# Gesture Authentication Mechanism on Android

By
Pankaj Kumar Anuragi
(2014073)
&
Sudhir Kumar
(2014107)

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Engineering
On April 18th, 2018

BTP Track: Research cum Engineering

BTP Advisor
Dr. Arun Balaji Buduru

**Indraprastha Institute of Information Technology**
**New Delhi**

## Student's Declaration

I hereby declare that the work presented in the report entitled **"Gesture Authentication Mechanism on Android"** submitted by us for the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of Dr. Arun Balaji Buduru. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

**……..** **Place & Date: ……...**

**(Pankaj Kumar Anuragi)**

**……..**

**(Sudhir Kumar)**

## Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**……..** **Place & Date: ……...**

**(Dr. Arun Balaji Buduru)**

# Abstract

The Password Patterns are mostly used authentication scheme on android phones because of it is high usable and memorable, But in terms of security it is vulnerable to shoulder surfing and smudge attack. In this Project we are a proposing a Gesture based authentication mechanism. It works on the mainly how the user interacted with the phone using distinguishing features such as Area covered by finger, Pressure how much user insert on screen while interacting, finger velocity, acceleration, and stroke time and many more. Even if attackers see what gesture a user performs, they cannot reproduce the behavior of the user doing gestures through shoulder surfing or smudge attacks. Here we have built an Android app to collect user's data. We have used SVM( Support Vector machine) for classification and after a certain level of training it is able to correctly distinguish between the owner and any other person with a high level of accuracy of 92.3%.

## Acknowledgments

We would like to express our sincere gratitude to our supervisor Dr. Arun Balaji Buduru for providing their invaluable guidance, comments and suggestions throughout the course of the project. We would specially thank him for constantly motivating us to work harder.

## Work Distribution

We have maintained a week wise log report of what has been done and at which week we have what work we have done.

Link :
https://docs.google.com/spreadsheets/d/12TQr8c5OTuWidrn3m1BKIarrTRtYu3nmRCFzG3bVk50/edit#gid=0

# Contents

# Chapter 1

# Introduction

Nowadays, threats to personal information stored on mobile devices have begun to increase as these devices are more important in day-to-day life. Current authentication methods present their own set of concerns. Text-based passwords have many problems, ranging from password reuse to weak password selection. Pattern Based password vulnerable to shoulder surfing. Biometric methods have difficulties from revoking the authentication token to misidentification (e.g. cost associated with sensor, fingerprint scanner scanning a scarred finger).

This project presents the implementation of a real-time classification system to correctly distinguish between the legitimate and illegitimate user with a high level of accuracy. Our system uses the Motion sensors(Accelerometer, Gyroscope) data from the smartphone to achieve this objective.

Here we have developed an android application based on the gesture extracted from user while interacting with phone. At the time of registration users have to draw about 30 instances of a three different gesture patterns and save it. Now exit the system and try to authenticate by drawing the same gesture, Now depending on user it will show success or fail message.It implements a supervised learning classifier, which, after some passive training using the smartphone's Motion sensors( Accelerometer, Gyroscope) data can understand and differentiate between the owner and any other person.

# Chapter 2

# Related work

The idea of using Gesture for user authentication is not completely new, there has been work done and still going on.

Floren Alexis et al.[1] aimed to model a free-form gesture user authentication mechanism with the integration of some other factors that may influence user identification. 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. Initial results show that the Neural Network classifier performs the best giving the recognition rate of 97%.

Mario Frank et al. [2] proposed a user authentication system based on touch information of the user's scroll pattern. Thirty kinds of features are extracted from touchsc- reen scroll pattern, and they are used in continuous user authentication. k-NN (k-Ne- arest Neighbors) and SVM (Support Vector Machine) are utilized to classify the touc- hscreen scroll patterns of users. In their experiments, the EER (Equal Error Rate) of the system was between 0% and 4%.

Can Liu et al.[3] Proposed a gesture authentication systems have implemented classification methods from Support Vector Machine to Dynamic Time Warping. Author have used the Protractor [4] 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**.** 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.

