Biometric ID, based on Phone Accelerometer usage

Govardana Sachithanandam Ramachandran

Introduction

Two level authentications are slowly getting adapted in mainstream. Since password based authentication are vulnerable. Prime example would be Sunet authentication at Stanford. Since smart phones are getting very pervasive and since everyone moves differently when they use their phone. This project aims to use phone accelerometer usage data for validating a user.

This type of authentication is very subtle, compare to finger print & facial recognition since finger print or face can be captured or forged easily by different means.

This project is an adaptation of the kaggle.com competition - "Accelerometer Biometric Competition" (http://www.kaggle.com/c/accelerometer-biometric-competition) . This project investigates the feasibility of using accelerometer data as a biometric for identifying users of mobile devices.

Data

Seal(the kaggle sponsor) has collected accelerometer data from several hundred users over a period of several months during normal device usage. To collect the data, Seal has published an app on Google's Android PlayStore that samples accelerometer data in the background and posts it to a central database for analysis.

They have uploaded approximately 60 million unique samples of accelerometer data collected from 387 different devices. These are split into equal sets for training and test. Samples in the training set are labeled with the unique device from which the data was collected. The test set is demarcated into 90k sequences of consecutive samples from one device. The training data is of the form

Field name	Description
T	Unix time (milliseconds since 1/1/1970)
X	Acceleration measured in g on x co-ordinate
Y	Acceleration measured in g on y co-ordinate
Z	Acceleration measured in g on z co-ordinate
DeviceId	Unique Id of the device that generated the samples

The test set has T,X, Y, Z & sequence Id. The objective was to determine if the sequence id is that of a device ID.

Leaks

1. It was apparent from the start there were potential leaks in the test & train data. The most apparent ones being the samples from a device were equally divided equally into test & train set. Just the numbers of sample alone were enough to predict the results with very high accuracy.
2. The sampling on a device was done almost at the same time of the day. Hence time had high correlation.

I decided not to use these features & others which I believed that had leaks.

Feature selection

One of the challenges faced, was to convert the time series data into feature vectors. It was observed that there were series of dense period of activities. It was decided to break the time series data into Segments. Start of session is defined by time between activity is more than a second. Below is an example of breaking series into Segments

T	X	Y	Z	Device	Segment
1336645084843	0	8.539958	4.372131	7	1
1336645085030	0.272407	8.430995	4.290409	7	
…	…	…	…	7	
1336645087668	0.313268	8.308413	4.099723	7	
1336645087918	0	8.62168	4.290409	7	
1336645116579	-0.88532	8.049625	4.944186	7	2
1336645116719	-0.91256	8.117727	5.012287	7	
..	..	..	..	7	
1336645129705	-0.42223	8.117727	4.521955	7	
1336645145770	-0.65378	8.240311	4.33127	7	3
1336645146026	-1.07601	8.349273	4.603676	7	
1336645146209	-0.91256	8.19945	4.603676	7	

Tab: Segmenting sequence for aggregative features& for better prediction

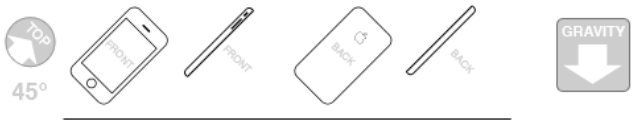
Some measures to capture the characteristic of each segment were devised. These were:

Distribution - position	mean values of X,Y & Z
	Variance of values of X, Y & Z
	Magnitude $(X^2 + Y^2 + Z^2)^{1/2}$
Distribution - rate of change	mean - rate of change in acceleration on (X,Y & Z axis)
	Variance - rate of change in acceleration on (X,Y & Z axis)
Temporal	Duration of the session
	# of activities in a session
	mean time between 2 activities
	Variance in time between 2 activities

