Real-time identification using gait pattern analysis on a standalone wearable accelerometer

GUGLIELMO COLA, MARCO AVVENUTI AND ALESSIO VECCHIO

Dipartimento di Ingegneria dell'Informazione, University of Pisa,
Largo Lazzarino 1, 56122 Pisa, Italy
Email: guglielmo.cola@iet.unipi.it, marco.avvenuti@unipi.it, alessio.vecchio@unipi.it

Wearable devices can gather sensitive information about their users. For this reason, automated authentication and identification techniques are increasingly adopted to ensure security and privacy. Furthermore, identification can be used to automatically customize operations according to the needs of the current user. A gait-based identification method that can be executed in real time on devices with limited resources is here presented. The method exploits a wearable accelerometer to continuously analyze the user's gait pattern and perform identification. Experiments were conducted with ten volunteers, who carried the device in a trouser pocket and followed their daily routine without predefined constraints. In total, ∼ 98 hours of acceleration traces were collected in uncontrolled environment, including 3073 gait segments. User identification results show a recognition rate ranging from 95% to 100%, depending on the mode of operation. It is demonstrated that the method can be executed on a standalone device with less than 8 KB of RAM. In addition, the energy consumption is evaluated and compared with an architecture that requires the presence of an external computing unit. Results show that the proposed solution significantly improves the lifetime of the device (approximately +70% for the considered platform), hence fostering user acceptance.

1. INTRODUCTION

The rise of wearable devices provides the opportunity to gather, both continuously and unobtrusively, large amounts of information about their users. This, in turn, fosters the development of novel applications in many industrial areas, from wellbeing to telemedicine [1–5].

Information produced by wearable devices is strictly personal and generally sensitive. Thus, it is essential to provide adequate authentication and identification methods. Moreover, some applications may benefit from knowing the identity of the user, as the parameters of operation can be customized according to the current user's preferences and needs. To this purpose, a number of biometric techniques based on users' movements have been developed during the last years. These techniques rely on the fact that every individual performs the activities of daily living with a unique pattern. As a consequence, movement information collected when the device is worn can be compared to pre-acquired "signatures" in order to identify and/or authenticate

users with a reasonable degree of accuracy. The vast majority of proposed methods are based on the analysis of gait, since it is highly specific [6]. In addition, walking is a frequent activity and thus provides numerous "hints" about the identity of the user. Biometric features, such as gait or heart rate variability, have also been proposed as a novel means to provide secure communication in body area networks [7].

This paper focuses on *gait-based identification*. We agree with the distinction between identification and authentication as introduced in [8]. In *identification* the purpose is to recognize the current user among the set of possible users of the device. Consequently, the model used for gait-based identification is built using labeled instances belonging to each of these users. Differently, *authentication* aims to recognize whether the current user is authorized, and the gait-model is typically built relying only on the authorized user's data (unauthorized users are not known in advance by the system).

Identification methods are used when a group of individuals (e.g., athletes, patients, members of a

family, etc.) share one or more wearable devices. Thanks to gait-based identification, the shared device can automatically customize its way of operation without manual intervention. For example, an activity tracker could automatically recognize the current user, and then update his/her statistics at the end of the session. It is worth to note that the user interface of wearable devices is in many cases rather rudimentary, because of their small form factor. Thus, the presence of automatic identification mechanisms can be useful to make the interaction between humans and devices less cumbersome.

Wearable devices are characterized by batteries with limited capacity. Moreover, the processing unit is in general not particularly powerful, as a consequence of the scarce energy budget. These constraints in terms of energy and computing power should, in our opinion, be considered during the design and implementation of gait-based identification methods. Even when gait-based identification is executed on a more powerful device, such as a smartphone, the adoption of lightweight mechanisms is preferable, as the impact on the battery can be reduced.

Unfortunately, existing literature about gait-based identification does not deal with these issues. A large fraction of recent gait identification systems is smartphone-based. This is due to the widespread availability and ease of programming of these devices. Nevertheless, despite their increasing computing power, very few studies include on-device identification of users. Most of the times, the smartphone is used to collect acceleration traces, whereas classification is performed off-line on a standard PC. Some studies have been carried out using small dedicated devices, but also in this case they have been used only for data collection (classification is again executed off-line on more powerful machines). Thus, real-time identification on a standalone wearable device, possibly characterized by limited hardware resources, is substantially unexplored. In addition, since the analysis of collected data is generally executed off-line, the problems related to the energy efficiency of currently available identification techniques have been scarcely investigated. Only in a few works (such as [9]) the computational requirements were estimated.

Another aspect not thoroughly investigated is the behavior of identification methods when executed in the real world. Almost all existing methods have been evaluated using datasets made of short traces, collected in laboratory sessions. We believe that a better understanding of gait identification methods can be achieved by using longer traces collected during everyday life: this makes possible to study the effects caused by unexpected situations and behaviors on the recognition process.

The main contribution of this article with respect to previous work in the field of gait-based identification can be summarized as follows:

- A gait-based identification method specifically designed for being executed in *real time* on devices with limited resources is presented. All the steps needed to identify the user (gait detection, feature extraction, classification, and multi-instance fusion) have been carefully devised having in mind *miniaturized* devices as targets. As a result, the proposed method is able to perform real-time identification on a standalone device embedding a microcontroller unit (MCU) with 8 MHz clock and less than 8 KB of RAM. Real-time identification enables the device to promptly adapt its operation mode to the current user.

- For the first time, gait-based identification is evaluated in *uncontrolled environment*. The dataset used to evaluate the method consists of acceleration traces collected by volunteers during their normal routine and without any supervision or predefined constraint. In total, 3073 gait segments were collected in 98 hours of experiments. This is a step forward for a deeper understanding of gait-based identification in realistic settings, as previous works typically relied on short, supervised experiments performed in controlled environment.

- Notwithstanding the real-time execution requirement and the challenging experimental setting, the proposed method delivers a *correct identification rate* that is in line with the best-performing techniques currently available. As mentioned above, state-of-the-art techniques typically require off-line processing on an external unit, and were evaluated in controlled environment (e.g., volunteers were asked to walk in a straight corridor).

- The *energy consumption* of the proposed method, which can be entirely executed on the wearable device, has been evaluated and compared with an architecture that requires the presence of a base station. Our solution, besides being preferable as it is not limited to the communication range imposed by the base station, improves significantly the lifetime of the device (approximately +70% for the considered platform).

2. METHOD FOR GAIT-BASED IDENTIFICATION

A flowchart representation of the gait-based user identification method is shown in Figure 1. Acceleration is continuously sampled by a wearable device embedding a tri-axial accelerometer. Acceleration samples are processed by a *Walking Detection* algorithm, which is designed to detect *gait segments*. A gait segment is a vector of tri-axial acceleration samples collected while the user is walking. Each time walking activity is detected by the Walking Detection algorithm, the gait segment is used as input to *Gait-based User Identification*. The first step of gait-based identification consists in extracting a set of features from the raw acceleration samples
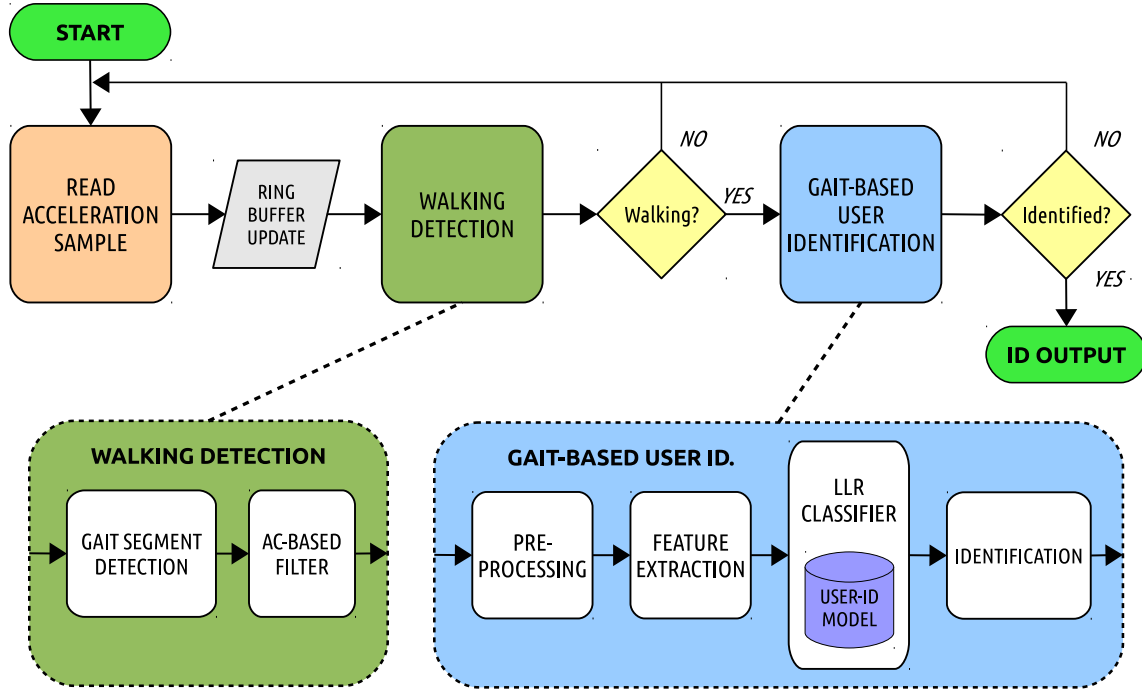
FIGURE 1. Flowchart representation of the proposed method.

of the gait segment (*preprocessing* plus *feature extraction*), so as to reduce data complexity and prepare the inputs for the classifier. The vector of features forms a *gait instance*. Gait instances are then used to feed the *classifier*, which is based on a previously generated model. The result of classification is a set of probability values, one for each user to be identified (*class probabilities*). Finally, according to an *Identification* technique, the class probabilities are analyzed to produce, if possible, an *identification output*.

In the following, all the components of the proposed method are presented in detail.

2.1.   Walking detection

Walking detection includes two phases: Gait Segment Detection and Autocorrelation-based filtering.

Gait Segment Detection is based on the algorithm that we proposed in [10] (the reader is forwarded to the referred article for further details, not included here for the sake of clarity, as the focus is here on identification). A gait segment is found when eight consecutive steps are detected, and a regularity test based on standard deviation is passed. Eight was selected as a trade-off between fast identification and reliable detection of gait patterns.

The second phase – *Autocorrelation-based Filter* – is aimed at discarding gait segments characterized by reduced regularity, which are less suitable to properly represent the user's typical gait patterns. For example, irregular segments are produced while climbing stairs, or when a sudden change in pace occurs. The test is based on the autocorrelation coefficient at the second

TABLE 1. Selected features.

AAV$_H$	AAV$_V$	AC-C1$_M$	duration
IQR$_V$	kurtosis$_M$	kurtosis$_V$	MAD$_H$
max$_H$	mean$_H$	RMS$_M$	RMS$_V$
skewness$_M$	st. dev.$_H$		

dominant period. The autocorrelation coefficients are found as described in Section 2.2.

The algorithm has low computational and memory requirements, as it is shown in Section 4.

2.2.   Preprocessing and feature extraction

In terms of data, a gait segment is a vector of tri-axial acceleration samples (*acceleration vector*). These raw samples are first low-pass filtered at 20 Hz by using a 2-nd order Butterworth filter. Then, three new vectors are calculated: (i) *acceleration magnitude*, (ii) *vertical* acceleration, and (iii) *horizontal* acceleration. Each element of the *acceleration magnitude* vector is found as the Euclidean norm of the corresponding sample in the acceleration vector. The *vertical* and *horizontal* acceleration vectors represent the vertical and horizontal dynamic components of acceleration, respectively. These two vectors are found according to the technique described in [11], which is independent from the orientation of the device with respect to the user's body. Hence, users are allowed to carry the device without caring about its orientation.

The three acceleration vectors are used as inputs to *feature extraction*. Hereafter, we use the suffixes M, V, and H to refer to a feature calculated

on acceleration magnitude, vertical acceleration, and horizontal acceleration, respectively (for example, the feature $mean_H$ is obtained by computing the mean value of horizontal acceleration). Also, we use the term *gait instance* to refer to the feature vector obtained through feature extraction.

The feature set was selected from an initial set with 41 features, which included well-known statistical features as well as features derived from autocorrelation coefficients. The selection procedure was based on the wrapper method [12]. Wrapper methods exploit the target classifier to assign a score to a feature set – the score corresponds to the estimated accuracy achieved by the classifier using that specific feature set. Wrappers generally give better results than methods relying only on statistical characteristics of features (filter methods), as the selection procedure is tailored to the classifier. A forward selection hill climbing technique was used to search through the feature space and select the feature set. The search started with an empty set, and the feature providing the best accuracy was selected. Then, each of the remaining features was tried in conjunction with the best performing feature, in order to select the best pair. The procedure was repeated by adding to the set the feature leading to the highest increase in estimated accuracy. The search was terminated when two consecutive iterations did not lead to an improvement in terms of estimated accuracy.

In our approach, the feature set was selected for the LLR classifier (described in Section 2.3) by using the above described procedure. The selected features are listed in Table 1. *InterQuartile Range (IQR)*, *kurtosis*, *max*, *mean*, *Root Mean Square (RMS)*, *skewness*, and *standard deviation* are statistical measures that have been widely used in activity recognition systems. *Duration* is the duration of the gait segment in seconds. *Median Absolute Deviation (MAD)* is a robust measure of statistical dispersion [13].

Average Absolute Variation (AAV) is found as:

$$AAV = \sum_{i=1}^{N-1} \frac{|s_{i+1} - s_i|}{N},$$

where N is the number of samples in the gait segment, and s_i is the i-th sample in the segment. AAV has been previously used for human fall detection and gait analysis [10, 14, 15].

AC-C1 is the autocorrelation coefficient at the first dominant period, and it is used to evaluate the regularity among consecutive steps [16]. Unbiased autocorrelation coefficients are calculated as follows:

$$AC_k = \frac{1}{N-k} \sum_{i=1}^{N-k} s_i * s_{i+k},$$

where AC_k is the k-th unbiased autocorrelation coefficient; N is the number of acceleration samples in the gait segment; s_i is the i-th sample minus the mean

of the samples in the gait segment. The coefficients are then normalized to one at zero lag ($AC_0 = 1$). Finally, the dominant periods in the autocorrelation signal are found by means of a peak detection algorithm.

2.3. Linear LogitBoost Regression (LLR) classifier – gait instance classification

The classifier used for the identification of gait instances is based on *multinomial logistic regression*. In particular, the technique used in this paper relies on the *LogitBoost* algorithm to fit linear regression functions to a logistic transformation of posterior class probability [17, 18]. Hereafter, we refer to this classification technique as *Linear LogitBoost Regression* (LLR). The idea behind LLR is briefly described in the next paragraph.

