This is the author's accepted manuscript (AAM) of an article published in *Pervasive and Mobile Computing (Elsevier)*. The final published version is available at: `https://doi.org/10.1016/j.pmcj.2021.101483`**.**

Continuous authentication through gait analysis on a wrist-worn device

Guglielmo Cola[a,b,*], Alessio Vecchio[b], Marco Avvenuti[b]

[a] *Istituto di Informatica e Telematica, CNR, Via G. Moruzzi, 1, 56124, Pisa, Italy*
[b] *Dip. di Ingegneria dell'Informazione, University of Pisa, Largo L. Lazzarino 1, 56122, Pisa, Italy.*

Abstract

Being distinctive of every individual, gait can be used as a biometric feature to authenticate the owner of a wearable device. This paper proposes and evaluates an authentication method that relies on the acceleration signal acquired at the user's wrist. During the training phase, the wrist-worn device automatically learns the gait patterns of the legitimate user, by exploiting a set of acceleration-based indicators. Subsequently, unauthorized users are detected by observing the occurrence of anomalous gait patterns. Experimental results carried out with 20 volunteers show that the proposed method is able to recognize the legitimate user with an equal error rate of $\sim 2.5\%$. The method is sufficiently lightweight to be executed in real time on a wearable device with limited resources. This enables continuous authentication without requiring the presence of an external device (e.g., a smartphone). Furthermore, the provided evaluation of power consumption shows that the completely on-node solution is also more energy efficient with respect to off-loading computation to an external device.

Keywords: Gait-based continuous authentication, Smartwatch, Wearable sensor, Wrist-worn accelerometer.

1. Introduction

Smartwatches and other wrist-worn devices are used for a growing variety of activities, from monitoring fitness and well-being to contactless payments and remote control of home appliances [1]. The intimate relationship between these devices and their owners obviously poses some concerns in terms of security and privacy. Collected information is extremely personal, and could be used to derive the user's behavior and health condition. In some cases, wrist-worn devices are also used to automatically unlock a smartphone, as their proximity implicitly testifies the presence of the owner.

*Corresponding author
Email addresses: guglielmo.cola@iit.cnr.it (Guglielmo Cola), alessio.vecchio@unipi.it (Alessio Vecchio), marco.avvenuti@unipi.it (Marco Avvenuti)

Hence, a stolen smartwatch or wristband can be used to gain access to the victim's data on the smartphone, where other sensitive data are frequently stored.

In the last years, high-end smartphones have started to include authentication methods based on biometric features, such as fingerprint and iris. These techniques make authentication faster and more user friendly, but require additional sensors and cannot be easily implemented on wearable devices with a small form factor. Moreover, they only authenticate users when they start interacting with the device, and do not represent a solution for continuous authentication [2].

In this context, the use of acceleration patterns as a biometric feature has been explored to passively and continuously authenticate the user of a wearable device, without requiring explicit interaction [3]. Most of these methods rely on walking activity to recognize the user. The reason is twofold: i) walking occurs frequently throughout the day, thus providing numerous hints about user's identity; ii) gait, i.e. a person's manner of walking, is highly specific [4], thus increasing the chance of detecting unauthorized users.

Continuous authentication is also useful for wearable devices used in healthcare [5]. As observed in some telemedicine applications, remotely monitored patients may be tempted to cheat caregivers by giving their devices to other people (e.g., to reach the prescribed activity levels [6, 7]). Similarly, during unsupervised clinical trials it should be verified that the expected patient is wearing the device at all times [8]. In these situations, gait-based authentication is useful to detect misbehaving users and ensure data integrity.

Research on gait-based biometric methods applied to wearable devices has focused on two aspects: identification and authentication [9]. In *identification*, the group of possible users of a shared device is known in advance, and the system aims to find the identity of the current user (within the group) [10, 11]. The typical approach is supervised learning: the gait patterns of the different users are learned during the initial training phase and then used to build a recognition model.

This paper is focused on *authentication*, which aims to understand if the current user is the legitimate one or someone else. Authentication is a "one vs. all" problem: the system only knows the device owner's gait pattern, whereas other possible users (impostors) are not known in advance. The gait style of the legitimate user is learned during the initial period of use, subsequently anomalies occurring in a set of acceleration-based features are used to detect possible impostors.

The method we propose leverages acceleration data captured at the user's wrist, for instance when the used wears a smartwatch or a wristband. Authenticating the user by means of a wrist-worn device can be useful in several situations. For instance, let us consider a user who brings her smartphone in a bag or a backpack. In this case, the smartphone cannot be used for authenticating the user as the gait style cannot be properly captured, but this is still possible using the wrist-worn device. Moreover, it is important to notice that we do not consider wrist-based authentication to be useful only in combination with a smartphone. Authenticating the user can be valuable for the wrist-worn device itself or in combination with other devices. A smart-wristband

can be used, for instance, to grant access to a building or to a car. In such a scenario, traditional cryptographic techniques can be strengthened thanks to continuous and unobtrusive user's authentication. Since hands are generally subject to more spurious movements, the analysis of acceleration at the wrist poses new difficulties with respect to those methods relying on a sensor closer to the user's center of mass (e.g., over the trunk or inside a trouser pocket). For this reason, gait-based authentication performed by means of devices like smartwatches has been scarcely investigated in previous work.

The contribution of this work in the field of gait-based authentication with wearable sensors can be summarized as follows:

- Despite the challenges deriving from the wrist-based position, the proposed method showed high authentication accuracy. According to our experiments, the use of a wirst-worn device does not lead to significantly worse results than using a device in a front trouser pocket.
- The method was designed to be not demanding in terms of computational and memory resources. All operations can be executed on a microcontroller with 8 MHz clock and 10 KB of RAM. This is a key step in the direction of continuous authentication on wearable devices, as previous studies typically relied on smartphones and/or performed authentication tasks off-line.
- The power consumption of a completely on-node solution was compared with a solution that relies on the presence of an external device for executing the authentication task. The comparison shows that, with current technologies, the on-node solution is more energy efficient (+60% of battery duration).

The method for user authentication is presented in Section 3. Section 4 describes the experiments, whereas authentication results are presented and discussed in Section 5. Section 6 presents the implementation in miniaturized devices and the evaluation of power consumption. Finally, Section 7 concludes the paper.

2. Related work

A wide variety of gait-based methods to recognize the users of wearable devices have been proposed during the last years [12, 13]. The vast majority of these works relied on dedicated devices or smartphones placed near the user's waist, while the use of wrist-worn devices has not been thoroughly explored.

One of the first studies in this field is [14]: the acceleration trace of the current user was compared with the typical step of the legitimate user using cross-correlation. Experiments involved 36 users and a waist-mounted accelerometer, and the method achieved a correct authentication rate of 88%. Another pioneering work is [15], where different metrics for evaluating the distance between gait segments were discussed. Fifty volunteers collected six different gait segments while wearing an accelerometer inside a trouser pocket. Absolute distance proved to be the most discriminating metric, with an equal error rate (EER) of $\sim 7\%$.

Three existing methods for gait-based authentication ([16, 17, 18]) and a novel one were evaluated on a set of traces collected from 744 users in [19]. For each user two gait segments were acquired: the first to train the system and the second for evaluation. The large number of users makes the evaluation significant in terms of users' heterogeneity. However, the limited number of gait segments per user, and their acquisition in a short time frame, make obtained results less conclusive if analyzed along other dimensions.

Verification for smartphones using gait-based information is described in [20]. The method was based on two Gaussian Mixture Models: a user gait model (trained on the owner), and a Universal Background Model (trained on other subjects). The output from the two models is used to understand whether the current user is the legitimate one. Authentication experiments were carried out in controlled conditions by 12 volunteers. Each subject carried the phone according to his/her preferences, and carried out some activities, including sitting, standing, walking, and biking. The method obtained an EER of about 14%. An additional experimental evaluation was carried out in uncontrolled conditions (eight subjects for two/three weeks).

The impact of device position on gait-based authentication accuracy was also studied [21]. Four possible positions were considered for a smartphone: left/right hand and left/right pocket. The effects caused by the actual position on the feature set and, in turn, on authentication results were evaluated with 30 volunteers. The authors concluded that knowing the position of the device greatly improves the accuracy of the authentication procedure. Obviously, this also makes the overall method more complicated, as it has to automatically determine the position where the smartphone is held. The use of multiple accelerometers placed at different body positions was studied in [22]. Experiments in controlled settings with 30 users show that combining information generated at different position improves identification accuracy. This, however, comes at the cost of reducing usability.

The problem of authenticating/identifying the user of a smartwatch was faced in [23], which considered acceleration and angular speed features as inputs to a supervised classifier. Experiments consisted in five-minute traces collected by 59 volunteers. Ten-second gait instances were extracted from these traces. Authentication was based on a supervised classifier, hence the training set required instances from both classes – the legitimate user and the impostors. For each user, an authentication model was built using his/her gait instances (legitimate user) and instances from four random subjects (impostors). Then, impostor detection rate was evaluated on four other subjects. Acceleration features combined with Random Forest or Rotation Forest obtained authentication accuracy levels of 98%, whereas identification with Rotation Forest showed 84% accuracy.

An example in the healthcare domain is [24]. The vertical component of acceleration is sampled at 50 Hz using a smartphone. Then, gait cycles are extracted using the correlation coefficient and interpolated to be aligned with a reference template. The verification procedure, when executed on the mobile device, relies on weighted Pearson correlation coefficients. Another possibility is the use of a Support Vector Machine on

an external server. The method was experimentally evaluated on a dataset collected involving 26 users for six months. Results show an 80% detection rate when using the procedure executed on the mobile device. When using the procedure on the external server the detection rate increases to 90%.

LiSA-G is an authentication system based on gait that relies on information collected at the wrist using a commercial smartwatch [25]. Data collected by the accelerometer and gyroscope are transferred from the smartwatch to a smartphone via Bluetooth, and then forwarded to an authentication server. The latter extracts the features and produces the result through a trained classifier. More recently, the possibility of using the accelerometer embedded within smart-glasses has been considered [26]. Glasses are worn at a convenient position for gait analysis, as their orientation is fixed and head is less subject than hands to spurious movements. However, the use of smart-glasses is nowadays extremely reduced, and it is still not clear if they will be widely adopted by the general public in the near future. The possibility of authenticating the user through information collected by smart-shoes was studied in [27]. In particular, data collected by IMUs placed at the user's feet and distance data originated by Ultra-WideBand transceivers proved to be effective for the considered problem. Evaluation was carried out off-line, on a standard PC. Similarly to the glass-based solution, the approach based on smart-shoes testifies the interest in wearable solutions that are not smartphone-based. Others studied the possibility of authenticating the owner using not only acceleration, but also the pattern formed by veins in the wrist region [28]. This supports the validity of the considered body position for authentication purposes (accuracy can be improved by adopting multi-modal solutions).