Tab: Features that were considered

Model selection

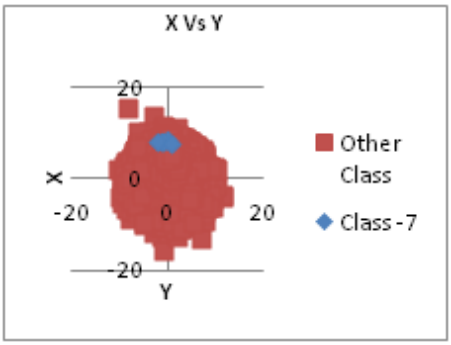
Model Position:

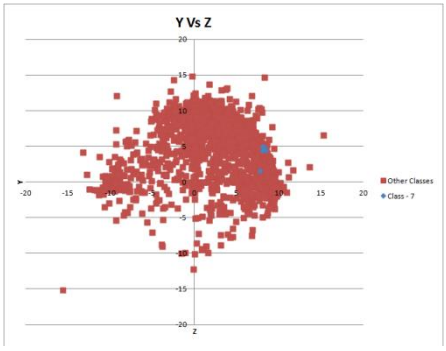


x =	0.75	x =	0	x =	-0.75	x =	0
y =	-0.75	y =	0.75	y =	-0.75	y =	-0.75
z =	0	z =	0.75	z =	0	z =	-0.75

Fig: Phone positions & their corresponding Accelerometer values

Position with which a user holds the phone was found to be strongest compared to any other feature. This was very apparent when running mutual information & PCA of the positions against the labels





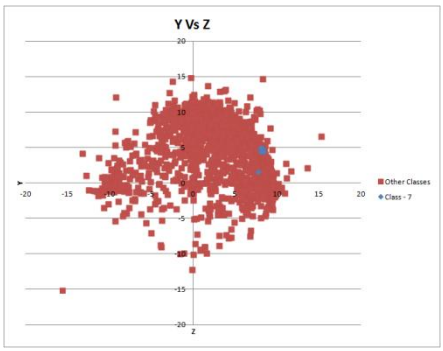


Fig: X,Y,Z segment mean correlation between themselves

Other features though sounded promising, later found to be noisy & didn't have signal in them to be selected as features. These includes Variance, rate of changes of G, time variance etc.,

