2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

Beware of SMOMBIES: Verification of Users based on Activities while Walking

1st Eric Klieme
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
eric.klieme@hpi.de

2nd Christian Tietz
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
christian.tietz@hpi.de

3rd Christoph Meinel
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
christoph.meinel@hpi.de

***Abstract*—Several research evaluated the user's style of walking for the verification of a claimed identity and showed high authentication accuracies in many settings. In this paper we present such a system that successfully verifies a user's identity based on many real world smartphone placements and yet not regarded interactions while walking. Our contribution is the distinction of all considered activities into three distinct subsets and a specific one-class Support Vector Machine per subset. Using sensor data of 30 participants collected in a semi-supervised study approach, we prove that unsupervised verification is possible with very low false-acceptance and false-rejection rates. We furthermore show that these subsets can be distinguished with a high accuracy and demonstrate that this system can be deployed on off-the-shelf smartphones.**

***Index Terms*—gait, authentication, smartphone, activities, verification, behavioral, continuous**

I. INTRODUCTION

Smartphones became a very personal asset over the last years. Some of the most sensible applications on smartphones provide access to web platforms as, for instance, social networks, email, crypto currency wallets, or trading platforms. In addition, a lot of sensitive data are stored on the phone itself, like exchanged text messages or personal photos. To prevent access by malicious users, the devices are protected by personal identification numbers (*pin*), touchgestures, fingerprint readers, or face scans. These protection mechanisms suffer from different vulnerabilities which can be exploited by smudge-attacks [1], shoulder-surfing [2], or face masks [3]. Furthermore, they only provide an *one-off* authentication after which any action can be taken. To tackle these problems, existing research proposed alternative authentication methods over the last years.

Some of these protection mechanisms are part of an authentication method based on *something-you-are* in terms of biometrics that encompass physiological (e.g. fingerprints or iris characteristics) and behavioral biometrics. The latter include the authentication of users based on the way they walk (gait). In addition to a higher difficulty for attackers to imititate this behavior, it helps authenticating the legitimate user in a *continuous* manner. If an attacker steals a phone and walks away, protection is still possible as the system can detect the attacker because of a different style of walking. From another point of view, continuous authentication can enable the smartphone as an authenticator that continuously provides a kind of *(trust) score* for web service authentication [4].

Smartphones are good candidate devices for authentication based on biometrics:

- *Data input*: Smartphones are equipped with many movement sensors, for instance, accelerometer, or gyroscope, and can collect raw data input for biometric classification.
- *Processing power*: Current smartphones are very powerful in terms of CPU or RAM and provide sufficient storage space. They can collect, store, and process raw sensor data for biometric classification on the phone itself.
- *Data availability*: Due to the sensible data stored on smartphones, they became very important companions for users. As a result, users carry their phone along with them almost all day and provide many interaction data.

Over the last years, a lot of research that also covered aspects like sensor positioning [5], different activities apart from walking (e.g. biking or sitting) [6], [7], and the integration of smartwatches as a second factor [8], [9] has been published in that field. They proved high authentication accuracies in user studies of different sizes with traditional machine learning techniques [10], [11] and Neural Networks [12].

A. Motivation

Particularly because smartphones are so convenient and provide access to a lot of services, they are often used actively while walking, too. Browsing the news, reading and typing messages, or recording a voice message are just a few tasks people perform while walking. These people are sometimes referred to as *smartphone-zombies* (SMOMBIES) [13] as any environmental events are rarely recognized anymore and they seem to be controlled by their phones only. While these activities are a health hazard to people and their environment [14] and while this behavior is already considered chargeable in some cities [15], these users define additional robustness requirements to any authentication or *trust* level system using gait biometrics: the device's movement may differ substantially depending on all possible locations and interactions while the user is walking in comparison to simply walking with the phone located in one's front side trousers

2324-9013/18/31.00 ©2018 IEEE
DOI 10.1109/TrustCom/BigDataSE.2018.00096

pocket. These different levels can aggravate a correct authentication and possibly increase ***F****alse-****A****cceptance-****R****ates* (FAR) or ***F****alse-****R****ejection-****R****ates* (FRR) [11], [16]. With a higher FAR, the *security* level decreases and attackers may circumvent a gait authentication system by keeping the phone in their hands while walking, for example. If the FRR is increased, *usability* may worsen as users are detected as attackers during interactions while walking. Depending on a possible implementation, that could trigger traditional fallback mechanisms like passwords or pins *continuously* which would need explicit user interactions every time.

B. Contribution

In this paper we present a verification system that is resistant against different locations and interactions connected to the act of walking. It allows continuous authentication based on a person's gait independent from the location or interaction. Our detailed contributions are as follows:

- We collected sensor data of 30 users in a semi-supervised user-study with a yet unseen approach to simulate real world situations linked to walking. These situations include interactions such as texting or reading and typical placements of devices while walking, like bags or pockets (section III). To our best knowlegde this work is the first evaluation of up to 14 different placements and interactions per person in terms of user verification while walking.
- We prove that unsupervised user verification is possible with a high performance that needs only 15-20 seconds of training time per activity. We reach F1-scores of up to 97% for single activities resulting from three different activity subsets and specific *one-class Support Vector Machines* (SVM) per subset (section V).
- We further illustrate that our subsets can be distinguished with a mean accuracy of 88.82% using SVMs (section V-E).
- We demonstrate that our system can be deployed on off-the-shelf Android devices and performs well with low sampling rates, too (section VI).

Starting with an overview on related work (section II) this paper closes with a conclusion and an outlook regarding future research (section VII).

II. Background

The evaluation of a person's gait has been the topic of research for some years already, wherein it has been analyzed in different contexts, for example, activity detection [17]–[20], sensor positioning [5], smartwatch integrations [8], [9], or health issues such as detecting spoofing attacks aiming for healthcare benefits [21]. In this section we give a more detailed overview on related work in terms of (unsupervised) user authentication based on a user's gait with a focus on real world situations.

In 2014, [22] conducted three user studies to evaluate (unsupervised) user verification by applying a *Gaussian Mixture Model Universal Background Model (GMM-UBM)*. In the first study they trained a supervised walking activity classifier with 47 subjects performing different activities such as walking, biking, or running with the smartphone in different positions. In the second study they collected data to evaluate supervised training with the help of 12 subjects who were carrying the phone in at least two different positions on their body like belts or jacket pockets. Lastly, they collected unlabeled data from 8 subjects over two to three weeks. They reached up to 98% precision for walking detection using a decision tree, an EER of up to 14% for supervised gait verification using 20% of the participants training data, and an accuracy in the unsupervised scenario that is 5% lower than in the supervised scenario. They explicitely note that actions with tight coupling to body motions (trousers pocket) are easier to verify in comparison to loosely coupled ones like holding in hands. In terms of features, they include time and frequency-domain features with a focus on *Compressed Sub-band Cepstral Coefficients (CSCC)* based on the *Mel Frequency Cepstral Coefficients* (MFCC) which are typically applied in speech detection. They extracted these features using windows from extracted accelerometer data with a length of 512 samples and an overlap factor of 50% using 100*Hz* sampling. Finally, they showed that their setup can be deployed on off-the-shelf smartphones with low CPU and RAM usage. In this process, feature extraction and gait analysis are the heaviest operations.

