Liveness is Not Enough: Enhancing Fingerprint Authentication with Behavioral Biometrics to Defeat Puppet Attacks

Cong Wu, Kun He, and Jing Chen, *Wuhan University;* Ziming Zhao, *Rochester Institute of Technology;* Ruiying Du, *Wuhan University*

https://www.usenix.org/conference/usenixsecurity20/presentation/wu

This paper is included in the Proceedings of the 29th USENIX Security Symposium.

August 12–14, 2020

978-1-939133-17-5

Open access to the Proceedings of the 29th USENIX Security Symposium is sponsored by USENIX.

Liveness is Not Enough: Enhancing Fingerprint Authentication with Behavioral Biometrics to Defeat Puppet Attacks

Cong Wu[1,2], Kun He[1*], Jing Chen[1,2*], Ziming Zhao[3], Ruiying Du[1]

{cnacwu, hekun, chenjing, duraying}@whu.edu.cn, zxzics@rit.edu

[1]*School of Cyber Science and Engineering, Wuhan University*,
[2]*Shenzhen Institute of Wuhan University*, [3]*Rochester Institute of Technology*

Abstract

Fingerprint authentication has gained increasing popularity on mobile devices in recent years. However, it is vulnerable to presentation attacks, which include that an attacker spoofs with an artificial replica. Many liveness detection solutions have been proposed to defeat such presentation attacks; however, they all fail to defend against a particular type of presentation attack, namely *puppet attack*, in which an attacker places an unwilling victim's finger on the fingerprint sensor. In this paper, we propose FINAUTH, an effective and efficient software-only solution, to complement fingerprint authentication by defeating both synthetic spoofs and puppet attacks using *fingertip-touch* characteristics. FINAUTH characterizes intrinsic fingertip-touch behaviors including the acceleration and the rotation angle of mobile devices when a legitimate user authenticates. FINAUTH only utilizes common sensors equipped on mobile devices and does not introduce extra usability burdens on users. To evaluate the effectiveness of FINAUTH, we carried out experiments on datasets collected from 90 subjects after the IRB approval. The results show that FINAUTH can achieve the average balanced accuracy of 96.04% with 5 training data points and 99.28% with 100 training data points. Security experiments also demonstrate that FINAUTH is resilient against possible attacks. In addition, we report the usability analysis results of FINAUTH, including user authentication delay and overhead.

1 Introduction

In recent years, fingerprint sensors have been widely integrated into smartphones and tablets. Combined with Fast IDentity Online (FIDO) [11] and other protocols, a fingerprint sensor enables applications [71], such as mobile banking, to locally authenticate end users instead of asking them to type passwords on a small touchscreen [1, 7]. It is estimated that 920 million global shipments of smartphones (about 64%) were equipped with a fingerprint sensor in 2017, and the number will increase to 1.25 billion (about 75%) by 2020 [8].

However, fingerprint authentication is vulnerable to presentation attacks [70], where attackers bypass the authentication using artificial crafts, e.g. gummy fingers that have fingerprint impressions, or human-based instruments [39]. To defend against presentation attacks, hardware-based solutions rely on additional hardware to acquire biological traits, such as blood pressure [42], odor [15], oxygen saturation [59], heartbeat [10], and electrocardiograph [40]. And, software-based solutions utilize image processing to extract more discriminative physical characteristics, such as the size of fingerprint ridges [55], density [26], continuity [58], texture [27], and train the detection model via machine learning methods to enhance the security against fingerprint spoofs [30, 56].

Unfortunately, existing methods to enhance the security of fingerprint authentication only focus on liveness detection, which determines whether the input fingerprint comes from a live human being. These systems are powerless against *puppet attacks*, in which an attacker places an unwilling but legitimate victim's finger on the fingerprint sensor, e.g., the victim is sleeping or passed out. Puppet attack was highlighted in ISO/IEC 30107 [39], and of increasing concern because it is easy to perform [2]. Because the fingerprint and other biological traits are collected from the real and legitimate user in puppet attacks, existing liveness detection methods all fail [4].

Even though combining fingerprint with behavioral biometrics is a promising approach in defeating puppet attacks, existing behavioral biometrics, including keystroke dynamic [34], gesture pattern [65], and gait pattern [49], are not suitable to enhance the security of fingerprint authentication due to the following reasons: i) these methods place extra usability burdens on users by requiring additional gestures; ii) these methods rely on behavioral biometric information collected in a relatively long time, e.g. more than 1 second [65], while fingerprint authentication happens in 0.29 seconds on average based on our experiments (Section 7.4). The key challenge in designing a practical puppet-attack-resistant fingerprint

*The corresponding authors are Kun He and Jing Chen.

authentication is to detect impostors promptly without undermining the usability of fingerprint authentication.

To overcome this challenge, we utilize the intrinsic *fingertip-touch* characteristics to model users' movements in legitimate authentications to defend against all presentation attacks, including the puppet attack. The term of fingertip-touch in this paper refers to the behavior completed in an instant when a user gets the mobile device in hand and applies his/her finger to fingerprint sensors. We model these movements with *acceleration* and *rotation angle*, which can be retrieved from built-in sensors, such as accelerometer, magnetometer, and gyroscope. This is inspired by the fact that users place their fingers on a fingerprint sensor to perform authentication repeatedly (average 50 times a day [72]) and these habitual behaviors form stationary and unique muscle memory [9, 63]. We identify latent time- and frequency-domain features, and use the convolutional neural network (CNN) to extract discriminative features from characterized behavior, i.e., accelerations and rotation angles. We develop an effective and efficient authentication system named FINAUTH, which can be easily deployed on mobile devices as auxiliary authentication for fingerprint authentications without introducing additional hardware or gestures.

Attack Models. We consider the following three types of attacks: i) *Artificial replica attack:* the attacker can forge fake fingerprints to spoof the fingerprint system [17]; ii) *Puppet attack:* the attacker can put an unwilling victim's finger on the fingerprint sensor [39]; iii) *Mimicry attack:* the attacker knows how our approach works and attempts to defeat our approach by mimicking the victim's movements in authentication [34]. FINAUTH can defeat the first two types of attackers. Also, it is difficult for the third type of attackers to bypass FINAUTH.

The contributions of this paper are summarized as follows:

- We propose FINAUTH to complement fingerprint authentication for defending presentation attacks, including the puppet attack. FINAUTH models a user's intrinsic fingertip-touch behavior during fingerprint authentication. FINAUTH uses built-in sensors and does not require additional hardware.
- To evaluate the performance of FINAUTH, we collected a dataset of fingertip-touch behavior data from 90 subjects. Our experimental results show that FINAUTH can achieve a balanced accuracy of 96.04% with only 5 training data points, while the balanced accuracy can be improved to 99.28% with 100 training data points.
- We demonstrate the security of FINAUTH in defeating three types of attacks, including artificial replica attacks, puppet attacks, and mimicry attacks. Experiment results show that attack success rates are all below 0.3% under the authentication model trained using 100 data points.

The rest of this paper is organized as follows. Section 2 presents the overview of FINAUTH. In Section 3, we introduce the data preprocessing and the method to characterize fingertip-touch behaviors. Sections 4 and 5 illustrate feature processing and classification approaches. We describe details of experiment design and data collection in Section 6. Section 7 reports experimental results of reliability, security, and usability. We review related work in Section 8, and discuss our study in Section 9. Section 10 concludes this paper.

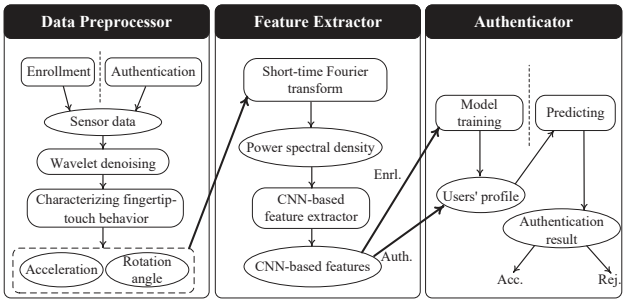


Figure 1: The workflow of FINAUTH.

2 Overview of FINAUTH

Similar to most authentication schemes, FINAUTH consists of two phases: enrollment and authentication. In enrollment, FINAUTH builds a user profile from the first successful fingerprint authentications. After a user profile is built, FINAUTH enters the authentication phase, in which FINAUTH assists the fingerprint sensor to authenticate a user.

FINAUTH only employs built-in sensors on smart devices, including accelerometer, gyroscope, and magnetometer, to sense phone movements incurred by fingertip-touch behaviors. The accelerometer and gyroscope are motion sensors, which can monitor device movement. The magnetometer is a position sensor to determine a device's physical position in the real frame of reference, which is leveraged for data calibration to acquire more precise motion information.

As shown in Figure 1, FINAUTH consists of three modules, including *data preprocessor*, *feature extractor*, and *authenticator*. The data preprocessor runs in the background to monitor fingerprint authentication events. Upon detecting fingerprint-inputting, data preprocessor starts to collect accelerometer, gyroscope, and magnetometer data. Then, data preprocessor uses wavelet denoising method to reduce noise. FINAUTH characterizes fingertip-touch behaviors using accelerations and rotation angles. For the feature extractor, FINAUTH generates power spectral density for characterized fingertip-touch behavior information using short-time Fourier transform (STFT), and then uses CNN-based feature extractor to extract features. To profile legitimate users with only successful login data points, FINAUTH trains a machine learning model based on a one-class classifier, which is later used for authentication.

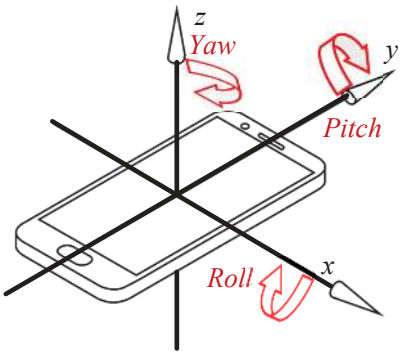


Figure 2: Roll, pitch, and yaw.

3 Data Preprocessing

In this section, we present the data collection and preprocessing approaches adopted by FINAUTH. We also illustrate how FINAUTH characterizes fingertip-touch behaviors.

3.1 Data Collection and Denoising

Data collection. Once a user places her finger on the fingerprint sensor, FINAUTH starts to collect accelerometer, gyroscope, and magnetometer data for a short period t with the sampling rate f_s. For each authentication attempt, FINAUTH collects n ($n = t \times f_s$) samples of sensor data. Each sample is 9-dimensional denoted as $(a_x^r, a_y^r, a_z^r, g_x^r, g_y^r, g_z^r, m_x^r, m_y^r, m_z^r)$, where r stands for raw data, a, g, m represent accelerometer, gyroscope, and magnetometer data respectively, and x, y, and z represent the three axes. We use a row vector, e.g. $\boldsymbol{a}^r = (a_x^r, a_y^r, a_z^r)$, to denote a data sample from a sensor and use a column vector, e.g. $\boldsymbol{a}_x^r = (a_{x,1}^r, ..., a_{x,n}^r)^T$, to represent all n samples at one axis (e.g. x-axis).

