

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Identification of walker identity using smartphone sensors: an experiment using ensemble learning

ANTONIO ANGRISANO¹, MARIO LUCA BERNARDI², MARTA CIMITILE³, SALVATORE GAGLIONE⁴, AND MARIO VULTAGGIO⁵,

¹GIUSTINO FORTUNATO UNIVERSITY, ITALY (E-MAIL: A.ANGRISANO@UNIFORTUNATO.EU)

²GIUSTINO FORTUNATO UNIVERSITY, ITALY (E-MAIL: M.BERNARDI@UNIFORTUNATO.EU)

³UNITELMA SAPIENZA UNIVERSITY, ROME, ITALY (E-MAIL: MARTA.CIMITILE@UNITELMASAPIENZA.IT)

⁴PARTHENOPE UNIVERSITY, NAPLES, ITALY (E-MAIL: SALVATORE.GAGLIONE@UNIPARTHENOPE.IT)

⁵GIUSTINO FORTUNATO UNIVERSITY, ITALY (E-MAIL: M.VULTAGGIO@UNIFORTUNATO.EU)

Corresponding author: Marta Cimitile (e-mail: marta.cimitile@unitelmasapienza.it).

ABSTRACT Nowadays MEMS sensors, like accelerometers, gyroscopes, and magnetometers, are spreading in a wide range of applications, because of their small size, cheapness and increasing performance. For instance, smartphones are currently equipped with this kind of sensors, which could be used to improve the user experience of the phone itself or the navigation functionalities. In this work, accelerometers, gyros, and orientation measurements are exploited to provide advanced information about the walker bringing the phone. In particular, smartphone sensors outputs are used to recognize the identity of the walker and the pose of the device during the walk. The aforesaid information, if known, could be used to improve specific smartphone functionalities. For instance, the recognition of walker identity can be used for theft protection or the device pose can be used to improve the performance of the pedestrian navigation. Machine learning algorithms have been effectively adopted in several fields to solve problems involving classification, time series prediction, pattern recognition, and object detection. Herein, a novel hierarchical approach for classification is applied to data produced by smartphone sensors in order to recognize the previously described contexts, obtaining effective results.

INDEX TERMS Navigation; Activities detection; Machine learning; MEMS sensors

I. INTRODUCTION

MEMS (Micro-Electro-Mechanical Systems) are small, light and cheap sensors, whose spread is continuously growing in several sectors (Bogue, 2013). Pressure and inertial sensors are largely used in the automotive industry since the nineties, for engine management, car dynamics control, and safety systems. MEMS had a further proliferation in the consumer electronics since the mid-2000s, in particular for smartphones and tablets (Lammel, 2015); in these devices, inertial sensors are used, for instance, for automatic screen rotation or recognition of gesture-based command. Currently, fitness trackers, smartwatches and virtual reality headsets further enrich the market of MEMS.

All smart-devices (phone, tablet, clocks) have an increasing number of embedded sensors that measure motion, orientation and several environmental parameters. As a result, smart-devices could be considered very powerful mobile sensor platforms (Lane et al., 2010). These sensors are

enabling new applications across a wide variety of domains, such as navigation, healthcare, IoT, safety, environmental monitoring and they are giving rise to a new area of research called mobile phone sensing. The outputs of smart-devices sensors are raw data with a high rate that directly provide information on a physical quantity (physical or hardware sensors) or that compute a quantity estimate by processing several measure sources (virtual or synthetic or software sensors). Three broad sensor categories could be considered: motion, environmental and position sensors (Yan et al., 2014; Addabbo et al., 2016). The first category includes physical sensors as accelerometers and gyroscopes, which can measure acceleration and angular velocity, while, as virtual sensors, gravity, linear acceleration, step detector, and counter. Environmental sensors (barometers, photometers, and thermometers) measure various environmental parameters, such as ambient air temperature and pressure, illumination, and humidity. The last category refers to sensors

providing the physical position of a device like GNSS, magnetometers and proximity sensor; the device orientation is usually provided by a virtual sensor, which is based on a fusion of accelerometers, gyroscopes and magnetometers measurements (Fareed, 2015).

The scope of this research is to recognize, within a set of possible walkers, the walker that is currently bringing the smartphone using the signals from sensors embedded into the device (i.e., accelerometers, gyroscopes, and magnetometers). The identification of the walker is the main purpose of the work and, to our best knowledge, there is no significant previous work on the topic. Its natural application is the theft protection, i.e. a smartphone with this functionality would be able to detect if the person bringing it is the owner or not. The pose of the smartphone during the walk, i.e. how the device is carried, is recognized too. Previous researchers (Susi et al., 2012; Pei et al., 2012; Kamisaka et al., 2011; Zhang et al., 2015) worked on this topic, mainly because knowing the device pose could improve pedestrian navigation performance. Differently from the above approaches, in this study, we mainly focus on the topic of walker identification and device pose recognition. Specifically, the approach described in this study aims at identifying the device pose and the walker's identity with a hierarchical ensemble of classifiers using a set of internal base classifiers to improve the overall performances. According to this, here we focus on comparing the performances of a base classifier with a hierarchical one that combines two base classifiers: a device pose classifier and a walker identity classifier.

Moreover, data from eighteen walkers (9 males and 9 females), with two device models (i.e. iPhone 7 and Samsung S6) in four different poses (phoning, pocket, texting, and hand as defined in the methodology section), are used to train the hierarchical learning classifiers; after the training step, the resulting classifiers are tested for the identification, in real-time, of both the actual walker among the possible ones and the device pose. The classifiers are assessed on real data collected from walking sessions and exhibit very effective identification performances. The percentage of correct recognition of device pose is about 98%. The performance in recognizing the walker's identity, regardless of the device model and pose, is very good but not always satisfying (about 93% of correct classifications). Consequently, a hierarchical approach, to first identify the device pose and then the walker's identity with two cascaded classifiers of different kinds, is carried out. With this approach, the accuracy of successful identifications drastically increases to at least 98.72%. Also, to understand the role of each considered sensor in the identification process, all the combinations of accelerometers, gyroscopes and orientations are tested; the analysis demonstrates that the sensor more incident for the identification is orientation, while the less incident is the gyroscope. Finally, the effect of time window size on the identification performance is analyzed too. With just less than half a second of data, it

is possible to obtain very good identification performance (with an accuracy of 93.8% and 98.2% respectively for simple and hierarchical classifiers). Further improvements are obtained by increasing the window size: processing five or more seconds of measurements allows to obtain $\approx 100\%$ of correct identifications when subjects are all known (i.e., the classifier has been trained on walking sessions produced by all potential walkers). To study the behavior when subjects are not known in advance, a case study to assess the performance of the hierarchical classifier in a real binary classification scenario (to discriminate between the "owner" or a "stranger") is proposed. The hierarchical classifier exhibits good generalization capabilities providing an accuracy above 92.5% with windows of two seconds and the best accuracy of 97.5% for windows of at least six seconds.

II. BACKGROUND ON DECISION TREE CLASSIFICATION

Classification algorithms aim to select from a set of categories, the category to which a new observation belongs. This section presents a short description of the decision tree classification algorithms adopted in this study.

A. MACHINE LEARNING ALGORITHMS

Machine Learning (ML) is a subset of the artificial intelligence discipline aiming at realizing systems that can learn how to behave from data (Mitchell, 1997). ML techniques are classified into two main categories: supervised learning and unsupervised learning. The difference between these categories is how the learning process is performed and what kind of information it needs. In particular, supervised learning algorithms infer a function that maps a set of input data and the desired output (it is called "training" dataset). The function, after inferred, can be used to map new observations of the phenomenon under study. Supervised learning can be effectively adopted in classification problems to identify the class labels for new observations. In this case, the learning process consists of selecting, among all the possible functions, the best one capable to identify correct class labels for unseen input data (i.e., data not included in the training set). When conducting supervised learning, a critical aspect is model complexity. Usually, a low-complexity model is advisable to allow the system to make correct predictions on new samples (i.e., it is capable of "generalize"). Usually, high-complexity learned models are said to "over-fit" since they are too much linked to the specific instances of training samples. Such models are not able to generalize, often performing well on the training/test data but exhibiting bad performances on new samples that are never used in the learning process. Unsupervised learning differs from supervised learning because it does not need to be trained with true class data (no explicitly-provided labels are used). In unsupervised learning, the model is explicitly defined and, usually, it is characterized by several parameters. The learning process, in this case, consists of

the estimation of the parameters based on real data. This section explores the supervised learning algorithms since the proposed approach is based on this kind of classifier.

Supervised learning approaches can be grouped in the following broad categories:

- Decision Trees (DT): (Mihăescu *et al.*, 2015; Iba and Langley, 1992; Hoeglinger and Pears, 2007; Breiman, 2001; Zhao and Zhang, 2008; Nadiammai and Hemalatha, 2013; Landwehr *et al.*, 2005),
- Bayesian Networks (BN): (Friedman *et al.*, 1997; Lee and Shimoji, 1993; John and Langley, 1995; McCallum and Nigam, 1998; Kibriya *et al.*, 2004),
- Multinomial Logistic Regression (MLR) and Regression based Classifiers (RC): (Freedman, 2009),
- Rules (R): (Kégl, 2013; Aha and Kibler, 1991),
- Lazy (L): (Aha and Kibler, 1991).

