applied sciences

MDPI

Article

UniMiB AAL: An Android Sensor Data Acquisition and Labeling Suite

Davide Ginelli, Daniela Micucci *, Marco Mobilio and Paolo Napoletano

Department of Informatics, Systems and Communication, University of Milano-Bicocca, 20126 Milan, Italy; davide.ginelli@unimib.it (D.G.); marco.mobilio@disco.unimib.it (M.M.); paolo.napoletano@disco.unimib.it (P.N.)

* Correspondence: daniela.micucci@disco.unimib.it; Tel.: +39-02-6448-7866

Received: 16 June 2018; Accepted: 26 July 2018; Published: 31 July 2018

Abstract: In recent years, research on techniques to identify and classify activities of daily living (ADLs) has significantly grown. This is justified by the many application domains that benefit from the application of these techniques, which span from entertainment to health support. Usually, human activities are classified by analyzing signals that have been acquired from sensors. Inertial sensors are the most commonly employed, as they are not intrusive, are generally inexpensive and highly accurate, and are already available to the user because they are mounted on widely used devices such as fitness trackers, smartphones, and smartwatches. To be effective, classification techniques should be tested and trained with datasets of samples. However, the availability of publicly available datasets is limited. This implies that it is difficult to make comparative evaluations of the techniques and, in addition, that researchers are required to waste time developing ad hoc applications to sample and label data to be used for the validation of their technique. The aim of our work is to provide the scientific community with a suite of applications that eases both the acquisition of signals from sensors in a controlled environment and the labeling tasks required when building a dataset. The suite includes two Android applications that are able to adapt to both the running environment and the activities the subject wishes to execute. Because of its simplicity and the accuracy of the labeling process, our suite can increase the number of publicly available datasets.

Keywords: dataset; Android application; ADL recognition; falls detection

1. Introduction

In recent years, researchers have been increasingly interested in systems for human activity monitoring based on machine learning techniques. In particular, the scientific community is paying particular attention to the definition and experimentation of techniques that use inertial signals captured by wearable devices to automatically recognize activities of daily living (ADLs) [1] and/or promptly detect falls [2,3]. These techniques have proven to be effective in many application domains, such as physical activity recognition and the estimation of energy expenditure [4,5], the monitoring of Parkinson's disease development [6], and the early detection of dementia disease [7].

Very recently, these techniques have been embedded in smartphone applications and rely on data acquired by the hosted sensors [8]. For instance, the applications by Sposaro et al. [9], Tsinganos et al. [10], Pierleoni et al. [11], and Casilari et al. [12] promptly detect falls; the application by Reyes-Ortiz et al. [13] classifies ADLs; and the application by Rasheed [14] both detects falls and classifies ADLs.

The success of these techniques depends on the effectiveness of the machine learning based algorithm employed, which results in being penalized by at least two factors: (i) the lack of an adequate number of public datasets to evaluate the effectiveness; and (ii) the lack of tools to be used to record labeled datasets and thus increase the number of available datasets.

Despite the strong interest in publicly available datasets containing labeled inertial data, their number is still limited. Casilari et al. [15] and Micucci et al. [16] analyze the state of the arts and provide a nearly complete list of those currently available, which includes, for example, UniMiB SHAR [16], MobiAct [17], and UMAFall [18]. For this reason, many researchers experiment with their techniques using ad hoc built datasets that are rarely made publicly available [19–21]. This practice makes it difficult to compare in an objective way the several newly proposed techniques and implementations because of a lack of a common source of data [16,21–23].

Almost none of the applications used to collect data are made publicly available [24]. To the authors' knowledge, the only exception is that used in the Gravity project, which, however, records falls only [25]. Thus, we searched for applications that also record inertial data in well-established digital marketplaces, such as Android's Google Play Store [26], Apple's App Store [27], and Microsoft's Windows Phone App Store [28]. We decided to focus on this market because we were interested in smartphones and smartwatches, and thus we neglected all the applications, such as that presented in [29], that take into consideration wearable devices not equipped with a mobile operating system. In the marketplaces we explored, we found some applications for registration, but all of them presented some limitations, for example, the kind of sensors used as the data source (e.g., G-sensor Logger [30]), the lack of support for labeling (e.g., Sensor Data Logger [31]), and the possibility of recording only one type of activity at each registration (e.g., Sensor Log [32]).

In this article, we present UniMiB AAL (Acquisition and Labeling), an Android suite that provides support for the acquisition of signals from sensors and their labeling. The suite is available for download from the following address http://www.sal.disco.unimib.it/technologies/unimib-aal/ [33]. The suite can manage the sensors hosted by both an Android smartphone and, if present, a paired Wear OS smartwatch. The suite is composed by two applications: *acquisition*, which collects signals from the available sensors while a human subject is performing activities, and *labeling*, which allows the user to specify which type of activity he/she is carrying out, such as walking, running, or sitting. The two applications are designed to run on two different smartphones; thus the user can specify each time an activity performed and its beginning and end timings using a smartphone without having to handle the other smartphone that is acquiring the data. This solution allows signals and assigned labels as synchronous as possible with the exact activity carried out by the subject to be obtained. Indeed, the smartphone that records signals is not moved from its position when the subject specifies the beginning and the end of each activity with the labeling smartphone. The suite discovers, independently of the hosting device, the available sensors and all their settings. The definitions of the types of activities that the user is requested to perform are easily configurable, as their definitions are stored in external files. Moreover, UniMiB AAL also collects information related to the sensor types and sampling rates that can be used to perform any kind of normalization that may be required to standardize the data with other data acquired with different devices. Finally, in addition to the use for which the suite was designed, it can also be used as a tool to label data from wearable devices that are not directly connected with smartphones and that provide data from different kinds of sensors, including those related to health. Because these devices are widely used, this could allow a potential amount of labeled data to be obtained. This can be achieved by placing the device (smartphone or smartwatch) on which the acquisition application is running next to the wearable device. The data acquired by the two devices is then analyzed in order to identify time series that refer to the same activity. The labels captured by the labeling application are then assigned to the wearable device data on the basis of these analyses. If the wearable device can be synchronized with a reference time (e.g., through the Network Time Protocol (NTP) [34]), only the labeling application is required, because the labels can be assigned directly to the data acquired by analyzing the timestamps only.

The article is organized as follows. Section 2 discusses differences between some existing applications and the UniMiB AAL suite; Section 3 describes the method that guided the design and implementation of the suite; Section 4 describes the applications that are included in the UniMiB

AAL suite; Section 5 describes how signals and labels are stored and managed; Section 6 provides information about the usability tests we performed; finally, Section 7 sketches the conclusions.

2. Related Work

This section discusses some of the available mobile applications for data acquisition and labeling. The search was performed on Google Scholar and Google Play Store, on July 15, 2018, considering applications available from 2015 to the end of 2017 with the following keywords: "sensor recording", "sensor log", "activity recognition", and "sensor labeling activity". In the case of Google Scholar, we also added to these the keyword "Android".

We looked for applications that are able to collect and label signals from sensors embedded in a smartphone (and from a possibly paired wearable device), that provide the opportunity to select the sensors to be used during recording, and that may support the user by showing the sequence of activities he/she has to execute to be compliant with the design of a dataset collecting campaign.

Thus, in order to be considered, the applications had at least the following characteristics, which made them usable and complete: (i) compliancy with the human–computer interaction (HCI) guidelines [35] in order to guarantee both the applications's correct use and its correct use by the subjects; (ii) provision of the possibility to record signals from a wearable device in order to collect more signals or signals from different positions; (iii) a selectable pool of sensors to be used for recording in order to choose those most suitable for the current acquisition campaign; (iv) in the case of acquisition campaigns with a sequence of predefined activities that the subjects have to perform, help provided to the user proposing time after time the activity that he/she must carry out, in order to relieve the user from having to remember the protocol (i.e., the sequences of activities) of the acquisition campaign; and finally; (v) labeling and acquisition activities executed in different devices to obtain signals and assigned labels as synchronous as possible with the exact activity carried out by the subject. Executing the two activities on different devices allows the user not to have to move the device that acquires when he/she specifies the beginning and the end of each activity from its position relative to the device used for labeling.

We clearly did not consider applications that do not store the acquired signals, because these were useless for our scope. We considered only applications that make the acquired data available for offline processing related to the experimentation of activity recognition techniques. Moreover, we did not exclude from our research applications that record signals only, that is, applications that do not provide a built-in labeling feature.

At the end of the research, we found only seven applications that met the requirements. Table 1 summarizes their characteristics. We include the UniMiB AAL suite in the table in order to compare it with the applications we found.