Denoising. Because slight vibrations, even sounds, can introduce measurable noise to the built-in sensors [43], it is important to reduce the noise from the sensed data. We apply wavelet denoising [79], which is widely used in signal processing, on the column vectors of the sensed data (e.g. $\boldsymbol{a}_x^r$). A denoised sample is represented as $(a_x, a_y, a_z, g_x, g_y, g_z, m_x, m_y, m_z)$.

3.2 Characterizing Fingertip-touch Behaviors

From the denoised data, we use accelerations and rotation angles to characterize fingertip-touch behaviors.

Accelerations. Accelerations of a device can represent the dynamic force acting upon a device from a user. We use the accelerations along the three axes at the device coordinate system (a_x, a_y, a_z) and the net acceleration $(a' = \sqrt{a_x^2 + a_y^2 + a_z^2})$ to model fingertip-touch characteristics. The coordinate system of a smartphone is shown as Figure 2.

Rotation angles. A fingertip-touch behavior also causes a device to rotate slightly. As shown in Figure 2, we use the classical Euler angle parameterization to represent the rotations, which are denoted as *roll* (ϕ), *pitch* (θ), and *yaw* (ψ). We compute the rotation angles using the sensed data through the following steps [16, 73]:

1) the coarse angles $(\phi^c, \theta^c, \psi^c)$ are computed using accelerometer and magnetometer data as shown in Eq. 1, 2, and 3 [75].

$$\phi^c = \arctan\left(\frac{-a_y}{\sqrt{-a_x^2 + a_z^2}}\right) \tag{1}$$

$$\theta^c = \arctan\left(\frac{-a_x}{a_z}\right) \tag{2}$$

$$\psi^c = \arctan \frac{sin(\phi^c)sin(\theta^c)m_x + cos(\phi^c)m_y + sin(\phi^c)cos(\phi^c)m_z}{cos(\theta^c)m_x + sin(\theta^c)m_z}. \tag{3}$$

2) to get more accurate angles, we then use the gyroscope data to get the partial derivatives of ϕ, θ, ψ with respect to time $(\frac{\partial(\phi)}{\partial(t)}, \frac{\partial(\theta)}{\partial(t)}, \frac{\partial(\psi)}{\partial(t)})$. The gyroscope measures the angular velocity, and the dynamic angle can be obtained by integrating the angular velocity, which is given in Eq. 4 [57].

$$\begin{bmatrix} \frac{\partial(\phi)}{\partial(t)} \\ \frac{\partial(\theta)}{\partial(t)} \\ \frac{\partial(\psi)}{\partial(t)} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi^c)tan(\theta^c) & sin(\phi^c)tan(\theta^c) \\ 0 & cos(\phi^c) & -sin(\phi^c) \\ 0 & sin(\phi^c)/cos(\theta^c) & cos(\phi^c)/cos(\theta^c) \end{bmatrix} \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \tag{4}$$

3) we then use extended Kalman filter (EKF) to perform sensor data fusion, which is widely used for state estimation and tracking due to its robustness in nonlinear dynamic environments [52]. The EKF method takes time-varying drift into account via defining an error metric and updating covariance metric iteratively to minimize this error. Specifically, the system state vector $\boldsymbol{x}$ of EKF in our work is given as Eq 5.

$$\boldsymbol{x} = [\boldsymbol{q}^T, \boldsymbol{w}^T]^T = [q_0, q_1, q_2, q_3, \frac{\partial(\phi)}{\partial(t)}, \frac{\partial(\theta)}{\partial(t)}, \frac{\partial(\psi)}{\partial(t)}]^T \tag{5}$$

where T denotes the transpose operator, $\boldsymbol{w}^T = [\frac{\partial(\theta)}{\partial(t)}, \frac{\partial(\psi)}{\partial(t)}, \frac{\partial(\psi)}{\partial(t)}]$, which are estimated values with Eq. 4. $\boldsymbol{q}$ is the quaternion (four-element vector), which can be acquired based on the relationship between Euler Angles and quaternion as shown in Eq. 6.

$$\boldsymbol{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} cos\frac{\phi^c}{2}cos\frac{\theta^c}{2}cos\frac{\psi^c}{2} + sin\frac{\phi^c}{2}sin\frac{\theta^c}{2}sin\frac{\psi^c}{2} \\ sin\frac{\phi^c}{2}cos\frac{\theta^c}{2}cos\frac{\psi^c}{2} - cos\frac{\phi^c}{2}sin\frac{\theta^c}{2}sin\frac{\psi^c}{2} \\ cos\frac{\phi^c}{2}sin\frac{\theta^c}{2}cos\frac{\psi^c}{2} + sin\frac{\phi^c}{2}cos\frac{\theta^c}{2}sin\frac{\psi^c}{2} \\ cos\frac{\phi^c}{2}cos\frac{\theta^c}{2}sin\frac{\psi^c}{2} - sin\frac{\phi^c}{2}sin\frac{\theta^c}{2}cos\frac{\psi^c}{2} \end{bmatrix} \tag{6}$$

where ϕ^c, θ^c, and ψ^c are estimated with the fusion of both accelerometer and magnetometer based on Eq. 1, 2, and 3. q_1, q_2, q_3, q_4 are elements of the unit quaternion.

Table 1: Time- and frequency-domain features and their normalized fisher's scores.

Domain	Feature	Description	Normalized Fisher Score of $(\boldsymbol{a}_x, \boldsymbol{a}_y, \boldsymbol{a}_z, \boldsymbol{a}', \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi})$
Time	Mean	The mean of the time series.	$(0.45, 0.01, 0.22, \mathbf{0.68}, \mathbf{0.86}, \mathbf{0.84}, \mathbf{0.84})$
	Standard deviation	The standard deviation of the time series.	$(0.24, 0.56, 0.31, 0.41, 0.58, 0.32, \mathbf{0.74})$
	Relative standard deviation	The extent of variability in relation to its mean.	$(0.34, 0.15, 0.12, 0.56, \mathbf{0.71}, \mathbf{0.64}, \mathbf{0.82})$
	Sum of absolute differences	The sum over the absolute value of consecutive changes in the time series.	$(0.32, 0.27, \mathbf{0.72}, 0.52, 0.53, \mathbf{0.72}, \mathbf{0.78})$
	Absolute energy	The absolute energy of the time series.	$(\mathbf{0.63}, \mathbf{0.98}, \mathbf{0.85}, 0.57, \mathbf{0.72}, 0.57, 0.37)$
	Autocorrelation	The autocorrelation of the time series.	$(0.00, 0.14, 0.15, 0.21, \mathbf{0.94}, \mathbf{0.62}, \mathbf{0.64})$
Frequency	Spectral centroid	The center of mass of the spectrum is located.	$(0.34, 0.21, 0.38, 0.12, \mathbf{0.78}, \mathbf{0.98}, \mathbf{0.78})$
	Spectral spread	The average spread of the spectrum in relation to its centroid.	$(\mathbf{0.66}, 0.36, 0.32, \mathbf{0.78}, 0.46, \mathbf{0.82}, \mathbf{0.96})$
	Spectral skewness	The measurement of the asymmetry of the probability distribution of a real-valued random variable about its mean.	$(\mathbf{0.85}, 0.45, 0.58, \mathbf{0.84}, 0.56, \mathbf{0.85}, \mathbf{1.00})$
	Spectral kurtosis	The shape of a probability distribution.	$(0.34, 0.17, \mathbf{0.70}, \mathbf{0.86}, \mathbf{0.62}, 0.51, 0.42)$
	Power spectral density	Average of distribution of power into frequency components.	$(\mathbf{0.90}, \mathbf{0.71}, \mathbf{0.86}, 0.26, \mathbf{0.85}, \mathbf{0.68}, \mathbf{0.82})$
	Spectral entropy	The complexity of the signal in the frequency domain.	$(\mathbf{0.94}, 0.32, \mathbf{0.82}, 0.21, \mathbf{0.96}, \mathbf{0.82}, \mathbf{0.89})$

We compute accurate quaternions, where the detailed steps are presented in Appendix B due to the page limit. Finally, rotation angles can be computed based on Eq. 7.

$$\begin{cases} \gamma = & \arctan\left(\frac{2q_2q_3+2q_0q_1}{2q_0^2+2q_3^2-1}\right) \\ \theta = & -\arcsin\left(2q_1q_3-2q_0q_2\right) \\ \psi = & \arctan\left(\frac{2q_1q_2+2q_0q_3}{2q_0^2+2q_1^2-1}\right) \end{cases} \tag{7}$$

The outcome of characterizing fingertip-touch behaviors is represented as $(\boldsymbol{a}_x, \boldsymbol{a}_y, \boldsymbol{a}_z, \boldsymbol{a}', \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi})$, where each of element is an n-dimensional vector.

4 Feature Extraction

We present two methods to extract discriminative features from fingertip-touch behaviors.

4.1 Time- and Frequency-domain Features

We extract features in the time- and frequency-domain from $(\boldsymbol{a}_x, \boldsymbol{a}_y, \boldsymbol{a}_z, \boldsymbol{a}', \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi})$. As shown in Table 1, we extract six statistical features in the time domain, including mean, standard deviation, relative standard deviation, sum of absolute differences, absolute energy, and autocorrelation. In addition, we apply fast Fourier transform and extract another six features in the frequency domain. These features include spectral centroid, spread, skewness, kurtosis, power density, and entropy. These time- and frequency-domain features are widely used for time series analysis [24, 44, 46].

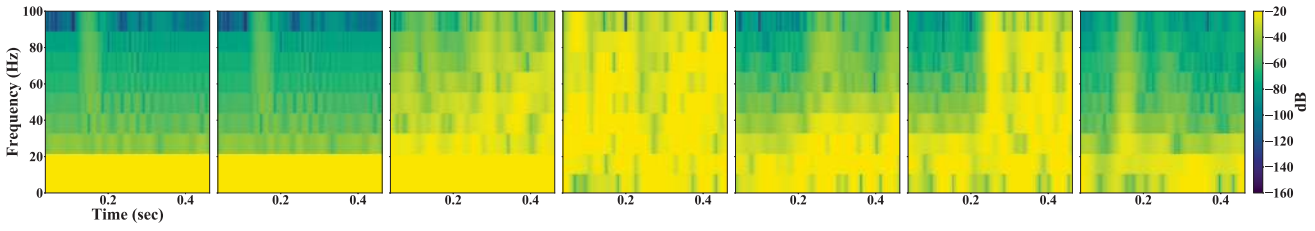
Selected Features. We computed the Fisher's scores [35] for all aforementioned 84 features with 45,000 data points collected from 90 users to select the most discriminative features. As the results show in Table 1, the features from rotation angle have higher Fisher's score than features from acceleration. Features with a normalized Fisher's score higher than 0.6 are selected. The output of features extraction and selection in time and time-domain is a 43-dimensional feature vector.