Starting from the analysis of the above-mentioned approaches, we decide to focus on decision-tree approaches. Specifically, the proposed approach is based on the RandomForest as formulated in Breiman (2001) for its capability to combine concepts of bootstrapped aggregation (or bagging) with simple decision tree-based classifiers. Since the RandomForest algorithm has been effectively used in similar classification tasks that are focused on behavior detection and identification providing best performances among simpler classifiers, we selected it as our baseline approach.

B. RANDOM FORESTS

Random forest operates by constructing a multitude of decision trees at training time and gives as output the membership, for each observation, to a class (for a classification problem) or the mean prediction (for a regression problem). In particular, the leaf nodes of the trees correspond to reached decisions and each node of a binary decision tree corresponds to a decision criterion (hence two branches contain an alternative set of decisions based on actual data). Looking at the classification problems, decision trees are very useful since each considered metric can be represented as a different node and each leaf node represent a decision about the membership of current observation. Multiple decision trees capture different types of classification rules and are more suitable to represent complex domains. These types of decision trees are referred to as a decision forest (DF) and can be considered as an example of ensemble classifiers. For instance, in categorical problems, each tree proceeds to vote for a different class and the leaf nodes report, as answers, the percentages of trees voting for each possible class. Randomness is essential for the construction of a decision forest. In general, the random forest approaches are characterized by a random selection of the features metrics used to build decision trees and by bootstrapped aggregation that repeatedly sample data with replacement from the original training set to obtain multiple separate training sets as described in (Breiman, 1996). The trees

built can be trained by using the CART methodology as shown in (Russell and Norvig, 2016). It consists of a metrics selection step based on information theory. The number of built tree and the maximum depths of the decision tree can be monitored thought some parameters (some of these parameters can be set by cross-validation to maximize performances). A limitation of an individual decision tree is that it produces predictions having low bias but high variance. Extremely randomized trees algorithm answers to this issue and attempts to achieve this goal by generating an ensemble of independent and uncorrelated decision trees.

III. RELATED WORK ON LEARNING ALGORITHMS FOR MOBILE SENSORS DATA

Complex activity recognition is a very explored topic (Liu *et al.*, 2018). A huge amount of studies explore the adoption of machine learning and pattern recognition to extract useful information from the mobile sensor data (San-Segundo *et al.*, 2018).

Several approaches are based on unsupervised learning algorithms since no prior information is required (Riboni *et al.*, 2016; Fiandrino *et al.*, 2017; Cardone *et al.*, 2013). Other studies explore semi-supervised learning approaches allowing to minimize the number of fully-labeled needed data (Hong *et al.*, 2016). In this work, we focus on supervised learning approaches. These approaches are largely applied to recognize human activities from a set of data extracted by using sensors (Bulling *et al.*, 2014). In Susi *et al.* (2012) decision tree algorithm is used to perform a classification among four smartphone poses by analyzing accelerometers and gyros signals. Similarly, authors in (Pei *et al.*, 2012) propose an approach useful to recognize eight common motion states during indoor navigation by using a Least Square-Support Vector Machines (LS-SVM) classification algorithm. A similar approach is proposed in Liu *et al.* (2012) where the support vector machine (SVM) is used to classify data from accelerometers and GPS sensors performing a physical activity recognition. The evaluation of the approach in a real context shows that the SVM algorithm gives higher accuracy than k-nearest neighbors and nearest neighbors algorithms. In Radu *et al.* (2016); Hammerla *et al.* (2016); Ronao and Cho (2016); Ordonez and Roggen (2016) deep learning techniques are adapted to recognize human activities on the base of the data extracted from accelerometer and gyroscope sensors. In particular, authors in Ronao and Cho (2016) achieved very high classification performance on moving activities by exploiting the inherent characteristics of human activities and 1D time-series signals. All the above-discussed approaches mainly aim to identify human activities and smartphone poses (Incel, 2015) by using sensor data and they are at the base of user identification. However, our proposal starts from the above-discussed approaches and goes to identify the walker identity and the pose of the device during the walking, basing on data revealed by smartphone sensors. In this direction go approaches proposed in Zhang *et al.* (2015);

Hoang et al. (2015). However, in Hoang et al. (2015) an SVM classifier is introduced on novel gait recognition. This method outperformed an equal error rate of 2.45% and an accuracy rate of 99.14% in terms of gait identification. In Zhang et al. (2015) human step modes and device poses are identified by processing accelerometers, gyroscopes, magnetometers and pressure sensors signals. However, the topic of walking identification is marginally discussed in the above studies (they are mainly focused on walking mode identification).

This topic is instead faced in Kwapisz et al. (2010); Lee and Lee (2016) and Hoang et al. (2015). In Kwapisz et al. (2010) a model allowing to use of accelerometer data from smartphones has been proposed for user identification. The authors propose a dataset and a feature vector. They use WEKAs J48 and Neural Network models to perform the classification obtaining respectively an accuracy of 0.909 with Neural Network and 0.84 with J48. In Lee and Lee (2016) data from both smartwatches and smartphones are used. Here, the feature vector is selected giving a higher focus on the integration of smartphone and smartwatch. Another directly related approach is reported in Singha et al. (2017), where authors use a random forest ensemble classifier to recognize users using data from phones embedded accelerometer sensors. This approach is evaluated on a dataset composed of 100 samples and shows an accuracy of 0.9679 and Area under Curve (AUC) of 0.9822. Differently from the discussed approaches, our method introduces a decision tree-based hierarchical architecture for the walker classifier that exploits a set of internal base classifiers to improve the overall performances reducing detection times. Moreover, we adopted a wider range of sensors (i.e., accelerometers, gyroscopes, and magnetometers) whereas previous studies for walker identification rely only on accelerometers. This architecture exhibits very good performance in walker recognition: using the smallest time window size the precision is equal to 0.98 and recall is equal to 0.99. Moreover, in this study, we performed an analysis of the impact, on the resulting classification performance, of:

- the set of sensors considered as features;
- the time-window size used for classification.

This impact analysis is useful to clarify which aspects are more critical for both the quality of classification and the time needed to perform it. Finally, the study includes a case study showing the application of the proposed approach in a real binary classification scenario.

IV. METHODOLOGY

This section describes the proposed method focusing on: i) the adopted features model and ii) the classification approach.

A. FEATURES MODEL

The proposed approach is based on the assumption that a set of MEMS sensors typically installed in smartphones can be

TABLE 1. Considered sets of features

Feature	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇
Acceleration <i>x</i>	●	○	○	●	●	○	●
Acceleration <i>y</i>	●	○	○	●	●	○	●
Acceleration <i>z</i>	●	○	○	●	●	○	●
Gyroscope <i>x</i>	○	●	○	●	○	●	●
Gyroscope <i>y</i>	○	●	○	●	○	●	●
Gyroscope <i>z</i>	○	●	○	●	○	●	●
Orientation <i>x</i>	○	○	●	○	●	●	●
Orientation <i>y</i>	○	○	●	○	●	●	●
Orientation <i>z</i>	○	○	●	○	●	●	●

used to capture the walking behavior of the device user and this information can be used to identify him among different users in real-time (i.e., during walking sessions).

The sensors considered as features are the accelerometer, the gyroscope triads (both physical sensors), and the orientation (virtual sensor, i.e. roll, pitch and yaw angles).

Based on such considerations, we considered the sets of features described in the first column of Table 1. In addition to the complete features model (composed by all the features available using all the device sensors), we also test some smaller feature sets (as reported in the table) to understand which features are more important to identify the walker. According to this, the last seven columns in the table reports, for each feature, the sets (from S₁ to S₇) that include it.

The first three sets are used to test the classifications performances of single sensors (i.e., S₁ for acceleration, S₂ for gyroscope and S₃ for orientation).

Sets from S₄ to S₆ are needed to test classification performances considering couple of sensors (i.e., accelerometers and gyroscope, accelerometers and orientation and gyroscope and orientation).

Finally the set S₇ contains all the sensors and represents the complete feature set.

B. CLASSIFICATION APPROACH

This section discusses the proposed classification approach to the walkers and their device pose identification using data extracted from smartphone sensors. The overall classification process is reported in Figure 1. It consists of two main sub-processes: (a) the generation of the datasets (b) the training and time-series classification. The two sub-processes will be described in the remaining of this section.

