Smartphone active use recognition by movement sensors data

Miguel Bermeo-Ayerbe, David Reixach-Perez
Master's degree in Automatic Control and Robotics
Escola Tècnica Superior d'Enginyeria Industrial de Barcelona (ETSEIB)
Universitat Politècnica de Catalunya (UPC)
{miguel.bermeo.ayerbe, david.reixach}@estudiant.upc.edu

Abstract **This paper proposes a method to recognize if a smartphone device is being actively used by the user, in a specific time, by analysing movement sensor signals and making use of a Neural Network, a well-known machine learning technique. The problem is stated as a state identification problem which intends to characterize the position in which the user carries the device, hand or pocket. The features obtained from the signals, the architecture of the Neural Network and the training algorithm are explained in detail. Also a motivation application for such method is explained.**

Keywords—Machine Learning; Pattern Recognition; Smartphone sensors; Signal Processing; Feature extraction; Neural Networks; Backpropagation; Classification.

I. INTRODUCTION

More than a decade of development in smartphones technology leverages devices with enhanced sensing capabilities. They include different sensors such acceleration or velocity (movement), cameras, microphones and air pollution sensors among others. Also their accuracy has been largely increased.

This situation provides a new board of opportunities to develop new functionalities that lay on making the device aware of its context. For example, using information from the movement sensors the smartphone could learn to recognize when the user is travelling, thus limit the use of the GPS tracking system to this cases in order to save energy.

In this work we use movement data (accelerometer and angular velocity) to predict whether the smartphone is being actively used by the user. We assume the user could not be having interaction with the device but making use of it, for example reading the screen or being short-time apart from it. We inspire in providing a new security functionality that makes the device remain unlocked when it is being actively used by the user, otherwise locks it. We tackle our problem as a state identification problem and we principally want to differentiate between states in which the user carries its device on its hand or in its pocket.

Our input data will be a fixed length time-series record of sensor values. We will extract some features over this data and build a model to classify between states.

This paper is structured as follows. Section II explains related works analysed, Section III explains problem formulation, Section IV explains the experiment developed and Section V obtains several conclusions on the work.

II. STATE OF THE ART

Similar works have been done regarding this machine learning application [1][2][3][4]. Normally they handle it as a human activity recognition problem, possibly including user identification (by means of gait recognition) [1]. In this case they build a classification model to recognize which activity is being done and they build an identification model in order to determine which the user is. Actually they build a model per user that only identifies if the data comes from its specific user or not. This method provides the capability of learning specific patterns among users.

Other types of works, such [4], focus on identifying devices context, the position in which the device is being carried by the user. In this case they build an aggregated model that includes data from different sensors (microphone, camera …) and merges all this information. Thus, building specific models per instance and including high data processing intermediate steps. Features they obtain from each data need also important data processing, such signal analysis (Fourier decomposition) and others.

In our case we build a single model that inputs the same type of raw data and outputs a classification label to a state identification, similar to the first step of [1]. We make use of movement sensors (acceleration and angular velocity) and implement a Neural Network as a classifier.

III. Proposed Approach

A. *Problem formulation*

We represent our reality with states and transitions. Our aim is to recognize the situation in which the user is found at a given time. In other words, we want to know if the input data comes from a specific state or transition. We treat states and transitions equally as classification labels.

We define five states and two transitions (between two specific states). Our states are: the device being carried by the user actively using it, standing and moving (states 1 and 2); with the device in its pocket, standing and moving (states 3 and 4) and the device being left on a surface (e.g. table, state 5). Transitions are defined between states 2 and 4, so while the user moving, stopping making active use of the device (leaving it inside its pocket) and vice versa (transitions 1 and 2). Making active use of the device within a time domain means holding the device in a fixed position relative to the user, Specifically in our case, horizontally perpendicular to the floor (e.g. as watching a video). A graphic scheme regarding our formulation can be seen in Fig. 1.

B. *Data (Sensors + type of data)*

The sensors we use are the accelerometer and the angular velocity sensors. They provide these magnitudes along three dimensions (x,y,z) so the sensors measure is a 6 dimension data vector.

We record these data along a continuous fixed time of twenty seconds what provides the raw time-series signals. Then we extract features over this data and we will build a model to classify them. An example of these signals can be seen in Fig. 2.

