Feet Fidgeting Detection Based on Accelerometers Using Decision Tree Learning and Gradient Boosting

Julien Esseiva[1], Maurizio Caon[1], Elena Mugellini[1], Omar Abou Khaled[1], Kamiar Aminian[2]

[1] University of Applied Sciences and Arts Western Switzerland, Fribourg, Switzerland
{julien.esseiva, maurizio.caon, elena.mugellini, omar.aboukhaled}@hefr.ch

[2]Laboratory of Movement Analysis and Measurement, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland
kamiar.aminian@epfl.ch

Abstract. Detection of fidgeting activities is a field which has not been much explored as of now. Studies have shown that fidgeting has a beneficial impact on people's healthiness as it burns a significant amount of energy. Being able to detect when someone is fidgeting would allow to study more closely the health impact of fidgeting. The purpose of this work is to propose an algorithm being able to detect feet fidgeting period of subjects while sitting using 3-D accelerometers on both shoes. Initial results on data from 5 subjects collected during this work shows an accuracy of 95% for a classification between sitting with fidgeting and sitting without fidgeting.

Keywords: Fidgeting detection, Decision Tree, Boosting, Accelerometers, Footwear, Wearable, Machine Learning

1 Introduction

Physical inactivity is more and more frequent in today society. It is known that sedentary has a negative impact on health, notably weight increase as energy is conserved as body fat. Health specialists therefore recommend to people to do physical activities to stay healthy. Recent studies have shown that people doing a light physical activity like walking, household activities or fidgeting could add up to several hundred of calories burnt per day, which has an impact on human well-being. While the field of activity and gesture recognition has seen a lot of research the past years, mainly due to the growing of fields such as human-computer interaction (HCI) or health-monitoring devices, the problem of detecting fidgeting periods has seen much less research. Activity and gesture recognition might initially seem to be a problem similar to fidgeting detection. There are however some key differences, mainly in the nature of movements. Activities are spanning on a longer period while fidgeting is generally short and spontaneous. This has an impact on the discriminative power of features, the mean of a signal is frequently used in posture recognition as it averages out over a long time, while it might not be as important for fidgeting detection due to the short and spontaneous nature of the movement. An algorithm allowing to detect whether a subject is fidgeting is of great interest, it would allow to further assess the impact of fidgeting on energy consumption or it could be used to encourage fidgeting. Therefore, the purpose of this work is to propose an algorithm detecting periods where the subject is fidgeting with its feet by using smart shoes composed of tri-axial accelerometers. First a selection of similar work is presented. The dataset as well as the data collection protocol and instrumented shoe system is discussed. The algorithms chapter presents the pipeline of the data, the features extracted from the signals as well as trained models. Tests done during the work, which include determining an optimal window size and reproduction of a baseline algorithm are then presented, followed by a discussion of the result. The work ends with a conclusion and possible outlook.

2 Related Works

Various algorithms were proposed for the tasks of activity and gesture recognition. Ayumi explored the application of Gradient boosting to action recognition over multiple dataset using a Kinect and depth sensor camera. While the nature of the data is different in our work, Ayumi compared gradient boosting to methods such as SVM or naive Bayes. The gradient boosting method outperformed the other two in most cases [3]. Zhang et al. worked on activity classification using decision tree and tri-axial accelerometers on instrumented shoe sensors [2]. El Achkar et al. proposed a system for activity classification using Instrumented shoe and a tree-based algorithm [5]. In these two studies data are similar to our work while movements are different, which impacts the choice of features and windowing. Tapia presents a detailed analysis of activity recognition using machine learning algorithm based on accelerometers data, exploring questions such as the processing and windowing of the signals, impact of multiple features, various models such as decision tree, naive Bayes, nearest neighbor and the differences between subject-dependent and subject-independent training [6]. Lugade et al. similarly proposed an algorithm which detects movements with a sensitivity greater than 85% using accelerometers which were placed on the waist and thigh. Amongst the activities, they were trying to detect quiet sitting and sitting with fidgeting. While the proposed algorithm could get good performances on activity recognition, they do mention that their lowest accuracy was for activities where the subject was fidgeting while sitting or standing [1]. As suggested by Lugade et al, it is necessary to do more research on human motion while fidgeting which is the focus of this work. Zhang et al. proposed an algorithm to detect feet fidgeting while the subject is sitting by using a 3D accelerometer. This is, to our knowledge, the only algorithm which is performing feet fidgeting recognition while seated and will therefore be used as a baseline algorithm for this work [4].