Table 1 includes the following information:

- *App*: The name of the application and its reference (i.e., the URL (Uniform Resource Locator) used to download the application or a reference to the article presenting the application).
- *Google Play Store*: The availability of the application on Google Play Store.
- *Source code*: The availability of the source code of the application.
- *Multi-devices (Wear OS)*: Support for recording sensors mounted on external devices such as smartwatches equipped with Wear OS.
- *Multi-sensors*: The possibility of recording data acquired from a set of sensors that are not only the inertial sensors.
- *Sensor selection*: The possibility to choose which sensors to use to record data.
- *Labeling*: The possibility to apply labels to the recorded data.
- *Multi-devices (Acquisition and Labeling - Acq. & Lab.)*: The possibility to run the acquisition and the labeling activities on different devices.
- *List of activities*: The possibility of supporting the user with the list of activities he/she has to perform (i.e., the protocol defined by the acquisition campaign if defined).

- *Material Design Guidelines*: Indicates if the application follows the HCI guidelines in order to meet the requirements related to usability. In particular, because we considered Android applications only, we checked if the application followed the Material Design Guidelines [36], which are a set of usability guidelines for the Android environment.

Table 1. Sensor recording and labeling applications.

Application	Google Play Store	Source Code	Multi-Devices (Wear OS)	Multi-Sensors	Sensor Selection	Labeling	Multi-Devices (Acq. & Lab.)	List of Activities	Material Design Guidelines
Activity Recognition Training [37]	Yes	—	No	Yes	No	Yes	No	No	No
ExtraSensory [38]	No	Yes	Yes	Yes	—	Yes	No	No	No
G-Sensor Logger [30]	Yes	—	No	No	No	No	No	No	No
Sensor Data Collector [39]	Yes	—	Yes	Yes	Yes	Yes	No	No	No
Sensor Data Logger [31]	Yes	—	Yes	Yes	Yes	No	No	No	Yes
Sensor Log [32]	No	—	No	Yes	Yes	Yes	No	No	No
Sensor Sense [40]	Yes	—	No	Yes	Yes	No	No	No	Yes
UniMiB AAL [33]	Not yet	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

All columns are populated with *Yes*, *No*, or — if the corresponding characteristic was respectively present, not present, or unknown.

Some of the applications provided a very limited set of features. For example, G-Sensor Logger [30] collects accelerations only, and Sensor Sense [40] acquires signals from only one sensor at a time, which can be chosen by the user. On the other hand, there were also more advanced applications, such as Sensor Data Logger [31], which allows signals to be registered from more than one sensor at a time (chosen by the user) and which can be paired with a Wear OS device.

Apart from UniMiB AAL, only four applications allowed labeling of the acquired data: Sensor Data Collector [39], Activity Recognition Training [37], ExtraSensory [38], and Sensor Log [32].

Sensor Data Collector and Activity Recognition Training are both available on Google Play Store.

Sensor Data Collector allows the user to specify the activity he/she is carrying out and his/her environment (e.g., home or street). The application saves the data in an SQLite database, which is stored in a non-public folder of the device.

Activity Recognition Training is an application designed to generate datasets that can be used for activity recognition. The application allows the user to specify the beginning and the end of each activity. The data are saved in a single text file stored in a public folder of the device. Each row in the file is a pair: the sensor type and the corresponding acquired signal.

ExtraSensory and Sensor Log are not available on Google Play Store, despite that the latter was present in the store until at least January 2018.

ExtraSensory is an open-source project, and the source code for both the server-side and the Android application is available. The application was designed in order to build the ExtraSensory Dataset [38] and was then released to ease further data collection. The application combines unsupervised labeling (exploiting classifiers trained with the ExtraSensory Dataset) and self-reported labels by the user, both before the actions and as a post-activity inquiry.

Sensor Log was developed to ease the process of collecting and labeling sensory data from smartphones. The recorded signals are stored in an SQLite database and can be exported in CSV (Comma-Separated Values) format.

All four applications supporting labeling, in addition to supporting the acquisition of signals from inertial sensors, also allow data to be registered from other built-in sensors, such as temperature and pressure. Of these four applications, only ExtraSensory and Sensor Data Collector allow signals to be recorded from the sensors of both the smartphone and the possible paired Wear OS smartwatch; only Sensor Data Collector and Sensor Log allow the user to select the sensors to be used for acquisition.

However, even if all four of these applications allow signals to be labeled, they are lacking in terms of satisfying some of the characteristics that we list at the beginning of this section, which are, on the contrary, fulfilled by UniMiB AAL. These characteristics are detailed in the following:

- **Multi-Devices (Acq. & Lab.):** The acquisition and labeling of the data happen on the same device. In the case of Sensor Log, Sensor Data Collector, and Activity Recognition Training,

these happen during the acquisition, thus introducing some incorrect labels in the dataset, as explained in Section 3.1. ExtraSensory relies on its own oracle to label data, which may not be accurate, and improves its accuracy by asking the user about the performed activities. While this gives better knowledge of the acquired data, there are inherent synchronisation issues between the offline labeling and the data.

- **Activities List:** None of the four applications support a predefined list of activities (i.e., protocols) in order to ease the data acquisition. Each time the user wishes to perform an activity, he/she has to choose the activity from the list of configured activities (or give feedback about it afterwards, in ExtraSensory); moreover, each activity is performed and recorded separately, thus removing from the acquired dataset the transitions between the activities, which may decrease performance once the online recognition takes place.
- **Material Design Guidelines:** Developing an end-user application that follows the HCI guidelines increases the end-user usage. On the opposite side, the user will experience frustration that will lead to him/her refraining from using the application [41]. Furthermore, depending on the marketplaces, users are familiar with specific guidelines (e.g., Material Design [36] for Android, and Human Interface Guidelines [42] for iOS). Therefore, developers should ensure consistency with these and not bore the user with new ways of interacting for common features.
  Despite this, none of the four applications follow the Material Design Guidelines, hence not guaranteeing usability. For example, Sensor Data Collector does not provide floating action buttons [43] (which represent the primary action of a screen) to control the application with a single action; it forces the user to leave the main screen and move to a settings panel to start the recording, and it fails in providing correct feedback to the user. In contrast, Sensor Sense follows the Material Design principles, for example, by providing a floating action button to start the recording and by using cards [44] (which contain content and actions about a single subject) to present a list of elements.

3. Method

The aim of our work is to provide the scientific community with a system capable of recording data from smartphone sensors (and the possibly paired smartwatch) to be easily used in evaluating and training machine learning techniques. The evaluation and training of these techniques require the availability of datasets containing signals recorded by sensors while subjects are carrying out activities, such as walking, sitting, or running. The subjects range from children to the elderly, with a scarce or marked attitude toward the use of computer technologies. Signals can range from inertial sensors data (e.g., accelerometers, gyroscopes, and compasses) to any kind of physiological or environmental sensor (e.g., pressure, temperature, and heart rate). In addition, these signals may come from sensors hosted in smartphones and/or smartwatches that differ from each other in terms of characteristics (e.g., precision, acquisition rate, and value ranges) depending on their manufacturer. Such signals must be labeled with the true activities carried out by the subjects, who can autonomously choose which activity to carry out and for how long or follow a protocol (i.e., a fixed sequence of actions and relative durations) defined by the researchers.

Thus, the definition of such datasets requires systems that are easy to use even by inexperienced users and that are easily configurable. To this aim, simple system configurations (e.g., a list of smartphone and smartwatch positions) should be possible without the applications having to be recompiled and deployed to the device. The data acquisition component must be designed in such a way that it does not depend on the physical characteristics of specific sensors. This allows a portable application that automatically adapts to the actual execution context to be obtained. The data labeling should be as consistent as possible to the activity the user is performing. Finally, the data and their labels should be accompanied by a set of additional information to be used in the technique evaluation. Such information may include, for example, the sensor type, position, and acquisition frequency.

An acquisition and labeling system must therefore address the following requirements: *labeling accuracy:* the data must be labeled according to the corresponding performed activity; *usability:* the system must be easily used to avoid mistakes and must be easily configured without the presence of a programmer; *self-adaptability:* the system must be able to be used with different smartphones, smartwatches, and sensors; *completeness:* the system must acquire and provide additional information to improve the dataset usage in the experimentations. Such requirements result in the following guidelines for the design of the system: *data reliability*, *configurability*, *user friendliness*, *hardware adaptability*, and *data meaningfulness*. Figure 1 sketches the identified requirements, the corresponding guidelines, and the design and implementation choices that fulfil them. The following subsections detail the guidelines and the identified design and implementation solutions.