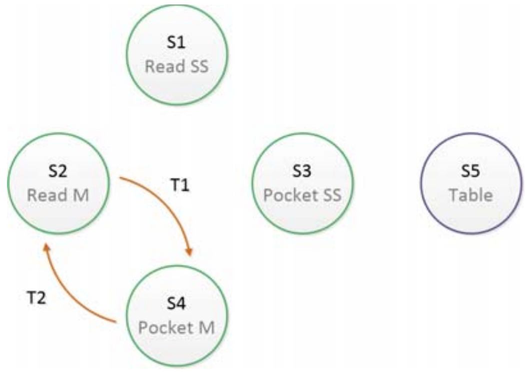

Fig. 1 Problem formulation graph, states and transitions

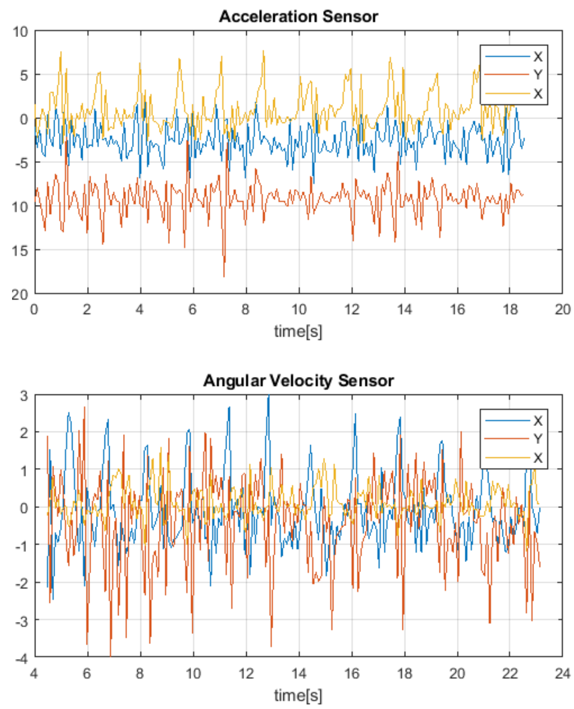

Fig. 2 Acceleration sensor and angular velocity sensor measurement example corresponding to state 1.

C. *Implementation*

In this section we first explain the features we extract from the raw data and secondly, we describe the model we build to make our classifier.

We extract 14 features from the two 3-dimension signals:

- Acceleration average [3]: Average value for each component within the time series.
- Acceleration standard deviation [3]: Standard deviation value for each component within the time-series.
- Acceleration energy [1]: Euclidean norm of the aggregated values along the time-series. Eq. (1).

$$\sqrt{\left(\sum_{i=0}^{N} x_i\right)^2 + \left(\sum_{i=0}^{N} y_i\right)^2 + \left(\sum_{i=0}^{N} z_i\right)^2} \quad (1)$$

- Angular velocity average [3]: Average value for each component within the time series.
- Angular velocity standard deviation [3]: Standard deviation value for each component within the time-series.
- Angular velocity energy [1]: Euclidean norm of the aggregated values along the time-series. Eq. (1).

We use an Artificial Neural Network trained using backpropagation as a classifier. The classifier inputs the 14-dimension extracted feature vector from the raw data and outputs a class probability. The output label will be assigned to the output with highest probability.

The architecture we choose is a wide used Neural Network for Pattern Recognition. It has a ten-node hidden layer with a hyperbolic tangent function and a 7 node output layer with a *Softmax* function to normalize probabilities. The architecture is shown in Fig. 3.

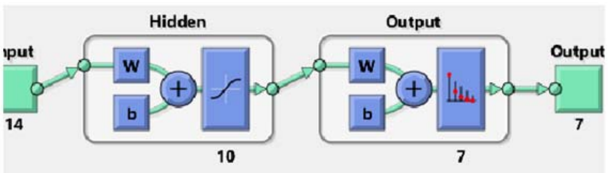

Fig. 3 Neural Network architecture scheme (MATLAB® representation)

IV. Experimentation

A. *Experiment design*

Our experiment will be based on data from two different subjects for every state and transition. We record twenty samples per subject and per instance, what imply more than 90 minutes of time-series data. The 60% of this data will be used for model training, and the rest for evaluation later.

We develop a framework for data collection, data processing and model training and testing in MATLAB®. For the classification model we make use of MATLAB® native functions [5].

The training algorithm does a second level of data separation itself. The division for internally training, testing and validation we choose is 70, 15 and 15% respectively. We implement a scaled conjugate gradient backpropagation algorithm [5].

