Smartphone Mode Recognition For Improved Pedestrian Dead Reckoning

Itzik Klein*, Yuval Solaz and Guy Ohayon
Rafael-Advanced Defense Systems Ltd. Haifa, Israel
Email: *itzikkl@rafael.co.il

***Abstract*—Smartphone mode recognition is becoming a key aspect for many applications such as daily life monitoring, health care and indoor navigation. In the later, one of the approaches to perform navigation is known as Pedestrian Dead Reckoning (PDR) commonly applied using the pedestrian smartphone sensors. A step length estimation is a critical stage in PDR. It requires a prior calibration phase, prior to PDR application, to determine appropriate gains which are very sensitive to different smartphone modes, such as talking, texting, swing or pocket. Inaccurate gain will result in a position error. In this paper, we employ machine learning classification algorithms to recognize the smartphone modes and thereby enabling the choice of a proper gain value to improve PDR navigation accuracy. To that end, a methodology of training on a single user and testing on multiple users is as well as unique features for the classification process is proposed. Experimental results obtained using thirteen participates walking in an inhomogeneous environments and smartphone modes show successes of more than 95% in classifying the smartphone modes and as a consequence improvement in navigation performance.**

***Index Terms*—Mode Recognition, Machine Learning, Inertial Sensors, Pedestrian Dead Reckoning**

I. Introduction

Smartphone mode recognition is becoming a key aspect for many applications. Among them are health care and safety services, commercial and emergency applications and indoor navigation [1]. While in outdoors, the positioning of a person is usually based on Global Navigation Satellite Systems (GNSS) [14]. However, in indoor environments the availability of satellite signals cannot be guaranteed and GNSS based services can be highly degraded or totally denied. In such situations, two main approaches using inertial sensors exist to calculate a pedestrian position: 1) utilizing classical inertial navigation approaches using shoe mounted inertial sensors [3], [2] and 2) Pedestrian Dead Reckoning (PDR) [4], [5]. PDR can be applied using the smartphone low-cost sensors [9], [15] mounted on the user foot [12], [13], waist [11] or helmet [10]. Also, PDR may be applied by using external sensors such as ultra wide band positioning [6] or WiFi signals [7], [8], however such solutions relies on external transmissions and infrastructures.

When using smartphones, commonly the accelerometer, gyroscope and magnetometer measurements are used in the PDR algorithm. In, general, PDR algorithm has four steps: 1) accelerometers (sometimes gyros instead) are used to detect pedestrian steps and then 2) pedestrian step length is estimated using an empirical formula. 3) the walking direction is obtained from the gyroscope and/or magnetometer readings and thus 4) by using initial conditions, current heading (from step 3) and step length estimation (from step 2) the current pedestrian position can be determined. In order to calculate the step length from the accelerometer readings an appropriate calibration phase must be made prior to PDR application. In the calibration phase, an optimal gain (or gains) is found given the chosen step length estimation approach for the PDR process. These gains are very sensitive to pedestrian and smartphone modes [15], [16]. Inaccurate gain results in an erroneous step length estimation which in turn will cause an erroneous position update of the pedestrian position. That is, for accurate PDR one requires different gain values for every pedestrian (walking, running, elevator and etc) and smartphone (hand-held in a pocket and etc) modes.

Qain [15] employed the slope of the accelerometer output to distinguish between walking and running modes and than to select appropriate model parameters based on the user mode. In [20] four types of user modes - walking, running, bicycle and vehicle where classified using Machine Learning (ML) approaches for various locations of the sensors. There, the dataset was generated using 17 persons and then divided by half where one half of the samples was used as training data and the other half for testing. The accuracy of identifying running or walking modes was more than 93% depending on the sensors locations. Recently, a comprehensive survey paper on motion mode recognition [21] summarized published results on various motion mode types including (but not only) walking, running and stairs modes which are most relevant for PDR scenarios. The papers listed therein include the usage of accelerometers, gyros, magnetometers and GNSS, while those sensors where located in different locations on the user, and in overall managed to achieve above 90% accuracy in mode classification. SmartMTra was proposed in [22] to improve trajectory tracing using smartphones. Motion recognition was applied there for eight types of pedestrian modes and seven types of smartphone modes, where distinction was made for texting with one or two hands and big or small arm swing. The true positive rate reported was more than 88%. When addressing not only the pedestrian mode but also the the smartphone mode, [16] proposed to distinguish between four cases: static user, quasi-stable device to include texting, talking and bag carrying, swing and irregular motion. The dataset

was generated using four men and four women and the classification process was made using decision trees achieving 95% of accuracy. Later, [23] used finite state machine to classify three smartphone modes: swing, holding (texting) and pocket. Using 17 participations, they reported an accuracy above 89% in mode classification. Yand and Wang [**?**], designed a random forest classier to classify four smartphone modes: hand (swing), bag, coat pocket and trouser pocket to achieve and accuracy of 93.7%. A phone mode identification algorithm that identifies the swing mode of phone usage is developed in [24] to assist the integration of the accelerometer and gyroscope pedometers. A threshold based approach was applied to achieve an accuracy of 94% to distinguish between presence or absence of swing mode. Klein et al [25], used only the magnitudes of the gyroscopes and accelerometers measurements to classify the smartphone modes while testing on six participates to achieve an accuracy of 86.7%.