Watanabe and Sara presented an activity and user detection evaluation in 2016 that was based on a former study from 2014 [10]. In this evaluation, 15 participants executed 5 different activities (pocket, pretend to call, pretend to look at device, upstairs, downstairs) each for about a minute in a 50 m corridor. They applied different feature selection methods to determine the most appropriate subset from a total of 52 features which they extracted from nonoverlapping windows of 300 accelerometer sensor events collected with a sampling rate of 100*Hz*. The Random Forest algorithm outperformed Neural Networks, Bayes Net and Support Vector Machines (SVM) provided by Weka Machine Learning suite in all classification scenarios. These scenarios included activity detection (5 activities), user detection (15 subjects), and *user and activity* detection (75 combinations). They reported correct classification rates of up to 90% which they reached in the classification of 5 activities [16].

Reference [11] analyzed 10 different one-class approaches for continuous authentication including K-Nearest-Neighbours (KNN), one-class SVMs, and 3 different distance-based classifiers (Euclidean, Manhattan, Mahalanobis) with different configurations for gait authentication. For evaluation they collected data of 10 participants executing 25 rounds of data collection, each based on 5 placements (right upper arm, right hand, right jacket pocket, right trousers pocket, waist) and 5 activities (ascending and descending stairs, walking, jogging, and jumping). Using *linear acceleration* and gyroscope sensor data recorded with 20*Hz*, they extracted 89 features from time-domain, frequency-domain and wavelet-domain based on a 1 second window with a 50% overlap. They evaluated an *activity-aware* and a *placement-aware* scenario. Overall, Manhattan and Mahalanobis distance classifiers performed better than the other and showed that placement-aware situations are more difficult. For the activity-aware scenario and the placement-aware scenario the best EERs of 2.4% and 5.5% were reached for *jogging* activity and *right side pocket*, respectively.

With *IDNet*, Gadaleta and Rossi present a gait authentication system that uses neural networks as a feature extractor and one-class SVMs (ocSVM) for user recognition. They used a *Convolutional Neural Network* (CNN) to extract feature vectors with 40 dimensions out of determined walking cycles from a data set of 50 participants. For each of the subjects they collected 'front right trousers pocket' walking activity data over a 6-months period. The activity also included different shoes and clothes. They reported that their extracted features outperform other traditional schemes, such as classification trees or KNN by comparing classification accuracies on the test set. In addtion, they stated that even the traditional schemes would perform better with the CNN-based features. Finally, they used an ocSVM to evaluate whether their feature extractor can be used for yet unconsidered test samples in a real-world setting. Further reducing the feature dimensions using *Principal Component Analysis* they showed that a ocSVM performs best with less feature dimensions than the original 40. In the end, they reach false-positive rates and false-negative rates of less than 0.15% based on a multi-stage authentication approach considering multiple subsequenct walking cycles [12].

We will extend the approaches presented above by focussing on interactions occuring while walking that have not yet been considered. Most of them became increasingly common with the rising success of smartphones and apps over the last 3-5 years such as texting or voice message recording. To our best knowledge, no in-depth analysis of user verification exists at the moment which includes different placements and interactions that occur while walking. Moreover, our work focuses on a real world deployment and thus only considers unsupervised learning for the verification. In addition, it involves a deployability evaluation that is left out in most of the other works. Nevertheless, our study does not include data collected over a longer time in comparison to, e.g, [12] or [22].

III. Semi-Supervised Data Collection

With respect to the user's clothes and manually observed interactions, the following activities and phone positions were considered for data collection:

- *7 Interaction activities*: reading a text (portrait), texting (portrait), texting (landscape), watching a video (landscape), listening to a voice message (portrait), recording a voice message (portrait), and answering a call (portrait)
- *9 Side-specific wearing activities (right/left)*: holding in hand if no pockets are available, trousers frontside pocket, trousers backside pocket, trousers knee-level pocket, jacket outer pockets (usually at the bottom end of the jacket), shoulderbag over shoulder, shoulderbag diagonal, handbag over shoulder, and handbag holding in hand
- *3 Wearing activities with central placements*: jacket breast pocket, jacket inner pocket, and backpack

To our best knowledge no public data set with the respective activities included exists. Therefore, we prepared a specific data collection application that covers the different locations and interactions (section III-A). The overall experiment took place in a large parking lot with very little car activities. We explained the purpose of the study to the participants and asked them to walk a speed they consider *normal*. Overall, we collected data from 30 participants (10 female) ranging from 18-42 (σ=6,3) years and heights from 163-201 (σ=9,6) *cm* we recruited on-campus. We used a Google Pixel Smartphone and a Huawei Smartwatch to collect 30 seconds of pure walking per *wearing* or *interaction* activity (table I).

A. Data Collection Application

In addition to the approaches from related work like [16], we wanted participants to actually *interact* instead of only *pretending* to perform a specific activity. Moreover, we wanted to allow users to behave natural and to reduce possible bias by repeated instructions of a supervisor. Thus, we implemented the app to enable the participants to navigate through the collection process on their own with as few interruptions as possible. For implementation we extended the ResearchStack framework [23] to collect raw *accelerometer* and *gyroscope* sensor data in the background. In addition, we created six real-world survey step interfaces to simulate situations for instant messaging (simple chat bot), reading articles (web view of Google research blog), watching videos (lecture video), recording and listening to voice messages (similar to WhatsApp) and telephoning. With regard to the user interface, each possible wearing or interaction survey step is represented by either two or three screens referring to the *instruction*, *execution*, and *closure* phases (Fig. 1).

1) *Instruction*: This screen describes the possible places in which to put the device while walking for *wearing* activities. All side-related locations were recommended based on the initial information in case a person was left-handed or right-handed. For *interaction* activities, such as *reading an article*, a short summary of the *execution* part is given in addition, e.g. "read a text".
2) *Execution* (interaction only): This screen shows the interaction interface, e.g. a chat view to exchange messages or a web view to read articles.

TABLE I
Total quantity of participants for wearing and interaction activities during data collection

	Trousers front right	*Trousers front left*	*Trousers back right*	*Trousers back left*	*Holding in hand right*	*Holding in hand left*	*Jacket outer right*	*Jacket outer left*	*Jacket inner*	*Jacket breast*	*Backpack*	*Reading a text*	*Watching a video*	*Texting Landscape*	*Texting portrait*	*Listening voice message*	*Recording voice message*	*Answering a call*
Total	**19**	**9**	**22**	**4**	**25**	**5**	**24**	**6**	**17**	**9**	**21**	**28**	**30**	**30**	**30**	**30**	**30**	**30**

3) *Closure*: This screen indicates that the required time of activity execution is over. It allows the participants to stop the specific step and proceed to the next one.

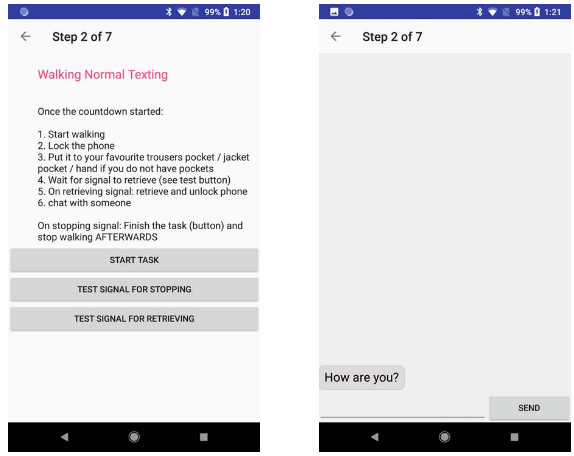
The screens are connected by either two or three different timers depending on the type of activity:

1) *Lock and locate device timer* (5,15 seconds): Within this time the participant should lock the device, start walking, and put it in the place requested on the *instruction* screen. It is started by the participant from the *instruction* screen. For trousers or jacket pockets we decided on a 5 second timer, for handbags, backpacks and shoulderbags the timer spanned 15 seconds.
2) *Retrieve and unlock device timer* (only interaction, 8 seconds): Once the timer passed, a notification sound plays on the smartphone and the smartwatch vibrates to signal the participant that he or she needs to retrieve the device from the respective location he or she put it before, and to unlock it. This second timer runs in the background after the previous one finishes.
3) *Activity Execution timer* (30 seconds): This timer starts after the *lock and locate* timer for *wearing* activities and after the *retrieve and unlock* timer for *interaction* activities, respectively. Once finished, the *closure* screen shows. For all *wearing* activities a notification sound plays again as the device still remains in the respective location and needs to be retrieved and unlocked to finish the step.

Although a more thorough evaluation regarding the participant's perceived level of reality was left out for this study, the approach worked out quite well in our observation in terms of less supervision. After a supervised introductory sample task, all participants navigated themselves through the collection process with only a very few interruptions. These were mostly necessary due to technical problems while only two participants misunderstood the *holding in hand* task. Please note that a supervisor observed the collection from a distance of 10-50 meters for the whole time and took notes or interrupted in case of any problem. To our best knowledge, this semi-supervised approach is yet unseen. Although [20] used an app that guided participants through the collection process in a similar manner, their study was conducted with the help of a threadmill and fewer activities.

B. Sampling Configuration

To access sensor data on Android for specific applications, a respective *sensor delay* needs to be defined. This setting determines the approximate delay or sampling rate with which the sensor data are available for that application. The final delay is, furthermore, only a *suggestion* for the system and can be changed by Android depending on available CPU processing power or battery level, for instance [24]. For this study we decided to collect the data with the lowest delay possible which results in a sampling frequency of approximately 400Hz on our device.



(a) Basic instructions for the *texting* activity. Participants could test all signals for that task prior to the start.

(b) Chat view to exchange messages with a simple chat bot that asks a new question after each response

Fig. 1. Screen flow of the *execution* phase and the *closure* phase of the *texting* activity.

C. Data Preprocessing

The following processing steps were taken prior to our evaluation:

- *Sliding window*: With respect to the above presented work, we decided to apply a sliding window approach for feature extraction. After manual evaluations, we chose a window length of 2 seconds that is extracted every 200 milliseconds resulting in an overlay factor of 90%. To our best knowledge, this is a novel approach with regards to related work mentioned before.
- *Activity time extraction*: As stated above, we collected 30 seconds of each *wearing* or *interaction* activity. To have raw activity time with no interference from a delayed placement or access of the device, we skipped the initial 8 seconds of the collected data. Based on our sliding window specifics, we extracted 100 windows of 2 seconds out of the last 21.8 seconds of each activity. We extracted this data based on the end of the *activity execution timer*.
- *Filter*: We applied a savgol filter of a window length of 21 with a polyorder of 3 to minimize hardware errors and noise in the raw sensor data.
- *Feature extraction*: We extracted 18 features in the time domain per sensor dimension (*X, Y, Z-dimension* per sensor respectively) resulting in 108 features in total per window. Per dimension, we extracted the 11 bin edges after applying equal-width binning for 10 bins, the peak-to-peak difference based on the first and last edge (min, max), variance, standard deviation, median, mean, average median deviation, and average mean deviation.

IV. Evaluation Approach

A. Detecting the legitimate User

We considered the following process to evaluate whether the *legitimate* user is successfully detected in which we represented situations where the *legitimate* user is fully enrolled and performs one of the abovementioned activities while walking:

1) **Choose** support activities for training: We considered the first 75% of extracted feature sets from all activities by a single participant as given by an assumed enrollment. For training, we initially left out the data for a particular activity we wanted to evaluate in terms of verification performance later on. To give an example, we included 75 feature sets from all wearing activities and interaction activities performed by a participant for training except *answering a call*. We refer to *answering a call* as the *target activity*.
2) **Train** and test *target activity* verification: The first 75% of the target activity feature sets were taken as further input for training. The yet unseen remaining 25% of the target activity data were used to test. For further verification, we cross-validated by repeating this process for all 75%/25% splits of the available 100 feature sets of the target activity. For a realistic validation situation we only took 25% shares of feature sets that were created from subsequent windows, whereas the 75% for training were taken from before and after the test data. In terms of 100 available feature sets per activity we repeated training and test thus 76 times and conducted $76 \times 25 = 1900$ tests of our model. For each test, the training set consisted of 75 feature sets of the current target activity training/test split and all training feature sets of the support activities.
3) **Repeat** for other activities: We repeated this process for each other recorded activity by the specific participant and for each other participant, respectively.