Cross-entropy [5] function evaluates the performance of the classifier. This function penalizes non-homogeneously the classification error (i.e. heavily penalizes outputs extremely far away from the target and modestly does it to more fair classifications).

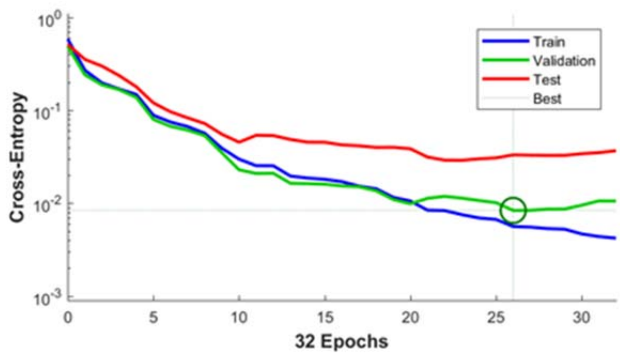

Fig. 4 Classifier performance over the training evolution.

In Fig. 4 the training evolution is shown. While the algorithm uses training data with different initializations to obtain different parametrizations, and testing data to choose the best among them, validation data is used to decide when the training stage has to be stopped. It obtains a maximum stable performance (minimum Cross-Entropy) after 26 epochs.

B. *Experiment result*

After the model is obtained we test it with the remaining data we separated at the beginning. The labels correspond from 1 to 4 to states 1 to 4, labels 5 and 7 to transitions 1 and 2 and label 6 to state 5.

The results are shown in Fig. 5. We observe an outstanding performance. The vast majority of the classes are well recognized despite their different nature (states and transitions). We observe that the confusion majorly happens between states and transitions independently (despite a 4 state is misclassified as transition 1). Confusion mainly happens between states and specifically the model tends to misclassify to state 1. There is low confusion between transitions.

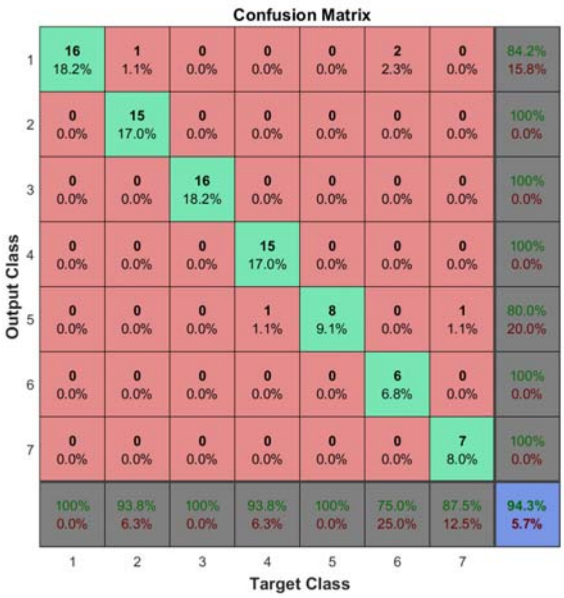

Fig. 5 Confusion matrix of the evaluation of the classifier using test data.

V. CONCLUSIONS

In this paper we proposed an approach to recognize if a user is making active use of a mobile device by means of movement sensors. Our motivation comes from developing a security application in order to automate device locking.

Our method relays on analysing sensors measurements along a time window. We process the signals obtaining some features and we run a classifier.

Despite its simplicity, the results obtained show availability of the method and provide a wide open field for improvement, such learning better features in order to characterize the events. Also reducing the length of the time-series would need to be achieved in order to fulfil application requirements.

REFERENCES

[1] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” SIGKDD Explor. Newsl., vol. 12, no. 2, pp. 74–82, Mar. 2011.

[2] L. Bao and S. Intille, “Activity recognition from user-annotated acceleration data,” in Pervasive Computing, ser. Lecture Notes in Computer Science, A. Ferscha and F. Mattern, Eds. Springer Berlin Heidelberg, 2004, vol. 3001, pp. 1–17.

[3] P. Siirtola and J. Roning, “Recognizing human activities user- ¨ independently on smartphones based on accelerometer data,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. 1, no. 5, pp. 38–45, 06 2012.

[4] Miluzzo, E., Papandrea, M., Lane, N.D., Lu, H., & Campbell, A.T. (2010). Pocket, Bag, Hand, etc. - Automatically Detecting Phone Context through Discovery.

[5] MathWorks®, (2018). Neural Network Toolbox: User's Guide (R2018a). Retrieved March 2018 from https://es.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf