Smartwatch User Identification as a Means of Authentication

Scott Davidson, Derrick Smith, Chen Yang, and Sebastian Cheah
Department of Computer Science and Engineering
University of California San Diego
La Jolla, California 92093

***Abstract*—User provided passwords are a sources of insecurity in existing computer systems due to a compromise between security and ease of use. Multi-factor authentication attempts to strengthen this weakness, but users fail to implement it due to inconvenience. One possible solution is to use gestures as a form of authentication. We show that it is possible to distinguish between gestures as well as distinguishing between users performing the same gesture. Thus, a gesture based second level authentication scheme is a viable solution.**

1. Introduction

The community has long known that user provided passwords are a major source of insecurity in computer systems [13]. This source stems from the fact that users tend to prefer ease of use over security. This desire for ease leads to weak and insecure passwords that can be easily guessed or brute forced. One promising solution to this problem is the use of multi-factor authentication. However, current multi-factor authentication schemes fail to adequately account for ease of use. For example, Google authentication requires users to type in a six digit one time code every time they log into a new or untrusted device. At first glance this seems like a fair trade-off of ease of use for security, but anecdotal evidence suggests that everyday users do not use this authentication scheme. We believe that this scheme suffers from the same drawbacks that cause passwords to be insecure.

Recent work has suggested that gait can be used as a form of second level authentication [1], [2], [12]. This work traditionally deals with sensors being placed on the body [12] or with the use of mobile phone sensors [10] to build a gait template. Yet, as indicated by [1], these approaches suffer from an error rate that is unacceptable for use in the wild. With the invention of smart watches, the error rate of accelerometer based schemes had decreased significantly [1]. However, the signals used to build movement based schemes are still too unreliable for general use.

We present a study that shows that differentiating between gestures is possible with similar or better error rates when compared to gait based schemes. We also show that it is possible to distinguish between users performing the same gesture with similar, if not better, error rates than gait based schemes. With these results we postulate that a gesture based second level authentication scheme is not only viable, but more promising than gait based schemes.

2. Related Work

A great deal of research has been performed on investigating gait as a bio-metric. The main idea is that the way a person walks is unique [1]. Gait based bio-metric research is split into two categories. The first is to capture gait externally, such as using cameras to record movement. The second is to measure gait using sensors, like accelerometers and gyroscopes, attached to the body to measure gait. To analyze gait, we use well studied machine learning techniques. One major issue with gait based bio-metrics is that many external factors affect how we walk. There is current work that uses a voting system and a sliding window to increase the reliability of these measurements, but they are still too unreliable when measurements are taken over multiple days.

Another related body of research is the use of accelerometers to learn a users password [4], [5], [6], [7]. This work is generally used as an attack, but we are interested in taking advantage of these techniques to measure how users input their passwords. This is similar to the attack specified in ”Keyboard acoustic emanations revisited” by Zhuang et al, which uses sound waves and machine learning algorithms to determine what a person typed. While we leave this to future work, from our current research, we believe that it is much easier to authenticate based on how users type than determining what they type.

The most closely related work is [3]. This work identifies gestures that are then used to perform some action. The example the authors used is to identify hand-shakes to exchange contact information via smart watch. This differs from our work because we are interested in not only identifying gestures, but in also identifying individuals performing said gestures.

3. Design

A smart watch, with AndroidWear OS runs a custom application that records the accelerometer data to a text file. The raw accelerometer data is processed to extract time domain features and frequency domain features. These features are then combined in to 3 different feature vectors. The Gait Feature vector and Time Feature vector solely consists time domain features. The Time and Frequency vector consists of features from both time and frequency domains. These feature vectors are then passed through a

K-nearest neighbor and Random Forest classifiers. Then resulting true and false positive rates for classification are calculated (1).

4. Implementation

In our implementation, we aim to analyze the accuracy of classifying a user or gesture given some learning data. In supervised learning, we conduct the machine learning task of inferring a function predicting users or gestures from correspondingly labeled training data. To complete this task, we must collect raw accelerometer data, analyze raw data for features, train a classifier given this data, and analyze our prediction accuracies given different scenarios.

4.1. Data Collection

Subjects were asked to perform 5 different gestures: walking, opening a door, typing a predetermined phrase, lifting a cup, checking the watch. Each user wore a smart watch on their dominant hand, and performed each gesture with said hand when applicable.

An Android application was written for the watch to record the accelerometer readings. It started recording once the user tapped the screen, and stopped once 500 samples were recorded.

As these actions were performed, the smart watch logs the magnitude of the accelerometer vector along the x, y, and z axes. For each action, 5 seconds of raw accelerometer data was collected. At a sample rate of 100 Hz, this leads to 500 raw accelerometer vectors to analyze. We refer to this type of data as the sample window. We chose 5 seconds as the duration of the window because it is long enough to capture a full range of motion of most small gestures without introducing too much noise such as transitioning from completing the gesture. Each sample window is labeled according to both user and gesture.

Our final dataset includes 10 users and 5 gestures. We note that 6 users are male and 4 are female. Among the users, 8 are right handed, and 2 are left handed. Each user performed each gesture 5 times. Overall, there are 250 total sample window instances, or 125,000 raw accelerometer vectors.

4.2. Feature Extraction and Selection

In our research, we identified features we can extract from a sample window in hopes of finding correlations with each user. We focused on 3 different feature vectors: gait features, time domain features, and both time and frequency domain features [1] [8] [9].

Each sample window was processed to identify features such as mean, variance, standard deviation, and differences between the axis, as in [8]. In addition, each data set was processed in MATLAB to extract features in the frequency domain such as Fourier Coefficients, Spectral Energy, and Spectral Entropy. This post processing generates a feature vector for use in classification.

Our final feature representations can be referred to in Table 1.

4.3. Classification

We leverage the different classifiers available in the WEKA data mining software provided by the University of Waikato [14]. We take note that each classifier will have varying results depending on what we are predicting: users or gestures. The training process would involve learning in relation to the label we would want to predict. In our experiments, we focused on comparing the performance of two classifiers: k-Nearest Neighbors and Random Forest.

4.3.1. k-Nearest Neighbors. When classifying a given test feature vector, k-Nearest Neighbors will find the k nearest training feature vectors given some distance function (typically L2-norm), look at all k training labels, and predict the label as the majority of the k labels. A nice advantage of k-Nearest Neighbors is its robustness against noisy data. Also, as the training set size increases, the classifier will overfit less. However, this comes at a high computation cost, and the parameter k will have to be tuned in each experiment.

4.3.2. Random Forest. Random forest is a variant of decision trees that performs to some degree a weighted nearest neighbors. To explain decision trees, the input space is first separated by class regions. In determining a decision tree, nodes are generated with decision functions that branch depending on the output of the decision. As one traverses from root to leaf, the classifier effectively narrows the prediction space until it reaches its final prediction at the leaf. In describing this, there are concerns with overfitting strongly to the training data. Random forest alleviates this by randomly sampling the training data to generate multiple decision trees, with a final prediction generated by the majority vote of all decision trees. Random forest brings a fast and scalable implementation, and without the need to tune parameters.

4.4. Experiments

The output of the classifier results is a strength indicator of our ability to predict which user is wearing the smart watch, or which gesture is being performed. To extract meaningful results of our predictions given our dataset, we perform 5-fold cross validation. In this experiment, the data is split into 5 equal subsets. A single subset is chosen as a validation set, with the other 4 subsets used as the training data to be fed into a classifier. Classifier results are generated with this setup with every instance (5 second window of accelerometer data) in the validation set being classified against the training sets. This entire process is repeated, by picking each subsequent subset as a validation set with the reaming as the training set, leading to a total of 5 experiments, which are weighted for the final results.

We perform 5-fold cross validation experiments with variants of the 3 feature representations and 2 classifiers to identify users and gestures.

Figure 1. Block Diagram of System Flow

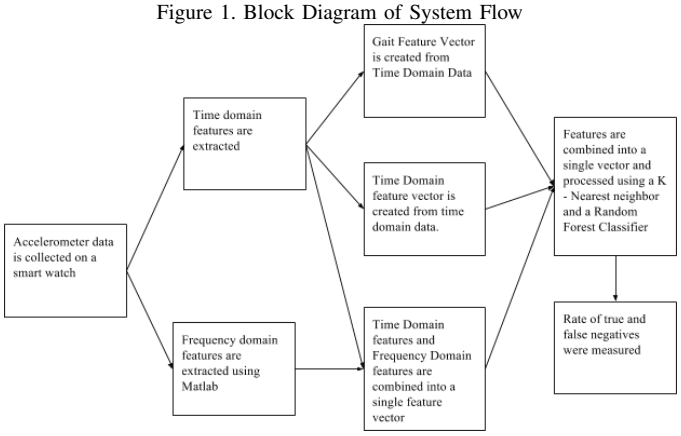


TABLE 1. FEATURE REPRESENTATIONS

Gait	Time	Time + Frequency
Mean (x, y, z)	Mean (x, y, z)	Mean (x, y, z)
Standard Deviation (x, y, z)	Standard Deviation (x, y, z)	Standard Deviation (x, y, z)
Average Absolute Distance (x, y, z)	Variance (x, y, z)	Variance (x, y, z)
Time Between Peaks (x, y, z)	Sample Differences (x, y, z)	Sample Differences (x, y, z)
10 - Point Binned Probability Distribution (x, y, z)	Minimum (x, y, z)	Minimum (x, y, z)
Average Resultant Acceleration	Correlation (xy, xz, yz)	Correlation (xy, xz, yz)
		Spectral Energy (x, y, z)
		Fourier Coefficient Sums (x, y, z)
		Spectral Entropy (x, y, z)

5. Results

The results will primarily focus on our true positive and false positive rates. While we obtained many more statistical points, these true and false positive rates represent two important aspects of authentication. True positives represent how usable of an authentication scheme it would be. A low true positive rate indicates that many legitimate attempts to authenticate would fail thus making this too much of a burden to use. False positives represent how secure this would be as an authentication scheme. A high false positive rate means illegitimate users could bypass the security and authenticate when they were not suppose to. Thus, we are always aiming to achieve high true positive rates and low false negative rates.

5.1. Classification Method

Throughout our experiments we focused on two classifiers: k-nearest neighbors and random forest algorithms. Figure 2 and Figure 3 shows the effect that the different classifier algorithms have on the average true and false positive rates for all three feature vectors classifying by gestures and by users. These graphs show that for time domain features and time frequency domain features the k-nearest neighbors algorithm out performs the random forest algorithm. However, when using gait features the random forest out performs the k-nearest neighbors algorithm. While the random forest algorithm does better while using the gait feature vector, the k-nearest neighbors algorithm performs significantly better while paired with the time domain and time frequency domain feature vectors.

5.2. Classification by Gesture

Table 2 shows a comparison of the true and false positive rates for the three different feature vectors when classifying by gesture. These are the results of using the k-nearest neighbors algorithm with $k = 1$.

Looking at true positive rates, we see that the time domain feature vector outperforms the gait feature vector with a higher true positive rate by 33.2%. Combining Time domain feature vectors and frequency domain feature vectors increases the true positive rate by another 0.4%. The time frequency feature vector performed the best and was able to achieve a 99.6% average true positive rate across all gestures performed. The only action that had any error was walking. We found one instance where a walking gesture was classified as a door opening gesture. This is likely do to users walking forward while pushing the door open causing some confusion in the classification.

Looking at false positive rates, we once again see the time domain feature vector outperforms the gait feature vector with a decrease of 8.2% in the false positive rate, and the time-frequency domain features was able to remove another 0.1% from the false positive rate over the time domain feature. The best feature vector was once again the time-frequency feature vector achieving a average false

Figure 2. Classifier Algorithm Comparison: Average True Positives

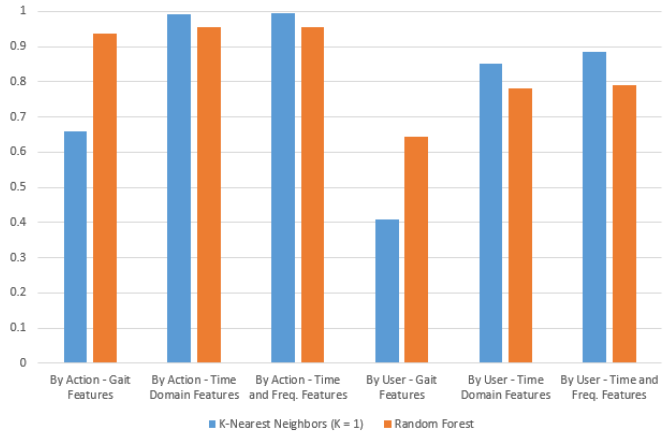


Figure 3. Classifier Algorithm Comparison: Average False Positives

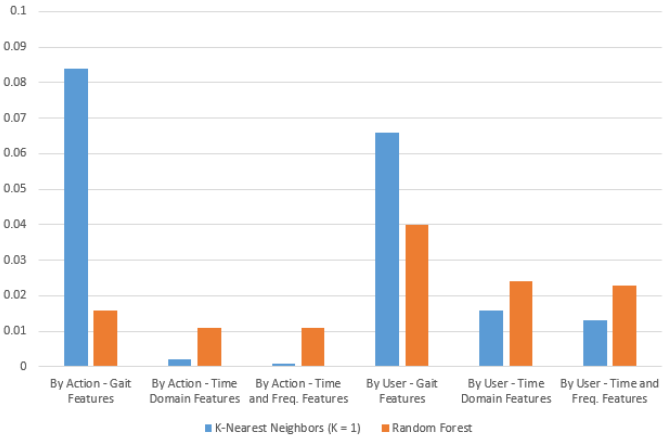


TABLE 2. TRUE AND FALSE POSITIVE RATES WHILE IDENTIFYING ACTION.

	Gait Features		Time Domain Features		Time + Freq. Domain Features	
	True Positive	False Positive	True Positive	False Positive	True Positive	False Positive
Cup	0.882	0.080	1.000	0.000	1.000	0.000
Door	0.540	0.160	0.980	0.000	1.000	0.005
Typing	0.939	0.184	1.000	0.000	1.000	0.000
Walking	0.400	0.000	0.980	0.005	0.980	0.000
Watch	0.540	0.000	1.000	0.005	1.000	0.000
Average	0.660	0.084	0.992	0.002	0.996	0.001

TABLE 3. TRUE AND FALSE POSITIVE RATES WHILE IDENTIFYING USER.

	Gait Features		Time Domain Features		Time + Freq. Domain Features	
	True Positive	False Positive	True Positive	False Positive	True Positive	False Positive
Chen	0.577	0.112	0.962	0.004	1.000	0.004
Chris	0.360	0.027	1.000	0.018	1.000	0.009
Derrick	0.333	0.035	0.875	0.031	0.833	0.027
Scott	0.320	0.049	0.880	0.022	0.960	0.013
Sebastian	0.400	0.133	0.720	0.004	0.840	0.004
Matt	0.280	0.022	0.840	0.004	0.880	0.004
Justine	0.480	0.151	0.720	0.027	0.800	0.013
Jennifer	0.480	0.058	0.840	0.040	0.800	0.036
Jackie	0.480	0.044	0.920	0.004	0.920	0.004
Sabrina	0.360	0.027	0.760	0.009	0.800	0.013
Average	0.408	0.066	0.852	0.016	0.884	0.013

positive rate of 0.1% over all studied gestures. This is a promising result, as this is the same false positive rate of guessing a truly random 4-digit pin.

5.3. Classification by User

Table 3 shows a comparison of the true and false positive rates for the three different feature vectors when classifying by user. These are also the result of using the k-nearest neighbors algorithm with $k = 1$.

Looking at true positive rates, we see the time domain feature vector once again out performing the gait feature vector with an increased true positive rate of 44.40% and the time-frequency domain feature vector gaining another 3.20% increase in true positive rate over time domain features. Our best result using the time frequency domain feature vector achieved an average 88.4% true positive rate. This is not as high as we would hope for an authentication scheme, meaning that about 1 out of every 9 legitimate attempts to authenticate would be rejected.

Looking at false positive rates, we see the time domain feature vector was able to reduce false positive rates by 5.0% over the gait feature vector. The time frequency domain feature vector was still the best with another 0.3% decrease in false positive rate over the time domain feature vector. The time frequency feature vector was able to achieve an average false positive rate of 1.3%. This is not as low as the false positive rate when classifying by gesture which was able to achieve a 0.1% false positive rate. A 1.3% false positive rate is higher than we would prefer for an

Figure 4. User Classification Per Gesture: True Positives

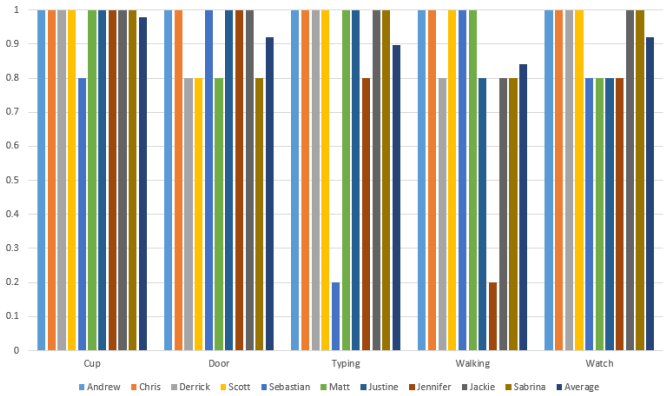
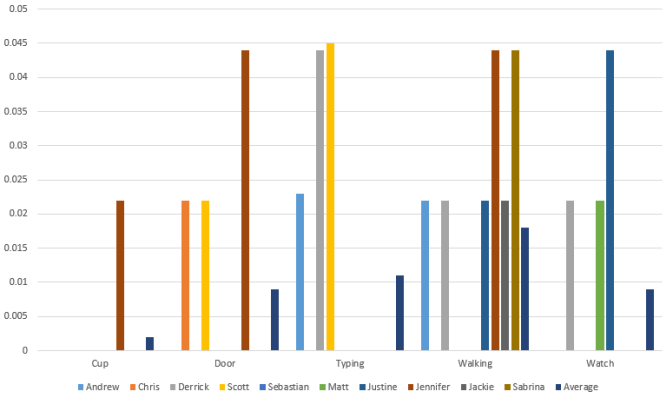


Figure 5. User Classification Per Gesture: False Positives



authentication scheme but shows that with some additional work, it has the potential to have a false positive rate that would be acceptable.

5.4. Hybrid Classification

For authentication, it would be beneficial to have classification by user be better than our results currently are showing. One potential improvement would be to combine classification of gestures and users. First, a normal classification would be performed to try and identify which gesture was being done. Then, knowing what gesture was done, we can try to identify which user performed the gesture by classifying the user only over the identified gesture. The idea here is that gesture classification is significantly better than user classification so we might be able to use gesture classification to remove noise and thus better classify which user was performing the gesture.

Figures 4 and 5 show the true and false positive rates of user classification after determining what gesture was performed. The results are done using the time frequency feature vector with a k-nearest neighbors algorithm ($K = 1$) as these performed the best in the previous classifications.

TABLE 4. AVERAGE TRUE AND FALSE POSITIVE RATES WHILE CLASSIFYING USER OVER A SINGLE GESTURE.

	Avg. User True Pos. Rate	Avg. User False Pos. Rate
Cup	0.980	0.002
Door	0.920	0.009
Typing	0.898	0.011
Walking	0.840	0.018
Watch	0.920	0.009

From Table 4 we can see that the average user classification true positive rates are greater over a single gesture than over all gestures for each gesture except walking which decreased by 4.4%. Similarly, the average user classification false positive rates are smaller over a single gesture that overal all gestures for each gesture except walking which increased by 0.5%. This also shows that classification of a user based on how they lift a cup is the most accurate of all the gestures with an average true positive rate of 98.0% and an average false positive rate of 0.2%. This leads us to believe that with a further study of different gestures as well as different hybrid classification models a hybrid classification could potentially be used as an authentication scheme.

6. Limitations and possible attacks

The main limitation to movement based identification is that the signal used to determine an identity is inconsistent. For example, in [1], the authors mention that the reliability of the signal they use (gait) degrades significantly between days. We believe that this is due to the fact that many external factors, like mood, affect how someone moves.

Current attacks on accelerometers and gyroscopes create keyloggers based on vibrations caused by key presses [4], [5], [6], [7]. These attacks tend to fall into two categories: attacks using an external device and attacks using an attacked device. However, they all have the same underlying goal: use accelerometer data to act as a key logger. Well this can be used as a side channel attack, as we discuss below, it maybe possible to use the way a person types as a form of gesture based authentication.

When dealing with new forms of authentication, an immediate concern is replay attacks. A simple replay attack on smart watch based authentication is for a malicious application on the smart watch records all of the accelerometer data. It is not immediately clear that this is an issue, but modern cryptographic techniques, such certificates, can protect against replay attacks. We leave defenses for replay attacks to future work.

7. Future work

The goal of this paper is to show that identifying repeated gestures can be just as, if not more, effective at providing a second factor of authentication as gait based approaches. As we have shown, it is both possible to distinguish between both individuals performing a gesture and

between unique gestures. An immediate idea we have for future work is to determine how users types their password and use that data as a second step of authentication. This idea is not new, as there is existing research that shows it is possible to uses accelerometer data to determine a users password. However, we are not interested in determining a unique password, we are interested in identifying how a person types his or her password. Preliminary results from the typing gesture indicate that this is indeed a promising area of research.

Further, as discussed above, a key limitation to movement based authentication is the variability of the signal. Future work will focus on studying potential gesture sequences that increase the consistency of the signal. Initially we used a similar approach as the one taken in [1]. However, this long window is subject to many external factors. We then used a simple nearest neighbors algorithm and our results for all of the non gait based gestures are measurably better than the gait based results.

When using the nearest neighbors algorithm, our results are promising for both identifying users performing the same action and identifying between actions. However, this is used on a relatively small data set. Our data set includes 10 users performing 5 gestures 5 times per gesture. In order to verify our results, we used a freely available data set from UCI. This data set does not distribute classes uniformly. This has led to a set of results are roughly inline with the results of our data set. Future work would entail building out our data set with more users, gestures, and training runs.

Finally, our data set is collected with only an LG smart watch running the AndroidWear OS. It would be useful to use different watches running different operating systems. Just like [15], used different keyboards to measure the how their system performed on different keyboards, we believe it is useful to measure gesture recognition on a wide verity of smart watches.

8. Conclusion

Smart watches are becoming increasingly popular. This popularity has forced the community to study the security implications of these small and powerful devices. It has been suggested that gait based authentication combined with smart watches is possible. We argue that gait based authentication is just one instance of a larger class of movement based bio-metrics. We also performed a test using multiple movement based gesture including walking (gait), opening a door, and typing. The results of our experiment suggest that indeed gait is in fact an instance of a larger class of movement based bio-metrics. They also suggest that other gestures maybe more accurate than gait. Further, our results suggest that one promising bio-metric is how a person types. When we combine the results in the time domain and the frequency with the nearest neighbors algorithm, the true positive rate is 100 percent with a false negative rate of 0 percent.

References

[1] A. Johnston, G. Weiss. Smartwatch-based biometric gait recognition. Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference, 2015.

[2] W. Vlaenderen, et.al. WatchMe: A Novel Input Method Combining a Smartwatch and Bimanual Interaction. Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems, 2015.

[3] A. Ferrari, D. Puccinelli, S. Giordano. Gesture-based soft authentication. Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference, 2015.

[4] P. Marquardt, A. Verma, H. Carter and P. Traynor, (sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers, Proceedings of the ACM Conference on Computer and Communications Security (CCS), October, 2011.

[5] T. Beltramelli and S. Risi, Deep-spying: Spying using smartwatch and deep learning, CoRR, vol. abs/1512.05616, 2015. [Online]. Available: http://arxiv.org/abs/1512.05616

[6] He Wang, Ted Tsung-Te Lai, Romit Roy Choudhury,MoLe: Motion Leaks through Smartwatch Sensors,, ACM MobiCom, September 2015

[7] Xiangyu Liu, Zhe Zhou, Wenrui Diao, Zhou Li, Kehuan Zhang. When Good Becomes Evil: Keystroke Inference with Smartwatch. In Proceedings of the 22nd ACM Conference on Computer and Communications Security (CCS15), Denver, USA.

[8] S. Chernbumroong, A. S. Atkins and H. Yu, "Activity classification using a single wrist-worn accelerometer," Software, Knowledge Information, Industrial Management and Applications (SKIMA), 2011 5th International Conference on, Benevento, 2011, pp. 1-6. doi: 10.1109/SKIMA.2011.6089975

[9] Davide Figo, Pedro C. Diniz, Diogo R. Ferreira, Joo M.P. Cardoso, Preprocessing techniques for context recognition from accelerometer data, Personal and Ubiquitous Computing, vol.14, no.7, pp.645-662, 2010

[10] S. F. Darwaish, E.Moradian, Tirdad Rahmani, Martin Knauer, Biometric Identification on Android Smartphones, Procedia Computer Science, 35, 832-841,(2014).

[11] Vildjiounaite, E., Kyllnen, V., and Ailisto, H. (2009). Empirical evaluation of combining unobtrusiveness and security requirements in multimodal biometric systems. Image and Vision Computing, 27(3), 279-292. doi:10.1016/j.imavis.2007.12.001

[12] Casale, P, Pujol, O, and Radeva, P 2012, 'Personalization and user verification in wearable systems using biometric walking patterns', Personal and Ubiquitous Computing, 16, 5, pp. 563-580, Academic Search Complete, EBSCOhost, viewed 13 March 2016.

[13] J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," in Proceedings of the IEEE, vol. 63, no. 9, pp. 1278-1308, Sept. 1975. doi: 10.1109/PROC.1975.9939

[14] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[15] Li Zhuang, Feng Zhou, and Doug Tygar. Keyboard Acoustic Emanations Revisited. In Proc. of ACM CCS, November 2005.