In a problem with J classes, let us define x as the feature vector extracted from an observed event to be classified (input variable). LLR models the posterior class probability of class j given x as

$$p_j(x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}, \quad j = 1, \ldots, J \quad (1)$$

where $F_j(x)$ is a linear regression function. $F_j(x)$ is thus described by a vector of C regression coefficients β_j, where C is the number of features, and by an additive value α_j :

$$F_j(x) = \alpha_j + \beta_j \cdot x.$$

Building the classification model consists in fitting the J regression functions, one per each class in the problem. In LLR, fitting is performed by means of the LogitBoost algorithm as described in [17]. After, the class probabilities of an unseen instance can be estimated using Equation (1). In the specific context of gait-based identification, the input variable x is a gait instance, and a linear regression function $F_j(x)$ is fitted for each user j in the dataset (user identity is the class to be recognized).

In Section 3.2 we demonstrate that LLR achieves high classification accuracy in the context of gait-based identification. In addition, in Section 4 we show that LLR enables real-time classification of gait instances on miniaturized devices with limited RAM.

2.4. Identification

The final step in gait-based identification consists in the technique used to interpret the class probabilities provided by the LLR classifier and actually produce an *identification output*. In this study, two different approaches are defined and evaluated: *single-instance* and *multi-instance* identification. In single-instance identification the output is based on a single gait instance and its respective class probabilities. Instead, multi-instance identification aggregates the

class probabilities of multiple instances belonging to the same user to reduce identification errors.

More specifically, three identification techniques are evaluated: one based on the single-instance approach, and two slightly different techniques based on the multi-instance approach.

2.4.1. Single-instance identification technique

The single-instance technique for identification is very simple. The class probabilities produced by the LLR classifier are directly used to produce an identification output. This is achieved by selecting the user j^* with the highest class probability:

$$j^* = \arg \max_j p_j(x).$$

In the single-instance technique, an identification output is produced as soon as a single gait instance is found. This ensures fast identification, but may lead to sporadic errors if the user is not walking according to his/her usual gait pattern.

2.4.2. Multi-instance identification techniques (basic and adaptive)

In *multi-instance* identification, the system delays identification until a predefined number of gait instances have been gathered. By reducing the frequency of identification outputs, the system is able to reduce errors and achieve higher identification accuracy. The instances must belong to the same user, and thus the device must be capable of inferring that the user has not changed during the data collection process. For example, this assumption can be made in case of bursts of gait activity with back-to-back gait instances. Alternatively, a proximity sensor or acceleration-based techniques may be exploited to automatically detect when the user might have changed.

A multi-instance identification technique is described by: (i) the size of the window of instances used for identification (*window size* parameter); (ii) a function to *aggregate* the class probabilities produced by LLR for each instance in the set; (iii) a *rule* used to perform identification based on the aggregate class probabilities.

Two slightly different multi-instance techniques are proposed and evaluated: *basic* and *adaptive* multi-instance identification. Both techniques collect a window of consecutive instances with predefined size, and single-instance class probabilities are averaged in order to aggregate LLR results. The two techniques actually differ in the rule used to produce an identification output based on the aggregate result.

The *basic* approach performs identification by choosing the user with the highest probability, according to the average class probabilities found over the window of gait instances. Let W be the window size, and $X = \{x_1, x_2, ..., x_W\}$ the set of gait instances in the considered window. A set of class probabilities for user j is then calculated as $P_j = \{p_j(x_1), p_j(x_2), ..., p_j(x_W)\}$.

The average value $\bar{p}_j = avg(P_j)$ is computed for all the users. Finally, the user j^* with the highest average class probability is found as

$$j^* = \arg \max_j \bar{p}_j.$$

Figure 2(a) illustrates how the technique works with an example. For the sake of simplicity, in the presented example there are four users in the system, and the window is composed of three instances. It is also supposed that the instances actually belong to user 1 (ground truth). The first window – comprising instances from number 1 to number 3 – leads to an identification error, because user 2 has the highest average probability over the window. Instead, in the second window – instances from number 4 to number 6 – the user is correctly identified. The rate of correct identification outputs is thus 50% (one correct identification out of two identification outputs).

The *adaptive* multi-instance technique starts by finding the average probabilities over a fixed-size window, like in the basic approach. However, this time there is an additional test to decide whether the current window is suitable for reliable identification. More precisely, the difference (*gap*) between the highest probability and the second-highest probability in the window is found. Only if the gap is above a predefined threshold, identification is actually performed by choosing the user with the highest probability. Conversely, when the gap is below the threshold, identification is delayed until a new gait instance is found. When a new instance is available, the window is moved forward by removing the oldest gait instance and the same procedure is repeated. More formally, the average probability $\bar{p}_j$ for user j is calculated in the same way as in the basic multi-instance method. Let us define $\bar{P} = \{\bar{p}_1, \bar{p}_2, ..., \bar{p}_J\}$, where J is the number of users. Then, $m' = max(\bar{P})$ and $m'' = max(\bar{P} \setminus m')$ are computed. Only if the $gap = m' - m''$ is higher than a predefined threshold, the procedure returns the user j^* with the highest average class probability as the identification output.

Figure 2(b) illustrates how the adaptive technique works, using the same example shown for the basic technique. Again, it is supposed that the window size is three instances and that all the instances belong to user 1. The gap threshold is 60%. Differently from the basic technique, this time the first window does not lead to an error, because it is not used by the system to produce an identification output. Indeed, the gap between the highest probability (user 2, 43%) and the second-highest probability (user 1, 41%) is far below the required threshold. The second window includes instances from number 2 to number 4 (with respect to the first window used, instance number 1 is discarded and the system adds a new instance to the window). The second window produces a correct output (user 1 is leading by a 62% gap). The system is thus capable
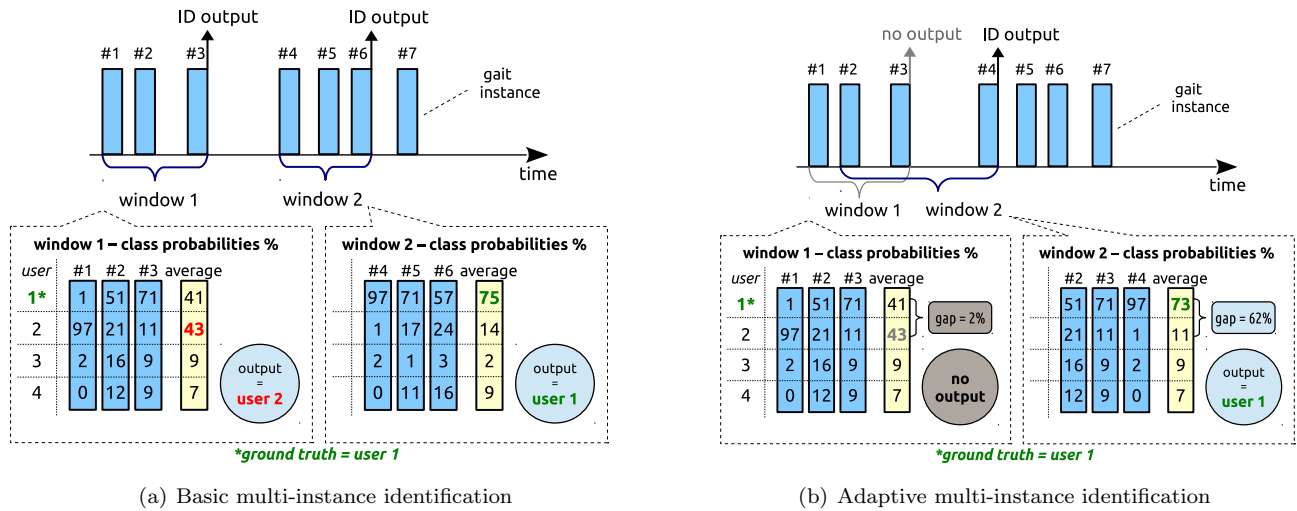
(a) Basic multi-instance identification   (b) Adaptive multi-instance identification

FIGURE 2. Multi-instance identification example with the *basic* (a), and *adaptive* (b) techniques.

of producing a correct identification output after four instances, without identification errors.

The gap between the two leading probabilities indicates whether there is a clear winner in terms of classification. When the user is walking according to his/her usual pattern, consistent identification results over consecutive instances are expected, marked by a relatively high gap. Hence, by analyzing the gap the system is able to understand when the currently available window of samples does not represent a reliable input for identification, and the identification output should be delayed to reduce the occurrence of identification errors.

3. EXPERIMENTAL SETUP, RESULTS AND DISCUSSION

In this section we investigate how the gait-based identification method performs with real-world acceleration data. More precisely, the following subsections present: (i) the wearable device used in the experiments; (ii) the data collection campaign and the results of walking detection; (iii) the results of LLR classification and a comparison with some other classification schemes; (iv) the results obtained by the three identification techniques – single-instance, basic multi-instance, and adaptive multi-instance. Finally, the results are discussed.

3.1. Wearable device

The device used in the experiments is a Shimmer2r [19], embedding a TI MSP430 microcontroller (up to 8 MHz clock, 10 KB RAM) and a Freescale MMA7361 tri-axial accelerometer with ±6 g range per axis. A microSD card is also available for internal data logging up to 2 GB.

TABLE 2. Volunteers' characteristics.

User	Gender	Age	Height [cm]	Weight [kg]
1	F	26	160	55
2	F	26	166	50
3	F	34	170	60
4	F	57	166	66
5	F	61	166	77
6	M	22	168	58
7	M	27	180	75
8	M	28	175	62
9	M	40	177	81
10	M	68	175	95

3.2. Data collection and walking detection results

Ten volunteers (5 females, 5 males , age 38.9 ± 16.0, height 170.3 ± 5.9 cm, weight 67.9 ± 13.1 kg) were involved in a data collection experiment. Gender, age and physical characteristics of each volunteer are shown in Table 2. The volunteers carried a Shimmer 2r in a front trouser pocket for about ten hours on average. Acceleration was sampled with 51.2 Hz frequency, and samples were saved on the SD memory of the Shimmer device, thus enabling repeatable off-line evaluation of the collected traces.

During data collection, the volunteers performed their habitual activities without any kind of supervision or constraint (hereafter, we also refer to the volunteers as *users*). Change of footwear was allowed. To minimize the burden, the volunteers were asked to roughly annotate performed activities and locations, using basic labels like *Home*, *Office*, *Transport*, and *Outdoors*. Home and Office mainly include short walks and long intervals of physical inactivity. Transport is used to indicate the use of personal or public transportation. Outdoors indicates time spent walking outdoors or briefly visiting bars/shops.

TABLE 3. Experiment details and walking detection results.

User	Duration [h]	Home %	Office %	Transport %	Outdoors %	Segments
1	8.2	0	76.6	4.1	19.3	489
2	11.9	0	82.9	0	17.1	694
3	9.4	7.1	68.7	11.4	12.8	135
4	9.2	94.7	0	2.2	3.2	174
5	8.8	100	0	0	0	115
6	10.6	82.4	0	0	17.6	454
7	10.0	0	83.8	7.3	8.8	410
8	12.3	54.3	36.7	9.0	0	329
9	9.2	5.9	82.5	8.8	2.8	134
10	8.6	18.9	0	43.6	37.6	139
Average	9.8	36.3	43.1	8.6	11.9	307.3

Table 3 shows some details about the experiments as well as the results of walking detection. The second column, *Duration*, shows the duration in hours of each experiment – in total ~ 98 hours of acceleration samples were collected. Columns from 3 to 6 show the percentage of time spent in the different activities/locations. The last column, *Segments*, shows the number of gait segments detected by the walking detection algorithm. About 307 gait segments were detected on average, 3073 in total. Each of these gait segments consists of eight steps and passed the regularity tests based on standard deviation and autocorrelation. A gait segment lasted 4.2 s on average, while the shortest and longest gait segments lasted 3.3 s and 5.9 s, respectively. It was verified that the walking detection algorithm did not detect gait segments when the volunteers were using transportation, which is a common issue with simple step detection techniques.

3.3. LLR classification results

The gait segments were processed with the feature extraction algorithms described in Section 2.2. As a result, 3073 gait instances were obtained, consisting in vectors with fourteen features. Each gait instance was labeled with the ID of the user who produced it, and the ID itself represents the class of our classification problem. Ten-fold cross-validation was then used to train the LLR classifier and, at the same time, obtain class probabilities for each gait instance in the dataset. Since the number of instances produced by different users is imbalanced (some users walk frequently, whereas others are much more sedentary), a technique based on random sampling was applied to each training fold in order to obtain homogeneous training sets [20].

The class probabilities were then used to find the classification results obtained by the LLR classifier, in terms of average *recall*. For a given class c, the recall indicates the likelihood that an instance belonging to that class will be classified correctly:

$$recall_c = \frac{true\ positives_c}{instances_c},$$

where $true\ positives_c$ is the number of instances belonging to c that are classified correctly, and $instances_c$ is the total number of instances belonging to c. The average recall for LLR was 95.3%.

The LLR classification performance was compared against some other widely-used learners, namely Neural Network, Random Forest, Rotation Forest, and Nearest Neighbor. For a fair comparison, the feature selection method mentioned in Section 2.2 was re-executed for each classification scheme, so as to find scheme-specific feature sets. The same evaluation procedure based on ten-fold cross-validation was then used.

Results are shown in Figure 3, in terms of average recall and number of selected features. LLR and Neural Network achieved the highest recall, while the number of selected features does not vary significantly among the considered learners.

3.4. Identification results

We here present the results related to the three proposed identification techniques: *single-instance* identification, *basic multi-instance* identification, and *adaptive multi-instance* identification. The metric used to evaluate the identification results is the *Correct Identification Rate (CIR)*:

$$CIR_j = \frac{Correct\ Identifications_j}{Identification\ Outputs_j},$$

where $Correct\ Identifications_j$ and $Identification\ Outputs_j$ are the number of correct identifications and the total number of identifications made for user j, respectively.

3.4.1. Single-instance identification
The *single-instance* identification technique produces an identification output for each gait instance. After that the LLR classifier has generated the class probabilities, identification is performed by simply assigning the instance to the user with the highest probability. Therefore, the CIR in this case corresponds to the proportion of gait instances belonging to a specific user that were classified correctly, and is thus
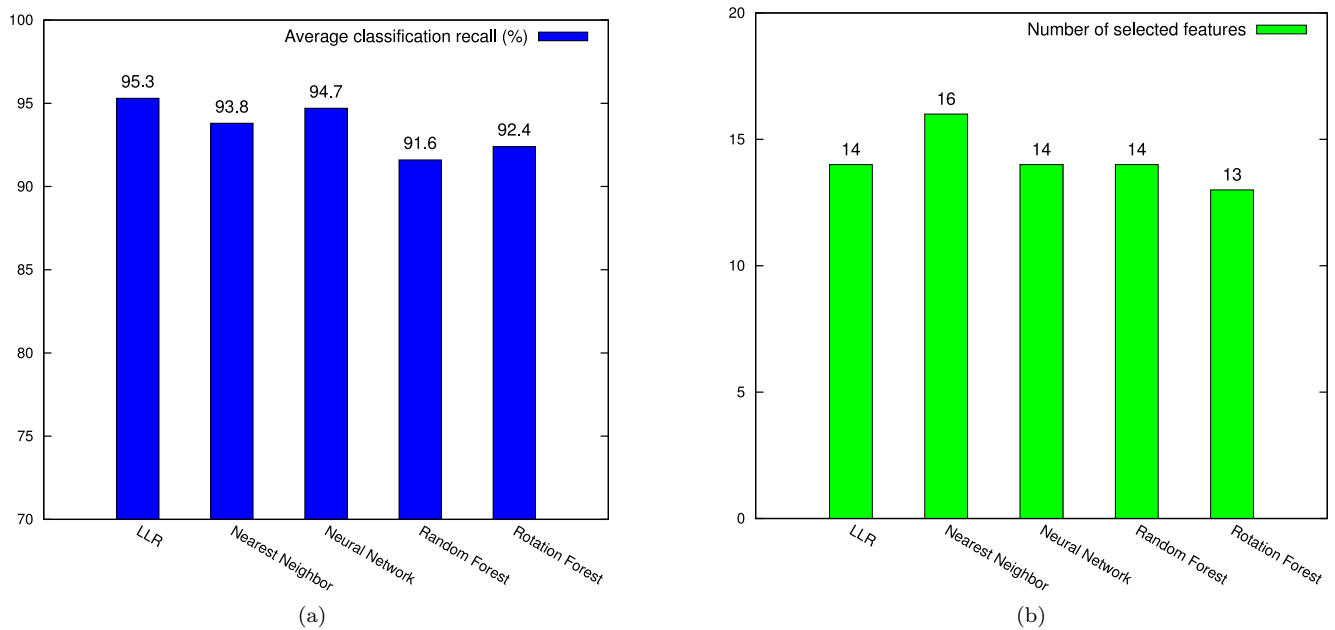
FIGURE 3. Classification schemes comparison in terms of average recall (a), and number of selected features (b).

TABLE 4. Single-instance identification results.

User	CIR %
1	97.2
2	91.3
3	93.8
4	97.9
5	97.0
6	92.5
7	96.7
8	97.7
9	91.6
10	96.9
Average	95.3

equivalent to the classification recall mentioned in Section 3.3.

Per-user identification results achieved by the single-instance technique are reported in Table 4 in terms of CIR. The proposed technique achieved more than 90% correct identifications on all the users in our dataset, and the average result was 95.3%.

3.4.2. Basic multi-instance identification

The *basic* multi-instance technique finds the average class probabilities over a window of gait instances belonging to the same user, and performs identification by choosing the user with the highest probability. This technique was applied to each user's data – the gait instances, with their respective class probabilities, were grouped into windows according to their chronological order.

Per-user results are shown in Table 5 according to different window size values. For this evaluation,

we actually used a sliding window to consider all the possible orderly sets with a specific window size, each window representing a possible input to the basic multi-instance technique. The second column (window size = 1) reports the same result shown in Table 4 (single-instance identification) and is shown here for direct comparison with the results achieved as the window size is increased. For most users it was possible to achieve perfect accuracy using less than 8 gait instances. For user 2, however, 40 gait segments per window were required to avoid errors.

3.4.3. Adaptive multi-instance identification

The *adaptive* multi-instance technique produces an identification output only if the probability *gap* between the users with the highest and second-highest probability values is above a predefined threshold. Otherwise, the oldest gait instance in the window is discarded and the procedure is repeated as soon as a new gait instance is found.

We tested the adaptive technique on our dataset using a window size of 6 gait instances and a gap threshold equal to 60%. These thresholds were found empirically. Results are shown in Table 6. The second column indicates the rate of correct identifications achieved with this method, while the third, fourth, and fifth columns indicate the average, standard deviation, and worst-case number of gait instances read before producing an identification output, respectively. This technique achieved perfect identification for all the users, requiring just 30 seconds of walking data on average (less than 7 gait instances, each lasting about 4 s).

TABLE 5. Basic multi-instance identification results.

User	CIR % with increasing window size								
	1	2	4	6	8	10	20	30	40
1	97.2	99.2	99.8	100	100	100	100	100	100
2	91.3	94.5	96.4	97.0	97.4	97.7	98.9	99.5	100
3	93.8	96.1	99.4	100	100	100	100	100	100
4	97.9	99.1	100	100	100	100	100	100	100
5	97.0	100	100	100	100	100	100	100	100
6	92.5	96.0	98.1	98.3	98.9	100	100	100	100
7	96.7	98.4	99.3	99.9	100	100	100	100	100
8	97.7	99.1	100	100	100	100	100	100	100
9	91.6	94.0	98.4	100	100	100	100	100	100
10	96.9	98.9	100	100	100	100	100	100	100
Average	95.3	97.5	99.1	99.5	99.6	99.8	99.9	99.9	100

TABLE 6. Adaptive multi-instance identification results.

User	CIR %	Required gait instances		
		Average	St. dev.	Worst case
1	100	6.1	0.5	13
2	100	8.0	6.8	53
3	100	6.5	1.7	18
4	100	6.1	0.7	11
5	100	6.0	0.2	12
6	100	7.0	2.6	25
7	100	6.2	1.0	14
8	100	6.3	1.5	23
9	100	7.4	3.1	26
10	100	6.1	0.7	11
Average	100	6.6	1.9	20.6

3.5. Discussion

Ten volunteers were involved in this study. This number is in line with other works in the field of gait-based identification [21–25]. However, differently from previous works, gait segments were extracted during the users' daily routine and without predefined constraints such as footwear, terrain/surface, or pace. These gait segments inevitably include a variety of gait patterns for each user, for example due to walking at different speed or on a different surface. This realistic setting makes the identification process much more challenging with respect to considering gait segments produced in controlled environment.

To tackle the presence of highly irregular gait patterns in real-world settings (for example produced while climbing stairs), the walking detection algorithm included filters based on standard deviation and autocorrelation. Notwithstanding these filters, it was possible to provide the learner with frequent "identification opportunities". Indeed, more than 100 gait segments per user were found in about ten hours of data collection, even for those users who spent most of their time indoors (e.g., user 4 and 5).

Acceleration features with low memory and computational requirements were considered for the feature selection process. Given the classification results achieved by different learning schemes (average recall > 90%), these features proved able to capture the most relevant characteristics from each user's gait pattern. Among considered learners, LLR showed the highest classification recall (95.3%).

The single-instance approach achieved more than 90% CIR for all the users in the dataset. This result is in line with the best performing algorithms in the field of gait-based identification, which typically relied on short gait experiments made in controlled environment. Single instance identification ensures fast identification, as it only requires eight consecutive steps.