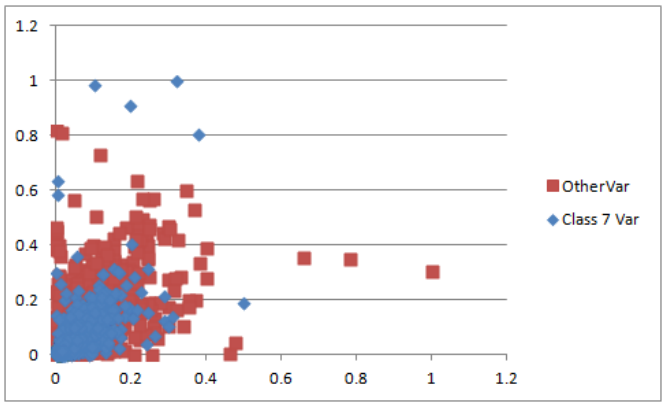


Fig: X-axis Variance distribution

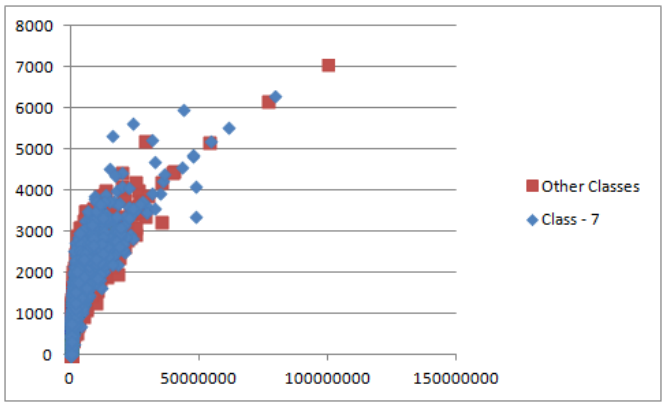


Fig: Time mean Vs Variance distribution

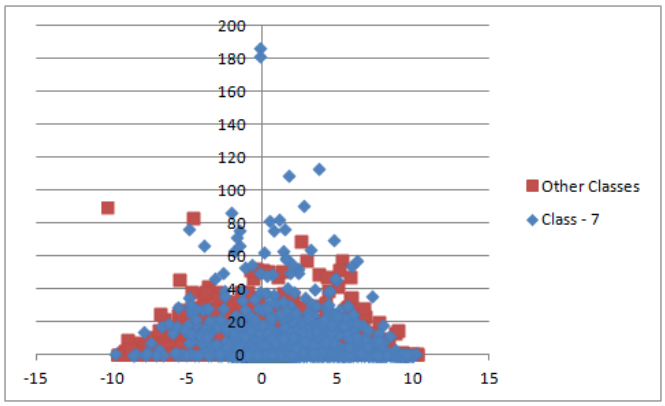


Fig: Normalized rate of change

As first cut, the problem was reduced to multi-class multivariate binomial classifier. As it made more engineering sense, to have each device store their respective model(s) locally on the phone. But It soon became apparent that the sample from 'Other Class' had very high density & started to skew the results.

Since it was obviously clear there could be no linear boundaries between the above selected features and clearly there were correlation between axes, SVR was used to model the problem. The reason to use SVR instead of SVM(SVC) is that it provide probabilistic interpretation. Since each test sequence has almost 750 segments. The probabilities of each of these segments are calculated separately. The probability for the entire sequence is determined by taking the log likelihood of each of the segment classes predicted for the sequence.

Monte Carlo by grid search was used on 10-fold cross validation set to determine the parameters for the model while maximizing the accuracy of the results.

$$\text{argmax}(\text{Accuracy})$$
$$C, \gamma, \varepsilon, p$$
$$\text{s.t } fp <= .01$$

Below are the results for individual segment

Model	SVR
Kernel	RFB
C	8
γ	.18
ε	1e-3

Tab: Model parameters

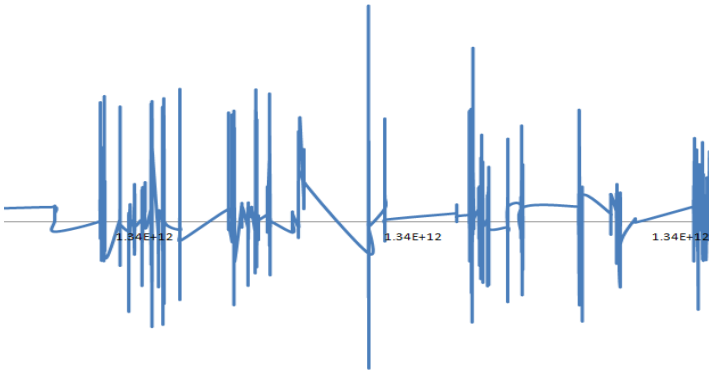
Measures	Value
Accuracy	0.85323
F-measure	2.1533

Tab: Results for individual segments

Measures	Value
Accuracy	0.954

Tab: Results for entire sequence (for 10 classes)

Model Harmonics:



From analyzing the input signals, It was apparent that there might be some harmonics either voluntary or involuntary emitted, which might be unique to the user. It seems body has natural frequency of 5 – 80 Hz , To capture this the sample rate as to at least 125ms. Due to loss of data sample rate was set to 250ms.

Since the signals seems to have been sampled at varying sample rate. Non-Uniform Discrete Fourier Transformation was used to transpose to frequency domain.

As promising as it had appeared, the result were not that hopeful, this was due to the fact that they are very low variance, after Z-score normalizing most of the frequencies' amplitude were the constant. A better approach would have been to get the first 5 order of harmonics as features.

