



# Smartphone Mode Recognition During Stairs Motion <sup>†</sup>

Lioz Noy <sup>1</sup>, Nir Bernard <sup>1,\*</sup> and Itzik Klein <sup>2</sup>

<sup>1</sup> Department of Electrical Engineering, The Technion - Israel Institution of Technology, Haifa 32000, Israel  
lioznoy@campus.technion.ac.il

<sup>2</sup> Department Of Marine Technology, University of Haifa, Haifa 32000, Israel, iklein@technion.ac.il

\* Correspondence: nir4560@campus.technion.ac.il

<sup>†</sup> Presented at the 6th International Electronic Conference on Sensors and Applications, 15–30 November 2019; Available online: <https://ecsa-6.sciforum.net/>

Published: 14 November 2019

**Abstract:** Smartphone mode classification is essential to many applications, such as daily life monitoring, healthcare, and indoor positioning. In the latter, it was shown that knowledge of the smartphone location on pedestrians can improve the positioning accuracy. Most of the research conducted in this field is focused on pedestrian motion in a horizontal plane. In this research, we use supervised machine learning techniques to recognize and classify the smartphone mode (text, talk, pocket and swing) while accounting for the movement up and down stairs. We distinguish between the going up and the down motion, each with four different smartphone modes, making 8 states in total. This classification is based on the use of an optimal set of sensors that varies according to battery life and the energy consumption of each sensor. The classifier was trained and tested on a data set constructed from multiple user measurements (total of 94 min) to achieve robustness. This provided an accuracy of more than 90% in cross validation method and 91.5% if the texting mode is excluded. When considering only stairs motion, regardless of the direction, the accuracy improves to 97%. These results may assist many algorithms, mainly in pedestrian dead reckoning, in improving a variety of challenges such as speed and step length estimation and cumulative error reduction.

**Keywords:** supervised machine learning; mode recognition; stairs movement classification; smartphone sensors; pedestrian dead reckoning

---

## 1. Introduction

The need for identifying the "smartphone mode", i.e. the way a person is holding the smartphone, is becoming more and more significant in many applications such as healthcare services, commercial usages, emergency and safety applications, etc. [1]. One of the main usages of smartphone mode recognition is to improve the capabilities of indoor navigation algorithms, particulallry pedestraind dead reckoning (PDR) approaches. Such apporaches are based on the measurement of step length and heading calculation. The formar is based on ampirical or biomechanical models which are highly affected by the smartphone mode[2]. Hence the importance of a fast and accurate recognition algorithm that can classify between multiple possible smartphone modes.

The smartphone mode is characterized by the relative location of the phone, the phone movement along periods of time, different relative angles of the phone (yaw, pitch, roll), sound levels, luminous intensity, and more. Those measurements are calculated by the phone physical sensors, e.g accelerometer, gyroscope, magnetometer, etc. [4]. Those can be presented to the user and sorted on the device with

variety of applications. In this paper we aim to classify four smartphone modes – texting, talking, swing and pocket while the pedestrian is going up or down the stairs.

Each of the smartphone modes above was divided to two groups: walking up the stairs and walking down the stairs, as presented in Figure 1, resulting with an overall of eight different modes to classify. While the recognition of the four common modes were explored in the past, the ascent and descent separation is a relatively unexplored field. Results show an accuracy of 90.25% for the eight states and over 96.75% for the four main modes mentioned.

The rest of the paper is organized as follows: Section 2 presents the problem and our approach to solve it. Section 3 describes the data collection process, experimental setup and results, while Section 4 provides the conclusions.



**Figure 1. Smartphone modes illustrations.** Total of 8 different smartphone modes divided to 4 main groups: (a) phone in hand, (b) phone in pocket, (c) walking while talking on the phone, and (d) walking while texting.

## 2. Methodology

### 2.1. Problem Formulation

Let  $\mathbf{x}_t \in R^d$  a  $d$  dimension vector represent the data collection calculating by the phone sensors in time  $t$ . Corresponding to  $\mathbf{x}_t$  is the label  $y_t$  which is determined by the smartphone mode at that time. We define a time window of size  $n$  that will be the data gathered from the sensors from time  $t - n + 1$  to time  $t$ , the window label  $y_n$  must be similar to all data samples composing this window. And so, we have the following matrix:

$$\mathbf{W}_n = \begin{pmatrix} \mathbf{x}_{t-n+1} \\ \mathbf{x}_{t-n+2} \\ \vdots \\ \mathbf{x}_t \end{pmatrix}_{n \times d} \rightarrow y_n \equiv y_{t-n+1} = \dots = y_t \tag{1}$$

The quantity of time windows that are extracted from the data is a function of window size  $n$ , number of samples of data  $T$ , and the overlap percentage between the time windows  $\frac{1}{n}$ . We notice that the overlap between the time windows can only be executed backwards due to the fact that we observe a causal system and we can't use future windows in real life applications. Our goal is to design an algorithm that can classify a given time window to the correct smartphone mode label.

