Gait-Watch: A Context-aware Authentication System for Smart Watch Based on Gait Recognition

Weitao Xu
School of Information Technology and Electrical Engineering
University of Queensland
CSIRO Data61,Australia
w.xu3@uq.edu.au

Yiran Shen*
College of Computer Science and Technology
Harbin Engineering University
China
shenyiran@hrbeu.edu.cn

Yongtuo Zhang
School of Computer Science and Engineering
University of New South Wales
Australia
ytzhang@cse.unsw.edu.au

Neil Bergmann
School of Information Technology and Electrical Engineering
University of Queensland
Australia
n.bergmann@itee.uq.edu.au

Wen Hu
School of Computer Science and Engineering
University of New South Wales
Australia
wenh@cse.unsw.edu.au

ABSTRACT

With recent advances in mobile computing and sensing technology, smart wearable devices have pervaded our everyday lives. The security of these wearable devices is becoming a hot research topic because they store various private information. Existing approaches either only rely on a secret PIN number or require an explicit user authentication process. In this paper, we present Gait-watch, a context-aware authentication system for smart watch based on gait recognition. We address the problem of recognizing the user under various walking activities (e.g., walking normally, walking with calling the phone), and propose a *sparse fusion* method to improve recognition accuracy. Extensive evaluations show that Gait-watch improves recognition accuracy by up to 20% by leveraging the activity information, and the proposed *sparse fusion* method is 10% better than several state-of-the-art gait recognition methods. We also report a user study to demonstrate that Gait-watch can accurately authenticate the user in real world scenarios and require low system cost.

CCS CONCEPTS

•Human-centered computing →Ubiquitous and mobile computing design and evaluation methods; •Security and privacy →*Authentication;*

KEYWORDS

Gait recognition, wearable device, accelerometer, sparse representation

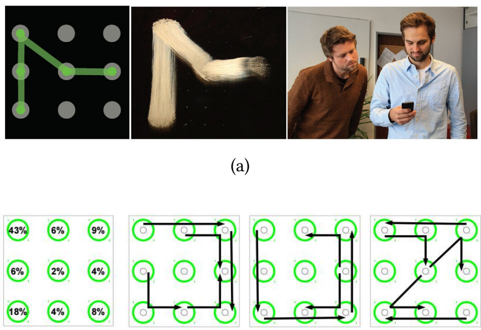
*Corresponding author

ACM Reference format:
Weitao Xu, Yiran Shen, Yongtuo Zhang, Neil Bergmann, and Wen Hu. 2017. Gait-Watch: A Context-aware Authentication System for Smart Watch Based on Gait Recognition. In *Proceedings of The 2nd ACM/IEEE International Conference on Internet-of-Things Design and Implementation, Pittsburgh, PA USA, April 2017 (IoTDI 2017),* 12 pages.
DOI: http://dx.doi.org/10.1145/3054977.3054991

1 INTRODUCTION

With recent advances in embedded computing technologies, smart wearable devices such as Apple Watch and Fitbit wristband, have become increasingly popular and play significant roles in our daily lives. With the pervasiveness of these devices, security is becoming crucial over time as they accumulate a large variety of sensitive data about the users. In particular, with sensors embedded in smart watches, the collected sensory data can be explored for the understanding of user's physical and mental health states. For instance, the accelerometer information collected by the smart watch can be mined to uncover user's daily life activities [24].

Traditional authentication methods such as passwords do not offer good user experience because of the need for keyboard. A mostly common deployed method in current smart watches is the so called Unlock Pattern scheme as in Fig 1(a). In this scheme, the user is presented a 3×3 grid and the secret (password) of a user is a drawing on that grid (i.e., a sequence of lines connecting the dots). During enrollment, a user has to choose a pattern and during the authentication phase, he has to recall his pattern and draw it on the screen. However, the pattern-based unlocking scheme has three major weaknesses. First, they are susceptible to smudge attacks, where imposters extract sensitive information from recent user input by using the smudges left by fingers on touch screens. Recent studies have shown that finger smudges (i.e., oily residues) of a legitimate user left on touch screens can be used to infer pattern [3]. In addition, they are susceptible to shoulder surfing attacks. Smart watches are often used in public settings (such as subway stations, schools, and cafeterias) where shoulder surfing often happens either purposely or inadvertently, and patterns are easy to spy [27]. Third, despite of numerous possibilities in pattern selection, researchers have found that there is a high bias in the pattern selection process, e.g., people often pick the top left corner as a starting point and prefer straight lines in their pattern. The results in [32] indicate that the security offered by this scheme is less than the security of only three digit randomly-assigned PINs for guessing.



(a)

(b)

Figure 1: Android pattern lock authentication system [32]

Motivated by the above issues, we aim to develop an unobtrusive, continuous, and implicit authentication system for smart watch based on gait recognition. Biometric gait recognition refers to verifying or identifying persons by their walking style. Extensive studies from psychology and biometrics have demonstrated that biometric gait contains distinctive patterns that can be used for security purposes [2, 20, 26]. Gait based authentication offers several advantages over traditional authentication system. For instance, it is non-intrusive, does not require explicit authentication process, and provides continuous authentication while walking. On the other hand, gait, as a biometric trait, is hard to be forged and replicated. An imposter can observe how the genuine user walks but still have difficulty in replicating the walking patterns.

Although gait-based recognition has been well explored in the literature, there remains several challenges on smart watch. Firstly, due to the high freedom of arm, people could walk in a variety of ways (e.g., walk normally or walk with calling the phone). The large majority of existing studies on accelerometer-based gait recognition have used a very restrictive experimental setup where the performance evaluation was conducted on a dataset collected from a controlled laboratory environment and the participants are asked to walk normally. As the pervasiveness of smart watches in the wild, these is a need for robust and efficient authentication system in a realistic environment. Besides, the smart watches have limited energy and moderate computing power, thus to improve energy efficiency and reduce computational cost is a crucial task for authentication system on resource-constrained smart watches.

In this paper, we tackle the challenges above and propose a context aware gait-based authentication system on smart watches. Specifically, we implement an activity detector in the system to deal with different activities, and then identification is performed on corresponding training dictionaries according to the output of activity detector. In addition, we employ several recently developed algorithms in Sparse Representation based Classification (SRC) to improve computation efficiency and propose a *sparse fusion* method to improve recognition accuracy. To the best of our knowledge, this is the first work for gait-based authentication using commecial smart watches. The main contributions of this paper are threefold:

- We propose a context-aware authentication system in which the authentication system infers and leverage activity information during authentication. Evaluation results show that compared to the traditional methods which do not take the activity into consideration, the improvement of recognition accuracy can be up to 20%.
- We propose a novel *sparse fusion* method to combine information from several gait cycles together to improve recognition accuracy. Evaluation results show that the proposed method outperforms several state-of-the-art classification methods and improves recognition accuracy by up to 10%.
- We implement Gait-watch on Samsung smart watch, and conduct a user study to evaluate the performance in real world environments. The results show that Gait-watch can authenticate the genuine user with 95.2% true positive rate. In addition, we report the system overhead to demonstrate the feasibility of Gait-watch for contemporary smart watches.

The rest of this paper is organized as follows. In Section 2, we introduce technical background on sparse fusion. We provide an overview of Gait-watch in Section 3 and detail the system architecture in Section 4. In Section 5, we evaluate the performance of Gait-watch on datasets. We then implement the system on smart watches and conduct user study to evaluate the system in Section 6. Finally, Section 7 discusses the related work and Section 8 concludes the paper.

2 BACKGROUND ON SPARSE FUSION

In this section, we introduce the SRC [36] algorithms, as the proposed *sparse fusion* model is based on SRC.