3 Datasets

Data in this study data were gathered using instrumented shoe sensors which were provided by [6].

3.1 Instrumented Shoe

The instrumented shoe system is composed of insoles with force sensors, a Physilog which includes 3D accelerometer, 3D gyroscope and 3D magnetometer with a sampling frequency of 200 Hz. The system can easily be inserted in any shoe. The orientation and position of Physilog sensors is not primordial as calibration of the signal is done. Fig. 1 shows the instrumented shoe once set up as well as the orientation of accelerometer and gyroscope axes.

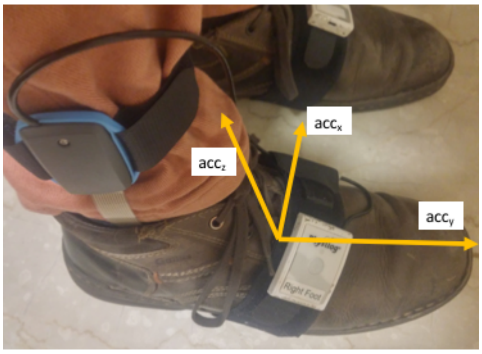

Fig. 1. Instrumented shoe setup with axes orientation

3.2 Collected dataset

A dataset was collected for this study using the instrumented shoe in [6]. Data from 6 different subjects were collected. Each subject had to follow a predetermined protocol after a calibration of the sensors.

Collection protocol. Table 1 shows the protocol followed by each subject. They were sitting on a chair for the whole time and asked to perform four different leg gestures generally considered as fidgeting, with 5s of quiet standing in-between each fidget. During the quiet sitting period, the subject was asked not to move. This protocol was performed twice for each subject, one time for each foot. A supervisor was present during the collection protocol to note the time at which subjects where changing between tasks so that it is possible to label the data later.

Task	Duration (s)
Quiet sitting	5
Fidget 1	10

Quiet sitting	5
Fidget 2	10
Quiet sitting	5
Fidget 3	10
Quiet sitting	5
Fidget 4	10

Table 1. Data Collection Protocol

The total time of the protocol is 1 min of data per subject, repeated once for each foot. Considering 6 subjects, this results in a dataset which contains a total of 12 minutes of data.

- Fidget 1 (upper leg swinging): The subject is moving its thighs left and right, either one or both at the same time. The two feet are constantly touching the ground. This gesture could be identical for each foot, depending on the subject.
- Fidget 2 (up and down leg bouncing): The subject has his two feet on the floor and is repetitively moving the heel of one leg up and down (along the z axis) with the toe still touching the ground.
- Fidget 3 (Lower leg swinging): The heel of one leg stands on the knee of the other one. The subject was asked to move its foot.

Fidget 4 (Foot jiggling): The legs of the subject are crossed with its thighs being one over the other. The lower leg of the subject is swinging, usually along the y axis.

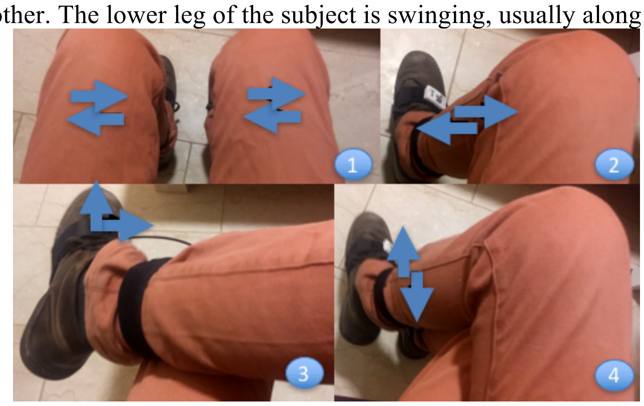

Fig. 2. Illustration of the 4 types of fidgeting recorded

Collected Features. Given the multiple sensors present on the instrumented shoe, it was possible to collect a total of 15 signals per foot as listed below.

- 3D acceleration [g]
- 3D gyroscope [deg/s]
- Force of the eight insole sensors [V]
- Pressure [KPa]

Collected data were labeled to train supervised models. In that regard, signals were shown on a plot where the supervisor could specify fidgeting periods as noted during the collection protocol.