1) The datasets generation

The datasets generation sub-process is described in Figure 1-(a). The process starts with the cleaning and the normalization of the data produced by the smartphones to obtain a consistent dataset that is suitable for statistical inference. This activity consists to remove all the incomplete and wrong data values by applying a set of techniques allowing to i) fill missing values, ii) filter out the noise, and iii) correct (or remove) the inconsistent values from the dataset. This activity is very critical since real-world data tend to be rather noisy, incomplete or even inconsistent. The adopted cleaning

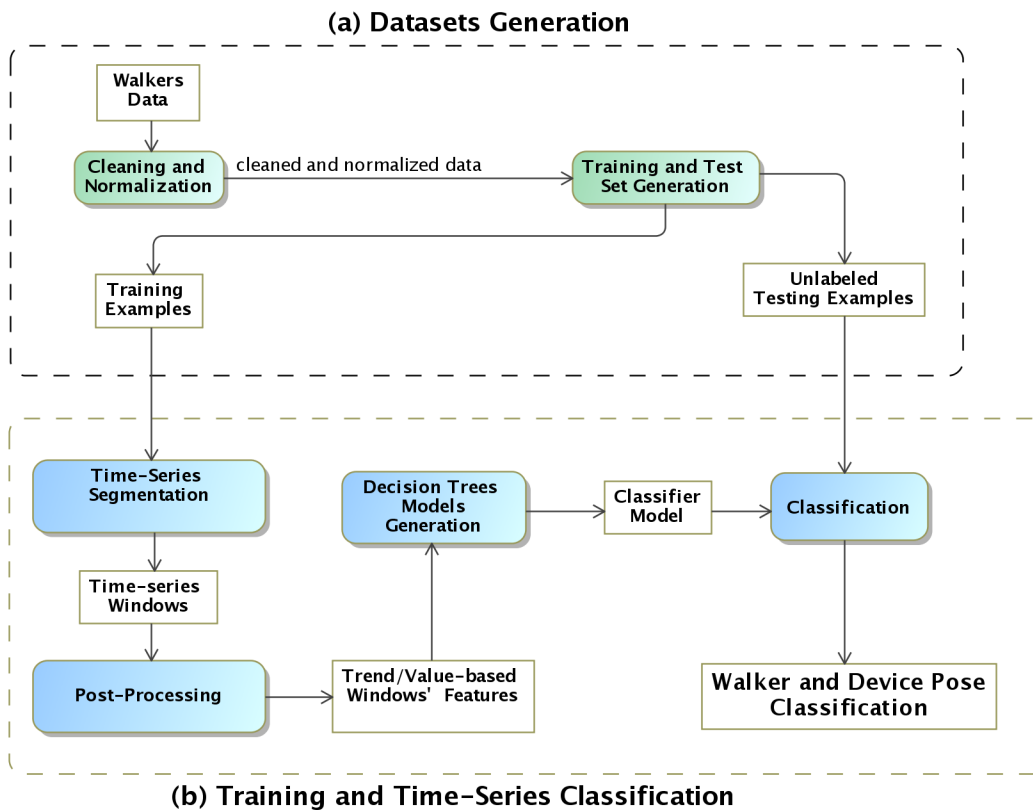


FIGURE 1. The overall classification process.

and normalization activity can be splitted in the following steps:

- fix missing values;
- remove noise;
- remove special character or values;
- verify semantic consistency;
- normalize.

The normalization is conducted by using a Min-max technique allowing to performs a linear transformation of the original smartphone data.

If \min_X and \max_X are respectively, the minimum and maximum values for the attribute X, the min-max normalization maps a value v_i of X to a v'_i in the range $\{\text{newMin}_X, \text{newMax}_X\}$ by computing:

$$v'_i = \frac{v_i - \min_X}{\max_X - \min_X} (\text{newMax}_X - \text{newMin}_X) + \text{newMin}_X$$

Finally, the cleaned and normalized dataset becomes the input for the training and test set generation activity. This activity allows splitting the data into two sets. The first is called the training set and is used to train the classifier. The second is called the test set and it is used to assess the performance of the classifier.

2) Training and Time-Series Classification

Figure 1-(b) describes the training and time-series classification sub-process. As shown in the figure, the process is mainly divided into the following steps:

- Time Series Segmentation: it consists to analyze time series and divide them into segments (i.e. time series windows);
- Post-processing: for each window it generates a representation based on values features and trends features;
- Decision Trees Model Generation: it consists to train a simple (or ensemble) decision trees-based classifier;
- Classification, in this step the trained classifier performances are tested on new time-series samples.

In the time series segmentation step, a sliding window approach is used to incrementally divide the multivariate time series into a sequence of segments across the time series values. In particular, we adopted a fixed sliding window approach consisting to define several windows of increasing range size (from 0.32 sec. to 10 sec.). In the post-processing step, for each time series window, identified during the segmentation, a set of features is evaluated. Exactly, we associate to each window w_i , the following

sequence of features:

$$(Fv_1, \dots, Fv_n, Ft_1, \dots, Ft_m)$$

The values feature Fv_j represents a discretized value of the time series contained in the $[0,1]$ range. The features Ft_k represent the trend of the time series local to the window and it can correspond to shape-based metrics (i.e., standard deviation, mean, average energy, entropy, skewness, and kurtosis). According to Esmael et al. (2012), the values feature is described by a single feature while trend features are described by two metrics: standard deviation and skewness metrics. The decision tree model generation step consists to perform classification by using a decision trees-based classifier (Breiman, 1996). The classifiers' inputs are all the vectors representing the window components (i.e., data for each sensor involved in this study). The classifier is trained using the class labels available for each set of value-based and trend-based information of each window. Finally, the trained classifier is used to classify new data and its performances are evaluated on new samples. For training the classifiers, we defined T as a set of labeled traces (M, l), where each M is associated to a label $l \in \{W_1, \dots, W_n\}$ (where W_n represents the n -th walker). For each M we built a feature vector $F \in R_y$, where y is the number of the features used in training phase ($y = 9$ for all the sensors data taken into account).

In the learning phase, the dataset assessment is performed by using a K-Fold Cross-Validation approach (Stone, 1974) consisting to split the data into k equally sized subsets using random sampling. A subset is retained as a validation dataset to assess the trained model whereas the remaining $k - 1$ subsets are exploited to perform training. Such a process is repeated $k = 10$ times: during the ten iterations, each of the k subsets has been used once as the validation dataset. To obtain a single reliable estimate, the final results are evaluated by computing the average of the results obtained during the ten iterations. The process starts by partitioning the dataset in k slices. Then, for each iteration i , we train and evaluate the effectiveness of the trained classifier following the steps reported below:

- 1) the training set $T_i \subset D$ is generated by selecting an unique set of $k-1$ slices from the dataset D ;
- 2) the test set $T'_i = D - T_i$ is generated selecting the remaining k^{th} slice (it can be evaluated as the complement of T_i to D)
- 3) a classifier is trained on set T_i ;
- 4) the trained classifier is applied to T'_i to evaluate accuracy.

Since $k = 10$, each iteration i is performed using the 90% of the dataset D as the training set (T_i) and the remaining 10% as test set (T'_i). Moreover, great representativeness of each subset is ensured by stratifying the data before being split into subsets. This model selection method, according to Kohavi (1995) provides less biased estimation of accuracy.

C. THE ADOPTED CLASSIFIERS

Different classifiers are implemented in this study:

- a generic walker classifier (C_1) able to distinguish the identity of the walkers; it can be trained, based on the needs, in two ways:
 - using walking sessions in which device is in any pose;
 - using walking sessions in which device pose is fixed;
- a device pose classifier (shortly called C_2) able to distinguish the pose of the phone (i.e., texting, pocket, phoning, swinging);
- a hierarchical classifier (C_3), obtained as the combination of one C_2 classifier to detect the pose and four C_1 classifiers (each trained using walking sessions of a fixed device pose).

The main architecture of the hierarchical classifier is reported in Figure 2. The classifier C_3 consists of two layers: in the first layer, the smartphone pose identification is performed through the C_2 classifier. In the second layer, on the base of the identified smartphone pose, the C_1 classifier for the identified device pose is used to perform the actual walker identification. As shown in Figure 2, each C_1 classifier in the C_3 is associated with a specific device pose: texting walker classifier, pocket walker classifier, phoning walker classifier, swinging walker classifier. The texting walker classifier allows identifying the walker when he is texting on the smartphone. Similarly, the pocket walker classifier allows identifying the walker when the device is in the pocket. The phoning walker classifier allows identifying the walker if he is using its device to make a phone call. Finally, the swinging walker classifier allows identifying the walker if he is freely swinging during the walk with the device in his hand. The main idea is that different walkers have different behavior for the above activities and these differences can be captured and used to improve walker identification. All the considered classifiers are preliminary tested on a partition of the data used cross-validation and, successively, on an external dataset of real walking sessions (to assess robustness).

V. EVALUATION

In this section, we describe the experiments we performed (i.e., the classification goals of each trained classifiers) and the related evaluation settings (i.e., the context of such experiments and the metrics used for validation).