Follow our initial work on the subject, in this paper, we present an approach to accurately classify four common smartphone modes: texting, talking, in pocket and swing. Our motivation for smartphone mode classification is to improve PDR algorithms and enhance user positioning accuracy. Since, by proper classification of the smartphone modes, appropriate gains are chosen for the step length estimation phase. To emphasize this, we demonstrate the consequence of ignoring the mode recognition phase which results in an inaccurate user position estimation by using two different approaches for step-length calculation.

The main contribution of the paper is smartphone modes recognition for improved PDR performance. To achieve that goal:

- A novel Machine learning (ML) methodology of training on a single user with a single smartphone and testing on data collected from multiple users and smartphones is proposed. This methodology was initially suggested and implemented due to technical reasons, however, as presented in the paper, obtained excellent results in smartphone modes classification using a dataset generated from thirteen people.
- New features where suggested making a total of 18 different feature types applied on the smartphone three axes accelerometers and gyroscopes measurements as well on their magnitudes.
- Inhomogeneous environment and smartphone modes were used to collect the data required for the training and testing phases. The environment includes: varying walking speeds, walking on uneven pavements, walking on roads with road humps, transitions between pavement and roads. The smartphone modes includes: varying hand swing (small to big), various location of the smartphone in pocket, texting and talking with a single hand in different positions relative to the user.

After the training process was completed, the chosen classifier was applied on experimental data and showed success in classifying correct smartphone modes and thereby improving PDR accuracy. In addition, the chosen classifier showed robustness while classifying the smartphone modes even when the recording test dataset was made while going up or down the stairs. Finally, feature selection was applied to reduce the number of features while maintaining the same accuracy level as in the complete set. In that manner, insight can be gained into the most important features of the motion.

The rest of the paper is organized as follows: Section II describes the PDR algorithm and demonstrate the influence of appropriate gain selection as a function of the smartphone mode. Section III describes the machine learning methodology used for the mode recognition process. Section IV presents the experimental setup and results and Section V gives the conclusions.

II. Pedestrian Dead Reckoning

A. PDR Algorithm

Usually (but not always) PDR is defined in an horizontal plane (2D) only. Given an initial position of the pedestrian x_k, y_k at step k , the current step length s_k and heading ψ_k, the current position is calculated by

$$\begin{aligned} x_{k+1} &= x_k + s_k \cos\psi_k \\ y_{k+1} &= y_k + s_k \sin\psi_k \end{aligned} \tag{1}$$

This procedure is illustrated in Figure 1, where the following steps are used: 1) use accelerometers to detect the pedestrians travelling steps step detection, 2) estimate step length using an empirical formula which requires an initial calibration phase, 3) estimate user heading using gyroscopes/magnetometers and finally 4) propagate the users position. Notice, that the motion mode recognition does not appear in the basic PDR algorithm but is required to improve PDR performance is will be explained herein. In the next section step length estimation

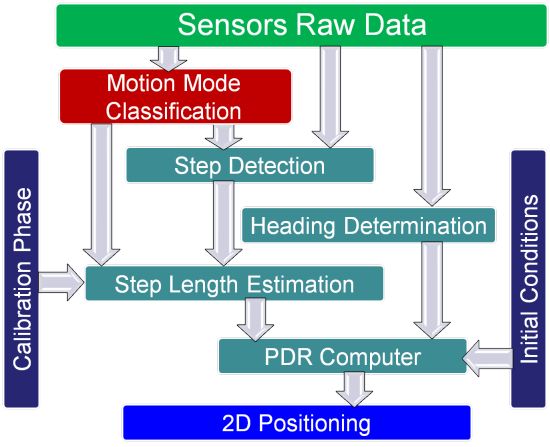

Fig. 1. Overview of pedestrian dead reckoning algorithm

approaches are described to show their dependency on a certain gain that needs to be calibrated before using any PDR algorithm.

B. Step Length Estimation

Several empirical methods for estimating the step length exist in the literature [17] [9]. Among them, we use two common approaches 1) Weinberg approach [19] and 2) Kim approach [18].

The Weinberg's approach is based on the amplitude of the vertical acceleration during the step and is given by [19]

$$s_k^1 = K_1 \left(\max a_D - \min a_D\right)^{1/4} \tag{2}$$

where K_1 is a gain which needs to be calibrated and a_D is the vertical acceleration.

Kim's approach is based on the average acceleration during the step and defined by [18]

$$s_k^2 = K_2 \left(\frac{\sum_{k=1}^{N} |a_D[k]|}{N} \right)^{1/3} \tag{3}$$

where N is the number of acceleration measurements during the step.

In both approaches, in order to find the appropriate gain the pedestrian needs to walk across a known distance. Given such a distance and number of steps made, the gain can be determined. of course, for accurate gain estimation this procedure must be repeated several times for a given pedestrian dynamics. That is, for example, a gain for normal walking and a different gain for fast walking must be calculated otherwise when walking fast and using a normal walking gain, the step length will be poorly estimated. Therefore, as illustrated in Figure 1 motion mode recognition is applied to detect the pedestrian mode and use the appropriate gain for each mode. However, it appears that not only the pedestrian modes has influence on the gain but also the smartphone modes as shown in the next section.

C. Effect of Smartphone Mode

To demonstrate the effect of the smartphone mode on the step length estimation performance and as a result on the total travelled distance we consider five different cases of smartphone modes and walking dynamics as shown in the first two rows of Table I. A distance of $21.4[m]$ was used as a reference distance, where in each smartphone mode that distance was travelled five times with constant pace. The gain to estimate the step length was found for each run. Finally, a mean gain of each scenario, that is walking and smartphone modes, was obtained for Weinberg and Kim approaches as shown in the third and forth rows of Table I. Without mode detection, a basic approach is to pick the gain of the most expected mode. Yet, this gain will increase the error in all other smartphone modes. Another approach is to use the mean gain of all possible expected modes as the only gain to be used while estimating the step length. When doing so, in each scenario an erroneous step length estimation will result and as a consequence an error in the walking distance estimation. Using Weinberg approach mean gain for all five modes a mean position error of 9% was obtained as shown in Figure 2. The same procedure using Kim's approach mean gain resulted with an error of 10% as shown in Figure 3.