It was observed that here few correlation though on two frequency spectrums

Fig: Correlation of 3-Classes on 2 frequency spectrums

Parameters	Values
Kernel	Sigmoid
C	0.01
γ	0.01
ε	1e-3

Tab: Parameters of the model

Measures	Values
F-Measure	0.9544699
Accuracy	0.704407

Tab: The segment level measures where

Model by Relative state change:

It might be argued that one might have quirks that would make the mobile move in a pattern between the axes. This would require pattern recognition.
For this I've hypothesized the model as viterbi formulation of Hidden Markov Model
Given a series of observed series of vectors [x y z]

$$\arg\max_{s_1,\dots s_T} P(s_1,\dots s_T|o_1,\dots o_T)$$

$$= \frac{\alpha_t(s)\beta_t(s)}{\sum_{s'}\alpha_T(s')}$$

The above formulation would provide the most likely state at T. Again here the Segmented sequence of events are used. While training, Since the states are know each of the observed vector. Instead of using just the Believes the actual states are use for training.

Baum-Welch algorithm is used to train the Hidden Markov Model.

Parameters	Comments
$P(S_i)=1/N$	Where N is # of classes
$P(A_{ij})=0 \ \forall i \neq j$	
$P(A_{ij})=1 \ \forall i = j$	
$P(S_{T-1}..S_1 \mid O_1..O_T)=1$	This is done while training. As these are labeled set

Tab: Model parameters

In order to avoid exuberant event & state transition matrix, The [x y z] are pruned to first 2 significant digits

Ensemble

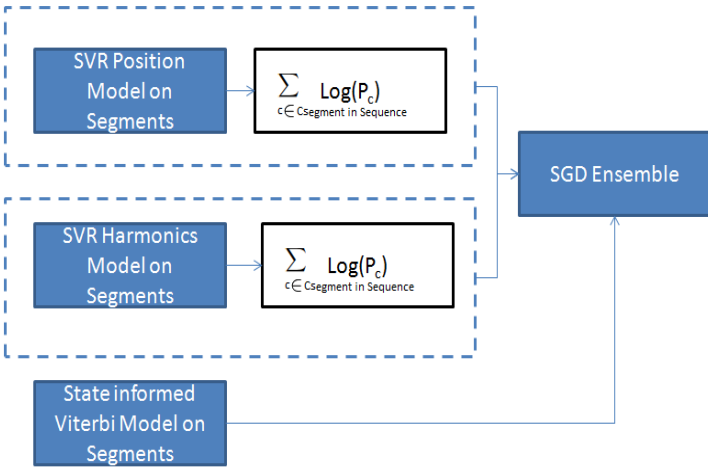


Fig: The Ensemble Models

The result of these models where then fed to an Ensemble classifier. SDG logistic regression was use to determine the final results.

Conclusion

Though there were some obvious leaks in the dataset. With accuracy >0.98 clearly shows the possibility of further research & adaptation into real world. The one obvious thing about the dataset is there were almost 90,000 sample for a device, In real world application this might not be possible, Further advancement could be done to help reduce the sample required to detect fraud.

It might also be noted the sample doesn't fully profiles the user as it was taken at one instance in time. The user might as well use the device on a table top, the device position will be fixed and there would be very few signals to go with, all the above models will fail. Harmonics analysis showed the importance of accuracy and sample rate to capture human subtle signal as noise from device overwhelms it. One thing this competition has showed is the importance of accurate sampling of data set.

Reference

1. "The Viterbi Algorithm" by "G. DAVID FORNEY, JR." - http://cbio.ensmp.fr/~jvert/svn/bibli/local/Forney1973Viterbi.pdf
2. "Whole-Body Vibration Building Awareness in SH&E" by Helmut W. Paschold and Alan G. Mayton http://www.asse.org/professionalsafety/pastissues/056/04/030_035_F1Paschold_0411Z.pdf
3. "Machine Learning Paradigms for Speech Recognition : An Overview" by Li Deng, Fellow, I and Xiao Li http://research.microsoft.com/pubs/189008/tasl-deng-2244083-x_2.pdf
4. "An Introduction to Hidden Markov Models by L. R. .Rabiner and B. H. Juang" http://blog.digitalagua.com/2008/07/15/accelerometer-xyz-based-on-iphone-position/
5. "Accelerometer Biometric Competition" by "Seal Mobile ID" http://www.kaggle.com/c/accelerometer-biometric-competition
6. "Baum–Welch algorithm " http://en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm