Detailed Analysis of Papers

#1. Modeling Free-form Handwriting Gesture User Authentication for Android Smartphones
By : Floren Alexis, Guillermo Gohan, Larry A. Vea

This paper aimed to model a free-form handwriting gesture user authentication mechanism with the integration of some other factors that may influence user identification.
Data Collection : They gathered data from 30 volunteer participants, ages 18 years old and above. Each participant were asked to draw a pattern 10 times and store with the participant names. In which 8 out of the 10 security patterns were used as part of the training set while the other 2 were used as part of the test set.

Working : Using a RapidMiner tool, model development was conducted by training some well-known classifiers such as Decision Tree, Naive Bayes and Neural Networks, and using 10-fold cross-validation to validate the model.we trained the features on mentioned classifiers using three features sets:
(1) using dynamic features only
(2) using static features only;
(3) using the combined static and dynamic features.
This was done to determine which feature-set gives the most acceptable model. Initial results show that the Neural Network classifier performs the best and followed closely by Naïve Bayes classifier.

Static features : Height, Width, Total pixel cover, Ratio of width and height etc.
Dynamic Feature : Size of fingerprint, Pressure, Timestamp, Total finger uptime, Total finger downtime, Speed.
Result : The model showed a very promising recognition rate of 96.67%.

#2. A Study of Touching Behavior for Authentication in Touch Screen Smart Devices
By : Ala Abdulhakim Alariki, Azizah Bt Abdul Manaf.

Data Collection : They Collected the raw data from 18 user whose age ranging from 20 to 40 and each user has 6 instances. These instances include finger size, pressure, x-coordinate, y-coordinate, time taken, Pressure and acceleration.

Working : Paper distributes working in three main phases :
(1) android application phase : from collection to storing of all the data (touch gesture) into a database for extracting the captured features.

(2) Microsoft Excel phase : Make the data into CSV file, which will be inserted into Weka tool for the most significant features extraction.
(3) Weka tool phase : Making a comparison between the current user's behaviors with relevant generated authentication signatures to decide if the user is a genuine or an impostor. They mainly focused on Random Forest classifier, which consists of many decision trees and outputs for the individual trees.

Result : Acceleration feature correctly classified 100 instances with 92.59% accuracy. In addition, time feature correctly classified 99 instances with 91.67% accuracy.Based on the obtained result both features acceleration and time have a superior performance compared to all other features. The best result obtained was combining all features which correctly classified 106 instances with 98.14% accuracy.

#3. A Behavioral Authentication Method for Mobile Gesture Against Resilient User Posture
By : Qiang Liu, Mimi Wang, Peihai Zhao, Chungang Yan, Zhijun Ding

This paper aimed to model a gesture user authentication mechanism with taking into consideration of user posture like standing, sitting, lying, walking, and the position of equipment containing hold in hand, put on the table which may influence user identification.

Data Collection : They are collecting two kinds of behavioral data.The first kind of data includes touch screen data which includes the X, Y coordinate of finger position, pressure, size of contact area and the timestamp .The other kind of data includes the X, Y, Z coordinate of screen orientation, mobile's acceleration(acc x , acc y , acc z).

Working : They used posture cluster algorithm, which uses K-means algorithm and silhouette coefficient to cluster user's postures on the data collected by mobile's orientation sensor and acceleration sensor, and gives a method to determine whether a posture is belonged to the legal user or not. Then they train a gesture authentication sub-model for each posture based on data collected by touch screen. The model judges the rightness of inputted password pattern firstly, and then legitimacy of current posture. If the posture is legal, then it uses the gesture authentication sub-model of the posture to judge the legitimacy of gesture behavior.

Result : THis model achieves 4.36% False Acceptance Rate (FAR) and 5.03% False Rejection Rate (FRR).

#4. Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication
By: Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song.

They proposed a model to authenticate users while they perform basic navigation steps on a touchscreen device and without any dedicated and explicit security action that requires attention from the user. Our goal is to analyze how robustly such schemes operate and if they are sufficiently reliable to be used on commodity devices.

Data Collection : Experiment carried out on Android phones where users asked first to read some text and based on that give some answer and second compare images. The main purpose of this experiment was to motivate users to produce many navigational strokes in a natural way. After one week later same study was conducted. There were 41 participants in the study and four different smart phones with similar specifications were used where each participants spent around 25 to 50 minutes in reading and answering the and 3 to 4 minutes to compare images. During experiment 30 features of each participants recorded like: for each finger x,y coordinates, pressure on screen, area covered, orientation of screen, velocity etc.

