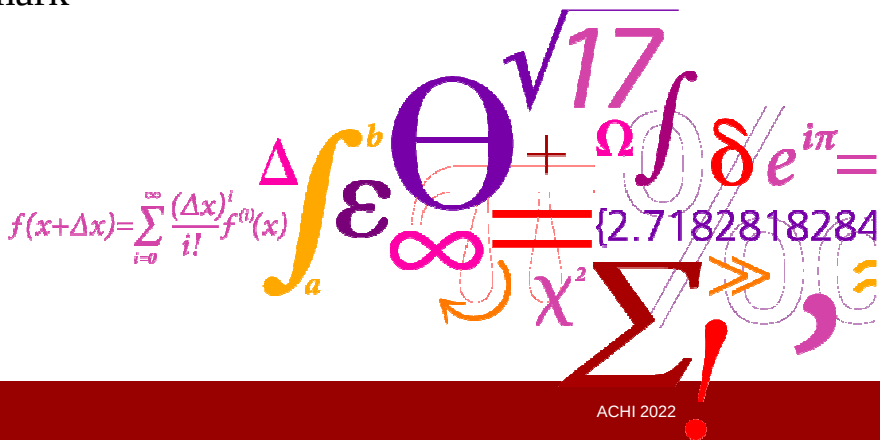
The Fifteenth International Conference on Advances in Computer-Human Interactions
(ACHI 2022), June 26, 2022 to June 30, 2022

Touch Behavioral Smartphone User Authentication based on Social Networking Applications

Weizhi Meng

Department of Applied Mathematics and Computer Science
Technical University of Denmark, Denmark



Research Directions

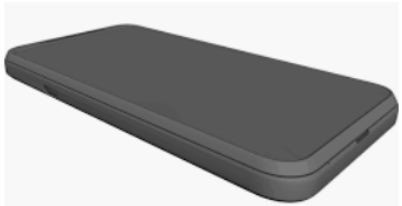
Weizhi Meng
weme@dtu.dk

- Intrusion Detection
- Biometric Authentication
- Trust Management
- HCI Security (Smartphone Security
- Blockchain

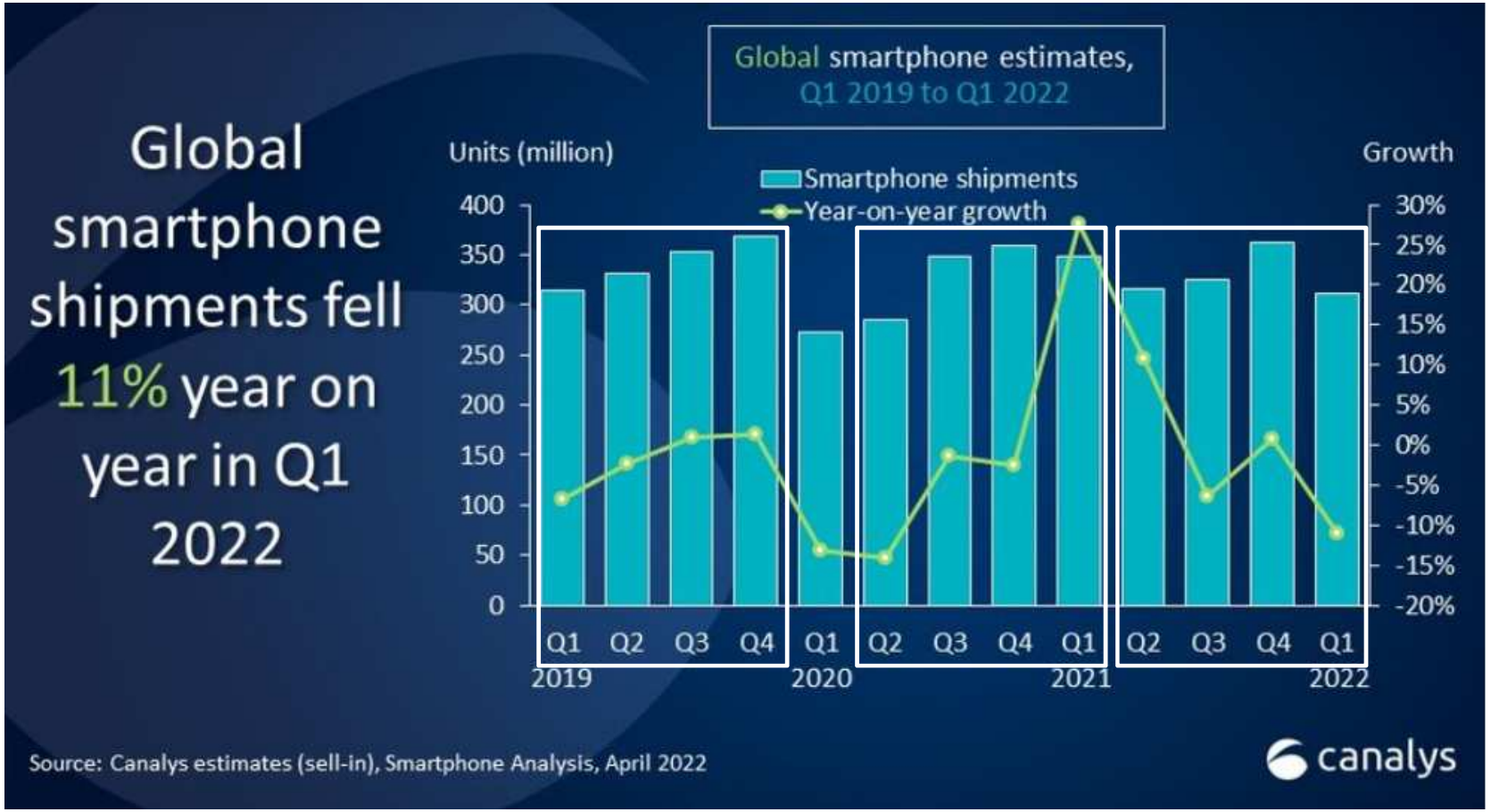
http://www.staff.dtu.dk/weme

Outline

- Background and Motivation
- SocialAuth
- Open Challenge / Discussion



Smartphone Shipments -- CANALYS



Sources: https://canalys.com/newsroom/worldwide-smartphone-market-Q1-2022

Popularity of Smartphones

- Due to the capabilities and convenience, smartphones have been widely adopted by individuals.

- These devices have become a personal assistant, i.e., working as a social connection and work facilitator. A survey showed that nearly 40 percent of respondents play with their phones for three hours or more each day*.

- As modern smartphones can work like a mini-computer, users are willing to store personal data and complete sensitive tasks on the phones, such as personal photos, credit card information, transactions, etc.
  - **62 percent** of phone users in Denmark were using their phone for viewing bank account and online payment (Source: Global Mobile Consumer Survey 2017)
  - During the 2018 holiday season in the US, users purchased almost **40%** of all e-commerce products via a smartphone (Source: OuterBoxDesign)
  - **Up to 85%** of travellers use mobile devices to book travel activities (Source: Adweek)

*https://www.abc.net.au/news/science/2017-10-13/smartphone-survey-results-show-fascinating-differences-in-usage/9042184

User Authentication

- Smartphones are becoming a more private device, cyber-criminals are always trying to exploit the stored data on smartphone.
- **User authentication mechanisms** become very important to protect phones from unauthorized access.

Biometric Authentication

Biometric authentication

Physiological

uses measurements from the human body

1. One-time verification
2. Additional hardware

Behavioral

use measurements from human actions

Continuous verification
No need for Additional hardware

1. False rate
2. Not Commercialized

State-of-the-Art

- There are many touch behavioral authentication schemes available in the literature, but how to design a behavioral authentication scheme for a long-term period still remains a challenge.

- Feng, T., Liu, Z., Kwon, K.-A., Shi, W., Carbunary, B., Jiang, Y., Nguyen, N.: Continuous mobile authentication using touchscreen gestures. In: Proc. of the 2012 IEEE Conference on Technologies for Homeland Security (HST), pp. 451- 456 (2012)
- Meng, Y., Wong, D.S., Schlegel, R, Kwok, L.-F.: Touch Gestures Based Biometric Authentication Scheme for Touchscreen Mobile Phones. In: Proceedings of the 8th China International Conference on Information Security and Cryptology (INSCRYPT), pp. 331-350, Springer, Heidelberg (2012)
- **Frank, M., Biedert, R., Ma, E., Martinovic, I.,Song, D.: Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. IEEE Transactions on Information Forensics and Security 8(1), pp. 136-148 (2013)**