A. DESCRIPTION OF THE EXPERIMENTS

The main topic of the experimentation is to evaluate the effectiveness of the proposed classifiers (C_1, C_2, C_3) to recognize the identity of a walker bringing a typical smartphone along with the device pose. Each evaluation consists to perform the classification process (as described in Section IV-B2) on the studied classifier. The evaluation is performed by considering different features models (Table 1 describes the feature models $S_1, S_2, S_3, S_4, S_5, S_6, S_7$) and different

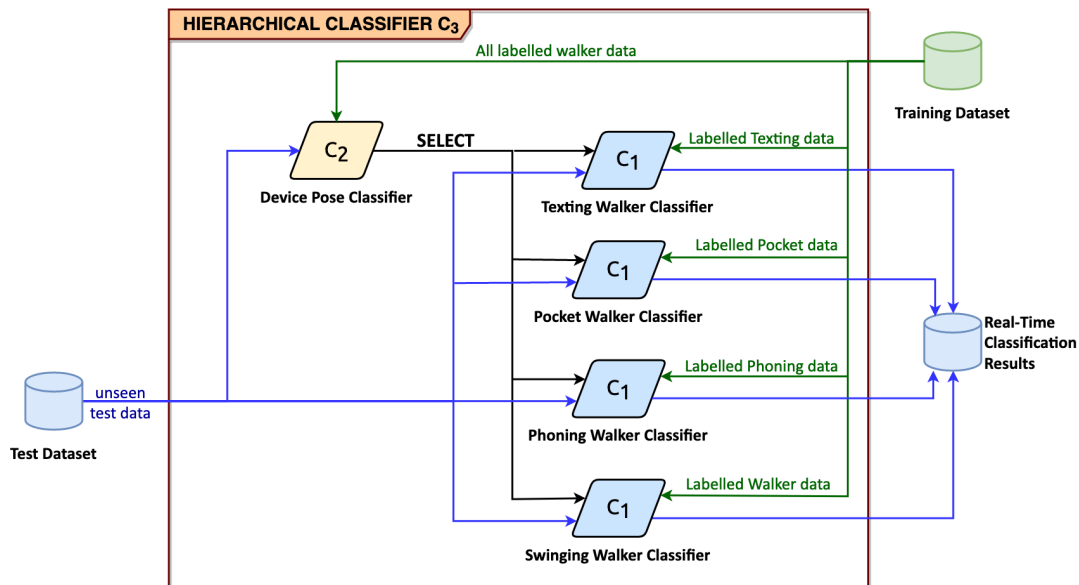


FIGURE 2. The two layers architecture of the hierarchical classifier C_3 .

time window size (as described in Section IV-B2 we adopted a fixed sliding window approach for the training and time series classification). This allows investigating the optimal time window and the best features model. According to the described goals, the following experiments are performed:

- the classifiers C_1 with the feature set S_7 is used to perform walker identification regardless of the device pose and using increasing time window sizes (ranging from 0.32 to 10 seconds);
- the classifier C_1 with different feature set ($S_1, S_2, S_3, S_4, S_5, S_6, S_7$) is used to perform walker identification regardless of device pose and using the shortest time window size (0.32 seconds);
- the classifier C_2 with the feature set S_7 is used to perform a fixed device pose identification using the shortest time window size (0.32 seconds);
- the classifiers C_3 with the feature set S_7 is used to perform walker identification regardless of the pose and using increasing time window sizes (ranging from 0.32 to 10 seconds).

Finally, further analysis has been carried out, collecting additional data from one person, included among the 18 walkers, after he was involved in an injury that caused a distorted gait (for this person we collect data of several walking sessions when he had both a limpid gait and, later on, a distorted one). The purpose of this last analysis is to understand if the classifier, trained on data related to the normal behavior of a person, can identify the subject when he walks with a distorted gait.

The classification analysis is performed by using Weka¹, a well-known framework written in Java to solve machine learning tasks.

¹<http://www.cs.waikato.ac.nz/ml/weka/>

B. EVALUATION SETTING

The training data consists of raw measurements collected from 18 different persons (9 males and 9 females). Each person walked indoor with the smartphones in 4 different poses (texting pose, phoning pose, pocket pose, swinging pose) for about 40 meters. As already pointed out, in this study we considered, as discriminating features, the accelerometer and gyroscope triads (both physical sensors), and the orientation (virtual sensor), i.e. roll, pitch and yaw angles. A data rate of 100 Hz is adopted. Two smartphones, from different brands and of different grades are used: the medium grade S6 from Samsung and the high-grade iPhone7plus from Apple. The considered data are logged using the "Matlab mobile" application from Mathworks, which allows a simple acquisition from the sensors into the smartphones and the storage into cloud memory.

TABLE 2. RandomForest selected parameters.

Hyperparameters	Values
Bootstrap	Yes
Max Depth	110
Max Features	Auto
Min Samples Leaf	2
Min Samples Split	2
Number of Estimators	1800

For what concerns the Random Forest settings, we refer to the following parameters:

- **Bootstrap**: it is a technique allowing to improve the stability and accuracy of machine learning algorithms. It also reduces variance and helps to avoid overfitting;
- **Max Depth**: maximum number of the network layers;
- **Max Features**: is the size of the random subsets of features to consider when splitting a node. If "Max

Features" is equal to "Auto", all the features are considered. Differently, Max Features can be equal to "SQRT" (meaning that "Max Features" is calculated as the square root of the total number of the features) or equal to "Log2" (meaning that "Max Features" is calculated as the logarithm to the base 2 of the total number of the features);

- **Min Samples Leaf:** minimum number of nodes that have no children;
- **Max Samples Split:** the minimum number of samples required to split an internal node;
- **Number of Estimators:** Random forest, as a meta estimator, fits several decision-trees classifiers on the sub-samples of the dataset and exploits averaging to improve accuracy and to reduce overfitting. The number of estimators is the number of such trees in the forest.

The experiment was conducted with the best parameters reported in Table 2 found using a Sequential Bayesian Model-based Optimization (SBMO) approach implemented using the Tree Parzen Estimator (TPE) algorithm as defined in (Bergstra et al., 2011).

Five known metrics have been used to evaluate the classification results: Precision, Recall, ROC AUC, Accuracy, and F1-score.

Precision (P) has been evaluated as the proportion of the examples that truly belong to the class of a specific walker among all those who were assigned to the class. The recall (R) has been evaluated as the proportion of examples assigned to the class of a specific walker among all the examples that truly belong to that class.

ROC AUC represents the degree of separability. It tells how much model is capable of distinguishing between classes.

The accuracy has been evaluated as a description of systematic errors. It is computed as the ratio of the sum of true positive and true negative to the total number of records. Finally, the F1-score is a measure that combines precision and recall in the harmonic mean.

VI. RESULTS AND DISCUSSION

The main metrics adopted to assess the performance of the classifiers are the accuracy, precision, recall, and ROC AUC; to deepen the results, the confusion matrix is considered too. The walkers are indicated by the letter "w", followed by an identification number and by the letter "m" or "f", depending on whether the walker is a male or a female; for instance, w03m indicates a male walker identified by the number 03. The selected walkers' profiles involved in the study are listed in Table 3.

Considering all the walkers bringing any phone in any pose, among the considered ones, the classifier C_1 provides, for the smallest window size, a percentage of correctly classified instances of 93.89; precision and recall metrics are respectively 0.98 and 0.91 whereas ROC AUC is 0.91.

The considered metrics demonstrate the good performance of C_1 in recognizing the walker's identity.

The confusion matrix of C_1 as shown in Table 4 reports quite good classification results. In particular, it is interesting to observe that both male and female walkers are well identified. Only one case of identification failures is greater than 3%.

We have also investigated the accuracy of smartphone detection finding that the accuracy of Samsung S6 correctly classified instances is 99.1%. For the iPhone7plus results are similar (97.9%).

To analyze the impact of each sensor on the classifier performance, all possible combinations of sensors are considered; the comparison among the obtained results is shown in Table 5. As predictable, the best performances are obtained with the configuration including all the considered sensors; the configuration including both accelerometers and gyros provides significantly worse results with respect to orientation coupled with accelerometers or gyros. The main contribution to the walker recognition seems to come from the orientation sensor, with 91.3% of successful identifications, while only accelerometers or gyros allow respectively only 58.1% and 32.4% of correct classifications.

The classifier C_2 objective is to distinguish the pose of the phone among four possible ones: texting, pocket, phoning and swinging. The correct classified instance percentage is about 98.2% and the related metrics are shown in Table 6.

The confusion matrix, in this case, is shown in Table 7. All the poses are quite easy to detect. For the swinging and the phoning poses, we obtain 98.4% of correct classifications. On the other hand, for the texting and pocket pose, we obtain 98.5%. In the worst-case 1.3% of texting poses are confused with pocket ones and 1.3% of swinging pose are confused with phoning pose.

To further test the robustness of classifier C_1 , it has been applied on data collected from four walkers, w05m, w06m, w12f, and w18f, in a real-world context: an outdoor walking session on a pedestrian street on a typical working day. The percentage of correct identification drastically decreased to 91.2. From the confusion matrix, in Table 8, it is evident that all the walkers are still correctly identified. Respectively, we obtain 91.9%, 89.7%, 92.02 and 91.12 for w05m, w06m, w12f, and w18f.