TABLE I
WALKING AND SMARTPHONE MODE SCENARIOS

Walking Mode	Normal	Normal	Normal	Normal	Fast
Smartphone Mode	Talking	Swing	Texting	Pocket	Pocket
Weinberg Gain	0.242	0.259	0.218	0.209	0.266
Kim Gain	0.376	0.479	0.383	0.360	0.440

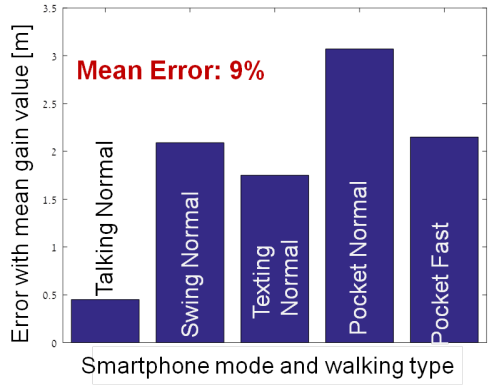

Fig. 2. Distance error for each scenario using mean gain value for Weinberg approach

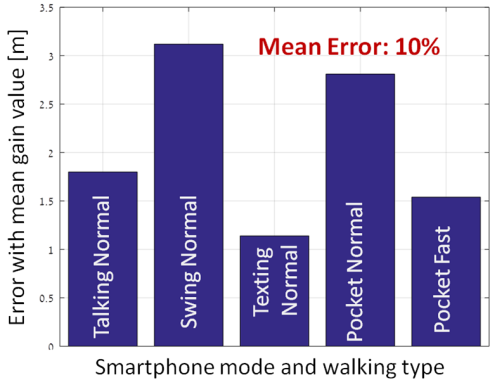

Fig. 3. Distance error for each scenario using mean gain value for Kim approach

III. Machine Learning Methodology

A. Strategy

As in our initial work [25], in this research we use a single user with a single phone for collecting sensory data used in the training process to determine an appropriate classifier. That is, our training model is solely based on a single user with single smartphone even though we believe that collecting data from multiple users and/or multiple smartphones would

probably make the classifier more robust. Initially, a single user approach was implemented due to operational reasons however, as we show in the results section, it appears that the classification performance achieved on multiple users and/or multiple phones is quite satisfactory. This learning strategy is illustrated in Figure 4. Indeed, the entire training process is made on a single user/single smartphone yet the chosen classifier is verified on multiple users with multiple smartphones.

Applying this strategy, the classification learning process

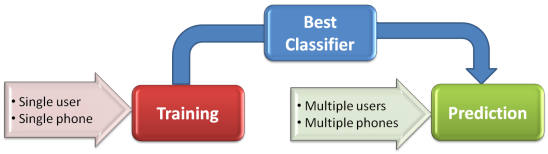

Fig. 4. Research strategy.

is presented in Figure 5. In the data acquisition phase, the collected sensory data is based only on the smartphone accelerometers (three orthogonal axes) and gyroscopes (three orthogonal axes) raw data (specific force and angular velocity, respectively). Other smartphone sensors are not used in this research, each form a different reason. Magnetometers are very sensitive to the user environment and therefore their output may deceive the classifier. The processed Barometer output yields the user altitude which may be very helpful to distinguish user modes (stairs, elevator) however not necessarily relevant to smartphone mode recognition. Light sensor are not available in all smartphones types and therefore also are not used herein. After data collection, it is preprocessed before learning is initiated. The preprocessing normally includes handling missing data, normalization, noise reduction and outliers rejection. In some situations, preprocessing activities may harm the classification performance and therefore care must be given and results are to be feedback examining performance with and without such activities. For example, normalization may sometimes cause loss of information for a given feature disabling its usage in the classifier design.

Next, features are extracted based on specific force and angular velocity vectors components and also based on their calculated vector magnitudes. The magnitude values may help overcome the sensitivity to the smartphone orientation in the user hand or pocket. Given the feature set, several machine learning classification algorithms are applied to find the appropriate one for the given problem while optimizing its parameters. The classifier is designed to distinguish between four common smartphone modes: 1) pocket, 2) swing, 3) texting and 4) talking under varying conditions such as user speed (low, normal high) user clothes (jeans, sport) and user types (gender, weight, height). The classifier is then used in prediction phase on multiple users data and outputs the accuracy of recognizing the smartphone mode.

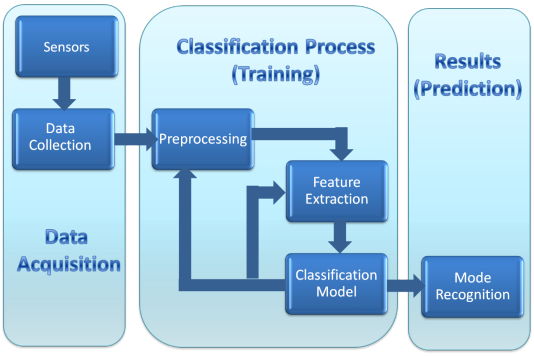

Fig. 5. Overview of the classification process for smartphone mode recognition.