### 2.2. Feature Extraction

The time window  $\mathbf{W}_n$  is composed from  $d$  time vectors  $\{\mathbf{u}_i\}_1^d$  of length  $n$  respectively to the  $d$  data points in  $\mathbf{x}_t$ . On each of these vectors, we will calculate a collection of features to achieve more information on our data and improve the classification process. We distinguish between two groups of features [1,3]:

1. **Statistical features:** Will be calculated by executing statistic analysis on each vector. Examples: mean, standard deviation, median, max, min, bias, etc.
2. **Time features:** Will be calculated by counting and searching for specific conditions on the data points in the vector. Examples: peaks count, mean / median crossing, amount of similar argument, zeros count, etc.
3. **Frequency features:** Statistical and counting features above calculating on a the absolute value and the angle of a Fourier transform that been executed on each time window.
4. **Cross measurements features:** Statistical and counting features above calculating on the magnitude ( $\sqrt{f_x^2 + f_y^2 + f_z^2}$ ) of 3 axes measurements, i.e. acceleration measurements, gyroscope measurements, and magnetic field measurements.

Let  $\sigma$  represent the number of statistic features and  $\tau$  represent the amount of time features. Thus, from each time window  $\mathbf{W}_{n,i}$  we can produce a feature vector  $\chi_i$  with the corresponding label of that time window  $y_i$ . Eventually, the full feature matrix is obtained as shown in Equation 2 for  $T$  data samples, time window size  $n$ , and overlap between the windows of  $n - 1$  samples:

$$\hat{\mathbf{X}} = \begin{pmatrix} \chi_1 \\ \chi_2 \\ \vdots \\ \chi_n^T \end{pmatrix}_{\frac{T}{n} \times d \cdot (\sigma + \tau)} \rightarrow \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n^T \end{pmatrix} \quad (2)$$

### 2.3. Classification

In our research, we compare a wide variety of machine learning algorithms to determine which is the best classifier for this problem. Each classifier was trained and tested on our data set that contains sensors output from 6 different pedestrians with 6 different smartphones, and the accuracy was calculated using cross validation method as shown in Section 3. After the initial classification test, we wish to improve the results and the robustness of the algorithm using the following methods.

First, we optimize the hyper-parameters of each classifier to the parameters that suit best with the nature of our data. The next step is to perform a feature selection process in order to extract the main features from the feature  $\hat{\mathbf{X}}$  matrix. this action used to avoid over-fitting and spurious correlations, shorten training and testing time and improve the results. All physical sensors except Thermometer, Barometer and Sound level meter generates cross-sensors measurements. For example, the combination of the Accelerometer and the Gyroscope generates linear acceleration and gravity. Because of the massive amount of features we first used feature selection method which takes into account the best subset of measurements for each physical sensor, once all combinations have been tested. Afterwards, "tsfresh" dedicated module was used, which propose a p-value based approach that inspects the significance of the features individually [11].

The optimal subset of sensors and measurements to use in the classification process is also calculated. Filtering the sensors subset is significant to achieve the optimal results for many reasons. First, some of the sensors output may be not relevant for our type of classification and moreover might impair the results due to over-fitting. Second, the application of our experiment, specifically closed spaces navigation using the smartphone, required the classification to be executed quickly, accurately, and the energy consumption should be minimized, i.e. use as less sensors as possible while preserving the best results.

### 3. Setup and Results

#### 3.1. Data Collection and Processing

As mentioned in Section 2, the data from the smartphone sensors was collected by 6 users considering all eight different modes. Each user used a different smartphone with an Android operating system, the application for reading the sensors output was identical for all measurements. The sensors output sampling rate has been set to 10[Hz], if a device had a faster sampling rate the data was re-sampled accordingly. On the data sets, windows with different lengths and no overlap were applied in the purpose of executing the feature extraction process. The total amount of data collected and windows extracted for each mode is presented in Table 1.