All the results shown so far are obtained processing the samples included in a very short time window of 0.32 seconds, to allow a real-time application of the method. Increasing the time window, and consequently, the number of samples used for the recognition could improve the performance of C_1 classifier. To demonstrate that, a sliding window approach has been adopted with fixed windows of increasing sizes up to 10 seconds. The obtained results, in terms of accuracy and error, are shown in Figure 3. It is evident that the accuracy increases with the window size and consequently the error decreases with it; starting from a window size of 4.5 seconds, the accuracy becomes $\approx 100\%$. The choice of the time window size is related to

TABLE 3. Walkers profiles (9 men and 9 females)

ID	Gender	Age [years]	Height [cm]
w01m	Male	23	169
w02m	Male	23	170
w03f	Female	23	173
w04m	Male	40	192
w05m	Male	22	183
w06m	Male	42	180
w07m	Male	31	183
w08f	Female	31	152
w09f	Female	34	154
w10m	Male	27	175
w11f	Female	27	172
w12f	Female	21	172
w13m	Male	28	175
w14f	Female	26	161
w15f	Female	27	160
w16f	Female	26	159
w17m	Male	31	177
w18f	Female	18	163

TABLE 4. Confusion matrix of classifier C_1 : all walkers with any phone in any pose

Classified →	w01m	w02m	w03f	w04m	w05m	w06m	w07m	w08f	w09f	w10m	w11f	w12f	w13m	w14f	w15f	w16f	w17m	w18f
w01m	93.80	0.60	0.00	0.28	0.33	1.36	0.30	0.00	0.22	0.00	0.00	0.00	0.00	0.60	0.15	0.40	0.00	0.00
w02m	0.75	92.88	0.00	0.40	1.25	1.44	0.90	0.00	0.00	0.00	0.00	0.00	0.48	0.00	0.00	0.00	0.00	0.00
w03f	0.00	0.00	91.50	0.00	0.00	0.00	0.00	0.20	0.20	0.20	0.10	0.20	0.10	0.10	0.10	0.20	0.20	0.20
w04m	0.00	1.70	0.00	93.12	0.35	0.47	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.39	0.00
w05m	0.18	2.28	0.00	1.52	92.68	0.38	0.50	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00
w06m	1.30	0.32	0.00	1.43	1.30	91.40	1.86	0.00	0.23	0.20	0.00	0.30	0.05	0.00	0.00	0.00	0.00	0.00
w07m	0.30	0.94	0.00	1.00	0.90	0.09	91.14	0.00	0.00	0.00	0.00	0.23	0.00	0.00	0.00	0.00	0.00	0.00
w08f	0.00	0.00	1.80	0.00	0.00	0.00	0.00	91.65	0.60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
w09f	0.00	0.00	3.50	0.03	1.12	0.12	0.00	1.10	92.80	0.00	0.00	0.27	0.00	0.00	0.00	0.00	0.00	0.16
w10m	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	92.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.80
w11f	1.20	0.00	0.00	1.12	0.00	0.00	0.00	0.00	0.00	0.00	94.55	0.00	0.23	0.00	0.00	0.00	0.00	0.00
w12f	0.00	0.00	0.00	0.00	0.00	1.45	2.60	0.00	0.00	0.00	0.00	92.70	0.00	0.10	0.55	1.90	0.13	0.00
w13m	0.00	0.00	2.00	0.00	1.03	0.00	0.00	0.03	0.10	0.00	0.00	0.00	91.94	0.00	0.00	0.00	0.00	0.00
w14f	0.00	1.13	0.00	1.10	0.00	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.00	92.99	0.00	0.00	0.37	0.00
w15f	0.31	0.00	0.00	0.00	0.00	2.78	2.40	0.01	0.00	0.00	0.20	0.17	0.00	0.00	92.92	0.00	0.00	0.00
w16f	0.00	0.00	1.20	0.00	1.04	0.00	0.00	0.06	0.45	0.00	0.00	1.50	0.00	0.00	0.00	93.35	0.00	0.10
w17m	1.24	0.15	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.20	0.20	0.13	0.00	0.00	0.00	0.00	94.28	0.13
w18f	0.92	0.00	0.00	0.00	0.00	0.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.72
↑ Actual																		

TABLE 5. Impact of each sensor on the classifier performance.

Feature Sets	Sensors	Accuracy	Precision	Recall	ROC AUC
S_1	A	58.10%	0.61%	0.65%	0.64%
S_2	G	32.40%	0.34%	0.31%	0.30%
S_3	O	91.30%	0.90%	0.95%	0.95%
S_4	A and G	78.00%	0.81%	0.77%	0.84%
S_5	A and O	92.70%	0.92%	0.91%	0.89%
S_6	G and O	91.20%	0.89%	0.92%	0.87%
S_7	A, G and O	93.80%	0.92%	0.93%	0.92%

TABLE 6. Performance metrics of classifier C_2 , to identify phone pose.

Accuracy	Precision	Recall	ROC AUC
98.2%	98.2%	98.2%	99.6%

the application; if a real-time response is required, a short window is necessary (allowing more errors), otherwise, a larger one can be considered.

To increase the robustness of walker recognition, we adopted a different approach based on identifying the walker once the phone pose is known. This hierarchical classifier, name C_3 , is described in Section IV-C. The results obtained

for C_3 are resumed in Table 9 and show a significantly improvement in performances with respect to classifier C_1 . With the smartphone in the pocket pose, the accuracy of walker identification reaches its highest level (99.16%), obtaining the lowest in the swinging pose (with still a very good accuracy of $\approx 96.92\%$).

The time window size impact is analyzed also for C_1 classifiers with fixed device poses (since these are the base classifiers used in C_3) and the accuracy/error behaviors are shown in Figure 4. From those, it is evident that the accuracy increases more rapidly with window size with respect to the C_1 classifier trained on the device in all poses; a window of 2sec provides an accuracy very near to 100% for texting,

TABLE 7. Confusion matrix of classifier C_2 , for all walkers with any phone.

↓ Actual phone pose	Classified phone pose ↓			
	Texting	Pocket	Phoning	Swinging
Texting	98.50	0.10	0.30	1.10
Pocket	1.30	98.50	0.00	0.20
Phoning	0.10	1.20	98.40	0.30
Swinging	0.10	0.20	1.30	98.40

TABLE 8. Confusion matrix of classifier C_1 , applied to walkers w05m, w06m, w12f and w18f with any phone.

↓ True / Pred →	w01m	w02m	w03f	w04m	w05m	w06m	w07m	w08f	w09f	w10m	w11m	w12f	w13m	w14f	w15f	w16f	w17m	w18f
w05m	0.66	0.12	0.66	1.34	91.90	1.50	0.20	0.10	0.30	0.17	0.50	0.12	0.60	0.30	1.30	0.00	0.20	0.03
w06m	0.20	0.10	0.40	2.50	0.40	89.70	0.80	0.25	0.80	1.20	0.20	0.15	1.60	0.90	0.00	0.60	0.10	0.10
w12f	0.26	0.11	1.20	1.11	0.10	1.50	0.20	0.10	0.30	0.17	0.50	92.02	0.60	0.30	1.30	0.00	0.20	0.03
w18f	0.10	0.40	0.39	2.03	0.44	0.22	0.50	0.25	0.80	1.20	0.20	0.15	0.60	0.90	0.00	0.60	0.10	91.12

FIGURE 3. Accuracy and error of C_1 classifier, for all device poses, with respect to window size.

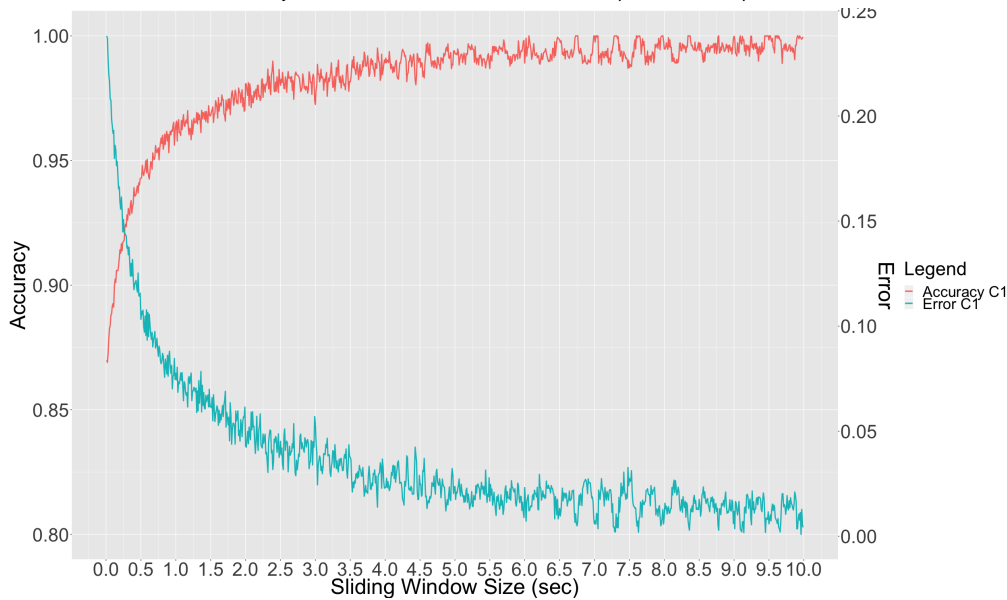


TABLE 9. Performances of base classifiers for walkers with fixed device modes and the walker classifier C_3 .