The SRC method solves a single-label classification problem, which aims to return the class that best matches a given test sample. The method is featureless in the sense that the number of elements in a training example is of the same order of magnitude as the amount of raw data describing the example. Note that because training examples and test samples are vectors, we will also refer to them as training vectors and test vectors. A key idea behind SRC is to assemble all the training vectors from all classes into a *dictionary* matrix A, and SRC is applied to solve the traditional linear equation:

$$y = Ax \tag{1}$$

where $y \in \mathbb{R}^q$ is the test vector; $A \in \mathbb{R}^{q \times (N \cdot K)}$ is the dictionary consisting of K classes and each class contains N q-dimensional training vectors. With the knowledge of A and y, ℓ_1 optimization can be applied to solve the linear equation with the *sparse assumption*:

$$\hat{x} = \arg\min_x \|x\|_1 \quad \text{subject to } \|y - Ax\|_2 < \epsilon \tag{2}$$

where ϵ is used to account for noise and the sparse assumption holds when the test vector can be represented by one of the classes in A. Due to the large dimensionality of the training vectors, solving Eq. (2) can be computationally intensive. Inspired by the recent information theory technique of Compressive Sensing (CS) [6], a random projection matrix $R \in \mathbb{R}^{p \times q}$ ($p \ll q$) can be applied to improve the efficiency of the ℓ_1 optimization. In particular, the projection matrices are randomly generated from Bernoulli or Gaussian distributions because of their information preserving properties:

$$\hat{x} = \arg\min_x \|x\|_1 \quad \text{subject to } \|Ry - RAx\|_2 < \epsilon \tag{3}$$

After obtaining the sparse representation vector $\hat{x} \in \mathbb{R}^{N \cdot K}$, the class results can be determined by checking the residuals based on the Euclidean distance. The definition of the residual for class i is:

$$r_i(y) = \|y - A\delta_i(\hat{x})\|_2 \tag{4}$$

where $\delta_i(\hat{x}) \in \mathbb{R}^{N \cdot K}$ contains the coefficients related to class i only (the coefficients related to other classes are set to be zeros). Then the final result of the classification will be:

$$\hat{i} = \arg\min_{i=1,\ldots,K} r_i(y) \tag{5}$$

i.e., the right class produces the minimal residual.

To further improve the performance of SRC, [29] proposes a heuristic algorithm to find the optimal projection matrix instead of the random one. The classification accuracy is improved by up to 12% with the optimized projection matrix. They also improve the efficiency of SRC by casting the residual calculation to a significantly lower dimensionality by introducing the *compressed residual*:

$$r_i(y) = \|R_{opt}y - R_{opt}A\delta_i(\hat{x})\|_2 \tag{6}$$

where $R_{opt} \in \mathbb{R}^{p \times q}$ is the optimized projection matrix. The classification accuracy will be preserved at the significantly lower dimensionality as described in [29].

3 SYSTEM OVERVIEW

In this section, we will introduce the proposed system by walking through an example scenario and then describe the architecture in detail.

Example Scenario. One day morning, a user gets up and wears his smartwatch on his wrist. After he walks several steps, the smart watch recognizes the identity of the user and provides the user with personal services such as weather information, meeting notification. However, we are aware that gait-based authentication system cannot provide a absolutely secure way to protect the data in smart watch. We imagine in the future to have many levels of security (e.g., two factor authentication) that tradeoff usability and accessibility given the risk imposed. For instance, accessing a user's bank account clearly requires higher security than retrieving the time of the next meeting and occurs less frequently, hence requiring additional authentication scheme (e.g., PIN) after Gait-watch grants user the access to use the smart watch.

As shown in Fig 2, the flow chart of Gait-watch consists of the following components.

Offline Training. During the offline dictionary training phase, gait cycles are first segmented from time series accelerometer signal and then interpolated into the same length. All detected cycles are passed to unusual cycles deletion to remove outliers of gait cycles. The obtained gait cycles are passed to column reduction algorithm [35] to form the training dictionary A. After obtaining A, we apply the projection optimization algorithm [29] to obtain a optimized projection matrix R_{opt}. Then the reduced training dictionary $\tilde{A} = R_{opt}A$ is used in the classifier as described in Section 2.

Online Processing.

Walking Detector To save energy consumption and extend battery life, Gait-watch is only activated when the user is walking. Therefore, we first apply a walking detector which indicates the subject is walking or not. Once Gait-watch detects the user is walking, the accelerometer data will be processed further.

Activity Classifier In realistic environment, the user can perform different activities during walking, which poses a major challenge for gait based recognition on smart watch. To deal with this issue, Gait-watch adopts activity classifier to infer the specific activity of the user and leverages the activity information while performing recognition.

Segmentation and Interpolation After walking detector and activity classifier, we apply gait cycle segmentation and interpolation to obtain the gait cycles from the test signal (accelerometer data). The same optimized projection matrix (as used for training) is used to reduce the dimension of the test signal and provide the measurement vector $\tilde{y}_i = R_{opt}y_i$, $i = 1, 2, \cdots, M$, and M is the number of obtained gait cycles.

Sparse Fusion based Classification Now both the training dictionary $\tilde{A}$ and the measurements $\tilde{y}_i$ are passed to the classifier. The ℓ_1 classifier first finds the sparse coefficient vector x_i. Then the vectors of different gait cycles are fused based on a novel *sparse fusion* model, and the fused sparse vector is used to calculate the residuals and confidence level. Finally, the confidence level is used to recognize whether the walker is the genuine user or imposter.

4 SYSTEM DESIGN

In this section, we detail the design of signal pre-processing, offline dictionary training, and classification in turn.

4.1 Signal Pre-processing

4.1.1 Walking Detector. As the first step, walking detector is essential to avoid applying expensive recognition algorithms during periods of non-motion. The raw accelerometer signals along three axises of smart watch fluctuates greatly and irregularly due to the changing direction of smart watches and arbitrary body movements. However, we observed that the acceleration along gravity direction exhibits regular patterns because of the repetitive nature of walk. Fig. 3 depicts the acceleration values along gravity of different activities. We can see that the acceleration along gravity shows a rhythmic pattern because of the repetitive nature of walk. The intuition is that the smart watch bounces maximally along the gravity dimension, and the bounce-peaks correspond well with the heel strikes. Based on this observation, we apply the method in [25] on acceleration along gravity direction to detect whether the user is walking or not. It is worth mentioning that the repetitive nature of walk was first utilized in NASC [25] to count steps on unconstrained smartphones. We borrow their idea and make some improvements. We give a description of the improved walking detection method in Algorithm 1.

The initial status of the user is set as *NOT WALKING* (include *IDLE* and *RUNNING*). Given an accelerometer signal sequence $a(t)(t = 1, 2, \cdots, N_{acc})$, we compute the normalized auto-correlation of $a(t)$

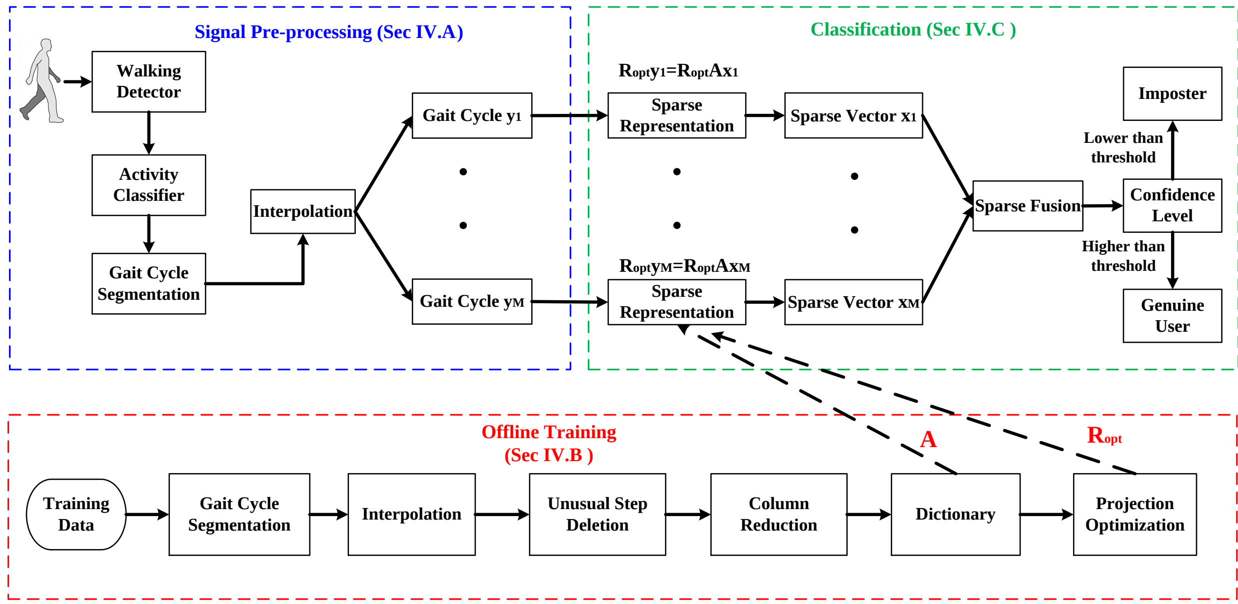


Figure 2: System flow chart of Gait-watch

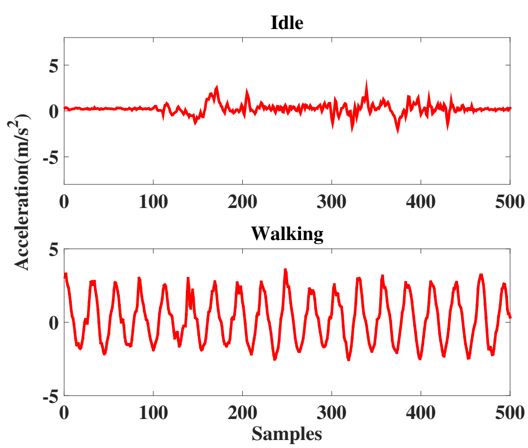


Figure 3: Acceleration along gravity

for lag τ at sample i as [25]:

$$R(i,\tau) = \frac{\sum_{k=0}^{k=\tau-1}\left[\begin{array}{c}(a\,(i+k)-\mu\,(i,\tau))\\ (a\,(i+k+\tau)-\mu\,(i+\tau,\tau))\end{array}\right]}{\tau\delta\,(i,\tau)\,\delta\,(i+\tau,\tau)} \quad (7)$$

where $\mu\,(k,\tau)$ and $\delta\,(k,\tau)$ represent the mean and standard deviation of the samples $\langle a(k), a(k+1), \ldots, a(k+\tau-1)\rangle$. The auto-correlation $R(i,\tau)$ measures the similarity between acceleration signals as a function of the time lag τ. $R(i,\tau)$ is in the range $[-1,1]$ and should be close to 1 when a subject is walking and τ is equal to the cycle of walking pattern. As the period of the subject's walking is unknown, we vary τ between τ_{min} and τ_{max} to find the τ_{opti} which makes the auto-correlation maximum (i.e., τ_{opti} is the period of walking):

$$R_{max}(i) = max_{\tau=\tau_{min}}^{\tau=\tau_{max}} R(i,\tau) \quad (8)$$

Algorithm 1 Walking Detection

Input: acceleration samples $a(i)(1 \leq i \leq N_{acc})$, lag τ.
Initialization: search window $(\tau_{min}, \tau_{max}) = (40, 65)$, walking frequencies $(f_{min}, f_{max}) = (1.2, 2.3)$, auto-correlation threshold R_{th}=0.7, spectral energy threshold $F_{th} = 400$, *STATUS=NOT WALKING.*
for $i = 1 : N_{acc}$ **do**
  for $\tau = \tau_{min} : \tau_{max}$ **do**
    $R(i,\tau)$=Auto-correlation(i,τ)
  end for
  $R_{max}(i) = max\ R(i,\tau)$
  if $R_{max}(i) \geq R_{th}$ **then**
    $F_f = STFT(a(j))\ (1 \leq j \leq N)$
    $F_{walk} = \sum_{f=f_{min}}^{f=f_{max}} F_f$
    if $F_{walk} \geq F_{th}$ **then**
      STATUS=WALKING
      $\tau_{opti} = \arg\max_{\tau} R(i,\tau)$
      $\tau_{min} = \tau_{opti} - 15, \tau_{max} = \tau_{opti} + 15$
    end if
  end if
end for
Output: *STATUS*

As described in Section 4.1.3, normal gait duration lies in 0.8-1.3s which produces 40-65 samples (the sampling rate of accelerometer is 50Hz). Therefore, the initial value of (τ_{min}, τ_{max}) is set to (40, 65). Once the period of the subject's walking pattern is found in the first few steps (i.e., τ_{opti}), the search window is reduced to ($\tau_{opti} - 15, \tau_{opti} + 15$).

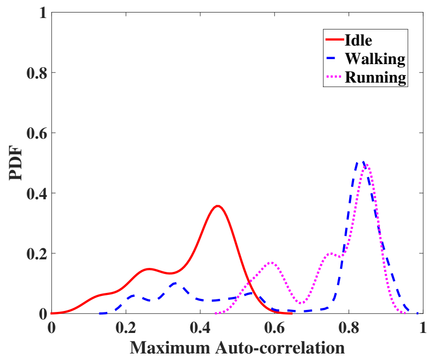


Figure 4: Distribution of maximum auto-correlation of different activities.

Fig. 4 depicts the distribution of $R_{max}(m)$ for *IDLE*, *WALKING* and *RUNNING*, we can see that the probability that the person is *IDLE* is extremely low when $\psi(t) \geq 0.7$. However, we noticed that the maximum auto-correlation values is similar between *WALKING* and *RUNNING* as both of them are repetitive activities. We notice that the major difference between *WALKING* and *RUNNING* is the frequency. In order to distinguish different repetitive activities, we apply Short Term Fourier Transform (STFT) on the acceleration samples and calculate the spectral energy F_{walk} (i.e. the magnitude of the STFT coefficients) at typical walking frequencies (f_{min}, f_{max}), we label the subject as walking if F_{walk} is greater than a predefined threshold F_{th} (400 in our system). Note that this method can also be used to detect other repetitive activities such as running by simply changing frequencies (f_{min}, f_{max}) to typical running frequency ranges; however, in this paper we consider walking only.

4.1.2 Activity Classifier. After walking detection, we build a activity classifier to detect how the user is walking. In this work, we focus on 7 of the most common activities while people are walking and divide them into two categories as shown in Table 1. The first category is walking with arm swing which include normal walk, walk upstairs and walk downstairs. The second category is walking without arm swing which include walk with texting the phone, calling the phone, and hand in jacket/pant pocket.

Table 1: Normal activities during walking

With arm swing	Without arm swing
normal walk	walk with texting the phone
walk upstairs	walk with calling the phone
walk downstairs	walk with hand in jacket pocket
	walk with hand in pant pocket

The real time data from an accelerometer contains much noise that needs to be filtered out before using it for activity recognition. Thus a moving average filter of order 3 is applied for noise removal. After noise reduction, continuous sensor data is segmented into 2s sliding windows with 50% overlap. The window size of 2s is chosen to balance between classification accuracy and latency as discussed in Section 5.2. The overlap in sliding window is used to capture changes or transitions around the window limits. For each window, we extract a number of features as listed in Tab 2. Previous studies used features from both time and frequency domain representations of signals for activity recognition [30]. For computation efficiency, we consider light-weight time domain features only in the system. Each window is represented as a feature vector of length 18. These features are then used to train the classifier. As the evaluation in Section 5.2, we choose k-NN classifier as it achieves higher recognition accuracy than SVM and Decision Tree.