Another closely related work is [29], where the accelerometer and gyroscope sensors available on common smartwatches and smartphones are used for behavioral authentication. Such study however, addresses different research questions compared to the ones faced in this article: Weiss et al. investigate the possibility of authenticating the user when performing some activities of daily living, such as walking, typing, clapping, eating pasta, and others. Moreover, they evaluated the authentication accuracy that can be reached using the single sensors (accelerometer/gyroscope) and the different devices (smartwatch/smartphone), or their combinations. All the considered activities performed reasonably well in terms of biometric effectiveness and gyroscopes proved to be less useful than accelerometers. Their results also show that walking is consistently one of the best activities from the point of view of authentication. Considering that it is also practical to be collected, we believe walking is the activity that should be considered as the first choice for the continuous authentication problem. As mentioned, our work starts from the idea of designing a method that can be executed on devices with extremely constrained resources and that can be executed completely on-node to save energy, two aspects that were not considered in [29].

In [30], a method for authenticating the users by observing their hand movements when typing is presented. The method, like most behavioral biometrics approaches, is meant to be an additional authentication factor rather than a standalone authentication system, and relies on information collected at the wrist using a smartwatch.

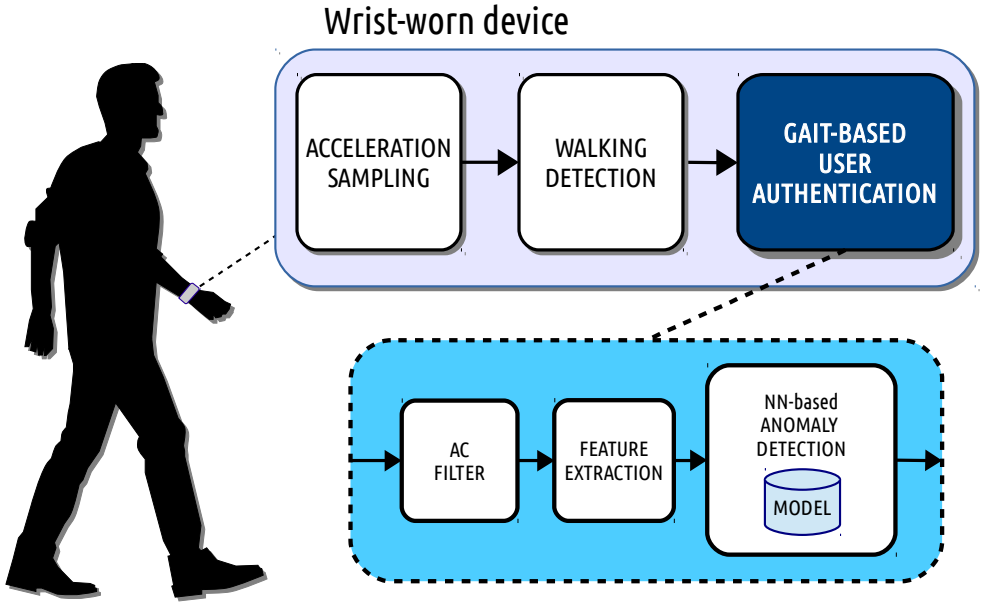


Figure 1: Flowchart representation of the proposed method.

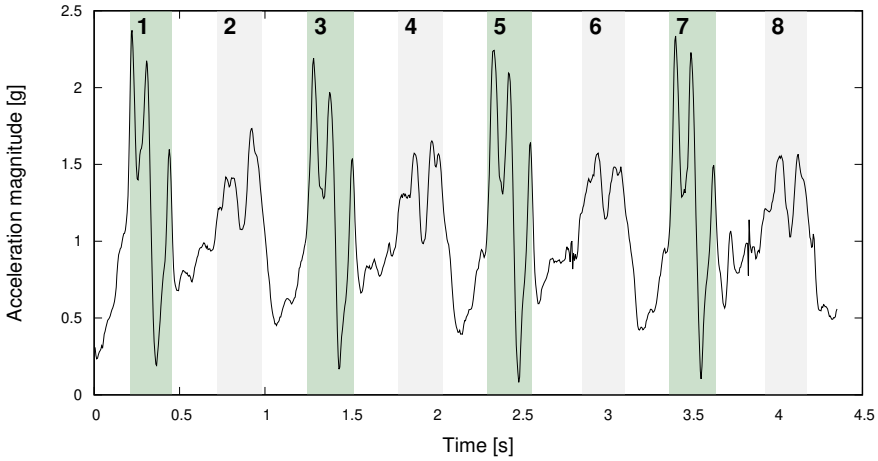
Data produced by the accelerometer and gyroscope during keystrokes is forwarded to a server where features are extracted and compared to the user profile. This again confirms the relevance of authentication using wearable devices, albeit in a different scenario (e.g., when the user is in front of a workstation).

3. Method

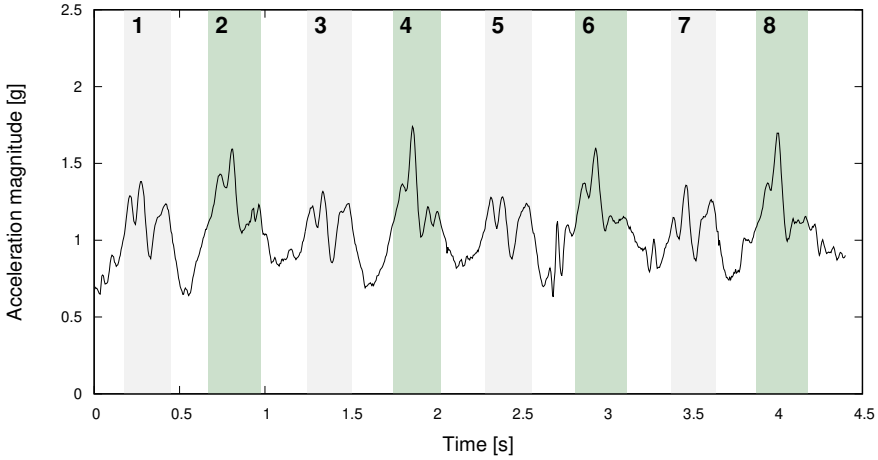
The proposed method for gait-based authentication is described in Figure 1. The user wears a device embedding a tri-axial accelerometer at his/her wrist. Acceleration samples are used as inputs to *walking detection*, which detects *gait segments* made of four consecutive gait cycles (i.e., eight steps). The core of the method is represented by the *gait-based user authentication* module, which provides authentication exploiting a gait segment. Each gait segment is processed by an autocorrelation-based filter (*AC filter*), aimed at discarding segments that are not suitable for reliable authentication. If a segment is not discarded, it is processed by *feature extraction* to obtain a feature vector. Hereafter, we refer to the feature vector resulting from feature extraction as *gait instance*. Finally, *anomaly detection* is used to understand if a gait instance has been produced by a legitimate user or by an unauthorized one.

3.1. Acceleration sampling and walking detection

The raw components of acceleration (x, y, z) are sampled at the wrist with a frequency of 51.2 Hz and converted to g units. The walking detection module takes as input the acceleration magnitude (Euclidean norm) $m = \sqrt{x^2 + y^2 + z^2}$. This reduces the computational load as a single signal has to be processed instead of three. In



(a) Pocket trace (front trouser pocket)



(b) Wrist trace

Figure 2: Comparison of the acceleration signal produced by gait in a pocket and at the wrist.

particular, the walking detection module is based on the peak-detection algorithm described in [31]: groups of peaks occurring in the acceleration signal at each foot contact (step) are detected, and when eight consecutive steps are found, a simple regularity test based on standard deviation is performed. If the test is passed, a new gait segment is detected. This walking detection technique is not computationally demanding and can be executed in real time on constrained devices. As reported in [31], this method is very accurate in detecting gait segments.

Such a technique was originally conceived to use the acceleration collected by a pocket-worn device (or other positions in proximity of the waist). Thus, we had to tune the technique to make it work properly when acceleration is collected at the wrist. The reason is that, at the wrist, the amplitude of the signal is generally smaller than the one obtained when the collection is carried out in a pocket or, more generally, on the lower trunk. This is due to the fact that the body acts as a transmission medium

and wrists are at a greater distance from the sources of vibrations (the feet impacting the ground). Attenuation is particularly large when the foot impacting the ground is the one of the contralateral leg.

An example is shown in Figure 2, which depicts the acceleration signal of the same gait segment sampled in a pocket (front trouser pocket) and at the wrist. The groups of peaks occurring at each step are highlighted with shaded areas and numbered. Green areas identify the steps made with the foot closer to the device. In the example, the sensors were worn at opposite sides (e.g., one in the left pocket, the other one on the right wrist), hence the steps marked in green for pocket and wrist are represented by odd and even steps, respectively. As mentioned above, these steps produce a sensibly higher acceleration amplitude, as the sensor is closer to the impact of the foot with the ground. For the same reason, the peaks in the pocket trace are sensibly higher than those in the wrist trace.

To cope with this problem, we lowered the threshold used for detecting peaks in the acceleration (m) signal. The new threshold was determined experimentally to ensure that both the peaks produced by the foot on the same side of the wrist-worn accelerometer and the ones produced by the contralateral leg are detected. Unfortunately, this simple approach may lead to a reduction in specificity, as spurious hand movements are more likely to be erroneously detected as steps. This new problem is tackled by the AC filter described in the following subsection.

3.2. AC filter

The AC filter was designed to discard gait segments that are not suitable to describe the legitimate user's gait pattern and may hinder authentication accuracy. There are two main examples: firstly, fake gait segments detected while the user was not walking, for instance because of spurious hand movements; secondly, gait segments detected while the user was actually walking, but marked by a highly irregular pattern. The latter ones may be caused by hand movements, by a sudden change in pace, or because of temporarily losing balance.

The filter consists of two regularity tests based on the autocorrelation coefficients of acceleration magnitude: only if both tests pass the gait segment is retained for authentication.