Algorithm Improvement

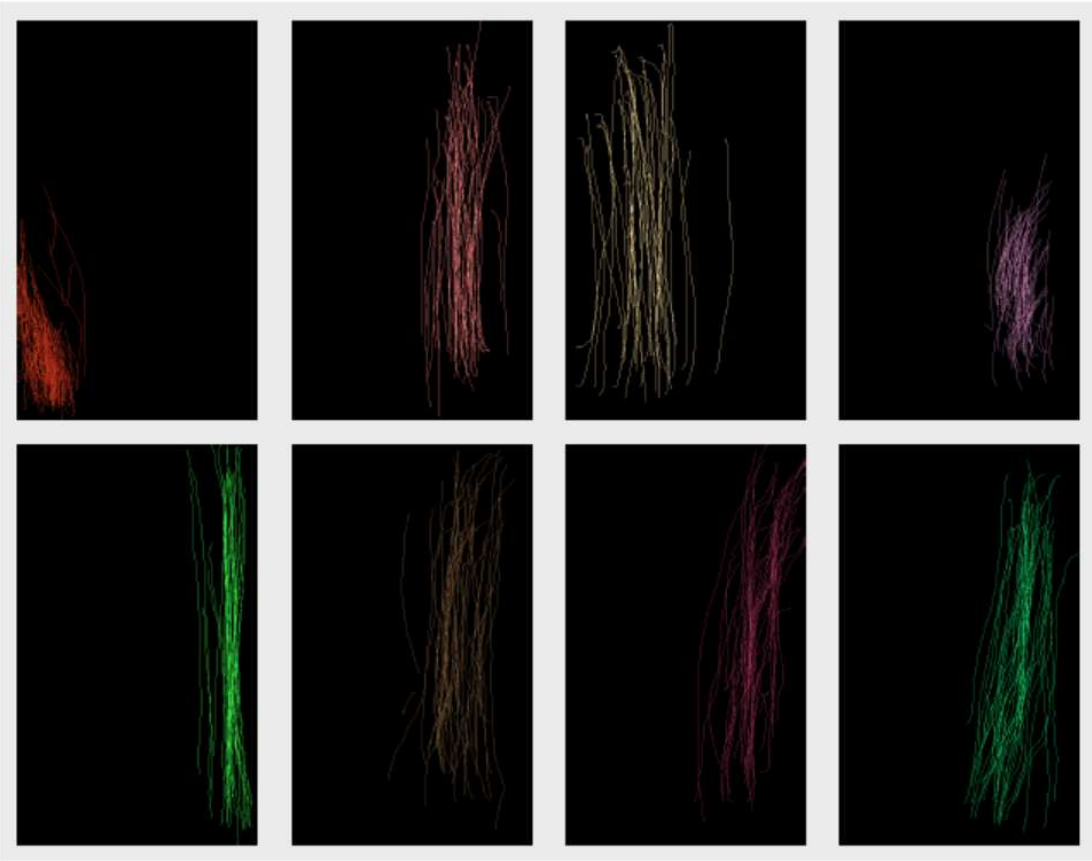
Refine Touch Actions (Swipe, Zoom)

Hybrid Scheme

2012/2013

Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, Dawn Song, Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication, IEEE Transactions on Information Forensics and Security (Vol. 8, No. 1), pages 136-148, IEEE 2013.

- This work investigated whether a classifier can continuously authenticate users based on the way they interact with the touchscreen of a smart phone.
- Feasibility and Stability.



- Each user's interaction behavior on touchscreens can be quite unique. This figure depicts strokes recorded from eight different users, each reading three different texts on an Android phone. Geometric patterns that discriminate the users from each other are already apparent. Other differences might come from different stroke timing, pressure, and area covered on screen.

Enrollment Phase

- The main hypothesis of this study is that continuously recorded touch data from a touchscreen is distinctive enough to serve as a behavioral biometric.

Define two particular user actions and call them `trigger-actions'.

- Sliding horizontally over the screen. Usually, one does this to browse through images or to navigate to the next page of icons in the main screen.
- Sliding vertically over the screen to move screen content up or down. This is typically done for reading email, documents or web-pages, or for browsing menus.

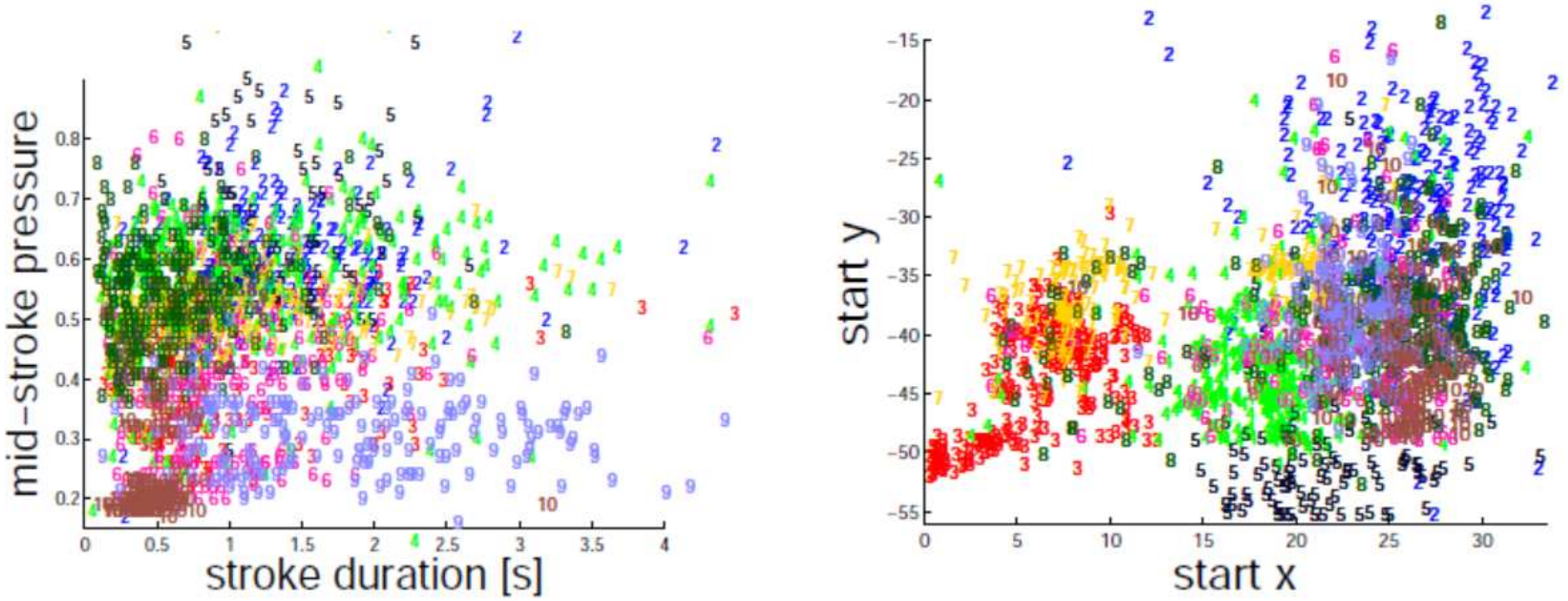
Continuous Authentication Phase

- Feature-extraction is to divide up the data records into **individual strokes**. A stroke is a sequence of touch data that begins with touching the screen and ends with lifting the finger.

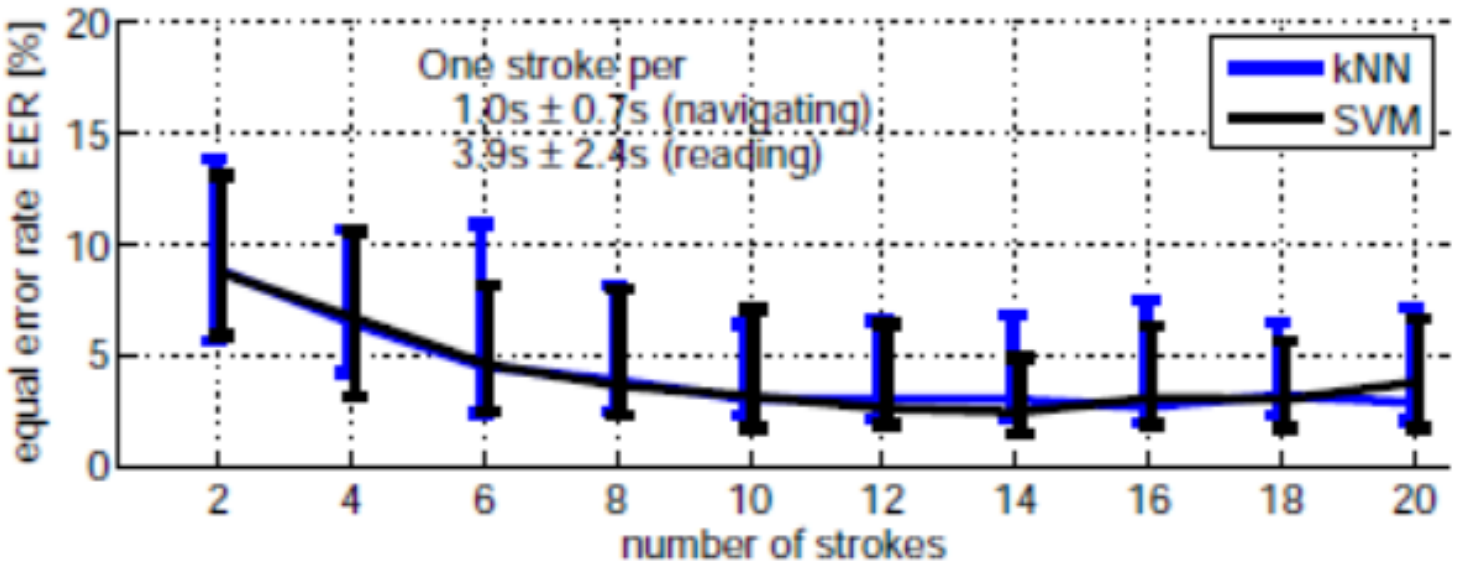
- Once the classifiers are trained, the device begins the authentication phase. During this phase, the system continuously tracks all strokes and the classifier estimates if they were made by the legitimate user

Rel. mutual information	Feature description
20.58%	mid-stroke area covered
19.63%	20%-perc. pairwise velocity
17.28%	mid-stroke pressure
11.06%	direction of end-to-end line
10.32%	stop x
10.15%	start x
9.45%	average direction
9.43%	start y
8.84%	average velocity
8.61%	stop y
8.5%	stroke duration
8.27%	direct end-to-end distance
8.16%	length of trajectory
7.85%	80%-perc. pairwise velocity
7.24%	median velocity at last 3 pts
7.22%	50%-perc. pairwise velocity
7.07%	20%-perc. pairwise acc
6.29%	ratio end-to-end dist and length of trajectory
6.08%	largest deviation from end-to-end line
5.96%	80%-perc. pairwise acc
5.82%	mean resultant lenght
5.42%	median acceleration at first 5 points
5.39%	50%-perc. dev. from end-to-end line
5.3%	inter-stroke time
5.14%	80%-perc. dev. from end-to-end line
5.04%	20%-perc. dev. from end-to-end line
5.04%	50%-perc. pairwise acc
3.44%	phone orientation
3.08%	mid-stroke finger orientation
0.97%	up/down/left/right flag
0%	change of finger orientation