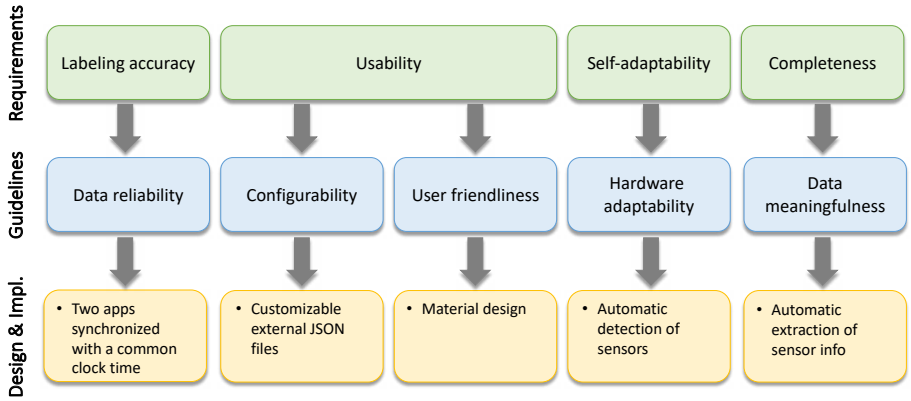


Figure 1. From requirements to design choices.

3.1. Data Reliability

The two main functionalities that the system must provide are *data acquisition* and support for *data labeling*. *Data acquisition* includes interfacing with the sensors embedded in the smartphone, managing the connection with the paired smartwatch, and storing the data acquired by the sensors. *Data labeling* includes the user interface setting the beginning and end times of a given activity performed by a human subject.

These two functionalities may result in a single application running on a smartphone. Using only one application would result in the following usage scenario: The user uses the application's user interface to specify the activity he/she will perform and then starts the registration and puts the smartphone in the designated place, for example, in her trouser pocket. When the user completes the activity, he/she picks up the smartphone from his/her trouser pocket and then stops recording. This usage scenario has a significant drawback: the data recorded from when the user starts recording to when he/she puts the smartphone in her pocket and, vice versa, the data recorded from when he/she picks up the smartphone from her pocket to when he/she stops recording are incorrectly labeled. These data are in fact labeled as belonging to the activity performed. To overcome this problem, we decided to operate a separation of concerns by assigning the responsibility of data acquisition and data labeling to two different software components, each of them reified by a different application. The idea is that these two applications should be executed in two different smartphones. In this way, the user exploits the acquisition application to start the recording of sensors. Once started, the user may place the smartphone in the designated location, for example, his/her trouser pocket. Then, the user exploits the labeling application running on a second smartphone to specify from time to time the activity that is being performed.

Two key aspects need to be considered to ensure that the process of recording and labeling data is properly implemented: Firstly, the two smartphones hosting the acquisition and labeling applications

should be synchronized in accordance with a common clock time; secondly, the data recorded by the sensors and the activities carried out by the subjects should have a unique subject identifier.

Clock synchronization between the two applications allows acquired data to be easily matched with the beginning and end time labels of the corresponding activity. For the implementation, we decided to use the `currentTimeMillis()` [45] method of the `System` class provided by the Java SDK (Java Software Development Kit), which allows the current time to be obtained in milliseconds. For the correct synchronization of the two smartphones, it is necessary that their wall-clock times are set by the same NTP server, so that the wall-clock times are the same and the `currentTimeMillis()` method returns consistent values from the two devices.

Concerning the identifiers, users are required to input their unique identifiers each time they start a new acquisition and labeling session. Identifiers along with timestamps are then used to store recordings as described in Section 5.

Clearly, the two applications can run on the same smartphone, but this use case would cause incorrect labels to be assigned to the recorded data.

3.2. Configurability

The application should be easily configured by people who are not programmers of mobile applications. This facilitates the use of the application by researchers mainly interested in developing and experimenting with new machine learning techniques for human activity monitoring.

When performing an activity or a set of activities, the subject has to specify both the kind of activity he/she will execute and where the smartphone and/or smartwatch will be placed during the execution of the activity. It has been demonstrated that the information regarding the position of the recording device (e.g., trouser pocket, wrist, or waist) is exploited in the machine learning literature to study the robustness of recognition methods to changes in device position [46].

The type of activity and device position are chosen from predefined lists and are not inserted by hand. This permits the user's interaction with the application to be simplified and, more importantly, the list of possible activities allowed during the recording session to be predefined. Thus, we have defined two sets of lists: one set contains the lists related to the positions, and the other set contains the lists related to the activities.

The set related to positions includes two lists: one list contains the positions where the smartphone can be placed, and the other contains the positions where the smartwatch can be placed.

The set related to activities also contains two lists, one containing a simple list of activities (e.g., walking, running, and sitting), and the other containing a list of predefined protocols. A protocol is an ordered sequence of activities and their durations, for example, running for 5 min, then jumping for 1 min, then standing for 2 min, and so on. Indeed, we identified two distinct ways in which the subject could carry out the activities: by selecting the activity to be performed from a list (i.e., operating a *free choice* among the list of activities) or by following a protocol (i.e., a *predefined list* of activities and durations). The former offers to the subject the possibility to choose which activities he/she wishes to carry out and for how long. On the contrary, the latter does not allow the subject to change the order in which the activities have to be carried out or their duration.

The four configuration lists are turned into four corresponding JSON (JavaScript Object Notation) files: the smartphone positions' list, the smartwatch positions' list, the list of activities for the free choice mode, and the list of protocols in the predefined list mode.

Listings 1 and 2 show examples for the smartphone and smartwatch positions' lists, respectively.

Listing 3 shows an example list related to the free choice of activities, while Listing 4 shows an example list containing three protocols.

Because the positions are related to the acquisition devices, we decided to assign the functionality regarding the choice of positions to the acquisition application. Because activities are used to correctly label signals, we decided to assign the functionality regarding the choice of the activity or the protocol to the labeling application. It follows that the JSON files related to positions are located in the

smartphone running the acquisition application, while the JSON files related to the activities are located in the smartphone running the labeling application.

Listing 1. Example of the configuration file that includes the list of the smartphone positions.

```
{
  "positions": [
    "Bag",
    "Belt waist/hip (left-front)",
    "Belt waist/hip (right-front)",
    "Belt waist/hip (center-front)",
    "Belt waist/hip (left-back)",
    "Belt waist/hip (right-back)",
    "Belt waist/hip (center-back)",
    "Shirt pocket (left-front)",
    "Shirt pocket (right-front)",
    "Trouser pocket (left-front)",
    "Trouser pocket (right-front)",
    "Trouser pocket (left-back)",
    "Trouser pocket (right-back)"
  ]
}
```

Listing 2. Example of the configuration file that includes the list of smartwatch positions.

```
{
  "positions": [
    "Wrist (left)",
    "Wrist (right)"
  ]
}
```

Listing 3. Example of the file that includes the list of activities from which the user makes their selection.

```
{
  "free_choice_activities": [
    "Bicycling",
    "Going downstairs",
    "Going upstairs",
    "Jumping",
    "Laying",
    "Running",
    "Sitting",
    "Sitting down",
    "Standing",
    "Standing up",
    "Walking"
  ]
}
```

To avoid having to compile the application every time these files are modified, the JSON files are located in a public folder of the smartphone called `configuration` in both the acquisition and labeling applications. Because these files are in public storage, it is possible to access them in an easy way either by connecting the smartphone to a computer or directly from the mobile device. This allows some positions and/or activities or protocols to be changed, added, or removed in accordance with the needs. Every change to these four files is automatically reflected in the acquisition and labeling application, and thus there is no need to either rebuild the application or make a new installation.

The `configuration` folder of the acquisition application is placed in the `AAL - Acquisition` folder. On the left of Figure 2 is shown the folder hierarchy and the two JSON files for the device positions. The `configuration` folder of the labeling application is located in the `AAL - Labeling` folder of the labeling application. On the right of the Figure 2 is shown the folder hierarchy and the two JSON files for the activities.

Listing 4. Example of the file that includes the list of protocols from which the user makes a selection.

```
{
  "protocols": [
    {
      "name": "Relax",
      "description": "This protocol ...",
      "exercise": [
        {
          "description": "Walking",
          "duration": 300
        },
        {
          "description": "Sitting",
          "duration": 60
        },
        {
           "description": "Standing",
           "duration": 120
        }
      ]
    },
    {
      "name": "Soft Sport",
      "description": "This protocol ...",
      "exercise": [
        {
          "description": "Running",
          "duration": 180
        },
        {
          "description": "Jumping",
          "duration": 120
        }
      ]
    },
    {
      "name": "Hard Sport",
      ...
    }
  ]
}
```

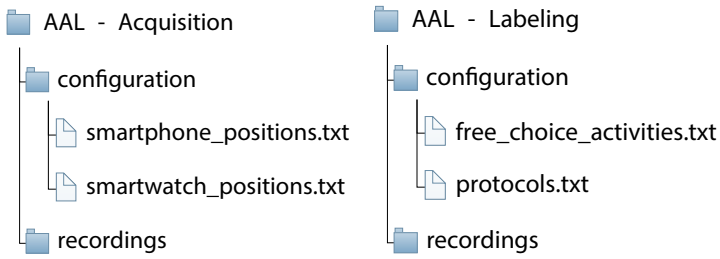


Figure 2. (**Left**) The folder organization in the acquisition application. (**Right**) The folder organization in the labeling application.