4.2 CNN-based Feature Learning

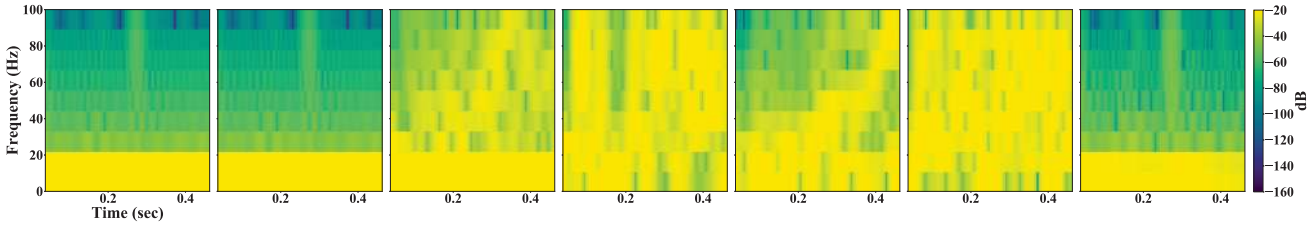
Besides the extracted time- and frequency-domain features, we also resort to CNN-based feature learning. To this end, we first apply STFT and convert the time series data (e,g., $\boldsymbol{a}_x$) to a two-dimensional power spectral density matrix. Then, we concatenate these matrices and rely on CNN models to extract features from them. Figure 3 shows three users' spectrograms from $\boldsymbol{a}_x$, $\boldsymbol{a}_y$, $\boldsymbol{a}_z$, $\boldsymbol{a}'$, $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$, which are visual representations of power spectral density matrices.

The basic idea of feature learning with CNN is to leverage the output of the model's intermediate layer as features thanks to the powerful feature representation of deep learning method [13, 67]. In particular, we train the CNN model to distinguish different users with collected FINAUTH data, and employ the first k layers of the trained model as the feature extractor. Even though the model is trained from a limited dataset, it can be used to extract generalized features because of the feature learning ability of CNN, which is also known as *transfer learning* [77].

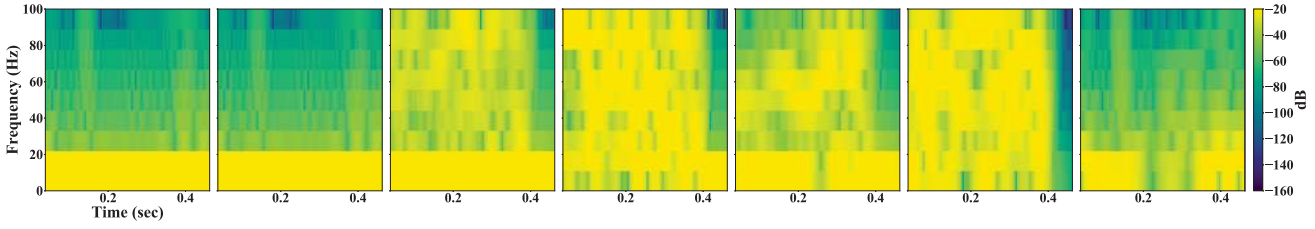
Table 2 shows the structure of our used CNN model. We use leaky rectified linear units (Leaky ReLu) as the activation functions for two-dimensional convolution (Conv2d) layers and fully connected (FC) layers, since it can tackle the vanishing gradient problem during the model training phase [50]. For the pooling layers, we use the max-pooling method to down-sample the input, which controls over-fitting and saves computational costs by reducing the number of parameters for training. To avoid over-fitting, we add dropout layers after each pooling layer. Furthermore, we also consider batch normalization (BN) layers to normalize the output of the previous layer, which accelerates model training and increases



(a) User A.



(b) User B.



(c) User C.

Figure 3: Characterized fingertip-touch behaviors of three users under STFT. From left to right, spectrograms of $\boldsymbol{a}_x$, $\boldsymbol{a}_y$, $\boldsymbol{a}_z$, $\boldsymbol{a}'$, $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$.

the stability of the model. The softmax layer is added as the last layer for prediction, which outputs the categorical probability distribution of each class. Specifically, the kernel size of Conv2d and pooling layers is set as 3×3 and 2×2 respectively, because of their better non-linear feature representation gaining popularity in start-of-art models [36, 38, 67]. The detailed output shape and the number of parameters of each layer are given as Table 2. The total model contains 202,974 parameters, including 202,438 trainable and 536 non-trainable parameters.

Table 2: The structure of base CNN model.

# Layer	Layer Type	Output Shape	# Para
1	Conv2d + LeakyReLu	$62 \times 126 \times 24$	1,536
2	Conv2d + LeakyReLu	$60 \times 124 \times 24$	5,208
3	Pooling + Dropout +BN	$30 \times 62 \times 24$	96
4	Conv2d + LeakyReLu	$28 \times 60 \times 48$	10,416
5	Conv2d + LeakyReLu	$26 \times 58 \times 48$	20,784
6	Pooling + Dropout +BN	$13 \times 29 \times 48$	192
7	Conv2d + LeakyReLu	$11 \times 27 \times 16$	6,928
8	Conv2d + LeakyReLu	$9 \times 25 \times 16$	2,320
9	Pooling + Dropout +BN	$4 \times 12 \times 16$	64
10	Flatten	768	0
11	FC+LeakyReLu	180	139,140
12	FC+ Softmax	90	16,290

5 Authentication With One-class Classifiers

In real-world fingerprint authentication settings, the training dataset only contains the legitimate user's data points. Therefore, it is a one-class classification problem. We use four methods to profile the legitimate user: i) Pearson correlation coefficient-based similarity comparison (PCC), ii) one-class support vector machine (OC-SVM), iii) local outlier factor (LOF), and iv) isolation forest (IF).

PCC is a similarity metric to measure the linear correlation between two variables. The coefficient is between +1 and -1, where +1/-1 denotes a total positive or negative linear correlation, and 0 represents none linear correlation. Specifically, after feature extraction, we compute the mean PCC between the extracted feature vector and fingertip-touch templates (i.e., saved feature vector during the register phase). The computed mean PCC is then used to decide whether the user is authorized.

OC-SVM, an extended algorithm of SVM, maps data points into high-dimensional feature space with the kernel func-

Table 3: Summary of the compiled datasets

Dataset	Week of Collection	# of Subjects / Attackers	Postures	Device	# of Data Points
1	1 †, 8 and 9 ‡	90	Sitting, standing, lying, walking, running	OnePlus3	63,000
2A	2, 3, 5, 7 †	24, 24, 22, 21	Sitting	OnePlus3	18,200
2B	10, 11, 12, 13 ‡	62, 61, 59, 53			47,000
3	Added Aug. 2019	64	Sitting	Xperia XZ1, Oneplus5, Vivo X21	3,200
4A	2 †, 10 and 11 ‡	15	Sitting	OnePlus3	3,600
4B					3,600
4C					3,600

†: Data collected at the university; ‡: data collected at the company.

tion and finds the surface of a minimal hyper-sphere which contains the objective data points as many as possible. The distance between data points and the hyper-sphere is the classification score, which is leveraged to conduct prediction. OC-SVM has been successfully applied to many anomaly detection problems, such as utterance verification [37], malware detection [31], and online fault detection [78].

LOF measures the local deviation of the data point to its neighbors [18]. It decides whether a data point is an outlier using the anomaly score depending on the local density. Specifically, locality density is estimated by k-nearest neighbors based on a given distance metric. A data point with a substantially lower density than their neighbors will be regarded as an outlier.

IF is a rapid one-class classification method for high-dimensional data based on ensemble learning, which assumes that abnormal data points are easier to isolate from given one-class instances [47]. IF detects abnormal data points by subsampling the dataset to construct *iTrees*, and further integrate multiple iTrees into a forest to detect abnormal data. A data point is seen as abnormal when these random trees collectively produce shorter path lengths for it.

6 Experiment Design and Data Collection

To collect the experiment data, we develop a prototype system on Android 7.1 (API level 25). Specifically, our implementation hooks the `authenticate()` method from the `FingerprintManager` class. We set the data collection time (t) as 0.5 seconds and the sampling rate (f_s) as 200 Hz.

After receiving the IRB approval from our university in June 2018, we started recruiting subjects for the data collection, which lasted for 5 months. To qualify for the experiment, a subject must self-identify as a frequent smartphone user who had been using fingerprint authentication for more than a year. 90 subjects were involved in finger-tip behavior data collection, who were aged from 22 to 45. 39 subjects were female, and 51 were male. 24 of them were students in our university, and the rest were employees in a company. Another 15 subjects (4 from our university, 11 from the company), including 4 females and 11 males, were recruited to play the role of an attacker to carry out artificial replica attack, puppet attack and mimicry attack on the 90 subjects.

We explained to each subject the purpose of this research project, the data we collect, and the steps we take to protect their personal identifiable information. During the data collection, we asked each subject to hold a smartphone in hand as they normally unlock their own devices. To help collect more distinct data points, we also suggested that they hold the device in different angles and directions. Table 3 summarizes the compiled 4 datasets:

1) *Dataset-1*. For this dataset, we used one smartphone (OnePlus 3 with 6G RAM) to eliminate factors that could be introduced by different phones. This device has a capacitive fingerprint sensor that is integrated with the home button. In week 1, the 24 subjects from our university were first asked to enroll their fingerprints on the phone. Then, a subject needed to perform successful fingerprint logins for 500 times while sitting (stationary), and for 50 times while standing (stationary), lying (stationary), walking (moving), and running (moving), respectively. Note that we only collect the finger-tip behavior data when a login is successful. In week 8 and 9, the 66 subjects from the company went through the same data collection procedure. Each subject spent 13 - 17 minutes to finish this task. As a result, we collected $90 \times 700 = 63,000$ data points for the *dataset-1*.

2) *Dataset-2*. To evaluate the consistency of the fingertip-touch behavior features over the long term, we compiled the *dataset-2* with the same subjects after some time intervals: i) *dataset-2A*. The 24 subjects from our university came in week 2, 3, 5, 7 to perform 50 successful fingerprint authentications while sitting; ii) *dataset-2B*. The subjects in the company did the same thing in week 10, 11, 12 and 13. Some subjects did not show up for all the collections. As a result, we collected 65,200 data points in total for the *dataset-2*.

3) *Dataset-3*. To evaluate the generalization of FINAUTH on different devices, we collected the *dataset-3* on 3 smartphones: Xperia XZ1 (side fingerprint sensor), Oneplus 5 (back fingerprint sensor), and Vivo X21 (in-screen fingerprint sensor). The 22 subjects from our university were assigned to Xperia XZ1, while the 42 subjects from the company were

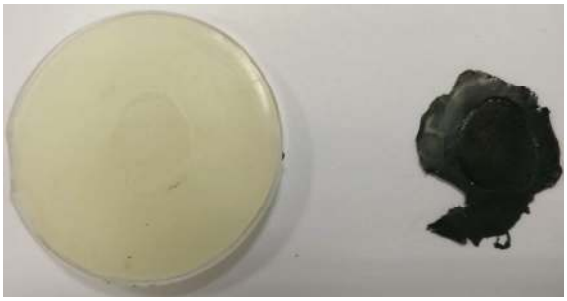


Figure 4: Artificial fingerprint replica. The left is the mold used to capture fingerprint; the right is a fake fingerprint crafted using silicone rubber.

assigned to the other two devices randomly. Each subject was asked to conduct 50 authentications while sitting. As a result, we collected 3,200 data points for the *dataset-3*.