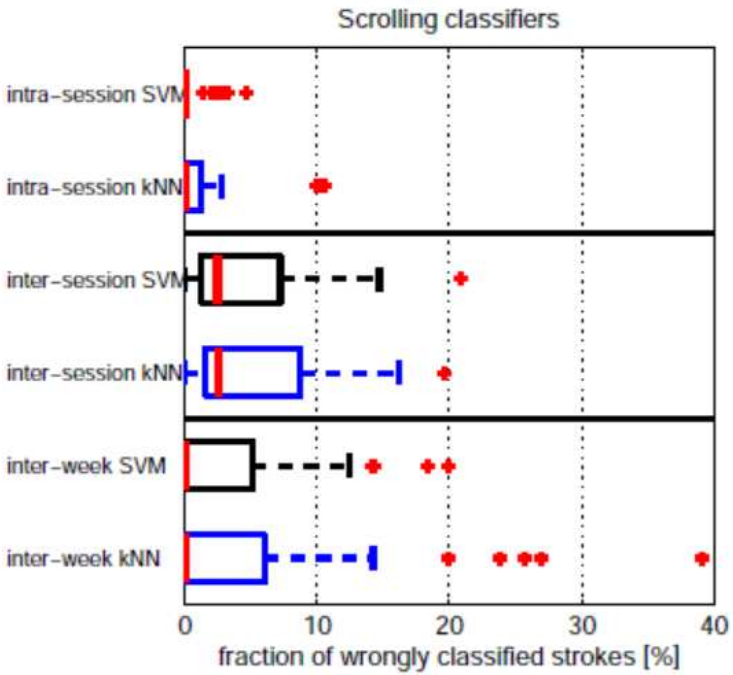
- A list of 30 features
- stroke velocity, fingertip pressure on screen and the direction of the stroke

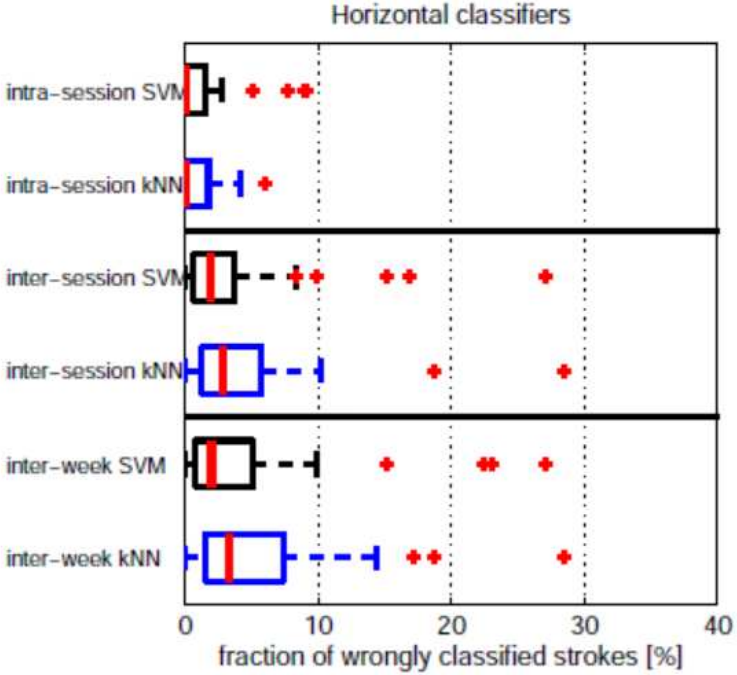


- Stroke features projected on a 2D-subspace. The user ID is given as a colored number. Already in these low-dimensional feature spaces, a class separation is apparent.
- The data depicted here was collected from users reading three Wikipedia articles in three different sessions. The left plot contrasts the finger pressure on the screen at the middle of the stroke against the stroke duration. The right plot shows the xy-positions where the fingertip first touches the screen.

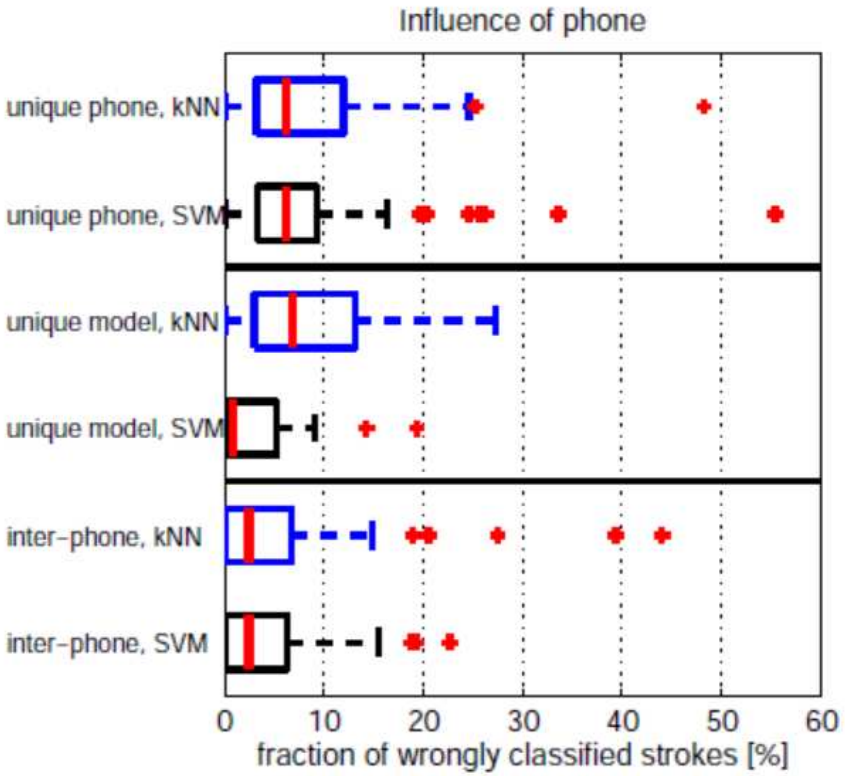


- Two classifiers: Support vector machines (SVM) and k-nearest-neighbors (kNN)
- When deciding with a single stroke only, the EER is approximately 13%. Both classifiers obtain a lower error when increasing the number of strokes used to provide a classification output.
- At a level of 11 to 12 strokes, the EER converges to a range between 2% and 3% and stays there up to using 20 strokes.





- The median EER ranges from 0% to 4% across all usage scenarios. The median intrasession errors are 0%, whereas few outliers can reach a 10% EER.
- It seems that, within one session, most users do not considerably change their touch behavior.
- The inter-session EER reaches from 2% to 3% and the inter-week EER reaches from 0% to 4%, depending on the scenario and the classifier used.



The error rates for users on the same phone are on average 2% higher than for user data collected on multiple phones.

”While our experimental findings disqualify this method as a standalone authentication mechanism for long-term authentication, it could be implemented as a means to extend screen-lock time or as a part of a multi-modal biometric authentication system.”

2012/2013

Y. Meng, D.S. Wong, R. Schlegel, and L.F. Kwok, Touch Gestures Based Biometric Authentication Scheme for Touchscreen Mobile Phones, In: Proc. of the 8th China International Conference on Information Security and Cryptology (INSCRYPT), pp. 331-350, LNCS, Springer, Heidelberg, 2012.

- In this work, we develop a user authentication system based on touch dynamics, including 21 touch gesture-based features.

Touch dynamics (3)

- In this paper, we classify inputs as captured by the touchscreen on a mobile phone into four categories:

  – Single-Touch (ST): the input starts with a touch press down, followed by a touch press up without any movement in-between.
  – Touch-Movement (TM): the input starts with a touch press down, movement (also called drag), followed by a touch press up.
  – Multi-Touch (MT): an input with two or more simultaneous, distinct touch press down events at different coordinates of the touch screen (i.e., two fingers press down on the touchscreen simultaneously), either with or without any movement before a touch press up event.
  – No input: there is no input on the touchscreen.

Architecture (1)

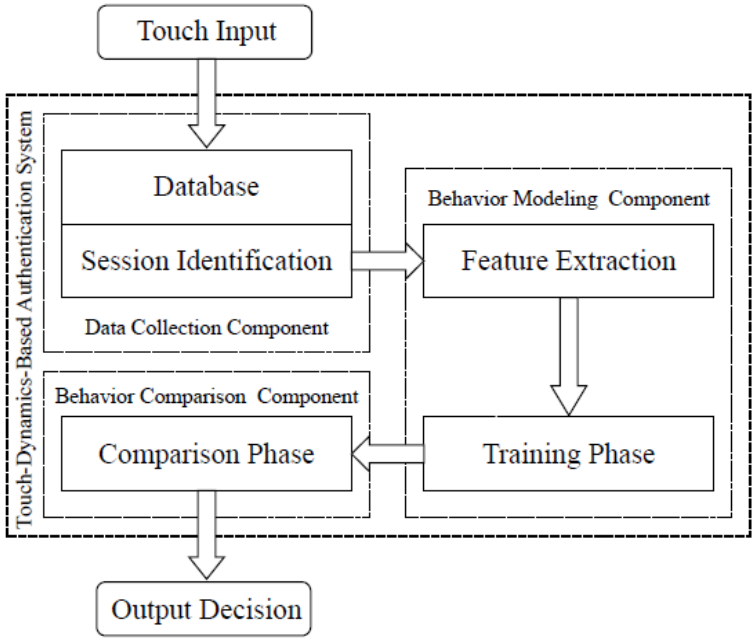


Figure shows the architecture of the touch-dynamics-based authentication system.