3.3. User Friendliness

The application must have a pleasant, intuitive, and clean user interface in order to promote its use and to minimize errors in its use.

To fulfil this aim, we followed two directions. On one hand, we analyzed the layouts of other applications with similar functionalities. In this way, we were able to identify the positive and negative aspects of every application according to our point of view. On the other hand, we followed the guidelines defined in the Material Design [36] specification, which is the way Google recommends Android applications be developed. The Material Design Guidelines, as well as providing recommendations about the patterns to follow to create a well-designed user interface, also gives advice on the basis of accessibility studies.

Taking into consideration the positive aspects of the analyzed applications and the indications provided by the Material Design Guidelines, we obtained a user interface that provides a good user experience and that is consistent with the operating system and other Android applications. These aspects allow the subject not to be disoriented when he/she uses both applications and to think only about carrying out the activities and recording the data.

Section 4 provides some screenshots of both of the applications to show their adherence to the Material Design Guidelines.

3.4. Hardware Adaptability

Ensuring hardware adaptability requires that the acquisition application should not embed a list of predefined sensors to interface with but should be able to discover those that are available in the target smartphone and in the paired smartwatch. If available in the smartwatch, the automatically populated list should include not only classical inertial sensors (e.g., accelerometer, gyroscope, and compass), but also different sensors for the monitoring of, for example, the ambient temperature, ambient level of humidity, heart rate, and others.

To achieve this goal, we used the Android API. In particular we exploited the `SensorManager` class that provides the `getSensorList (int type)` [47] method, through which it is possible to obtain a list of the available sensors in a device. In this way, the acquisition application can adapt to the smartphone in which it is executed and offers the possibility to choose all the sensors or some of them. Moreover, if the smartphone is paired with a smartwatch, the application can also adapt to the smartwatch, thus also making it possible to select the sensors available in the smartwatch itself, increasing the information associated to the activities carried out by the subject.

This ensures extreme flexibility in the choice of the data type that can be recorded. Moreover, the subject can select only the sensors that are hosted by the smartphone or the smartwatch.

3.5. Data Meaningfulness

Signals acquired by the sensors can be used without any additional information. However, the knowledge of information about the devices (smartphone and smartwatch) and the sensors used to acquire the signals may help in improving the data processing.

Concerning the devices, additional information may include, for example, the version of the operating system, the manufacturer, and the model of the device. Concerning the sensors, the manufacturer and the model can be very useful to search for the technical datasheet of the sensor. In addition, operational information of the sensors, such as the resolution, the delay, and the acquisition rate, can be very useful for processing purposes.

As for hardware adaptability, to achieve this goal, we used the Android API (Application Programming Interface). In particular, we exploited the `Build` class [48], which makes available a set of constant variables whose values are initialized according to the device in which the application is running. Such constants include, for example, the manufacturer, the brand, the version release, and much other information about the device. In this way, the application can automatically retrieve such information each time a new acquisition is started.

On the other side, we exploited the `Sensor` class [49], which represents a sensor. The class is initialized with the specific characteristics of the represented sensor that can be accessed by means of ad hoc methods. For example, the method `getMaximumRange()` makes available the maximum range of the sensor in the sensor's unit, and the method `getMinDelay()` makes available the minimum delay allowed between two events in microseconds.

As for the `Build` class, the `Sensor` class automatically adapts the status of its instances according to the sensor the instance is representing. Thus, the application can automatically retrieve sensors' information each time a new acquisition is started, as the corresponding sensor instances are initialized accordingly.

4. UniMiB AAL Suite

This section describes the functionalities of the UniMiB AAL suite and provides some screenshots captured during its execution.

The two main functionalities are assigned to two different applications: *data acquisition* and *data labeling*. As discussed in Section 3, this architectural choice was made in order to obtain signals with labels that are as correct as possible, that is, signals with labels that reflect the exact activity carried out by the subject. With two separate applications, the subject can start the recording application on a smartphone, which is then placed in its designated position and is not moved until he/she decides to stop recording. Once the subject has placed the smartphone with the acquisition application running, he/she starts the labeling application on another smartphone and uses the application to label signals (i.e., he/she specifies the activity, its beginning time, and its end time). This usage scenario is sketched in Figure 3. Timestamped signals recorded by the acquisition application can be easily coupled to the labels assigned by the subject through the labeling application because of the fact that the clocks of the two smartphones are synchronized. However, the fact that the suite includes two applications does not preclude it from running on a single smartphone.

4.1. The Acquisition Application

When the subject runs the application, he/she is asked to specify his/her ID, the devices to use in the acquisition phase, and the position of the devices (see Figure 4a).

The ID of the subject is any alphanumeric string that is used to link signals, labels, and the identity of the subject and that is maintained externally from the suite. We imagine that a subject who plans an acquisition session will store in their own file (e.g., a CSV file) information about each aspect, such as, for example, age, weight, height, lifestyle, and any other information that can be useful for training and testing machine learning techniques.

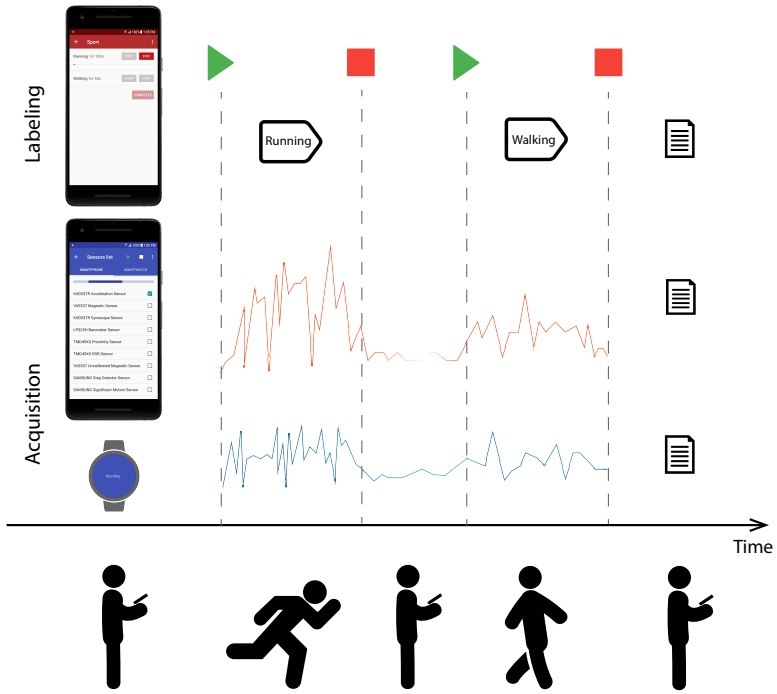


Figure 3. The UniMiB AAL (Acquisition and Labeling) suite.

Concerning the selection of the devices and their positions, the subject selects these from the list (see Figure 2) placed in public storage as specified in Section 3.2. In Figure 4a, it is possible to see how Listing 1 is rendered by the application. The subject is then asked to select the sensors for recording signals among those available in the previously selected devices (see Figure 4b). The sensors presented to the subject are those that have been obtained by asking the available sensors using the Android API as discussed in Section 3.5. In case both the smartphone and the smartwatch are selected, a tab layout allows the user to switch from the smartphone to the smartwatch list of the available sensors to select those to be used.

The subject can then start the acquisition of signals from the selected sensors by pressing the `play` button on the top of the activity, as shown in Figure 4b. Once the recording has been started, an indeterminate progress bar informs the subject that the signals are being captured (Figure 4c). At this time, the subject should place the devices in the position that he/she has specified in the configuration phase (see Figure 4a).

If the application cannot acquire signals from the sensors, it notifies the subject of the problem by emitting a sound signal.

