HARDroid: Human Activity Recognition using a collaborative approach

| Alberto Gimenez  Facultad Politécnica - UNA  San Lorenzo, Paraguay  albergimenez@gmail.com | Santiago Yegros  Facultad Politécnica - UNA  San Lorenzo, Paraguay  santiago.yegros@gmail.com | Joaquín Lima  Facultad Politécnica - UNA  San Lorenzo, Paraguay  joaquin.lima@pol.una.py |
| --- | --- | --- |

# ABSTRACT

Human Activity Recognition (HAR) is a research topic broadly covered in the last decade for its relevance in areas where the users’ context is important to build interactive applications. Smartphone applications have the capability to collect data from the environment and along with algorithms that take advantage of context-aware information becomes a powerful development platform. In this paper, we propose a HAR System denominated HARDroid that is specifically designed to detect common user activities. Furthermore, data collected from users on ground are taken into account to improve the activity recognition classifier. HARDroid is freely available as a library that may be included in Android applications. Finally, an evaluation that comparing the initial classifier with an improved classifier is presented, achieving a recall of 91% and a precision of 92%.

## Author Keywords

## Location-Aware/Contextual Computing; Collaboration; Mobile Devices: Phones/Tablets; Quantitative Methods; Prototyping/Implementation; Machine Learning; Sensors Wearable Computers; Contextual Inquiry; Survey; Artifact or System Dataset.

## ACM Classification Keywords

D.2 SOFTWARE ENGINEERING (K.6.3)  
 D.2.2 Design Tools and Techniques  
 Software libraries

# INTRODUCTION

Human Activity Recognition (HAR) is a research topic in constant development for more than a decade and that covers the design of algorithms that collect data from people interacting with their environment to provide contextual information [2]. The common example of using these algorithms is to recognize basic ambulatory activities, which are, when an individual is walking, running, standing or sitting, all through some type of sensor or camera available for that purpose.

As smartphones began to become more widespread, HAR-based applications have been propitious to be developed in order to determine user interactivity and interact with them. This allows the use of contextual information available for various purposes such as data mining and predicting activities for various types of intelligent applications in different fields, for example in medicine, security, entertainment or military use, etc. [12].

A smartphone is equipped with varied sensors and may include a GPS, microphone, camera, luxometer, thermometer, barometer, compass and accelerometer. There are also other sensors more varied depending on the model, manufacturer or accessories that can be paired with the device. The accelerometer is the most common sensor in these devices and can measure the movement in two or three axes as well as detect the orientation of the device. The main use of provided sensors information is the recognition of human activities.

Along with the above, there has also been a breakthrough in the state of the art for the human activities recognition with sensors. This includes recognition techniques, methods of data capture and signal processing, and the application of artificial intelligence techniques such as Machine Learning [10, 11].

On the other hand, despite the large amount of software and applications that have been developed in the field of human activities recognition, there is still a lack of a software HAR component that can be extensible and be available for free use or for its improvement. That is, without relying on private Application Programming Interfaces (APIs), Software as a Service (SaaS) platforms, or third-party applications of free use but of closed definition, such as Google Play Services and Apple Health Kit, among others.

This proposal contemplates the study of the human activity recognition techniques on smartphones with focus in provide a HAR system in the form of a library that is free to use or to improve. Moreover, interactive user participations are taken into account to do a collaborative improve of the recognition classifier. The generated components are validated through experimental tests and the collected data shows the effectiveness of the resultant library.

# STRUCTURE OF HAR SYSTEMS

As in other automated learning applications, the recognition process is divided into two well-known stages, training and testing (or evaluation).

The Figure 1 shows the common phases of these two stages [12]. The training stage initially requires a set of data collected in a time series with the attributes measured from individuals performing each activity. The series are divided into time windows to apply sample extraction thereby filtering the relevant information from the raw signals.

