Daily activity recognition based on meta-classification of low-level audio events

Theodoros Giannakopoulos and Stasinos Konstantopoulos
Institute of Informatics and Telecommunication, National Center for Scientific Research 'Demokritos', Athens, Greece
tyiannak@gmail.com, konstant@iit.demokritos.gr

Keywords: audio event recognition, Raspberry PI, audio sensors, activity recognition, fusion

Abstract: This paper presents a method for recognizing activities taking place in a home environment. Audio is recorded and analysed realtime, with all computation taking place on a low-cost Raspberry PI. In this way, data acquisition, low-level signal feature calculation, and low-level event extraction is performed without transferring any raw data out of the device. This first-level analysis produces a time-series of low-level audio events and their characteristics: the event type (e.g., 'music') and acoustic features that are relevant to further processing, such as *energy* that is indicative of how loud the event was. This output is used by a meta-classifier that extracts long-term features from multiple events and recognizes higher-level activities. The paper also presents experimental results on recognizing kitchen and living-room activities of daily living that are relevant to assistive living and remote health monitoring for the elderly. Evaluation on this dataset has shown that our approach discriminates between six activities with an accuracy of more than 90%, that our two-level classification approach outperforms one-level classification, and that including low-level acoustic features (such as energy) in the input of the meta-classifier significantly boosts performance.

1 INTRODUCTION

Home automation is quickly becoming a widely available commodity, with an increasing variety of sensing and actuation hardware available in the market. This is an important enabler for telehealth and assisted living environments, transforming in marketable products the considerable volume of research on remotely collecting medically relevant information. Advancements in artificial intelligence and intelligent monitoring have been explored as a way to prolong independent living at home (Barger et al., 2005; Hagler et al., 2010; Mann et al., 2002). Several methods have been used to detect activities of daily living in real home environments, focusing on elderly population (Vacher et al., 2013; Vacher et al., 2010; Costa et al., 2009; Botia et al., 2012; Chernbumroong et al., 2013; Stikic et al., 2008; Giannakopoulos et al., 2017). In addition, special focus has been given in fall detection (El-Bendary et al., 2013) and physiological monitoring (Li et al., 2014; Petridis et al., 2015b; Petridis et al., 2015a).

These opportunities for prolonging independent living at home are needed to sustain our aging population, but their application also raises concerns regarding the acquisition and processing of the data collected. Commercial solution always deploy very simple units at the customers' location that record and transfer raw content to cloud services operated by the solution manufacturer. This is ethically questionable even in cases of providing non-health related automation services, but becomes even more so in the face of medical problems and the collection of data that pertains to such problems.

Additionally there is an obvious issue with user acceptance if raw audio or visual data is sent to the cloud. In this work, we demonstrate a framework that uses audio information in order to extract low-level events and then recognizes a series of higher-level activities usually performed in the context of a real home environment. Towards this end, we have integrated a Raspberry PI device that records and analyses audio in real time. The result of this procedure is a sequence of low-level *audio events*, i.e., a sequence of labels that characterize short segments of the audio signal. Naturally, the selection of labels depends on the application at hand. For the application and use cases presented here, we have labels for the sounds typically occurring in a home environment. At a second stage, a meta-classifier maps the low-level events to high-level and long-term activities (e.g. watching TV, cleaning up the kitchen, etc.) The conceptual ar-

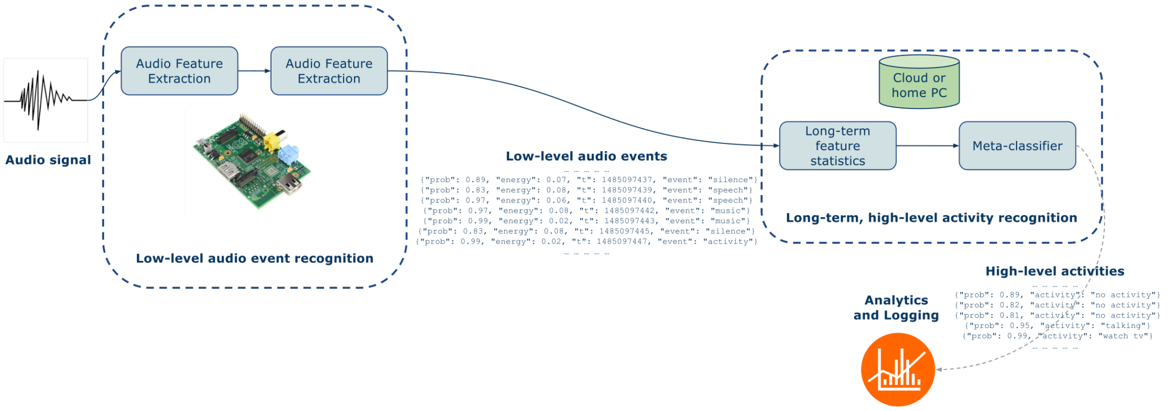

Audio signal
Audio Feature Extraction
Audio Feature Extraction
Low-level audio event recognition
Low-level audio events
… … … … …
{"prob": 0.89, "energy": 0.07, "t": 1485097437, "event": "silence"}
{"prob": 0.83, "energy": 0.08, "t": 1485097439, "event": "speech"}
{"prob": 0.97, "energy": 0.06, "t": 1485097440, "event": "speech"}
{"prob": 0.97, "energy": 0.08, "t": 1485097442, "event": "music"}
{"prob": 0.99, "energy": 0.02, "t": 1485097443, "event": "music"}
{"prob": 0.83, "energy": 0.08, "t": 1485097445, "event": "silence"}
{"prob": 0.99, "energy": 0.02, "t": 1485097447, "event": "activity"}
… … … … …
Cloud or home PC
Long-term feature statistics
Meta-classifier
Long-term, high-level activity recognition
High-level activities
… … … … …
{"prob": 0.89, "activity": "no activity"}
{"prob": 0.82, "activity": "no activity"}
{"prob": 0.81, "activity": "no activity"}
{"prob": 0.95, "activity": "talking"}
{"prob": 0.99, "activity": "watch tv"}
… … … … …
Analytics and Logging

Figure 1: Conceptual architecture of the proposed scheme

chitecture of this rationale is illustrated in Figure 1. The contribution of the work described here is the following:

- a state-of-the-art sound analyser is integrated on a low-cost sensor that functions both as data acquisition and pattern analysis module
- experimentation on a real-world dataset from a join living-room and kitchen space has proven that the proposed metaclassifier is very accurate (more than 90% on 6 basic activities), and more accurate than directly recognizing the high-level activities from the raw audio features

2 LOW-LEVEL AUDIO EVENT RECOGNITION

Audio-based low-level event recognition is achieved through a Raspberry PI device that performs both audio acquisition and real-time signal recognition. In particular, the Raspberry PI captures audio samples from the attached microphone device and executes a set of real-time feature extraction and classification procedures, to provide online audio event recognition. We have presented more details of this architecture in the context of a practical workflow that helps the technicians setup the device through a fast, user-friendly and robust tuning and calibration procedure (Siantikos et al., 2016). In this way, ‘quotestraining of the device is performed without any need for prior knowledge of machine learning techniques. In particular, in order to train the classifier that discriminates between low-level audio events, we developed Graphical User Interface (GUI) for Android. This app acts as a ‘remote control’ for the Raspberry Pi device and through that, the user can: (a) Record and annotate sounds: the user notifies the Raspberry Pi device that a particular audio event (e.g., music playing) is currently taking place, so that the Raspberry Pi can collect recordings of each class (b) Train the audio classifier: The user triggers the training procedure, which results in a probabilistic Support Vector Machine classifier (SVM) of audio segments (Platt, 1999).

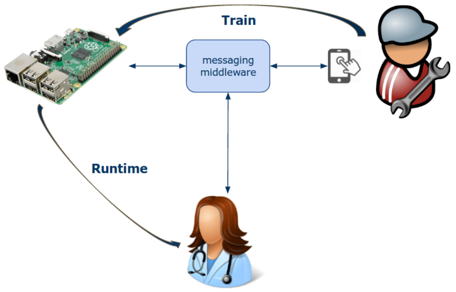
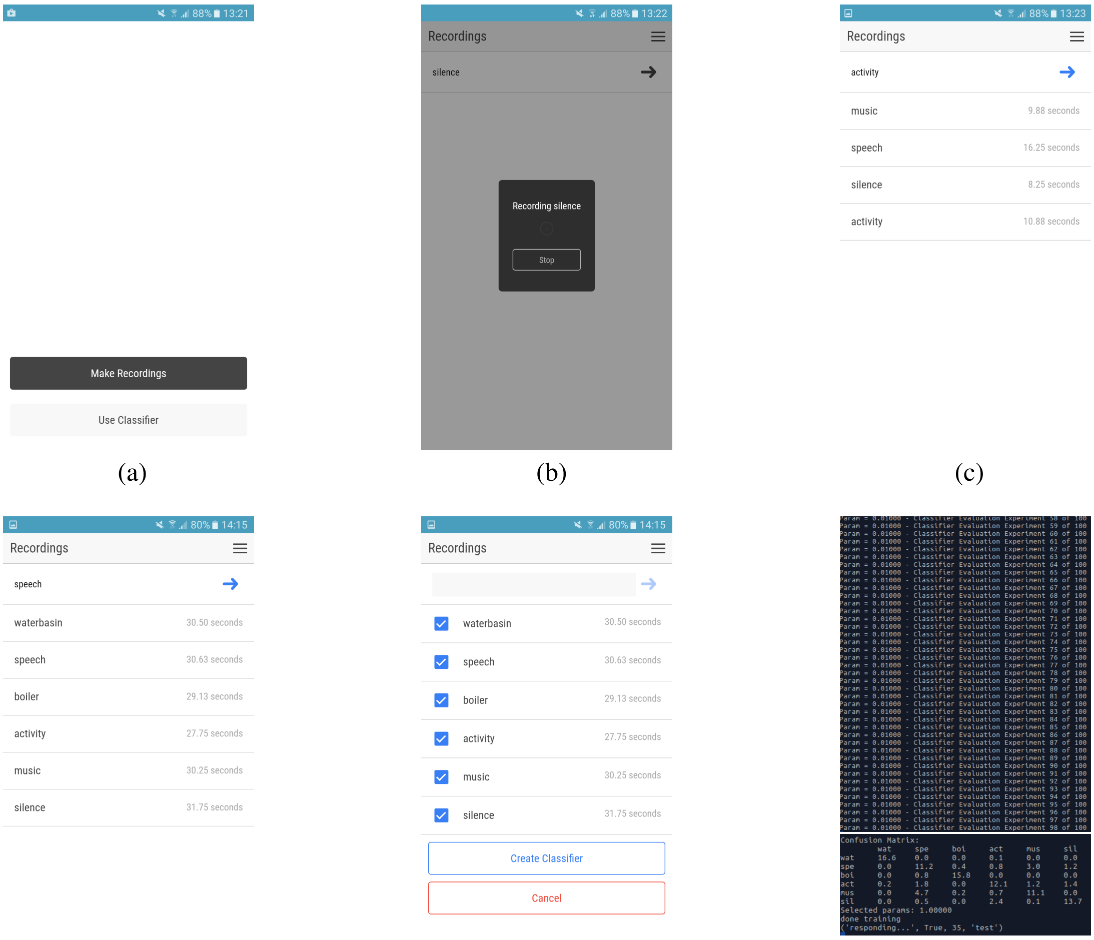