- **Data collection**: collects raw data from the touchscreen (i.e., recording and storing all touch gesture data into a database) and converting the raw data into meaningful information (i.e., identifying sessions).
- **Behavior modeling**: analyzes collected data, extracts features to generate authentication signature for a legitimate user, models a user's touch behavior.
- **Behavior comparison**: compares the current user's behavior with the relevant generated authentication signatures, and makes an output.

Architecture (2)

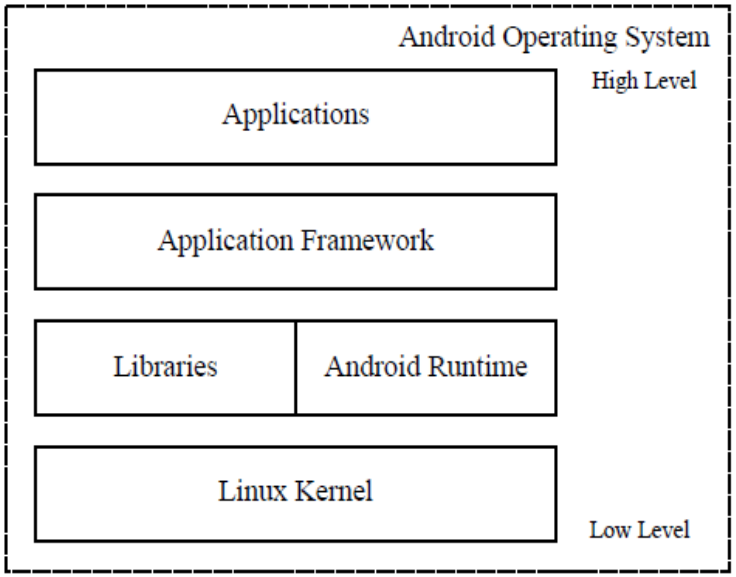


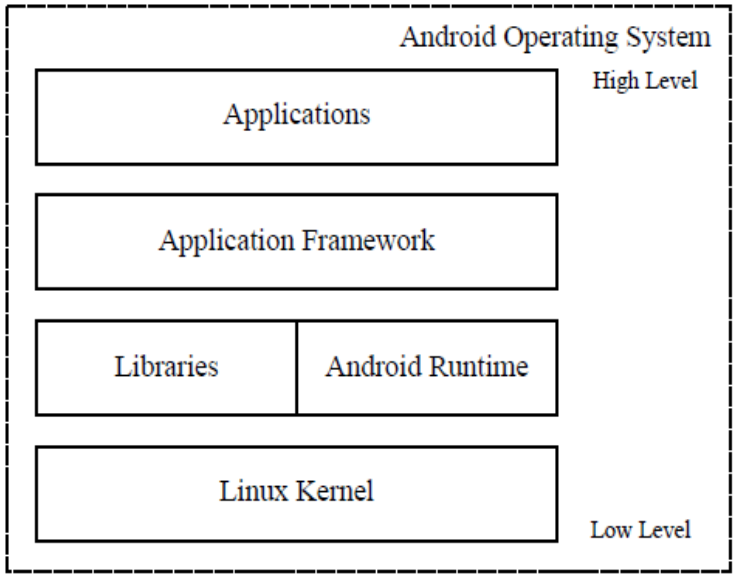
Figure . The architecture of the android operating system and its layers.

• **Linux kernel**. Android relies on Linux version 2.6 for core system services such as security, memory management and drivers. This layer contains drivers for devices such as USB, display, camera, Bluetooth chip and flash memory. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

• **Libraries**. Android includes a set of C/C++ libraries such as the System C library, media libraries and 3D libraries, which are all used by various components of the Android system.

•**Android runtime**. Android includes a runtime which contains a set of core libraries that provide different functionalities. In addition, every Android application runs in its own process, with its own virtual machine instance.

Architecture (3)



- **Application framework**. The Android application framework is a high-level layer to provide the developer with a development platform for creating new Android applications. Developers can access location information, run background services, add notifications to the status bar, and access lots of other information and functionality.
- **Applications**. This is the highest level of the Android operating system architecture. Android ships with a set of core applications and widgets including an email client, messaging application, calendar, maps, browser, contacts and others. Users can also easily add more applications.

In our case, modifying the application framework layer allows us to implement the desired functionality without the need to modify any applications, and it is more applicable to develop a system by programming the application framework as an interface is provided by Android.

Data Collection (1)

- We used a Google/HTC Nexus One Android phone (CPU: 1GHz, Memory: 512 MB) with a capacitive touchscreen (resolution 480X800 px) to perform the experiments.
- The advantage of this particular phone is that the stock Android operating system installed on it can be replaced with a modified custom version of the Android OS. In particular, we updated the phone with a modified Android OS version 2.2 based on CyanogenMod.
- The modification consists of changes to the application framework layer to record raw input data from the touchscreen, such as the timing of touch inputs, the coordinates x and y, and the type of the input (e.g., single-touch, multi-touch or movement).
- In addition, we installed a separate application, which allowed us to easily extract the recorded data from the phone.

Data Collection (2)

Table gives a sample of raw data collected from touchscreen inputs.

Input Type	X-Coordinate	Y-Coordinate	Time (ms)
Press Down	475.46866	659.6717	1770785
Press Move	472.56793	660.3004	1770807
Press Move	470.2978	660.9292	1770814
Press Move	466.76645	662.0609	1770852
Press Move	470.55002	659.9232	1770898
Press Move	472.56793	658.6658	1770910
Press Up	471.6851	658.9172	1770933

- Each record consists of at least the following four fields: input type, x-coordinate, y-coordinate, and system time (S-time).
- The system time in Table 1 is relative to the last start-up of the phone.
- The duration of each touch input can then be calculated by taking the difference in system-time.
- These four fields allow us to precisely determine the type of touch inputs, their coordinates and their duration.

Data Collection (3)

- Session identification: the purpose is to determine when a new session starts.

- The specific length of a session can be configured.
  - A new session starts when a touch input is recorded and the last session has ended.
  - A session ends if the duration of the current session has reached or exceeded the maximum session duration time. For instance, if we choose a session duration time of 10 minutes, then our scheme will terminate a session and start a new session when the duration time of the current session reaches or exceeds 10 minutes.

Feature Extraction (1)

- In this work, we extract 21 features to construct an authentication signature for user authentication.

- The features are the following:

1. Average touch movement speed per direction (8 directions)
2. Fraction of touch movements per direction (8 directions)
3. Average single-touch time
4. Average multi-touch time
5. The number of touch movements per session
6. The number of single-touch events per session
7. The number of multi-touch events per session.

Feature Extraction (2)

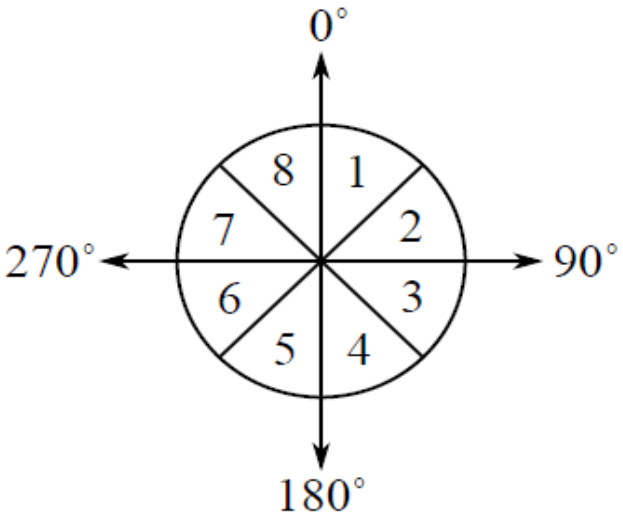


Figure shows the 8 different directions of a touch movement.

Average Touch Movement Speed per Direction:

•After categorizing the touch movements according to their direction, we then calculate the average touch movement speed (denoted ATMS) for each of the 8 directions, represented by *ATMSi* (e.g., ATMS1 represents the ATMS in direction 1, ATMS3 represents the ATMS in direction 3).

• Touch movement speed (**TMS**):

$$TMS = \frac{\sqrt{(x2-x1)^2+(y2-y1)^2}}{S2-S1}$$

•Touch movement angle:

$$\textit{Touch movement angle: } \theta = \arctan\frac{y2-y1}{x2-x1}, \theta \in [0, 360^{\circ}]$$

Feature Extraction (3)

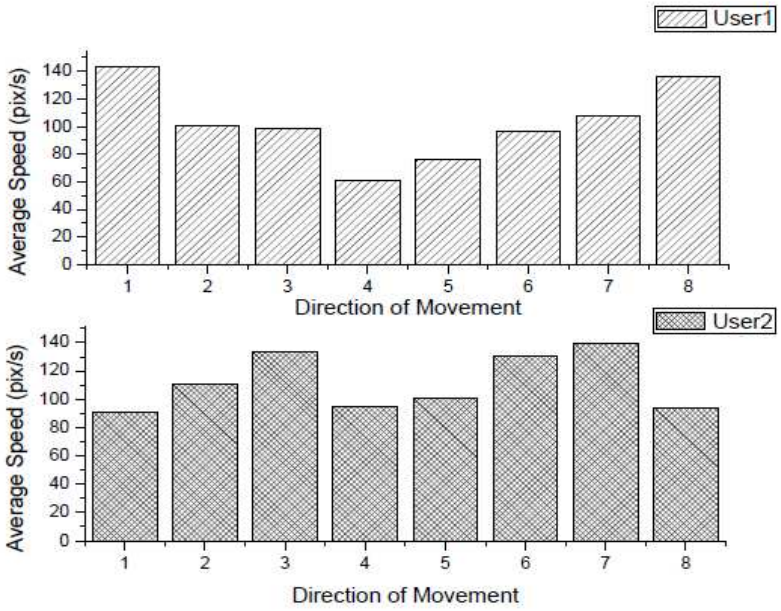


Figure shows the average touch movement speed versus the direction of movement for 2 different users.