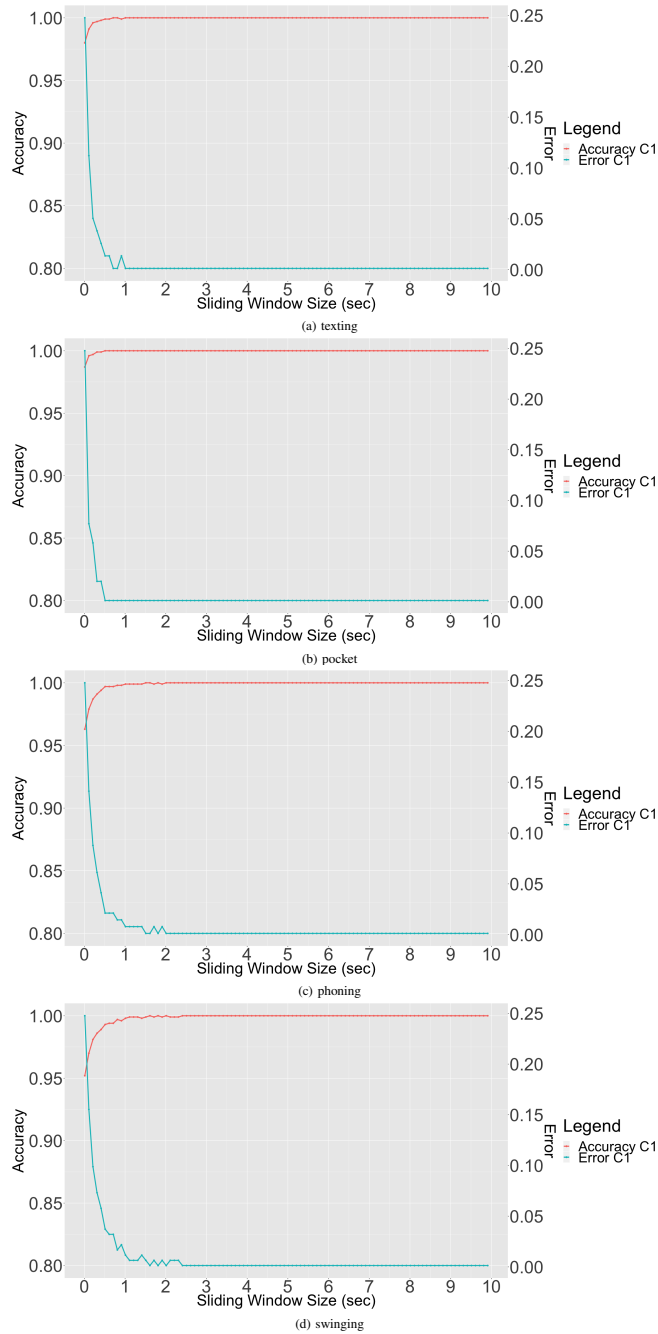
Evaluated classifiers	Accuracy	Precision	Recall	ROC AUC
Generic Walker Classifier (C_1)	93.89%	0.98	0.91	0.92
Texting-pose Classifier	98.78%	0.99	0.97	0.98
Pocket-pose Classifier	99.16%	0.99	0.98	0.97
Phoning-pose Classifier	97.61%	0.99	0.95	0.96
Swinging-pose Classifier	96.92%	0.98	0.94	0.93
Hierarchical Classifier (C_3)	98.72%	0.98	0.99	0.90

pocket and phoning posed, whereas a window of 2.5sec is necessary for swinging.

To further assess the robustness of the classifier C_3 , an additional test is carried out, classifying measurements from the walker w04m that suffered a minor accident causing a slight lameness. The goal of the test is to verify if C_3 can recognize w04m, despite his altered gait. The results, as shown in Figure 5, demonstrate that the injured walker is not anymore distinctly identifiable by C_1 classifier in these conditions: a very low accuracy of $\approx 65\%$ is obtained with the smallest window size. The study of the performances of this case using both the plain classifier (C_1) and the

hierarchical one (C_3) reveals that the window size has a bigger impact (with improvement of more than 20%) on the resulting accuracy with respect to the classifier’s architecture (with the improvement in accuracy slightly below 5% with respect to the plain one). It’s worth observing that using the best classifier (the C_3 with a window size of 8.5sec) the injured walker is identified with an accuracy of 95.89% exhibiting a very high tolerance to data alteration. Another interesting comparison among C_1 and C_3 concerns the window size needed to reach a "good" accuracy (i.e. at least 90%): in this case, the plain classifier C_1 requires a window size of ≈ 5.5 sec whereas the classifier C_3 requires

FIGURE 4. Accuracy (red line) and error (blue line) behavior of C_1 classifiers.



a window size of ≈ 3.1 sec (almost half of the size) to reach the same performances: this means that the C_3 classifier, even if more difficult to train and complex to set up, is a good fit for applications that require both robust and quick identification. Finally, we study the impact of the walker height on the accuracy of classifier C_3 . Figure 6 reports the accuracy for three groups of walkers: short (<165 cm), regular (included in the range [165 cm-176 cm]) and tall (>176cm). The figure highlights that there is no significant difference between the obtained accuracy values.

FIGURE 5. Accuracy and error behavior of plain (C_1) and hierarchical (C_3) classifiers with

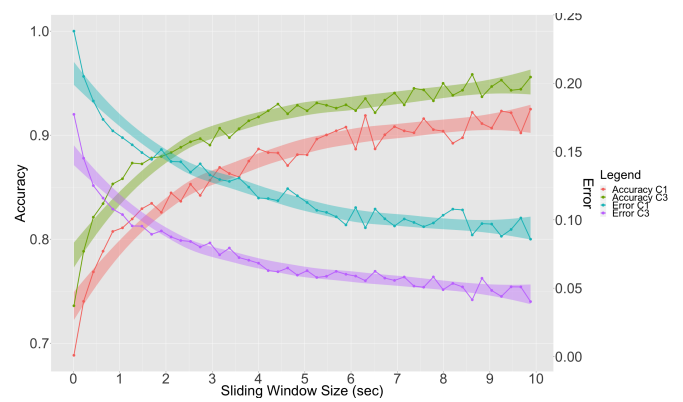
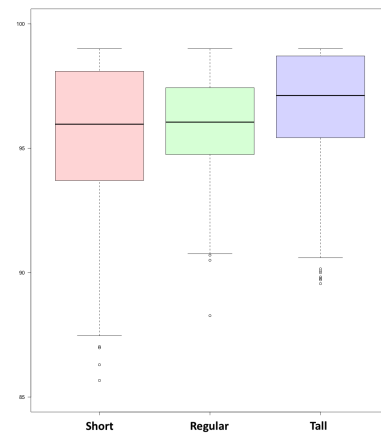


FIGURE 6. Accuracy of C_3 classifier for different walker heights.



VII. BINARY CLASSIFICATION CASE STUDY

TABLE 10. Hyper-parameters Optimization and selected ranges.

Hyperparameters	Ranges
Bootstrap	Yes , No
Max Depth	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Max Features	Auto , SQRT, Log2
Min Samples Leaf	1, 2, 4, 8, 10
Min Samples Split	2, 5, 10, 15, 20
Number of Estimators	from=500, to=2000, step=50

In this section, we discuss the results of applying the C_3 hierarchical classifier in a real binary classification scenario where people, except the device "owner", could be not known in advance. For this use case, it is only important to know if the device is carried by its legitimate owner or not. This could be very useful for security reasons: if the device detects that the user is a stranger, it could be configured to lock, to enable microphone and camera sending captured data and its GPS position to the owner account (or even to wipe its content, giving the chance to insert the pin to avoid wiping). Specifically, we performed the training for each user considering the remaining subjects as potential thieves.

FIGURE 7. Performances of C_3 binary classifier.

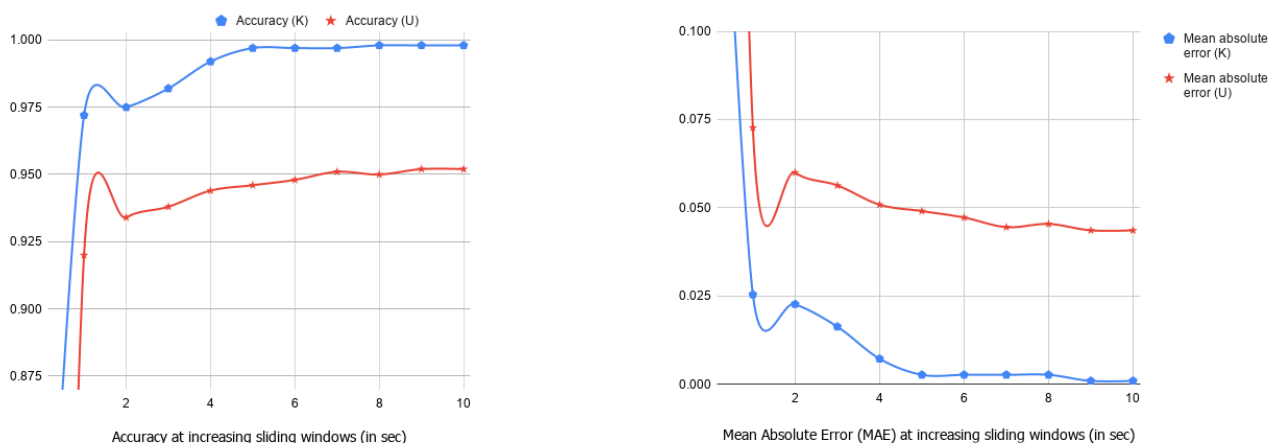


TABLE 11. Accuracy of C_3 classifier for binary "owner or thief" use case.