Figure 2: MQTT-based calibration procedure

Communication between the mobile app and the Raspberry PI device is achieved through MQTTm which is is a lightweight messaging protocol. In our use case, it is used both for sending commands to the Raspberry PI (for example to start/stop recording a sample of a given class) and for remotely receiving the processing results.

An example of screenshots of such a training procedure is shown in Figure 3. Here, the technician successively records sounds from all 6 classes (Figure 3 a–d). As more recordings are added, the total duration dynamically changes in the respective GUI area. When the data are sufficient (typically at least 20 seconds are recorded from each class), the user triggers the classification training by selecting the appropriate audio classes and pressing ‘Create Classifier’ (Figure 3e). Then, the SVM tuning procedure is executed on the Raspberry PI device. As soon as this step is executed, the confusion matrix of the internal cross-validation procedure is returned (Figure 3f) and the



(a)

(b)

(c)

music 9.88 seconds
speech 16.25 seconds
silence 8.25 seconds
activity 10.88 seconds

(d)

waterbasin 30.50 seconds
speech 30.63 seconds
boiler 29.13 seconds
activity 27.75 seconds
music 30.25 seconds
silence 31.75 seconds

(e)

☑ waterbasin 30.50 seconds
☑ speech 30.63 seconds
☑ boiler 29.13 seconds
☑ activity 27.75 seconds
☑ music 30.25 seconds
☑ silence 31.75 seconds

(f)

```
Param = 0.01000 - Classifier Evaluation Experiment 58 of 100
Param = 0.01000 - Classifier Evaluation Experiment 59 of 100
Param = 0.01000 - Classifier Evaluation Experiment 60 of 100
Param = 0.01000 - Classifier Evaluation Experiment 61 of 100
Param = 0.01000 - Classifier Evaluation Experiment 62 of 100
Param = 0.01000 - Classifier Evaluation Experiment 63 of 100
Param = 0.01000 - Classifier Evaluation Experiment 64 of 100
Param = 0.01000 - Classifier Evaluation Experiment 65 of 100
Param = 0.01000 - Classifier Evaluation Experiment 66 of 100
Param = 0.01000 - Classifier Evaluation Experiment 67 of 100
Param = 0.01000 - Classifier Evaluation Experiment 68 of 100
Param = 0.01000 - Classifier Evaluation Experiment 69 of 100
Param = 0.01000 - Classifier Evaluation Experiment 70 of 100
Param = 0.01000 - Classifier Evaluation Experiment 71 of 100
Param = 0.01000 - Classifier Evaluation Experiment 72 of 100
Param = 0.01000 - Classifier Evaluation Experiment 73 of 100
Param = 0.01000 - Classifier Evaluation Experiment 74 of 100
Param = 0.01000 - Classifier Evaluation Experiment 75 of 100
Param = 0.01000 - Classifier Evaluation Experiment 76 of 100
Param = 0.01000 - Classifier Evaluation Experiment 77 of 100
Param = 0.01000 - Classifier Evaluation Experiment 78 of 100
Param = 0.01000 - Classifier Evaluation Experiment 79 of 100
Param = 0.01000 - Classifier Evaluation Experiment 80 of 100
Param = 0.01000 - Classifier Evaluation Experiment 81 of 100
Param = 0.01000 - Classifier Evaluation Experiment 82 of 100
Param = 0.01000 - Classifier Evaluation Experiment 83 of 100
Param = 0.01000 - Classifier Evaluation Experiment 84 of 100
Param = 0.01000 - Classifier Evaluation Experiment 85 of 100
Param = 0.01000 - Classifier Evaluation Experiment 86 of 100
Param = 0.01000 - Classifier Evaluation Experiment 87 of 100
Param = 0.01000 - Classifier Evaluation Experiment 88 of 100
Param = 0.01000 - Classifier Evaluation Experiment 89 of 100
Param = 0.01000 - Classifier Evaluation Experiment 90 of 100
Param = 0.01000 - Classifier Evaluation Experiment 91 of 100
Param = 0.01000 - Classifier Evaluation Experiment 92 of 100
Param = 0.01000 - Classifier Evaluation Experiment 93 of 100
Param = 0.01000 - Classifier Evaluation Experiment 94 of 100
Param = 0.01000 - Classifier Evaluation Experiment 95 of 100
Param = 0.01000 - Classifier Evaluation Experiment 96 of 100
Param = 0.01000 - Classifier Evaluation Experiment 97 of 100
Param = 0.01000 - Classifier Evaluation Experiment 98 of 100
```

```
Confusion Matrix:
        wat     spe     boi     act     mus     sil
wat     16.6    0.0     0.0     0.1     0.0     0.0
spe     0.0     11.2    0.4     0.8     3.0     1.2
boi     0.0     0.8     15.8    0.0     0.0     0.0
act     0.2     1.8     0.0     12.1    1.2     1.4
mus     0.0     4.7     0.2     0.7     11.1    0.0
sil     0.0     0.5     0.0     2.4     0.1     13.7
Selected params: 1.00000
done training
('responding...', True, 35, 'test')
```

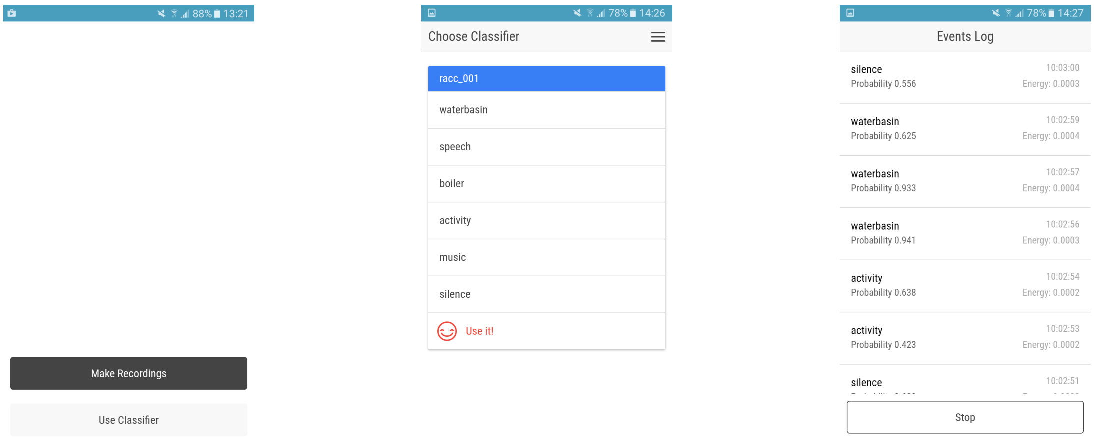
Figure 3: Screenshots of the training (calibration) Graphical User Interface

Raspbery PI device is now equipped with the audio classifier, and ready to function as an audio event detector. When the audio segment classifier is trained, the Raspberry PI device is ready to record sounds from the microphone and (in an online mode) recognize low-level audio events. Figure 4 shows three screenshots of the GUI when the Raspberry PI device in on the 'testing mode': first the user selects a pretrained classifier (in our case the classifier trained as shown in Figure 3) and then the Raspberry PI device starts recording and classifying audio segments. The results of this classification procedure are pushed to the mobile GUI interface (Figure 4c).

Audio feature extraction and classification has been achieved using the open-source pyAudioAnalysis library (Giannakopoulos, 2015), that implements a wide range of audio analysis functionalities. The complete set of features and more details on the classification procedure is presented in (Siantikos et al., 2016), where this architecture has been adopted for recognizing low-level audio events in the context of a bathroom activity monitoring scenario. Finally, we have selected to adopt a 1 second of signal length for each classification decision. In other words, in the runtime mode the Raspberry PI device broadcasts a classification decision every one second.