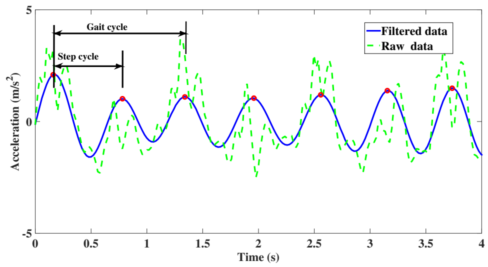
Table 2: List of features used for activity classification

Feature	Accelerometer axis
Mean	X,Y,Z
Min	X,Y,Z
Max	X,Y,Z
Standard Deviation	X,Y,Z
Variance	X,Y,Z
Ratio	X/Y,X/Z,Y/Z

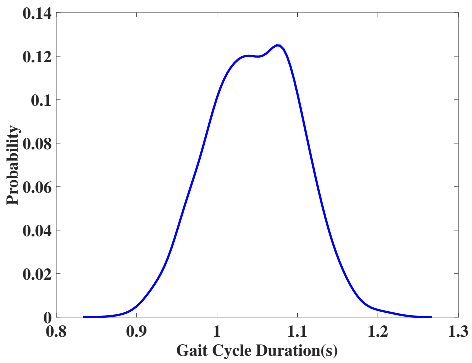
4.1.3 Gait Cycle Segmentation. In order to recognize a gait signal, it is essential that we separate the time series of walking periods into segments, such that each segment contains a complete gait cycle. The gait cycle can be obtained by combining two successive step cycles together as technically the gait cycle is across a *stride* (two steps). Previous studies have developed approaches that reliably detect and count steps on smartphones [25, 37]. A common way is to combine (x, y, z) acceleration data together and then detect peaks as in [37]; however, we found that this method is not suitable for smart watch due to various arm movements. Large body movements such as turning a corner bring in additional acceleration, thus decrease the step extraction accuracy. To address this issue, we compute the acceleration along the gravity. Assume the linear acceleration signals along three orthogonal directions of smart watches are Acc_x, Acc_y,and Acc_z respectively, the linear acceleration in the world coordinate system can be computed as:

$$\begin{bmatrix} Acc_E \\ Acc_N \\ Acc_G \end{bmatrix} = R \cdot \begin{bmatrix} Acc_x \\ Acc_y \\ Acc_z \end{bmatrix} \tag{9}$$

Where Acc_E,Acc_N,and Acc_G are linear acceleration along three directions (East, North and Gravity) in the world coordinate system, and R is the rotation matrix from the device coordinate system to the world coordinate system which can be obtained by Android API. The advantage of this method is that Acc_G is independent of the watch's orientation and user's arm motion, thus can address challenges raised in dynamic mobility scenarios. The raw sensor data of Acc_G are deteriorated by high frequency noise. As mentioned in Section. 4.1.1, typical step frequencies are around 1-2Hz, we thus apply a band-pass Butterworth filter [4] on the sampled data to eliminate out-band interference. The lower and upper cutoff frequency is set as 1Hz and 2Hz separately (filter order is 4). After filtering, the step cycles are separated by finding peaks associated with the heel strike as shown in 5(a). Thereafter, the gait cycle is obtained by combining two consecutive step cycles together.



(a) The time series of Acc_G



(b) Distribution of cycle duration

Figure 5: Illustration of gait cycle detection

After gait cycle extraction, the sensor data are segmented into short gait cycles based on the peak detection. Fig. 5(b) presents the distribution of cycle duration (i.e. time length of stride) for 20 healthy subjects walking at their normal speed. We can see that most of the gait cycle ranges between 0.8s-1.3s (40-65 samples). This results in turn can be used to omit unusual gait cycles and exclude the cycles not produced by walking, i.e., the cycles which last less than 0.8s and exceed 1.3s are dropped.

4.1.4 Linear Interpolation. Detected cycles are normalized to equal length by linear interpolation because SRC require vectors of equal length as input. As mentioned above, normal gait duration lies between 40 and 65 samples, we apply linear interpolation on the samples to ensure that they achieve the same length of 80 samples. We also evaluate the performance of different interpolation methods and find that they exhibit comparable recognition accuracy, we choose linear interpolation due to its less computational cost on resource-constrained smart watches.

4.2 Offline Training

The training data are also passed to gait cycle segmentation and linear interpolation to obtain gait cycles with same length. In addition, we delete unusual cycles and optimize projection matrix to further improve recognition accuracy.

4.2.1 Deletion of Unusual Cycles. Unusual cycles caused by occasional abnormalities like temporary walking pauses or turning contains much noise that will deteriorate the recognition accuracy. Apart from deleting unusual cycles using cycle durations, the detected cycles are also passed to a function which further deletes unusual cycles. This function uses Dynamic Time Warping (DTW) distance scores to remove outliers from a set of cycles. Specifically, we first compute the DTW distance between the detected cycle and typical cycle. Thereafter, we delete unusual cycles by a simple threshold method, i.e., if the DTW distance of detected cycle and typical cycle is higher than a predefined value (12 in the proposed system), the detected cycle will be dropped. The typical cycle is the one which is assumed to represent the subject's gait signal. This is obtained by computing the average of all cycles in the training data.

4.2.2 Column Reduction and Projection Optimization. After unusual cycles removal, the remaining gait cycles obtained from training data are used to form the training dictionary A. According to the formation of ℓ_1−Homotopy, the computational complexity is $O(s^3 + sq(N \cdot K))$, where s is the sparsity of the solution ($s \ll N \cdot K$), q is the number of equations, and $N \cdot K$ is the number of unknowns, i.e., the number of columns in the training dictionary. We can see that the computation of ℓ_1 optimization is also proportional to the number of columns $(N \cdot K)$ in the dictionary A. The gait cycles in the same class are highly correlated and lead to intra class redundancy. To reduce the intra class redundancy in the dictionary while retaining the most informative columns, we apply the columns reduction approach [35] to improve the efficiency of G-watch. Furthermore, motivated by a recent work [29], we apply the projection matrix optimization algorithm to reduce the dimensionality of A while retaining the high classification accuracy. The projection matrix R_{opt} is learned from dictionary A based on Tabu search. We refer the reader to [29] for more details.

4.3 Sparse Fusion Model for Classification

SRC aims to solve the classification problem of one test vector, however, the evaluation results in Section 5.3.1 show that the recognition accuracy of using one gait cycle can achieve 86% only. To overcome this limitation, we propose a novel *sparse fusion* model which fuses the sparse coefficients vectors from multiple consecutive gait cycles to further improve recognition accuracy.

The key assumption behind the proposed method is that gait cycles obtained from consecutive gait cycles tend to have a high agreement on the sparse representations because each of the gait cycles from the same person should be linearly represented by the same class in the dictionary. Suppose we have acquired a set of M gait cycles from the test signal. Following the single test vector approach described in Section 2, we can obtain a set of estimated coefficients vectors $\hat{X} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_M\}$ by solving the ℓ_1 optimization problem for each gait cycle. Theoretically, a precise sparse representation will only contain the non-zero entries at the locations related to the specific class. However, noise exists in the empirical estimations. Therefore, the estimated coefficients vector of the m-th test gait cycle can be expressed as: $\hat{x}_m = x + \epsilon_m$, where x is the theoretical sparse representation of the test vector and ϵ_m is used to account for noise.

The test vector could be misclassified due to low Signal to Noise Ratio (SNR). To enhance the SNR of the classification system, we propose a new sparse representation model by exploiting the information from multiple gait cycles. The new sparse representation model can be expressed as: $\hat{x}_{sum} = \sum_{m=1}^{M} \alpha_m \hat{x}_m$, where α_m is the weight assigned to $\hat{x}_m$ based on the *Sparsity Concentration Index* (SCI) defined in [36]:

$$SCI(\hat{x}_m) = \frac{K \cdot max_j \|\delta_j(\hat{x}_m)\|_1 / \|\hat{x}_m\|_1 - 1}{K - 1} \in [0, 1] \quad (10)$$

the SCI measures how concentrated the coefficients are in the dictionary. $SCI(\hat{x}_m) = 1$, if the test vector can be strictly linearly represented using training vectors from only one class; and $SCI(\hat{x}_m) = 0$, if the coefficients are spread evenly over all classes. The weight of $\hat{x}_m$ is obtained by normalizing the SCIs among the obtained M gait cycles:

$$\alpha_m = SCI(\hat{x}_m) / \sum_{n=1}^{M} SCI(\hat{x}_n) \quad (11)$$

With the knowledge of $\hat{x}_{sum}$, the compressed residual of each class is computed as:

$$r_i(y_{sum}) = \|R_{opt} y_{sum} - R_{opt} A \delta_i(\hat{x}_{sum})\|_2 \quad (12)$$

where $y_{sum} = \sum_{m=1}^{M} \alpha_m y_m$ is the weighted summation of all the test vectors. Following the same approach in [29, 36], the final classification result is obtained by finding the minimal residual.