Working : For the classification purpose they have used -nearest-neighbors(kNN) and a support-vector machine with an rbf-kernel (SVM).

Result : The classifier achieves a median equal error rate of 0% for intrasession authentication, 2%–3% for intersession authentication, and below 4% when the authentication test was carried out one week after the enrollment phase.

#5. Implicit Authentication On Mobile Devices
By : Stockinger, Tobias.

In this paper author explains about the importance of the phone popularity and due to which sensitive information like confidential documents are highly insecure without a protected authentication system in the smartphones where author explains about password authentication are highly unreliable therefore author suggested continuous authentication for the smartphones. Author introduces his finger gesture authentication system which uses touch data features like finger pressure,trajectory,speed,acceleration ,matching x,y coordinates.
In the end this method hasn't completely mitigate the unauthorized use and the scheme also take much more time and not suitable for instantaneous authentication.

#6. Touch Gestures Based Biometric Authentication Scheme for Touchscreen Mobile Phones
By: Yuxin Meng, Duncan S. Wong, Roman Schlegel, and Lam-for Kwok

Proposed touch gestures based biometric authentication system.This system utilizes only simple touchscreen data like input-type, x-coordinate, y-coordinate and system time.
The system collects all touch gesture data during 10 minutes and extracts 21 types of features and features are characterized according to drawing directions.In their experiments, they collected and analyzed touch gesture data of 20 users.And they optimized the neural network classifier by using Particle Swarm Optimization to deal with variations in user’s usage pattern.

As a result,the average error rate of PSO was 3%. However, in this experiments,the touch gesture data is affected on his or her using service during 10 minutes.And 20 user's data may be insufficient to evaluate the experimental results.

#7. Robust Gesture-Based Authentication for Mobile Systems
By: Can Liu, Gradeigh D. Clark, Janne Lindqvist

Datasets Used:
1. Freeform gesture dataset (http://dx.doi.org/10.1145/2594368.2594375)
2. Demo dataset (http://dx.doi.org/10.1145/1294211.1294238)
and many more.

Working : Proposed gesture authentication systems have implemented classification methods from Support Vector Machine to Dynamic Time Warping. Author have used the Protractor [32] recognizer, a popular algorithm for free-form gesture authentication to analyze the effects of three gesture invariances: scale, rotation, and location.Author have designed and implemented two novel Multi-Expert (ME) recognizers: Garda and SVM Garda recognizers. We also implemented another eleven popular recognizer methods: EDR, LCS, DTW, HMM groups, SVM etc. where each recogniser tested on 3 parameters:
1) Authentication performance (i.e. equal error rate) with five datasets.
2) Imitation attacks with two datasets.s
3) Brute force attacks.

Result : Shows that Garda achieved the lowest average error rate (0.015) for authentication performance, the significantly lowest average error rate (0.040) for imitation attacks, and resistance to all brute-force attacks.

We going to refer paper 1,3,7 mostly but we also take some references from other research papers.

Week 5 :

As we have discussed last time you have told us regarding partitioning of dataset so that dataset become unbiased. So we have partition dataset according to no of user, Here we count how many instances of user1 have occurred now take the same no of instances for the remaining ones and make a dataset. Similarly we have done this for all the users and use weka tool to check the accuracy. For testing purpose we have rbf algorithm which gives around 70% accuracy.

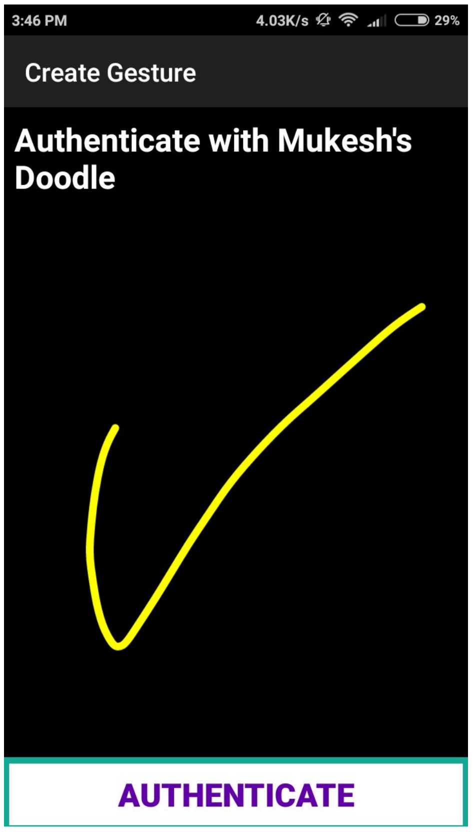
Now we are developing an android application which can collect finger gesture while interacting with phone. Here we are collecting data like x,y coordinates, timestamp (time duration to perform action), Area covered , pressure, velocity etc. Except pressure we have collected these data and calculated mean and variance for all of these.