4 Algorithms

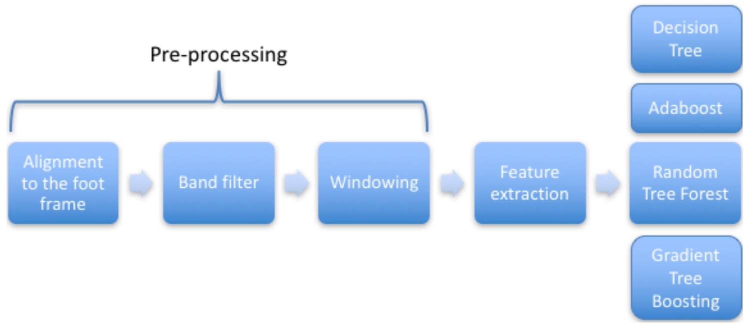

Fig. 3. Algorithm pipeline of the processed data.

4.1 Signal processing.

Filters. Raw signals extracted from sensors are usually noisy. It is important to preprocess them before extracting features from them. Removing frequencies considered to be noise is an important first step. A Butterworth bandpass filter with the following specification was applied to signals:

- Passband: [0.1 20] Hz
- Stopband: [0.01, 99] Hz
- Maximum passband ripple: 1 dB
- Minimum stopband attenuation: 60 dB

Windowing. A second important step is to split signals in multiple windows. An optimal window length is not universally defined and seems to depend on the problem to be solved. In the context of this work it is necessary to detect fidgeting which are usually short spontaneous movements. A small window therefore seems to be more appropriate to the problem. Both non-overlapping and 50% overlapping windows will be extracted from the signals to compare the two versions in terms of performance metrics.

4.2 Feature extraction

The present work focused on using accelerometers, therefore only the six accelerometers signals, x, y and z for each foot, were used. For each signal, multiple features inspired by the work of Tapia [6] were computed as presented below. Both time and frequency features were extracted. For the frequency domain features, the 0 Hz term of the FFT was omitted when computing features related to the frequency domain. Features are extracted on a window-basis. A total of 62 features are extracted for each window and the whole set of features was used to train classification models.

FFT Peaks. The frequency component with the highest magnitude. A tuple which contains the frequency and its magnitude is extracted.

Main frequency energy ratio. This feature is computed using the equation 1, where mag_{hf}^2 is the highest magnitude across all frequencies of the signal. The maximum frequency is 20 Hz.

$$R = \frac{mag_{hf}^2}{\sum_{f=0}^{\max\_freq} mag_f^2} \quad (1)$$

Entropy. The entropy is a commonly used feature in signal processing measuring the signal complexity. Equations 2 and 3 show how the entropy is computed, with B being the number of FFT bins.

$$H = -\sum_{b=1}^{B} p_b * \log_2 p_b \quad (2)$$

$$p_b = \frac{mag_b}{\sum_{j=0}^{B} mag_j} \quad (3)$$

Root mean square. The root mean square is a commonly used feature in statistics and is easy to compute. The signal is detrended before computing this feature, with N being the number of samples in the window and a_n is the measured acceleration at index n in the window.

$$RMS = \sqrt[2]{\frac{1}{N}\sum_{n=1}^{N} |a_n|^2} \quad (4)$$

Absolute area. This feature is the sum of the absolute value of each sample in the window.

Absolute mean. Like the absolute area, the absolute mean corresponds to the mean of the samples in the window.

Autocovariance. The autocovariance corresponds to the cross-covariance of a signal with itself. Only the highest and the lowest values were kept for each signal which results in 2 features per signal.

Approximate entropy. The approximate entropy is a feature which is useful for capturing signal complexity and the evolution of complexity and predictability of the signal. A value close to 0 suggests that the signal is predictable and regular. The parameters m=2 and r=0.01 were used to compute this feature [8].

Total SVM. The signal vector magnitude of the signal is extracted as a feature by using the equation 5

$$SVM = \frac{1}{N}\sum_{n=1}^{N} \sqrt[2]{\sum_{i=1}^{nb\_axes} a_{n\_i}^{2}} \quad (5)$$

Total absolute area. This feature corresponds to the sum of the absolute area feature of the six accelerometer signals used. It is computed once per window instead of once per signal.

4.3 Models

Multiple supervised models were trained and compared in this work, which includes decision tree, random forest, AdaBoost and gradient tree boosting. The training was done using K-fold cross-validation with K=10. A grid search was performed on each model to optimize hyperparameters. Both binary and multiclass models were trained using a single subject-independent dataset. The purpose of binary models was to classify between fidgeting and no fidgeting while multiclass models should recognize each fidgeting gesture independently. For each category (binary / multiclass) of models, the four machine learning algorithms were trained once using non-overlapping window and a second time with 50% overlapping window. The choice of the optimal window size is detailed in 5.2.