B. Feature Extraction

On each working window which consists several sensor measurements (as will be defined in the following section) three types of features are calculated: 1) statistical features, 2) time-domain features and 3) cross-sensor features. In order to define the features, let the specific force vector $\mathbf{f}_j$ at time index j be defined as

$$\mathbf{f}_j = [f_{x,j}\ f_{y,j}\ f_{z,j}]^T \tag{4}$$

and the angular velocity vector ω_j at time index j be defined as

$$\omega_j = [\omega_{x,j}\ \omega_{y,j}\ \omega_{z,j}]^T \tag{5}$$

Features were calculated on each specific force and angular velocity vector components (4)-(5) and also on the magnitude of the specific force vector

$$f_m(j) = \sqrt{f_x^2(j) + f_y^2(j) + f_z^2(j)} \tag{6}$$

and the magnitude of the angular rate vector

$$\omega_m(j) = \sqrt{\omega_x^2(j) + \omega_y^2(j) + \omega_z^2(j)} \tag{7}$$

That is, features are calculated on the dataset $\mathbf{X} \in \mathbb{R}^8$ where

$$\mathbf{X} = \{x|\ f_x\ f_y\ f_z\ f_m\ \omega_x\ \omega_y\ \omega_z\ \omega_m\} \tag{8}$$

1) Statistical Features:

- **Mean**. The mean of a signal $\mathbf{x}$ of length n is defined as

$$\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j \tag{9}$$

- **Median**. The median x_m is the middle value separating the higher half of a data sample $\mathbf{x}$ from the lower half. Let $\mathbf{x}_s$ be an ascending sorted vector of $\mathbf{x}$ and j be an element index. Thus the median is defined by

$$x_m = \begin{cases} \mathbf{x}_s(j/2+1/2) \text{ if } j \text{ is odd} \\ \frac{\mathbf{x}_s(j/2)+\mathbf{x}_s(j/2+1)}{2} \text{ if } j \text{ is even} \end{cases} \tag{10}$$

- **Standard deviation**. The square root of the variance (measure of the spread of data around the mean) is defined as

$$\sigma_x = \left[\frac{1}{n-1} \sum_{j=1}^{n} (x_j - \bar{x})^2 \right]^{1/2} \tag{11}$$

- **Average Absolute Difference**. Measure of the spread of data around its mean, taking the absolute difference between values and the mean.

$$AAD_x = \frac{1}{n} \sum_{j=1}^{n} |x_j - \bar{x}| \tag{12}$$

- **Interquartile range**. It is the difference between 75th percentile and 25th percentile of the data where percentile of $Y\%$ is the value separating the higher $100 - Y\%$ of a data sample from the lower $Y\%$ of the data.
- **Skewness**. A measure of the asymmetry of the probability distribution of a signal.

$$skew_x = \frac{\left[\frac{1}{n} \sum_{j=1}^{n} (x_j - \bar{x})^3 \right]}{\left[sqrt \frac{1}{n} \sum_{j=1}^{n} (x_j - \bar{x})^2 \right]^3} \tag{13}$$

- **Kurtosis**. A measure of the tailedness of the probability distribution of a signal.

$$kurt_x = \frac{\left[\frac{1}{n} \sum_{j=1}^{n} (x_j - \bar{x})^4 \right]}{\left[sqrt \frac{1}{n} \sum_{j=1}^{n} (x_j - \bar{x})^2 \right]^2} \tag{14}$$

- **Energy**. The sum of the squares of signal values.

$$ENE_x = \sum_{j=1}^{n} x_j^2 \tag{15}$$

- **Magnitude Area**. The sum of absolute values of a signal.

$$SMA_x = \sum_{j=1}^{n} |x_j| \tag{16}$$

- **Maximum Value**. The maximum value in the window of the signal.
- **Minimum Value**. The minimum value in the window of the signal.
- **Amplitude**. The absolute difference between the maximum value and minimum value.

$$AMP_x = |\max \mathbf{x} - \min \mathbf{x}| \tag{17}$$

2) Time-Domain Features:

- **Number of peaks**. The count of the number of maximum points within the desired window of the signal where the maximum points should be above a predefined value and located after w samples from the last maximum point.
- **Median-crossing rate**. A count of how many times within a window the signal crosses its median value.
- **g-crossing rate**. A count of how many times within a window the accelerometer signal crosses the gravity value.

3) Cross Sensor Features: let $\mathbf{x}$ represent a single member of the accelerometer components in the dataset (8) and $\mathbf{y}$ represent single member of the gyroscope components in the dataset (8). The cross-sensor features, calculated for both magnitudes and vector components, are:

- **Gyroscope-Accelerometer Correlation**. Is the cross-correlation coefficient between the gyroscope and acceleration sensors defined by

$$\rho_{x,y} = \frac{\sum_{j=1}^{n} (x_j - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_{j=1}^{n} (x_j - \bar{x})^2 \sum_{j=1}^{n} (y_j - \bar{y})^2}} \tag{18}$$

- **Gyroscope-Accelerometer Maximum**. The multiplication result of the gyroscope and acceleration maximum values.

$$GAM_{x,y} = \max(\mathbf{x}) \max(\mathbf{y}) \tag{19}$$

- **Gyroscope-Accelerometer Standard Deviation**. The multiplication result of the gyroscope and acceleration standard deviation values.

$$GAS_{x,y} = \sigma_x \sigma_y \tag{20}$$

- **Gyroscope-Accelerometer Amplitude**. The multiplication result of the gyroscope and acceleration amplitudes

$$GAA_{x,y} = AMP_x AMP_y \tag{21}$$

To summarize, we have a total of 111 features used in the analysis where all of the statistical and time-domain features where calculated for each member in the dataset (specific force and angular velocity magnitudes and components) and cross sensor features where calculated between the specific force and angular velocity magnitudes and components.