Although SRC is a multi-class classification approach, we can modify it to achieve authentication purpose. Generally, an authentication system only requires to collect data from the genuine user. To meet the sparse requirement of SRC, we can store gait data from other subjects before releasing the system to user. Then the user only requires to collect his own gait to make the system work. To identify whether the walker is the genuine user or imposter, we adopt the same principle in [29] by using confidence level defined as:

$$confidence = \left(\frac{1}{K} \sum_{i=1}^{K} r_i - \min_{i=1,\ldots,K} r_i \right) / \frac{1}{K} \sum_{i=1}^{K} r_i \quad (13)$$

The confidence level is in the range of $[0, 1]$ and the verification decision can be made by:

$$confidence \begin{cases} \geq C & \text{genuine user} \\ < C & \text{imposter} \end{cases}$$

where C is a threshold we set empirically. An appropriate threshold can be chosen by data-driven approach to make the recognition system robust to imposters.

5 EVALUATION

5.1 Goals, Metrics and Methodology

In this section, we evaluate the performance of Gait-watch via simulation. The goals of the simulation are threefold: 1) to evaluate the performance of activity classifier; 2) whether the proposed sparse fusion classification outperform state-of-the-art classification methods; 3) whether Gait-watch achieves high accuracy in differentiating imposters and genuine users.

In this paper, we focus on the following metrics:

- **Recognition accuracy**: it represents the percentage of correct classifications which can be calculated as the percentage of the total number of tests that resulted in correct classifications.
- **False positive rate (FPR)**: it is the measure of probability that the authentication system incorrectly accepts the access request by an imposter.
- **False negative rate (FNR)**: it is the measure of the probability that the authentication system incorrectly rejects the access requests from the genuine users.

In general, FPR relates to the security of the system, while FNR to the usability. An interesting point in the Decision Error Trade-off (DET) curve is the Equal Error Rate (EER) where FPR=FNR. For instance, an EER of 5% means that out of 100 genuine trials 5 is incorrectly rejected, and out of 100 impostor trials 5 are wrongfully accepted.

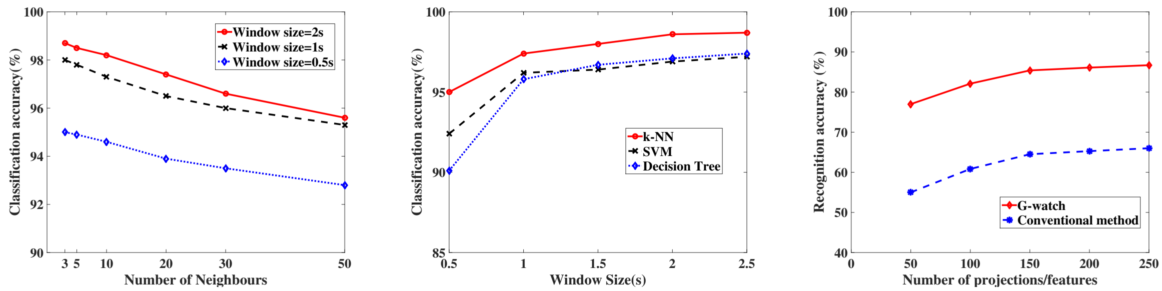
Data Collection. To evaluate the performance of Gait-watch, we collected data (accelerometer data) with Samsung smart watch in both controlled environment and real-world environment[1].

Activity dataset. The activity dataset is used to evaluate the performance of activity classifier. It consists of 15 subjects (11 males and 4 females) with different skin tones and heights. The participants perform the activities listed in Table 1 on two different days. For each activity, the participants were requested to walk 3 minutes at their normal speed. In total, we collected about 5 hours of accelerometer data for activity recognition.

Gait dataset. The Gait dataset is used to evaluate the recognition accuracy of Gait-watch. During data collection phase, the participants wore the smart watch at one of their wrists at will, and were asked to perform the fore-mentioned activities with equal proportion. The data collection is performed in several environments (indoor and outdoor) in order to capture different terrains. There were 20 volunteers (14 males and 6 females) participating in data acquisition. An illustration of indoor environment and outdoor environment is shown in Fig 7(a) and Fig 7(b). The terrain of the chosen outdoor environment varies including plain, grass and asphalt. Each volunteer participated in two data collection sessions that was separated by 1 week. We will refer to these sessions as *session 1* and *session 2* respectively. A participant might dress in different ways (e.g. different clothes and shoes) for the two sessions. Based on the above description, the gait dataset is close to a realistic environment as it includes the natural gait changes over time and different environments (indoor and outdoor). In total, we obtained about 10 hours of gait data to evaluate the performance of Gait-watch.

In the following evaluations, we perform 10-fold cross-validation on the collected dataset and plot the results of the average values. Specifically, we randomly split the dataset into 10 folds with equal size. Then, each fold is retained as the validation data for testing the classifier, and the remaining 9 folds are used as training data. The cross-validation process is then repeated 10 times, with each of the 10 folds used exactly once as the testing data.

[1] Ethical approval for carrying out this experiment has been granted by the corresponding organization



(a) (b) (c)

Figure 6: Evaluation results of activity classification: (a) accuracy of different number of neighbors. (b) comparison of different methods. (c) improvement of recognition accuracy by using activity classifier

(a) Indoor experiment (b) Outdoor experiment

Figure 7: Illustration of data collection.

5.2 Performance of Activity Classifier

In this section, we evaluate the recognition accuracy of different classifiers based on activity dataset. Then, we demonstrate that Gait-watch improves recognition accuracy by leveraging the activity information.

5.2.1 Comparison of Different Activity Classifiers. We evaluate the recognition accuracy of three mostly used classifier in online activity recognition: k-NN, SVM and Decision Tree [16]. We first evaluate the accuracy of k-NN by varying k (the number of nearest neighbor) from 3 to 50. As shown in Fig 6(a), k-NN achieves best accuracy when $k = 3$. After k is determined, we evaluate the accuracy of different methods by varying window size from 0.5s to 2.5s. As we can see from Fig 6(b), k-NN achieves the classification accuracy by up to 98.6% when window size is chosen as 2s, and shows higher accuracy than SVM and Decision Tree at all feasible window size. Based on this experiment, we choose k-NN as activity classifier, the window size and k value is chosen as 2s and 3 respectively. It should be noted that k-NN is used to detect the activity of the user rather than verifying the identity of the user.

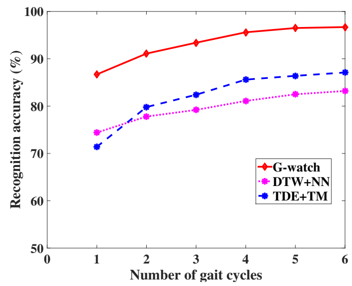
5.2.2 Improvement of Recognition Accuracy by Using Activity Classifier. Traditional gait-based recognition system assumes the user walks normally. In order to deal with the problem of various activities, a common method is to train a large dictionary by assembling gait cycles from all activities. We term this method as *conventional method*, and compare it with context-aware Gait-watch. For *conventional method*, we use all gait cycles of each subject to form the training dictionary. Regarding to Gait-watch, we first perform activity detector to check the user's activity. Based on the result of activity classifier, user recognition is performed based on the appropriate training data. For example, if the test gait cycle is labeled with walking upstairs, we use gait cycles from walking upstairs to form the training dictionary only. We use one gait cycle as test vector and vary the number of projections/features d (i.e., the number of rows in R_{opti}) from 50 to 250. As shown in Fig 6(c), we can observe a significant accuracy improvement of Gait-watch over *conventional method*, the improvement can be up to 20%. The results suggest that it is critical to know the user's activity for authentication, and the proposed context-aware authentication system improves recognition accuracy significantly by leveraging the activity information.

5.3 Performance of Sparse Fusion Classification

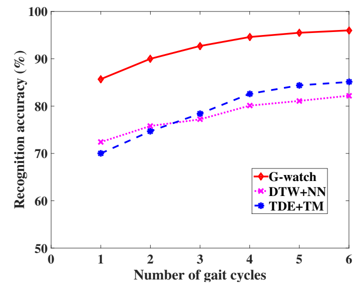
In this set of experiments, we compare the proposed sparse fusion classification method with several state-of-the-art gait recognition methods on gait dataset.

5.3.1 Comparison with Other Gait Recognition methods. We compare Gait-watch with several state-of-the-art gait recognition methods, namely, DTW+NN [5] and TDE+TM [9]. We perform evaluation in two categories: same day evaluation and different days evaluation. Same day evaluation means the training set and test set are chosen from the sessions of the same day while different days evaluation chooses the sessions of different days. We set $d = 200$ as Fig 6(c) shows that there is no significant improvement when $d > 200$. We vary the number of gait cycles M from 1 to 6 and calculate the recognition accuracy of different methods.

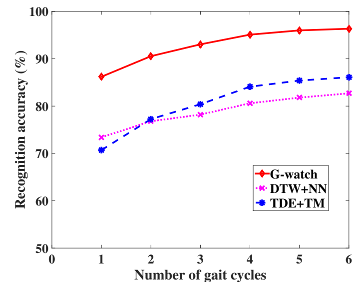
Fig 8(a) presents the evaluation results from same day evaluation, we can see that Gait-watch achieves the best recognition accuracy. Fig 9(b) plots the recognition accuracy from different days evaluation. The accuracy of all the gait recognition methods is lower than the same day evaluation as the different days evaluation tends to produce more dynamics between the training data and test data. The observations from the same day evaluation still hold true in the different days evaluation. The overall recognition accuracy of different methods are shown in Fig 8(c). We can see that Gait-watch is 10% better than TDE+TM and 14% better than DTW+NN when



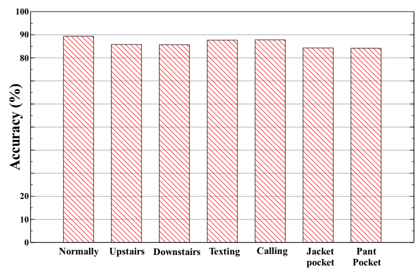
(a) Same day evaluation



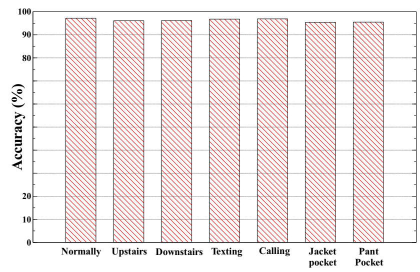
(b) Different day evaluation



(c) Overal results

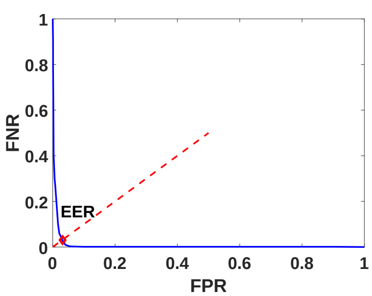


(d) Accuracy of different activities (M=1)

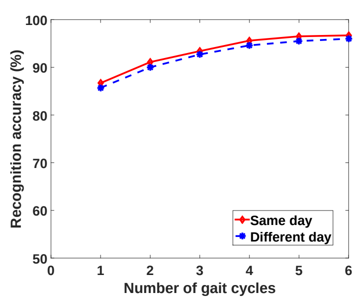


(e) Accuracy of different activities (M=5)

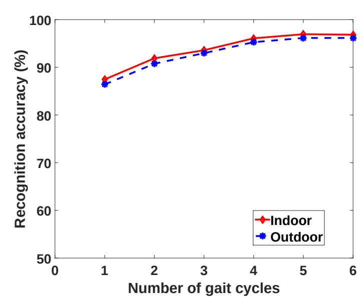
Figure 8: Evaluation results



(a) Verification accuracy



(b) Different day



(c) Different environment

Figure 9: Robustness to attackers and gait variations.

$M = 5$. We also notice that the recognition accuracy of Gait-watch increases when more gait cycles are fused for recognition, this is intuitive as more information can be obtained by fusing multiple gait cycles together. The overall recognition accuracy of Gait-watch is over 96% when 5 or more consecutive gait cycles are used. We define the number of gait cycles as a user or application defined parameter; more accurate recognition will be achieved when the user would like to make more efforts.

5.3.2 Impact of Different Activities. We now evaluate the recognition accuracy under different activities to explore the impact of activity on Gait-watch. For each activity, we use the gait cycles extracted from the activity to perform 10-fold cross-validation. We set $d = 200$ and calculate the recognition accuracy of different activities by choosing $M = 1$ and $M = 5$, respectively. From the results in Fig 8(d), we can see that walking normally shows better performance than other activities if we use only one gait cycle for recognition. Walking with hand in jacket/pant pocket performs the worst because the subjects wore different clothes at different days during data collection phase. The results indicate that, if we do have control over the users, walking normally is a good activity to use for identification purposes. However, when we fuse several gait cycles to perform recognition, different activities show comparable recognition accuracy as shown in Fig 8(e). This result demonstrates the advantage of the proposed sparse fusion method. Also, it indicates that Gait-watch can recognize the user under various activities at a high accuracy.

5.4 Robustness Against Attackers

As mentioned previously, an attacker can imposter the genuine user to gain access to the smartwatch. To evaluate the robustness of Gait-watch against the imposter attack scenario, we group the 20 subjects into 10 pairs. Each subject was told to mimic his/her partner's walking style and try to imitate him or her. Firstly, one participant of the pair acted as an attacker, the other one as a target, and then the roles were exchanged. The genders of the attacker and the target were the same. They observed the walking style of the target visually, which can be easily done in a real-life situation as gait cannot be hidden. Every attacker made 5 active impostor attempts. As an authentication system, Gait-watch is sensitive to false acceptance rate. Therefore, we set $M = 8$ to reduce false positive rate. We vary the confidence threshold C and plot DET curve in Fig 9(a). The red dash line stands for the possible points where FPR is equal to FNR. The crossover (marked as a diamond) of the red dash line and FPR-FNR curve stands for the location of the EER. We notice that EER of Gait-watch is as low as 3.1%, which means out of 100 impostor trials only 3 are wrongfully accepted. The setting of more steps indicates more time it will take before collecting sufficient step cycles to detect the suspect. Therefore, we define the number of steps as a user-defined parameter: more accurate authentication will be achieved if the users would like to pay more efforts on the walking.

5.5 Robustness to Gait Variations

To evaluate the robustness of Gait-watch to gait variations, we conduct the following two experiments: different day evaluation and different environment evaluation. In this experiment, same day evaluation means the training set and test set are chosen from the sessions of the same day while different days evaluation chooses the sessions from two different days separated by 1 week. Similarly, in different environment evaluations, indoor evaluation means the training set and test set are chosen from indoor environment while outdoor evaluation chooses training data and test data from outdoor environment. As the results in Figure 9(b), the accuracy of different day is lower than the same day evaluation as the different days evaluation tends to produce more changes to gait. However, Gait-watch can still achieve the accuracy of 95% when more than 5 steps are used. This observation holds in the different environment evaluation. From Figure 9(c), we can see outdoor environment achieves lower accuracy than indoor environment because it includes several different terrains such as grass path and asphalt road.

Many factors exist that may impact the accuracy of a gait-based recognition system, such as shoe, clothes, walking speed and terrain. Previous studies have shown that the accuracy will decrease when the test and training samples of the person's walking are obtained using different shoe types and clothes [8]. Indeed, as shown in Section 5.5, the accuracy of Gait-watch decrease when session 1 is used for training and session 2 is used for testing. The dataset used in the experiment is challenging as it includes the natural gait changes over time (two sessions separated by 1 week), as well as gait variations due to changing in clothes, terrain and shoes. However, Gait-watch can still achieve the accuracy of 95% by the proposed sparse fusion model, which in turn demonstrate the robustness of Gait-watch to gait variations. Due to space limitation, we defer the analysis of different factors as our future work. In fact, there has been several attempts to study the relationship between recognition performance and different factors [8, 19]. For example, in terms of walking speed, Muhammad and Claudia [19] found that normal walk has best results and fast walk is a bit better than slow walk. As for different types of terrains, they reported that gravel walk has better results than grass and inclined walk. We encourage the reader to refer [8, 11, 19] for more details.

6 USER STUDY IN THE WILD

Our final study aims to evaluate the performance of Gait-watch in real world scenarios.