3 HUMAN ACTIVITY RECOGNITION

The procedure described in Section 2 trains a classifier integrated on the Raspberry PI device, using an Android mobile device as a 'remote controller' for making the whole process faster and easier. As soon as this process is completed, the Raspberry PI de-



silence 10:03:00
Probability 0.556 Energy: 0.0003

waterbasin 10:02:59
Probability 0.625 Energy: 0.0004

waterbasin 10:02:57
Probability 0.933 Energy: 0.0004

waterbasin 10:02:56
Probability 0.941 Energy: 0.0003

activity 10:02:54
Probability 0.638 Energy: 0.0002

activity 10:02:53
Probability 0.423 Energy: 0.0002

silence 10:02:51

Figure 4: Screenshots of the testing Graphical User Interface.

vice can record audio streams from the microphone and automatically classify audio segments of 1 second length to any of the 6 audio classes. The goal of this work is to extract higher-level labels regarding everyday human activities that typically occur the living room and kitchen area. In the context of this work, we have defined the following higher-level events: (1) no activity (NO), (2) talking (TA), (3) watching TV (TV), (4) listening to music (MU), (5) cleaning up the kitchen (KI) and (6) other activity (OT). Our task here is to infer these activities in a long-term rate (e.g. every one minute) using the short-term, low-level events. Towards this end, we have adopted a simple meta-classification procedure according to which low-level audio segment decisions are used to extract feature statistics that are fed as input to a supervised model, namely a Support Vector Machine meta-classifier. In particular, the adopted features are the following:

- $F_i = \sum_{j=0}^{N} 1 : \lambda(j) = i$, where $i = 1,..6,$, $\lambda(j)$ is the class label of the j-th audio segment and N is the total number of audio segments in a session (recording). These first 6 features are actually the distribution of short-term, low-level audio event labels as extracted by the audio classifier
- F_7 is the average number of transitions from silence to any of the non-silent audio classes per second
- F_8 to F_{11} are the mean, max, min and std statistics of the normalized segment energies.

This 11-dimensional feature vector is used to classify the whole session (recording) in terms of its respective activity. As shown in the experiments section of the paper, the simple low-level energy statistics add to the performance of the low-level event statistics (i.e. the first 6 high-level features). The classification methods adopted, in order to classify the high-level features to any of the activity classes are the following:

- k-Nearest Neighbor classifier
- Probabilistic SVMs (Platt, 1999)
- Random forests is an ensemble learning method used for classification and regression that uses a multiude of decision trues (Ho, 1995), (Pal, 2005)
- Gradient boosting: an ensembling approach interpreted as an optimization task, widely used both in classification and regression tasks (Breiman, 1997; Friedman, 2002)
- Extremely Randomized Trees (or Extra trees) is a modern classification tree-ensemble approach that is based on strong randomization of both attribute and cut-point choice while splitting the tree node (Geurts et al., 2006), and has been adopted in various computer vision and data mining applications.

It has to be noted that this meta-classification stage can be either executed as a cloud or a home-PC service, or even as an independent submodule in the Raspberry PI device. In any case, the core concept of the proposed approach is that the joint audio feature extraction-audio event recognition module broadcasts only low-level events and no raw information, in order to be be complied with the privacy requirements of a health monitoring or assisted living application.

Furthermore, in order to compare the proposed approach with the straightforward classification of high-level activities from low audio features, the pyAudioAnalysis library has also been used to train and evaluate an SVM classifier that learns to discriminate between high-level activities *based on the low-level time* *sequences of short-term features*. As shown in the experiments section, this approach achieves much lower performance rates compared to the proposed meta-classification module.

4 EXPERIMENTS

4.1 Datasets

In order to train and evaluate the proposed methodology, three separate datasets have been compiled and manually annotated. The first two of the datasets are associated with the traing and the evaluation stage of the audio segment classification submodule (i.e. the low-level audio event classifiers), while the third dataset is used to train and evaluate the meta-classifier (using cross-validation). In mode detail, the following datasets have been adopted:

Audio segment classification dataset (Training): This dataset consists of one uninterrupted recording for each class, recorded through the mobile calibration GUI (Section 2). The total duration of each recording is 200 seconds, giving 20 minutes in total for all six classes. This is the total time required to 'calibrate' the audio recognition model, plus the time required for the SVM model to be trained.

Audio segment classification dataset (Evaluation). This dataset consists of 600 audio segments (100 from each audio class). Each audio segment is 1-sec long, since this is the selected decision window duration. This dataset is used to *evaluate* the audio classification module. Obviously the segments of these datasets are totally independent to the segments of the previous dataset to avoid bias in evaluation.