- It is clearly visible that the distributions for these two users are different: the touch movements of User1 in direction 1 and 8 are performed with a higher speed than other directions, while the touch movements of User2 have a higher speed in direction 2, 3, 6, and 7. This illustrates nicely that the feature ATMS per direction (total of 8 features) can be used to model the characteristics of a user's touch behavior.

Feature Extraction (4)

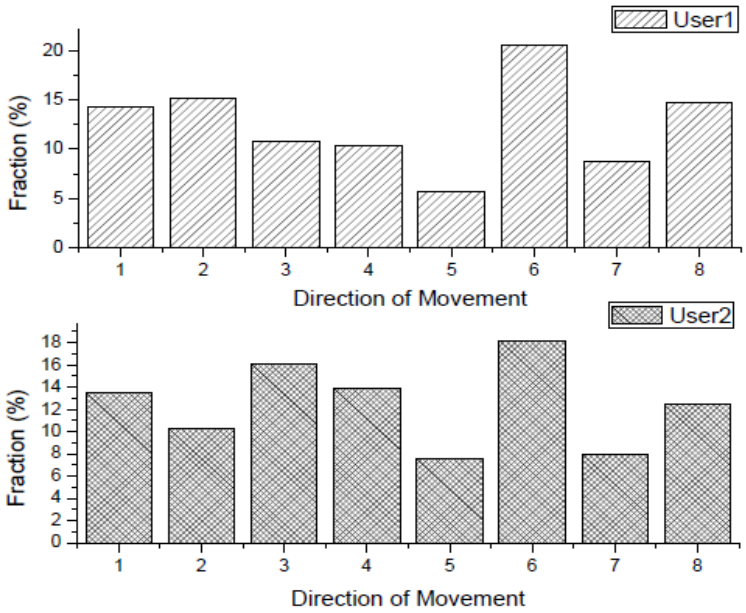


Figure shows the fraction of touch movements versus the direction of movement for 2 different users.

Fraction of Touch Movements per Direction (**FTM**)

- We observe that there are usually certain directions that contain more touch movements than other directions and that for different users the fraction per direction varies.
- It shows the distribution of the fractions of touch movements (denoted FTM) versus the direction of a touch movement for User1 and User2.
- User1 performed relatively more touch movements in direction 1, 2, 6 and 8, while User2 performed more touch movements in direction 1, 3, 4, 6, and 8.
- The FTM in 8 directions (total of 8 features) can be used to characterize the touch behavior of a user.

Feature Extraction (5)

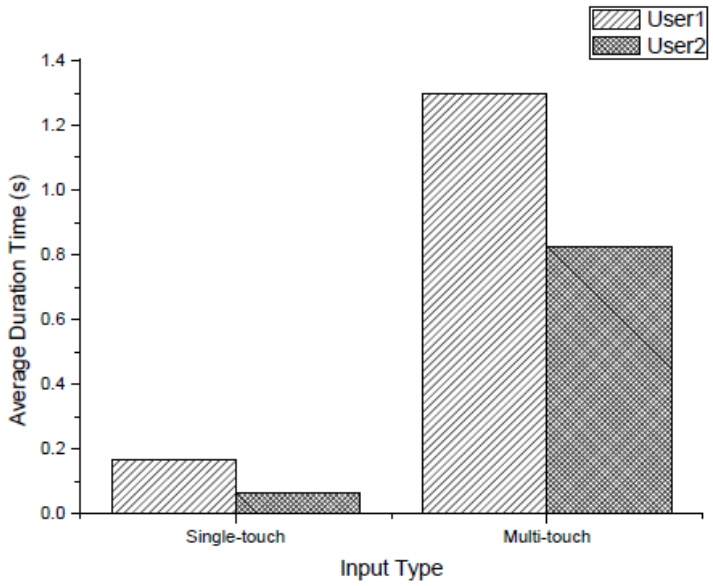


Figure shows the average single-touch time and the average multi-touch time for 2 different users.

Average Single-touch/Multi-touch Time (**AST/MTT**)

- In addition to touch movements, single-touch and multi-touch are also two important types of touch inputs. We observe that the average duration time of a single-touch or multi-touch is different for different users.
- It shows the histogram for these two features, Average Single-touch time (denoted AST) and Average Multi-touch time (denoted MTT) again for the two users User1 and User2.
- User1 on average spent a longer time for AST and MTT compared to User2, showing that these two features can also be used to characterize and hence distinguish the touch behavior of different users.

Feature Extraction (6)

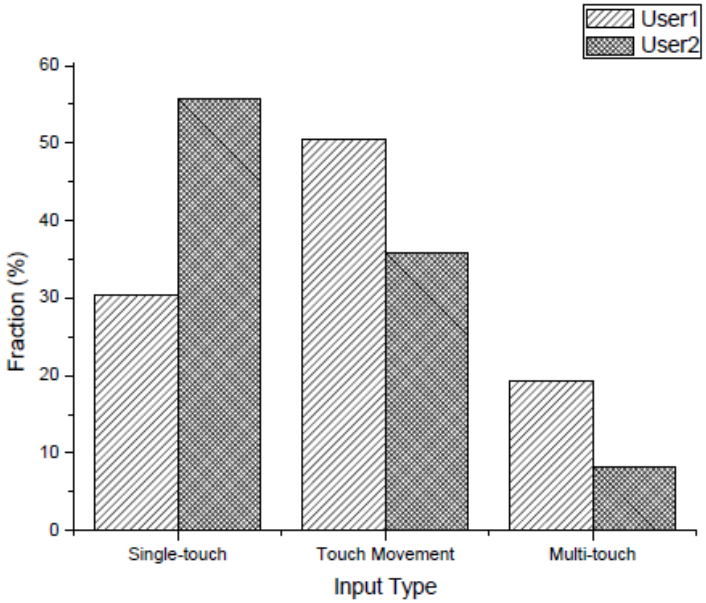


Figure shows the number of single-touch events, touch movements and multi-touch events per session for 2 different users.

Number of Touch Action Events (**AST/MTT**)

• Single-touch, touch movement and multi-touch events are three major input types on a touchscreen.

•We observe that the total number of these three touch events over one session varies for different users.

• We therefore distinguish the three features number of touch movements per session (denoted NTM), number of single-touch events per session (denoted NSTE), and number of multi-touch events per session (denoted NMTE).

We can find that User1 performed more touch movements and multi-touches than User2, while User2 performed more single-touches than User1. It is also clearly visible that the numbers differ significantly between the users, making this also a suitable feature to distinguish between users' touch behavior.

Evaluation (1)

- We investigate the performance of 5 existing classification schemes when applied to our system: Decision tree (J48), Naive Bayes, Kstar, Radial Basis Function Network (RBFN) and Back Propagation Neural Network (BPNN).

- **J48** is a decision tree classifier that classifies data items by generating decision trees from training data.
- **Naive Bayes** is a probabilistic classifier based on the assumption that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.
- **Kstar** is a statistical classifier based on the assumption that similar instances will have similar classes. Thus, it uses similarity functions to create instance-based classifications.
- **RBFN** and **BPNN** are neural network classifiers. RBFN is an artificial neural network that uses radial basis functions as activation functions. Its approximation capabilities are used to model complex mappings. The BPNN classifier has two main steps: (1) to present input and propagate it forward through the network to compute the output values for each output unit; (2) to perform backward passes through the network and calculate appropriate weights.

Evaluation (2)

- To remove any implementation related bias, we performed our evaluation using WEKA (using default settings), which is an open-source machine learning software that provides a collection of machine learning algorithms.

- *Methodology*. We had 20 Android phone users (12 female and 8 male) participate in our experiments and among the participants were students (85%) as well as professionals (15%).
- All participants were regular mobile phone users and ranged in age from 20 to 48 years.
- Before starting the collection, we described our objective to all participants and showed what kind of data would be collected. We asked participants to use the Android phones the same way they would use their own phones during the data collection period.
- Participants were asked to do the actual data collection outside of the lab, allowing them to get familiar with the phone first.
- Participants were asked to complete the collection of 6 sessions (with each session lasting 10 minutes) within 3 days, and they could use the phone freely as their own phones (e.g., using it to browse the web, install new software, etc.) during the entire collection period.

- **Evaluation measures**

  – False Acceptance Rate (FAR): indicates the probability that an impostor is classified as a legitimate user.

  – False Rejection Rate (FRR): indicates the probability that a legitimate user is classified as an impostor

• In practice, a trade-off is usually made between the false acceptance rate (security) and the false rejection rate (usability).

• In general, a false rejection is less costly than a false acceptance, since a higher false acceptance rate will lower the security level of the authentication system, while a higher false rejection rate will frustrate a legitimate user, which is still unfortunate but arguably less problematic than a lower security level.

• In terms of security and usability, both lower FAR and FRR are desirable

Evaluation Results

Table 2. Evaluation results for the tested classifiers.

Measure	J48	NBayes	Kstar	RBFN	BPNN
FAR (%)	22.43	22.45	14.11	7.08	8.85
FRR (%)	25.01	18.36	16.69	8.34	14.3
Avg. err. rate	23.72	20.41	15.4	7.71	11.58
SD in FAR	16.46	18.1	12.3	6.4	7.72
SD in FRR	21.33	7.63	13.73	6.83	10.6

1. The evaluation results show that for the data collected from our participants, the two neural network classifiers (RBFN and BPNN) have the best performance with an average error rate of 7.71% and 11.58%, respectively, compared to the other classifiers, which have average error rates of between 15% and 24%.