First of all, magnitude samples are processed by using a 2nd-order Butterworth low-pass filter with a cut-off frequency equal to 20 Hz. Then, autocorrelation coefficients are found as indicated in [32]:

$$AC_k = \frac{1}{N-k} \sum_{i=1}^{N-k} r_i * r_{i+k}, \tag{1}$$

where AC_k is the autocorrelation coefficient at time lag k; N is the number of magnitude samples in the gait segment; r_i is the ith magnitude sample minus the mean of the magnitude samples in the gait segment. The mean is removed from acceleration magnitude to remove any offset from the signal. Before being analyzed, the coefficients are normalized to 1.0 at zero lag.

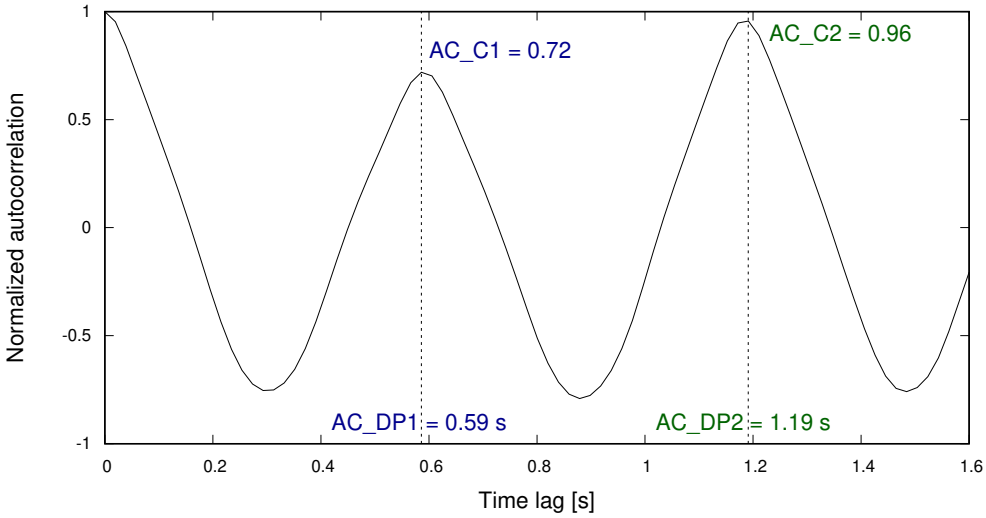


Figure 3: Autocorrelation example.

A cyclic signal will produce autocorrelation coefficients with peak values for lags equivalent to the cyclical components of the signal (often referred to as dominant periods). In gait segments, there are two cyclical components: steps and strides (i.e., gait cycles). By plotting the autocorrelation coefficients of a gait segment against the time-lag, it is possible to easily identify two peaks: the first peak occurs at a time lag that represents an estimate of the average duration of single steps (first dominant period, AC_DP1), while the second peak occurs at a time lag approximately equal to the average duration of strides (second dominant period, AC_DP2). Hence, AC_DP1 and AC_DP2 can be used to estimate the average duration of steps and strides. Moreover, the normalized value of these two peaks, AC_C1 and AC_C2, can be used to determine the regularity of steps and strides, respectively. For instance, if AC_C2 is close to 1, the subject's gait cycles are extremely regular. On the other hand, a relatively low peak value indicates a less regular gait pattern. In the case of AC_C1, however, it should be observed that a low value may occur if the sensor is positioned on a side (e.g., on a wrist, or in a trouser pocket), as this positioning causes inevitable differences in the measured acceleration signal for consecutive steps.

An example is shown in Figure 3, where normalized autocorrelation of gait acceleration measured at the wrist is plotted against the time lag. The two dominant periods are clearly visible at lag 0.59 s and 1.19 s. The normalized values of the peaks at the first and second dominant periods are 0.72 and 0.96, respectively. This denotes a relatively high gait cycle regularity, whereas the lower AC_C1 value is expected as the sensor is positioned on a side (wrist), producing different patterns for left and right steps. In our method, the two peaks, and thus the two dominant periods, are identified by using a simple peak detection techinque. Peak detection searches for a local maximum in two time lag intervals determined considering the typical min/max values for steps and strides [33].

The first test aims to verify if the duration of a gait segment is consistent with the

second dominant period of its autocorrelation (AC_DP2). Considering that a segment is made of four gait cycles, its duration should be:

$$duration = (4 \times AC_DP2) + \epsilon, \tag{2}$$

where ϵ is the approximation error. The gait segment is retained only if the absolute error $|\epsilon|$ is below a predefined threshold ϵ_{th}. A relatively high absolute error may be due to gait cycle detection errors. For example, the peaks belonging to two consecutive steps are merged into a single step, or a single step is split into two steps. These kinds of errors typically occur because of spurious hand movements.

The second test is based on gait cycle regularity. If the normalized coefficient AC_C2 is below a threshold AC_C2$_{th}$, the gait segment is discarded. A low regularity indicates that the user was not walking according to a consistent pattern.

Both thresholds (ϵ_{th} and AC_C2$_{th}$) were determined experimentally as a trade-off between gait detection sensitivity and authentication accuracy.

3.3. Feature extraction

A gait segment is a vector of acceleration samples collected on the three axes. The three components ($\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$) are measured according to the *local* coordinate system of the wrist-worn accelerometer. When the acceleration is collected using a wearable accelerometer, two factors must be considered: i) the device where the accelerometer is contained can be worn on different parts of the body and can be oriented in different ways with respect to the considered body part; ii) the part of the body where the device is worn on is subject to additional movements/rotations. As far as the first factor is concerned, let us consider a smartphone: it can be placed in a trouser pocket or in a jacket pocket, and even when it is always carried in the same place, say the trouser pocket, it can be oriented in different ways, as the user may extract the smartphone from the pocket, check the email, and then put the smartphone again in the pocket but now upside-down. This does not happen with a smartwatch, as it is always worn on the wrist and it is always oriented in the same way with respect to the wrist itself (we suppose that the watch is always worn on the same arm). As the second factor is concerned, let us consider a device placed on the user's trunk and another device attached to an arm. Let us also suppose that they both do not change their orientation with respect to the part the body they are attached to. The second one is going to be subject to additional movements and rotations (it is attached to one of the limbs) compared to the first one (closer to the center of mass). For the considered problem, i.e. authenticating the user, the first factor can be more troublesome than the second one: if acceleration is collected wearing the device on different parts of the body the user's gait patterns can be more difficult to be recognized. In addition, the possibly different orientations of the device have to be detected and managed. On the contrary, the additional movements and rotations at the wrist are part of the gait style of the user and thus are less a confounding factor (arm swinging can be considered part of the gait style of the specific user). Spurious movements that are not related with gait can be filtered out thanks to the regularity test.

The acceleration magnitude vector **m**, derived from (**x**, **y**, **z**), is insensitive to the orientation and rotations of the device. This positive aspect comes at the cost of discarding some information, as the acceleration direction is lost. We also considered the possibility of extracting features from a *global* reference system based on the direction of gravity. Such a reference system is relevant because it does not assume a specific orientation of the device with respect to the user's body. Thus, two additional vectors were derived: the *vertical acceleration* **v**, defined as the acceleration along the direction of gravity, and the *horizontal acceleration magnitude* **h**, defined as the magnitude of the acceleration projected on a plane orthogonal to gravity (the horizontal plane). The technique used to find **h** and **v** is described in [34].

As a result, features can be extracted from six vectors: **x**, **y**, **z**, **m**, **h**, and **v**.

In the remaining, a suffix is used to make clear which is the vector involved in computing a feature. For instance, $mean_x$ indicates that the mean is computed on the **x** vector.

The list of considered features was built using 12 widely-used functions from the statistical and signal processing domains: max, min, mean, median, kurtosis, skewness, standard deviation (SD), interquartile range (IQR), mean crossing rate (MCR), median absolute deviation (MAD), root mean square (RMS), and peak-to-peak amplitude (P2P). Average absolute variation (AAV) has been used in other works for the analysis of gait and fall detection [35]. AAV is found as:

$$AAV = \sum_{i=1}^{N-1} \frac{|s_{i+1} - s_i|}{N-1}, \tag{3}$$

where N is the number of samples in the segment, and s_i is the ith sample in the segment. The four autocorrelation features (AC_C1, AC_C2, AC_DP1, AC_DP2) were found only on acceleration magnitude, as described in Section 3.2. The duration of a gait segment is also considered a feature. Overall, for each gait segment, 83 features are produced: the 12 functions and AAV computed on the 6 vectors, the 4 autocorrelation features computed on **m**, and the duration of the gait segment. We then considered three subsets as the starting points for feature selection as described in Section 4.2.

The result of feature extraction is a *gait instance*, which consists in a vector of features.

3.4. Detection of anomalies

The authentication technique relies on *semi-supervised* anomaly detection [36]. It is a binary classification task, where the set used to train the classification model (training set) includes only instances belonging to the "normal class" (one-class classification). In the context of gait-based authentication, the normal class is represented by the gait instances belonging to the legitimate user.

Differently from supervised classification, this approach does not require to learn and model the gait patterns of unauthorized users, and only the legitimate user has to provide an initial set with his/her gait instances (training phase). This can be easily

accomplished in the first few days of use, by temporarily adopting alternative means for authentication to ensure that the collected instances are genuine.

More specifically, the proposed approach uses the Euclidean distance to the nearest neighbor (NN) in the training set to assign an anomaly score (AS) to a new gait instance. The instance is then classified as *normal* (legitimate user) or *abnormal* (unauthorized user) by comparing AS against a threshold. NN analysis has long been used to perform anomaly detection [36, 37, 38].

More formally, let $T = \{t_1, ..., t_M\}$ be the set of instances belonging to the legitimate user and collected during the training phase (training set). The Euclidean distance between two gait instances a and b is indicated as $dist(a, b)$. To find AS for a gait instance g, the first step is to calculate the distance to its NN in T:

$$distNN(g) = dist(g, n_g), \tag{4}$$

where n_g is the NN of g in T. Then two normalization parameters are derived from the M training set instances, namely the average NN distance

$$\mu_T = \frac{1}{M} \sum_{i=1}^{M} distNN(t_i), \tag{5}$$

and the standard deviation

$$\sigma_T = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (distNN(t_i) - \mu_T)^2}. \tag{6}$$

Finally, AS is found for the gait instance g as follows:

$$AS_g = \frac{distNN(g) - \mu_T}{\sigma_T}. \tag{7}$$

For instance, if μ_T in the training set is 1.0 and σ_T is 0.2, then an instance with NN distance to the training set of 1.4 is characterized by an AS = 2. A high AS value means that the gait instance is more distant from the legitimate user's training data. As a consequence, it was more likely produced by an unauthorized user. Normal and abnormal instances are distinguished by a threshold (AS_{th}). The value of the threshold is set as a trade-off between the capability of detecting anomalies and the problem of generating false positives (i.e., legitimate user's instances that are classified as anomalies).