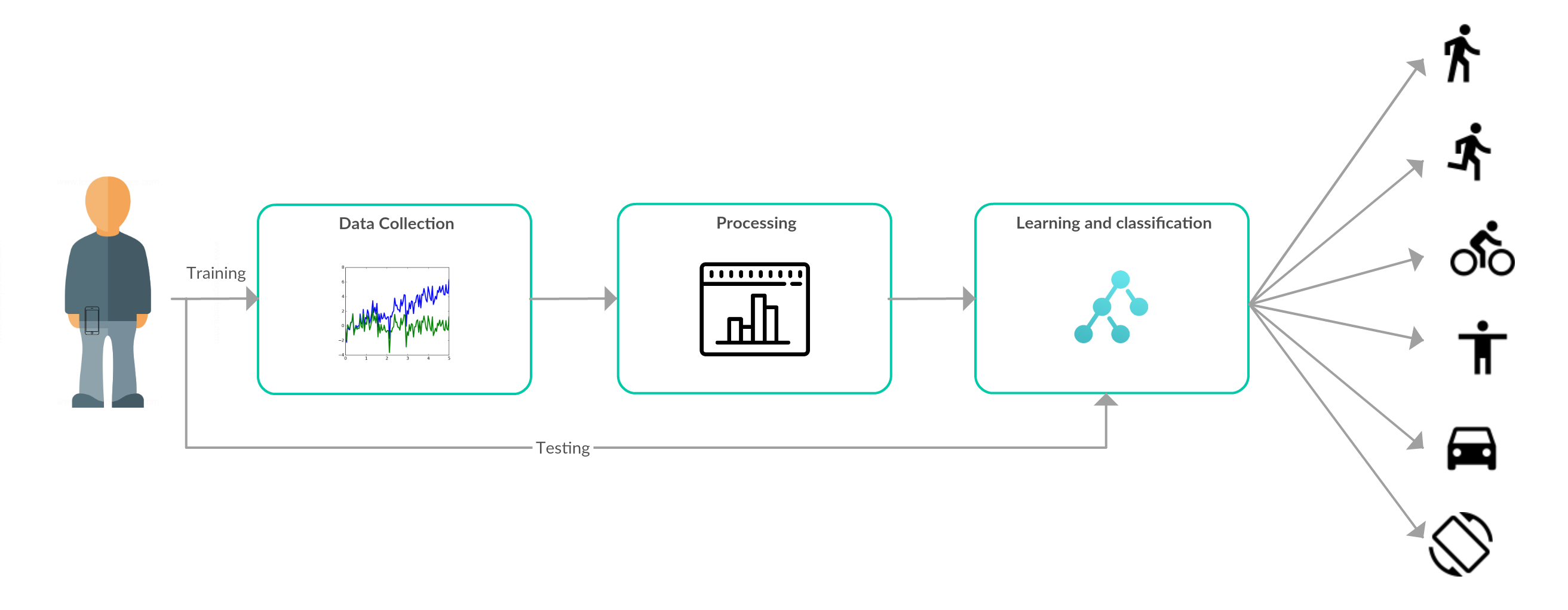


Figure 1. General structure of human activity recognition.

Later, learning methods are used to generate an activity recognition model from the data set collected through the calculated characteristics. Similarly, for the test or evaluation stage, data are collected during a time window, which is used to extract the same characteristics used in the model; these are evaluated in the previously trained learning model, generating a label of the predicted activity.

## Data Collection

The definition of the data collection method is an important point in a HAR system, since an incorrect collection can add noise to the data causing models with bad precision.

The first step in the recognition process is to collect of signals obtained from sensors that continuously are sensing users; these are attached to the body; on the waist, wrist, breastplate, thighs or on the head. Also, the sensors could be carried by the user since they are commonly embedded in devices like modern mobile phones, in watches or smart lenses.

The recording method consists of capturing the signals from a sensor and separating the measurements into one or more variables depending on the type of sensor. The organization of the records is done with respect to time. The timestamp is usually measured milliseconds and depending of the sensor the interval between measurements can vary in the same order, for example an output rate of 60 Hz have 60 measurements in a second.