C. Classification Algorithms

A brief description of five classification approaches used in this work for smartphone mode recognition is presented.

1) K-nearest Neighbors: KNN is refereed as a lazy learner [27] because it does not learn a discriminative function from the training data but memorizes the training dataset instead. The algorithm can be summarized by the following steps: a) choose the number of k and a distance metric, b) find the k nearest neighbours of the sample that we want to classify and c) assign the class label by majority vote. Besides its simplicity the advantage of KNN is an immediate adaptation of new training data while its major disadvantage is its computational complexity.

2) Multilayer Perceptron: (MLP) is a feedforward neural network consisting of three layers: input, hidden and output. The units of the hidden layer are fully connected to the input layer and the output layer is fully connected to the hidden layer.

3) Support Vector Machine: (SVM) objective is to maximize the margin, which is defined as the distance between the separating hyperplane (decision boundary) and the training samples that are closest to this hyperplane [27], [26]. SVM usually tend to have high accuracies however, SVM require significant processing power.

4) Random Forest: (RF) can be considered as an ensemble of decision trees [28]. It consists of a large group of decision trees with each decision tree classifying a subset of the data, and each node of each decision tree evaluates a randomly chosen subset of the features. In evaluating a new data sample, all the decision trees attempt to classify the new data sample and the chosen class is found by majority voting [26]. RF is useful in handling data sets with a large number of features or unbalanced data sets or data sets with missing data.

5) Gradient Boosting: (GB), from the family of boosting algorithms, uses a set weak learners, typically decision trees, to make predictions and an additive model to add the week learners to minimize the loss function[29], [26] and thereby turn them into a strong learner.

IV. Experimental Results and Discussion

A. Experimental Setup

Both in the training and prediction phases the accelerometer and gyroscope data were collected in a sampling rate of 25-30Hz. After acquiring the sensors raw data, their corresponding magnitudes were calculated using (6)-(7). In the preprocessing part, an outliers rejection algorithm was applied to remove samples which are over 3 standard deviations from the signal mean in each component of the feature set (8). It is noted, that normalization was also made but it turnout it causes loss of information and makes the classification task much more difficult. Therefore, no normalization was made on the data. In the same manner, no noise reduction algorithms were applied. In the training phase, for a single user, optimal parameters could be found in order to reduce the noise from the recorded data. However, the same parameters did not apply to all other users in the prediction phase and for each user different optimal parameters were found. Since in real life problems the test set is not always available, noise reduction parameters cannot be determined a priori. Therefore we decided not to use any noise reduction algorithms also in the learning phase to make the classier robust to noise.

On the remaining data (after outliers rejection) a sliding window with constant length of T seconds was applied with overlapping of all samples except one. The features were calculated using all samples in the window. We examined cases where the window width varied from 1 second to 6 seconds.

The sensory data for the training phase was collected from a single smartphone with a single user during four smartphone modes: 1) pocket, 2) swing, 3) texting and 4) talking. Inhomogeneous environment and smartphone modes were used to collect the data required for the training and testing phases. The environment includes: varying walking speeds, walking on uneven pavements, walking on roads with road humps, transitions between pavement and roads. The smartphone modes includes: varying hand swing (small to big), various location of the smartphone in pocket, texting and talking with a single hand in different positions relative to the user. For the prediction phase, the same data collocation approach was applied only for different users including ten men and three women. The total number of samples in each mode is presented in Table 1 for the training and prediction datasets.

TABLE II
Number of samples used for each smartphone mode in the training and prediction datasets

Mode	Training samples	Prediction samples
Pocket	37092	33485
Swing	58212	9651
Talking	26697	7781
Texting	24277	32504

B. Analysis and Experimental Results

We examined five types of machine-learning classifying algorithms as described in Section III-C. All classifiers where trained on the training dataset with a five fold validation approach. Accuracy was chosen as the performance measure in the presented analysis. Accuracy is the measure of the proportion between all cases which have been correctly classified out from the total cases. After parameter tuning for all classifying algorithms, the best performance classifiers of each approach was tested with the test dataset to examine its abilities.

C. Influence of Dataset Length

In this part, we set the window width to be $T = 5$ seconds. Our aim was to examine if features should be calculated on the complete dataset (8), or on partial set from it. The motivation is to find a minimum dataset which enables to obtain satisfactory performance - more than 90% accuracy. Since a minimum dataset can gain insight into the parameters involved in the classification process and reduces computational load.

To that end, three difference datasets where examined. Set 1, consists only of the specific force magnitude value (6) with all statistical and time-domain features. This case, Set 1, represents the minimum set of data. In the next set, Set 2, we used in addition to the specific force magnitude value (6), also the angular velocity magnitude value (7). The addition of the angular velocity magnitude enables the usage of cross sensor features. Lastly,in Set 3, the complete dataset (8) including specific force and angular velocity components and magnitudes where used. This case represents the maximum usage of the collected data and features. The accuracy results of the three sets are given in Table III. For all classifiers, as the dataset was increased, first to include the angular velocity features (Set 2) and next to include the vector components (Set 3), the performance was also increased. In particular, RF and GB managed to achieve the desired result of more than 90% accuracy. When examining each mode performance of both RF and GB classifiers, it was found that most errors occurred when the true smartphone mode was in pocket while the classifier recognized it as in a swing mode. This may occur due to the varying clothing of the users since in sports pants, usually, while in pocket the smartphone still has a huge amount of movement unlike jeans where the pocket holds the smartphone tight.

TABLE III
ACCURACY OF THE THREE SETS TESTED ON FIVE DIFFERENT CLASSIFIER TYPES

Classifier	KNN	MLP	SVM	RF	GB
Set 1 Accuracy %	44.2	47.8	43.6	63.7	56.8
Set 2 Accuracy %	58.8	63.2	53.9	62.3	57.6
Set 3 Accuracy %	58.5	65.8	57.1	90.0	91.1

D. Influence of Window Width

Trying to avoid the misclassification between pocket and swing modes, we examine the influence of window width arguing that smaller width may help the classifier to overcome this error. The underlying assumption is that in a smaller window width the unique characteristic of each mode are more significant and therefore easier to identify, in particular in an inhomogeneous environment and smartphone modes the data was recorded . Based on the results presented in Table III we limit the discussion only to RF and GB classifiers as they obtained the best performance (the other classifiers where also examined but, as before, produced poor results compare to RF and GB). The accuracy of classifying the correct smartphone mode as a function of the window width is given in Table IV. As can be seen, for window width between 1

TABLE IV
RF AND GB ACCURACY AS A FUNCTION OF THE WINDOW WIDTH

Window Width [sec]	1	2	3	4	5	6
RF Accuracy %	94.3	94.4	94.4	94.1	90.0	86.5
GB Accuracy %	95.1	95.4	95.3	95.2	91.1	88.6

to 4 seconds the accuracy of each classifier is almost the same. As the window width is further increased, the performance is decreased. The best accuracy of 95.4% was achieved by GB for a window width of 2 seconds. This results supports our initial claim that in shorter window width the unique characteristics of each mode are more significant and also the inhomogeneous environment effects have less influence. The resulting confusion matrix is presented in Figure 6. There, it can be seen that the misclassification between pocket and swing modes, described in Section IV-C, was solved. The following parameters where used in the GB classifier: number of estimators = 80, learning rate = 0.1, minimum samples split = 500, min samples leaf = 50, maximum depth = 8, subsample = 0.8 and maximum features were chosen using log2 method.

E. Best Classifier Analysis

Using the best classifier to the problem at hand, as obtained in Section IV-D, we further looked into three different problems to examine its robustness and means to improve its performance. The first, was to examine the performance when changing the number of overlapping samples between two consecutive windows. In this work, we added only one new sample to each window as explained in Section IV-A.

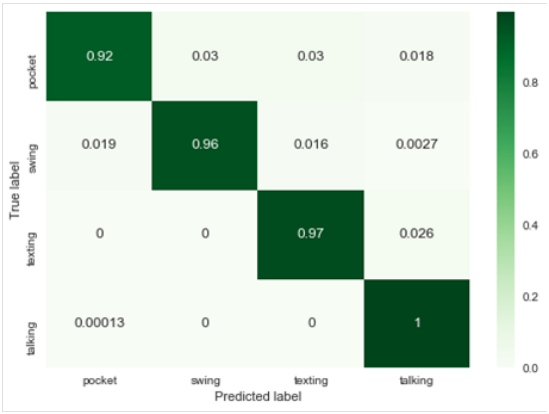

Fig. 6. Confusion matrix presenting the accuracy of GB classifier in classifying the four smartphone modes

Yet, it turn out that the number of overlapping samples in two consecutive windows had almost no effect on the classifier accuracy and therefore this issue is not elaborated here.
The second, was to use the inherit feature importance attribute of GB classifier and to perform feature selection. By applying feature selection and obtaining a smaller feature set, more intuition can be gained in understanding the dominant features behaviour. Also, a reduce feature set may yield better performance since some features may have a negative effect on the model. Moreover, a reduce set may avoid the problem of overfitting the model. The total number of features used in the analysis was 111. Figure 7 presents a feature importance bar of the 20 most dominate features. The most dominant feature was the magnitude area of the y-axis accelerometer however, due to the amount of features used the difference between the feature importances is very small. To chose a reduced

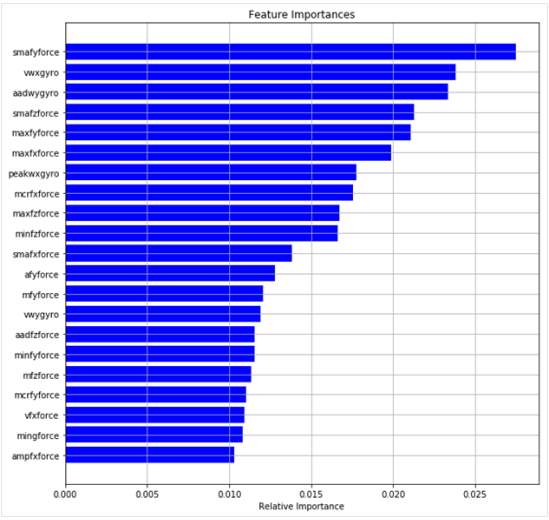

Fig. 7. Feature importance bar of the 20 most dominate features

feature set, a varying threshold on the feature importance

was made to select the number of features as presented in Figure 8. When using the three most dominant features an accuracy of 60% was achieved. Increasing the number of features used to 56, which is about half of the original set, an accuracy of 95% was achieved compared to 95.4% of the whole feature set. Also when, observing the curve in 8, using a 12 feature set, which is about 10% of the original feature set, managed to obtain accuracy of 93.8% which is very close to the performance of the original set. These 12 feature set are the 12 dominant features as presented in Figure 7.Notice, that the 12 features where calculated on the three accelerometer components (x, y, z) and two gyroscope components (x, y) reducing the complete set 8 into a reduced set of five dimensions