Validation Metrics	Best Hyperparameters							
	K	U	Use Bootstrap	Max Depth	Max Features	Min Samples Leaf	Min Samples Split	Number of Estimators
Training Time (min)	11.25		True	80	AUTO	2	5	1800
Accuracy	0.97	0.92						
F1	0.95	0.89						
ROC AUC	0.93	0.90						
Training Time (min)	8.72		True	70	SQRT	2	2	1400
Accuracy	0.96	0.91						
F1	0.93	0.89						
ROC AUC	0.92	0.90						
Training Time (min)	16.61		False	80	AUTO	4	10	1750
Accuracy	0.95	0.88						
F1	0.91	0.89						
ROC AUC	0.91	0.87						

In Table 10 are reported the optimized hyper-parameters and their ranges.

Table 11 shows the results for the three best sets of parameters. The first column of the table reports the validation metrics (i.e., training time, classifier accuracy, F1 score, and ROC AUC) evaluated at the smallest window size. The next two columns (i.e., U and K) are related to test sets adopted for validation. The column labeled with 'U' (that stands for "Unknown") provides validation metrics for a test set produced with subjects that, except for the owner, were never seen by the trained network. The column labeled with 'K' (that stands for "Known") provides validation metrics for a test set produced with subjects that also produced data to train the network. As we can see from the results, even at the smallest window, for known subjects, accuracy and F1 are never below 0.9 and making the classifier very robust. The best result of 0.99 is achieved with 1650 estimators and trees of max depth equals to 110. It is interesting to note that increasing estimators did not produce any further improvement in final validation (but with much worse training times). In the context of unknown subjects, results are, as expected, not as good as for the K test set. In this case, the best accuracy at the smallest

window size is 0.92 (obtained, for the same parameter permutations) whereas the worst one dropped to 0.88. We studied also the accuracy and mean absolute error (MAE) of the best classifiers for increasing window sizes. The results are reported in Figure 7. As we can see from the trends, for the known group after six seconds we obtain nearly perfect classification results whereas for the unknown group the best result is slightly better than 0.975 of accuracy with an MAE of 0.015 (as shown by the Accuracy and Mean Absolute Errors curves). These results show how the best classifier is effective and has good generalization capabilities also in real-world scenarios where peoples involved are not all known in advance.

VIII. CONCLUSIONS

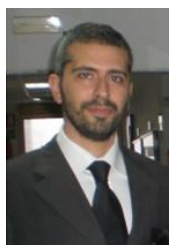
In this research, machine learning techniques are applied to measurements from smartphone sensors, specifically accelerometers, gyroscopes, and orientation, to retrieve information about the walker bringing the device. The information of interest is the identity of the walker and the pose of the device. The most relevant sensor for identity recognition is the orientation, the less one is the gyroscope; using all three sensors is anyway the best choice. The walker identification is performed with several time windows of measurements and it has been demonstrated that increasing the window size produces significant improvements of the results; specifically, with time windows over 4.5 seconds, the percentage of correct identification is ≈ 100 . The recognition of the device pose is obtained with satisfying results and the pose classifier is also considered as a part of a hierarchical one that can identify both walker identity and device pose. The performance of the hierarchical classifier in identity recognition overcome the ones of the simple classifier in terms of correct identification percentage; moreover, the accuracy increases rapidly with window size, obtaining $\approx 100\%$ with a window of 2.5 seconds for the multi-class scenario when the classifier is trained on all involved

subjects. In the binary case, when involved subjects could be not known in advance performances are still good with the best accuracy of $\approx 97.5\%$ using a window of at least six seconds.