The sensor signals can be classified according to movement, position, environment and physiological [12]:

### Motion Signals

Motion sensors provide highly informative signals for HAR system because they measure the acceleration and rotation forces on three axes when carried by their users. In this category of sensors are accelerometers and gyroscopes.

### Position Signals

Position sensors provide signals with additional information that can be used to HAR system and context applications with location-based services. In this category are orientation sensors (or compass), magnetometers and GPS.

### Environmental Signals

These sensors alone, though, might not provide sufficient information as individuals can perform each activity under diverse contextual conditions in terms of weather, audio loudness, or illumination. Therefore, environmental sensors are generally accompanied by accelerometers and other sensors.

### Physiological Signals

Physiological sensors provide signals of vital signs of an individual. Information on heart rate, respiration rate and body temperature could be combined to enrich the context during recognition in certain specific applications such as health-oriented.

## Processing

The next step in the recognition activities is the processing of the signals obtained by the sensors and extract relevant characteristics of the raw data. The recognition model is constructed from a sample assembly and compared using machine learning methods in the training scenario.

Sample processing relies on three distinct tasks that are performed automatically in both stages of the HAR process, in addition a manual task is performed during the training stage called labeling and then each task is detailed.

### Labeling

The automatic learning process requires a moderate amount of data collected from users while performing human activities at our study. These data should be collected and labeled using an application designed for the case, for example, in this work is used an Android phone and the application *SensorLog* [1].

The collection protocol consists of enlisting a group of people who carry a Smartphone while performing a specific set of activities and recording data through the application. The activities of interest described in Table 1 should be carried out by carrying the phone in the pocket where each individual takes a walk, run, bike or drive a vehicle for a period of 10 to 15 minutes.

| Activities | **Label** |
| --- | --- |
| ic_activity_walk.png | WALKING |
| ic_activity_run.png | RUNNING |
| ic_activity_still.png | STILL |
| ic_activity_tilt.png | TILTING |
| ic_activity_bike.png | ON\_BICYCLE |
| ic_activity_car.png | IN\_VEHICLE |

Table 1. Labeled Activities.

### Signal Filter

The electronic sensors can introduce some instability in the signal (known as jitter) causing a result in the readings due to errors in the measurement affecting the quality of the data. Therefore, even though a motion sensor device is completely stationary, the readings could record noise in the data, for example errors of the order of ±0.005.

Then, to reduce signal noise, one or more filters must be applied. The filter allows smoothing the signal by means of a simple function such as the moving average that is used in this work. There are also other methods such as Butterworth [13].

### Sampling

Human activities are performed for long periods of time, in the order of seconds or minutes. A simple measurement captured in an instant does not provide enough information to describe what activity a person is doing. Therefore, human activities must be recognized from samples drawn in time windows ***w*** rather than using a single instantaneous measure in ***t***.

Time windows cause signals to be segmented into discrete samples used as activity recognition units. To increase the number of samples, a 50% overlap with consecutive windows with a size of 2.56 seconds is used, as recommended by other HAR works such as [2, 13]. Overlap prevents certain events from being lost and activities are truncated.

### Feature Extraction

The extraction process consists in extracting characteristic values ​​in each window by means of the relevant information display and the calculation of values ​​that identify in a certain way the signals. This translates into characteristic vectors (feature vectors) with relevant information that compose several metrics calculated based on windows in the time domain. Later also, the windows are transformed into the frequency domain with discrete Fourier methods using FFT algorithms with real numbers.