2. Although these experimental results are encouraging for the feasibility of our scheme, an average error rate of about 7.8% is still very high for real world systems. The reason for an error rate of around 7.8% is that the performance of the classifiers decreases as the variance of the feature datasets increases. Table 2 shows the standard deviation of the FAR and FRR for each classifiers, ranging from 7% to 22%.

A more ideal classifier suitable for our system should therefore meet the following requirements:

(1) The classifier should provide a relatively small FAR and FRR (less than 5% each).

(2) The classifier should be economical in terms of computational power required, considering that it will be run on mobile devices with limited resources

(3) The classifier should be able to deal with the sometimes significant variations in the feature dataset

PSO-RBFN Classifier (1)

- To improve the performance of the classification when working on data with significant variations in a user's behavior, we applied an algorithm that combines Particle Swarm Optimization (PSO) and an RBFN classifier.
- In this work, the RBFN classifier was selected for two reasons:

  (1) RBFN has the lowest FAR and FRR compared to the other classifiers, as shown in Table 2;

  (2) Comparing the two neural network classifiers (RBFN and BPNN), RBFN has better accuracy and is faster when authenticating a user (e.g., fast in constructing models), which is a desirable property for applications that are run on resource-limited devices such as mobile phones.

PSO-RBFN Classifier (2)

- PSO was selected for the following two reasons:

  (1) PSO is one of the most commonly used evolutionary algorithms used to optimize the structure of neural networks (e.g., RBFN);

  (2) PSO can achieve faster convergence speed and requires fewer optimized parameters compared to other evolutionary algorithms such as Genetic algorithms, which benefits the implementation on a mobile phone. The principle of the PSO-RBFN classifier is described below.

- In hybrid PSO-RBFN, PSO can be used to enhance the RBFN training by optimizing the radial activation function and weighted sum of RBFN with a population-based iterative search procedure, so that PSO-RBFN can better deal with variations in a user's touch behavior compared to regular RBFN

PSO-RBFN Classifier (3)

Table 3. the experimental results of comparing the PSO-RBFN classifier against the regular RBFN classifier.

Measure	RBFN	PSO-RBFN
FAR (%)	7.08	2.5
FRR (%)	8.34	3.34
Average error rate	7.71	2.92
SD in FAR	6.4	1.22
SD in FRR	6.83	1.89

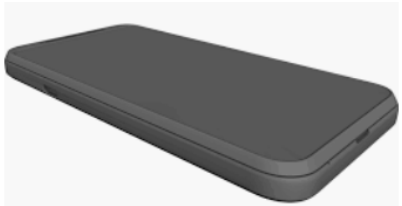
The numbers clearly show that using a combination of PSO and RBFN significantly improves the accuracy, reducing the average error rate from 7.71% for RBFN to 2.92% for PSORBFN.

An FAR of 2.5% and FRR of 3.34% mean that the possibility of identifying an impostor as a legitimate user and the possibility of identifying a legitimate user as an impostor are low.

Furthermore, both the FAR and the FRR are below 5% when using the PSO-RBFN classifier and the standard deviation is also significantly lower compared to RBFN.

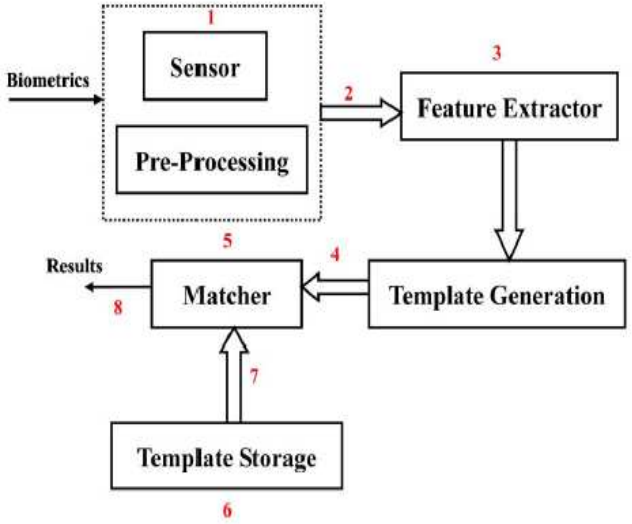
Outline

- Background and Motivation
- SocialAuth
- Open Challenge / Discussion



How to design a robust touch behavioral authentication?

- Pay attention to the above vulnerable points, but system improvement is only one aspect!



- Template/data security & privacy
- Side channel attack
- Hardware security

- It is more important to guide phone users.

How to design a robust touch behavioral authentication?

Given Conditions

- It is hard to build a model by means of free touches
- Find out the most stable touch gestures under a given condition

User Guidelines

- Users have to keep their touch habits

SocialAuth: Touch gesture-based authentication

- In this work, we focus on social networking applications and design a touch behavioral authentication scheme called ***SocialAuth***.
- An ideal touch behavioral authentication scheme has to continuously monitor the behaviors and make an alert (or lock the phone) when any anomalies are detected. The high-level architecture of touch gesture-based auth**en**tication system is presented in Fig. 1.

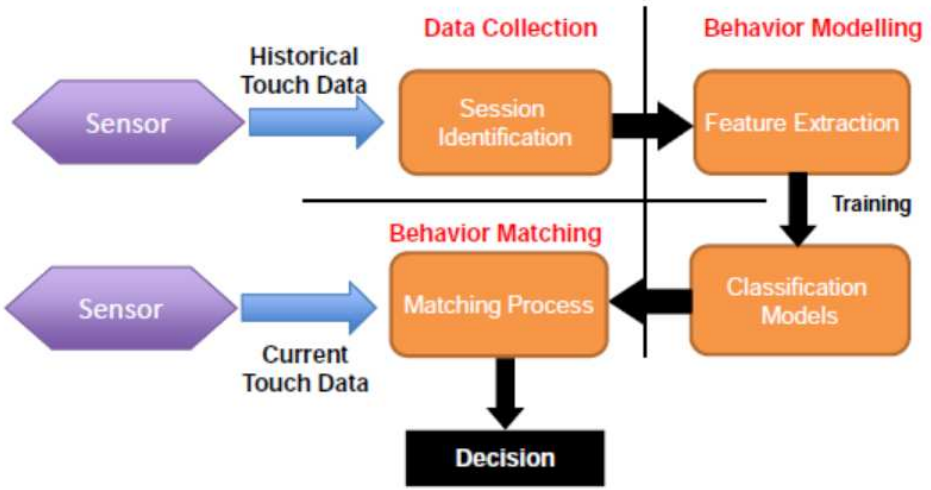


Figure 1. The architecture of touch gesture-based authentication system.

Three major phases:

Data collection

Behavior modelling

Vehavior matching

Weizhi Meng, Wenjuan Li, Lijun Jiang, and Jianying Zhou. SocialAuth: Designing Touch Behavioral Smartphone User Authentication based on Social Networking Applications. The 34th IFIP International Conference on Information Security and Privacy Protection (IFIP SEC 2019), pp. 180-193, June 2019.

Touch Gesture Types

- Modern smartphones can provide a wide range of touch gestures, such as tap, swipe left or right, swipe up and down, and so on. Generally, these gestures on touchscreen can be categorized into the following types.

- **Single-Touch (ST):** this touch event starts with a touch-press down, and ends with a touch-press up without any touch movement in-between, like single-finger tap
- **Touch-Movement (TM):** this touch event starts with a touch-press down, followed by a touch movement, and ends by a touch-press up, like swipe up and down
- **Multi-Touch (MT):** this touch input starts with two or more simultaneous and distinct touch-press down events at different coordinates of a touchscreen, either with or without any touch movement before a touch press up event, like zoom, pinch and rotate.

Touch Features - 1

- In this work, we adopt and revise a touch behavioral authentication scheme on smartphones with up to 22 features, based on the work [9]. While we add one extra touch feature, namely **touch pressure** into the scheme, as many studies have proven its effectiveness.

- **Average Touch Movement Speed per Direction**. Suppose a touch movement can be divided into different features. If we assume there are two points (x1, y1) and (x2, y2) in a touch movement' trajectory with relevant system time S1 and S2 (suppose S1 < S2).

- Then we can have **touch movement speed (TMS)** and **touch movement angle.** Let ATMS denote average touch movement speed. It is easy to calculate each feature based **on the angles.**

$$TMS = \frac{\sqrt{(x2-x1)^2+(y2-y1)^2}}{S2-S1}$$

$$Touch\ movement\ angle:\ \theta = \arctan\frac{y2-y1}{x2-x1}, \theta \in [0, 360^{\circ}]$$

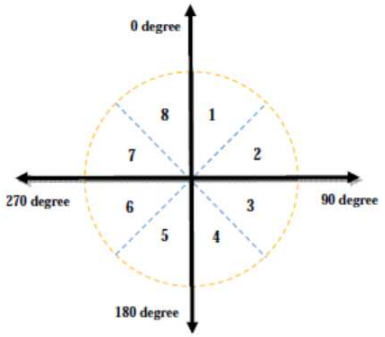


Fig. 2. Different directions for a touch action.

Touch Features - 2

- **Fraction of Touch Movements per Direction**. Intuitively, users may perform a touch movement more often in some certain directions. Therefore, the fraction of touch movements per direction varies among users and can be used for user authentication.

- **Average Single-Touch and Multi-Touch Time**. Single-touch and multi- touch are two types of touch gestures when users interact with their phones. Let AST denote average single-touch time and MTT denote average multi-touch time. The touch duration would be different between a single-touch and a multi-touch action.