In contexts where identification errors must be further reduced, the proposed *multi-instance* techniques are able to minimize the probability of identification errors. The *basic* multi-instance technique exploits a window of consecutive gait instances in order to perform identification on the average class probabilities. Thanks to the high single-instance CIR, by using 10 consecutive instances it is possible to eliminate errors for almost all the users (see Table 5). However, user 2, despite a single-instance CIR higher than 90% and similar to user 9, required a window with 40 consecutive instances to actually eliminate all the errors. The reason behind this result is that abnormal gait patterns may occur in bursts lasting a few minutes, for example when the user is temporarily carrying a load or performing any other unusual activity that the classifier has not learned properly. The use of a large window of

instances somehow fixes the problem, but has two main drawbacks: (i) using a relatively large window is unnecessary in most cases, and may add a delay to the identification process; (ii) a longer than expected burst of abnormal activity may still lead to identification errors.

The *adaptive* multi-instance technique addresses these shortcomings. Results have shown that the adaptive approach requires less than 7 gait instances, on average, to achieve perfect identification for all the users. That corresponds to less than 30 seconds of gait data (each gait segment lasts for about 4 seconds). When bursts of abnormal gait were found, identification was properly delayed in order to avoid errors, until the required level of confidence (probability gap) was reached. The worst case occurred, as expected, for user 2. Nevertheless, the technique achieved correct identification in about 30 seconds of gait on average, and in less than 4 minutes in the presence of the above mentioned burst of unusual gait activity.

To the best of our knowledge, this is the first time that a *self-contained* system for gait-based identification – thus including all the necessary components, from walking detection to classification – has been tested on data collected in uncontrolled environment. The results shown in this section prove the soundness of the proposed approach, and suggest that this technique may be effectively used for *long-term* and *trustworthy* identification of users in real-word settings.

4. IMPLEMENTATION AND EVALUATION OF POWER CONSUMPTION ON A MINIATURIZED DEVICE

In this section we show that the proposed techniques for gait-based identification can be implemented and executed on a miniaturized device such as the Shimmer 2r. In particular, the Shimmer embeds a TI MSP430 microcontroller with 8 MHz clock, 10 KB RAM, 48 KB flash memory, and a 450 mAh battery.

In the next subsections, we first demonstrate that the required computations can be executed within the required time and memory constraints. Then, we compare the proposed solution against continuous streaming of samples to an external device, in terms of energy consumption and battery duration.

4.1. On-node implementation and feasibility of real-time processing

The proposed method was implemented as a TinyOS 2.x event-driven application. TinyOS is characterized by two fundamental computational abstractions: *async events* and *tasks*. Tasks are executed in the background using a FIFO policy and are non-preemptive. Conversely, async events interrupt task execution when they are raised. Events can also schedule new tasks for later execution.

In our application there is a single *async* event, which is raised each time a new sample is read from the accelerometer (every $\sim$ 19.5 ms, corresponding to 51.2 Hz sampling frequency). This event is responsible for: (i) correcting the raw acceleration values of the new sample according to calibration parameters; (ii) saving the new sample into a circular buffer, by overwriting the oldest sample; (iii) updating the walking detection finite state machine, which implements *Gait Segment Detection*. Acceleration values coming from the ADC are corrected by using six calibration parameters, two for each axis, to compensate the slightly different scale and offset at rest (small differences between accelerometers are introduced by the manufacturing process). Corrected acceleration values are represented as 16-bit integers to reduce the computational load (with respect to using floating point). The circular buffer used to store the samples can contain up to 512 integers per each axis, corresponding to 10 seconds of data.

Considering the flowchart representation described in Figure 1, the async event only implements the first subtask of Walking detection – *Gait Segment Detection*. The rationale is that only this subtask can be executed in a time significantly shorter than the sampling period, and can thus be implemented in an *atomic* block of code. Actually, *Gait Segment Detection* is performed in less than 1 ms. Conversely, the autocorrelation-based filter requires more than a sampling period to be executed and must be implemented in a background task.

When a possible gait segment is detected, the async event schedules a new task. This task is responsible for performing the remaining computations required for identification, namely Autocorrelation-based filtering, preprocessing, feature extraction, LLR classification, and identification. The task is interrupted each time a new sample is read. To ensure that the system is able to keep up with the stream of acceleration data and perform real-time analysis, it is key to satisfy two time constraints: (i) a gait segment must be processed in less than the duration of the gait segment itself; (ii) the task must be completed before the gait segment under analysis is overwritten with new samples in the circular buffer.

As described in Section 3.2, the duration of gait segments in our experiments ranged from 3.3 to 5.9 s, while the average duration was 4.2 s. With a conservative approach, our algorithm was tuned to accept gait segments lasting up to 7.0 s. Taking into account the size of the circular buffer (10 s), it must be verified that the longest possible gait segment (7 s) is processed in less than three seconds, thus ensuring that the gait segment's samples are processed before overwriting occurs in the circular buffer. According to our experiments, the Shimmer's MCU is capable of performing single-instance identification on the longest gait segment in about one second. It is thus able to

keep up with the stream of acceleration samples and perform real-time identification of users. The additional processing required by the multi-instance identification techniques is negligible.

The entire application requires less than 8 KB of RAM with single-instance identification and 10 users. More specifically, the LLR classifier model requires only 60 bytes per class/user (fifteen 32-bit coefficients per class/user), and thus a total 600 bytes in our 10-user scenario. In multi-instance identification, it is necessary to save the class probabilities of window-size instances. With 10 users, each set of class probabilities is composed of ten 16-bit integers, corresponding to 20 bytes. For example the adaptive method, which requires 6 instances per window to achieve perfect identification, would lead to an additional RAM occupation of just 120 bytes.

This evaluation demonstrates that real-time execution of the proposed method on Shimmer2r devices is feasible in terms of both memory and processing requirements.

4.2. Power consumption – on-node processing vs radio streaming

The power consumption on the Shimmer device was evaluated according to two different implementation strategies: (i) *On-node Processing*, the strategy proposed in this paper, where all the required processing is performed on the wearable device; (ii) *Radio Streaming*, the opposite approach, where the wearable sensor is only responsible for collecting acceleration data and wirelessly transmits samples to an external device. In order to find the power consumption associated with each strategy, we first evaluated the power consumption associated to all the possible subtasks – the results are shown in Table 7.

The *Sampling* subtask consists in collecting acceleration samples at 51.2 Hz and saving them into a circular buffer. According to our experiments, *Sampling* requires ~ 3.0 mW. This baseline consumption cannot be avoided, regardless of the strategy adopted.

Radio represents the wireless transmission of samples collected at 51.2 Hz. It was implemented using the CC2420 IEEE 802.15.4 radio module of the Shimmer device. According to the CC2420 specifications, power consumption was estimated considering a 52.2 mW consumption while transmitting. A simple transmission protocol was implemented to ensure reliable transmission of samples. The radio is switched on every 5 s to send the samples collected in the last interval. The receiver is supposed to be always listening, and sends an acknowledgement to confirm the start and the end of each communication. The average consumption is ~ 2.9 mW.

Gait Segment Detection (GSD) is the first subtask performed to detect gait segments suitable for identification, as described in Figure 1. GSD requires

~ 0.5 ms of processing for acceleration sample, leading to an average consumption of ~ 0.4 mW.

The remaining subtasks – *Autocorrelation-based Filter (ACF)*, and *Gait-based Identification (GI)* – are applied only to detected gait segments, and thus consume energy only when the user is walking. More precisely, ACF is applied to each detected gait segment, while GI is applied only to those segments that are not filtered out by the autocorrelation-based filter. For these two subtasks, Table 7 shows the estimated power required during walking activity. The actual identification technique – single-instance or multi-instance – is not specified because the difference in terms of power consumption is negligible.