The statistical metrics with respect to time and frequency used in this work are shown in Table 2. [13]:

| Function | **Methods** |
| --- | --- |
| mean(s) | Mean |
| std(s) | Standard deviation |
| max(s) | Largest values |
| min(s) | Smallest value |
| skewness(s) | Frequency signal Skewness |
| kurtosis(s) | Frequency signal Kurtosis |
| energy(s) | Average sum of the squares |
| entropy(s) | Signal Entropy |
| irq(s) | Interquartile range |
| autoregression(s) | 4th order Burg  Auto regression coefficients |
| meanFreq(s) | Frequency signal weighted  average |

Table 2. Summary of feature extraction method for acceleration signals.

The processed signals of the set ***S*** correspond to the acceleration attribute.

In order to minimize such effects caused by orientation changes, is calculated the magnitude of a sensor, from de dimensions ***x***, ***y***, ***z***. This choice was motivated about orientation-independence in activity recognition, because the magnitude feature is less sensitive to orientation changes.

## Learning and Classification

A HAR system is similar to any automatic learning application (Machine Learning, ML) where an algorithm is required to extract information from the data. The main purpose of the algorithm is to classify the unknown data.

The primary requirement for the learning task is to choose the appropriate algorithm for classification. There are many classification algorithms, such as Decision Trees, SVM, Bayes Classifiers, Markov Models, Neural Networks and others. This work classifies unknown instances by a model built with C4.5 algorithm based on the decision tree approach whose implementation is based on Java implementation J48 [4].

One of the primary topics of ML algorithms is the way in which data are processed by classifying unknown data and used to construct an effective recognition model. In dealing with the data, it is a good strategy to divide the available input instances into a training set and a remnant known as a test set. The validation set allows evaluating the classifier by determining if the results are over-adjusted and with how accurate the model is.

# DESCRIPTION OF HARDROID

HARDroid is a system that classifies human activities on smartphones based on Android platform. In general, the system design matches that of existing systems such as Google Play Services [9], thus providing the same functionality to recognize activities by taking into account concepts from the state of the art of HAR.

## Design

The system has two main components: an application-programming interface (API) that exposes capabilities and a background service that performs key calculations. Like any API, a well-documented list of function signatures is provided so that any third-party application can be integrated with the HAR system. The background service is an Android application that implements data preprocessing and human activity recognition algorithms.

The purpose of this work is to have a decoupled design in order to achieve an extensible HAR system. As with any software system, a suitable design allows for easy evolution and maintenance without affecting the operation of client dependent applications.

## Implementation

The background service is a resource manager that can be viewed as a utility service in the layer of the Android *Application Framework* [7]. This recognition service is capable of update the classifier dynamically. A utility service has the advantage that third-party applications keep up with the latest improvements thanks to automatic updates from the Google Play store.

Figure 2 shows the HARDroid service integration scheme that is common in the development of distributed applications for Android, such as the Google Play Services implementation [9]. Third-party applications are confined in different operating system processes, each on an independent *Dalvik Virtual Machine* (DVM) [3]. Inter-process communication (IPC) is carried out via an interface called *Android Interface Definition Language* (AIDL) part of the *IPC Binder* mechanism [14].

The recognition service is composed of the following modules.

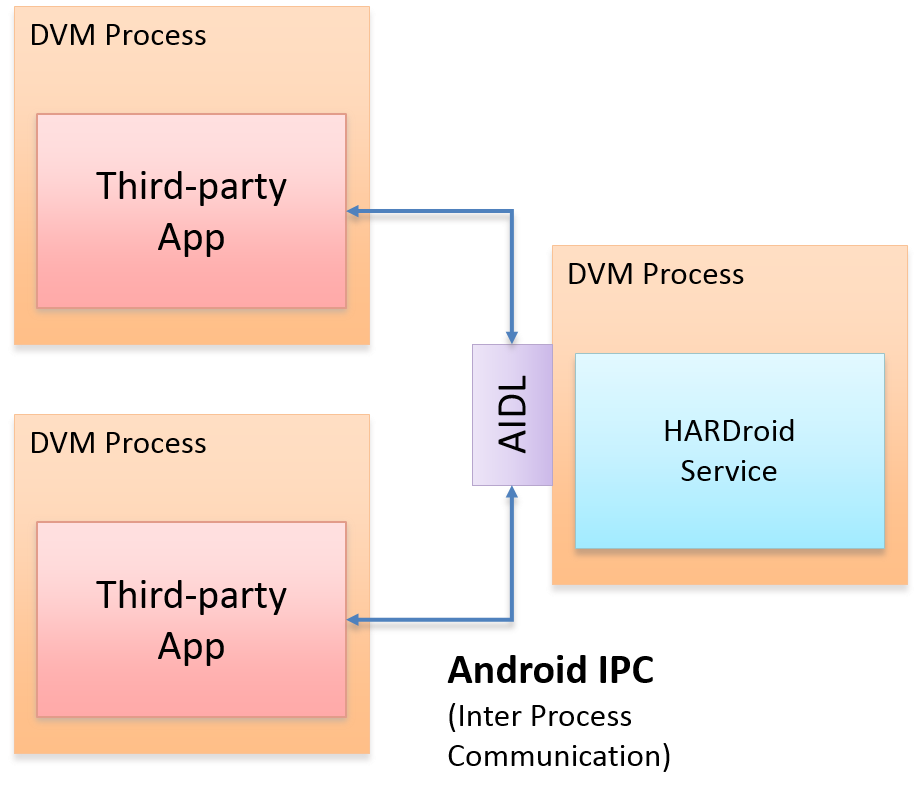


Figure 2. HARDroid service integration.

### Service Interface

The module defines a domain model and a service layer that conforms the API [8]. The domain model is composed of entities *Client*, *Connection*, *Human Activity*, *Feature* and *Activity Recognition Result*. The service layer is defined by a set of calls that the clients can execute like *Connect*, *Disconnect*, *Subscribe* to periodic recognition events and *Get* the last recognized activity.

In order to clarify the interface provided by the service layer a short API signature is described in the following listing:

* requestSingleUpdate (callbackIntent: PendingIntent): void public
* requestActivityUpdates (detectionIntervalMillis: long, listener: ActivityRecognitionListener): void public
* requestActivityUpdates (detectionIntervalMillis: long, callbackIntent: PendingIntent): void public
* removeActivityUpdates (listener: ActivityRecognitionListener): void public
* removeActivityUpdates (callbackIntent: PendingIntent): void public

### Connection Handler

The module is responsible for handling remote procedure calls from clients. The business logic is divided into two features that are manage connection lifecycle and notify subscription events through the artifacts ActivityRecognitionService and ActivityRecognitionSubscription respectively.

The server manages a subscriber list where each client is registered with the notification preference for a detected activity event.

### Processing Engine

The module is responsible for recognizing human activity that is executed at periodic time intervals as long as there are registered subscriptions. The recognition engine always produces an overall result of human activity by means of the following artifacts:

* ActivityRecognitionWorker: performs key calculations of data collection, data processing and activity classification to issue a new result.
* FeatureProcessing: a utility for feature set calculations.
* SignalProcessing: a utility for signal processing calculations.

The recognition engine starts along with the background service and performs its tasks in a predefined time interval according to the shortest time recorded in the subscriptions. When a recognition event occurs, a new Android Intent is posted for interested clients.

### Classifier

The module is responsible for classifying human activities from feature sets and issuing an estimate of the detected activity. First, a generic class ActivityClassifier is defined for common behaviors and two implementations are provided:

• DecisionTreeClassifier: an implementation based on decision trees generated with the WEKA tool [6].

• DumbClassifier: a simple implementation that produces an unknown result.

In addition, a DexModelLoader is provided to get a dynamic classifier that can frequently update the model thanks to collaboration. This utility is to download from the Internet future improvements of WEKA-generated classifiers packaged in a secured JAR library [5].