- **Fraction of Touch Action Events**. It is observed that users could have their own habit when interacting with the phone. Three relevant features can be derived: the fraction of touch movements per session (denoted FTM), the fraction of single-touch events per session (denoted FSTE), and the fraction of multi-touch events per session (denoted FMTE).

- **Touch Pressure**. With the development of modern smartphones, sensors are becoming more accurate and sensitive.

Data Collection - 1

- We employ an Android phone - Google/HTC Nexus One for data collection, which has a capacitive touchscreen of 480x800 px. This type of phone is selected because its OS can be replaced with a modified OS version. In this work, we updated the phone with a modified Android OS version 2.2 based on CyanogenMod*.

- The changes were mostly on its application framework layer by inserting system level command to record raw data from the touchscreen, such as the timing of touch inputs, the coordinates x and y, and the touch pressure and various gestures like single-touch, multi-touch and touch movement. (inserted Slog.v command to two java source files (InputDevice.java and KeyInputQueue.java) regarding the Application framework layer, and then recompiled the whole source codes)

- A separate logcat application was installed to help extract and record the captured data from the phone.

* http://www.cyanogenmod.com/

Data Collection - 2

- A sample of collected raw data from the phone is depicted in Table 1. Each record contains five major items: **input type, x-coordinate, y-coordinate, touch pressure, and system time** (S-time).

Table 1. A sample of raw data collected from touchscreen on the Android platform.

Input Type	X-Coordinate	Y-Coordinate	Touch Pressure	Time (ms)
Press Down	478.5686	658.6726	0.090196080	1870785
Press Move	473.5593	660.5503	0.101960786	1870807
Press Move	471.2780	660.9001	0.101960786	1870814
Press Move	468.7645	662.0188	0.125686300	1870852
Press Move	470.5872	660.5211	0.125686300	1870898
Press Move	472.8723	658.5432	0.125686300	1870910
Press Up	470.6778	660.6223	0.125686300	1870933

The system time is relevant to the start-up of the phone and is managed by the phone itself, while the duration of each touch gesture can be computed by measuring the difference in system-time between touch press down and up.

Session Identification

- To build a behavioral profile, session identification is an important factor that could affect authentication performance.

- The **purpose** of session identification is to help decide the length of a session. To ensure the collection of enough touch gestures, in this work, we adopted an event-based session identification includes a total of **120 touch** gestures in each session.

- A session ends if the number of touch gestures reached the pre-define value and then a new session starts. For implementation, session start and end can be easily determined by checking the raw data record.

User Study – Methodology - 1

- In the study, we recruited a total of 50 regular Android phone users (including 26 female and 24 male), who were aged from 18 to 61 years. Participants have a diverse background including students, senior citizens, researchers and business people.
- During the study, each participant was provided with an Android phone (a Google/HTC Nexus One) equipped with our modified OS version. The main purpose is to ensure that all data were collected under the same settings.

- Table 2 details the background information of participants.

Table 2. Background of participants in the user study.

Occupation	Male	Female	**Age**	Male	Female
Students	14	16	18 - 30	14	16
Business people	2	3	31 - 40	5	5
Researchers	7	5	40 - 50	2	3
Senior citizen	1	2	Above	3	2

User Study – Methodology - 2

- Before the study, we described our research objective to all participants, introduced how to perform data collection, and explained what kind of data would be collected, i.e., we emphasized that no personal data would be collected during the study.
- Further, we seek **approval** from each participant for gathering and analyzing the data, before they started the experiment.

- All participants were required to use the Android phones freely as the same way they would use the phones in their daily lives. By considering the limitations of a lab study, we allowed participants to do the actual data collection out of the lab, motivating them to have enough time to get familiar with the phone.

- In this study, we mainly consider two situations for data analysis. **For the first situation (S1),** our scheme analyzes all recorded touch gestures when participants use the phones, whereas **for the second situation (S2),** our scheme only considers the touch gestures when participants play with any social networking applications.
- Each participant was required to complete 15 sessions for each situation (each session contains 120 touch gesture events) within 3 days. As a result, we could collect up to 1500 sessions of raw data, that is, 750 sessions for each situation. All participants could get a $20 gift card.

Machine Learning Classifiers and Metrics

- As a study, we employed five commonly used classifiers in the comparison: namely, Decision tree (J48), Naive Bayes, Radial Basis Function Network (RBFN), Back Propagation Neural Network (BPNN) and Support Vector Machine (SVM).
- To avoid any unexpected implementation bias, we extracted the above classifiers from **WEKA** (using default settings), which is an open-source collection of machine learning algorithms.

- There is a need to balance false acceptance rate and false rejection rate in real-world applications.
  - **False Acceptance Rate (FAR):** indicates the probability that an impostor is categorized as a legitimate user.
  - **False Rejection Rate (FRR):** indicates the probability that a legitimate user is classied as an intruder.

Result Analysis - The effectiveness of features -1

- Our authentication scheme is comprised of 22 touch features such as ATMS1, ATMS2, ATMS3, ATMS4, ATMS5, ATMS6, ATMS7, ATMS8, FTM1, FTM2, FTM3, FTM4, FTM5, FTM6, FTM7, FTM8, AST, MTT, FTM, FSTE, FMTE and ATP.

- In this part, we analyze the collected data regarding the **effectiveness of features**, **touch behavioral deviation between two groups**, **authentication accuracy**, and **long-term performance after two weeks**.

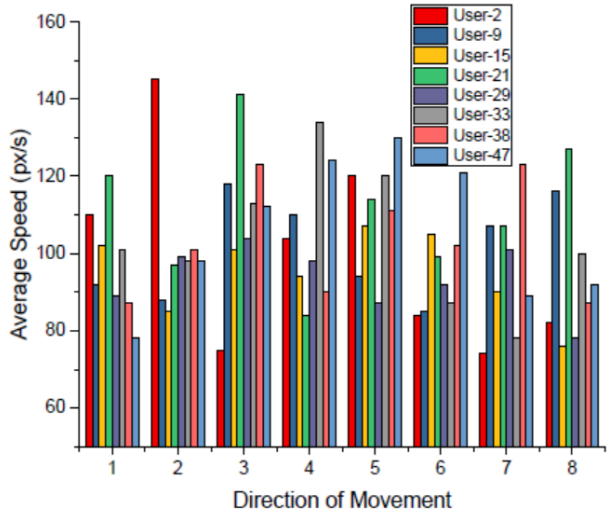


Fig. 3. The average touch movement speed per direction for eight different users.

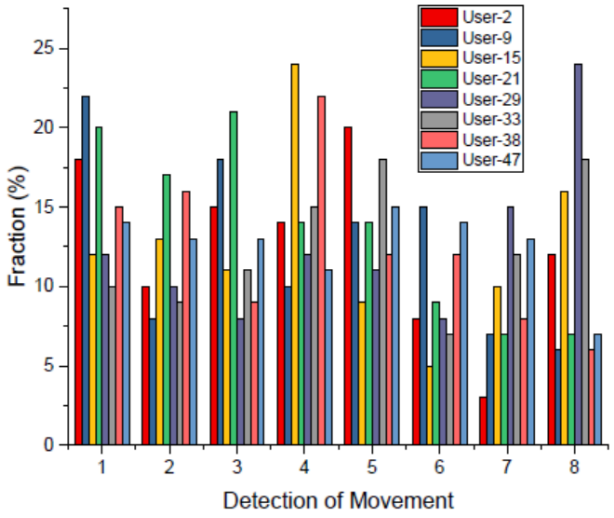


Fig. 4. The fraction of touch movements per direction for eight different users.

Result Analysis - The effectiveness of features -2

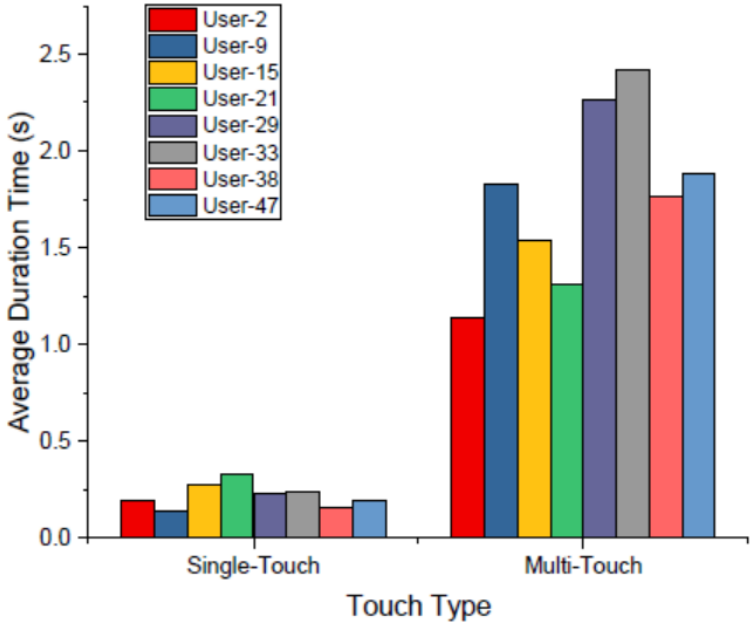


Fig. 5. The average duration time regarding single-touch and multi-touch for eight different users.