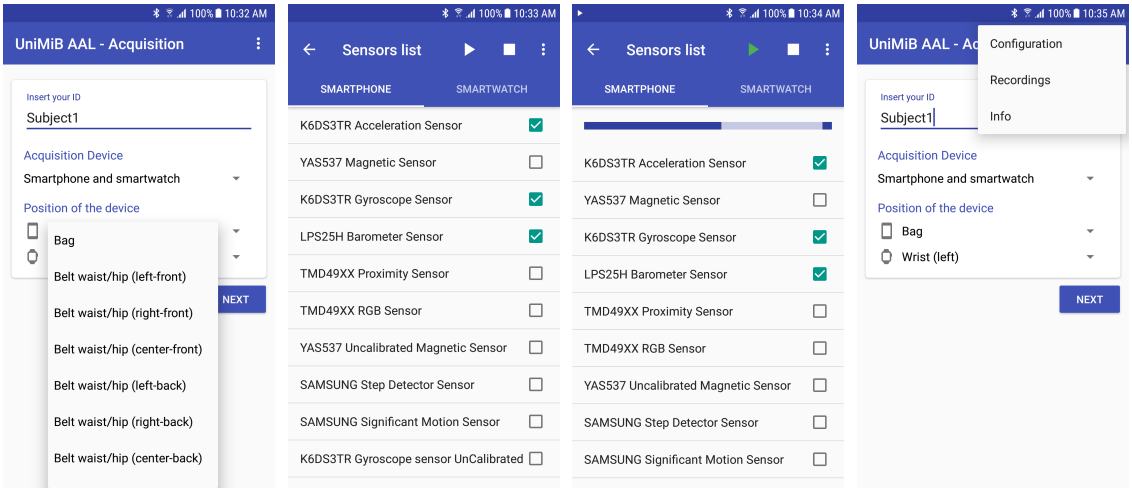
Finally, once the subject has stopped recording, he/she can perform another acquisition session by changing the configuration if necessary (e.g., using only a smartphone and selecting its accelerometer sensor), or the subject can access the recordings folder to retrieve the files containing all the data related to the acquisition (see Figure 4d).

4.2. The Labeling Application

When the subject starts the application, he/she is asked to specify her ID, which must be equal to that provided in the acquisition application in order to associate the files generated by the acquisition application to the corresponding files generated by the labeling application (see Figure 5a).

In the same configuration screen, the subject also selects the acquisition mode. As introduced in Section 3.2, the suite supports two kind of acquisition modes: *free choice* and *predefined list*. The free choice acquisition mode allows the subject to select the time of each activity he/she wants to carry out, for example, walking, standing, and sitting down. The predefined list acquisition mode constraints the

subject to carry out a predefined sequence of activities, for example, walking for 300 s, running for 180 s, and stretching for 60 s.



(**a**) (**b**) (**c**) (**d**)

Figure 4. Screenshots of the acquisition application.

The labeling application reifies the free choice acquisition mode by showing the subject the list of activities from which he/she selects the activity he/she wants to carry out and for how long (see Figure 5b). The list shown in Figure 5b was populated by processing the JSON file in Listing 3. Once the subject has selected the activity (e.g., Running), he/she can signal its beginning by pressing the `start` button (see Figure 5c). In this way, the application acquires the timestamp for which the subject starts the execution of the selected activity. When the subject decides to stop the execution of the activity, he/she presses the `stop` button, so that the labeling application can acquire the timestamp for the end of the activity. As shown in Figure 5d, the labeling application shows the subject the list of activities he/she has carried out and their durations.

With regard to the predefined list acquisition mode, the labeling application shows the subject the list of available protocols, as shown in Figure 5e. The list is populated by processing the JSON file in Listing 4. Once the subject chooses the protocol, the labeling application shows the list of activities defined in the selected protocol and their durations (see Figure 5f). This list is also populated by means of the information in Listing 4. The subject starts each activity by pressing the `start` button, which signals to the application the beginning time of the current activity. Automatically, when the time for the current activity elapses, it saves the end time of the just-completed activity. Figure 5g shows the execution of the Relax protocol.

Finally, once the subject has finished the execution of the last activity defined in the selected protocol, he/she can perform another labeling campaign, changing the configuration if necessary (e.g., changing the protocol or selecting the free choice acquisition mode), or the subject can access the recordings folder to retrieve the files containing all the data related to the labeling (see Figure 5h).

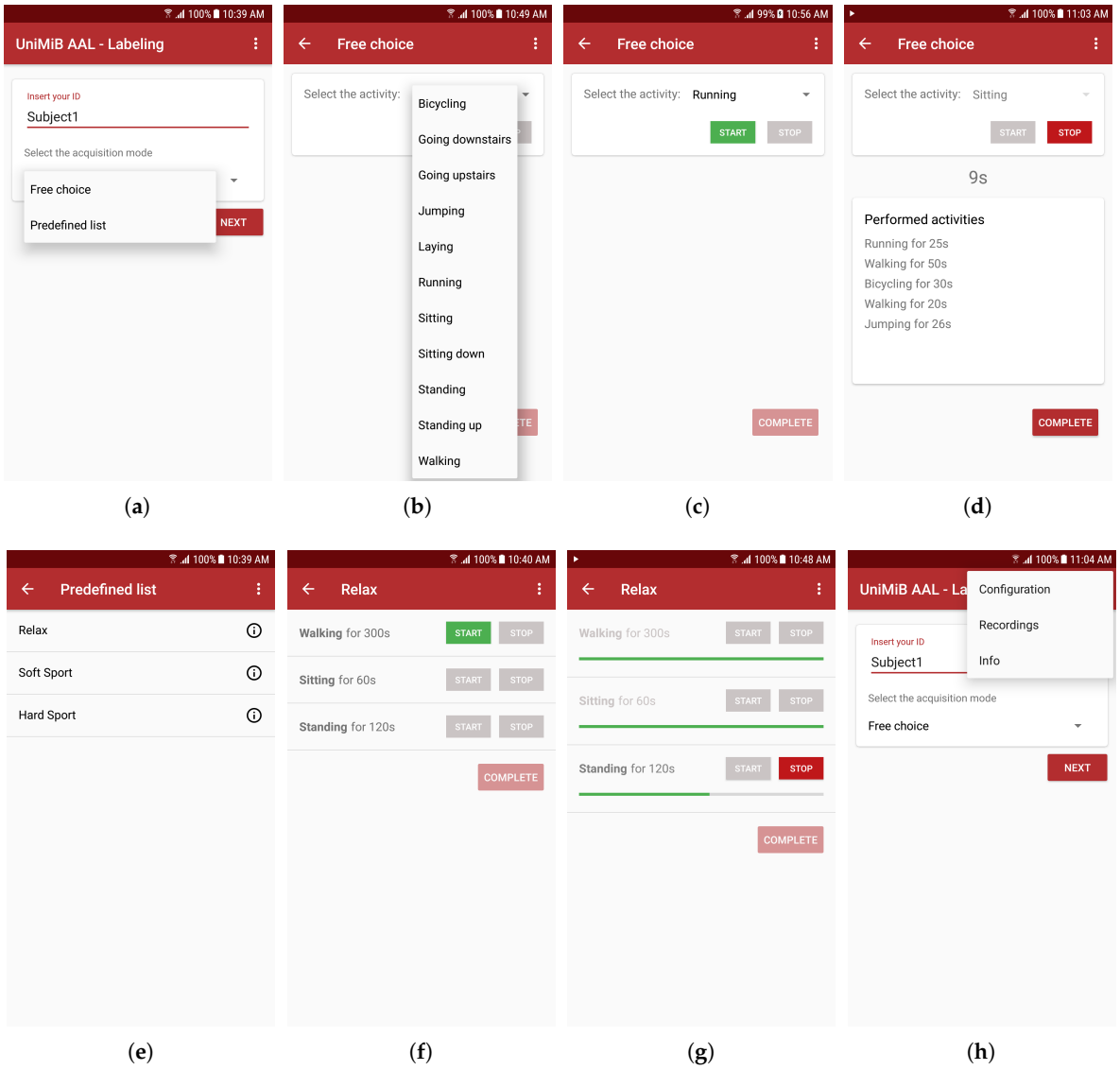
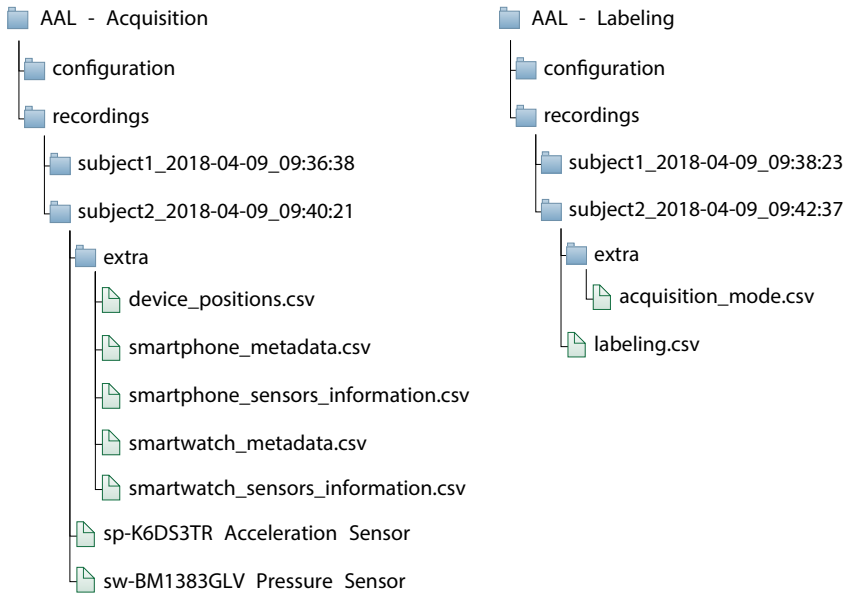


Figure 5. Screenshots of the labeling application.