4. Experiments and evaluation procedure

We performed experiments with accelerometers in a pocket and at the wrist. Collected data were used to evaluate the accuracy of the proposed method and make a direct comparison between the two sensor positions.

Table 1: Main characteristics of the group of volunteers involved in data collection (avg.±std.dev.)

Number	Age	Height	Weight
15 males, 5 females	26.3±3.6 years	174.7±8.5 cm	68.6±12.9 kg

4.1. Collection of experimental data

Data collection exploited two Shimmer3 devices, which embed a tri-axial accelerometer [39]. One device was worn like a watch (wrist trace), whereas the other was carried in a front trouser pocket (pocket trace). Twenty volunteers were involved: their main characteristics are shown in Table 1. The experiment consisted in walking a corridor six times: two times at preferred pace, two times at fast pace, and two times keeping the hand inside the trouser pocket. Hence, three different gait styles were collected for each user. At the end of the walking session, each volunteer was also asked to perform random, but periodic, movements with his/her hand (such as drawing an "8" in the air repeatedly). The aim was to produce fake gait detections in the wrist trace.

Acceleration was sampled at 51.2 Hz and saved on the persistent memory of the device. Then, traces were transferred onto a PC where they were used as input for the proposed method. This allowed us to test the behavior of the method when using different values for parameters and thresholds. The walking detection algorithm was used to automatically extract gait segments from these traces.

4.2. Feature selection

Feature selection was performed as indicated in [40], where a preliminary description of the technique was presented[1]. In particular, feature selection relies on the following two steps:

1. inputs are reduced by selecting a subset of the six vectors—this reduction is based on whether features derived from local or global coordinates are used in addition to magnitude $\mathbf{m}$ (magnitude was included in all subsets as autocorrelation features are computed only on that vector);
2. remaining features are further reduced by using Correlation-based Feature Subset Selection with greedy hill climbing search [41].

As shown in Table 2, three different feature sets were selected to evaluate three different approaches, each using only a subset of the acceleration vectors.

POCKET was used to evaluate the authentication method on the pocket acceleration trace. When a device is carried into a pocket, like a smartphone, it would be unreliable to assume a predefined orientation of the device. In fact, orientation is likely to change several times throughout the day depending on the way it is put in the pocket. Hence, the vectors of acceleration samples measured on the local coordinate

[1]The authentication technique is here described in more detail, the results are computed on a larger number of subjects, different anomaly detection methods are considered, the effects produced by the size of the training set are evaluated, the energy consumption of the method is estimated.

Table 2: Evaluated approaches and respective feature sets

Approach	Input	Selected features
POCKET	**m**, **h**, **v**	AC_C1, AC_DP2, kurtosis$_m$, kurtosis$_h$, kurtosis$_v$, MAD$_m$, max$_h$, MCR$_m$, MCR$_v$, mean$_h$, median$_h$, median$_v$, min$_m$, min$_v$, SD$_h$, skewness$_m$, skewness$_h$, skewness$_v$
WRIST_G	**m**, **h**, **v**	AAV$_m$, AC_C1, AC_DP1, AC_DP2, duration, IQR$_m$, kurtosis$_m$, MCR$_h$, MCR$_m$, median$_m$, median$_v$, RMS$_m$, SD$_h$, skewness$_m$, skewness$_h$
WRIST_L	**x**, **y**, **z**, **m**	AAV$_y$, AC_C1, AC_DP2, kurtosis$_x$, max$_x$, max$_z$, MCR$_m$, MCR$_y$, MCR$_z$, mean$_x$, mean$_y$, mean$_z$, median$_x$, median$_z$, median$_m$, min$_x$, RMS$_z$, skewness$_y$, skewness$_m$

system were excluded. The POCKET approach represents a baseline against which the wrist-based solution is compared.

Two variations of the wrist-based approach were considered.

WRIST_G used *global* coordinates on a wrist-worn sensor to enable a direct comparison with POCKET. However, it should be considered that the technique used to estimate the horizontal and vertical vectors relies on the direction of gravity [34]. The orientation of the device with respect to gravity is likely to vary significantly during a gait cycle, due to the swing of the user's arm. As a result, the estimation may be inaccurate and fail to provide useful information.

WRIST_L relies on the three axes of the *local* coordinate system plus magnitude. This approach takes advantage of the fact that a wrist sensor is typically carried with the same orientation on the user's arm. For example, a smartwatch is always worn the same way to be able to read the display properly. Differently from the pocket trace, this consistent orientation throughout use enables the adoption of features calculated with respect to the local coordinate system.

4.3. Evaluation procedure of the anomaly detection technique

The authentication performance of the method was evaluated using two metrics commonly adopted in biometric studies: the *false rejection rate* (FRR) and the *false match rate* (FMR), which describe two different kind of errors that may occur. FRR is the proportion of legitimate user's instances that are discarded (not authorized), whereas FMR is the proportion of instances from unauthorized users that are authorized by the system. For a volunteer v, his/her FRR and FMR can be estimated by using leave-one-instance-out cross-validation. In each cross-validation fold, all v's instances but one are included in the training set – FRR is estimated on the left-out v's instance, whereas FMR is estimated on the other users' instances. Finally, v's FRR and FMR are averaged over the cross-validation iterations.

To estimate FRR and FMR values for each user with the procedure described above, it is required to set a specific threshold AS$_{th}$. As an alternative, authentication accuracy can be evaluated as AS$_{th}$ is varied by using *ROC curve* analysis. In fact, a

ROC curve depicts the trade-off between the true positive rate and the false positive rate of a binary classifier as the discriminating threshold is varied [42]. In our context, the discriminating threshold is AS_{th}, and we define a *positive* as the detection of an unauthorized instance (anomaly), whereas a *negative* occurs when a gait instance is authorized. Following this definition, the true positive rate corresponds to the inverse of FMR (1-FMR), whereas the false positive rate corresponds to the FRR. Hence, the ROC curve plots (1-FMR) against FRR as AS_{th} is varied. The overall performance is then measured by means of two widely-used metrics: the equal error rate *EER* (the point of the curve where FRR = FMR) and the *area under the curve* (AUC).

5. Results and Discussion

In this Section, we first evaluate the results obtained by the combined use of the walking detection and AC filter modules, in terms of detected gait segments at the two considered positions, i.e. wrist and pocket. After, we show and discuss the performance of the proposed user authentication method for each of the approaches in Table 2. We also evaluate the impact of gait segment length and training set size on authentication accuracy. Finally, we compared the proposed anomaly detection technique with two other solutions from the state of the art.

5.1. Results on the detection of gait segments

The performance of the whole gait segment detection process, made of walking detection plus AC filter, is shown by the histogram in Figure 4, in terms of the average time required for recognizing a gait segment when the user is actually walking. On average, a gait segment was detected every 5.1 s in the pocket trace and every 5.5 s in the wrist trace. The worst case in the pocket experiment occurred for user 19 (6.0 s to detect a gait segment), whereas with the wrist-worn device it occurred for user 12 (7.7 s). The total number of detected segments was 689 in the pocket trace, and 643 in the wrist trace. The duration of a gait segment (four cycles) was 4.2 s on average.

5.2. Filtering of irregular gait segments and rejection of non-gait segments

The AC filter discards true gait segments that are characterized by a level of regularity that is unsuitable for authenticating the user. The amount of gait segments discarded by the AC filter is shown in Figure 5, for the two considered positions (wrist and pocket) and for the different users. As expected, the average rate of discarded segments at the wrist was significantly higher than in the pocket experiment (10.5% vs 5.3%). This was mainly due to the arm swing movement during gait, which in some occasions may affect the regularity of the acceleration signal. This aspect was particularly relevant for users 6 and 13, as about 30% of true gait segments were not considered suitable for authentication. It is interesting to notice that users are characterized by a relatively wide range, in terms of gait regularity, with a percentage of discarded gait segments that goes from 0% to 30%.

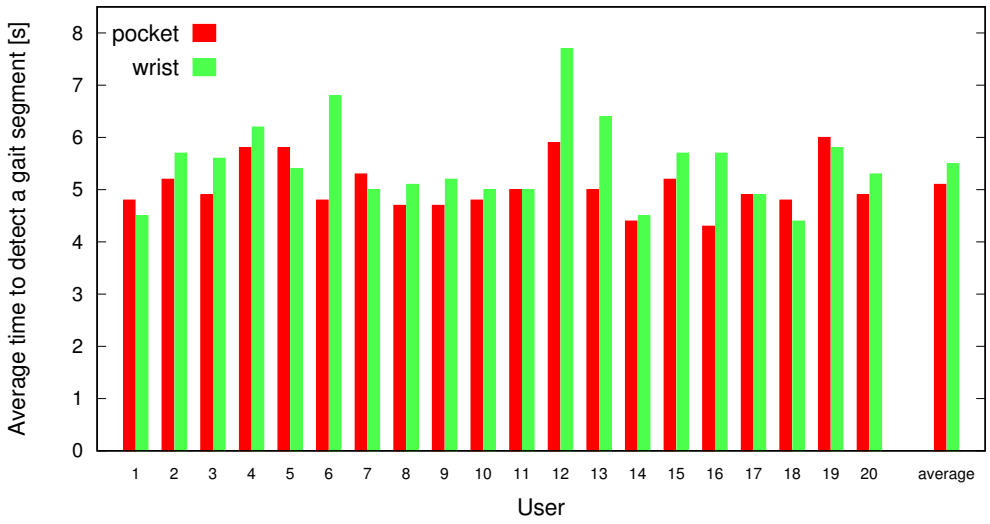


Figure 4: Time to detect a gait segment.

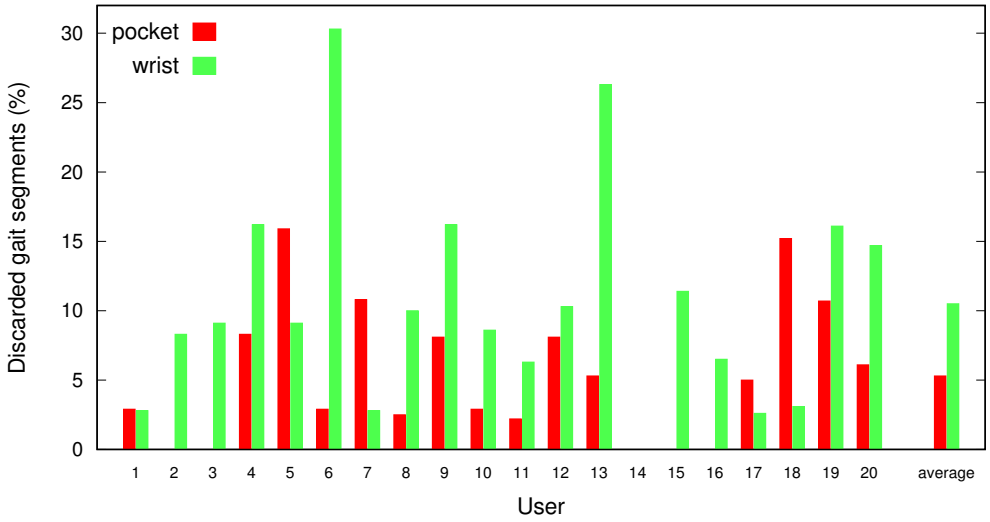


Figure 5: True gait segments discarded by the AC filter.

The AC filter also discards non-gait segments that are regular enough to pass the first simple filter, i.e. the one based on standard deviation [31]. As previously mentioned, volunteers were asked to perform some movements with their hands that could be confused with real gait segments. The AC filter was able to discard 35 out of 36 non-gait segments. The beneficial effect of the AC filter on authentication accuracy is demonstrated in Section 5.4.

It should be noted that this study was not focused on accurate gait detection, and that the detection rate shown in Figure 4 fully meets the requirements of an application to authenticate users as soon as they walk a few steps. From this point of view, the technique for gait segment detection performed fairly well even at the wrist: when the user is walking, an authentication opportunity is provided every ∼8 s in the worst

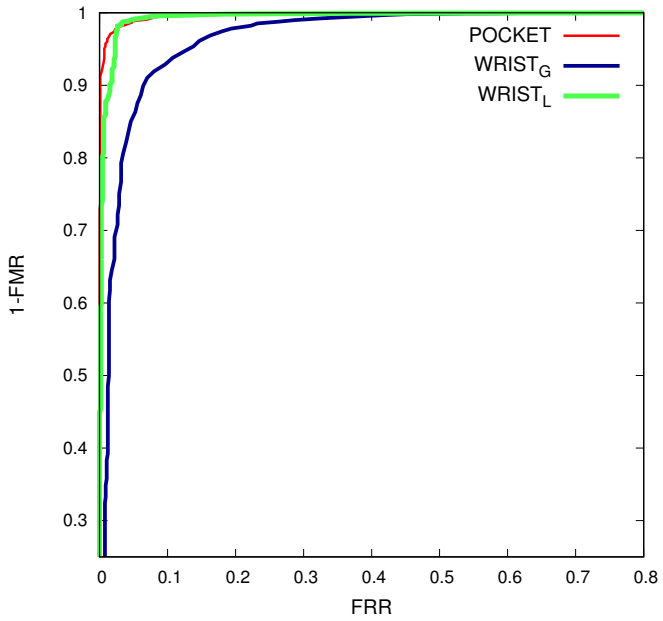


Figure 6: ROC plot for the three approaches.

Table 3: Results of the three approaches

Metric	POCKET	WRIST_G	WRIST_L
AUC (%)	99.7	96.9	99.5
EER (%)	2.3	8.1	2.5

case.

5.3. Results on authentication

We trained and cross-validated 20 gait-based authentication models, each time considering a different user from our dataset as the legitimate user. More precisely, we used leave-one-instance-out cross-validation: at each fold the classifier is trained on all the legitimate user's gait instances but one, which is left out for testing together with other users' instances. Given a number n of legitimate user's instances, the procedure is repeated n times, each time leaving out a different genuine instance for testing. As a result, an anomaly score is obtained for each genuine instance, whereas for unauthorized instances the procedure finds an anomaly score per each fold. At the end of cross-validation the anomaly score of an unauthorized instance is set as the average score for the instance over the n validation folds. From these scores it is possible to obtain a ROC curve describing the performance of gait-based authentication when a specific user from our dataset is selected as the legitimate user. Finally, in order to find metrics describing the overall performance of the gait-based authentication technique, we computed the average ROC curve over the 20 ROC curves (one per user in the dataset), by using the "threshold averaging" technique described in [42].

Table 4: Effect of autocorrelation-based filtering on EER and AUC (percentage point variation)

Metric	POCKET	WRIST_L
AUC variation	+1.0	+4.4
EER variation	−1.6	−7.3

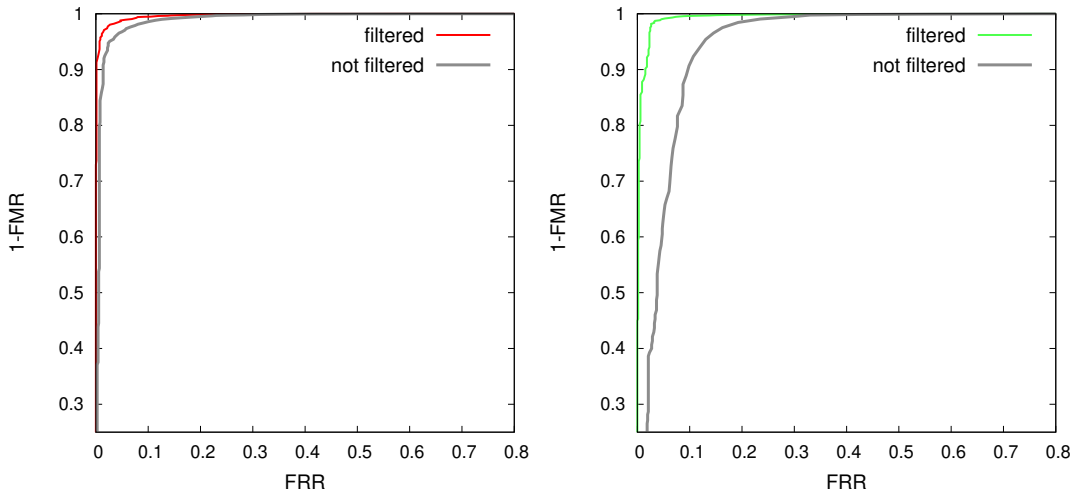
The plot in Figure 6 shows such average ROC curve for each of the three approaches considered, namely POCKET, WRIST_G, and WRIST_L. From the curves, it is clearly visible that the use of estimated global coordinates on a wrist-worn device (WRIST_G) leads to a substantially worse authentication accuracy. The reason is that the lightweight technique adopted to estimate vertical and horizontal directions cannot be applied correctly when the device's orientation with respect to gravity is not approximately constant. Indeed, when a person is moving around, the wrist-worn sensor is constantly rotated due to the typical arm swing and this hinders the estimation of global coordinates. As a result, the features calculated on $\mathbf{h}$ and $\mathbf{v}$ do not provide a sound description of the user's gait pattern and cannot be used for reliable authentication. A possible solution would be the use of more advanced techniques to estimate the orientation of the device in real time, for example including additional sensors like a gyroscope and a magnetometer. However, we decided not to investigate these techniques in this work, as the focus here is on designing a lightweight method that can be executed in real time in miniaturized devices and that ensures long battery life.

In order to achieve better accuracy, it is paramount to exploit the fact that a wrist-worn device is always worn the same way. Differently from a smartphone, the orientation of a smartwatch with respect to the user's arm is not changed throughout use. This enables the adoption of features derived from the acceleration samples measured on the device's local coordinate system. From Figure 6, it is evident that the performance of this approach (WRIST_L, green line) is very close to the one achieved by the pocket-worn device.

The metrics shown in Table 3 further support these considerations. The performance of POCKET and WRIST_L was very close in terms of AUC and EER. In particular, WRIST_L achieved 2.5% EER and 99.5% AUC. This result is in line with the best performing gait-based authentication systems in the literature, which typically exploited a more favorable body position, such as a device attached to the user's waist with a belt. Conversely, the EER of WRIST_G was more than three times higher with respect to the other approaches, with an AUC below 97%.

5.4. AC filter and authentication accuracy

The ROC curves in Figure 7a and 7b highlight the effect of the AC filter on POCKET and WRIST_L: the authentication accuracy of both approaches is substantially improved when irregular patterns are removed, but the effect is much more visible for WRIST_L. Numerical details are given in Table 4. As previously mentioned, at the wrist it is more challenging to provide reliable detection of gait cycles for two main reasons: i) hand movements during gait may lead to merge consecutive steps or split



(a) POCKET approach (b) WRIST_L approach

Figure 7: Effect of autocorrelation-based filtering on the ROC curve.

a single step; ii) spurious hand movements during other activities may lead to false detections. Both issues are addressed effectively by the proposed AC filter. The impact on EER is a reduction by 7.3 percentage points, whereas AUC is increased by 4.4 percentage points.

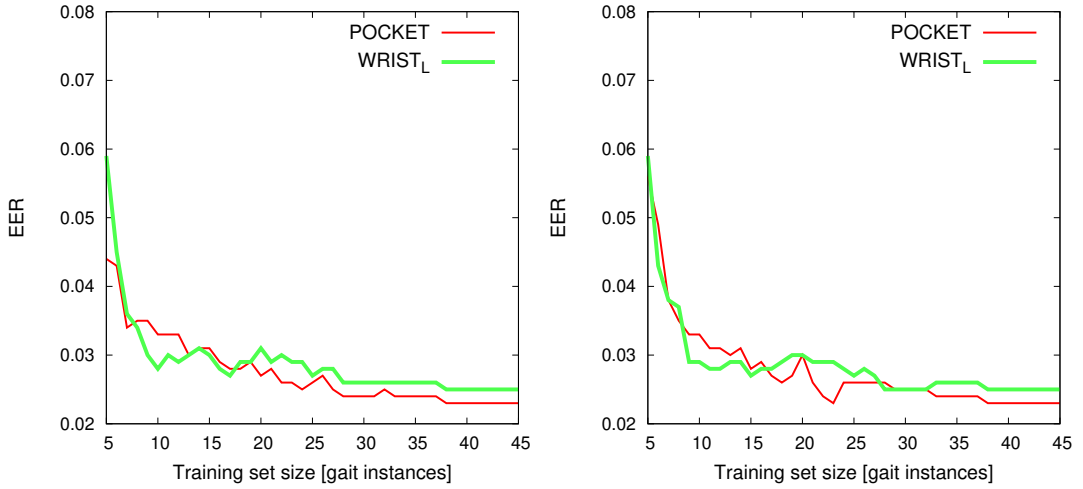
The importance of autocorrelation-based analysis is further confirmed by a test we performed to evaluate the contribution of individual features to authentication accuracy. For POCKET and WRIST_L, we tried to remove one feature at a time from the respective features sets, and evaluated the loss in accuracy. For POCKET, the most valuable feature is AC_C1, whose removal leads to an increase of 0.8 percentage points in the EER. For WRIST, the most valuable feature is AC_T2 (+0.4 percentage points in the EER), immediately followed by AC_C1 (+0.3 percentage points).