## Evaluation Tools

In order to verify the operation of the HARDroid service and to evaluate the results produced by the detection of human activities performed by a user, the application ActivitySurvey was created. The design of this mobile application is quite simple; it has the basic functionality to test the integration with HARDroid and use it as a tool to survey the actual users during the training sessions.

The responses completed by the user during the survey can be used to improve the classifier by synchronizing the data with a REST web service called Backend C4.5. It stores the results produced when using HARDroid in order to create improved learning models through feedback.

In Figure 3, the general view of the project architecture is shown.

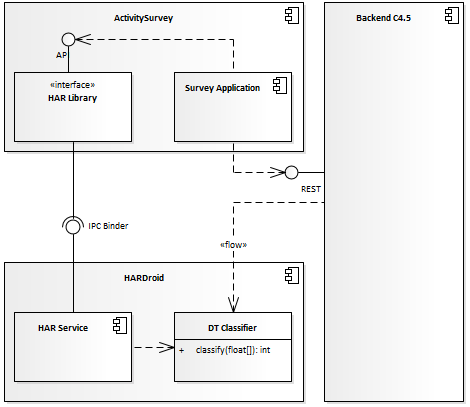


Figure 3. Evaluation project architecture.

The diagram is described in UML notation where the components represent two independent distributed mobile applications (HARDroid and ActivitySurvey) and a web server (Backend C4.5) used to collect experimental data.

# EVALUATION OF HAR SYSTEM

As discussed in previous section, in order to build and evaluate the recognition engine, experimental data collection and a training set are required.

## Experimental Data

The experimental data was gathered from a group of eight volunteers between the ages of 20 and 38. The data capture procedure was instructed with the use of the smartphone carried in the pocket or at the waist while performing a predetermined physical activity performed in order for a period of 2 to 15 minutes. *SensorLog* application [1] was used for data labeling according to the listing in Table 1 and shown in Figure 4.

A total of 6.904.165 measures were collected from seven different smartphones resulting in 12.012 labeled feature sets summarized in Table 3.

| **Activity** | **Features** |
| --- | --- |
| WALKING | 5.915 |
| RUNNING | 3.019 |
| STILL | 645 |
| TILTING | 485 |
| ON\_BICYCLE | 1.338 |
| IN\_VEHICLE | 610 |

Table 3. Calculated Features.

****

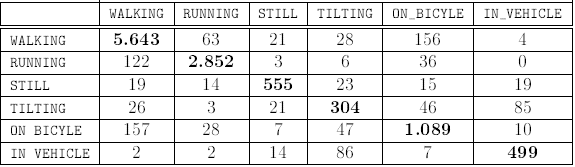
Figure 4. SensorLog application

## Classification Model

A classification model is constructed using the C4.5 algorithm for a simplified implementation in *Java* language. The training of the classification model is automated using WEKA with the following parameters [4]:

* Confidence threshold for pruning (confidenceFactor): 0.25
* Minimum number of instances per left (minNumObj): 2
* Number of folds (numFolds): 37
* Cross-validation folds: 10

The classifier is a *Java* class suitable for execution on Android platform. In addition, the assessment of the classification accuracy is evaluated on the training data from the output produced by the tool that is the confusion matrix in Figure 5 with an overall precision of 91.74% and recall 91.09%.

Figure 5. Confusion Matrix

## Verification

The HAR classifier model assessment was done by guided exercises performed by two individuals. First, each physical activity was performed over a period of 10 to 20 minutes using the ActivitySurvey application. ActivitySurvey queries the prediction accuracy of the human activity posted by the HARDroid service at regular intervals. During a session the information such as telephone, mail, age, gender, date and time, detected activity and user suggested activity is recorded.

Table 4 shows the count of successful and unsuccessful detected activities in proportion to the total number of detections collected during the survey sessions. The label TILTING is omitted because is not user basic activity.

| **Activity** | **Error** | **Success** | **Precision** |
| --- | --- | --- | --- |
| WALKING | 13 | 151 | 92.07% |
| RUNNING | 13 | 83 | 86.46% |
| STILL | 8 | 140 | 94.59% |
| ON\_BICYCLE | 12 | 59 | 83.10% |
| IN\_VEHICLE | 11 | 83 | 88.30% |

Table 4. Calculated Features.