5. The Recordings

This section details how recordings generated by the two applications are organized and how they can be accessed. As introduced in Section 3.2, both the applications have a main folder in the public storage of the smartphone: `AAL - Acquisition` for the application that acquires the signals from the sensors, and `AAL - Labeling` for the application that labels these data. Because these files are in public storage, a user can access them in an easy way either by connecting the smartphone to a computer or directly from the mobile device.

As shown in Figure 6, both the `AAL - Acquisition` folder and `AAL - Labeling` folder contain two subfolders: `configuration` and `recordings`. The `configuration` folder is discussed in Section 3.2 and contains the files that a user can change in order to adapt the applications according to researchers' needs, while the `recordings` folder contains the files with the signals and the labels recorded.



(**a**) Acquisition (**b**) Labeling

Figure 6. The folder organization in the acquisition application (**a**), and the folder organization in the labeling application (**b**).

Each time a subject starts a recording session, the acquisition application creates a new folder whose name is obtained by concatenating the *id* (identifier) of the subject and the *date and the time* of the beginning of the recording. For example, the folder `subject2_2018-04-09_09:40:21` in Figure 6a specifies that the recordings are related to a subject with ID "subject2" who recorded a session on April 9, 2018 at 9 am for 40 min and 21 s. Each folder related to an acquisition session performed by a subject contains as many CSV files as the number of sensors that have been selected for the recordings, each of them containing the acquired signals. The name of each file is characterized by an initial string that specifies if the sensor belongs to the smartphone (`sp`) or to the smartwatch (`sw`) and is followed by the name of the sensor as provided by the Android API. The files containing signals from smartwatch sensors are transferred to the smartphone via Bluetooth. For example, the file `sp-K6DS3TR Acceleration Sensor` in Figure 6a contains the acceleration signals acquired by the sensor identified as K6DS3TR hosted in the smartphone.

Similarly, the `recordings` folder of the labeling application contains the data related to the activities performed during application runs. Each time a subject starts to label his/her first activity, the labeling application creates a new folder following the same pattern of the acquisition application. For example, the folder `subject2_2018-04-09_09:42:37` in Figure 6b specifies that the labels are related to a subject with ID "subject2" who recorded a session on April 9, 2018 at 9 am for 42 min and 37 s. Each folder related to a labeling session contains a CSV file named `labeling` that contains the labels of the activities that the subject has performed (see Figure 6b).

The subfolders of the acquisition application `recordings` folder can be associated to the corresponding subfolders of the labeling application `recordings` folder because of the names assigned to the subfolders. That is, given a `recordings` subfolder in the acquisition application, the associated subfolder in the labeling application is that which has the same identifier for the subject and a time that is the closest to that of the acquisition subfolder. For example, the subfolder `subject2_2018-04-09_09:40:21` created by the acquisition application and the subfolder `subject2_2018-04-09_09:42:37` created by the labeling application can be associated because they have the same identifier and because the date of the subfolder in the labeling application is the closest to that of the folder in the acquisition application.

Figure 7a,b show respectively the fragments of the `sp-K6DS3TR Acceleration Sensor` and `sw-BM1383GLV Pressure Sensor` files stored in the `subject2_2018-04-09_09:40:21` folder. The first column indicates the acquisition time in milliseconds, the following columns contain the acquired values (x, y, and z for the acceleration sensor and the value of the pressure for the pressure sensor), and the last column specifies the accuracy of the values as returned by the Android API. In particular, the level of accuracy can assume one of five predefined values: -1 when the values returned by the sensor cannot be trusted because the sensor had no contact with what it was measuring (e.g., the heart rate monitor is not in contact with the user), 0 when the values returned by the sensor cannot be trusted because calibration is needed or the environment does not allow readings (e.g., the humidity in the environment is too high), 1 when the accuracy is low, 2 when the accuracy is medium, and 3 when the accuracy is maximum [47].

timestamp	x	y	z	accuracy
		...		
1523406183006,00	41.397.014	-11.253.091	8.842.057	3
1523406183025,00	41.325.183	-11.205.206	8.822.903	3
1523406183025,00	41.349.125	-11.468.576	8.820.508	3
1523406183025,00	4.048.719	-11.420.691	879.896	3
		...		
1523406188009,00	4.312.089	-11.013.664	8.411.087	3
1523406188009,00	4.314.483	-11.013.664	8.411.087	3
1523406188009,00	4.312.089	-10.941.836	8.415.876	3
1523406188015,00	4.336.032	-10.870.007	8.432.636	3
		...		

(**a**) Acceleration values.

timestamp	value	accuracy
	...	
1523406182994,00	97.986.914	3
1523406183042,00	979.791	3
1523406183094,00	97.974.023	3
1523406183142,00	97.974.414	3
	...	
1523406187883,00	9.798.047	3
1523406187931,00	97.972.656	3
1523406187982,00	9.798.164	3
1523406188030,00	9.798.262	3
	...	

(**b**) Pressure values.

Start	Stop	Activity
1523406171772,00	1523406179794,00	Walking
1523406183009,00	1523406188011,00	Sitting
1523406190160,00	1523406196183,00	Standing

(**c**) Labels.

Figure 7. The signals that were stored by the acquisition application and acquired by the acceleration sensor K6DS3TR (**a**) and by the pressure sensor BM1383GLS (**b**), and the corresponding labels that were stored by the labeling application (**c**).

Figure 7c shows the content of the paired `labeling` file (contained in the `subject2_2018-04-09_09:42:37` folder). The first and second columns respectively specify the beginning and the end of the activity, which are labeled in the third column. Red bounding boxes in Figure 7a,b indicate the signals that are labeled with "sitting" as they have compatible timestamps.

At each recording session, both the acquisition and the labeling application store in their `extra` folder additional information that can be useful for data processing (see Figure 6). The acquisition application includes the following CSV files:

- `device_positions` contains the positions of the devices indicated by the subject.
- `smartphone_metadata` and `smartwatch_metadata` respectively contain information about the smartphone and the smartwatch used to perform recordings, such as, for example, the manufacturer, the model, and the Android version. This information is automatically retrieved from the smartphone and from the smartwatch by the applications.
- `smartphone_sensors_information` and `smartwatch_sensors_information` respectively contain information about the sensors of the smartphone and the smartwatch that have been used to record the signals. Besides the name of the sensor, its vendor, and other information that allows the sensor's detailed specifications to be determined, the files include other data such as the minimum delay (i.e., the minimum time interval in microseconds that a sensor can use to

sense data), the maximum range (i.e., the maximum range of the sensor in the sensor's unit), and the resolution of the sensor in the sensor's unit. Figure 8 shows the content of these files in the recording session sketched in Figure 7. The red bounding boxes highlight the sensors that generated the signals in Figure 7a,b.

The labeling application includes the `acquisition_mode` CSV file that specifies the acquisition mode (i.e., free choice or predefined list) selected by the subject. In the case that the predefined list mode has been selected, the labeling application also specifies the name of the selected protocol.

Name	Vendor	Version	Min Delay	Max Range	Resolution	Power
K6DS3TR Acceleration Sensor	STM	1	2000	784.532	0.0023942017	0.25
YAS537 Magnetic Sensor	Yamaha Corporation	1	10000	1200.0	0.1	6.0
K6DS3TR Gyroscope Sensor	STM	2	2000	34.906.586	0.0012217305	6.1
...						

(**a**) Accelerometer sensor information.

Name	Vendor	Version	Min Delay	Max Range	Resolution	Power
BMI160 3-axis Accelerometer	Bosch Sensortec	1	10000	392.266	0.009576807	0.23
BMI160 3-axis Gyroscope	Bosch Sensortec	1	10000	34.906.586	0.0012217305	6.1
BM1383GLV Pressure Sensor	ROHM Semiconductor	1	50000	1100.0	5.0E-4	0.005
...						

(**b**) Pressure sensor information.

Figure 8. The information about the sensors used for recording in the smartphone (**a**) and in the smartwatch (**b**).