Yuxin Meng et al.[5], Ala Abdulhakim Alariki et al.[6] and Zhiping ZHOU et al.[7], Alexander De Luca et al.[8] and many more have worked in this field and proposed a way for Gesture authentication.

# Chapter 3

# System Architecture

First you have to add yourself as a user. Then you have to add instances of gesture by writing the number from the given set on the screen and save it ,you have to draw around 30 instances of a single gesture for training purpose of classifier, for three different pattern like for two, three and four. Now when you try to authenticate you have to draw the same gesture as mention on the screen , a number from the set of three will pop up and have to draw the same number on the screen. Motion sensors(Accelerometer, Gyroscope) collects the data and match with existing data using SVM classifier, Depending on the legitimacy of user it will show the message of success or fail.

Furthermore we have also given an option to retrain the user which help to predict the legitimacy with more accuracy ( larger the dataset more the accuracy ).

# Chapter 4

# Data Acquisition and Annotation

Our data is mainly focused on the touch gesture data and the motionEvent data which contains information representing the interaction of the fingers of the user while using the device .The Android software development kit (SDK) directly provides a set of motion event features that can be collected to form datasets for touch biometrics. A motion event is triggered whenever a user touches the touchscreen.The motion event contains information such as the pressure applied to the screen and the coordinates of the touch and the other gesture android class provides the stroke data of the finger while interacting with the device. After collecting the data for the each user ,we tried to identify the true owner of the device by using the machine learning techniques and have used Support Vector Machine(SVM) to predict the best results.

- **Training Phase:**
  During the training phase we asked the user to draw a predefined pattern on the application we created and extracted the features for our model classification data.
  We mainly focused on the predefined pattern to write by the user as it's more important to use it for authentication ,As this will directly solves the shoulder surfing drawbacks of other authentication issues faced by other techniques.

- **Testing Phase:**
  During the testing, we will be authenticating any user by collecting single gesture which presenting previously predefined pattern and will apply of the classification model to find whether the test gesture is matching the true user by several important features of the gesture and motion event parameters.

# Chapter 5

# Data Preprocessing and Feature Extraction

We created two data-set for the processing,First data-set is collected by the MotionEvent class provided by the Android OS which deals with the motion of the pattern drawn on the screen. Second dataset is created from the gesture class which deals with the gesture activity of the user and follow the change in the values of the pressure on the screen ,gesture stroke parameters.

After experimenting and reviewing each value given by the two android classes we filtered some extremely important raw data for our processing which can be easily used and computable .

**Raw Data**
The gesture provided during the training and testing is made of up of a list of motion-Events and Stroke,and Provides these parameters :

- x,y coordinates (all points)
- Timestamp (for each point)
- Stroke bounding height
- Stroke bounding width
- Number of points in Gesture
- Pressure
- Touch Size
- Orientation

# Chapter 6

# Feature Selection

Feature selection for any predictive analytic model is very important . It can be used to identify and remove unneeded, irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model or may in fact decrease the accuracy of the model. Benefits of performing feature selection on the data:

- Reduces Overfitting
- Improves Accuracy
- Reduces Training Time

## Feature Selection in Weka

Weka provides an attribute selection tool. The process is separated into two parts:

- **Attribute Evaluator:** Method by which attribute subsets are assessed. We have used CFSSubsetEval Algorithm.

- **Search Method**: Method by which the space of possible subsets is searched. We have used Greedy Stepwise Algorithm. Uses a forward (additive) or backward (subtractive) step-wise strategy to navigate attribute subsets.

Final Features selected for are :