High-level event dataset This dataset is used to evaluate the proposed high-level activity recognition method. It consists of 20 long-term sessions for each activity class (120 sessions in total). Each long-term session corresponds to a recording of 20 seconds to 2 minutes that has led to a series of short-term, low-level audio events. This dataset is openly provided.[1] Each recording-session corresponds to another file. All files follow the JSON format, with the following fields: (a) winner class probability (b) (normalized) signal energy (c) timestamp and (d) winner class. Note that these four fields are provided for each 1-second audio segment of the recording. Therefore, each file has several rows that correspond to several audio segments of the same recording-session. Training and classification performance evaluation has been achieved using repeated random sub-sampling using this dataset. In particular, in each sub-sampling 10% of the data is used for testing.

4.2 Experimental Results

4.2.1 Low-level audio event recognition evaluation

Table 1 presents the confusion matrix, along with the respective class recall, class precision and class F1 values for the audio segment classification task. The second dataset described in Section 4.1 has been used to this end. The overall F1 measure was found to be equal to 83.5%. This actually means that more than 8 out of 10 audio segments (1 second long) are correctly classified to any of the 6 audio classes, on average.

4.2.2 High-level activity recognition

First, we present the F1 measures for all meta-classifiers applied on (a) meta-features 1 to 6 (b) meta-features 1 to 7 and (c) all meta features. The best performance is achieved for the Support Vector Machine classifier, when all features are used. In addition, the 7th meta-feature (the average, per second, transitions from silence to other audio classes) adds 2%. Furthermore, if the energy-related features are also combined, the overall performance boosting reaches 4%. Note that for the SVM classifier, a linear kernel has been adopted. Probably this is the reason that the SVM classifier overperforms the more sophisticated ensemble-based classifiers, since it is proven to be more robust to overfitting due to low training sample size. Second, Table 3 presents the detailed evaluation metrics for the best high-level event recognition method, i.e. the SVM classifier for all adopted high-level features.

According to the results above, the proposed methodology achieves a very high recognition accuracy for the six activities, based on simple low-level audio decisions and signal energy statistics, especially when the SVM classifier is used. The highest confusion is achieved between 'watching TV' and 'talking'–'music', as well as between 'kitchen cleanup' and 'other activity'. As explained in the next section, our ongoing work focuses on the expansion of some of the high-level classes to more detailed representations: for example, the music-related activities could be expanded to 'listening to music' and 'dancing' and 'other activities' may be further split to 'eating', 'drinking', 'exercising' and 'cooking'. However such information requires both more detailed low-level audio representations and other modalities, e.g. visual and motion information, stemming from cameras and accelerometers respectively. An example of

[1] Available at `https://zenodo.org/record/376480`

Table 1: Audio segment classification results: Row-wise normalized confusion matrix, recall precision and F1 measures. Overall F1 measure: 83.5%

Confusion Matrix (%)

	Predicted					
True ⇓	activity	boiler	music	silence	speech	waterbasin
activity	79.3	0.0	0.0	5.2	6.9	8.6
boiler	32.4	51.4	2.7	13.5	0.0	0.0
music	0.0	0.0	88.0	0.0	12.0	0.0
silence	0.0	0.0	0.0	100.0	0.0	0.0
speech	0.0	0.0	12.9	0.0	87.1	0.0
waterwasin	0.0	0.0	0.0	0.0	0.0	100.0

Performance Measurements (%, per class)

Recall:	79.3	51.4	88.0	100.0	87.1	100.0
Precision:	71.0	100.0	85.0	84.3	82.2	92.1
F1:	74.9	67.9	86.5	91.5	84.6	95.9

Table 2: Comparison between classification methods and feature sets for the high-level activity recognition task. Performance is quantified based on the F1 measure

	Features		
Method	1-6	1-7	1-11 (all)
kNN	84	88	84
SVM	88	90	**92**
Random Forests	89	89	90
Extra Trees	88	89	88
Gradient Boosting	88	88	89

Table 3: High-level event recognition based on all meta-features for the best classifier (SVM) Overall F1 measure: 92%

Confusion Matrix (%)

	Predicted					
True ⇓	NO	TA	TV	MU	KI	OT
NO	100	0	0	0	0	0
TA	0	96	4	0	0	0
TV	7	11	71	5	0	5
MU	0	5	5	89	0	0
KI	0	1	0	0	95	4
OT	0.5	0	0	0	0.5	99

Performance Measurements (%, per class)

Recall:	100	96	71	89	95	99
Precision:	93	85	89	95	99	91
F1:	96	90	79	92	97	95

accelerometer information and an explanation of how it could be used for fusion with audio sensing is presented in the last section.

Finally, for comparison purposes, we have evaluated the ability of a one-level audio classifier that directly maps mid-term audio feature statistics to high-level activities. Using the same dataset, the F1 measure for this method was significantly lower (80%). Additionally, such an approach would require more training data, since it needs to much a high-dimensional audio feature space to semantically-high activity classes. This is not only computationally demanding, but impractical, since acquiring annotations for high-level activities is a laborious and intense task. On the other hand, the proposed approach has made training low-level, short-term multidimensional audio classifiers an easy task (through the calibration procedure described in Section 2), while the training of the meta-classifier requires only a few training samples, since it is based on audio segment classification decisions, not on very low-level and multi-dimensional audio features.