6. Usability Tests

Usability tests were conducted at a quiet and controlled location within the Department of Informatics, Systems and Communication of the University of Milano-Bicocca. We involved 10 users who tested both the acquisition and the labeling applications by performing and labeling activities for 20 min. The users' ages were around 20, 40, 50, and 60 and were equally distributed, and for their genders, half were female and half were male. The users were trained in the use of both the applications by a moderator who was an expert user of the application. The usability test included four tasks that were conducted individually, that is, one user at a time. After the test, to obtain the users' impressions about the usability of the applications, the users completed a questionnaire. The questionnaire included questions related to the functionalities of the two applications, and the questions were inspired by the System Usability Scale (SUS) questionnaire [50].

The tasks involved in the tests are described in the following.

1. Use the acquisition application to acquire the data from the sensors of the smartphone only, and use of the labeling application to label the activities performed following a predefined list of activities. For this task, the moderator instructed the users to follow the predefined list of activities denoted as "Relax", which included walking for 300 s, sitting for 60 s, and standing for 120 s.
2. Use of the acquisition application to acquire the data from the sensors of both the smartphone and the smartwatch, and use of the labeling application to label the activities performed following a predefined list of activities. For this task, the moderator instructed the users to follow the predefined list of activities denoted as "Soft sport", which included running for 180 s and jumping for 120 s.
3. Use of the acquisition application to acquire the data from the sensors of the smartphone, and use of the labeling application to label the activities performed that were freely chosen from the free choice list. For this task, the moderator instructed the users to choose at most three activities from the free choice list and to perform them for at most 15 s. The free choice list included running, walking, sitting, and going downstairs.

4. Use of the acquisition application to acquire the data from the sensors of both the smartphone and the smartwatch, and use of the labeling application to label the activities performed that were freely chosen from the free choice list. For this task, the moderator gave the same instructions as in the third task above.

The users were also instructed to move between tasks and to do whatever they would naturally do. The moderator observed all the test sessions and annotated the errors and the problems that occurred and the time taken by each user to complete a task.

Table 2 reports the results of the questionnaire obtained by taking the median of all the users' votes. The first 10 statements refer to both the applications, while the 13th and 14th statements refer to the acquisition and labeling applications, respectively. Generally, the applications evaluated the applications as easy to use. The users employed a very short time to start using the applications correctly, and they declared the use of the applications' interface to be very pleasant. Some users found the action of selecting the sensors to be recorded somewhat difficult when both the smartphone and the smartwatch were selected. For this, additional instructions by the moderator were needed.

Table 2. Usability questionnaire (numerical scale: 1: "strongly disagree"; 2: "disagree"; 3: "neutral"; 4: "agree"; 5: "strongly agree").

No.	Statement	1	2	3	4	5
1.	Overall, I am satisfied with how easy it is to use the application.				•	
2.	I found this application unnecessarily complex.		•			
3.	I would need the support of a technical person to be able to use the application.		•			
4.	The interface of this application was pleasant.				•	
5.	I found the various functions in this application were well integrated.				•	
6.	I thought there was too much inconsistency in this application.	•				
7.	I would imagine that most people would learn to use the application very quickly.				•	
8.	I found the application very cumbersome/awkward to use.		•			
9.	I felt very confident using the application.				•	
10.	I needed to learn a lot of things before I could get going with this application.		•			
11.	Whenever I made a mistake using the application, I could recover easily and quickly.			•		
12.	I believe I could become productive quickly using this system.				•	
13.	I found the action of selecting the sensors to be recorded too complex.			•		
14.	I found the action of selecting the activity to be recorded too complex.	•				

7. Conclusions

The lack of available applications to acquire and label signals acquired from sensors and the small number of public datasets containing labeled signals force researchers to develop their own applications whenever they need to experiment with a new activity recognition technique based on machine learning algorithms.

In this article, we present UniMiB AAL, an application suite for Android-based devices that eases the acquisition of signals from sensors and their labeling. The suite is not yet loaded in Google Play

Store, but it is available for download from the following address: http://www.sal.disco.unimib.it/technologies/unimib-aal/ [33]. The suite is composed by two applications (acquisition and labeling) in order to be executed on two different smartphones. This may reduce the number of signals erroneously labeled, as the smartphone acquiring signals is never moved from its position during the labeling task. Moreover, assigning labeling and acquisition to two separated and dedicated applications also becomes an advantage when an operator assists the subject during recording activities. In this scenario, having a second smartphone adds the possibility to have the operator handling the labeling while the subjects only have to perform the experiment. Indeed, our empirical experience when collecting the UniMiB SHAR [16] dataset suggested that subjects have the tendency to be less precise and make more errors than the designers of the experiments. Even if the configuration that exploits two smartphones helps in achieving correctly labeled signals, the two applications can be run on one smartphone only. This can be useful whenever the subject is unable to manage two smartphones.

Both the applications feature a simple and clear interface and can acquire data from a variety of sensors. The acquisition application automatically discovers the sensors hosted by the devices so that it can adapt to the true running environment. Moreover, the configuration information, including the list of activities or the positions of the devices, is stored in external accessible files. This allows the researches to personalize the configurations according to their needs, without having to recompile the applications.

In order to validate the usability, we conducted a number of tests involving 10 users equally distributed in terms of age and gender. The tests showed that a very short time was needed to understand how to use the applications and start using them.

The application scenarios of the proposed suite are numerous and mostly focus on human monitoring (see Section 2). The idea of implementing a suite for recording and labeling signals from smartphone and smartwatch sensors arose because of the difficulties we encountered in acquiring a dataset of accelerometer signals related to human activities and falls [16]. This dataset, as well as others in the state of the arts, are used to develop machine learning algorithms that are then able to predict whether a person's activity is a race or a walk. To obtain robust machine learning algorithms, a huge amount of data is needed, and more importantly, data needs to be correctly labeled. The manual work for data labeling would be tedious and noise-prone. One of the possible application scenarios of the proposed suite would be the following: needing to acquire signals from the accelerometer and the gyroscope of a smartphone. The idea is to record signals related to n activities (walking, running, etc.) performed by a group of m humans. Each activity is performed following a given story board: walking for 5 min, running for 3 min, and so on. Each person performing the activity holds the smartphone for data recording in his/her pocket and the smartphone for data labeling in his/her hands. Using the proposed labeling suite running on the latter smartphone, the human is able to easily label the activities that he/she is performing.

Future developments will include a simplification of the process of selecting sensors when both the smartphone and the smartwatch have been selected, as suggested from the tests. We plan to substitute the current tab layout with a simple activity that includes two buttons: one button is used to show a popup from which the subject can select the smartphone's sensors; the other button is used to show a popup from which the subject can select the smartwatch's sensors. Moreover, we plan to include in both the applications additional functionality that will allow the researchers/users to define the content of the configuration files so that they do not have to edit files in JSON format.

Finally, we will explore the possibility to integrate voice recognition as a way to ease the process of starting, stopping, and labeling activities. This improvement will release the user of the burden of manually signaling the beginning of each activity in the predefined list operation mode and of manually selecting and signaling the beginning and end timings of each activity in the free choice operation mode.

Author Contributions: All authors contributed equally to the design and development of the mobile applications; all authors contributed to the writing, proofreading, and final approval of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors wish to thank Luca Lo Russo and Giuseppe Paris for their support in developing the applications.