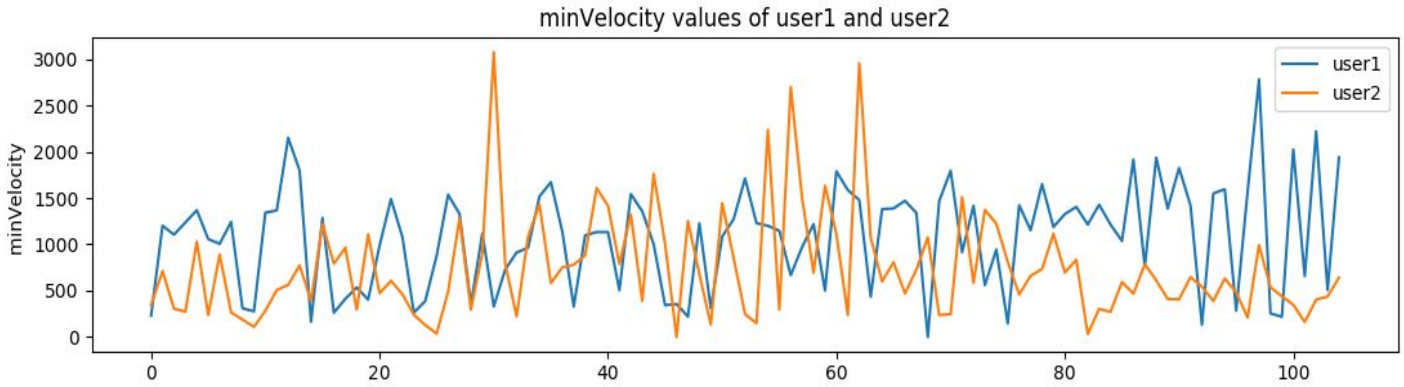
1. Start(X)
2. Start(Y)
3. End(X)
4. End(Y)
5. N : Number of event points or gesture points
6. Start Vx : for last n points
7. Start Vy : for last n points
8. End Vx : for last n points
9. End Vy : for last n points
10. Start Vavg : for first n points
11. End Vavg : for last n points
12. Max Velocity
13. Min Velocity
14. Timestamp
15. Start Pressure :for starting n points
16. End Pressure : for last n points
17. Start Touch Size :for starting n points
18. End Touch Size:for last n points
19. Start $Acc_x$ : for starting n points
20. End $Acc_x$ : for last n points
21. Start $Acc_y$ : for starting n points
22. End $Acc_y$ : for last n points
23. Start $Acc_{avg}$ : for last n points
24. End $Acc_{avg}$ : for last n points
25. Boundary height : height of the gesture
26. Boundary width : width of the gesture
27. Orientation

# Chapter 7

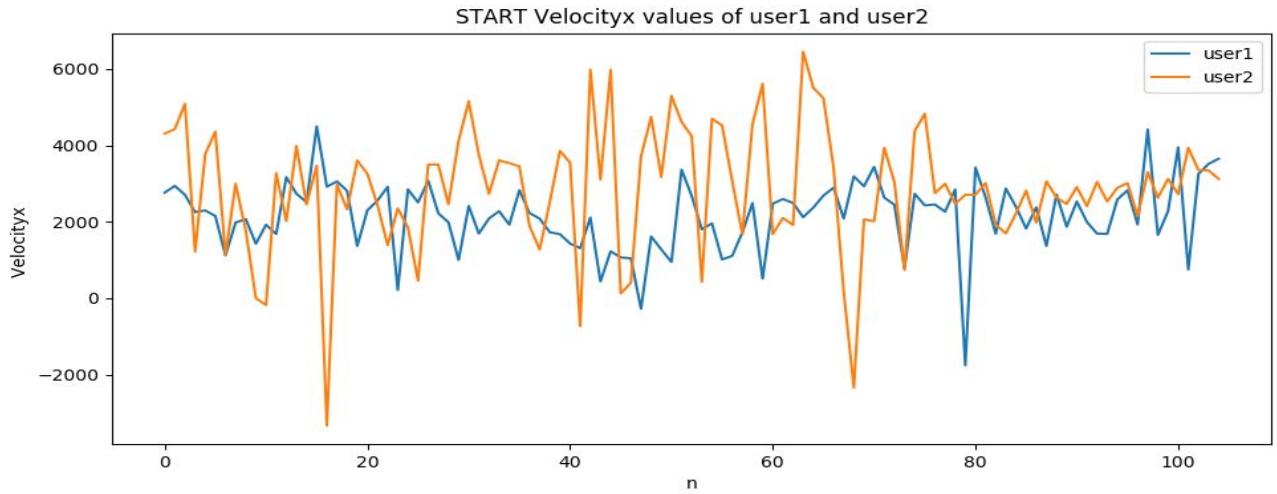
# Evaluation

After choosing some of the important features such as touch pressure, orientation and touch size, velocity values of touch that can be very useful in classification for the users as follow