REFERENCES

- R. Bogue, Recent developments in mems sensors: a review of applications, markets and technologies, *Sensor Review* 33 (2013) 300–304. doi:
- G. Lammel, The future of mems sensors in our connected world, in: 2015 28th IEEE International Conference on Micro Electro Mechanical Systems (MEMS), 2015, pp. 61–64. doi:
- N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, A. T. Campbell, A survey of mobile phone sensing, *IEEE Communications Magazine* 48 (2010) 140–150. doi:
- Y. Yan, S. Cosgrove, E. Blantont, S. Y. Ko, L. Ziarek, Real-time sensing on android, in: Proceedings of the 12th International Workshop on Java Technologies for Real-time and Embedded Systems, JTTRES '14, ACM, New York, NY, USA, 2014, pp. 67:67–67:75. URL: <http://doi.acm.org/10.1145/2661020.2661026>. doi:
- P. Addabbo, M. Focareta, S. Marcuccio, C. Votto, S. L. Ullo, Land cover classification and monitoring through multisensor image and data combination, in: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2016, pp. 902–905. doi:
- U. Fareed, Smartphone sensor fusion based activity recognition system for elderly healthcare, in: Proceedings of the 2015 Workshop on Pervasive Wireless Healthcare, MobileHealth '15, ACM, New York, NY, USA, 2015, pp. 29–34. URL: <http://doi.acm.org/10.1145/2757290.2757297>. doi:
- M. Susi, V. Renaudin, G. Lachapelle, Motion mode recognition and step detection algorithms for mobile phone user, *J. Locat. Based Serv* (2012).
- L. Pei, J. Liu, R. Guinness, Y. Chen, H. Kuusniemi, R. Chen, Using ls-svm based motion recognition for smartphone indoor wireless positioning, *Sensors* (Basel) (2012).
- D. Kamisaka, S. Muramatsu, T. Iwamoto, H. Yokoyama, Design and implementation of pedestrian dead reckoning system on a mobile phone, *IEICE Transactions on Information and Systems* E94.D (2011) 1137–1146. doi:
- H. Zhang, W. Yuan, Q. Shen, T. Li, H. Chang, A handheld inertial pedestrian navigation system with accurate step modes and device poses recognition, *IEEE Sensors Journal* 15 (2015) 1421–1429. doi:
- T. M. Mitchell, *Machine Learning*, 1 ed., McGraw-Hill, Inc., New York, NY, USA, 1997.
- M. C. Mihăescu, P. c. Popescu, D. D. Burdescu, J48 list ranker based on advanced classifier decision tree induction, *Int. J. Comput. Intell. Stud.* 4 (2015) 313–324.
- W. Iba, P. Langley, Induction of one-level decision trees, in: Proceedings of the Ninth International Workshop on Machine Learning, ML '92, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992, pp. 233–240.
- S. Hoeglinger, R. Pears, Use of hoefding trees in concept based data stream mining, in: 2007 Third International Conference on Information and Automation for Sustainability, 2007, pp. 57–62.
- L. Breiman, Random forests, *Machine Learning* 45 (2001) 5–32. URL: <https://doi.org/10.1023/A:1010933404324>. doi:
- Y. Zhao, Y. Zhang, Comparison of decision tree methods for finding active objects, *Advances in Space Research* 41 (2008) 1955–1959.
- G. V. Nadiammai, M. Hemalatha, Performance Analysis of Tree Based Classification Algorithms for Intrusion Detection System, Springer International Publishing, Cham, 2013, pp. 82–89.
- N. Landwehr, M. Hall, E. Frank, Logistic model trees, *Machine Learning* 59 (2005) 161–205.
- N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, *Machine Learning* 29 (1997) 131–163.
- S. Lee, S. Shimoji, Bayesnet: Bayesian classification network based on biased random competition using gaussian kernels, in: IEEE International Conference on Neural Networks, 1993, pp. 1354–1359 vol.3.
- G. H. John, P. Langley, Estimating continuous distributions in bayesian classifiers, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, UAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 338–345.
- A. McCallum, K. Nigam, A comparison of event models for naive bayes text classification, 1998.
- A. M. Kibriya, E. Frank, B. Pfahringer, G. Holmes, Multinomial naive bayes for text categorization revisited, in: Proceedings of the 17th Australian Joint Conference on Advances in Artificial Intelligence, AI'04, Springer-Verlag, Berlin, Heidelberg, 2004, pp. 488–499.
- D. A. Freedman, *Statistical models: theory and practice*, Cambridge University Press, 2009.
- B. Kégl, The return of adaboost.mh: multi-class hamming trees, *CoRR abs/1312.6086* (2013).
- D. Aha, D. Kibler, Instance-based learning algorithms, *Machine Learning* 6 (1991) 37–66.
- L. Breiman, Bagging predictors, *Machine Learning* 24 (1996) 123–140. URL: <https://doi.org/10.1007/BF00058655>. doi:
- S. J. Russell, P. Norvig, *Artificial intelligence: a modern approach*, Malaysia; Pearson Education Limited., 2016.
- L. Liu, S. Wang, B. Hu, Q. Qiong, J. Wen, D. Rosenblum, Learning structures of interval-based bayesian networks in probabilistic generative model for human complex activity recognition, *Pattern Recognition* 81 (2018). doi:
- R. San-Segundo, H. Blunck, J. Moreno-Pimentel, A. Stisen, M. Gil-Martín, Robust human activity recognition using smartwatches and smartphones, *Engineering Applications of Artificial Intelligence* 72 (2018) 190–202. URL: <http://www.sciencedirect.com/science/article/pii/S0952197618300812>. doi:
- D. Riboni, T. Szttyler, G. Civitarese, H. Stuckenschmidt, Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '16, ACM, New York, NY, USA, 2016, pp. 1–12. URL: <http://doi.acm.org/10.1145/2971648.2971691>. doi:
- C. Fiandrino, A. Capponi, G. Cacciatore, D. Kliazovich, U. Sorger, P. Bouvry, B. Kantarci, F. Granelli, S. Giordano, Crowdsensim: a simulation platform for mobile crowdsensing in realistic urban environments, *IEEE Access* 5 (2017) 3490–3503. doi:
- G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, R. Curtmola, Fostering participation in smart cities: a geo-social crowdsensing platform, *IEEE Communications Magazine* 51 (2013) 112–119. doi:
- J. Hong, J. Ramos, A. K. Dey, Toward personalized activity recognition systems with a semipopulation approach, *IEEE Transactions on Human-Machine Systems* 46 (2016) 101–112. doi:
- A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, *ACM Comput. Surv.* 46 (2014) 33:1–33:33. URL: <http://doi.acm.org/10.1145/2499621>. doi:
- S. Liu, R. X. Gao, D. John, J. W. Staudenmayer, P. S. Freedson, Multisensor data fusion for physical activity assessment, *IEEE Transactions on Biomedical Engineering* 59 (2012) 687–696. doi:
- V. Radu, N. D. Lane, S. Bhattacharya, C. Mascolo, M. K. Marina, F. Kawsar, Towards multimodal deep learning for activity recognition on mobile devices, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, UbiComp '16, ACM, New York, NY, USA, 2016, pp. 185–188. URL: <http://doi.acm.org/10.1145/2968219.2971461>. doi:
- N. Y. Hammerla, S. Halloran, T. Ploetz, Deep, convolutional, and recurrent models for human activity recognition using wearables, *CoRR abs/1604.08880* (2016). URL: <http://arxiv.org/abs/1604.08880>. arXiv:1604.08880.
- C. A. Ronao, S.-B. Cho, Human activity recognition with smartphone sensors using deep learning neural networks, *Expert Syst. Appl.* 59 (2016) 235–244. URL: <https://doi.org/10.1016/j.eswa.2016.04.032>. doi:
- F. J. Ordonez, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, *Sensors* 16 (2016). URL: <http://www.mdpi.com/1424-8220/16/11/115>. doi:
- Incel, Analysis of movement, orientation and rotation-based sensing for phone placement recognition, *Sensors* 15 (2015).
- T. Hoang, D. Choi, T. Nguyen, On the instability of sensor orientation in gait verification on mobile phone, in: 2015 12th International Joint Conference on e-Business and Telecommunications (ICETE), volume 04, 2015, pp. 148–159.
- J. R. Kwapisz, G. M. Weiss, S. A. Moore, Cell phone-based biometric identification, in: 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2010, pp. 1–7. doi:
- W.-H. Lee, R. Lee, Implicit sensor-based authentication of smartphone users with smartwatch, in: Proceedings of the Hardware and Architectural Support for Security and Privacy 2016, HASP 2016, ACM, New York, NY, USA, 2016, pp. 9:1–9:8. URL: <http://doi.acm.org/10.1145/2948618.2948627>. doi:
- T. B. Singha, R. K. Nath, A. V. Narsimhadhan, Person recognition using smartphones' accelerometer data, *CoRR abs/1711.04689* (2017).
- B. Esmael, A. Arnaut, R. K. Fruhwirth, G. Thonhauser, Multivariate Time Series Classification by Combining Trend-Based and Value-Based Approximations, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 392–403. URL: http://dx.doi.org/10.1007/978-3-642-31128-4_29. doi:
- M. Stone, Cross-validators choice and assessment of statistical predictions, *Journal of the Royal Statistical Society, Ser. B* 36 (1974) 111–147. Cited By 11.
- R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 1137–1143. URL: <http://dl.acm.org/citation.cfm?id=1643031.1643047>.
- J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, in: Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS'11, Curran Associates Inc., USA, 2011, pp. 2546–2554. URL: <http://dl.acm.org/citation.cfm?id=2986459.2986743>.

•••



ANTONIO ANGRISANO has obtained his M.Sc degree (Cum Laude) in Science of Navigation and his PhD in Geodetic and Topographic Sciences at University of Naples Parthenope. In 2010 he is visiting researcher at the University of Calgary PLAN (Position Location and Navigation) Group, researching algorithms for filtering and data fusion. In 2014-15 he works in Magneti Marelli, FCA Group, as a navigation system engineer.

Since 2015 he is an assistant professor at G. Fortunato University. His research interests include GNSS and Augmentation Systems, Inertial and Integrated Navigation, RAIM and Integrity.



SALVATORE GAGLIONE is Associate Professor at the Science and Technology Department of the University of Naples "Parthenope" and scientific director of the Parthenope Navigation Group (PANG) Laboratory. In 2010 he was Visiting Academic at the Department of Geomatics Engineering at the University of Calgary. In the last Academic year, He gives lectures in the field of Inertial and Integrated Navigation, Radio Navigation and Air Navigation. His research activities

are focused on GNSS positioning and other sensors (INS, Camera, etc.) integration algorithms for several applications. Professor Gaglione is member and delegate for Europe of the "Istituto Italiano di Navigazione"(AIVELA), a scientific non-profit organization that promotes navigation culture. He is the author of 60 publications on journals and international conferences.



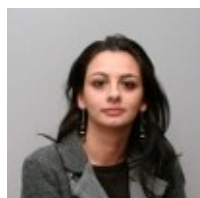
MARIO LUCA BERNARDI received the Laurea degree in Computer Science Engineering from the University of Naples "Federico II", Italy, in 2003 and the Ph.D. degree in Information Engineering from the University of Sannio in 2007. He is currently an assistant professor of Computer Science at the Giustino Fortunato University. Since 2003 he has been a researcher in the field of software engineering and his list of publications contains more than 60 papers published in journals and conference proceedings.

His main research interests include software engineering (maintenance, testing, business process management, reverse engineering and data mining on software systems, software quality assurance with particular interest on internal quality metrics and new paradigms for software modularity, including aspect-oriented software, component-based software, and model-driven development). He serves both as a member of the program and organizing committees of conferences, and as associate editor and reviewer of papers submitted to some of the main journals and magazines in the field of software engineering, software maintenance, and program comprehension.



MARIO VULTAGGIO is Professor of Navigation at G. Fortunato University; up to 2014 he was full Professor at Parthenope University of Naples. His field of research was in oceanography surveying between 1972 and 1979. Since 1979 his research interests were Navigation, Cartography, VTS (Vessel Traffic Services), Space and Astronomical Navigation. He was director of the Institute of Navigation "G.Simeon" of Faculty of Science and Technology and President of Degree of Nautical Science and Science and Technology of Navigation at Parthenope University.

His field of research was in oceanography surveying between 1972 and 1979. Since 1979 his research interests were Navigation, Cartography, VTS (Vessel Traffic Services), Space and Astronomical Navigation. He was director of the Institute of Navigation "G.Simeon" of Faculty of Science and Technology and President of Degree of Nautical Science and Science and Technology of Navigation at Parthenope University.



MARTA CIMITILE is Assistant Professor and Aggregated Professor at Unitelma Sapienza University of Rome (Italy). She received a Ph.D. in Computer Science on 06/05/2008 at the Department of Computer Science at the University of Bari and she received the Italian Scientific Qualification for the Associate Professor position in Computer Science Engineering in April 2017. She published more than fifty papers at international conferences and journals. Her main

research topics are Business Process Management and modeling, Knowledge modeling and Discovering, Process and Data Mining in Software Engineering Environment. In the last year, she was involved in several industrial and research projects and she is a founding member of the SpinOff of University of Bari named Software Engineering Research and Practices s.r.l. (www.serandp.com). She was in the program and organizing committees of several international conferences, she is a reviewer to some of the main journals and magazines in the field of Knowledge Management and Software Engineering, knowledge representation and transfer and data mining and she is in the Editorial Board of the Journal of Information and Knowledge Management, PeerJ Computer Science. She is an IEEE Member and IEEE Access Associate Editor.