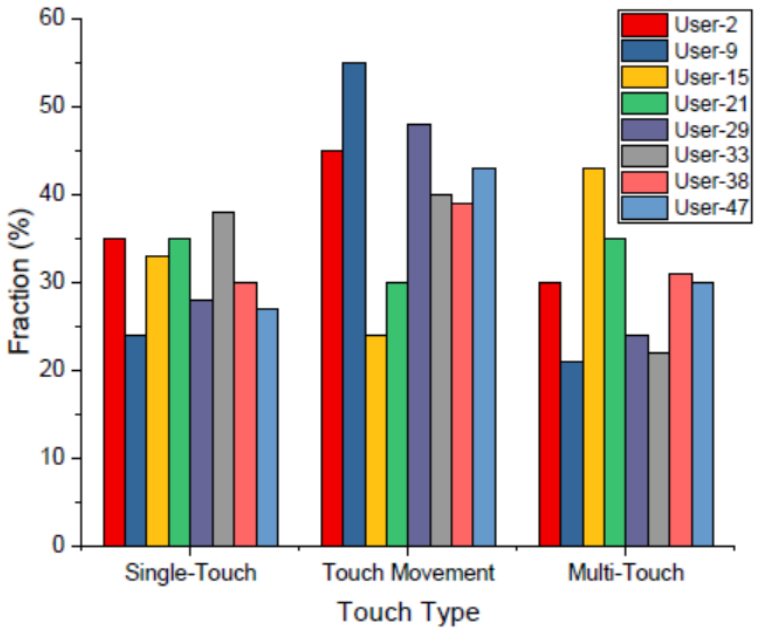


Fig. 6. The fraction of single-touch, touch movement and multi-touch for eight different users.

Result Analysis - Touch Behavioral Deviation - 1

- Under S1, we considered all touch behavioral events when participants used the phone, while under S2, we only considered the touch gestures when they were using social networking applications. Our major purpose is to investigate the touch behavioral deviation between the two situations.

- A total of four social networking applications were selected in the study: **WeChat, Facebook, Twitter and Instagram**.

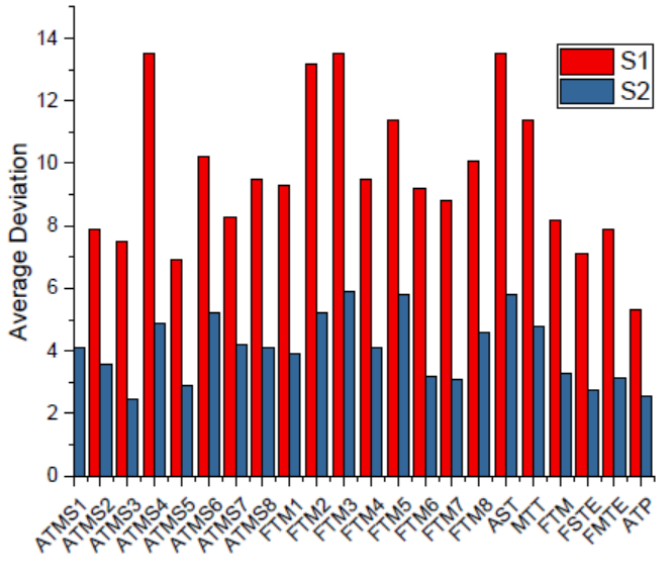


Fig. 7. The average behavioral deviation regarding all features under two situations.

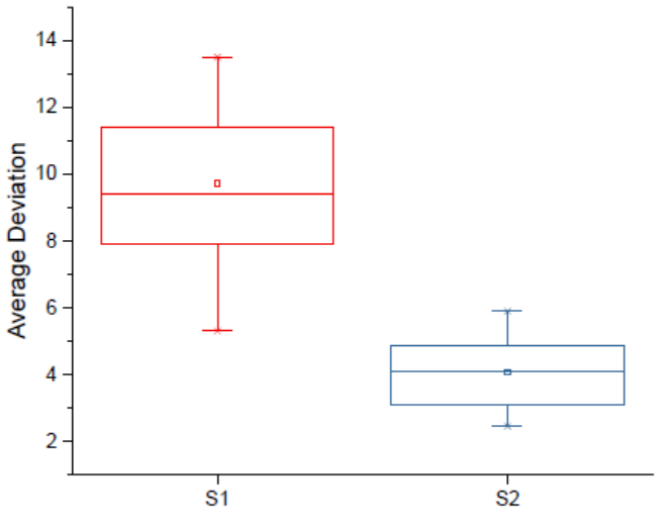


Fig. 8. The distribution of average deviation under two situations.

Result Analysis - Touch Behavioral Deviation - 2

- Intuitively, a higher deviation means that participants' touch gestures are more unstable, which may increase the difficulty of behavioral modelling.

- In contrast, **a smaller deviation makes it easier to build a robust touch behavioral authentication scheme**.

- We informally interviewed all the participants about their habits of phone usage. Based on their feedback, **most participants reflected that their touch behavior would be quite dynamic when they freely used the phone without a task**, whereas **their touch actions would become focused when they were using a particular application**, like social networking application.

- The feedback **validated** the observation that users' touch actions could become relatively stable under certain scenarios.

Result Analysis – Authentication Accuracy

- To investigate the authentication performance, we applied 18 sessions (up to 60% of the total sessions) as training data to help each classifier build a touch behavioral profile for each participant.

- Then we used the remaining sessions for testing. The test was run in 10-fold mode provided by the WEKA platform. The **false acceptance rate (FAR), false rejection rate (FRR), and average error rate (AER)** are presented in Table 3.

Table 3. Authentication performance for different classifiers under two situations.

S1	J48	NBayes	RBFN	BPNN	SVM
FAR (%)	22.55	18.66	9.72	9.12	5.22
FRR (%)	23.78	20.73	10.45	10.34	6.82
AER (%)	23.17	19.70	10.09	9.73	6.02
S2	J48	NBayes	RBFN	BPNN	SVM
FAR (%)	15.13	11.56	6.88	6.42	2.89
FRR (%)	16.55	13.23	7.11	7.88	3.24
AER (%)	15.84	12.40	7.00	7.15	3.07

*Users' touch behavior can become relatively stable under our scheme of SocialAuth, when they play with certain phone applications like a social networking application, as compared to the situation by considering all touches during the phone usage.

Result Analysis – Long-term Authentication - 1

- In the study, up to **16 participants** (seven males) chosen to attend our task on long-term authentication, in which they could keep using our provided phone and returned to our lab after two weeks. Our goal is to investigate the behavioral deviation after two weeks.
- They then required to complete **5 sessions** for **each S1 and S2** within two days. After the experiment, they could get a $30 gift card.

- Fig. 9 and Fig. 10 shows the average behavioral deviation regarding all features and the distribution of behavioral deviation after two weeks, respectively.

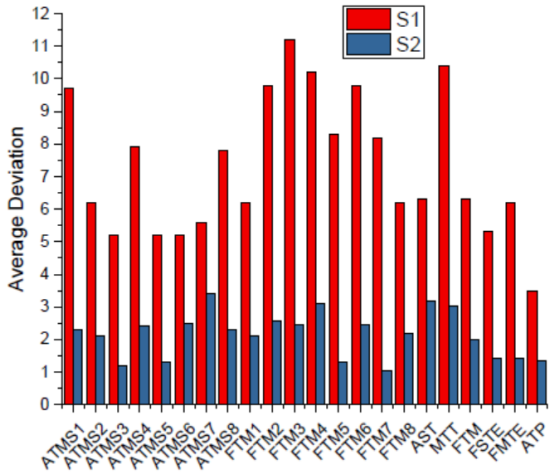


Fig. 9. The average behavioral deviation regarding all features after two weeks.

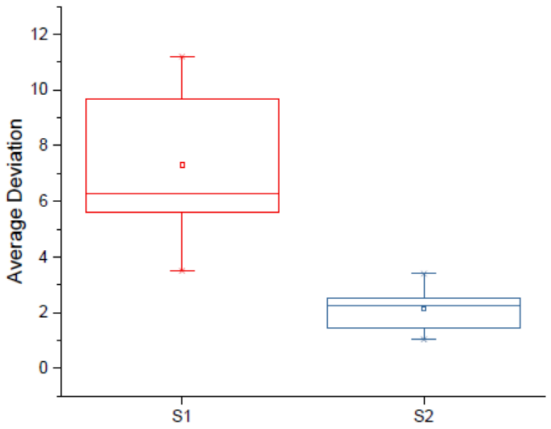


Fig. 10. The distribution of average deviation after two weeks.

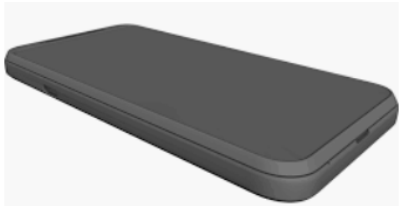
It is found that after two weeks, the behavioral deviation under S2 is much smaller than those under S1, i.e., some features' deviations are smaller than 2. In other words, users' touch gestures were much more stable under S2 than those under S1.

Result Analysis – Long-term Authentication - 2

- For authentication accuracy, we applied the same five classifiers on the new sessions without re-training. That is, we used the already built behavioral model (before two weeks) for each classifier.
- It is found that SVM still could achieve a smaller AER under two situations, but the rate is much different, i.e., it reached a rate of 3.68% and 9.82% under S2 and S1, respectively.
- The results validated that users' touch behavior could become relatively stable when they play with social networking applications, making it easier to build a robust authentication scheme for a long-term period.

Outline

- Background and Motivation
- SocialAuth
- Open Challenge / Discussion



Discussion & Limitations - 1

Touchscreen Size

- Android 2.2 vs. Android 9.0
- Much higher resolution that the one considered (1920×1080 vs 800x480)

Attacks Investigation

- How easy is to pretend being another user?
- Is it possible to learn the features of another user?
- How robust is the proposal when one of the users is malicious and wants to emulate a target user?

Discussion & Limitations - 2

Concrete Social Media Applications

- Each user may use different social media applications. Will this affect the overall authentication performance in terms of error rate ?

Phone Change

- When a use changes a phone, how many efforts are needed to adopt this solution on a new phone ?

Phone Type

- Phone type? iPhone?

Q&A

If you have any question, you can contact via

weme@dtu.dk