5.5. Gait segment length and authentication accuracy

The choice of the number of gait cycles to be included in a gait segment was made considering the trade-off between authentication accuracy and the ability to perform continuous authentication even in relatively small environments, where long walks are uncommon. A minimum of two gait cycles per segment are required to enable the analysis of regularity between cycles through autocorrelation. The use of a greater number of cycles per segment can make gait detection more specific, as spurious hand movements are less likely to be included in detected segments. Also, by using more gait cycles, it is possible to filter out the cycles produced when the user starts/stops walking, which tend to be less regular and not representative of the user's gait pattern. On the other hand, if the required gait segment is too long, the user may not provide frequent "samples" for authentication, especially if he/she spends most of the time in a relatively small environment.

Table 5: Authentication results based on the number of cycles per gait segment

Cycles per segment	POCKET		WRIST_L	
	AUC (%)	EER (%)	AUC (%)	EER (%)
2	99.0	4.3	97.1	7.9
3	99.0	4.3	98.4	6.1
4	99.7	2.3	99.5	2.5
5	99.5	2.8	99.3	2.9
6	99.3	3.4	99.5	3.2



(a) Reduction applied on all training instances

(b) Reduction applied for each new instance after the set size limit has been reached

Figure 8: Effect of training set size on the EER of POCKET and WRIST_L, using two different approaches for reducing the training set.

Table 5 reports the authentication accuracy results achieved using different gait segment lengths. According to these experiments, the use of segments made of four gait cycles is a reasonable solution to the trade-off, as it achieves high authentication accuracy while requiring a relatively short walk (about 6 meters considering an average step length of 0.75 meters). Indeed, using more than four cycles does not seem to improve authentication accuracy, while a reduced number of gait cycles leads to a substantially worse performance, especially for the wrist-based approach.

5.6. Training set size and authentication accuracy

In our experiments the average number of gait instances per user was 32.2 for the wrist-worn device and 34.1 for the pocket-worn device. In real use, the legitimate user is expected to wear the device for a few days, so that enough instances describing his/her typical gait patterns are captured and included in the training set. In the context of a miniaturized device with a relatively small RAM size, the adoption of a training set reduction technique will be necessary to ensure that the set fits into main memory.

Two different configurations can be identified: i) the device is equipped with persistent storage (e.g., the Shimmer has 2 GB of microSD storage); ii) the device can only save gait instances into RAM, even during the training phase. In the first configuration, thanks to the presence of persistent storage, all the gait instances found during the training phase can be stored. Hence, the reduction technique can be applied at the end of the training phase on all the found instances, in order to ensure that the final training set fits into RAM. In contrast, in the second configuration the device can only save a limited number of training instances during the training phase: once the maximum number of instances has been reached, the reduction technique will be used to decide whether each new instance should replace one of the instances already included in the training set.

In both configurations (reduction applied at the end of training or for each new instance), we propose a reduction technique that attempts to eliminate a gait instance belonging to a dense area of the feature space. More precisely, it is eliminated the instance with the smaller distance from its two nearest neighbors in the training set, considering the sum of the distances from the nearest neighbor and the second-nearest neighbour (2-NN sum algorithm). Such instance is likely to carry redundant information, and it is reasonable to expect that its removal has a minor impact on authentication accuracy.

We tested these two configurations on our dataset, considering training set sizes ranging from 5 to 45 (the maximum number of gait instances per user in our experiments was 44 for POCKET and 45 for WRIST). In the first configuration, the reduction technique is applied repeatedly on the entire training set, until the desired training set size is achieved. In the second configuration, given a desired training set size S, the first S instances are added to the training set, then the reduction technique is used for each additional instance to decide whether it should replace another instance in the training set. Results of this experiment are presented in Figure 8a (first configuration), and Figure 8b (second configuration). These plots show the achieved EER in relation to a specific training set size limit.

Some interesting conclusions can be drawn from this experiment. First, the two configurations achieve similar results, and thus it is not strictly necessary to save the entire training set on persistent storage before applying the reduction technique. This potentially enables the use of devices with extremely constrained specifications. Second, the reduction of the training size has a similar impact on the two device positions, POCKET and WRIST_L, in terms of authentication accuracy. Finally, in the considered experimental setting, when using a training set size of at least 10 gait instances the EER is always below 4%, and the loss with respect to using the full training set is ≤ 1 percentage point in terms of EER.

5.7. Comparison with other anomaly detection techniques

As mentioned in Section 3.4, the problem of gait-based authentication belongs to the field of semi-supervised anomaly detection, which requires to perform one-class classification. The classifier is trained only on the normal class (i.e., the legitimate

Table 6: Results of other anomaly detection techniques

Classifier	POCKET		WRIST_L	
	AUC (%)	EER (%)	AUC (%)	EER (%)
OCSVM	99.5	2.5	99.2	3.5
IF	95.4	11.0	95.0	11.7

user's instances). Then, it should be able to identify new instances as genuine or anomalies by assigning an anomaly score (AS). In the following we compare the proposed classifier, which is based on NN analysis, with two popular techniques in the field of one-class classification: One-Class Support Vector Machine (OCSVM) [43], and Isolation Forest (IF) [44].

Table 6 shows the results obtained by OCSVM and IF on our dataset, using the same evaluation procedure. For both classifiers we used the implementation of the Scikit-learn Python library with the default parameter values, the only exception being the ν parameter of OCSVM, which was fine-tuned to improve classification results (the selected value was $\nu = 0.2$). It can be observed that, on our experiment, the NN-based approach and OCVSM achieved excellent results, with NN being slightly superior on both POCKET and WRIST_L experiments. On the other hand, in this specific context IF was clearly outperformed, showing an EER above 10%.

These results support our selection of a classifier based on NN analysis. Moreover, it should be highlighted that NN-based classification offers a key advantage over other solutions for on-node implementation, which is among the main aims of this study. Indeed, the NN-based classifier does not require to explicitly build a model at the end of the training phase, as the model is represented by the training instances. This means that even the training phase can be performed on a device with extremely constrained resources. In contrast, training an OCSVM can be costly in terms of both processing power and memory requirements [45], while IF has relatively greater memory requirements for the trained model, as it is composed of an ensemble of binary decision trees (100 decision trees when using the default configuration).

5.8. Limitations

Our work is based on an experimental setting that is very common in this field: supervised walking performed by volunteers in a controlled environment. The presented results show that our method achieves EER results in line with (or better than) most of the studies from the state of the art that used a similar experimental setting (for example [25] relying on 500 s of walk for each user, or [29] where data collection occurred under the supervision of a researcher). The key contribution is that we proved that a lightweight approach for continuous authentication is possible, at least in this experimental context. Also, we showed that a wrist-based solution does not lead to worsened results compared to a device placed in a pocket, opening up the way to a broader use of smartwatches for gait-based authentication or even smart-wristbands, which are generally characterized by limited computing power.

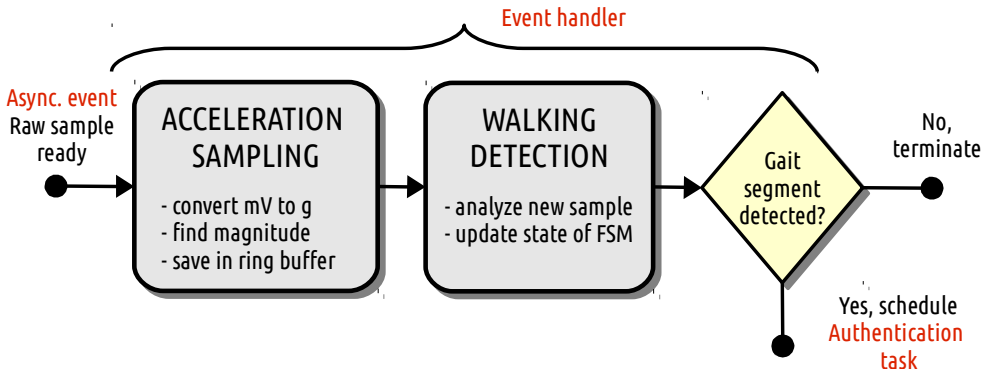


Figure 9: Event-driven implementation.

The evaluation of temporal robustness would require repeating the experiments on a different day under different conditions, or performing long-term experiments. Such long-term experiments are beyond the scope of this work. Indeed, as far as we know a very limited number of recent works attempted to apply gait-based authentication to real world data (for example [46], where 6 users wore a smartwatch 2 hours per day for 3 days obtaining 95.2% of true positive rate, or [47] where the data collected on two different days was used to understand the impact of such a factor). Long-term traces would also be useful for estimating the duration of the training period, in order to fully learn the user's gait pattern. Concerning this problem, the analysis presented in Section 5.6 allows a better understanding of the impact of a limited training size on the performance of the method (as a long-term collection could lead to an excessive growth of the user's model). It can thus be considered as an intermediate step in the direction of long-term experiments.

6. Real-time execution and power consumption

On-node processing is key for wearable-based applications aimed at the continuous monitoring of users' activities. The reason is twofold: i) the device can always offer the required service, even when not in the range of an external device; ii) on-node processing reduces the amount of radio transmissions required and ensures a significantly longer battery duration. This, in turn, improves usability and fosters user acceptance.

In this section, we first provide details on how the proposed method can be implemented and executed on devices with limited resources. This proves that the method is sufficiently *lightweight*, in terms of both computational and memory requirements, to be executed in real time in miniaturized devices. After, we also demonstrate the energy saving deriving from the on-node processing strategy.