4) *Dataset-4*. We used artificial replica attack, puppet attack and mimicry attack to evaluate the effectiveness of FINAUTH. It is infeasible to ask each attacker to attack all 90 subjects in all three experiments. To increase the chance of successful attacks, we collected the fingertip-touch data of the 15 attackers and used Pearson correlation distance matrix to compute the distance between each attacker and each subject. Then, we assign each attacker 6 subjects as his/her targets on the basis of fingertip-touch behavioral similarity:

i) *Dataset-4A: artificial replica attack*. We crafted a fingerprint spoof using the silicone rubber, as shown in Figure 4, for each of the 85 subjects (5 dropped out). The spoofs were tested to make sure they can spoof the original fingerprint authentication. After the experiments, the molds and synthetic spoofs were destroyed. Each attacker was asked to spoof the fingerprint sensor while sitting for 50 attempts per subject. We collected $50 \times 85 = 4,250$ data points for the *dataset-4A*;

ii) *Dataset-4B: puppet attack.* Each attacker was asked to hold the device in her/his hand and place a subject's finger on the fingerprint sensor 50 times while both of them in sitting. We collected 4,250 data points for the *dataset-4B*. Note that the unwillingness for this study is a subset of all possible puppet attacks since we do not have data on other kinds of unwillingness, e.g. the victim is sleeping or passed out;

iii) *Dataset-4C: mimicry attack*. Each attacker was asked to carefully observe a subject's hand and device movement in a close distance (no more than 2 feet). After the attacker was confident about what they observed, she/he would mimick the subject's fingertip-touch behavior with the crafted fingerprint spoofs for 50 times. We collected 4,250 data points for the *dataset-4C*.

7 Evaluation

In this section, we report the evaluation results of the proposed system. Section 7.1 presents the metrics we used in measuring the performance. Section 7.2 shows evaluation on how distinguishable users' fingertip-touch behaviors are under different conditions using *dataset-1, 2,* and *3*. Section 7.3 evaluates FINAUTH's effectiveness against presentation attacks using *dataset-4*. Section 7.4 presents system performance of FINAUTH. Section C reports user acceptance of FINAUTH. Section 7.5 illustrates other design considerations behind FINAUTH.

Specifically, the base CNN was trained using cross-entropy as the loss function based on half (22,500) data points of *dataset-1* (collected while sitting) containing fingertip-touch behavior data from 90 classes (subjects). We pre-trained base model on a PC with Intel i5-8300 CPU, 16GB RAM, GTX 1060 GPU, and the training process took 42 minutes. Keras with TensorFlow backend was used for training. The size of the total model is 1.54 MB, which is lightweight on mobile devices.

7.1 Evaluation Metrics

We use the following metrics to evaluate the effectiveness of FINAUTH. True acceptance (TA) means fingertip-touch behaviors from legitimate users are correctly identified. True rejection (TR) means fingertip-touch behaviors not from legitimate users are correctly declined. False acceptance (FA) means fingertip-touch behaviors not from legitimate users are incorrectly identified as legitimate. False rejection (FR) means fingertip-touch behaviors from legitimate users are incorrectly rejected. False acceptance rate (FAR) is defined as $\frac{FA}{FA+TR}$, which measures the proportion of illegal users who gain access. False rejection rate (FRR) is defined as $\frac{FR}{FR+TA}$, which measures the proportion of legitimate users who are denied access. Balanced accuracy (BAC) is a metric used for evaluating models trained from unbalanced data [19]. It is defined as the average between true rejection rate ($TRR = \frac{TR}{TR+FA}$) and true acceptance rate ($TAR = \frac{TA}{TA+FR}$). We also use receiver operation characteristic (ROC) curves to show dynamic changes of TAR against FAR at a varying decision threshold for performance comparison. The area under the ROC curve (AUC) is used to estimate the probability that prediction scores of authorized users are higher than unauthorized users. While in presentation attacks resistance evaluation, we leverage FAR, i.e., attack success rate, as the evaluation criteria, which is the ratio between the number of incorrectly identified data points and the number of all attack data points. It implies the probability of attackers bypassing the authentication system. Note that, FAR is more important in fingerprint authentication, e.g., achieving FAR as low as 10^{-6} while still maintaining an FRR of 1% [5].

7.2 Reliability Analysis

To find out how distinguishable each user's fingertip-touch behaviors are, we randomly split each user's data points, train

Table 4: BAC (%) under different k for CNN-based feature learning.

# Layer k	PCC	OC-SVM	LOF	IF
3	86.72	78.69	84.96	86.15
6	91.27	82.67	87.28	88.91
9	93.53	84.32	94.34	90.09
11	94.65	90.69	**97.99**	93.63

a model for each of them, and use her/his remaining data points and other users' data points to evaluate the model. We report the performance of using different feature sets, classifiers, training dataset size, and datasets in the rest of this section.

7.2.1 Different Feature Sets and Classifiers

CNN-based Feature Learning. We trained the base CNN with 22,500 sitting data points in the *dataset-1*, and then leveraged the output of the base model's intermediate layer (k_{th} layer) as extracted features. To find the optimal k, we evaluated each classifier's performance with 30 training data points from the first pooling layer (3_{rd} layer) to the first fully-connect layer (11_{th} layer). Table 4 shows the averaged BAC when using features extracted with different layers under different classifiers. As the results show, with the features from the 11_{th} layer, classifiers achieve higher BAC.

Results. After determining the best k for the CNN-based feature learning, we obtained three feature sets: i) time- and frequency-domain features (TFF) extracted via feature extraction and selection (Section 4.1); ii) CNN-based features (CNF) extracted with the pre-trained model (Section 4.2); and iii) the union of feature sets of the aforementioned two (UnF).

We used the grid search to find the best parameter combinations for each classifier. For OC-SVM, we found radial basis function works best with $\gamma = 0.25$ and $\nu = 0.1$. For IF, the optimal parameter of `n_estimators` was 20. For LOF, we used *Minkowski* distance as the distance metric with the optimal parameter of `n_neighbors` as 5.

Figure 5 shows ROC curves of using the three feature sets under different one-class classifiers. The results indicate that CNN-based features are more discriminative than time- and frequency-domain features. Specifically, for PCC and LOF, the BAC of models using CNN-features is significantly higher than using time- and frequency-domain features. However, the performance of OC-SVM and IF of CNN-based features is poorer. Another observation is that the union of two feature sets brings slight improvement over only one feature set. Table 5 shows the BAC, FAR, FRR, and AUC under different feature set and classifier combinations. Even though UnF + LOF has the best BAC, CNF + LOF is the most reliable model with low FAR. For the rest of the evaluations, we use the CNF + LOF approach.

Table 5: BAC (%), FAR (%), FRR (%), and AUC under three different feature sets and four different one-class classifiers.

Feature Set + Classifier	BAC	FAR	FRR	AUC
TFF + PCC	84.41	11.85	19.34	0.9169
TFF + OC-SVM	91.49	5.56	11.45	0.9656
TFF + LOF	93.28	4.32	9.13	0.9767
TFF + IF	96.07	2.51	5.35	0.9915
CNF + PCC	94.65	3.30	7.40	0.9871
CNF + OC-SVM	90.69	6.41	12.21	0.9532
CNF + LOF	**97.99**	**0.86**	3.16	0.9974
CNF + IF	93.63	3.72	9.02	0.9789
UnF + PCC	94.76	2.86	7.62	0.9888
UnF + OC-SVM	93.78	4.06	8.37	0.9806
UnF + LOF	**98.02**	1.52	2.43	0.9975
UnF + IF	96.88	2.03	4.21	0.9938

Table 6: Mean BAC (%), FAR (%), FRR (%), and AUC with non-overlapping subjects in training base CNN and testing.

Feature Set + Classifier	BAC	FAR	FRR	AUC
CNF + LOF	95.34	4.20	5.10	0.9805
UnF + LOF	95.59	3.35	5.47	0.9867

7.2.2 Performance with Non-overlapping Subjects

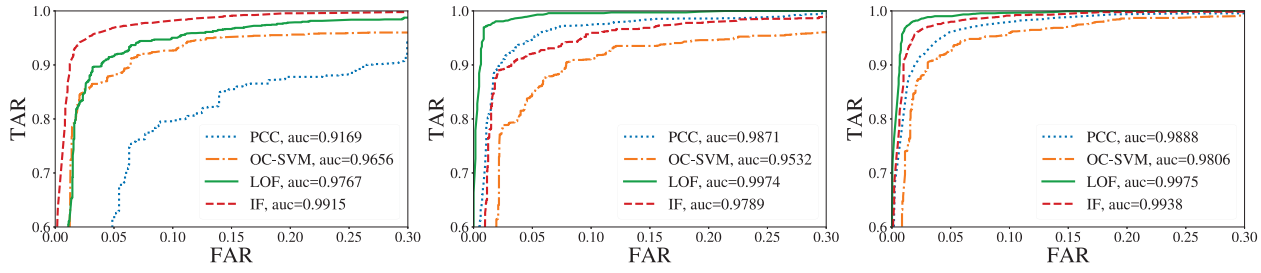
We also evaluated the performance of FINAUTH when using non-overlapping subjects in training the base CNN and evaluating the authentication models. We split these 90 subjects into two groups randomly and evenly. One was used to train the base CNN as the feature extractor, and the other was used to evaluate the performance of authentication models. 5-fold cross-validation was used in the testing phase. We used CNF + LOF and UnF + LOF on the sitting data points in *dataset-1*.

Table 6 shows the BAC, FAR, FRR, and AUC with non-overlapping subjects in training base CNN and testing. The mean BACs under CNF + LOF and UnF + LOF are 95.34% (compared with 97.99% in Table 5) and 95.59% (compared to 98.02%).

7.2.3 Impact of Different Postures

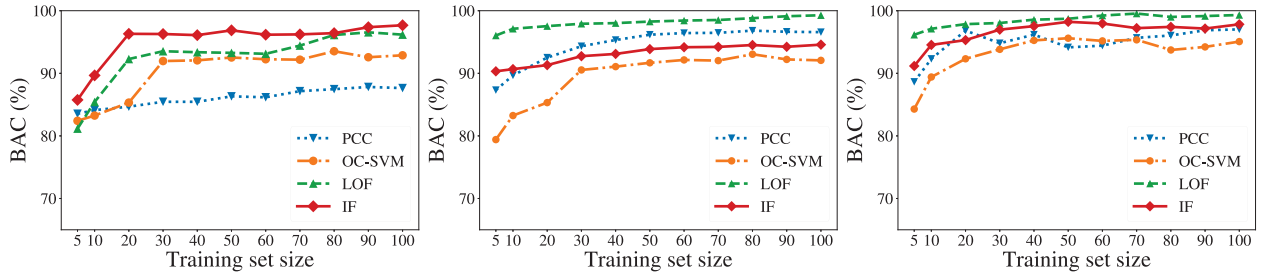
To find out how postures and moving affect the performance of FINAUTH, we used all of the 63,000 data points of *dataset-1*. For each user and each posture, we train a classifier using 30 data points in the training dataset. Specifically, for each participant, the authentication model was trained with regard to five different postures respectively. Next, the model was leveraged to evaluate the performance of different postures.