5 CONCLUSIONS AND FURTHER WORK

We have presented a low-cost solution for recognizing human activities in a home environment using low-level audio labels. Towards this end, we expanded our previous related work (Siantikos et al., 2016) by (a) training an audio segment classifier that distinguishes between low-level audio events that usually appear in the context of a joint living-room and kitchen environment (b) proposing and evaluating a meta-classification technique that uses low-level audio events and simple signal energy values, in order to recognize long-term and high-level activities. Both the dataset and the prototype implementation used for the experiments described here are publicly avail-

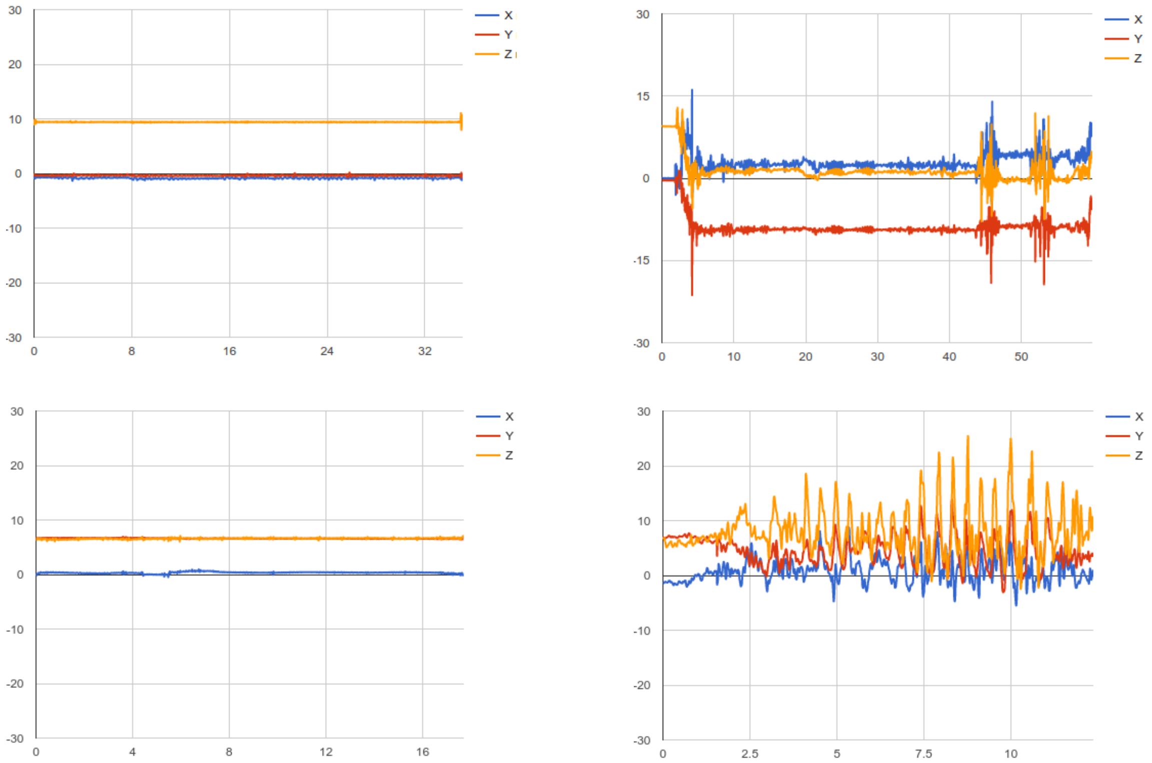


Figure 5: Example of accelerometer data for two activities: eating (upper left), kitchen cleaning up (upper right), listening to music (lower left) and dancing (lower right)

able.[2]

A range of modern classifiers has been evaluated on this meta-classification task and the results have proven that probabilistic SVMs that combine both low-level audio classification decisions and simple energy values of raw audio segments outperforms all other classifiers and feature combinations. In particular, this experimentation on a real dataset has shown that the proposed approach successfully classifies activities in 92% of the cases.

Further work on this topic aims to refine the activity taxonomy, to fuse acoustic results with other modalities, and to include temporal information about the low-level events. Specifically, it is obvious that the classes used here can be expanded to a more detailed *taxonomy* of types and sub-types of activities, but also that further top-level activities related to one's ability to function independently (such as preparing and consuming a meal, currently lumped under 'other') should be added. It is also obvious that as the taxonomy gets finer, the performance of acoustic-only classification will start dropping, as some activities cannot be distinguished using only-audio information; consider, for example, making the distinction between 'listening to music' and 'dancing to the music'.