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

References

1. Shoaib, M.; Bosch, S.; Incel, O.D.; Scholten, H.; Havinga, P.J. A Survey of Online Activity Recognition Using Mobile Phones. *Sensors* **2015**, *15*, 2059–2085. [CrossRef] [PubMed]
2. Casilari, E.; Luque, R.; Morón, M.J. Analysis of Android device-based solutions for fall detection. *Sensors* **2015**, *15*, 17827–17894. [CrossRef] [PubMed]
3. Micucci, D.; Mobilio, M.; Napoletano, P.; Tisato, F. Falls as anomalies? An experimental evaluation using smartphone accelerometer data. *J. Ambient Intell. Hum. Comput.* **2017**, *8*, 87–99. [CrossRef]
4. Shoaib, M.; Bosch, S.; Durmaz, I.O.; Scholten, H.; Havinga, P.J.M. Fusion of Smartphone Motion Sensors for Physical Activity Recognition. *Sensors* **2014**, *14*, 10146–10176. [CrossRef] [PubMed]
5. Hills, A.P.; Mokhtar, N.; Byrne, N.M. Assessment of Physical Activity and Energy Expenditure: An Overview of Objective Measures. *Front. Nutr.* **2014**, *1*, 5. [CrossRef] [PubMed]
6. Rovini, E.; Maremmani, C.; Cavallo, F. How Wearable Sensors Can Support Parkinson's Disease Diagnosis and Treatment: A Systematic Review. *Front. Neurosci.* **2017**, *11*, 555. [CrossRef] [PubMed]
7. Urwyler, P.; Stucki, R.; Rampa, L.; Müri, R.; Mosimann, U.P.; Nef, T. Cognitive impairment categorized in community-dwelling older adults with and without dementia using in-home sensors that recognise activities of daily living. *Sci. Rep.* **2017**, *7*, 42084. [CrossRef] [PubMed]
8. Su, X.; Tong, H.; Ji, P. Activity recognition with smartphone sensors. *Tsinghua Sci. Technol.* **2014**, *19*, 235–249.
9. Sposaro, F.; Tyson, G. iFall: An Android application for fall monitoring and response. In Proceedings of the 31th IEEE Annual International Conference of Engineering in Medicine and Biology Society (EMBC '09), Minneapolis, MN, USA, 3–6 September 2009.
10. Tsinganos, P.; Skodras, A. A smartphone-based fall detection system for the elderly. In Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis (ISPA '17), Ljubljana, Slovenia, 18–20 September 2017.
11. Pierleoni, P.; Pernini, L.; Belli, A.; Palma, L.; Valenti, S.; Paniccia, M. SVM-based fall detection method for elderly people using Android low-cost smartphones. In Proceedings of the 2015 IEEE Sensors Applications Symposium (SAS '15), Zadar, Croatia, 13–15 April 2015.
12. Casilari, E.; Oviedo-Jiménez, M.A. Automatic Fall Detection System Based on the Combined Use of a Smartphone and a Smartwatch. *PLoS ONE* **2015**, *10*, e0140929. [CrossRef] [PubMed]
13. Reyes-Ortiz, J.L.; Oneto, L.; Samà, A.; Parra, X.; Anguita, D. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* **2016**, *171*, 754–767. [CrossRef]
14. Rasheed, M.B.; Javaid, N.; Alghamdi, T.A.; Mukhtar, S.; Qasim, U.; Khan, Z.A.; Raja, M.H.B. Evaluation of Human Activity Recognition and Fall Detection Using Android Phone. In Proceedings of the 29th IEEE International Conference on Advanced Information Networking and Applications (AINA '15), Gwangju, Korea, 25–27 March 2015.
15. Casilari, E.; Santoyo-Ramón, J.A.; Cano-García, J.M. Analysis of public datasets for wearable fall detection systems. *Sensors* **2017**, *17*, 1513. [CrossRef] [PubMed]
16. Micucci, D.; Mobilio, M.; Napoletano, P. UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones. *Appl. Sci.* **2017**, *7*, 1101. [CrossRef]
17. Chatzaki, C.; Pediaditis, M.; Vavoulas, G.; Tsiknakis, M. Human Daily Activity and Fall Recognition Using a Smartphone's Acceleration Sensor. In Proceedings of the International Conference on Information and Communication Technologies for Ageing Well and e-Health (ICT4AWE '16), Rome, Italy, 21–22 April 2016.
18. Casilari, E.; Santoyo-Ramón, J.A.; Cano-García, J.M. UMAFall: A Multisensor Dataset for the Research on Automatic Fall Detection. *Procedia Comput. Sci.* **2017**, *110*, 32–39. [CrossRef]
19. Chen, J.; Kwong, K.; Chang, D.; Luk, J.; Bajcsy, R. Wearable sensors for reliable fall detection. In Proceedings of the 28th Annual International Conference of the Engineering in Medicine and Biology Society (EMBS '06), New York, NY, USA, 30 August–3 September 2006.

20. Li, Q.; Stankovic, J.A.; Hanson, M.A.; Barth, A.T.; Lach, J.; Zhou, G. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN '09), Berkeley, CA, USA, 3–5 June 2009.
21. Ojetola, O.; Gaura, E.; Brusey, J. Data Set for Fall Events and Daily Activities from Inertial Sensors. In Proceedings of the 6th ACM Multimedia Systems Conference (MMSys '15), Portland, OR, USA, 18–20 March 2015; pp. 243–248.
22. Leutheuser, H.; Schuldhaus, D.; Eskofier, B.M. Hierarchical, Multi-Sensor Based Classification of Daily Life Activities: Comparison with State-of-the-Art Algorithms Using a Benchmark Dataset. *PLoS ONE* **2013**, *8*, e75196. [CrossRef] [PubMed]
23. Vavoulas, G.; Pediaditis, M.; Chatzaki, C.; Spanakis, E.G.; Manolis, T. The MobiFall Dataset: Fall Detection and Classification with a Smartphone. *Int. J. Monit. Surveill. Technol. Res.* **2014**, *2*, 44–56. [CrossRef]
24. Luque, R.; Casilari, E.; Moron, M.J.; Redondo, G. Comparison and Characterization of Android-Based Fall Detection Systems. *Sensors* **2014**, *14*, 18543–18574. [CrossRef] [PubMed]
25. Vilarinho, T.; Farshchian, B.; Bajer, D.G.; Dahl, O.H.; Egge, I.; Hegdal, S.S.; Lones, A.; Slettevold, J.N.; Weggersen, S.M. A Combined Smartphone and Smartwatch Fall Detection System. In Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM '15), Liverpool, UK, 26–28 October 2015.
26. Google Play Store. Available online: https://play.google.com/store (accessed on 18 July 2018).
27. Apple Store. Available online: https://www.apple.com (accessed on 18 July 2018).
28. Windows Store. Available online: https://www.microsoft.com/en-in/store/apps/ (accessed on 18 July 2018).
29. Pirttikangas, S.; Fujinami, K.; Hosio, S. Experiences on data collection tools for wearable and ubiquitous computing. In Proceedings of the International Symposium on Applications and the Internet (SAINT '08), Turku, Finland, 28 July–1 August 2008.
30. G-Sensor Logger. Available online: https://play.google.com/store/apps/details?id=com.peterhohsy.gsensor_debug (accessed on 22 January 2018).
31. Sensor Data Logger. Available online: https://play.google.com/store/apps/details?id=net.steppschuh.sensordatalogger (accessed on 22 January 2018).
32. Sensor Log. Available online: https://apkpure.com/sensor-log/com.hfalan.activitylog (accessed on 19 July 2018).
33. UniMiB AAL. Available online: http://www.sal.disco.unimib.it/technologies/unimib-aal/ (accessed on 18 July 2018).
34. NTP: The Network Time Protocol. Available online: http://www.ntp.org/ (accessed on 18 July 2018).
35. Jacko, J.A. *Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*, 3rd ed.; CRC Press, Inc.: Boca Raton, FL, USA, 2012.
36. Material Design. Available online: https://material.io/guidelines/ (accessed on 19 July 2018).
37. Activity Recognition Training. Available online: https://play.google.com/store/apps/details?id=com.wirthual.artraining.anne (accessed on 19 July 2018).
38. Vaizman, Y.; Ellis, K.; Lanckriet, G.; Weibel, N. ExtraSensory App: Data Collection In-the-Wild with Rich User Interface to Self-Report Behavior. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18), Montreal, QC, Canada, 21–26 April 2018.
39. Sensor Data Collector. Available online: https://play.google.com/store/apps/details?id=de.unima.ar.collector (accessed on 19 July 2018).
40. Sensor Sense. Available online: https://play.google.com/store/apps/details?id=com.kristofjannes.sensorsense (accessed on 19 July 2018).
41. Ceaparu, I.; Lazar, J.; Bessiere, K.; Robinson, J.; Shneiderman, B. Determining Causes and Severity of End-User Frustration. *Int. J. Hum. Comput. Interact.* **2004**, *17*, 333–356. [CrossRef]
42. Human Interface Guidelines. Available online: https://developer.apple.com/design/human-interface-guidelines/ (accessed on 19 July 2018).
43. Buttons: Floating Action Button. Available online: https://material.io/design/components/buttons-floating-action-button.html (accessed on 18 July 2018).
44. Cards. Available online: https://material.io/design/components/cards.html (accessed on 18 July 2018).

45. Java Platform SE 8. Available online: https://docs.oracle.com/javase/8/docs/api/java/lang/System.html#currentTimeMillis-- (accessed on 22 January 2018).
46. Medrano, C.; Igual, R.; Plaza, I.; Castro, M. Detecting falls as novelties in acceleration patterns acquired with smartphones. *PLoS ONE* **2014**, *9*, e94811. [CrossRef] [PubMed]
47. Sensor Manager, Android Developers. Available online: https://developer.android.com/reference/android/hardware/SensorManager (accessed on 1 May 2018).
48. Build, Android Developers. Available online: https://developer.android.com/reference/android/os/Build (accessed on 1 May 2018).
49. Sensor, Android Developers. Available online: https://developer.android.com/reference/android/hardware/Sensor (accessed on 1 May 2018).
50. Brooke, J. SUS: A quick and dirty usability scale. *Usabil. Eval. Ind.* **1996**, *189*, 4–7.