**Table 1.** Data collection and the division to time windows of 1[s] with no overlap.

Description	Upstairs			Downstairs		
	Label	Minutes	Time windows	Label	Minutes	Time windows
Phone in hand	1	14.98	896	2	11.19	669
Phone in pocket	3	12.30	736	4	10.91	652
Talking on the phone	5	9.70	580	6	10.94	653
Texting	7	12.85	768	8	11.16	668
All labels	-	49.83	2980	-	44.2	2642

#### 3.2. Classification Process

To perform the classification process, feature extraction (see Section 2.2) is applied on the time windows (Section 2.1) constructed on the sensor raw measurements. The proposed algorithm performance is evaluated by performing a series of classification tests and comparisons. We compare four types of machine-learning classifying algorithms: K Nearest Neighbours (K-NN) [6], Decision Tree[7], Random Forest[8], and XGBoost[9].

The accuracy performances of the classifiers were determined by the portion of correct label prediction out of all the labels in the test set of each fold in the cross-validation process. For validating the proposed algorithm, the accuracy of the main modes without the stairs partition is also tested. Table 2 shows the best results of each classifier as produced with the parameters of the experiment. We notice that all the Random Forest classifiers had shown superior results relatively to the other classifiers –90.26%. Furthermore, we see that the accuracy for the 4 main modes classification has given results of over 96%, i.e. most of the false classifications occurred in the up stairs vs. downstairs labelling and not between main smartphone modes. We can verify the last statement in Figure 2 showing the confusion matrices of the Random Forest classifier. Also, the texting mode is the most inaccurate for the accent and decent division, if we test the other three modes, we achieve accuracy of 91.5%. A possible reason can be that in texting mode the user keeps the phone in a relatively static position that makes the classification more challenging than the other modes.

The accuracy of the classifiers is based significantly on the features that had been used in the classification process. Figure 3 presents the feature importance attribute of the random classifier. It is shown that the most important features in the classification are related to the specific force measurements produced by the accelerometer. Other dominant features were related to the acceleration calculation produced by the accelerometer and the atmospheric pressure produced by the barometer sensor. The influence of the barometer sensor was expected since it is often used for measuring elevation in many devices, thus the impact on stairs movement classification process. Figure 4 Shows the influence of the number of sensors used in the classification process on its performance. We see that the best results were

produced by using 8 out of the 10 sensors in the phone. It is also shown that we can obtain small regression with less battery consumption by using a different sensors subset.

Table 2. Best results of each classifier.

Classifier	Accuracy [%] - with up down division (8 labels)	Accuracy [%] - main modes (4 labels)
KNN	74.83	92.16
Decision Tree	78.61	90.90
Random Forest	90.25	96.75
XGBoost	90.22	95.74

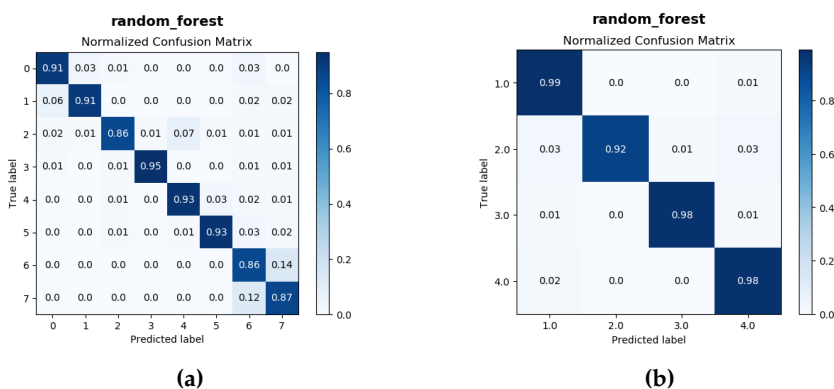


Figure 2. Random Forest confusion matrices. The accuracy of each state out of 8 labels with the stairs division according to the labels in Table 1 (a) and out of 4 labels (b) for only the main smartphone modes (1 - swing, 2 - pocket, 3 - talking, 4 - texting).