6.1. Implementation as an event-driven TinyOS application

A prototype of the proposed method was implemented for *Shimmer2r* devices, which are equipped with a microcontroller unit (MCU) running at 8 MHz and with 10 KB of RAM. Battery capacity is 450 mAh. The prototype was developed using

TinyOS version 2.1.2 and the nesC programming language [48]. TinyOS applications are event-driven and characterized by three computational abstractions: *commands*, *events* and *tasks*. Commands are used to require a specific service. Events are signaled on completion of a service, triggering the execution of the corresponding handler. The event handler must be responsive – extensive computations, if required, are deferred by scheduling the later execution of a task. Tasks are executed in background according to a FIFO policy, when there are no other handlers currently running.

The architecture presented in Figure 1 can be easily adapted to the event-driven paradigm. The boot procedure enables the accelerometer and the ADC, and requests acceleration sampling with 51.2 Hz frequency. After, the behavior of the application is described by an asynchronous event, an event handler and a task, as shown in Figure 9. The event signals that a new sample is ready to be processed.

The event handler implements the acceleration sampling and walking detection modules of our method. Raw acceleration components (x, y, z) are converted into g units, then the acceleration magnitude (m) is computed. The three acceleration components and magnitude are added to a ring buffer by overwriting the oldest acceleration sample. The ring buffer can store up to 512 values for each component, representing the last 10 s of acceleration data. Acceleration values are represented as 16-bit integers – with respect to floats, this cuts RAM occupation by 50% and also leads to a dramatic reduction in terms of computational cost (the MCU does not embed a floating point unit). Walking detection is implemented as a finite state machine. Each new sample is used as input to the machine, so as to update its inner state. If a new gait segment is not found, the event handler terminates and the device becomes idle until the sampling timer fires again. Conversely, when a gait segment is detected, the handler schedules the execution of the authentication task, which includes AC filter, feature extraction, and anomaly detection.

The application requires ∼7 KB of RAM without considering the training set. The remaining RAM can be used to load a training set with up to 75 gait instances, each being a gait pattern of the legitimate user.

6.2. Real-time processing

The application must process a gait segment before the respective part of the buffer gets overwritten. To this end, two requirements must be met. Given a gait segment g, let d_g be its duration and let p_g be the time required to process g for gait-based authentication. The first requirement is that $p_g < d_g$. This requirement guarantees that the application is able to keep up with the flow of acceleration samples without backlog, even in presence of back-to-back gait segments. The second requirement is that $p_g < bufCapacity - d_{max}$, where $bufCapacity$ is the interval of acceleration data that can be held in the ring buffer, and d_{max} is the longest gait segment accepted by the walking detection algorithm.

In our implementation the buffer can store up to 10 s of samples ($bufCapacity$). The duration of gait segments in the dataset ranged from 3.4 to 5.6 s, with a mean of 4.2 s. With a conservative approach, the walking detection algorithm was tuned to

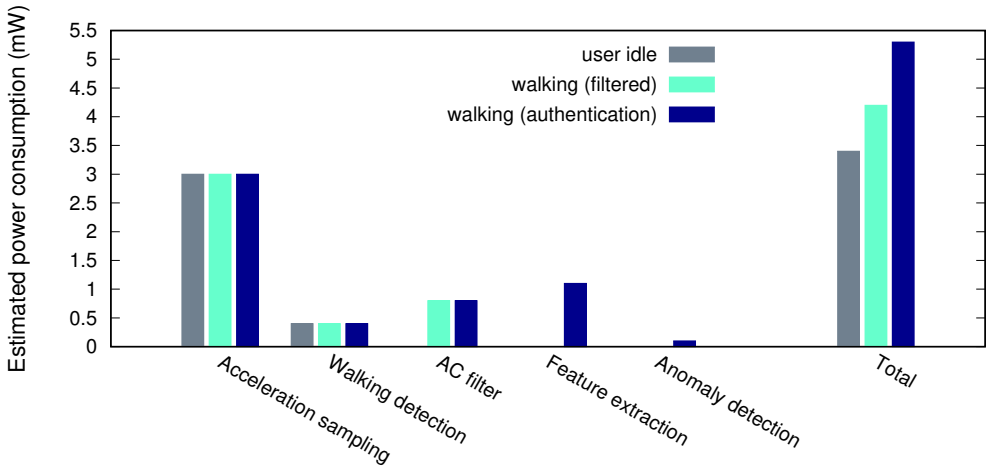


Figure 10: On-node processing strategy – estimated power consumption based on user's activity.

Table 7: Subtasks and average power consumption

Subtask	Average power consumption
Acceleration sampling	3.0 mW
Radio streaming	2.9 mW
Walking detection	0.4 mW
AC filter	0.8 mW
Feature extraction	1.1 mW
Anomaly detection	0.1 mW

detect segments lasting up to 7 s (d_{max}). Therefore, according to the second requirement, the time to process a segment (p_g) must be shorter than 3 s. The atomic code of the event handler requires < 1 ms to read and process an acceleration sample. Hence, the reminder of the sampling period ($\sim$ 19 ms) is used to execute the authentication task, when active. Under these conditions, the MCU can process a 7 s segment in less than 1 s and meets both the requirements mentioned above. This result proves that the proposed method can be executed in real time by a Shimmer device.

6.3. Power consumption and battery duration estimation

We estimated power consumption considering two opposite strategies: *on-node processing*, where all the required operations are executed in the wearable device, and *external processing*, where raw acceleration samples are sent to an external companion device (such as a smartphone). The analysis started by estimating the average consumption associated with each of the different subtasks that compose our method, as well as the power required to stream samples continuously to the companion device. The results are summarized in Table 7.

Acceleration sampling includes sampling the ADC at 51.2 Hz, converting acceleration components and saving them into the ring buffer. This requires an average consumption of $\sim$ 3.0 mW – this power requirement is a baseline that cannot be avoided, whatever the strategy adopted. *Radio streaming* was estimated as follows.

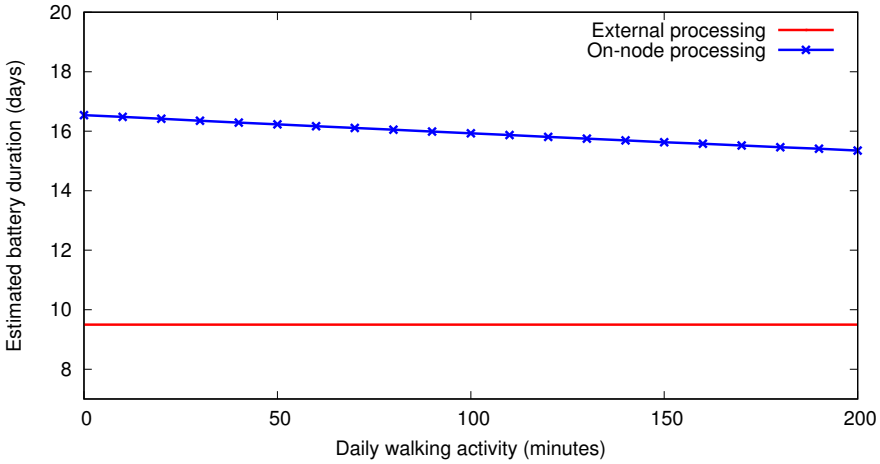


Figure 11: Estimated battery duration for the two strategies.

First, we implemented a simple protocol to transmit reliably a five-second window of samples every 5 s. Then, we measured the time required to transmit each window, so as to find the duty cycle of the radio module. Finally, we combined the duty cycle information with the transmission power consumption indicated in the datasheet of the Shimmer's radio module (CC2420). As a result, we estimated an average power consumption of 2.9 mW to send samples to a base station. The strategy that relies on an external device always shows the same power consumption, regardless of the user's activity, given by the sum of acceleration sampling and radio streaming ($\sim$ 5.9 mW).

Walking detection requires $\sim$ 0.5 ms of processing for each new acceleration sample, corresponding to an average consumption of $\sim$ 0.4 mW. In the on-node strategy, walking detection is performed continuously.

The power consumption of the remaining subtasks of the on-node strategy depends on the current activity performed by the user. This is better explained in Figure 10, which shows the estimated consumption associated with each subtask in three different scenarios: i) the user is idle (i.e., not walking); ii) the user may be walking and a gait segment is found, but the segment is discarded by the AC filter; iii) the user is walking and the gait segment is used for authentication. When the user is idle (gray bars), none of the gait authentication subtasks is executed, as the application only performs acceleration sampling and walking detection. Whenever the user is walking and a gait segment is found, the AC filter subtask is executed, with the power consumption indicated in Table 7. This value represents the average power required for the duration of the gait segment. If the gait segment is discarded by the filter (light blue bars in Figure 10), then feature extraction and anomaly detection are not necessary. Conversely, if the gait segment is used for authentication (blue bars), then all the subtasks are executed. During authentication the power consumption of the on-node strategy is $\sim$5.3 mW.

Finally, Figure 11 shows the estimated battery duration for the two strategies, as a function of the user's daily walking activity. The strategy that relies on an external

device has an expected battery duration of $\sim$ 9.5 days regardless of the user's activity level. Even considering a scenario where the user has been extremely active (e.g., walking 200 minutes a day), on-node processing shows a clearly superior performance. In fact, the estimated battery duration is more than 60% longer when authentication is executed in the wearable device. This result confirms the importance of developing a lightweight technique that can be implemented and executed in real time in miniaturized devices.

7. Conclusions

Smartwatches and other wrist-worn devices are used by an ever-increasing number of people. In this paper, an authentication method based on gait analysis at the wrist has been presented and evaluated. The method relies on the acceleration signal and uses anomaly detection to understand if the current user is the legitimate one or someone else. The experimental evaluation shows that the method is reliable and that authentication can be executed with an EER of 2.5%.

Previous research on gait-based authentication mostly focused on techniques tailored for being executed on smartphones. A direct transposition of such techniques to wrist-worn devices is destined to be sub-optimal, as demonstrated in Section 5. In fact, smartphones are generally carried in a user's pocket and are thus positioned near the center of mass. This makes smartphones less susceptible to spurious movements. On the contrary, wrist-worn devices are subject to frequent accelerations produced by the use of hands for everyday tasks and gesticulation. These accelerations are not related to gait and must be filtered out by the authentication method, making the process more challenging.

Another aspect that differentiates smartphones from wrist-worn devices is their orientation. Smartphones can be oriented in different ways with respect to the user's body, for example depending on the way they have been inserted in a pocket. Conversely, wrist-worn devices have a fixed orientation with respect to the user's arm, and this is an advantage as it does not require to compute features on a global reference system.