Along with the above, it can be seen that the classifier has a high success rate in most activities according to an average of 88.8% success. In addition, the reader may even notice a good performance for activities such as the bicycle and the vehicle that are more difficult to predict.

## Classifier Improvement

In general, ActivitySurvey is a tool for collecting processed information to improve classifier by taking into account feedback from users. According to this approach, the information collected during the evaluation survey is combined with the initial experimental data, thus improving the collaborative classifier.

The result is a collaborative classification model generated with the addition of correctly classified instances with a precision of 92.04%, recall of 91.34% and the features sets increasing to 12,578. Therefore, the collaborative classifier is an improvement over the initial classifier model by means of the feedback of instances collected in ground.

# Conclusion

In this work, we present a novel HAR system under the Android platform. The main contribution is a reusable open source library to recognize human activities. Furthermore, this component enables the iterative improvement of its performance through a collaborative scheme. The evaluation tests have encouraging results with a high success rate of 92%, and the possibility that the model can be improved with a collaborative effort.

# REFERENCES

1. Hasan Faik Alan, Bert Arnrich, Cem Ersoy and Burcu Cinaz. 2014. Sensor log: A mobile data collection and annotation application. In *22nd Signal Processing and Communications Applications Conference, SIU 2014 - Proceedings*, 1375-1378.
2. Ling Bao and Stephen S. Intille. 2004. Activity Recognition from User-Annotated Acceleration Data. *Pervasive Computing* (Springer International Publishing), 1 - 17.
3. David Ehringer. 2010. Dalvik Virtual Machine*.* Retrieved January 15, 2017 from http://davidehringer.com/software/android/The\_Dalvik\_Virtual\_Machine.pdf
4. Eibe Frank. J48 Implementation*.* 2017. Retrieved March 6, 2017 from http://weka.sourceforge.net/doc.stable/weka/classifiers/trees/J48.html.
5. Luca Falsina. Grab’n Run*.* 2014. Retrieved November 20, 2016 from   
   http://grab-n-run.readthedocs.io/en/latest/index.html.
6. Eibe Frank, Mark A. Hall and Ian H. Witten. 2017. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". In *Morgan Kaufmann*.
7. Marko Gargenta and Masumi Nakamura. 2014. *Learning Android: Develop Mobile Apps Using Java and Eclipse.* 2nd. O'Reilly Media, Inc.
8. Alberto Gimenez and Santiago Yegros. 2016. Activity Recognition API*.* Retrieved November 1, 2016 from http://hardroidpy.github.io/javadoc/index.html.
9. Google Inc. Google APIs for Android*.* 2016. Retrieved May 6, 2016 from https://developers.google.com/android/.
10. Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. 2011. Activity recognition using cell phone accelerometers. In *ACM SIGKDD Explorations Newsletter* (ACM) 12, no. 2, 74.
11. Oscar D. Lara and Miguel A. Labrador. 2012. A mobile platform for real-time human activity recognition. In  *IEEE Consumer Communications and Networking Conference (CCNC).* IEEE, 667-671.
12. Oscar D.Lara and Miguel A. Labrador. 2013. A Survey on Human Activity Recognition using Wearable Sensors. In *IEEE Communications Surveys & Tutorials* 15, no. 3, 1192-1209.
13. Jorge Luis Reyes Ortiz. 2015. *Smartphone-Based Human Activity Recognition.* Springer Theses. Springer International Publishing, Cham. http://dx.doi.org/10.1007/978-3-319-14274-6.
14. Thorsten Schreiber. 2011. *Android Binder*. Master Thesis. Retrieved November 11, 2016 from https://www.nds.rub.de/media/attachments/files/2011/10/main.pdf