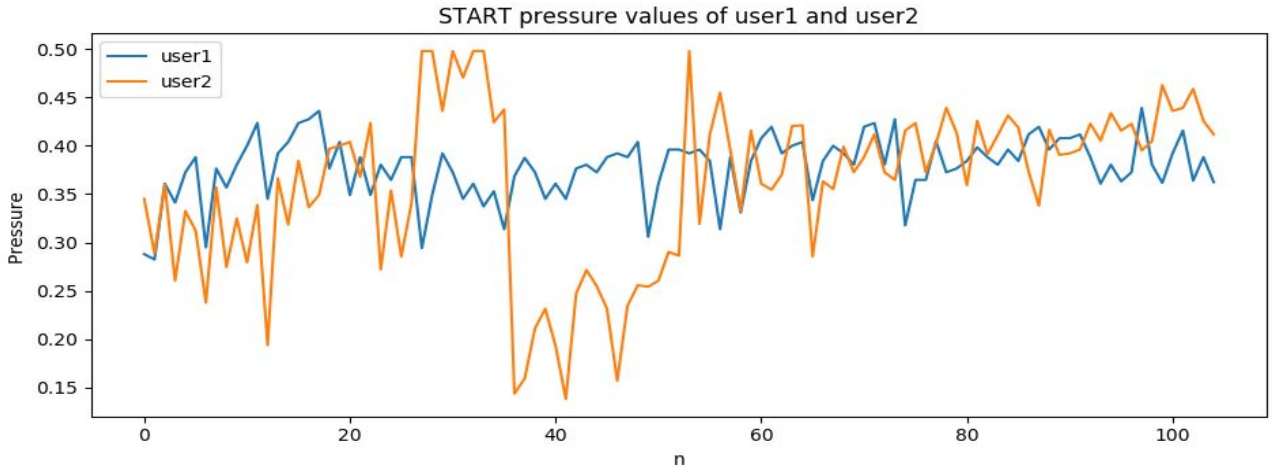
Min velocity of user1 and user2( set of all other users):




Starting Velocity of user1 and user2(set of other users)



Starting pressure values for user 1 and user 2




Here, we can see the differences in the values of the pressure and is important parameter in classification of correct user from training data available and
After all this feature collections , we started our choosing the best classification model and got the following results:

| **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 |
| SVM | 92.3000% | 7.30000% | 0.0360 |

After Choosing SVM( Support Vector Machine) which producing the least value of incorrect classification and further tries it on the bases of the number of splits and results are the following.

## Table 1 For 70% split

| | Split : 70% | | | | | |
|---|---|---|---|---|---|---|
| | TP Rate | FP Rate | Precision | Recall | F-Measure | class |
| | 0.981 | 0.1 | 0.946 | 0.981 | 0.964 | 1 |
| **User 1** | 0.9 | 0.019 | 0.0964 | 0.9 | 0.931 | 0 |
| | 1 | 0.33 | 0.947 | 1 | 0.97 | 1 |
| **User 2** | 0.667 | 0 | 1 | 0.667 | 0.8 | 0 |
| | 1 | 0.308 | 0.75 | 1 | 0.857 | 1 |
| **User 3** | 0.692 | 0.148 | 0.88 | 0.84 | 0.837 | 0 |
| | 0.75 | 0 | 1 | 0.75 | 0.857 | 1 |
| **User 4** | 1 | 0.25 | 0.833 | 1 | 0.909 | 0 |
| | 0.93 | 0.222 | 0.909 | 0.93 | 0.92 | 1 |
| **User 5** | 0.778 | 0.07 | 0.824 | 0.778 | 0.8 | 0 |