Figure 7 shows the BAC when using data points collected in different postures to train authentication models (x-axis) and evaluate performance (y-axis). The results indicate that FINAUTH achieves better performance in stationary postures (e.g., sitting, standing, and lying) than moving (e.g., walking and running). Authentication models trained in stationary pos-



(a) Time- and frequency-domain features (b) CNN-based features (c) The union of two feature sets

Figure 5: ROC curves of different feature sets under different one-class classifiers.



(a) Time- and frequency-domain features (b) CNN-based features (c) The union of two feature sets

Figure 6: BAC under different classifiers and different feature sets at varying training set sizes.

tures can be transferred to other stationary postures without downgrading obviously. If we ignore 'running', which is rare in real-life, FINAUTH achieves over 94% BAC when profiling a user with 30 data points collected while sitting.

7.2.4 Impact of Training Dataset Sizes

To investigate the impact of training set sizes, we changed the training set size from 5 to 100 in a step of 5 or 10 to profile the legitimate users. Figure 6 shows the BAC for different classifiers with different training set sizes. As expected, the results show that training with more data achieves a higher BAC. Using CNN-based features or the union of two feature sets, LOF outperforms the other three classifiers. With only 5 training data points and CNN-based features, LOF achieves the BAC of 96.04%, where its FAR is 1.12% and FRR is 6.80%. With 100 training data points, LOF achieves the BAC of 99.28%, where its FAR and FRR are 0.045% and 1.39% respectively.

7.2.5 Consistency Over Time

To find out how consistent users' fingerprint behaviors are over a long period, we used *dataset-2* and the 45,000 sitting data points of *dataset-1*. The training data points were selected from *dataset-1* (the first week of data collection), and test data points were from *dataset-2*.

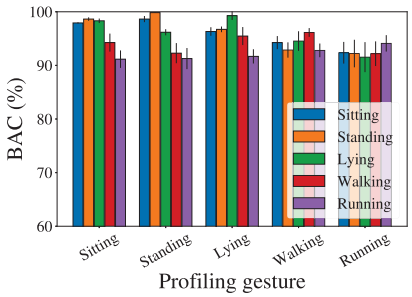
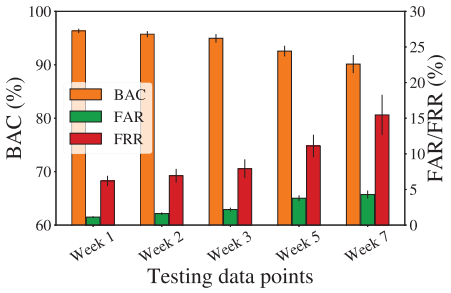
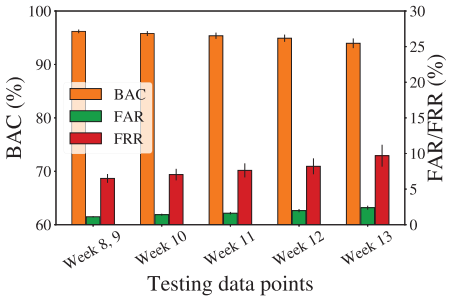


Figure 7: BAC of FINAUTH under different postures.

Figure 8 shows the mean BAC, FAR, and FRR over different weeks with regard to *dataset-2A* and *dataset-2B*. As the results show, behavior variability has an impact on the usability of FINAUTH, but little impact on security. In particular, as shown in Figure 8(a), the BAC decreases from 96.34% to 90.13% under *dataset-2A*, where its FRR increases from 6.20% to 15.46% in 7 weeks. While in Figure 8(b), the BAC decreases from 96.19% to 93.96% under *dataset-2B*, where its FRR increases from 6.50% to 9.69% in 5 weeks. The FAR is almost stable in *dataset-2A&B*. This demonstrates that FINAUTH is resilient against behavioral variability in a short period. In particular, we assume that, in real applications, the problem of behavioral variability can be tackled by



(a) Dataset-2A



(b) Dataset-2B

Figure 8: BAC of FINAUTH evaluated in different weeks using two datasets with different intervals.

Table 7: Mean/standard deviation of BAC (%), FAR (%), and FRR (%), tested on four smartphones (RAM/Snapdragon CPU) with the training set size as 30.

Device	Mean/Std BAC	FAR	FRR
Oneplus3 (6G/ 820)	97.99/0.37	0.87/0.07	3.16/0.74
Oneplus5 (6G/ 835)	98.41/0.56	0.27/0.04	2.91/1.13
XperiaXZ1 (4G/ 835)	96.83/0.52	1.69/0.11	4.65/0.99
VivoX21 (6G/ 660AIE)	98.64/0.18	0.58/0.05	2.13/0.36

retraining the authentication model with newly collected data, namely *model updating mechanism*, which was adapted in Face ID [3].

7.2.6 Impact of Different Devices

To find out how the fingertip-touch data on different devices would affect the robustness of FINAUTH, we evaluated with the 45,000 sitting data points of *dataset-1* and *dataset-3*. As shown in Table 7, the BAC on Oneplus3, Oneplus5, Xperia XZ1, and Vivo X21 are 97.99%, 98.41%, 96.83%, and 98.64%, respectively. There exist variances among different devices in terms of BAC. It achieves the best performance with a BAC of 98.64%, where its FAR and FRR are 0.58% and 2.13% respectively. The worst result on Xperia XZ1 achieves the BAC of 96.86%, where its FAR and FRR is 1.69% and 4.65% respectively.

Table 8: Mean/standard deviation of FAR (%) and prediction score under three types of attacks when tested using models trained with 100 legitimate data points to profile users.

Type	Artificial Replica Attack	Puppet Attack	Mimicry Attack
FAR	0.08/0.06	0.12/0.08	0.25/0.14
Score	−0.29/0.15	−0.62/0.13	−0.37/0.10

7.3 Evaluation of Presentation Attacks

To investigate the defense against presentation attacks, we utilize *dataset-4*. We report the FAR under CNF + LOF at varying training dataset sizes.

Figure 9(a) shows FAR under artificial replica attack using *dataset-4A* with varying training dataset size. The overall BAC is less than 3%. Specifically, the FAR is 2.01% when the model is trained with 10 data points, and it improves to 0.08% using 100 data points.

Figure 9(b) shows the FAR under puppet attack using *dataset-4B* with varying training dataset size. The results indicate that FINAUTH resists against puppet attack with mean FAR below 2%. Specifically, the mean FAR is 1.93% under the model trained with only 5 data points, and it is enhanced to 0.12% under the model trained using 100 data points.

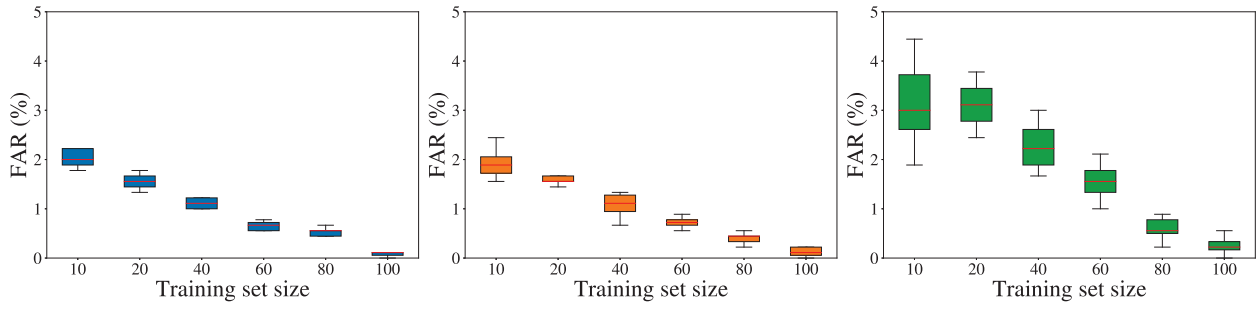
Figure 9(c) shows the FAR under mimicry attack using *dataset-4C*. The results show that it is very difficult for attackers to mimic the fingertip-touch behavior of users. The attack success rate is 3.10% under models trained with 5 data points, and it improves to 0.25% with 100 data points.

As the results show, FINAUTH is effective in defeating all three kinds of presentation attacks. Using more legitimate data points to train the authentication model can strengthen the defense against various attacks. FAR, and prediction scores under authentication models trained using 100 data points are shown in Table 8. In particular, for prediction scores of all attack data points, the distribution and its kernel density evaluated under Gaussian kernel are shown in Figure 10.

7.4 System Performance

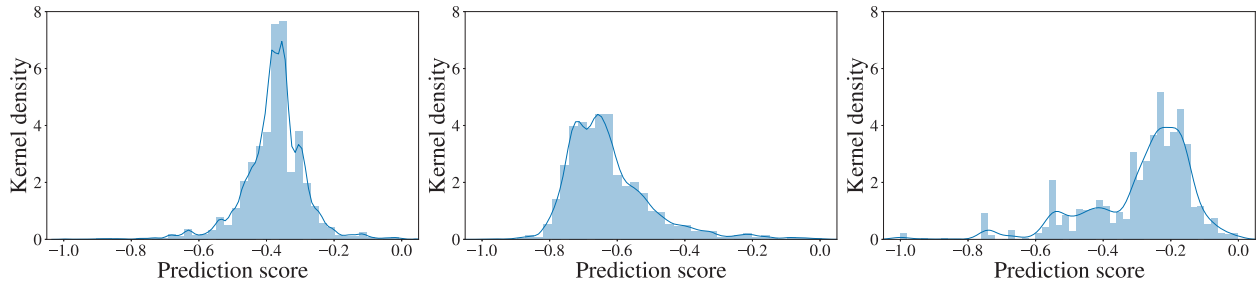
We analyzed the system performance of FINAUTH on Oneplus 3, Redmi Note 4X, Xperia XZ1, and Vivo X21. On each device, we performed authentication with the prototype for 50 times to evaluate the authentication delay, memory usage, and power consumption.

Authentication Delay. The delay is defined as the interval between the time when the authentication system detects the fingerprint authentication event to the time when the system generates the result. It consists of the time for data collection, data processing, and classification. Table 9 shows the delay of four smartphones. The average delay is 713.34 ms, 722.93 ms, 630.72 ms, and 692.15 ms of our method under the four smartphones respectively. Figure 11 shows cumulative dis-



(a) Artificial replica attack (b) Puppet attack (c) Mimicry attack

Figure 9: The FAR, i.e., attack success rate, under the authentication trained with different training set sizes.



(a) Artificial replica attack (b) Puppet attack (c) Mimicry attack

Figure 10: The kernel density of attack data points' prediction score under authentication models trained with 100 data points.

tribution function (CDF) of delay on different smartphones with and without FINAUTH. For 90% attempts, the delay of FINAUTH is less than 742.39 ms, 749.83 ms, 643.26 ms, and 714.54 ms for Oneplus 3, Redmi Note 4X, Xperia XZ1, and Vivo X21, respectively. Overall, FINAUTH only requires an average delay of 689.79 ms. In addition, the delay of our method is lower than existing methods on smartphones, such as PINs, pattern lock, and facial authentication. This implies that FINAUTH can authenticate users timely.