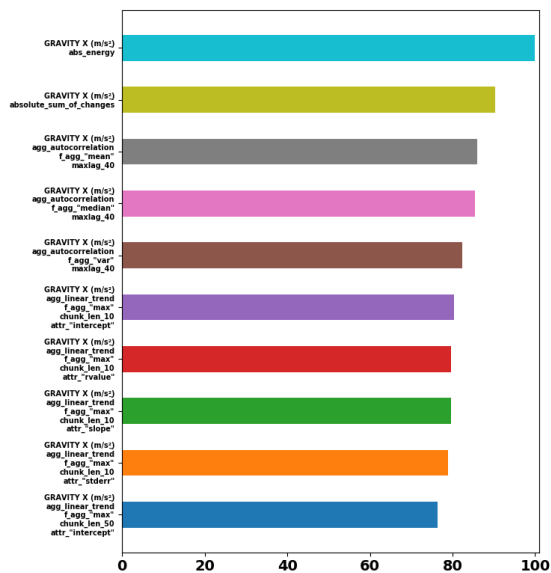


Figure 3. Feature importance. 10 most important features for the classification process.

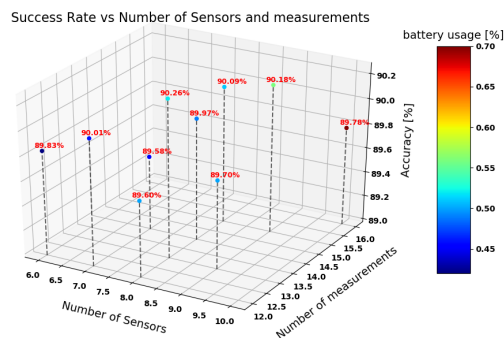


Figure 4. Sensors and measurement reduction. The accuracy of the classifiers as a function of number of sensors and phone measurements used. The color represents the battery usage percentage of the specific sensors subset used in the classification process. The power consumption was evaluated with the number of sensors and their average power consumption. [12] [13].

#### 4. Conclusions

In this research we addressed the problem of smartphone classification of four modes (texting, talking, swing and pocket) during stairs movement. A classification accuracy of 90.25% was obtained while classifying four main smartphone modes and divided them into climbing and ascending stairs – a relatively unexplored area of smartphone modes. In future research we aim to use neural network based approaches to improve the classification accuracy.

**Funding:** This research received no external funding.

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

#### References

1. I. Klein, Y. Solaz, and G. Ohayon. Pedestrian Dead Reckoning with Smartphone Mode Recognition, *IEEE Sens. J.* **2018**, *18*, 7577–7584.
2. I. Klein, Y. Solaz, and G. Ohayon. Smartphone Motion Mode Recognition. *Proceedings* **2017**, *2*, 145.
3. M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-liehr. Neurocomputing Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing*, **2018**, *307*, 72–77.
4. S. Rinaldi, A. Depari, A. Flammini, and A. Vezzoli. Integrating Remote Sensors in a Smartphone : the project "Sensors for ANDROID in Embedded systems". In proceedings of 2016 IEEE Sensors Applications Symposium, Catania, Italy, 20–22 April 2016.
5. R. Polikar. Ensemble based systems in decision making. *IEEE Circuits Syst. Mag.* **2006**, *6*, 21–45.
6. Y. Zeng, B. Wang, L. Zhao, and Y. Yang. The extended nearest neighbor classification. In proceedings of the 27th Chinese Control Conference (CCC), Kunming, China, 16–18 July 2008.
7. B. Kamiński, M. Jakubczyk, and P. Szufel. A framework for sensitivity analysis of decision trees. *EJOR* **2018**, *26*, 135–159.
8. A. Liaw and M. Wiener. Classification and Regression by randomForest. *R news* **2002**, *2*, 18–22.
9. T. Chen and C. Guestrin. Diagnosis of tuberculosis–newer tests, XGBoost: A Scalable Tree Boosting System. In proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
10. H. Li, G. Ji, and Z. Ma. A Nonlinear Predictive Model Based on Multilayer Perceptron Network. In proceedings of 2007 IEEE International Conference on Automation and Logisticspp, Jinan, China, 18–21 August 2007.
11. I. Guyon. An Introduction to Variable and Feature Selection 1 Introduction. *JMLR* **2003**, *3*, 1157–1182.
12. I. Khan, S. Khusro, S. Ali, and J. Ahmad. Sensors are Power Hungry : An Investigation of Smartphone Sensors Impact on Battery Power from Lifelogging Perspective Sensors are Power Hungry : An Investigation of Smartphone Sensors Impact on Battery Power from Lifelogging Perspective. *BUJICT* **2016**, *9*, 8–19.
13. Z. Horvath and H. Jenak. Battery consumption of smartphone sensors. *F. J Reliable Intell Environ* **2017**, *3*, 131–136 .



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).