## Table 2 For 90% split

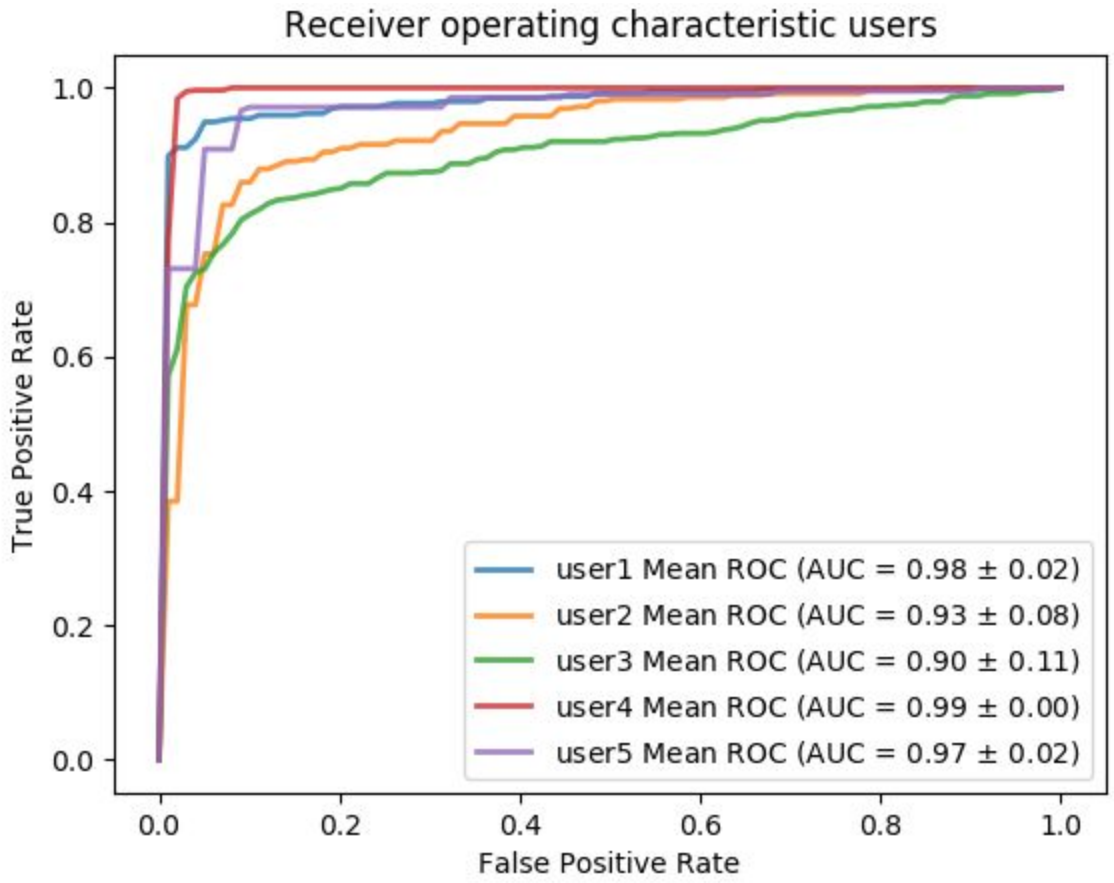
| | **Split :90%** | | | | | |
|---|---|---|---|---|---|---|
| | **TP Rate** | **FP Rate** | **Precision** | **Recall** | **F-Measure** | **class** |
| | **0.947** | **0.111** | **0.947** | **0.947** | **0.947** | **1** |
| **User 1** | **0.889** | **0.053** | **0.889** | **0.889** | **0.889** | **0** |
| | **1** | **0** | **1** | **1** | **1** | **1** |
| **User 2** | **1** | **0** | **1** | **1** | **1** | **0** |
| | **1** | **0.2** | **0.75** | **1** | **0.857** | **1** |
| **User 3** | **0.8** | **0** | **1** | **0.8** | **0.889** | **0** |
| | **0.75** | **0** | **1** | **0.75** | **0.857** | **1** |
| **User 4** | **1** | **0.25** | **0.833** | **1** | **0.909** | **0** |
| | **0.949** | **0.133** | **0.949** | **0.949** | **0.949** | **1** |
| **User 5** | **0.867** | **0.051** | **0.867** | **0.867** | **0.867** | **0** |