The presented method relies on a single accelerometer to reduce implementation complexity and cost. In future work, we will evaluate the adoption of additional sensors, such as gyroscope or magnetometer, from which additional information can only be beneficial in terms of authentication accuracy. However, it should be considered that the reduced complexity of the solution proposed in this work enables its implementation and real-time execution in microcontroller-based devices, characterized by limited computational capacity and storage. Furthermore, the evaluation of the energy usage by the method shows that executing authentication on the wrist-worn device is paramount to ensure that continuous use in days is possible on battery operated devices. More specifically, the on-node processing strategy improves battery usage by more than 60% with respect to streaming information to a more powerful device (e.g., a smartphone).

To enable direct comparison with the proposed method and foster further advancements in the field, the acceleration traces will be made publicly available at the following address upon publication:
http://vecchio.iet.unipi.it/gaitanalysis.

Acknowledgments

This research was supported by the PRA 2018_81 project "Wearable sensor systems: personalized analysis and data security in healthcare", funded by the University of Pisa, and by the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence).

References

[1] S. Zhang, P. McCullagh, H. Zheng, C. Nugent, Situation awareness inferred from posture transition and location: Derived from smartphone and smart home sensors, IEEE Trans. on Human-Machine Systems 47 (6) (2017) 814–821.

[2] M. Ehatisham-ul Haq, M. Awais Azam, U. Naeem, Y. Amin, J. Loo, Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing, Journal of Network and Computer Applications 109 (February) (2018) 24–35.

[3] K. Halunen, J. Häikiö, V. Vallivaara, Evaluation of user authentication methods in the gadget-free world, Pervasive and Mobile Computing 40 (2017) 220–241.

[4] L. Bianchi, D. Angelini, F. Lacquaniti, Individual characteristics of human walking mechanics, Pflügers Archiv 436 (3) (1998) 343–356.

[5] S. Patel, H. Park, P. Bonato, L. Chan, M. Rodgers, A review of wearable sensors and systems with application in rehabilitation, Journal of NeuroEngineering and Rehabilitation 9 (1).

[6] J. Kirby, C. Tibbins, C. Callens, B. Lang, M. Thorogood, W. Tigbe, W. Robertson, Young people's views on accelerometer use in physical activity research: Findings from a user involvement investigation, ISRN Obesity 2012.

[7] N. Alshurafa, J.-A. Eastwood, M. Pourhomayoun, S. Nyamathi, L. Bao, B. Mortazavi, M. Sarrafzadeh, Anti-cheating: Detecting self-inflicted and impersonator cheaters for remote health monitoring systems with wearable sensors, in: Proc. IEEE Int. Conf. on Wearable and Implantable Body Sensor Networks (BSN), 2014, pp. 92–97.

[8] S. Mauceri, L. Smith, J. Sweeney, J. McDermott, Subject recognition using wrist-worn triaxial accelerometer data, in: Machine Learning, Optimization, and Big Data, Springer International Publishing, 2018, pp. 574–585.

[9] J. Kwapisz, G. Weiss, S. Moore, Cell phone-based biometric identification, in: Proc. IEEE Int. Conf. on Biometrics: Theory Applications and Systems (BTAS), 2010, pp. 1–7.

[10] G. Cola, M. Avvenuti, A. Vecchio, Real-Time Identification Using Gait Pattern Analysis on a Standalone Wearable Accelerometer, The Computer Journal 60 (8) (2017) 1173–1186.

[11] A. Vecchio, G. Cola, Method based on uwb for user identification during gait periods, Healthcare Technology Letters 6 (5) (2019) 121–125.

[12] J. Unar, W. C. Seng, A. Abbasi, A review of biometric technology along with trends and prospects, Pattern Recognition 47 (8) (2014) 2673 – 2688.

[13] S. Sprager, M. Juric, Inertial sensor-based gait recognition: A review, Sensors 15 (9) (2015) 22089–22127.

[14] H. J. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, S.-M. Makela, Identifying people from gait pattern with accelerometers, in: Proceedings of SPIE, Vol. 5779, 2005, pp. 7–14.

[15] D. Gafurov, E. Snekkenes, P. Bours, Gait authentication and identification using wearable accelerometer sensor, in: Proc. IEEE Workshop on Automatic Identification Advanced Technologies, 2007, pp. 220–225.

[16] D. Gafurov, E. Snekkenes, P. Bours, Improved gait recognition performance using cycle matching, in: Proc. IEEE Int. Conf. on Advanced Information Networking and Applications Workshops (WAINA), 2010, pp. 836–841.

[17] L. Rong, Z. Jianzhong, L. Ming, H. Xiangfeng, A wearable acceleration sensor system for gait recognition, in: Proc. IEEE Conf. on Industrial Electronics and Applications, 2007, pp. 2654–2659.

[18] M. Derawi, P. Bours, K. Holien, Improved cycle detection for accelerometer based gait authentication, in: Proc. Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010, pp. 312–317.

[19] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, Y. Yagi, The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication, Pattern Recognition 47 (1) (2014) 228 – 237.

[20] H. Lu, J. Huang, T. Saha, L. Nachman, Unobtrusive gait verification for mobile phones, in: Proc. ACM Int. Symposium on Wearable Computers, 2014, pp. 91–98.

[21] A. Primo, V. V. Phoha, R. Kumar, A. Serwadda, Context-aware active authentication using smartphone accelerometer measurements, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, 2014, pp. 98–105.

[22] G. Pan, Y. Zhang, Z. Wu, Accelerometer-based gait recognition via voting by signature points, Electronics Letters 45 (22) (2009) 1116–1118.

[23] A. H. Johnston, G. M. Weiss, Smartwatch-based biometric gait recognition, in: Proc. IEEE Int. Conf. on Biometrics Theory, Applications and Systems (BTAS), 2015, pp. 1–6.

[24] Y. Ren, Y. Chen, M. C. Chuah, J. Yang, User verification leveraging gait recognition for smartphone enabled mobile healthcare systems, IEEE Trans. on Mobile Computing 14 (9) (2015) 1961–1974.

[25] P. Musale, D. Baek, N. Werellagama, S. S. Woo, B. J. Choi, You walk, we authenticate: Lightweight seamless authentication based on gait in wearable iot systems, IEEE Access 7 (2019) 37883–37895.

[26] Y. Shen, C. Luo, W. Xu, W. Hu, Poster: An online approach for gait recognition on smart glasses, in: Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys '15), 2015, pp. 389–390.

[27] L. Brombin, M. Gambini, P. Gronchi, R. Magherini, L. Nannini, A. Pochiero, A. Sieni, A. Vecchio, User's authentication using information collected by smartshoes, in: L. Mucchi, M. Hämäläinen, S. Jayousi, S. Morosi (Eds.), Body Area Networks: Smart IoT and Big Data for Intelligent Health Management, Springer International Publishing, Cham, 2019, pp. 266–277.

[28] B. M. Galloway, G. Niezen, G. P. Hancke, B. J. Silva, Multimodal biometric authentication in wireless sensor networks, in: Proc. IEEE Int. Conf. on Industrial Informatics (INDIN), 2016, pp. 1003–1007.

[29] G. M. Weiss, K. Yoneda, T. Hayajneh, Smartphone and smartwatch-based biometrics using activities of daily living, IEEE Access 7 (2019) 133190–133202.

[30] A. Acar, H. Aksu, A. S. Uluagac, K. Akkaya, A usable and robust continuous authentication framework using wearables, IEEE Transactions on Mobile Computing 20 (6) (2021) 2140–2153.

[31] G. Cola, A. Vecchio, M. Avvenuti, Improving the performance of fall detection systems through walk recognition, Journal of Ambient Intelligence and Humanized Computing 5 (6) (2014) 843–855.

[32] R. Moe-Nilssen, J. L. Helbostad, Estimation of gait cycle characteristics by trunk accelerometry, Journal of Biomechanics 37 (1) (2004) 121 – 126.

[33] N. Herssens, E. Verbecque, A. Hallemans, L. Vereeck, V. Van Rompaey, W. Saeys, Do spatiotemporal parameters and gait variability differ across the lifespan of healthy adults? a systematic review, Gait & posture 64 (2018) 181–190.

[34] D. Mizell, Using gravity to estimate accelerometer orientation, in: Proc. IEEE Int. Symposium on Wearable Computing, 2003, pp. 252–253.

[35] G. Cola, M. Avvenuti, A. Vecchio, G.-Z. Yang, B. Lo, An unsupervised approach for gait-based authentication, in: Proc. IEEE Int. Conf. on Wearable and Implantable Body Sensor Networks (BSN), 2015, pp. 1–6.

[36] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM computing surveys (CSUR) 41 (3) (2009) 15.

[37] E. M. Knorr, R. T. Ng, V. Tucakov, Distance-based outliers: algorithms and applications, The VLDB Journal 8 (3) (2000) 237–253.

[38] A. Zimek, P. Filzmoser, There and back again: Outlier detection between statistical reasoning and data mining algorithms, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (6) (2018) e1280.

[39] Shimmer, http://www.shimmersensing.com (2021).

[40] G. Cola, M. Avvenuti, F. Musso, A. Vecchio, Gait-based authentication using a wrist-worn device, in: Proc. ACM Int. Conf. on Mobile and Ubiquitous Systems (MOBIQUITOUS 2016), 2016, pp. 208–217.

[41] M. A. Hall, G. Holmes, Benchmarking attribute selection techniques for discrete class data mining, IEEE Trans. on Knowledge and Data Engineering 15 (6) (2003) 1437–1447.

[42] T. Fawcett, An introduction to ROC analysis, Pattern recognition letters 27 (8) (2006) 861–874.

[43] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM transactions on intelligent systems and technology (TIST) 2 (3) (2011) 1–27.

[44] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Trans. Knowl. Discov. Data 6 (1).

[45] I. W. Tsang, J. T. Kwok, P.-M. Cheung, N. Cristianini, Core vector machines: Fast svm training on very large data sets., Journal of Machine Learning Research 6 (4).

[46] W. Xu, Y. Shen, C. Luo, J. Li, W. Li, A. Y. Zomaya, Gait-watch: A gait-based context-aware authentication system for smart watch via sparse coding, Ad Hoc Networks 107 (2020) 102218.

[47] N. Al-Naffakh, N. Clarke, F. Li, P. Haskell-Dowland, Unobtrusive gait recognition using smartwatches, in: 2017 International Conference of the Biometrics Special Interest Group (BIOSIG), 2017, pp. 1–5.

[48] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, et al., Tinyos: An operating system for sensor networks, in: Ambient intelligence, Springer, 2005, pp. 115–148.