Memory Usage. We used Trepn Profiler [1] and Android Studio Profiler [2] to monitor the memory usage of FINAUTH. Table 9 shows the memory usage of FINAUTH without consideration of graphics on four smartphones. Specifically, the memory usages on four different smartphones are 62.99 MB, 57.82 MB, 48.77 MB, 81.19 MB. The average memory usage is 62.69 MB, which incurs additional 14.92 MB compared with the original fingerprint authentication.

Power Consumption. Trepn Profiler was employed to provide *mW*-level power consumption estimation. Power consumption is measured by subtracting screen power consumption while the screen is on. The average power consumption overhead is 23.13 *mW*, which incurs additional 6.90 *mW* compared with original fingerprint authentication (Table 9).

To sum up, FINAUTH achieves a low authentication delay of 689.79 ms on commercial smartphones. It requires a memory usage of 62.69 MB and power consumption of 23.13 *mW*. Compared with the original fingerprint authentication, it introduces very little overhead and short delay.

7.5 Other Design Considerations

To verify if our feature extraction is effective, we also attempted to construct another CNN-based feature extractor to extract features from denoised sensor data directly without characterizing fingertip-touch behavior. We employed a similar model structure as shown in Table 2 and pre-trained the model with power spectral matrices of denoised sensor data as input to distinguish different users. Then, we implemented end-to-end feature learning by inputting power spectral matrices of denoised sensor data to the model to extract features.

Figure 12(a) shows ROC curves when implementing end-to-end feature learning with CNN. Its best BAC is 61.10% with the training set size as 500. While under our designed fingertip-touch behavior characterizing method (Section 3.2), the BAC reaches 93.11% with only 50 training data points to profiling the legitimate user. As the results show, the step of fingertip-touch behavior characterizing significantly elimi-

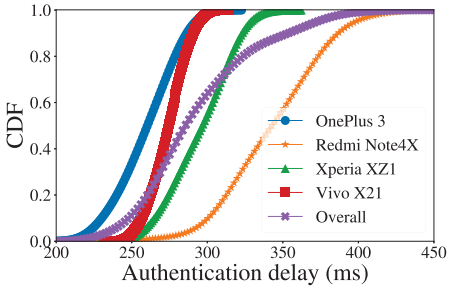
[1] https://developer.samsung.com/game/trepn

[2] https://developer.android.com/studio/profile/cpu-profiler

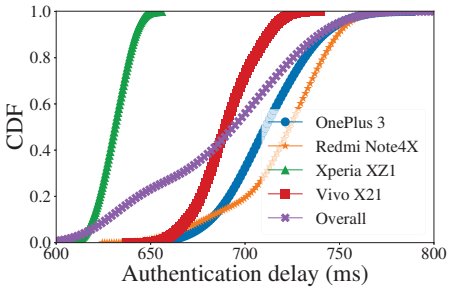
Table 9: Mean authentication delay (*ms*), memory usage (*MB*), and battery power consumption (*mW*) of FINAUTH on four different devices (CPU clock rate, *GHz*).

Device	With FINAUTH			Without FINAUTH		
	Delay	Memory	Power	Delay	Memory	Power
Oneplus 3 (2.15)	713.34	62.99	19.35	257.36	47.82	12.67
Redmi Note 4X (2.0)	722.93	81.19	28.41	342.83	43.56	19.25
Xperia XZ1 (2.45)	630.72	48.77	18.44	293.14	36.75	9.83
Vivo X21 (2.2)	692.15	57.82	26.32	271.16	62.94	23.18

Note that, the authentication delays for PIN, pattern lock, facial authentication are 1.25 [81], 3.14 [81], and 1.48 seconds [6].

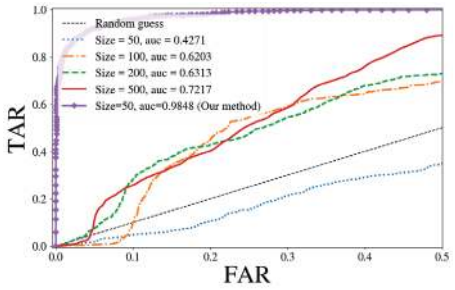


(a) Original fingerprint authentication

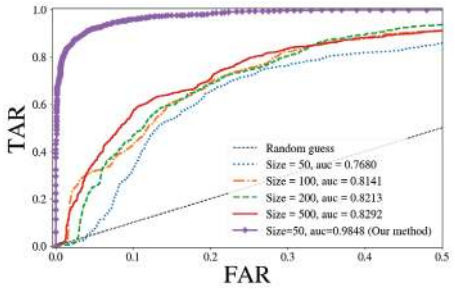


(b) FINAUTH

Figure 11: Authentication delay on different devices



(a)



(b)

Figure 12: (a): ROC curves when using CNN to learn features from denoised sensor data, (b): ROC curves when using *ALOCC* model as one-class classifier.

nates relying on deeper models and a larger number of training data points.

We also evaluated an approach that utilizes a deep learning classification model [23]. We utilized the ALOCC model [60], which was proposed to combine a generative adversary network and an autoencoder to achieve one-class classification. This model combines these two networks to learn the self-distribution of the input in the training phase. It determines whether a data point is an outlier by comparing the distance between its input and output with a threshold. In our experiments, the input of this model is power spectral matrices of accelerations and rotation angles.

Figure 12(b) shows ROC curves under different training set size. The best BAC to recognize different subjects is 76.14%, which is significantly poorer than our previous methods. We suspect the reason is that ALOCC relies on a large number of training data points to learn self-distribution from input one-class data to enable the network robust.

8 Related Work

Fingerprint Presentation Attack Detection. Fingerprint authentication is vulnerable to presentation attacks, which can be carried out easily at a low expense [39]. To enhance its security, various methods have been proposed, including the hardware-based and the software-based. Hardware-based methods acquire life signs to determine the liveness of the input fingerprint, such as blood pressure [42], odor [15], oxygen saturation [59], heartbeat [10], and electrocardiograph [40]. These methods rely on dedicated hardware integrated with fingerprint authentication systems. Software-based methods leverage image processing methods to extract discriminative features from fingerprint images and utilize machine learning techniques to enhance the defense against fingerprint spoofs. Some methods concentrate on the fine-grained characteris-

tics of captured fingerprint images, such as skin perspiration through the pores [54], skin deformation [12], and image quality [33]. Other methods resort to powerful deep learning-based approaches to learn features to distinguish between true and synthetic fingerprints [30, 56]. Existing hardware-based and software-based methods only focus on fingerprint liveness detection. They ignore the intended puppet attack, where the adversary may approach the victim and apply the victim's finger to the fingerprint sensor when the victim is unwilling, e.g., sleeping and fainting. The significantly overlooked problem motivates us to enhance the widely used fingerprint authentication method.

Behavioral Biometrics Authentication. Behavioral biometrics authentication authenticates users based on inherent and unique user's behavior patterns, such as keystroke dynamics [25, 34, 45, 65], signature [64], gesture [28, 65, 68], and gait patterns [49], where behaviors are captured through sensors on mobile devices. However, they are vulnerable to behavior variability in real applications. To handle this issue, behavioral biometric was also designed to fuse with physiological features to provide robust multi-touch authentication [69]. Besides, behavioral characteristics also served as complemented authentication factor to enable traditional knowledge-based authentication schemes (i.e. password/PINs, and pattern locks) resilient against security threats in a highly usable way [21, 41, 48]. Existing behavioral biometrics was designed to authenticate users when performing specific behaviors, such as typing or touching on a screen, writing a signature, or taking a walk. However, it is extremely unnatural to perform such behaviors during fingerprint authentication to enhance its security. Moreover, these methods are necessary to collect behavior data for a relatively long time (e.g., more than 1 second) [65], which will severely undermine the usability if combining these methods with fingerprint authentication. Our proposed system overcomes such challenges. We compare the differences in research question, authentication delay, feature extraction and classification methodologies of these systems in Appendix A.

9 Discussion

9.1 Alternatives to CNN

We chose to use CNN in FINAUTH, because Bai et al. showed that a simple convolutional architecture outperforms canonical recurrent networks across a diverse range of sequence modeling tasks and datasets [14]. Nevertheless, it is worthwhile to evaluate the performance of recurrent neural network (RNN) and long short-term memory (LSTM) networks in future work.

9.2 Limitations

Although we took great efforts to maintain our studies' validity, there are some limitations in our studies and experiments. For example, behavior variability and different postures may incur additional false rejection, and undermine the usability and robustness of our method. Also, FINAUTH requires the user to hand-hold the device. If the device is placed on a desktop stationarily, FINAUTH will fail to work. To solve this issue, FINAUTH can be improved by reminding users to pick the device during authentication if the device is detected not being handheld. It is feasible to detect whether the device is on-hand or on-table using the built-in accelerometer [29]. Also, FINAUTH may falsely reject a legitimate user if she/he uses one hand to register while the other hand to perform authentication. FINAUTH can also be enhanced by reminding users to get the device in the right hand if the device is not.

The datasets we collected were from limited subjects, in which demographic characteristics, e.g., genders, regions, ages, were not perfectly balanced. Fingertip-touch behaviors may differ between males and females, which we did not consider. Older users, who have worked with their hands a lot and even have fingerprints worn away, may also have different fingertip behaviors from the general public. In data collection, even though each subject was told to hold the device in different angles and directions to help collect more distinct data points, they were not required to place the phone down between attempts for their convenience. To enable FINAUTH to work in real applications, it should further be tested to find out other underlying influential factors, which might undermine the performance. As for these older users with their fingerprints worn away, the behavior-based methods might be effective for them. Another concern is user privacy security. Since the sensor data in FINAUTH is related to user behavior, preventing the sensor data from illegal access is of great significance.

9.3 Advanced Attacks

Besides the aforementioned three types of presentation attacks, there also exist the following advanced attacks:

1) *Sensor data injecting attack.* In FINAUTH, raw sensor data are acquired by calling operating system APIs, then processed and input into an authentication model. Due to the imperfection of machine learning models, the adversary can generate adversarial examples to fool and bypass the authentication model by querying models repeatedly [22]. Next, the attacker can inject adversarial data to the sensor dataflow by hijacking OS APIs. In this paper, we did not consider this type of attack.

2) *Adversarial input.* The following adversarial machine learning attacks are possible: i) *model reverse attack* [32]: the attacker aims to infer the training data points used to build the authentication model by querying the model interactively;

ii) *membership inference attack* [66]: the attacker aims to infer whether the constructed data points belong to train set; iii) *model stealing attack* [74]: the attacker aims to use as few queries as possible to compute an approximation model that closely matches the target authentication model; iv) *generating adversarial examples* [22, 53]: the attacker aims to generate adversarial examples to fool and bypass the authentication model by querying the target model interactively.

3) *Robotic attack.* Robotic attack is also a threat of behavioral biometrics [51]. For instance, the attacker can program the robotic arms, such as a Lego robot, to imitate legitimate user's fingertip-touch characteristics [61, 62]. In this attack scenario, even though the attacker has none knowledge of authorized user's fingertip-touch characteristics, he/she could conduct lots of trials. Eventually, it is possible for attackers to find out the correct behavior patterns and drive the robotic arms to perform this specific behavior. Defending against this type of attack is also beyond the scope of our work.

