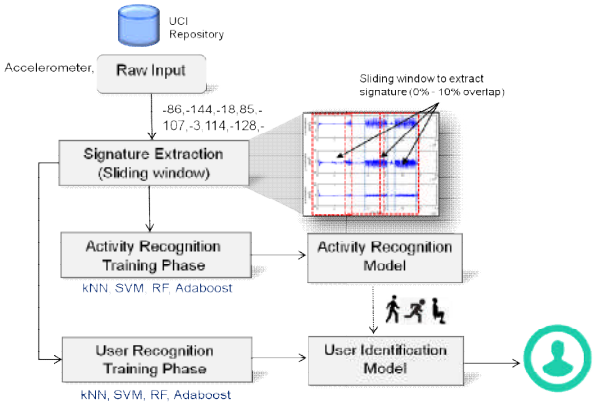
What can we learn from the accelerometer data? A close look into privacy

Team Member: Devu Manikantan Shila

Abstract: A handful of research efforts nowadays focus on gathering and analyzing the data from the end devices such as wearable's, smart phones to understand various user patterns and then customize their solutions based on the identified user patterns (e.g., health care industries monitors the walking pattern of the patients for early disease diagnosis). A key question is: *what else could we learn from the data besides the activity pattern*? The objective of this project is to apply state-of-the-art machine learning techniques on the raw activity (aka gait) data collected from the wearable devices (chest mounted accelerometer and also accelerometer mounted at multiple body locations) to recognize the ``user'' performing the specific activity. The proposed approach is based on a multi-layer (2-layer) classification problem: (a) In the *first* layer, we will identify the gait (irrespective of the user) and map into the most probable gait label; (b) In the *second* layer, we will identify the user with regard to the identified gait with a certain level of confidence. This project leverages supervised learning techniques such as Adaboost, SVM, kNN, Random Forest trees, NB for the multi-layer classification problem. For the experiments, the datasets from UCI repository [1, 2] were employed. The dataset mainly consists of the raw tri-axial acceleration (acceleration measured in three spatial dimensions *x, y and z*). The three dimensional data mainly captures the acceleration of the person's body, gravity, external forces like vibration of the accelerometer device and sensor noise; these characteristics may vary from one activity (or user) to another and serve as a useful measure for distinguishing users and activities. The experiment results showed that Random Forest and Adaboost performed well with identifying activities (accuracy of 82% for dataset 1 and 99% for dataset 2) and users (accuracy of 99% for datasets 1 and 2). We envision that this research project will have two key advantages: First, design a machine learning based technique for recognizing users based on the gait rather than relying on biometrics (fingerprints, facial, voice) or passwords/PINs. Second, enables researchers to think in a new direction: should we randomize or anonymize data in such a manner only the gait pattern can be learned without violating (leaking) the user privacy?

Approach: The proposed effort mainly encompasses three components: (a) *data gathering* - identifying the right dataset to use for gait and user classification experiments; (b) *signature (`feature') extraction*- deriving the right set of features for the machine learning algorithms from the raw tri-axial accelerometer data; (c) *learning and cross-validation of machine learning models*: identifying the right set of models and training the data on the "*training set*" and validating using "*test set*". The figure to the right shows our approach graphically.



Data gathering: We used the publicly available datasets from the UCI repository [1, 2]. Two datasets were used to confirm our findings related to gait/activity based user recognition: (dataset #1) is obtained from the wearable accelerometer mounted on the chest [1] and (dataset #2) is obtained from the wearable accelerometer mounted on four body locations – waist, left thigh, right arm and right ankle [2]. (***Dataset #***

1): The original dataset from [1] is collected from 15 participants (15 files, each belonging to a participant), performing seven activities (*Working at Computer, Standing Up, Walking and Going up down stairs, Standing, Walking, Going up down Stairs, Walking and Talking with Someone, Talking while Standing*). Due to intensive computing requirements, we used the data belonging to 10 participants (files). Each participant file consists of the following information: sequential number, *x* acceleration, *y* acceleration, and *z* acceleration and activity labels. The total number of samples per file (*Row*) differs and ranges from 120K to 160K and the number of dimensions (Columns) is 3 (excluding gait labels). The sampling frequency of the accelerometer is 52Hz. ***(Dataset # 2***): The dataset consists of 12-feature vector with time and frequency domain variables corresponding to tri-axial accelerations from four parts of the body. The real size of the dataset is 160K and each file consists of the following information: user, gender, age, height, weight, BMI, 12-feature vector. There are total of 5 activities (*sitting, walking, sitting down, standing and standing up*). The sampling frequency of the accelerometer was assumed to be 50Hz.

Feature extraction: The dataset consists of raw tri-axial accelerometer data and hence one may need to extract the useful features from this raw data to help identify the gait and the user performing the gait. The raw acceleration signals were first pre-processed by applying noise filters and are then separated into parts of several seconds using a fixed-width sliding window approach with 0-10% overlapping rectangular windows (using 5 *seconds* sliding window and sampling frequency of 50-52 Hz, we have 250-260 readings/window). Alternatively, original signal of length l is divided into segments of length *t*, and we used a length of 5 *seconds* for *t* (based on literature review, observed that we need to capture at least 5 *second* signal to extract the gait and corresponding user signature accurately). The segments at this stage are still represented as time series and hence, features are required to be extracted for each 5-*second* window. For dataset #1 and dataset #1, we extracted 24 and 36 statistical features, respectively, using the following metrics: RMS (root mean square of the *x, y and z* signal), signal correlation coefficient (correlation between *xy, yz and xz* signals), cross correlation (similarity between two waveforms), FFT (maximum and minimum of Fast Fourier transforms), vector magnitude (signal and differential vector magnitude), maximum, minimum, binned distribution (relative histogram distribution in linear spaced bins between the minimum and the maximum acceleration in the segment), zero crossings (number of sign changes in the window) and information entropy (a recommended metric to differentiate between signals that correspond to different activity patterns but similar energy signals). The statistical signature (feature) extraction module is implemented in MATLAB.

Machine learning models: As mentioned earlier, the proposed approach consists of two phases: (a) gait recognition; (b) user recognition based on the gait. Therefore, we call this approach as a two-layer multi-classification problem, where given the statistical features extracted from the 5 *second* test data sample, the model shall be able to identify the gait of the person and then use that results to identify the person performing the specific gait. Before training the model using the machine learning algorithms, the preprocessed datasets (#1,#2) are partitioned into two sets: (a) activity training set: XTRAIN with feature vectors and YTRAIN with activity labels; (b) user training set for each activity: XTRAIN with features and YTRAIN with user label performing a particular activity. To avoid the problem of over-fitting, each training set is further partitioned into testing and training data using the *cross_validation* package from Python Scikit. We have evaluated three cases: holding out 20%, 30% and 40% of the data for testing (evaluating) our classifiers. We used kNN, Adaboost, SVM, Random Forest Trees and Naïve Bayes algorithms for the classification purpose. Our experiments showed that the Naïve Bayes performed worst with 45% testing accuracy score and so, the results corresponding to Naïve Bayes are omitted from the tables and the discussion below. All the models were implemented in Python using the scikit machine

learning library. The performance of algorithms on recognizing gait and users was independently measured using confusion matrices (enabled us to extract the features that will distinguish two classes), testing accuracy, F1-score. The observations (accuracy and F1 scores) are given below for each dataset.

Optimal parameters for classifiers: Table [1] shows the parameters used for the classification algorithms. For instance, we used a Radial Basis Function (RBF) kernel for SVMs and a parameter selection using grid search from the Python's *GridSearchCV* package giving the combination of C=1 and Gamma = 0.001. Similarly, for Random Forest, Adaboost and kNN, using *sckit-learn*, we found the optimal values for the parameters *n_estimators*, *n_neighbors* by looping through a range of values and calculating the accuracy based on the holdout data. Furthermore, for kNN, we used a *uniform* weighing function that gives equal importance for all the neighboring *k* points. Besides parameter estimators, Tree based feature selection algorithm from *sklearn.ensemble* package was used to disregard irrelevant features by computing feature importances and to improve our running time. Though the tree-based selection algorithm produced low dimensional features (25% dimension reduction) for both dataset # 1 and #2, we found that using the reduced set of features corresponded to lower classification performance (4% drop in accuracy scores) for Random Forest classifier. Throughout our experiments, no feature selection algorithms were employed.

Models	Dataset # 1	Dataset#2
kNN	n_neighbors = 7	n_neighbors = 10
Adaboost	n_estimators =1000	n_estimators =1000
SVM	C=1, Gamma=0.001	C=1, Gamma = 0.001
Random Forest	n_estimators = 300	n_estimators = 150

Table 1: Optimal classifier parameters used for the experiments

Experiment Results:

1. (*Dataset # 1*): The sample and feature size for activity training set is (7*k* X 24). Once the activity is determined, only the file corresponding to activity class is trained and tested for person identification. The sample size of the user training set ranges from (*1k-2k* X 24). The classification algorithms generally performed well with training accuracy (gait and user identification) ranging from 0.99 to 1.0. However, we observed activity testing accuracy of an average 0.82 (see Figure [1]) for various classifiers (almost all classifiers produced the same behavior). For further reasoning of the results, we used the F1 score to understand the gaits/activities that were hard to recognize or contributed to the low scores. It

ML Models	Cross Validation 20%	30%	40%
kNN	0.82669	0.81717	0.80908
Adaboost	0.819277	0.831995	0.824837
SVM	0.821	0.81238	0.81327
Random Forest	0.819277	0.8214947	0.8276181

Figure 1: Testing accuracy of activity classification for CV splits

Cross Validation	Activity Classes: Working at Computer	Standing up, Walking and Going up-down Stairs	Standing	Walking	Going Up-down stairs	Walking and talking with someone	Talking while standing
20%	0.952218	0.478	0.61157	0.7821	0.45	0.40677	0.87064
30%	0.9522	0.4179	0.640211	0.77522	0.42	0.4782	0.87305
40%	0.95263	0.35294	0.63095	0.79373	0.41558	0.42718	0.878732

Figure 2: F1 scores of each activity (based on Adaboost) for various CV splits

stems from Figure [2] that classes 2, 5 and 6 performed the worst (scores of 0.35 – 0.45). Figures [3]-[4] show the classifier performance in classifying the user based on each activity for 20% and 30% cross validation. Generally, omitting activity 2, the algorithms performed very well in identifying the user (e.g., Random Forest gave user identification accuracy of 0.96 to 1). A close observation of activity 2 shows that it is a combination of several activities such as standing up, walking, going up-down stairs etc and that may be one of the reason the classifiers were unable to identify it properly.

	Activity Classes						
ML Models	Working at Computer	Standing up, Walking and Going up-down Stairs	Standing	Walking	Going Up-down stairs	Walking and talking with someone	Talking while standing
kNN	0.96928	0.84615	0.9921	0.994652	1	0.9787	1
Adaboost	0.959044	0.80769	0.96875	0.99465	0.96153	1	0.9923
SVM	0.94285	0.75	0.9332	0.9729	0.857	0.9457	0.95
Random Forest	0.969283	0.8846	1	0.994652	1	1	0.99744

Figure 3: Testing accuracy of user classification for 20% CV

	Activity Classes						
Person ID	Working at Computer	Standing up, Walking and Going up-down Stairs	Standing	Walking	Going Up-down stairs	Walking and talking with someone	Talking while standing
1	1	0.67	0.95	0.99	0.881	1	1
2	0.94252	0.8	0.94	1	1	1	0.96
3	0.96969	1	-	0.99	0.86	1	0.99
4	0.9842	0.34	1	1	0.86	1	0.987
5	0.96969	1	0.95	1	0.86	1	1
6	0.9523	0.67	1	1	1	1	0.97
7	0.97435	0.89	0.956	1	1	1	1
8	0.97916	0.86	1	1	1	1	1
9	0.9629	1	1	1	1	1	1
10	0.963636	1	1	1	1	1	1

Figure 4: F1 scores of identifying user/activity (based on Adaboost) for 30% CV

In short, the user classification performed very well compared to the activity classification and regarding the classifiers, Random Forest and Adaboost performed the best. One reason for the worst performance of activity classifier (classes 2, 5 and 6) will be the inaccuracy of the activity data itself (as said earlier, some activities are combinations of 2 or more activities). Other reasoning behind this observation may be the in-sufficient information provided by the single chest mounted accelerometer. This also implies that we might be able to obtain more accurate results, if multiple mounted wearable accelerometers are used.

	Cross Validation			
ML Models	10%	20%	30%	40%
kNN	0.99034	0.991155	0.99086	0.99043
Adaboost	0.99673	0.996679	0.996619	0.96609
SVM	0.97345	0.97498	0.97475	0.9741
Random Forest	0.994626	0.99471	0.994747	0.99396

Figure 5: Testing accuracy of activity classification for CV splits (10%-40%)

	Activity Classes				
Cross Validation	Sitting	Walking	Sitting Down	Standing	Standing Up
40%	0.9995	0.99655	0.9859	0.99755	0.98425
30%	0.999602	0.99706	0.987	0.99787	0.9862
20%	0.99603	0.997081	0.98798	0.99791	0.98817

Figure 6: F1 scores of each activity (based on Adaboost) for various CV splits

2. (*Dataset #2*): The observations from *dataset # 1* motivated us to use data from multiple mounted accelerometers [2]. The sample and feature size for activity training set is (10*k* X 36). The classification algorithms generally performed well with training (gait and user identification) accuracy ranging from

0.995 to 1.0. The testing accuracy (gait and user identification) also performed very well with an average of 99%, which corroborated our findings *that multiple accelerometers placed at various parts of the body and fewer (no) combinations of activities* may help to improve the classification accuracy. Among the algorithms, Random Forest and Adaboost gave the best performance [Figure 5]. For detailed understanding of the results, the F1 scores for various activities are given in Figure [6]. Figure [7] shows the classifier performance in classifying the user based on each activity for 20% - 40% cross validation splits. Generally, the algorithms performed very well in identifying the user (e.g., Random Forest gave accuracy score of 0.97 to 1). A close observation shows that users based on activity 2 (walking) were hard to recognize, compared to other activities.