In order to boost the classification performance for such cases, we have started experimenting with *fusing* audio information with accelerometer data, which has been lately used in human activity recognition for the elderly (Khan et al., 2010), (de la Concepción et al., 2017). Figure 5 illustrates the raw accelerometer values (along the three spatial axes) for four activities, namely: eating, cleaning up the kitchen, listening to music and dancing. It is obvious that these time series can be used to discriminate between the dancing and listening to music classes, once acoustic modelling has established the presence of 'music' in the environment.

A second opportunity to improve the discrimination ability of the classifier, is to include information about the *temporal evolution* of the low-level events. In future work, we will experiment with methods from *complex event recognition* (Artikis et al., 2014) to describe complex events in a way that takes into account temporal information.

[2] See `https://zenodo.org/record/376480` and `https://github.com/radio-project-eu/AUROS`

ACKNOWLEDGEMENTS

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 643892. Please see `http://www.radio-project.eu` for more details.

REFERENCES

Artikis, A., Gal, A., Kalogeraki, V., and Weidlich, M. (2014). Event recognition challenges and techniques. *ACM Transactions on Internet Technology*, 14(1).

Barger, T. S., Brown, D. E., and Alwan, M. (2005). Health-status monitoring through analysis of behavioral patterns. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(1):22–27.

Botia, J. A., Villa, A., and Palma, J. (2012). Ambient assisted living system for in-home monitoring of healthy independent elders. *Expert Systems with Applications*, 39(9):8136–8148.

Breiman, L. (1997). Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at Berkeley.

Chernbumroong, S., Cang, S., Atkins, A., and Yu, H. (2013). Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications*, 40(5):1662–1674.

Costa, R., Carneiro, D., Novais, P., Lima, L., Machado, J., Marques, A., and Neves, J. (2009). Ambient assisted living. In *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, pages 86–94. Springer.

de la Concepción, M. Á. Á., Morillo, L. M. S., García, J. A. Á., and González-Abril, L. (2017). Mobile activity recognition and fall detection system for elderly people using ameva algorithm. *Pervasive and Mobile Computing*, 34:3–13.

El-Bendary, N., Tan, Q., Pivot, F. C., and Lam, A. (2013). Fall detection and prevention for the elderly: A review of trends and challenges. *Int. J. Smart Sens. Intell. Syst*, 6(3):1230–1266.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

Giannakopoulos, T. (2015). pyaudioanalysis: An open-source python library for audio signal analysis. *PloS one*, 10(12):e0144610.

Giannakopoulos, T., Konstantopoulos, S., Siantikos, G., and Karkaletsis, V. (2017). Design for a system of multimodal interconnected adl recognition services. In *Components and Services for IoT Platforms*, pages 323–333. Springer.

Hagler, S., Austin, D., Hayes, T. L., Kaye, J., and Pavel, M. (2010). Unobtrusive and ubiquitous in-home monitoring: a methodology for continuous assessment of gait velocity in elders. *Biomedical Engineering, IEEE Transactions on*, 57(4):813–820.

Ho, T. K. (1995). Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE.

Khan, A. M., Lee, Y.-K., Lee, S. Y., and Kim, T.-S. (2010). A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE transactions on information technology in biomedicine*, 14(5):1166–1172.

Li, X., Chen, J., Zhao, G., and Pietikainen, M. (2014). Remote heart rate measurement from face videos under realistic situations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4264–4271.

Mann, W. C., Marchant, T., Tomita, M., Fraas, L., and Kathleen, S. (2002). Elder acceptance of health monitoring devices in the home. *Care Management Journals*, 3(2):91–98.

Pal, M. (2005). Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222.

Petridis, S., Giannakopoulos, T., and Perantonis, S. (2015a). Unobtrusive low-cost physiological monitoring using visual information. In *Handbook of Research on Innovations in the Diagnosis and Treatment of Dementia*, pages 306–316. IGI Global.

Petridis, S., Giannakopoulos, T., and Spyropoulos, C. D. (2015b). A low cost pupillometry approach. *International Journal of E-Health and Medical Communications*, 6(4):49–61.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. Citeseer.

Siantikos, G., Giannakopoulos, T., and Konstantopoulos, S. (2016). A low-cost approach for detecting activities of daily living using audio information: A use case on bathroom activity monitoring. In *Proceedings of the 2nd International Conference on Information and Communication Technologies for Ageing Well and e-Health (ICT4AWE 2016)*.

Stikic, M., Huynh, T., Van Laerhoven, K., and Schiele, B. (2008). Adl recognition based on the combination of rfid and accelerometer sensing. In *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, pages 258–263. IEEE.

Vacher, M., Portet, F., Fleury, A., and Noury, N. (2010). Challenges in the processing of audio channels for ambient assisted living. In *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, pages 330–337. IEEE.

Vacher, M., Portet, F., Fleury, A., and Noury, N. (2013). Development of audio sensing technology for ambient assisted living: Applications and challenges. *Digital Advances in Medicine, E-Health, and Communication Technologies*, page 148.