In the *Radio Streaming* approach two subtasks – *Sampling* and *Radio* – are executed continuously, regardless of what the user is currently doing. Conversely, *On-node Processing* leads to a different consumption depending on the user's behavior, as shown in Table 8. When the user is idle (i.e., not walking), the device continuously samples acceleration and searches for a new gait segment (S+GSD subtasks). When the user actually walks and a gait segment is detected, ACF is performed to test whether the gait segment is regular enough to be used for identification. Only in the latter case, the extra processing required for gait-based identification is actually executed.

Table 9 shows the average consumption and the estimated battery duration relative to the two considered implementation strategies. The estimation of *Radio Sampling* consumption is simply the sum of the consumptions due to sampling acceleration and transmitting the samples via radio. Instead, the average consumption relative to *On-node Processing* was found considering the average number of gait segments per hour produced by our volunteers, taking into account also those gait segments that were filtered by the ACF subtask.

The results confirm that the proposed strategy – On-node Processing – is feasible in terms of power consumption. Indeed, the expected battery duration is about 70% longer than the solution where all the samples are sent to an external device. In addition to improving battery duration, on-node processing enables identification even when the user is not in the transmission range of a gateway.

5. RELATED WORK

In this section we summarize the most relevant work concerning identification by means of mobile devices.

Gafurov et al. carried out pioneering work about gait-based identification using a wearable accelerometric sensor [26]. Fifty users were involved in the experiments. Each user walked for 20 meters while wearing a tri-axial accelerometer (operating at 100 Hz) in his/her trouser pocket (with one exception: a person was monitored by attaching the sensor to her

TABLE 7. Subtasks and associated consumption.

Subtask	Description	Average consumption
Sampling (S)	Sampling acceleration at 51.2 Hz	3.0 mW
Radio (R)	Radio transmission of samples	2.9 mW
Gait Segment Detection (GSD)	Detection of gait segments (8 consecutive steps)	0.4 mW
Autocorrelation-based Filter (ACF)	Filter to remove irregular gait segments	0.8 mW
Gait-based Identification (GI)	Preproc., feature extraction, classification, identification	1.8 mW

TABLE 8. On-node Processing approach consumption according to user's activity.

Activity	Subtasks	Average consumption
Idle	S+GSD	3.4 mW
Walking (filtered)	S+GSD+ACF	4.2 mW
Walking (identification)	S+GSD+ACF+GID	6.0 mW

belt). Six gait instances were collected for each person: four walking normally, and two carrying a backpack. Analysis was then executed on a PC. Results show a recognition rate of $\sim 86\%$ both with and without backpack. Such performance was obtained when using a method based on computing the average step cycle and comparing cycles on the base of absolute distance among samples.

Another prototypical system is described in [21]. Four users were monitored, in terms of tri-axial acceleration, using a Tmote-Sky node attached near one of the ankles and with predefined orientation. Sensor readings, collected at 20 Hz, were wirelessly sent to a computer where they were analyzed off-line. Perfect accuracy was obtained using a k-NN classifier and majority voting. However, the small number of subjects somehow reduces the significance of the achieved performance.

In [8] the use of an Android smartphone as a means for biometric identification is studied. Thirty-six individuals were enrolled in the study; each individual performed a set of predefined activities (walking, jogging, climb up/down stairs) in a supervised environment. Acceleration was collected at 20 Hz carrying the device in a front trouser pocket. Identification was performed off-line on a standard PC, using two classification techniques (J48 and neural networks). Results show that walking and jogging are the most important activities for identification purposes (they obtained accuracy levels in the $\sim 90-92\%$ range).

Another smartphone-based identification method is proposed in [22]. Gait data from ten volunteers were collected using a smartphone placed in a front trouser pocket. The orientation of the device was automatically estimated. Each volunteer was instructed to perform four walks. These gait instances were used to evaluate the proposed method, based on features such as frequency, symmetry, range and a similarity coefficient. Procedures were executed off-line. No identification errors were produced using a voting scheme that combined the above mentioned features.

In [23], techniques operating both in time and frequency domain are compared. Data were collected with participation of eleven volunteers, using the accelerometer embedded in a common smartphone. Sampling frequency was set at ~ 30 Hz and the smartphone was fixed, with known orientation, at the user's thigh. Twelve short walks (~ 30 s) were collected for each volunteer; five of them were used for training and the remaining ones for evaluating the performance. Support Vector Machine-based classification achieved $\sim 92\%$ recognition when using features in the frequency domain, and $\sim 79\%$ when using features in the time domain.

A method that does not rely on fixed placement of the sensor is described in [24]. Quantities that are invariant with respect to orientation, based on inner product and normalized cosine similarity, are computed on data collected using both an accelerometer and a gyroscope. A recognition rate of $\sim 85\text{-}87\%$ was obtained incorporating techniques borrowed from speaker recognition methods. The dataset comprised semi-naturalistic traces produced by 20 users on two separate sessions. Each session consisted in about fifteen minutes of walking. To measure acceleration, the users carried a smartphone in a trouser pocket without constraints in terms of device orientation.

The impact of different walking surfaces on gait identification is studied in [25]. Five subjects performed three walks on four different surfaces (ground, grass, gravel, stone plates); acceleration was collected at ~ 37 Hz using a smartphone attached to the right hip of subjects, and with predefined orientation. Continuous wavelet transformation, dynamic time warping, and principal component analysis were used to detect gait cycles, time align them, and estimate similarity. Results show that gait-based identification remains feasible even in the presence of different surfaces. The impact of phone placement is instead analyzed in [27], where 30 subjects were monitored while keeping the device in their right/left pocket and right/left hand. Determining phone position before user classification is investigated as a means to improve the performance of the system.

In [28], a gait recognition technique was implemented

TABLE 9. Implementation strategies and associated consumption.

Strategy	Average consumption	Expected battery duration
On-node Processing	3.5 mW	16.1 days
Radio Streaming	5.9 mW	9.5 days

on a commercially available smartphone. Thus, in contrast with the majority of other systems, both data collection and analysis were carried out on the mobile device. The proposed method does not depend on the orientation of the device, as it uses the Euclidean norm of acceleration. Samples were collected at approximately 150 Hz, with the device placed in the user's pocket, during three short walks in a controlled environment. Experiments were carried out with five known users, and 20 unknown ones (the system is able to both identify and authenticate users). The true positive rate ranges from $\sim$ 72%, when Manhattan is used as the distance metric and the technique is executed on the smartphone, to $\sim$ 89%, when other distance metrics are considered and executed on a standard PC.

A number of other works have faced the problem of authentication, which is solved using techniques that are similar to the ones used in identification. In [29] authentication is modeled as a one-class classification problem, since a model for non-authorized users cannot be easily built. An approach based on convex hull was evaluated on data collected using both a custom wearable system and a smartphone; a personalized activity recognition module was used to improve performance. Unobtrusive user authentication is discussed in [9]: the proposed solution, based on smartphones, does not assume a predefined position or orientation of the device, to improve usability. The study includes an experimental evaluation where eight subjects were monitored for a relatively long period in real world conditions.

Unfortunately, the performance indicated for the above mentioned works cannot be used to directly compare the different approaches, as they have been obtained using significantly different datasets (in terms of number of users, type and placement of device, length of monitoring, control of environmental conditions). Nevertheless, they provide a rough baseline for comparing future systems when operating in similar settings.

Almost all of the summarized studies evaluated their methods using datasets made of relatively short traces, collected in controlled or semi-controlled environments. To the best of our knowledge the only exception is [9], which focuses on authentication. With respect to the problem of identification, the longest traces – about 30 minutes per user – were used in [24]. The reason is due to the burden imposed on users when collecting long traces, and to the ease of setup for laboratory sessions.

6. CONCLUSION

Correct identification of users is a key element in a large number of applications for wearable devices, as it increases the security and usability of the system. We have presented a method for gait-based identification with little requirements in terms of both computational power and memory. More precisely, the method has been implemented and executed in real time on the Shimmer2r platform, which is equipped with a microcontroller unit running at 8 MHz and with 10 KB of RAM. Such low requirements, in turn, determine an extended lifetime on battery-operated devices. Despite being parsimonious, the proposed method provides excellent results in terms of correct identification rate ($\sim$ 95%). Even better results have been achieved by combining the output associated to a sequence of gait instances (multi-instance approach), eventually achieving perfect results. In particular, this can be obtained by either combining larger and larger sets of gait instances or by delaying the identification of the user until a given confidence level is reached.

It is important to highlight that the method has been evaluated in uncontrolled environments, where the users were free to move and act without restrictions. The collected dataset is rather large (3073 gait segments, $\sim$ 98 hours) with respect to many previous gait-based identification experiments, thus making the achieved results particularly significant. To the best of our knowledge, this is the first time that gait-based identification is tested in real-world settings. In the present work, the volunteers carried the device in a trouser pocket – future work will also consider different body positions, such as a wrist-worn device.

To encourage further advancements in the field and enable direct comparison with the proposed method, the acceleration traces are publicly available at the following address: `http://vecchio.iet.unipi.it/gaitanalysis`.

REFERENCES

[1] Kranz, M., Möller, A., Hammerla, N., Diewald, S., Plötz, T., Olivier, P., and Roalter, L. (2013) The mobile fitness coach: Towards individualized skill assessment

using personalized mobile devices. *Pervasive and Mobile Computing*, **9**, 203–215.

[2] Valenza, G., Nardelli, M., Lanat, A., Gentili, C., Bertschy, G., Paradiso, R., and Scilingo, E. P. (2014) Wearable monitoring for mood recognition in bipolar disorder based on history-dependent long-term heart rate variability analysis. *IEEE Journal of Biomedical and Health Informatics*, **18**, 1625–1635.

[3] Giordano, S. and Puccinelli, D. (2015) When sensing goes pervasive. *Pervasive and Mobile Computing*, **17**, 175–183.

[4] Raskovic, D., Martin, T., and Jovanov, E. (2004) Medical monitoring applications for wearable computing. *The Computer Journal*, **47**, 495–504.

[5] Barshan, B. and Yüksek, M. C. (2014) Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *The Computer Journal*, **57**, 1649–1667.

[6] Bianchi, L., Angelini, D., and Lacquaniti, F. (1998) Individual characteristics of human walking mechanics. *Pflügers Archiv*, **436**, 343–356.

[7] Karaoğlan, D. and Lev, A. (2014) A survey on the development of security mechanisms for body area networks. *The Computer Journal*, **57**, 1484–1512.

[8] Kwapisz, J., Weiss, G., and Moore, S. (2010) Cell phone-based biometric identification. *Proceedings of the Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS)*, Washington, DC, USA, September, pp. 1–7. IEEE.

[9] Lu, H., Huang, J., Saha, T., and Nachman, L. (2014) Unobtrusive gait verification for mobile phones. *Proceedings of the 2014 ACM International Symposium on Wearable Computers (ISWC)*, Seattle, WA, USA, September, pp. 91–98. ACM.

[10] Cola, G., Vecchio, A., and Avvenuti, M. (2014) Improving the performance of fall detection systems through walk recognition. *Journal of Ambient Intelligence and Humanized Computing*, **5**, 843–855.

[11] Mizell, D. (2003) Using gravity to estimate accelerometer orientation. *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC)*, Washington, DC, USA, October, pp. 252–253. IEEE.

[12] Hall, M. A. and Holmes, G. (2003) Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, **15**, 1437–1447.

[13] Leys, C., Ley, C., Klein, O., Bernard, P., and Licata, L. (2013) Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, **49**, 764 – 766.

[14] Cola, G., Avvenuti, M., Vecchio, A., Yang, G.-Z., and Lo, B. (2015) An on-node processing approach for anomaly detection in gait. *IEEE Sensors Journal*, **15**, 6640–6649.

[15] Cola, G., Avvenuti, M., Vecchio, A., Yang, G.-Z., and Lo, B. (2015) An unsupervised approach for gait-based authentication. *Proceedings of the 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, Cambridge, MA, USA, June, pp. 1–6. IEEE.

[16] Moe-Nilssen, R. and Helbostad, J. L. (2004) Estimation of gait cycle characteristics by trunk accelerometry. *Journal of Biomechanics*, **37**, 121 – 126.

[17] Landwehr, N., Hall, M., and Frank, E. (2005) Logistic model trees. *Machine Learning*, **59**, 161–205.

[18] Friedman, J., Hastie, T., Tibshirani, R., et al. (2000) Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, **28**, 337–407.

[19] Shimmer Research (Online, accessed December 9, 2016). Wearable Wireless Sensing Technology and Solutions. http://www.shimmersensing.com.

[20] Batista, G. E., Prati, R. C., and Monard, M. C. (2004) A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, **6**, 20–29.

[21] Annadhorai, A., Guenterberg, E., Barnes, J., Haraga, K., and Jafari, R. (2008) Human identification by gait analysis. *Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments (HealthNet)*, Breckenridge, CO, USA, June, pp. 11:1–11:3. ACM.

[22] Sun, B., Wang, Y., and Banda, J. (2014) Gait characteristic analysis and identification based on the iPhones accelerometer and gyrometer. *Sensors*, **14**, 17037–17054.

[23] Thang, H. M., Viet, V. Q., Thuc, N. D., and Choi, D. (2012) Gait identification using accelerometer on mobile phone. *Proceedings of the International Conference on Control Automation and Information Sciences (ICCAIS)*, Saigon, Vietnam, November, pp. 344–348. IEEE.

[24] Zhong, Y. and Deng, Y. (2014) Sensor orientation invariant mobile gait biometrics. *Proceedings of the IEEE International Joint Conference on Biometrics (IJCB)*, Clearwater, FL, USA, Sept, pp. 1–8. IEEE.

[25] Sprager, S. and Zazula, D. (2011) Impact of different walking surfaces on gait identification based on higher-order statistics of accelerometer data. *Proceedings of the IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Kuala Lumpur, Malaysia, November, pp. 360–365. IEEE.

[26] Gafurov, D., Snekkenes, E., and Bours, P. (2007) Gait authentication and identification using wearable accelerometer sensor. *Proceedings of the IEEE Workshop on Automatic Identification Advanced Technologies*, Alghero, Italy, June, pp. 220–225. IEEE.

[27] Primo, A., Phoha, V., Kumar, R., and Serwadda, A. (2014) Context-aware active authentication using smartphone accelerometer measurements. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Columbus, OH, USA, June, pp. 98–105. IEEE.

[28] Derawi, M. and Bours, P. (2013) Gait and activity recognition using commercial phones. *Computers & Security*, **39**, 137–144.

[29] Casale, P., Pujol, O., and Radeva, P. (2012) Personalization and user verification in wearable systems using biometric walking patterns. *Personal and Ubiquitous Computing*, **16**, 563–580.