	Cross Validation (20%-40%)														
	Activity 1			Activity 2			Activity 3			Activity 4			Activity 5		
ML Models	20%	30%	40%	20%	30%	40%	20%	30%	40%	40%	30%	20%	20%	30%	40%
kNN	0.9997	0.9998	0.9996	0.97304	0.97165	0.96964	0.998	0.998	0.9985	0.99842	0.9988	0.99873	0.98913	0.98926	0.98832
Adaboost	0.99408	0.9995	0.9996	0.9917	0.99078	0.98922	0.991	0.999	0.9983	0.99979	0.99971	0.99978	0.99756	0.99758	0.99778
SVM	0.96789	0.9679	0.9672	0.94673	0.94018	0.94016	0.969	0.96	0.9601	0.97809	0.97881	0.97732	0.9489	0.9493	0.94791
Random Forest	1	1	1	0.98583	0.9841	0.98179	0.998	0.998	0.9983	0.99931	0.99937	0.99916	0.99557	0.99678	0.99718

Figure 7: Testing accuracy of user classification for various activities given 20%- 40% CV splits

The F1 Scores of the user identification for various activities is given in Figure [8]. Compared to the various activities, user recognition based on walking provided an average of 98% accuracy.

	Activity Classes				
Person ID	Sitting	Walking	Sitting Down	Standing	Standing Up
1	0.999681	0.99257	0.9982	0.99974	0.999018
2	0.999648	0.98877	0.99752	0.99965	0.99662
3	0.999221	0.98786	0.99965	0.9999	0.9979
4	0.999671	0.9798	0.99653	1	0.99642

Figure 8: F1 scores of identifying user/activity (based on Adaboost) for 30% CV

3. Confusion Matrices: Figures 9 (a) and (b) corresponds to dataset #1 and the remaining graphs 9(c) and (d) corresponds to dataset #2. The confusion matrices (Figure [9]) clearly show that the performance of dataset # 2 activity classification outweighs dataset #1. Specifically, from 9(a), we observe that classes 2, 5 and 6 performed worst (maps to F1 scores in Figure [2]). Surprisingly, for both datasets, user identification performed very well, which indeed proves our concern related to privacy.

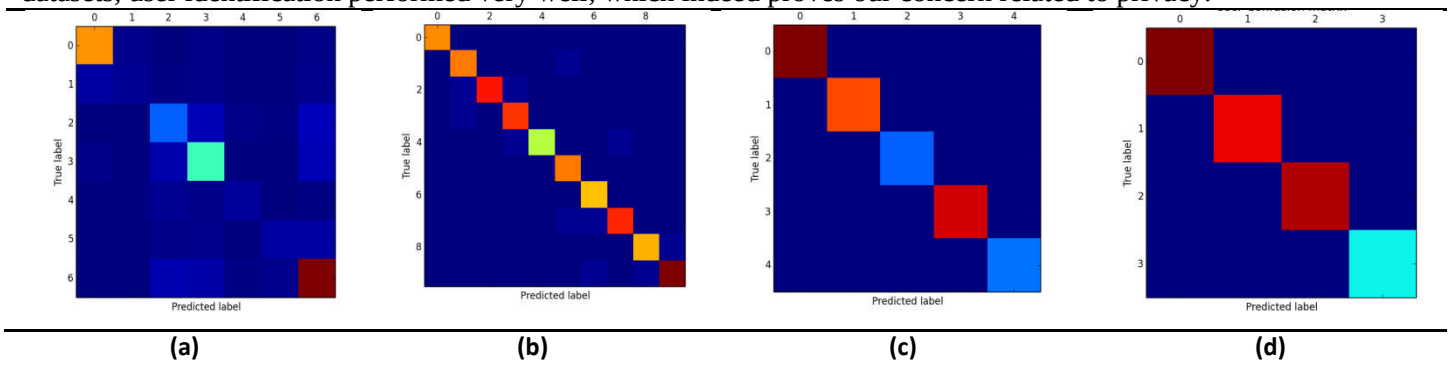


Figure 9: Confusion Matrices: (a) Dataset #1: activity classification (seven classes); (b) Dataset #1: user classification based on activity 1; (c) Dataset #2: activity classification (five classes); (d) Dataset #2: User classification based on activity 1

Future Work: In future, we would like to apply unsupervised learning techniques such as mixture of Gaussians and also, extract more useful features such as the speed, acceleration signal signs to improve the classification rates in a less user-interrupting manner. We will investigate the performance of our classifiers exposed to varying user behaviors (e.g., variable walking speeds depending on shoes).

References:

[1] https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer
[2] Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. *Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements*, in the proceedings of 21st SBIA, 2012.
[3] Python Scikit, http://scikit-learn.org/stable/index.html