For calculating pressure we need sensor which can gives us the pressure data but most of the does not come with the pressure sensor we are looking into alternate way to calculate the same.

Plans ahead :

After all of the feature collection which are mentioned above we will first test the dataset on weka for the accuracy and improvement then later we build a app for system login using human finger gesture.

App Screenshot :



Week 6

1. X, Y coordinate : View.getTop() View.getBottom(), View.getLeft(), View.getRight() this will give the coordinate wrt to parent view. or you could use event.getX() and event.getY(){ keyword : MotionEvent.ACTION_DOWN, MotionEvent.ACTION_UP}

http://www.41post.com/2382/programming/android-touchscreen-swipe-movement-detection

https://stackoverflow.com/questions/11315030how-can-i-get-both-start-and-ending-coordinates-on-android-touch-event-motion-e

2.direction:
https://stackoverflow.com/questions/3148741/how-to-capture-finger-movement-direction-in-android-phone

4.distance :
https://stackoverflow.com/questions/12559066/calculate-gesture-distance-in-android

5.velocity :
https://stackoverflow.com/questions/5815975/get-speed-of-a-ontouch-action-move-event-in-android

6.
https://developer.android.com/reference/android/view/MotionEvent.html
https://developer.android.com/training/gestures/movement.html(All in one solution)

http://www.mariofrank.net/touchalytics/extractFeatures.m this is matlab file to extracting feature

https://stuff.mit.edu/afs/sipb/project/android/docs/training/gestures/movement.html(read this)

List of features :

1.User_id

2.Start X

3.Start Y

4.Stop X

5.Stop Y

6.Stroke_duration(timestamp) - time to cover up swipe

7.Up/Down/Left/Right flag

8.Median velocity at last three point - when swipe is ending take last three points velocity and takes its median

9.Avg direction - we take it as from start to end point.

10.Length of trajectory

11.Avg. velocity - we can divide length of trajectory in 3 or 4 part takes velocity at any point in that part and takes mean of it.

12.Median acceleration at first 5 point - when swipe is ending take last five points acceleration and takes its median

14.Mid stroke pressure (we can select 5 points on trajectory and then take mean of it)

15.Mid stroke area covered(instead of this we can select 5 point on trajectory and then take mean of it)

16.Mid stroke finger orientation

17.Phone orientation

<u>Different Algorithms and its Result</u>

Algorithm	Correctly Classified Instances	Incorrectly Classified Instances	Root mean squared error
Naive Bayes	12.8906 %	87.1094 %	0.1822
BayesNet	57.0313 %	42.9688 %	0.115
Simple Logistic	60.1563 %	39.8438 %	0.1155
Multi class Classifier	61.0677 %	38.9323 %	0.1124
AdaboostM1	52.9948 %	47.0052 %	0.1371
Filtered Class Cla.	59.6354 %	40.3646 %	0.1134
Kstar	58.2031 %	41.7969 %	0.0783
LogitBoost	65.2344 %	34.7656 %	0.1057

REP Tree	60.026 %	39.974 %	0.115
Random Forest	80.7292 %	19.2708 %	0.0783

<u>AutoWeka Classifier Result:</u>

```
Classifier classifier = AbstractClassifier.forName("weka.classifiers.trees.RandomForest", new String[]{"-I", "10", "-K", "0", "-depth", "0"});
classifier.buildClassifier(instances);
```

```
Correctly Classified Instances         620               80.7292 %
Incorrectly Classified Instances       148               19.2708 %
Kappa statistic                  0.7291
Mean absolute error                    0.0134
Root mean squared error                0.0783
Relative absolute error                37.4605 %
Root relative squared error            59.0319 %
Total Number of Instances              768
```

Using motion event we can capture the following data(feature).

We have the list of motion events pointers.which gives x,y , time etc values for all the point.
Pointer=[(x1,y1,t1), (x2,y2,t2),...................................(xn,yn)]
velocity=x2-x1/t2-t1
acceleration=v2-v1/t2-t1