B. Detecting nonlegitimate Users

We followed a similar procedure to verify whether nonlegitimate users are detected. As a result, we represented situations where the *legitimate* user is fully enrolled and a *nonlegitimate* user gained access to the phone and started walking away.

1) **Train** model based on *legitimate* user: We picked one participant as the *legitimate* user and trained the model with 100% of all available feature sets extracted from any recorded activity of that participant. To give an example, 11 activities were used as a *training profile* for a participant that recorded all 7 *interaction* activities and 4 *wearing* activities (trousers front/back, holding in hand, jacket outer pocket). Overall, training was done with $11 \times 100 = 1100$ feature sets.
2) **Test** model: We considered a threat model with an attacker who put no effort in observing the user's favorite placement side or style of walking and could be categorized as a *random attack* [25]. Nevertheless, the attacker observed the favorite pockets of the legitimate user. Because of that, we tested the model with all feature sets extracted from activities recorded by all other participants that fit the activities of the training profile. Furthermore, we ignored any side-specific information in case a participant was right-handed or left-handed.
3) **Repeat** for other participants: We repeated this process so that each participant was the *legitimate* user once and a random *attacker* for all other participants, respectively.

C. One-Class Support Vector Machines

We used *one-class Support Vector Machines* (ocSVM) with an *rbf* kernel for all training and testing operations. Referring to related work like [11], [12] or [26], (one-class) SVMs with *rbf* kernel show a good performance in different settings. Moreover, implementations like LIBSVM are given for a vast amount of environments [27], which is important for a later real-world deployment. In general, *Support Vector Machines* (SVM) determine a hyperplane that separates vectors that represent feature sets of different target classes with a maximum margin in a *linear* space. If SVMs are used with a specific *kernel* they map these vectors into higher dimensional spaces and thus also can find nonlinear relations to determine the optimal hyperplane. Prior to our application, any training or test feature set was standardized accordingly using a *Min-Max Scaling* procedure based on the training data.

D. Sensor-based ocSVMs and Majority-Voting

Depending on the specific activity, the device's orientation may be more important than the overall acceleration applied to a device, especially in terms of *interaction* activities, such as texting or calling. For the evaluation we thus included four different decision schemes to cover these characteristics:

1) *Accelerometer-only* ocSVM: This ocSVM only considers training and test feature sets extracted from *accelerometer sensor data* (54 features).
2) *Gyroscope-only* ocSVM: This ocSVM only considers training and test feature sets extracted from *gyroscope sensor data* (54 features).
3) *Hybrid-sensor* ocSVM: This ocSVM considers training and test feature sets with the full feature set as described in section III-C extracted from *accelerometer* and *gyroscope sensor data* (108 feature vectors).
4) *Majority Voting on decision level*: Based on the decision of each separate ocSVM majority voting was applied. If both the *accelerometer-only* and the *gyroscope-only* ocSVM agree in their decision this did overwrite any decision of the *hybrid-sensor* ocSVM.

E. Activity-aware User Verification

As the presented work by [11] showed, specific evaluation scenarios that are either *placement*- or *activity*-aware can lead to different results. Furthermore, [22] mentions that the coupling of the device to the overall body motion influences the verification performance, too. This is especially the case when a user performs activities with *different* levels of coupling. In addition to considering all recorded activities as a whole, we also evaluated whether a split into specific activity subsets can increase verification performance. Based on our observations during the study and the mentioned related work, we focussed on a split resulting from the *the device's level of coupling to body motion* per activity.

V. RESULTS

A. Verification based on all available Activities

Overall, our results are promising including all recorded activities with all decision schemes as depicted in Fig. 2. In the end, the *accelerometer-only* approach slightly outperformed the *hybrid* and the *majority-voting*-based approach with a focus on a most balanced result that gains an overall FAR of 4.47% and FRR of 7.17%. For each decision scheme we used a coarse grid search and manual tuning to find the optimal hyper-parameters ν and γ that can be used to tune ocSVMs.

A different situation is observed for all decision schemes in terms of per-activity performance. Please refer to table II for per-activity performances based on the results of the *accelerometer-only* ocSVM. Activities #1-#11 perform quite well regarding protection against nonlegitimates with single per-activity FARs of 0.0%. At the same time, *holding in hand right* and *trousers front/back left* activities show high FRRs of up to 15.67%. Regarding *interaction activities*, *reading* and *texting portrait* activities demonstrate high FARs up to 15 % while having acceptable FRR rates of less than 5%. Any further tuning effort resulted in either lower FRRs for tasks #1-#11 and higher FARs for all remaining *interaction* activities or vice-versa. In conclusion, our results fit the observations made by [22] in terms of different levels of device-body movement coupling and can be extended to our set of activities while walking.

TABLE II
FAR, FRR, PRECISION, RECALL, AND F1-SCORE PER ACTIVITY FOR ν=0.01 AND γ=0.25 FOR ACCELEROMETER-ONLY OCSVM

#	*Activity*	**Precision**	**Recall**	**F1-Score**	**FAR**	**FRR**
1	Trousers front right	99.84	90.33	94.85	0.10	9.67
2	Trousers front left	99.98	89.25	94.31	0.01	10.75
3	Trousers back right	99.99	92.47	96.09	0.00	7.53
4	Trousers back left	100.00	89.12	94.25	0.00	10.88
5	Jacket outer right	99.89	93.65	96.67	0.07	6.35
6	Jacket outer left	99.99	90.59	95.06	0.01	9.41
7	Holding in hand right	97.75	84.33	90.54	1.27	15.67
8	Holding in hand left	98.97	86.78	92.47	0.59	13.22
9	Jacket inner	99.82	95.59	97.66	0.20	4.41
10	Jacket breast	99.99	92.86	96.29	0.02	7.14
11	Backpack	99.98	93.44	96.60	0.02	6.56
12	Texting portrait	81.17	98.56	89.02	14.98	1.44
13	Texting landscape	86.84	95.32	90.88	9.46	4.68
14	Reading article	85.01	98.41	91.22	12.21	1.59
15	Watching video	88.37	96.74	92.37	8.34	3.26
16	Answering a call	97.83	91.09	94.34	1.33	8.91
17	Listening to voice message	95.17	90.55	92.80	3.01	9.45
18	Recording voice message	89.94	94.47	92.15	6.92	5.53

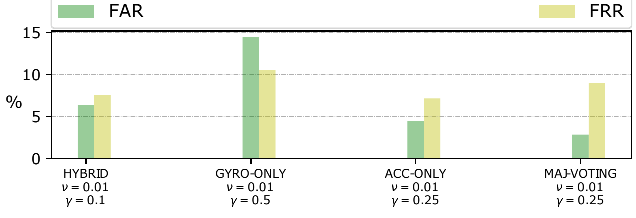


Fig. 2. *FAR* and *FRR* per decision scheme including all activities based on best hyper-parameter settings per decision scheme

B. Verification based on Coupling to Body Motion

We improved our system by splitting all activities into three subsets with specific ocSVMs as shown in table III, IV and table V, respectively. Each subset includes activities with a certain level of coupling between the device's movement and the body motion and the following three-fold split outperformed a two-fold split based on *wearing* and *interaction* activities only:

- *Screen attention activities*: For these activities users need to absorb all body movements in order to still be able to follow anything that happens on the screen, such as writing and reading texts, or watching videos.
- *Speech-related activities*: These activities encompass *recording of* and *listening to* voice messages, and *answering a call*. Although users still need to absorb body motion to a certain level they absorb less motion in comparison to *screen attention* activities. In the end, they only need to talk or listen and therefore they do not need full screen attention all the time.
- *Wearing only activities*: These activities have a tight coupling to body motion and include all of our *wearing* activities defined in section III.

Although the FARs for *reading an article* and *texting portrait* improved only a little, we reached an overall more balanced performance using the *accelerometer-only* ocSVM again. While the FRR of *speech-related* activities increases, some of the remaining *wearing only* activities' performance could be improved significantly. The FRR could be reduced by approximately 6% from 15.67% to 9.63% for the *holding in right hand* activity, for instance. Simultaneously, the performance of the other included activities remained on a high level reaching F1-scores of up to 97.71% for the *jacket inner pocket* activity.

In conclusion, we consider the three subsets of *wearing only*, *screen attention*, and *speech-related* as the optimal split for this evaluation:

- We consider the *wearing only* activities to be the most important group as we assume that the majority of people (and attackers) still merely walk in most of the cases.

TABLE III
FAR, FRR, PRECISION, RECALL, AND F1-SCORE PER SCREEN ATTENTION ACTIVITY FOR ν=0.01 AND γ=0.02 FOR ACCELEROMETER-ONLY OCSVM

#	*Activity*	**Precision**	**Recall**	**F1-Score**	**FAR**	**FRR**
1	Texting portrait	87.33	90.75	89.00	8.63	9.25
2	Texting landscape	87.32	93.20	90.16	8.87	6.80
3	Reading article	89.40	93.69	91.50	7.82	6.31
4	Watching video	89.91	92.48	91.18	6.80	7.52

TABLE IV
FAR, FRR, PRECISION, RECALL, AND F1-SCORE PER SPEECH-RELATED ACTIVITY FOR ν=0.01 AND γ=0.03 FOR ACCELEROMETER-ONLY OCSVM

#	*Activity*	**Precision**	**Recall**	**F1-Score**	**FAR**	**FRR**
2	Answering a call	94.08	89.86	91.92	3.70	10.14
1	Listening to voice message	93.89	89.40	91.59	3.81	10.60
3	Recording voice message	88.72	90.94	89.81	7.58	9.06

TABLE V
FAR, FRR, PRECISION, RECALL, AND F1-SCORE PER WEARING ONLY ACTIVITY FOR ν=0.01 AND γ=0.03 FOR ACCELEROMETER-ONLY OCSVM

#	*Activity*	**Precision**	**Recall**	**F1-Score**	**FAR**	**FRR**
1	Trousers front right	92.14	93.10	92.62	5.59	6.90
2	Trousers front left	93.68	95.33	94.50	4.53	4.67
3	Trousers back right	91.30	95.08	93.15	6.88	4.92
4	Trousers back left	89.73	95.54	92.54	8.31	4.46
5	Jacket outer right	94.19	96.36	95.26	4.03	3.64
6	Jacket outer left	98.93	97.03	97.97	0.71	2.97
7	Holding in hand right	92.52	90.37	91.43	4.79	9.63
8	Holding in hand left	98.77	92.18	95.36	0.75	7.82
9	Jacket inner	97.93	97.50	97.71	2.45	2.50
10	Jacket breast	98.73	94.97	96.81	3.32	5.03
11	Backpack	98.49	94.94	96.69	1.38	5.06

- All *screen attention* activities could be supported by a second factor in a later real-world deployment, like keystroke/touchstroke authentication approaches [28], or eye-movement-based authentication systems [29].
- All activities of the *speech-related* activity set except *listening to a voice message* could be supported by a second factor in a later real-world deployment like voice/speech recognition technologies [30].

Please note that depending on the overall focus of the verification system different activity subset splits may be sufficient.

C. Wrong Decisions in a Row

As explained in section III-C we used a sliding window approach that results in 5 decisions per second. We evaluated the number of wrong decisions in a row to gain further insights on all cases in which decisions were made wrong. In general, the mean length is sufficiently low with values that represent approximately *1* second (*3-6* wrong decisions) for both groups of users. Nevertheless, there exist single outliers which nearly correspond to the respective test window length referring to the legitimate users (*24* decisions) and up to *97* decisions for nonlegitimates in all subsets as shown in table VI.

D. Majority Voting on the Decision Level

Based on the row lengths presented above, we evaluated a majority voting scheme in addition to the four decision schemes described in section IV-D. With regard to section IV, multiple test sets of 25 decisions are considered to verify the legitimate user. Thus, we included only 7 to 17 *subsequent* decisions for majority voting to get *a)* a sufficient amount of majority voting results per test set and *b)* a realistic situation in terms of extracted walking data. Majority voting based on 17 decisions results in 9 *majority decisions* as the respective window of 17 decisions could be shifted 8 times based on 25 ocSVM decisions, for example.

Fig. 3 shows the performance of majority voting applied to the results of the *accelerometer-only* ocSVM for *speech-related* activities. We can see that precision values worsen

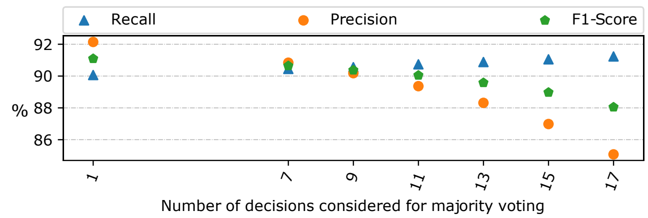


Fig. 3. Performance depending on different amounts of decisions considered for majority voting for *speech-related* activities based on the *accelerometer-only* ocSVM with ν*=0.01* and γ*=0.03*

a little more (from 92.14% to 85.07%) than the recall is improved (from 90.06% to 91.23%) with larger window sizes. This results in a slightly decreasing F1-Score from 91.09% to 88.04%. Apart from these metrics, FAR and FRR slightly improve from 5.03% / 9.93% to 4.49% / 8.76% in terms of *1* (no majority voting) and *17* included decisions.

We further evaluated a larger number of considered decisions for the detection approach of *nonlegitimate* users as these test sets included 100 decisions. Overall, a slight improvement of the accuracy is reached from 95.54% to 96.49% based on 1 to 51 considered decisions. We observed similar results for the other activity subsets for both groups of considered decisions as shown in table VII and VIII, respectively. In conclusion, majority voting can increase FAR and FRR by up to 2% for *wearing only* activities based on only a few seconds of consequent walking activity (17 decisions ≈ 3,4 seconds). In terms of nonlegitimate users only, the detection accuracy can be improved by up to approximately 3% in the case of *screen*

TABLE VI
MEAN, STANDARD DEVIATION, AND MAXIUM OUTLIER FOR THE LENGTHS OF WRONG DECISIONS IN A ROW BASED ON ACTIVITY SUBSET FOR LEGITIMATE AND NONLEGITIMATE USERS

	wearing only	**screen attention**	**speech-related**
$\bar{x}$ legitimate	3.05	3.8	3.91
σ legitimate	3.44	3.96	4.32
max outlier legitimate	24	24	24
$\bar{x}$ nonlegitimate	5.09	4.06	5.68
σ nonlegitimate	8.71	5.98	8.44
max outlier nonlegitimate	97	85	88

TABLE VII
PERFORMANCE RESULTS OF A MAJORITY-VOTING APPROACH THAT CONSIDERS DIFFERENT AMOUNTS OF SUBSEQUENT DECISIONS FOR WEARING ONLY AND SCREEN ATTENTION SUBSET

Activity Set	**Decisions**	**Precision**	**Recall**	**F1-Score**	**FAR**	**FRR**
Wearing only	1	88.46	92.51	90.44	8.04	7.49
Wearing only	17	80.58	94.15	86.84	6.48	5.85
Screen attention	1	94.63	92.69	93.65	3.85	6.35
Screen attention	17	89.49	94.15	91.76	3.47	5.85

TABLE VIII
DETECTION ACCURACIES OF NONLEGITIMATE USERS BASED ON DIFFERENT AMOUNTS OF SUBSEQUENT DECISIONS CONSIDERED FOR MAJORITY-VOTING

	Decisions considered					
Activity Set	**1**	**7**	**15**	**25**	**35**	**51**
Wearing only	94.97	95.17	95.43	95.70	95.92	96.06
Screen attention	91.96	92.54	93.32	93.99	94.44	94.84

attention activities if 51 decisions are considered for majority voting.

E. Activity Subset Detection

In a real-world scenario the proposed subset-based verification system needs to determine which activity is currently performed to decide about the sufficient subset classifier. We further evaluated SVMs with an *rbf*-kernel, Random Forests (RF) and Decision Trees (DT) for supervised detection of our activity subsets including all decision schemes. Therefore, we performed a 10-fold cross validation based on 30 participants and repeated it 10 times (100 runs in total).

Since the *speech-related* activity subset includes only three activities, this class had the lowest amount of available training/test samples. Thus, we randomly removed samples from the other classes to gain an equal amount of samples per class. We conducted a grid search to find the optimal hyper-parameter for the SVM but did not consider additional feature selection approaches for RF and DT. Overall, we reached the best mean accuracy of 88.82% using the SVM *majority-voting* scheme (C=1, γ=0.1). In general, *majority-voting* performed slightly better (1%-2%) than the *hybrid* and *accelerometer-only* scheme while *gyroscope-only* performed worst for all classification approaches. In comparison, the SVM performed a little better than the Random Forest (*majority-voting* 87.16%), and clearly outperformed the Decision Tree (*majority-voting* 83.69%) classification scheme.

Regarding the confusion matrix shown in table IX, the distinction between *speech-related* and *screen attention* activities is the most difficult as a lot of *speech-related* activities were classified as *screen attention* activities. Additionally,

TABLE IX
CONFUSION MATRIX FOR SUPERVISED ACTIVITY SUBSET DETECTION BASED ON SVM MAJORITY-VOTING AFTER 100 CROSS VALIDATIONS

	wearing only	**screen attention**	**speech-related**
wearing only	83121	1428	5451
screen attention	602	85266	4132
speech-related	2778	15804	71418

distinguishing *speech-related* activities from *wearing only* activities is more difficult than to distinguish them from *screen attention* activities. Referring to our data collection and the presented results of our study we assume that they are due to the following reasons:

1) Users do not need to absorb body movement while listening or speaking, because they do not need to see what is happening on the screen. Thus, the device's movements are not decoupled as heavyly as they are during *screen attention* activities. They may also be more similar to *wearing activities*.
2) Some of the participants recorded one long message while others recorded multiple short messages with some short *walking only* activity in between. Moreover, some participants talked rather loudly while the device was in approximately the same position as *reading an article* in front of the body. Another share of the participants talked with a normal conversation volume but made sure that the microphone pointed to their mouth. As the participants had to keep their finger on the recording button all the time while speaking (similar to WhatsApp), only two participants recorded voice messages similar to *answering a call*.
3) One group of participants listened to the voice message just like they would answer a call. Another group of participants changed the location of the device to listen properly: While some of them just increased the volume to keep the device in approximately the same position as *reading an article*, others moved the device next to their head and made sure that the speaker pointed to their left or right ear.

VI. Deployability Evaluation

As stated in section III-B we recorded sensor data with the lowest delay possible. To determine *deployability* [31], we evaluated *verification performance* related to different sampling rates and the on-device *processing complexity* of our approach.

- *Verification performance*: We simulated lower sampling rates by considering only every n-th sample of the data we collected initially. Although this downsampling approach does not consider possible aliasing effects, it is sufficient enough as our features are extracted solely from the *time domain*. Our target sampling rates focused hereby on the different built-in sensor delay settings from Android.
- *Processing complexity*: We implemented our approach in a prototype app that we deployed on our test device. We fully charged the phone and let the system run for one hour without any further active apps except the profiling app TrepN [32]. While this approach does not consider any influence from real world phone interaction while walking, it is sufficient for a basic understanding of *wearing only* activities.

A. Sampling Rate vs. Accuracy

To determine the default sampling rates for our test device we measured sampling rates based on the built-in sensor delay settings provided by Android resulting in approximately 400 Hz (fastest), 50 hz (game delay), 25 Hz (UI delay), and 6 Hz (normal delay). Thus, we simulated lower sampling rates of approximately 200, 100, 50, 25 and 6 Hz by considering only every 2^{nd}, 4^{th}, 8^{th}, 16^{th} and 64^{th} sample of our initially collected data. Please refer to Fig. 4 to see the best performances we could reach for the *screen attention* activity set with a focus on FAR and FRR.

On the one hand, we proved our verification system to be stable against lower sampling rates as we reached sufficient performances even with only 6 Hz simulated sampling frequency. On the other hand, hyper-parameter settings and the decision scheme needed to be changed with a decreasing sampling rate to still reach a good performance. While the accelerometer still outperforms the other schemes in most frequencies, the *majority-voting* approach provided the most balanced set for 6 *Hz*. Overall, FRR and FRR provided the best performance with a simulated rate of 25 *Hz* but the most balanced with 200 *Hz* and 400 *Hz*. Although FAR and FRR are still sufficient low with 6 *Hz*, they drop approximately 4% in comparison to 25 *Hz*. With only 6 measurements per second not all walking specifics may be covered anymore. We thus consider a sampling rate of 25 *Hz* as the best trade-off between sampling rate and performance. For further verification, we successfully cross-validated the 25 *Hz* setting by shifting the reference for every 16^{th} sample by 0-15 times prior to the feature extraction. Overall, we got a mean FAR and FRR of 5,66% (σ=0.07) and 6,87% (σ=0.15), respectively. For the *speech-related* and *wearing only* subset we reached sufficient FARs / FRRs for 25 *Hz* of 5,57% / 10.39% and 3,58% / 4,98% (accelerometer-only with ν=0.01, γ=0.05 for both), too.

The activity subset detection performed best with a mean accuracy of up to 88.43% based on the *hybrid* decision scheme using a SVM and slightly outperforms the *accelerometer-only* and *majority voting* scheme (1%-2%) again.

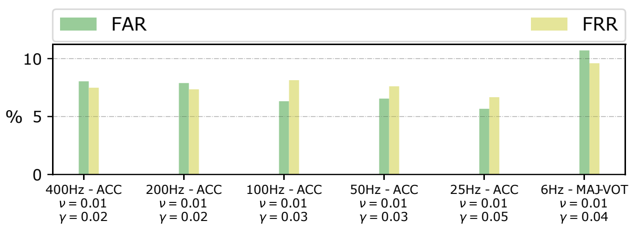


Fig. 4. *FAR* and *FRR* per simulated *frequency* for *screen attention* activity set based on best hyper-parameter settings per frequency

B. Processing Complexity

For complexity evaluations we used an open source LIBSVM [27] integration for Android [33] and built a prototype app based on our presented work. Since activity subset detection performance was only slightly worse with the *accelerometer-only* approach, we focused on that approach for activity subset detection and the final decision. Please refer to table X which summarizes the mean times needed for the different operations after a one hour test run based on 15077 decisions. On our test device, a verification process was triggered approximately every 238 *ms* based on 25 *Hz* sampling frequency and a threshold of 200 *ms* between decisions. To simulate a previous enrollment, we used an activity subset detection model trained based on 29 users. For the different activity subsets we picked a participant that contributed 5 wearing activities, 4 screen attention activities and 3 speech-related activities. The battery was drained by 4% after one hour and TrepN reported a mean per-CPU workload of 457 *MHz*, 467 *MHz*, 399 *MHz*, and 398 *MHz* and a mean RAM usage of 367 *MB*, respectively. Based on the measured times and the profiling results, we argue that our approach is deployable on off-the-shelf Android devices. In the end, decisions needed an average time-span of approximately 30 *ms* for all subsets including *feature extraction*, *subset determination*, and the final *subset-specific* scaling and decision process.

TABLE X
PERFORMANCE OF SUBSET BASED VERIFICATION SYSTEM AFTER A ONE HOUR TEST RUN WITH 25 *Hz*

Component	**Mean (*ms*)**	**Min (*ms*)**	**Max (*ms*)**	**σ (*ms*)**
Feature Extraction	10,45	1,62	79,54	1,73
Scaling *Subset Detection*	8,11	1,13	54,43	3,1
Prediction *Subset Detection*	2,56	0,3	36,09	1,73
Scaling *Screen Subset*	6,45	1,04	37,47	2,43
Prediction *Screen Subset*	2,41	0,34	24,59	1,43
Scaling *Wearing Subset*	6,31	1,03	34,96	2,13
Prediction *Wearing Subset*	2,31	0,36	27,58	1,41
Scaling *Speech Subset*	6,31	1,05	36,14	2,06
Prediction *Speech Subset*	2,32	0,32	22,54	1,43

VII. CONCLUSION AND FUTURE WORK

Based on a user study with 30 participants, we showed that unsupervised verification in a privacy-preserving scenario is possible using *one-class Support Vector Machines* (ocSVM) based on 15-20 seconds of walking in different real-world situations. These include activities like *reading a text* on the device while walking or placing it in different pockets. We reached promising results with FARs and FRRs of less than 10% for most activities. The main contribution of this work is the distinction of all possible activities into several activity subsets and the determination of a customized classifier per subset. Each subset represents a different level of coupling between the device and the body movement which resulted in *wearing only*, *screen attention* and *speech-related* activity subsets. For each subset we determined the best out of four decision schemes including three different ocSVMs and a *majority voting* scheme. Overall, *accelerometer-only* ocSVMs slightly outperform the *majority voting* and *hybrid* approach in terms of a most balanced FAR to FRR ratio for all activity subsets. We further showed that majority voting can improve verification performance if subsequent decisions over a longer time frame are considered. Nevertheless, a nonlegitimate user was not detected in single test cases.

We further proved that the three proposed subsets can be distinguished with a sufficient mean accuracy of 88.82% based on a SVM using the *majority voting* decision scheme. As a remaining challenge, *speech-related* activities such as *recording* or *listening to voice messages* are the most difficult activities to classify as our participants applied many different strategies and mixed several locations during the data collection. Finally, we evaluated *deployability* using the example of *screen attention activities* and demonstrated that our system performs well with lower sampling rates down to 6*Hz*. Moreover, we showed that an *accelerometer-only* approach can be deployed on off-the-shelf Android devices with an acceptable battery drain and low processing complexity based on a 25 *Hz* sampling rate. In terms of data collection we followed a novel approach with the help of a special survey-like application that simulated real-world interactions like *instant messaging* and enabled participants to collect data mostly by themselves under less supervision.

In future research, we will include smartwatch data and evaluate an improvement of the overall verification performance. Future studies will focus specifically on power users referring to possible activities (listening/recording voice messages) and placements (handbags, shoulderbags) while walking, but also include other border cases such as injuries, different types of ground or shoes. Furthermore, the proposed mechanism has to be evaluated regarding its deployability with respect to energy consumption and required processing power in a realistic usage over a longer time. The activity subset detection needs further improvement based on the various related work in that field, too. On the one hand, it is very important to choose the right subset verification classifier. On the other hand, a sufficient activity detection is needed for a proper enrollment. Due to the amount of different activities included, research for an enrollment approach with a high usability needs to be conducted, too.

References

[1] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge Attacks on Smartphone Touch Screens." *Woot*, vol. 10, pp. 1–7, 2010.

[2] M. Eiband, M. Khamis, E. von Zezschwitz, H. Hussmann, and F. Alt, "Understanding Shoulder Surfing in the Wild: Stories from Users and Observers." ACM Press, 2017, pp. 4254–4265.

[3] "Bkavs new mask beats Face ID in "twin way": Severity level raised, do not use Face ID in business transactions," 2017, accessed: 2018-03-09. [Online]. Available: http://www.bkav.com/d/top-news/-/view_content/content/103968/bkav%EF%BF%BDs-new-mask-beats-face-id-in-twin-way-severity-level-raised-do-not-use-face-id-in-business-transactions

[4] A. Ceccarelli, L. Montecchi, F. Brancati, P. Lollini, A. Marguglio, and A. Bondavalli, "Continuous and Transparent User Identity Verification for Secure Internet Services," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 3, pp. 270–283, May 2015.

[5] D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "An unsupervised approach for automatic activity recognition based on hidden Markov model regression," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 829–835, 2013.

[6] M. Altini, R. Vullers, C. Van Hoof, M. van Dort, and O. Amft, "Self-calibration of walking speed estimations using smartphone sensors," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*. IEEE, 2014, pp. 10–18.

[7] S. A. Antos, M. V. Albert, and K. P. Kording, "Hand, belt, pocket or bag: Practical activity tracking with mobile phones," *Journal of Neuroscience Methods*, vol. 231, pp. 22–30, Jul. 2014.

[8] W.-H. Lee and R. Lee, "Implicit Sensor-based Authentication of Smartphone Users with Smartwatch." ACM Press, 2016, pp. 1–8.

[9] Y. Zeng, A. Pande, J. Zhu, and P. Mohapatra, "WearIA: Wearable device implicit authentication based on activity information," in *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017 IEEE 18th International Symposium on*. IEEE, 2017, pp. 1–9.

[10] Y. Watanabe, "Influence of Holding Smart Phone for Acceleration-Based Gait Authentication." IEEE, Sep. 2014, pp. 30–33.

[11] Y. Chen, C. Shen, Z. Wang, and T. Yu, "Modeling interactive sensor-behavior with smartphones for implicit and active user authentication," in *Identity, Security and Behavior Analysis (ISBA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[12] M. Gadaleta and M. Rossi, "Idnet: Smartphone-based gait recognition with convolutional neural networks," *CoRR*, vol. abs/1606.03238, 2016.

[13] "Smartphone zombies are taking over our pavements. Am I the only person who thinks that's a good thing? | The Independent," 2016, accessed: 2018-03-09. [Online]. Available: http://www.independent.co.uk/voices/smartphone-zombies-are-taking-over-our-pavements-am-i-the-only-person-who-thinks-thats-a-good-thing-a6887551.html

[14] J. Mwakalonge, S. Siuhi, and J. White, "Distracted walking: Examining the extent to pedestrian safety problems," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 2, no. 5, pp. 327–337, Oct. 2015.

[15] "Honolulu now fines people up to $99 for texting while crossing road | Technology | The Guardian," 2017, accessed: 2018-03-09. [Online]. Available: https://www.theguardian.com/technology/2017/oct/25/honolulu-fines-people-for-texting-while-crossing-road

[16] Y. Watanabe and S. Sara, "Toward an Immunity-based Gait Recognition on Smart Phone: A Study of Feature Selection and Walking State Classification," *Procedia Computer Science*, vol. 96, pp. 1790–1800, 2016.

[17] M. A. Ayu, S. A. Ismail, T. Mantoro, and A. F. A. Matin, "Real-time activity recognition in mobile phones based on its accelerometer data," in *Informatics and Computing (ICIC), International Conference on*. IEEE, 2016, pp. 292–297.

[18] H. Guo, L. Chen, G. Chen, and M. Lv, "An Interpretable Orientation and Placement Invariant Approach for Smartphone Based Activity Recognition." IEEE, Aug. 2015, pp. 143–150.

[19] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones." ACM Press, 2013, p. 225.

[20] H. Martn, A. M. Bernardos, J. Iglesias, and J. R. Casar, "Activity logging using lightweight classification techniques in mobile devices," *Personal and Ubiquitous Computing*, vol. 17, no. 4, pp. 675–695, Apr. 2013.

[21] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "Smartphone based user verification leveraging gait recognition for mobile healthcare systems," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*. IEEE, 2013, pp. 149–157.

[22] H. Lu, J. Huang, T. Saha, and L. Nachman, "Unobtrusive gait verification for mobile phones," in *Proceedings of the 2014 ACM international symposium on wearable computers*. ACM, 2014, pp. 91–98.

[23] "ResearchStack," 2018, accessed: 2018-03-09. [Online]. Available: http://researchstack.org/

[24] "Sensors Overview | Android Developers," 2018, accessed: 2018-03-09. [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_overview.html

[25] A. Buriro, Z. Akhtar, B. Crispo, and S. Gupta, "Mobile biometrics: Towards a comprehensive evaluation methodology." IEEE, Oct. 2017, pp. 1–6.

[26] R. Kumar, P. P. Kundu, and V. V. Phoha, "Continuous Authentication Using One-class Classifiers and their Fusion," *arXiv:1710.11075 [cs]*, Oct. 2017, arXiv: 1710.11075.

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[28] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 1, pp. 136 –148, 2013.

[29] C. Song, A. Wang, K. Ren, and W. Xu, "EyeVeri: A secure and usable approach for smartphone user authentication." IEEE, Apr. 2016, pp. 1–9.

[30] W. Shi, J. Yang, Yifei Jiang, Feng Yang, and Yingen Xiong, "SenGuard: Passive user identification on smartphones using multiple sensors." IEEE, Oct. 2011, pp. 141–148.

[31] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano, "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes." IEEE, May 2012, pp. 553–567.

[32] "Trepn Profiler - Apps on Google Play," 2018, , Accessed: 2018-03-19. [Online]. Available: https://play.google.com/store/apps/details?id=com.quicinc.trepn

[33] Y.-C. Tung, "AndroidLibSVM: The well-known LibSVM on Android," Mar. 2018, original-date: 2015-08-31T15:34:20Z. [Online]. Available: https://github.com/yctung/AndroidLibSVM