6.1 System Implementation

The prototype of Gait-watch is implemented on Samsung Gear Live Smart Watch[2]. The CPU is a Qualcomm Snapdragon at 1.2GHz and the operating system is Android wear. The efficient implement of ℓ_1 optimization algorithm ℓ_1-*Homotopy* [7] is used. To reduce the expected response time, we implement Gait-watch in multiple threads. 6 threads are allocated for Gait-watch according to the system setting. One of the threads is responsible for the step detection. The rest of the threads are in idle before the sensor data of gait cycles are received. For each time a new step is detected, the corresponding sensor data of the step cycle is passed to activate a new thread. After all the sparse coefficients vectors are obtained by dynamic sparse representation, the authentication decision is determined by comparing the confidence level with a pre-defined parameter C as Eq 4.3. The prototype of Gait-watch uses full dictionary whose number of rows are 200 and number of columns for each class is 50. Besides, the number of gait cycles is set as $M = 5$ and the confidence level threshold is set as $C = 0.4$.

6.2 User Study

We recruited 3 users (2 males and 1 female) for this study. The users wore the smart watch for 2 hours per day for three different days. We asked the users to launch Gait-watch after they put the smart watch on and end it after they take it off. Gait-watch works continuously after it is launched, and logs the authentication decision after the required number of gait cycles (M=5) are detected.

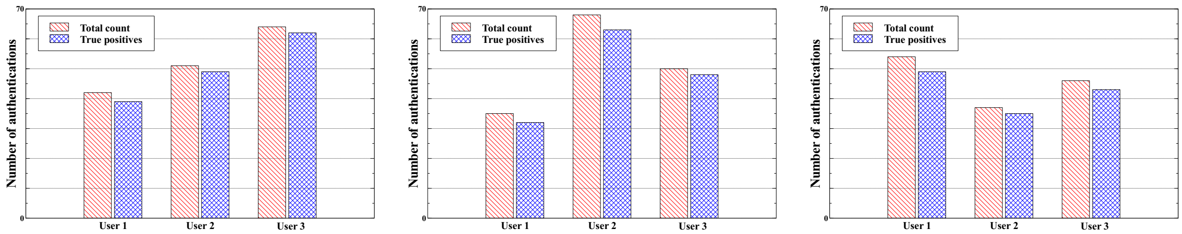
In practice, Gait-watch can run in a power saving mode. It starts as a background service when the user puts the smart watch on, and performs authentication when it detects the wearer is walking. If the wearer is accepted as genuine user, Gait-watch stops working and the wearer will be regarded as the owner continuously without additional authentication before the take-it-off action is detected. This is due to the fact that the smart watch must be on the same subjectfis wrist before a take-it-off activity is detected. The put-it-on and take-it-off activities can be detected by the take-on sensor in smart watch.

6.3 Experimental Results

Fig 10 shows the results of the user study. We notice that the total number of authentication attempts varied between 30 to 68 per day. of all the 447 detected authentications, Gait-watch authenticates the genuine users correctly by 427 times, i.e., the True Positive Rate (TPR) achieves up to 95.3%. The results demonstrate the effectiveness of Gait-watch in authenticating users in real world scenarios.

Table 3 shows the resource consumption of Gait-watch measured on Samsung Smart Watch. The computation time is obtained from

[2] A demonstration of Gait-watch can be found at the following URL (in the demo, the system is implemented on smartphone for better illustration.): https://youtu.be/LOqSq28oexA



(a) Day 1 (b) Day 2 (c) Day 3

Figure 10: Evaluation results of user study: total count means the total number of authentication attempts user performed and true positive means the number of accepted authentications.

Table 3: System overhead.

	Computation time	Energy consumption
Walking detection	120 ms	121 mJ
Activity classification	80 ms	20.6 mJ
Signal processing	24 ms	36 mJ
Classification	670 ms	467 mJ
Total	894 ms	644.6 mJ
Memory usage	35-42 MB	

the console of the Eclipse development environment and averaged by the results from 30 authentication attempts. The memory usage is measured by PowerTutor App (it was also used in [29]). The computation time of the four stages in the pipeline: walking detection, activity detection, segmentation and interpolation, classification take an average time of 120, 80, 24 and 670 ms, respectively. When the whole pipeline is fully engaged, Gait-watch takes about 4s to collect the required gait data and can respond in approximately 900 ms. Besides, it consumes 644.6 mJ only. The memory requirement to execute Gait-watch is modest and varies between 35-42 MB.

The user study results show that Gait-watch can recognize the user accurately in real world scenarios and require low system cost. Therefore, it is feasible to implement Gait-watch on off-the-shelf smart watches. However, we are aware that gait-based authentication system cannot provide a absolutely secure way to protect the data in smart watch. We imagine in the future to have many levels of security (e.g., two factor authentication) that tradeoff usability and accessibility given the risk imposed. For instance, accessing a user's bank account clearly requires higher security than retrieving the time of the next meeting and occurs less frequently, hence requiring additional authentication scheme (e.g., PIN) after Gait-watch grants user the access to use the smart watch.

7 RELATED WORK

In this section, we discuss the related work in two aspects: gait recognition and applications of SRC.

Gait Recognition: Gait recognition has been well studied in the literature. From the way how gait is collected, gait recognition can be categorized into three groups: vision based, floor sensor based, and wearable sensor based. In vision based gait recognition system, gait is captured from a remote distance using video-camera. Then, video/image processing techniques are employed to extract gait features for further recognition. A large portion in the literature belong to this category [13, 28]. In floor sensor based gait recognition, sensors (e.g., force plates), which are usually installed under the floor, are used for capturing gait features, such as ground reaction force (GRF) [21, 31] or heel-to-toe ratio [17, 22].

Compared with vision-based and other non-accelerometer based gait measurements, acceleration can reflect the dynamics of gait more directly and faithfully. For instance, accelerometer based gait recognition do not suffer from the existing problems for vision-based methods, like occlusions, clutter, and viewpoint changes. Existing works of wearable sensor based gait recognition are mainly based on the use of body-worn accelerometers. The first work of accelerometer based gait recognition is proposed by Ailisto et al. [1] and further developed by Gafurov et al. [10]. In the initial stages, dedicated accelerometers were used and worn on different body positions, such as lower leg [10], waist [1], hip [12], hip pocket, chest pocket and hand [33]. With the prevailing of smartphone, researchers have proposed several gait-based authentication systems by utilizing the built-in accelerometer [15, 20, 23, 26]. For example, in [15], the researchers have proposed an unobtrusive gait verification system for mobile phones. In [23], the authors studied the impact of different phone locations such as hand and pocket. In recent years, researchers start to use emerging energy harvester to achieve gait recognition [39]. In addition, the unique gait has been exploited for key generation to protect user's personal devices [38, 40].

Although various feature selection and classification methods have been proposed, most of the existing studies assume that the user is walking normally except [14]. In [14], the researcher aim to identify users based on the way during their multiple activities include walking, jogging, climb up stairs, and climb down stairs. However, the recognition accuracy is relatively low (70%-90%). To the best of our knowledge, this is the first work for gait-based authentication system that considers various activities. To overcome the challenge of various activities, we propose a context-aware authentication system and a *sparse fusion* method to improve the recognition accuracy.

Applications of SRC: SRC is an emerging classification method and has been widely used in recognition tasks of sensor areas. In [35], the authors developed an acoustic classification system on wireless sensor networks by applying SRC to improve the recognition accuracy. In [29], the researchers proposed opti-SRC by optimizing the random matrix used in SRC to increase the performance of face recognition system in smartphones. Several papers have exploited the sparsity of multiple measurements to improve the system performance. [18] used CS to compress GPS signals and exploits the information of various propagation paths to improve the SNR of GPS signals. In [34], the authors improved activity classification accuracy by fusing several channel state information (CSI) vectors. Weitao et al. also exploited the sparse representation of several face images from different views to improve face recognition accuracy [41].

8 CONCLUSION

In this paper, we address the problem of recognizing the user by emerging smart watches. Gait-watch provides an unobtrusive, continuous authentication way for smart watch users. Extensive evaluation results show that Gait-watch can recognize the user accurately in real world scenarios and outperforms several state-of-the-art gait recognition systems. We also perform a user study to demonstrate the feasibility of Gait-watch for commercial smart watches.

ACKNOWLEDGMENTS

We would like to thank our shepherd Assistant Professor Rui Tan and the anonymous reviewers for their helpful feedbacks on earlier versions of this paper. We also would like to thank the volunteers who participant in the experiments.