# Chapter 8
# ROC Plots for SVM Classifier :

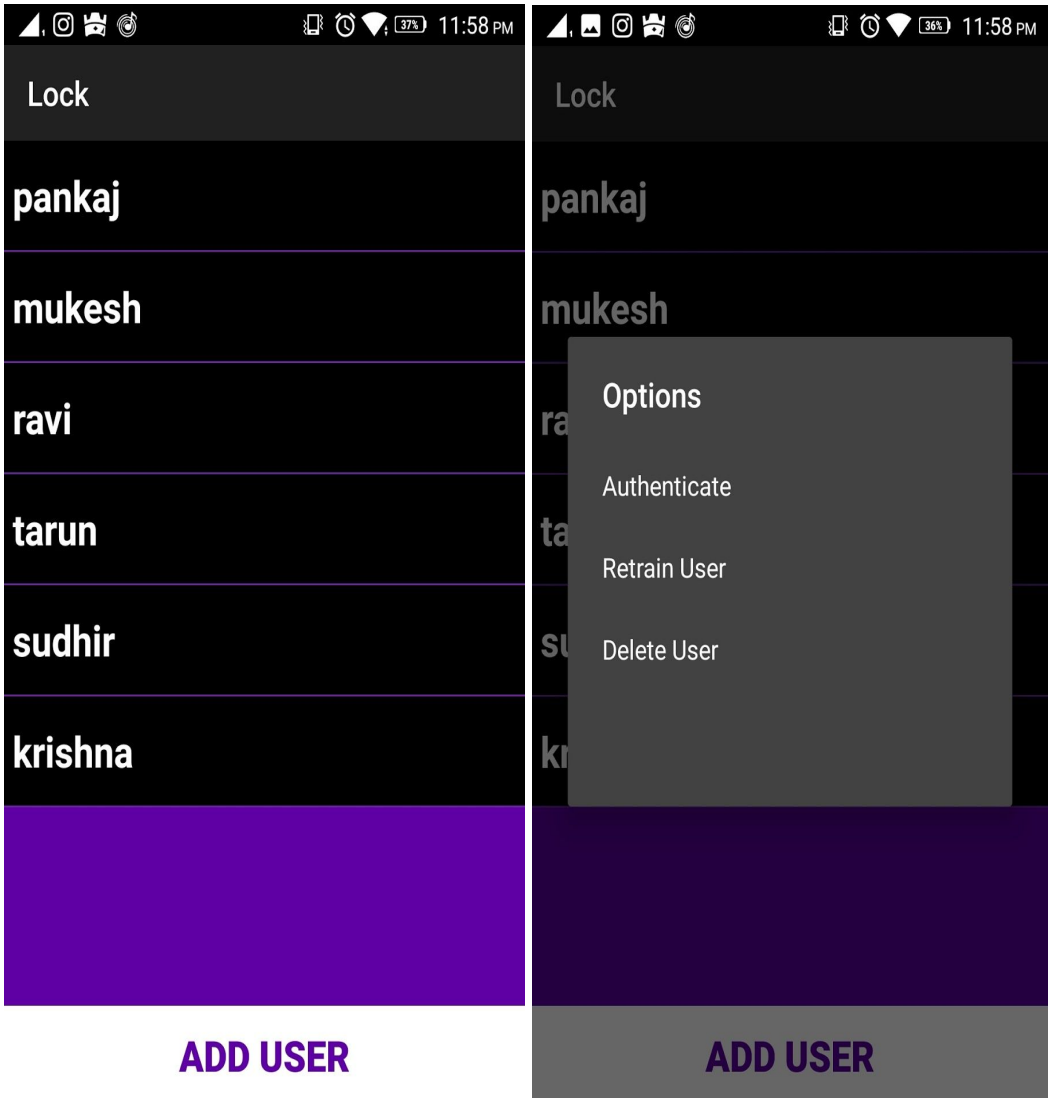
For Total data instance : 248

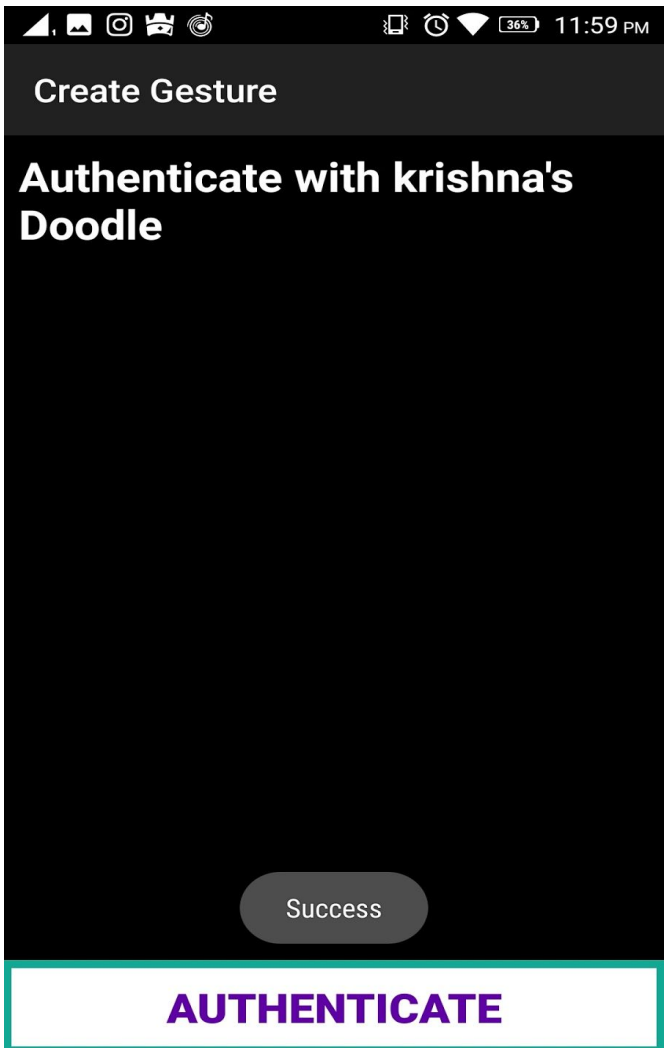
**X-axis : False Positive Rate**
**Y-axis : True Positive  Rate**

**Roc Plot for 5 User with 70 % split**



# Application ScreenShots :




11:59 PM
Create Gesture
Authenticate with krishna's Doodle
Success
AUTHENTICATE


9:18 PM
0.05K/s
4G
51%
Create Gesture
Retrain Pankaj's Doodle for (3)
SAVE
DISCARD
FINISH

# Chapter 9

# References

[1] Floren Alexis, Guillermo Gohan E. Guerrero III and Larry A. Vea. Modeling Free-form Handwriting Gesture User Authentication for Android Smartphones. IEEE/ACM International Conference on Mobile Software Engineering and Systems 2016.

[2] Frank, M. and Biedert, R. and Ma, E. and Martinovic, I. and Song, D.. 2013. Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. IEEE Transactions on Information Forensics and Security 8, 1 (2013), 136-148.

[3] Can Liu, Gradeigh D. Clark and Janne Lindqvist. Where Usability and Security Go Hand-in-Hand: Robust Gesture-Based Authentication for Mobile Systems. CHI 2017, May 6–11, 2017, Denver, CO, USA.

[4] Yang Li. 2010. Protractor: A Fast and Accurate Gesture Recognizer. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 2169–2172.

[5] Yuxin Meng, Duncan S. Wong, Roman Schlegel, and Lam-for Kwok. Touch Gestures Based Biometric Authentication Scheme for Touchscreen Mobile Phones

[6] Ala Abdulhakim Alariki and Azizah Bt Abdul Manaf. A study of touching behavior for authentication in touch screen smart devices. IEEE International Conference on Intelligent Systems Engineering 2016.

[7] Zhiping ZHOU and Minmin MIAO | Ziwen SUN. Mobile Terminal User Authentication Scheme Based on Dynamic Gesture. Proceedings of the 34th Chinese Control Conference July 28-30, 2015, Hangzhou, China.

[8] Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner and Heinrich Hussmann.Touch me once and I know it's you! Implicit Authentication based on Touch Screen Patterns. CHI 2012, May 5–10, 2012, Austin, Texas, USA.