5 Tests

5.1 Baseline algorithm

Zhang et al. proposed an algorithm to detect whether a subject is sitting or sitting and fidgeting [4]. In a 2-second window, the sum of the square of the three accelerometer signals is computed for each sample in the window. Only the maximum of these values in the window is kept and signals are not preprocessed in this algorithm. If the maximum value is above a given threshold, the window is classified as fidgeting. One limitation of this algorithm is that it can't be used for multiclass classification. This algorithm was reproduced and used as a baseline comparison for the proposed algorithm. As only few information on how to compute the decision threshold was available, a decision tree with the entropy splitting criterion and a maximum depth of 1 node was used to determine this threshold.

5.2 Optimal window length

Tapia [6] proposed a procedure to determine an adequate window length for a given dataset. Two features highly impacted by the window length, the Pearson correlation coefficient and FFT peaks were computed for each signal, using windows size ranging from 1s to 7s, with a step size of 0.2s. A CART Decision Tree was then trained on the two features previously mentioned for each combination of window length, window overlap and classification type. Performances were assessed using 10-fold cross-validation. The Fig. 4 shows the trend for the binary, non-overlapping windows model, other models are not shown here but similar results were obtained. The accuracy decrease as the window length increase which can be explained by the fact that increasing the window size drastically reduces the number of samples of the already small dataset therefore increasing the generalization error of the model.

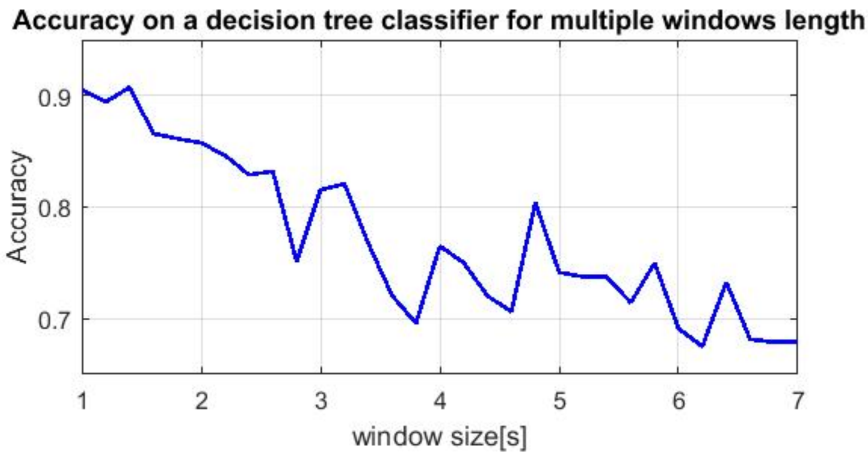

Fig. 4. Accuracy as function of window length for the binary, non-overlapping windows model.

6 Results and Discussion

Table 2 and 3 summarize the results obtained with the best performing model resulting of the grid search using 10-fold cross-validation. Metrics are computed using a macro-average. We see that models trained on overlapping window perform slightly better overall. Performances are good for both binary and multiclass models.

Regarding the binary problem, DT perform slightly worse than other models. The Matthews correlation coefficient (MCC) is significantly lower for the baseline algorithm. Because the classes are not perfectly balanced, metrics such as accuracy are slightly skewed while the MCC is more representative of the true performances.

	Accuracy		Precision		Recall		F-score		MCC	
	No	O	No	O	No	O	No	O	No	O
Base	0.80		0.83		0.81		0.81		0.58	
DT	0.89	0.93	0.89	0.93	0.89	0.93	0.89	0.93	0.72	0.83
RF	0.93	0.95	0.93	0.95	0.93	0.95	0.93	0.95	0.83	0.89
AB	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.86	0.85
GB	0.94	0.94	0.94	0.95	0.94	0.94	0.94	0.95	0.86	0.87

Table 2. Performances of binary models using rectangular windows. “No” stands for non-overlapping windows, “O” for 50% overlapping windows, “DT” is decision tree, “RF” is random forest, “AB” is AdaBoost, “GB” is gradient tree boosting.

The multiclass problem still yields decent performances. Gradient tree boosting has the best performances overall. While models might be overfitted, it is to be noted that boosting method such as gradient tree bosting trains multiple shallow trees and is therefore resilient to overfitting.

	Accuracy		Precision		Recall		F-score	
	No	O	No	O	No	O	No	O
DT	0.77	0.82	0.77	0.82	0.77	0.82	0.77	0.82
RF	0.85	0.90	0.85	0.89	0.85	0.89	0.85	0.89
AB	0.40	0.64	0.42	0.65	0.41	0.64	0.40	0.64
GB	0.88	0.91	0.88	0.91	0.88	0.91	0.88	0.91

Table 3. Performances of the multiclass models using rectangular window.

Fig. 5 shows the confusion matrix for the GB model with 50% overlapping window of length 1.2s, which results in a dataset of approximately 1200 samples. While other matrices are not presented here, the error distribution is similar in each one of them. We see that there is a confusion between the gesture “Lower leg swinging” and “Foot jiggling”. These two gestures are indeed quite similar, another source of misclassification is the “No fidgeting” class. A possible source of confusion here is that the “No fidgeting” class isn’t perfectly representative of the reality as subjects were asked not to move during the quiet sitting periods. A subject might do small movements that should not be considered as fidgeting. It is also to be noted that the classes are not perfectly balanced, with more “No fidgeting” samples compared to the other classes for the multiclass models. It is the opposite for the binary models, 2/3 of the samples are labelled as fidgeting.

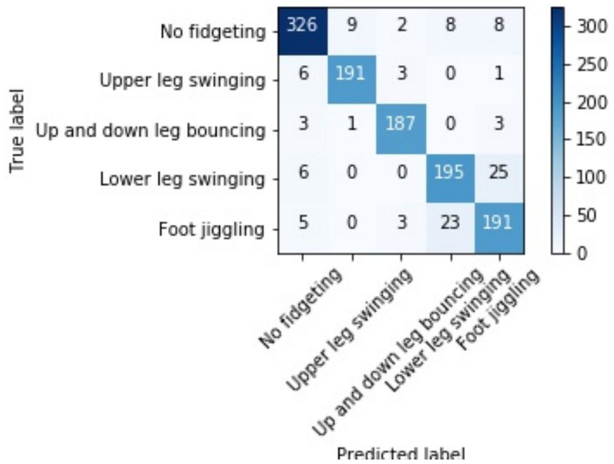

Fig. 5. Confusion matrix of the GB model trained on 50% overlapping 1.2s windows

7 Conclusion

In this work, we showed initial results on detecting fidgeting activities while seated. The algorithm can get decent performances on the multiclass problem and better performances when compared to the baseline algorithm for the binary classification problem. While more work is still necessary, this does provide initial promising results. The main limitation of this work is the collected dataset which is too small and leads to model overfitting. Therefore, the next step is to collect more data by having subjects in real-life situations instead of following a predefined protocol. Other possible outlook includes performing dimensionality reduction algorithm to select only the most relevant features and extracting features from other sensors such as the gyroscope or the force sensing insole.

References.

1. Lugade, Vipul, Emma Fortune, Melissa Morrow, and Kenton Kaufman. "Validity of using tri-axial accelerometers to measure human movement—Part I: Posture and movement detection." Medical engineering & physics 36, no. 2 (2014): 169-176.
2. Zhang, T., Tang, W., & Sazonov, E. S. (2012, August). Classification of posture and activities by using decision trees. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE* (pp. 4353-4356). IEEE.
3. Ayumi, V. (2016, December). Pose-based human action recognition with Extreme Gradient Boosting. In *Research and Development (SCOReD), 2016 IEEE Student Conference on* (pp. 1-5). IEEE.
4. Zhang, T., Klein, D. A., Walsh, T., Lu, J., & Sazonov, E. S. (2014, June). Android TWEETY—A wireless activity monitoring and biofeedback system designed for people with Anorexia Nervosa. In *Medical Measurements and Applications (MeMeA), 2014 IEEE International Symposium on* (pp. 1-5). IEEE.

5. el Achkar, C. M., Lenoble-Hoskovec, C., Paraschiv-Ionescu, A., Major, K., Büla, C., & Aminian, K. (2016). Instrumented shoes for activity classification in the elderly. *Gait & posture*, *44*, 12-17.
6. Munguia Tapia, E. (2008). Using machine learning for real-time activity recognition and estimation of energy expenditure (Doctoral dissertation, Massachusetts Institute of Technology).
7. Moufawad El Achkar, C., Lenoble-Hoskovec, C., Paraschiv-Ionescu, A., Major, K., Büla, C., & Aminian, K. (2016). Physical behavior in older persons during daily life: insights from instrumented shoes. *Sensors*, *16*(8), 1225.
8. Pincus, Steven M. "Approximate entropy as a measure of system complexity." *Proceedings of the National Academy of Sciences* 88.6 (1991): 2297-2301.