$$\mathbf{X}_R = \{x|\ f_x\ f_y\ f_z\ \omega_x\ \omega_y\} \tag{22}$$

instead of eight. Particularly, in the reduced set (22) the magnitudes of the specific force and angular velocity is not required, Our next goal is then, to design a new GB classifier using the reduced feature set (22) and obtain the same performance as with the complete feature set (8). After a new learning and testing processes, we managed to find a new classifier which obtained the same accuracy as with the complete 111 feature set, but only now with only 12 features.The following parameters where used in the new GB classifier: number of estimators = 80, learning rate = 0.1, minimum samples split = 500, min samples leaf = 50, maximum depth = 3, subsample = 0.58 and maximum features were chosen using log2 method. Finally, to examine the best classifier robustness we used it as

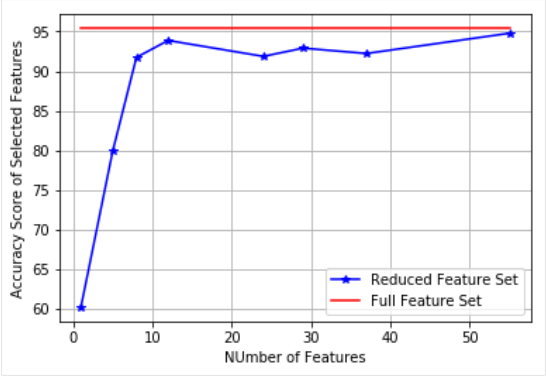

Fig. 8. Reduced feature set accuracy applied on the test dataset

is, on a different test dataset obtained using two participators while going up and down stairs with all four smartphone modes being applied. That is, a major change in the dynamics of the pedestrian. Notice, that the classifier was not trained to include stairs dynamics and we wanted to examine if it can handle a new type of pedestrian dynamics and accurately classify the smartphone modes. This examination was made even though it is well understood that if the classifier will be trained with stairs data it will be able to better cope with it when appearing in the test dataset. Our aim was to conduct an initial examination of the classifier robustness to untrained user dynamics. To that end, the same training set as in Table II with the same classifier as obtained in Section IV-D was used. The number of samples used in the new test dataset is presented in Table V. The accuracy performance in terms of the confusion

TABLE V
NUMBER OF SAMPLES USED FOR EACH SMARTPHONE MODE IN THE STAIRS PREDICTION DATASET

Mode	Pocket	Swing	Talking	Texting
Prediction samples	12837	4043	2535	8183

matrix for the stairs dataset is shown in Figure 9. Using the same classifier, an accuracy of 94.3% was achieved.

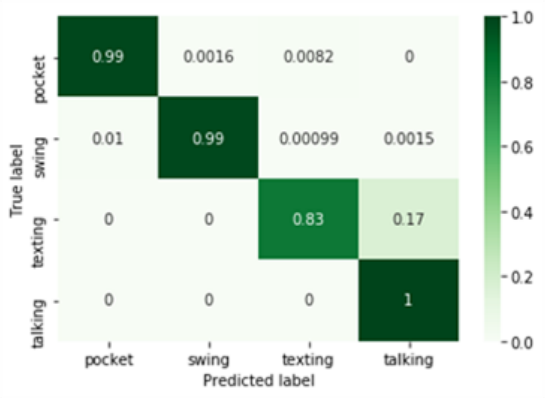

Fig. 9. Confusion matrix presenting the accuracy of GB classifier in classifying the four smartphone modes for the stairs dataset

V. CONCLUSIONS

The influence of gain selection based on the smartphone mode, for accurate step-length estimation and thus positioning accuracy of PDR algorithms was demonstrated. A methodology for classifying smartphone modes for improved PDR was proposed. A methodology of training on a single user with a single smartphone and testing on data collected from multiple users and smartphones was implemented and managed to achieve an accuracy above 95%. Both training and testing datasets were recorded in an inhomogeneous environment. The environment includes: varying walking speeds, walking on uneven pavements, walking on roads with road humps, transitions between pavement and roads. The smartphone modes includes: varying hand swing (small to big), various location of the smartphone in pocket, texting and talking with a single hand in different positions relative to the user. After determining an appropriate window width, a feature selection process was carried out resulting in a reduced feature set of 12 features instead of the original set of 111 features. In that manner, a reduced dataset containing only the three accelerometer components (x, y, z) and two gyroscope components (x, y) was used, making the calculations of the accelerometer and gyroscope measurements magnitudes unnecessary. After tuning

the classifier, the reduced feature set managed to achieve the same performance as the original one. An initial examination proved the robustness of the classifier to unfamiliar pedestrian dynamics when the original test dataset including walking dynamics was replaced by a new stairs dynamics only dataset.

REFERENCES

[1] Rainer, M. *Indoor positioning technologies*, Ph.D. thesis, Swiss Federal Institute of Technology Zurich, Switzerland, 2012.

[2] Skog, I., Nilsson, J. O., and Handel, P. *Pedestrian tracking using an IMU array*, In Electronics, Computing and Communication Technologies (IEEE CONECCT), 2014 IEEE International Conference on, pp. 1–4, 2014.

[3] Foxlin, E. *Pedestrian tracking with shoe-mounted inertial sensors*, IEEE Computer graphics and applications, Vol. 25(6), pp. 38–46, 2005.

[4] Cliff, C., Randell, D. and Muller, H. L. *Personal position measurement using dead reckoning*, in Proceedings of the Seventh International Symposium on Wearable Computers, IEEE Computer Society, October 2003, pp. 166-173.

[5] Beauregard, S. and Haas., H. *Pedestrian Dead Reckoning: A Basis for Personal Positioning*,. In Proc. of WPNC, 2006.

[6] Zampella, F., De Angelis, A., Skog, I., Zachariah, D. and Jimenez, A. *A constraint approach for UWB and PDR fusion*, In Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on, pp. 1–9, 2012.

[7] Li, Y., Zhuang, Y., Lan, H., Zhou, Q., Niu, X., and El-Sheimy, N. *A hybrid WiFi/magnetic matching/PDR approach for indoor navigation with smartphone sensors*, IEEE Communications Letters, Vol. 20(1), pp. 169–72, 2016.

[8] Chen, Z., Zou H., Jiang H., Zhu, Q., Soh, YC. and Xie, L. *Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization*, Sensors, Vol. 15(1), pp. 715–32, 2015.

[9] Pratama, AR. and Hidayat, R. *Smartphone-based pedestrian dead reckoning as an indoor positioning system*, In System Engineering and Technology (ICSET), International Conference on, pp. 1–6, 2012.

[10] Beauregard, S. *A helmet-mounted pedestrian dead reckoning system*, I Applied Wearable Computing (IFAWC), 2006 3rd International Forum on, pp. 1–11, 2006.

[11] Jirawimut, R., Ptasinski, P., Garaj, V., Cecelja, F. and Balachandran, W, *A method for dead reckoning parameter correction in pedestrian navigation system*, IEEE Transactions on Instrumentation and Measurement, Vo. 52(1), pp. 209-15, 2003.

[12] Jimnez, AR., Seco, F., Prieto, JC., and Guevara, J., *Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU*, In Positioning Navigation and Communication (WPNC), 7th Workshop on, pp. 135-143, 2010.

[13] Godha, S. and Lachapelle, G. *Foot mounted inertial system for pedestrian navigation*, Measurement Science and Technology, Vol. 19(7), pp. 075202, 2008.

[14] Groves, P. D. *Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems*, Second Edition, Artech House, 2013.

[15] Qian, L., Ma, J., Ying, R., Liu, P. and Pei, P. *An improved indoor localization method using smartphone inertial sensors*, International Conference on Indoor Positioning and Indoor Navigation(IPIN), pp. 1–7,2013.

[16] Susi, M., Renaudin, V. and Lachapelle, G. *Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users*, Sensors vol. 13, pp. 1539-1562, 2013.

[17] Sayeed, T., Sam, A., Catal, A. and Cabestany, J., *Comparison and adaptation of step length and gait speed estimators from single belt worn accelerometer positioned on lateral side of the body*, In Intelligent Signal Processing (WISP), 2013 IEEE 8th International Symposium on, pp. 14–20, 2013.

[18] Kim, J. W., Jang, H. J., Hwang, D-H. and Park, C., *A Step, Stride and Heading Determination for the Pedestrian Navigation System*, Journal of Global Positioning Systems, pp. 273–279, 2004.

[19] Weinberg, H., *Using the ADXL202 in Pedometer and Personal Navigation Applications*, Analog Devices AN-602 Application Note, 2002.

[20] Elhoushi, M., Georgy, J., Noureldin, A. and Korenberg, M. *Online motion mode recognition for portable navigation using low-cost sensors*, NAVIGATION Journal of the institute of navigation, vol. 62, no. 4, pp. 273-290, 2015.

[21] Elhoushi, M., Georgy, J., Noureldin, A. and Korenberg, M. *A Survey on Approaches of Motion Mode Recognition Using Sensors*, IEEE Transactions on intelligent transportation systems, Vol. 18 (7), pp. 1662–1686, 2017.

[22] Zhang P., Chen X., Ma X., Wu Y., Jiang H., Fang D., Tang Z. and Ma Y., *SmartMTra: Robust Indoor Trajectory Tracing Using Smartphones*, IEEE SENSORS JOURNAL, VOL. 17(12), pp. 3613–3624, 2017.

[23] Tian Q., Salcic Z., Wang K. and Pan Y., *A Multi-Mode Dead Reckoning System for Pedestrian Tracking Using Smartphones*, IEEE SENSORS JOURNAL, VOL. 16 (7), pp. 2079–2093, 2016.

[24] Kumar A., Gowda N. S., Babu R. S. and Sekaran C. D., *UMOISP: Usage Mode and Orientation Invariant Smartphone Pedometer*, IEEE SENSORS JOURNAL, VOL. 17(3), pp. 869–881 , 2017.

[25] Klein, I., Solaz, Y. and Ohayon, G. *Smartphone Motion Mode Recognition*, the 4th International Electronic Conference on Sensors and Applications (ECSA 2017), 2017.

[26] Hastie, T., Tibshirani, R. and Friedman, J. *The Elements of Statistical Learning, Data Mining, Inference and Prediction*, Second Edition, Springer, 2009.

[27] Raschka, S. *Python Machine Learning*, Packt publishing, 2016.

[28] Breiman L., *Random Forests*, Machine learning, Vol. 45(1), pp.5–32, 2001.

[29] Friedman J. H. *Greedy function approximation: A gradient boosting machine*, Annals of statistics, pp. 1189-1232, 2001.