REFERENCES

[1] Heikki J Ailisto, Mikko Lindholm, Jani Mantyjarvi, Elena Vildjiounaite, and Satu-Marja Makela. 2005. Identifying people from gait pattern with accelerometers. In *Defense and Security*. International Society for Optics and Photonics, 7–14.

[2] Mojtaba Alizadeh, Saeid Abolfazli, Mazdak Zamani, Sabariah Baharun, and Kouichi Sakurai. 2016. Authentication in mobile cloud computing: A survey. *Journal of Network and Computer Applications* 61 (2016), 59–80.

[3] Adam J Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M Smith. 2010. Smudge Attacks on Smartphone Touch Screens. *WOOT* 10 (2010), 1–7.

[4] Stephen Butterworth. 1930. On the theory of filter amplifiers. *Wireless Engineer* 7, 6 (1930), 536–541.

[5] Michael B Crouse, Kevin Chen, and HT Kung. 2014. Gait Recognition Using Encodings With Flexible Similarity Measures. (2014).

[6] D.L. Donoho. 2006. Compressed sensing. *IEEE Transactions on Information Theory* (2006), 1289–1306.

[7] D.L. Donoho and Y. Tsaig. 2008. Fast Solution of ℓ_1 -Norm Minimization Problems When the Solution May Be Sparse. *Information Theory, IEEE Transactions on* 54, 11 (2008), 4789–4812.

[8] Shuichi Enokida, Ryo Shimomoto, Tomohito Wada, and Toshiaki Ejima. 2006. A predictive model for gait recognition. In *2006 Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference*. IEEE, 1–6.

[9] Jordan Frank, Shie Mannor, and Doina Precup. 2010. Activity and Gait Recognition with Time-Delay Embeddings.. In *AAAI*. Citeseer.

[10] Davrondzhon Gafurov, Kirsi Helkala, and Torkjel Søndrol. 2006. Biometric gait authentication using accelerometer sensor. *Journal of computers* 1, 7 (2006), 51–59.

[11] Davrondzhon Gafurov and Einar Snekkenes. 2009. Gait recognition using wearable motion recording sensors. *EURASIP Journal on Advances in Signal Processing* 2009 (2009), 7.

[12] Davrondzhon Gafurov, Einar Snekkenes, and Tor Erik Buvarp. 2006. Robustness of biometric gait authentication against impersonation attack. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*. Springer, 479–488.

[13] Jinguang Han and Bir Bhanu. 2006. Individual recognition using gait energy image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, 2 (2006), 316–322.

[14] Jennifer R Kwapisz, Gary M Weiss, Samuel Moore, and others. 2010. Cell phone-based biometric identification. In *BTAS' 2010*. IEEE, 1–7.

[15] Hong Lu, Jonathan Huang, Tanwistha Saha, and Lama Nachman. 2014. Unobtrusive gait verification for mobile phones. In *ISWC' 2014*. ACM, 91–98.

[16] Andrea Mannini and Angelo Maria Sabatini. 2010. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors* 10, 2 (2010), 1154–1175.

[17] Lee Middleton, Alex Buss, Alex Bazin, Mark S Nixon, and others. 2005. A floor sensor system for gait recognition. In *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*. IEEE, 171–176.

[18] Prasant Kumar Misra, Wen Hu, Yuzhe Jin, Jie Liu, Amanda Souza de Paula, Niklas Wirstrom, and Thiemo Voigt. 2014. Energy efficient GPS acquisition with sparse-gps. In *IPSN' 2014*. IEEE Press, 155–166.

[19] Muhammad Muaaz and Claudia Nickel. 2012. Influence of different walking speeds and surfaces on accelerometer-based biometric gait recognition. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*. IEEE, 508–512.

[20] Claudia Nickel, Tobias Wirtl, and Christoph Busch. 2012. Authentication of smartphone users based on the way they walk using k-NN algorithm. In *IIH-MSP' 2012*. IEEE, 16–20.

[21] Robert J Orr and Gregory D Abowd. 2000. The smart floor: a mechanism for natural user identification and tracking. In *Proc. of CHI*. ACM, 275–276.

[22] F Pelisse, F Marie, and P Richard. 1990. The use of a microprocessor for the partial assistance of a prosthesis for above knee amputees. *Journal of Microcomputer Applications* 13, 2 (1990), 145–153.

[23] Abena Primo, Vir V Phoha, Ravindra Kumar, and Abdul Serwadda. 2014. Context-aware active authentication using smartphone accelerometer measurements. In *CVPRW' 2014*. IEEE, 98–105.

[24] Mashfiqui Rabbi, Shahid Ali, Tanzeem Choudhury, and Ethan Berke. 2011. Passive and in-situ assessment of mental and physical well-being using mobile sensors. In *Ubicomp' 2011*. ACM, 385–394.

[25] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. 2012. Zee: zero-effort crowdsourcing for indoor localization. In *Mobicom' 2012*. ACM, 293–304.

[26] Yanzhi Ren, Yingying Chen, Mooi Choo Chuah, and Jie Yang. 2013. Smartphone based user verification leveraging gait recognition for mobile healthcare systems. In *Secon' 2013*. IEEE, 149–157.

[27] Florian Schaub, Ruben Deyhle, and Michael Weber. 2012. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 13.

[28] Siraj A Shaikh and Joseph R Rabaiotti. 2010. Characteristic trade-offs in designing large-scale biometric-based identity management systems. *Journal of Network and Computer Applications* 33, 3 (2010), 342–351.

[29] Yiran Shen, Wen Hu, Mingrui Yang, Bo Wei, Simon Lucey, and Chun Tung Chou. 2014. Face recognition on smartphones via optimised sparse representation classification. In *IPSN '14*. IEEE Press, 237–248.

[30] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. 2015. A survey of online activity recognition using mobile phones. *Sensors* 15, 1 (2015), 2059–2085.

[31] Khairul Azami Sidek, Vu Mai, and Ibrahim Khalil. 2014. Data mining in mobile ECG based biometric identification. *Journal of Network and Computer Applications* 44 (2014), 83–91.

[32] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. 2013. Quantifying the security of graphical passwords: The case of android unlock patterns. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 161–172.

[33] Elena Vildjiounaite, Satu-Marja Mäkelä, Mikko Lindholm, Reima Riihimäki, Vesa Kyllönen, Jani Mäntyjärvi, and Heikki Ailisto. 2006. Unobtrusive multimodal biometrics for ensuring privacy and information security with personal devices. In *Pervasive Computing*. Springer, 187–201.

[34] Bo Wei, Wen Hu, Mingrui Yang, and Chun Tung Chou. 2015. Radio-based device-free activity recognition with radio frequency interference. In *IPSN' 2015*. ACM, 154–165.

[35] Bo Wei, Mingrui Yang, Yiran Shen, Rajib Rana, Chun Tung Chou, and Wen Hu. 2013. Real-time classification via sparse representation in acoustic sensor networks. In *Sensys' 2013*. ACM, 21.

[36] J Wright, A.Y. Yang, A Ganesh, S.S. Sastry, and Y. Ma. 2009. Robust face recognition via sparse representation. *PAMI* (2009), 210–227.

[37] Zhuoling Xiao, Hongkai Wen, Andrew Markham, and Niki Trigoni. 2014. Lightweight map matching for indoor localisation using conditional random fields. In *IPSN' 2014*. IEEE, 131–142.

[38] Weitao Xu, Chitra Javali, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. 2017. Gait-Key: A Gait-based Shared Secret Key Generation Protocol for Wearable Devices. *ACM Transactions on Sensor Networks (TOSN)* 13 (2017).

[39] Weitao Xu, Guohao Lan, Qi Lin, Sara Khalifa, Neil Bergmann, Mahbub Hassan, and Wen Hu. KEH-Gait: Towards a Mobile Healthcare User Authentication System by Kinetic Energy Harvesting. In *NDSS' 2017*.

[40] Weitao Xu, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. 2016. Walkie-talkie: Motion-assisted automatic key generation for secure on-body device communication. In *IPSN' 2016*. IEEE, 1–12.

[41] Weitao Xu, Yiran Shen, Neil Bergmann, and Wen Hu. 2016. Sensor-assisted Face Recognition System on Smart Glass via Multi-view Sparse Representation Classification. In *IPSN' 2016*. IEEE, 1–12.