9.4 Future Work

To make FINAUTH more reliable and secure, there are several improvements to pursue in the future: i) *enhancing the CNN-based feature extractor.* In our experiments, the CNN-based feature extractor is pre-trained with limited data points. Collecting data from more users will significantly generalize the feature extractor; ii) *mitigating the impact of postures.* Building the posture detection model using accelerometer data seems a promising method to tackle this problem [80]; iii) *eliminating the impact of behavioral variability.* This problem can be tackled by retraining user authentication models using newly collected data to update users' profiles with time elapsing. Similar approaches have been used in FaceID [3]; iv) *investigating reliability using more data points.* To make FINAUTH more reliable in real-world scenarios, we can continue the evaluation with a more diverse population in the long-term and improve its performance.

10 Conclusion

In this paper, we presented FINAUTH, which complements fingerprint sensors to defend against presentation attacks, especially the puppet attack. FINAUTH models the fingertip-touch characteristics when users apply their fingers to fingerprint sensors. It relies upon common built-in sensors to capture instant behavioral characteristics to authenticate different users. We designed effective methods to characterize the fingertip-touch behaviors and demonstrated that fingertip-touch behavior is distinguishable from person to person during fingerprint authentication. To evaluate the performance of FINAUTH, we compiled datasets from 90 subjects. The evaluation results demonstrate that FINAUTH is robust and can verify legitimate user with high BAC under minimum computation efforts while successfully denying the access requests from unauthorized users with a low false acceptance rate.

Acknowledgments

We thank Kevin Butler and the anonymous reviewers for their comments. This work is supported by the National Natural Science Foundation of China under Grant U1836202, Grant 61772383, Grant 61572380, Grant 61702379, the Joint Foundation of Ministry of Education under Grant 6141A02033341, the Foundation of Science, Technology and Innovation Commission of Shenzhen Municipality under Grant JCYJ20170303170108208, and the Foundation of Collaborative Innovation Center of Geospatial Technology.

References

[1] Alipay adds fingerprint authentication to mobile wallet. https://www.mobilepaymentstoday.com/news/alipay-adds-fingerprint-authentication-to-mobile-wallet/, 2014.

[2] Cybersecurity may be slipping through our fingers. http://www.chinadaily.com.cn/china/2016-12/20/content_27716237.htm, 2016.

[3] Face ID security. https://www.apple.com/business/docs/site/FaceID_Security_Guide.pdf, 2017.

[4] Face ID, touch ID, no ID, PINs and pragmatic security. https://www.troyhunt.com/face-id-touch-id-pins-no-id-and-pragmatic-security/, 2017.

[5] Fingerprints biometric technologies whitepaper. https://www.fingerprints.com/asset/assets/downloads/fingerprints-biometric-technologies-whitepaper-2017-revb.pdf, 2017.

[6] iPhone X face ID slower than touch ID. https://www.tomsguide.com/us/iphone-x-face-id-speed-up,news-26060.html, 2017.

[7] Visa biometrics payments study. https://usa.visa.com/dam/VCOM/global/visa-everywhere/documents/visa-biometrics-payments-study.pdf, 2017.

[8] Report: 920 million fingerprint-enabled smartphones shipped in 2017. https://www.androidheadlines.com/2018/01/report-920-million-fingerprint-enabled-smartphones-shipped-in-2017.html, 2018.

[9] ADKINS, D. L., BOYCHUK, J., REMPLE, M. S., AND KLEIM, J. A. Motor training induces experience-specific patterns of plasticity across motor cortex and spinal cord. *Journal of Applied Physiology* (2006).

[10] ALAJLAN, N., ISLAM, M. S., AND AMMOUR, N. Fusion of fingerprint and heartbeat biometrics using fuzzy adaptive genetic algorithm. In *Proc. of WorldCIS* (2013).

[11] ALLIANCE, F. Fido uaf architectural overview. https://fidoalliance.org/specs/fido-uaf-v1.1-ps-20170202/fido-uaf-overview-v1.1-ps-20170202.html, 2017.

[12] ANTONELLI, A., CAPPELLI, R., MAIO, D., AND MALTONI, D. A new approach to fake finger detection based on skin distortion. In *Proc. of ICB* (2006).

[13] ATHIWARATKUN, B., AND KANG, K. Feature representation in convolutional neural networks. *arXiv preprint arXiv:1507.02313* (2015).

[14] BAI, S., KOLTER, J. Z., AND KOLTUN, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018).

[15] BALDISSERRA, D., FRANCO, A., MAIO, D., AND MALTONI, D. Fake fingerprint detection by odor analysis. In *Proc. of ICB* (2006).

[16] BLANCO, J.-L. A tutorial on SE(3) transformation parameterizations and on-manifold optimization. *University of Malaga, Tech. Rep* (2010).

[17] BONTRAGER, P., ROY, A., TOGELIUS, J., AND MEMON, N. Deepmasterprint: fingerprint spoofing via latent variable evolution. *arXiv preprint arXiv:1705.07386* (2017).

[18] BREUNIG, M. M., KRIEGEL, H.-P., NG, R. T., AND SANDER, J. Lof: identifying density-based local outliers. In *Proc. of SIGMOD* (2000).

[19] BRODERSEN, K. H., ONG, C. S., STEPHAN, K. E., AND BUHMANN, J. M. The balanced accuracy and its posterior distribution. In *Proc. of CVPR* (2010).

[20] BROOKE, J., ET AL. Sus-a quick and dirty usability scale. *Usability evaluation in industry* (1996).

[21] BURIRO, A., CRISPO, B., DEL FRARI, F., AND WRONA, K. Touchstroke: smartphone user authentication based on touch-typing biometrics. In *Proc. of ICIAP* (2015).

[22] CARLINI, N., AND WAGNER, D. Towards evaluating the robustness of neural networks. In *Proc. of S&P* (2017).

[23] CHALAPATHY, R., MENON, A. K., AND CHAWLA, S. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360* (2018).

[24] CHEN, Y., JIN, X., SUN, J., ZHANG, R., AND ZHANG, Y. Powerful: mobile app fingerprinting via power analysis. In *Proc. of INFOCOM* (2017).

[25] CHEN, Y., SUN, J., ZHANG, R., AND ZHANG, Y. Your song your way: rhythm-based two-factor authentication for multi-touch mobile devices. In *Proc. of INFOCOM* (2015).

[26] CHENG, Y., AND LARIN, K. V. In vivo two-and three-dimensional imaging of artificial and real fingerprints with optical coherence tomography. *IEEE Photonics Technology Letters* (2007).

[27] CHUGH, T., CAO, K., AND JAIN, A. K. Fingerprint spoof buster: use of minutiae-centered patches. *IEEE Transactions on Information Forensics and Security* (2018).

[28] CONTI, M., ZACHIA-ZLATEA, I., AND CRISPO, B. Mind how you answer me! transparently authenticating the user of a smartphone when answering or placing a call. In *Proc. of ASIACCS* (2011).

[29] DAS, S., GREEN, L., PEREZ, B., AND MURPHY, M. Detecting user activities using the accelerometer on android smartphones, 2010.

[30] ENGELSMA, J. J., AND JAIN, A. K. Generalizing fingerprint spoof detector: learning a one-class classifier. *arXiv preprint arXiv:1901.03918* (2019).

[31] EVGENY, B., AND DMITRY, S. One-class SVM with privileged information and its application to malware detection. In *Proc. of ICDMW* (2016).

[32] FREDRIKSON, M., JHA, S., AND RISTENPART, T. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proc. of CCS* (2015).

[33] GALBALLY, J., MARCEL, S., AND FIERREZ, J. Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition. *IEEE transactions on image processing* (2013).

[34] GIUFFRIDA, C., MAJDANIK, K., CONTI, M., AND BOS, H. I sensed it was you: authenticating mobile users with sensor-enhanced keystroke dynamics. In *Proc. of DIMVA* (2014).

[35] GU, Q., LI, Z., AND HAN, J. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725* (2012).

[36] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. In *Proc. of CVPR* (2016).

[37] HOU, C., HOU, Y., HUANG, Z., AND LIU, Q. Overlapping one-class SVMs for utterance verification in speech recognition. In *Proc. of TrustCom* (2011).

[38] HUANG, G., LIU, Z., VAN DER MAATEN, L., AND WEINBERGER, K. Q. Densely connected convolutional networks. In *Proc. of CVPR* (2017).

[39] ISO/IEC. *ISO/IEC 30107-1:2016 information technology: biometric presentation attack detection - part 1: frameworkinformation technology*. ISO/IEC, 2016.

[40] KOMEILI, M., ARMANFARD, N., AND HATZINAKOS, D. Liveness detection and automatic template updating using fusion of ECG and fingerprint. *IEEE Transactions on Information Forensics and Security* (2018).

[41] KU, Y., PARK, L. H., SHIN, S., AND KWON, T. Draw it as shown: behavioral pattern lock for mobile user authentication. *IEEE Access* (2019).

[42] LAPSLEY, P. D., LEE, J. A., PARE JR, D. F., AND HOFFMAN, N. Anti-fraud biometric scanner that accurately detects blood flow, 1998.

[43] LAPUT, G., XIAO, R., AND HARRISON, C. Viband: high-fidelity bio-acoustic sensing using commodity smartwatch accelerometers. In *Proc. of UIST* (2016).

[44] LEE, W.-H., AND LEE, R. B. Implicit smartphone user authentication with sensors and contextual machine learning. In *Proc. of DSN* (2017).

[45] LI, L., ZHAO, X., AND XUE, G. Unobservable re-authentication for smartphones. In *Proc. of NDSS* (2013).

[46] LI, Y., HU, H., AND ZHOU, G. Using data augmentation in continuous authentication on smartphones. *IEEE Internet of Things Journal* (2018).

[47] LIU, F. T., TING, K. M., AND ZHOU, Z.-H. Isolation forest. In *Proc. of ICDM* (2008).

[48] LIU, J., WANG, C., CHEN, Y., AND SAXENA, N. Vibwrite: towards finger-input authentication on ubiquitous surfaces via physical vibration. In *Proc. of CCS* (2017).

[49] LU, H., HUANG, J., SAHA, T., AND NACHMAN, L. Unobtrusive gait verification for mobile phones. In *Proc. of ISWC* (2014).

[50] MAAS, A. L., HANNUN, A. Y., AND NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *Proc. of ICML* (2013).

[51] MAHFOUZ, A., MAHMOUD, T. M., AND ELDIN, A. S. A survey on behavioral biometric authentication on smartphones. *Journal of Information Security and Applications* (2017).

[52] MARKLEY, F. L. Attitude error representations for kalman filtering. *Journal of Guidance, Control, and Dynamics* (2003).

[53] MOOSAVI-DEZFOOLI, S.-M., FAWZI, A., AND FROSSARD, P. Deepfool: a simple and accurate method to fool deep neural networks. In *Proc. of CVPR* (2016).

[54] NIKAM, S. B., AND AGARWAL, S. Ridgelet-based fake fingerprint detection. *Neurocomputing* (2009).

[55] NIKAM, S. B., AND AGARWAL, S. Wavelet-based multiresolution analysis of ridges for fingerprint liveness detection. *International Journal of Information and Computer Security* (2009).

[56] NOGUEIRA, R. F., DE ALENCAR LOTUFO, R., AND MACHADO, R. C. Fingerprint liveness detection using convolutional neural networks. *IEEE Transactions on information forensics and security* (2016).

[57] PLOTNIKOV, P. Solution for the motion of a symmetric euler gyroscope for arbitrary initial values of the euler angles using epy kinematic differential poisson equations. *Advances in Theoretical and Applied Mechanics* (2014).

[58] RATTANI, A., AND ROSS, A. Automatic adaptation of fingerprint liveness detector to new spoof materials. In *Proc. of IJCB* (2014).

[59] REDDY, P. V., KUMAR, A., RAHMAN, S., AND MUNDRA, T. S. A new antispoofing approach for biometric devices. *IEEE Transactions Biomedical Circuits and Systems* (2008).

[60] SABOKROU, M., KHALOOEI, M., FATHY, M., AND ADELI, E. Adversarially learned one-class classifier for novelty detection. In *Proc. of CVPR* (2018).

[61] SERWADDA, A., AND PHOHA, V. V. When kids' toys breach mobile phone security. In *Proc. of CCS* (2013).

[62] SERWADDA, A., PHOHA, V. V., WANG, Z., KUMAR, R., AND SHUKLA, D. Toward robotic robbery on the touch screen. *ACM Transactions on Information and System Security* (2016).

[63] SHADMEHR, R. Neural correlates of motor memory consolidation. *Science* (1997).

[64] SHAHZAD, M., LIU, A. X., AND SAMUEL, A. Behavior based human authentication on touch screen devices using gestures and signatures. *IEEE Transactions Mobile Computing* (2017).

[65] SHEN, C., ZHANG, Y., GUAN, X., AND MAXION, R. A. Performance analysis of touch-interaction behavior for active smartphone authentication. *IEEE Transactions on Information Forensics and Security* (2016).

[66] SHOKRI, R., STRONATI, M., SONG, C., AND SHMATIKOV, V. Membership inference attacks against machine learning models. In *Proc. of S&P* (2017).

[67] SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In *Proc. of ICLR* (2015).

[68] SITOVÁ, Z., ŠEDĚNKA, J., YANG, Q., PENG, G., ZHOU, G., GASTI, P., AND BALAGANI, K. S. Hmog: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security* (2015).

[69] SONG, Y., CAI, Z., AND ZHANG, Z.-L. Multi-touch authentication using hand geometry and behavioral information. In *Proc. of S&P* (2017).

[70] SOUSEDIK, C., AND BUSCH, C. Presentation attack detection methods for fingerprint recognition systems: a survey. *IET Biometrics* (2014).

[71] SRINIVAS, S., KEMP, J., AND ALLIANCE, F. FIDO UAF architectural overview.

[72] TEH, P. S., ZHANG, N., TEOH, A. B. J., AND CHEN, K. A survey on touch dynamics authentication in mobile devices. *Computers & Security* (2016).

[73] TEREJANU, G. A., ET AL. Extended Kalman filter tutorial. *University at Buffalo* (2008).

[74] TRAMÈR, F., ZHANG, F., JUELS, A., REITER, M. K., AND RISTENPART, T. Stealing machine learning models via prediction apis. In *Proc. of Usenix Security* (2016).

[75] VERBOOM, J., TIJMONS, S., DE WAGTER, C., REMES, B., BABUSKA, R., AND DE CROON, G. C. Attitude and altitude estimation and control on board a flapping wing micro air vehicle. In *Proc. of ICRA* (2015).

[76] VHADURI, S., AND POELLABAUER, C. Multi-modal biometric-based implicit authentication of wearable device users. *IEEE Transactions on Information Forensics and Security* (2019).

[77] WANG, B., YAO, Y., VISWANATH, B., ZHENG, H., AND ZHAO, B. Y. With great training comes great vulnerability: practical attacks against transfer learning. In *Proc. of Usenix Security* (2018).

[78] YAN, K., JI, Z., AND SHEN, W. Online fault detection methods for chillers combining extended kalman filter and recursive one-class SVM. *Neurocomputing* (2017).

[79] YI, T.-H., LI, H.-N., AND ZHAO, X.-Y. Noise smoothing for structural vibration test signals using an improved wavelet thresholding technique. *Sensors* (2012).

[80] YÜRÜR, Ö., LIU, C. H., AND MORENO, W. Lightweight online unsupervised posture detection by smartphone accelerometer. *IEEE Internet of Things Journal* (2015).

[81] ZEZSCHWITZ, E. V., DUNPHY, P., AND LUCA, A. D. Patterns in the wild:a field study of the usability of pattern and pin-based authentication on mobile devices. In *Proc. of MobileHCI* (2013).

A Detailed Comparison with Other Methods

Besides the brief related work in Section 8, we also provide a detailed comparison between FINAUTH and typical methods published on top venues. The comparison consists of the following aspects, including design goal, attack models, used features, and classification, which is shown in Table 10.

B Sensor Fusion based on EKF

We present the method for sensor data fusion based on EKF:

1. Initialize quaternion as Eq. 6.
2. Define the system state vector $\boldsymbol{x}$ as Eq. 5.
3. Apply normalization to three sensor data.

$$\boldsymbol{z}_a = \frac{[a_x, a_y, a_z]^T}{||\boldsymbol{a}||} \tag{8}$$

$$\boldsymbol{z}_m = \frac{[m_x, m_y, m_z]^T}{||\boldsymbol{m}||} \tag{9}$$

$$\boldsymbol{z}_g = \frac{[g_x, g_y, g_z]^T}{||\boldsymbol{g}||} \tag{10}$$

4. Calculate the projection of the altitude vector along three axes.

$$\boldsymbol{z}_e = \begin{bmatrix} 2(q_1 q_3 - q_0 q_2) \\ 2(q_2 q_3 + q_0 q_1) \\ 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \tag{11}$$

5. Then calculate the estimate error.

$$\boldsymbol{e}_a = \boldsymbol{z}_a - \boldsymbol{z}_e \tag{12}$$

$$\boldsymbol{e}_m = \boldsymbol{z}_m - \boldsymbol{z}_e \tag{13}$$

$$\boldsymbol{e}_g = \boldsymbol{z}_g - \boldsymbol{z}_e \tag{14}$$

6. Define the angle matrix $\boldsymbol{H}$.

$$\boldsymbol{H} = \begin{bmatrix} -2q_2 & 2q_3 & -2q_0 & -2q_1 & 0 & 0 & 0 \\ 2q_1 & 2q_0 & 2q_3 & 2q_2 & 0 & 0 & 0 \\ 0 & -4q_1 & -4q_2 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{15}$$

Table 10: Comparison with other biometric authentication systems on mobile devices.

Paper	Design goal	Features	Classification
[28]	Using movements of devices when answering a phone call to authenticate users	Time-domain features from accelerometer and orientation sensor	DTW-D[1] DTW-S[2]
[45]	Using user's finger sliding gesture patterns to authenticate users	Sliding gesture behavioral features, such as moving distance, duration, etc. from multi-touch screen, accelerometer, orientation, and compass	Binary SVM
[68]	Using hand movement, orientation, and grasp to authenticate users	Time-domain features from accelerometer, orientation sensor and magnetometer	SM[3], SE[4], OC-SVM
[25]	Using the sequence of rhythmic taps/slides to authenticate users	Time-domain features from multi-touch screen	Binary SVM
[69]	Fusing hand geometry and hand gesture behavioral information on screen to authenticate users	Hand-gesture related behavioral features including velocity, pressure, angle, etc. from multi-touch screen	KNN, OCSVM
[48]	Using the physical vibration signal incurred by the finger-input to authenticate users	Spectral point-based features, MFCC-based features from vibration motor and receiver	DTW[5], EMD[6]
[76]	Using fitness data from wearable devices to authenticate users	Time- and frequency-domain features from step counts, heart rate, calorie burn, and metabolic equivalent of task	Binary SVM
FINAUTH	Defending against puppet attack in fingerprint authentication	Time- and frequency-domain features, CNN-based features from accelerations and rotation angles	OC-SVM, PCC, LOF, IF

[1] Dynamic Time Warping Distance. [2] Dynamic Time Warping Similarity. [3] Scaled Manhattan. [4] Scaled Euclidian. [5] Dynamic Time Warping. [6] Earth Moving Distance.

7. Update the covariance matrix of the estimate error $\boldsymbol{P}_e$.

$$\boldsymbol{P}_{e_k} = \boldsymbol{P}_{e_{k-1}} + \boldsymbol{HPH}^T \tag{16}$$

where $\boldsymbol{P}$ is the covariance matrix of the system, k is the timestamp. Both $\boldsymbol{P}_e$ and $\boldsymbol{P}$ are initialized with small values. We initialize $\boldsymbol{P}_e$ and $\boldsymbol{P}$ as $diag(10^{-4}, 10^{-4}, 10^{-4})$ and $diag(10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4})$ respectively, where $diag$ denotes diagonal matrix.

8. Update the gain of EKF with the covariance matrix $\boldsymbol{P}_{e_k}$.

$$\boldsymbol{K} = \boldsymbol{PH}^T \boldsymbol{P}_{e_k}^{-1} \tag{17}$$

9. Update the state vector with the updated Kalman filter's gain.

$$\boldsymbol{q}_k = \boldsymbol{q}_{k-1} + \boldsymbol{K}(\boldsymbol{e}_a + \boldsymbol{e}_m) \tag{18}$$

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + \boldsymbol{K}\boldsymbol{e}_g \tag{19}$$

where k is the timestamp.

10. Update the covariance matrix of the whole system.

$$\boldsymbol{P} = \boldsymbol{P} - \boldsymbol{KHP} \tag{20}$$

11. According to the state vector, acquire the accurate angles as Eq. 7.

C User Acceptance Study

To find out how users perceive FINAUTH, we recruited another 43 subjects, including 12 females and 31 males. These subjects did not participate in the data collections as shown in 3. The subjects were asked to use FINAUTH to perform authentication on their smartphones for one week, and then rate our system. Instead of using system usability scale [20] to measure usability, we focused on convenience, authentication delay, and FRR by asking the following three questions to all subjects:

Q1 Was it easy and convenient to use our system compared to original fingerprint authentication? (-2: Not at all, -1: Little, 0: Neutral, 1: Somewhat, 2: Very.)

Q2 Did you feel obvious delay during authentication compared to the original fingerprint authentication? (-2: Very, -1: Somewhat, 0: Neutral, 1: Little, 2: Not at all.)

Q3 How often were you rejected by the FINAUTH? (-2: Usually, -1: Often, 0: Seldom, 1: Rarely, 2: Never.)

For these questions, we employ 5 levels, from -2 to +2, to represent different levels of user preferences, where +2 corresponds to fully positive and -2 corresponds to fully negative about the system experience. The average ratings of the three questions are all positive at 1.93, 1.44, 1.81, respectively.