Hindawi
Security and Communication Networks
Volume 2022, Article ID 6339407, 13 pages
https://doi.org/10.1155/2022/6339407

Research Article

Who Is Using the Phone? Representation-Learning-Based Continuous Authentication on Smartphones

Huanran Wang,[1] **Hui He**,[1] **Chen Song**,[1] **Hao Tang**,[1] **Yanwei Sun**,[2] **Yanchen Qiao**,[3] **and Weizhe Zhang**[1,3]

[1] *School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China*
[2] *China Information Technology Security Evaluation Center, Beijing 100085, China*
[3] *Department of New Networks, Peng Cheng National Laboratory, Shenzhen, China*

Correspondence should be addressed to Hui He; hehui@hit.edu.cn

Received 26 September 2021; Accepted 16 February 2022; Published 11 April 2022

Academic Editor: Yunchuan Guo

Recently, mobile technology has become closely linked with our daily activities. Smartphones are used for multiple personal tasks involving private information, such as communication, healthcare, and banking. Therefore, there is a high demand for user-friendly authentication methods that prevent unauthorized access to sensitive information. This paper proposes a novel feature representation tactic for continuous authentication named Multiple Channels Biological Graph (MCBG). Unlike conventional techniques, MCBG divides the smartphone usage scenarios into more fine-grained cases, including the operation interval features. To this end, we extract the screen touch and handheld features from multiple built-in sensors without extra user interaction. We conduct experiments on 180 participants (130 adults and 50 minors) and investigate the sufficiency of different sensor combinations required to authenticate identity accurately. Results show that our MCBG-based model achieves 99.38% authentication accuracy within 1.9 seconds. Furthermore, MCBG also represents the intrinsic differences between grown-ups and minors, achieving 96% identification accuracy.

1. Introduction

With the rapid development of computing technology, mobile devices (e.g., smartphones) have become multifunction and portable. Since the outbreak of COVID-19, people have spent at least 4.2 hours a day on mobile phones, an increase of 30% over the previous two years. Smartphones assist people with privacy-related activities, ranging from entertainment and shopping to banking. Therefore, any unauthorized access will cause privacy disclosure risk to smartphone users [1].

Existing authentication mechanisms exploit collecting sensitive data to determine the legitimacy of users at the interactive entry point. For instance, an recorded audio clip [2, 3], facial images taken from a camera [4–9], or fingerprints [10] are most widely used in practice. However, current point-of-entry features lack continuous authentication capability. The effectiveness of using these features depends on external environmental factors. Besides, they are inherently vulnerable to shoulder surfing attack or smudge attack [11]. Studies [12, 13] expose the vulnerability of fingerprint recognition systems to attacks that have been highlighted in the biometrics literature.

On the academic side, studies are still mostly concentrated on sensor-based features. Studies [14, 15] demonstrate that different users have different physiological characteristics (e.g., hand size and finger length), which leads to unique features for each handheld device. Based on this finding, studies [16, 17] utilize multimodal physiological sensors (e.g., accelerometer and gyroscope sensors) to mine distinct patterns of users for continuous authentication. Besides, behavior-based continuous authentication using user gesture features has gained increasing attention in the security community [14, 18, 19]. However, the smartphone computational overhead restricts the authentication efficiency. Besides, these generalized authentication methods fail to cope with the increasing usage scenarios [11]. For instance, the same user shows completely different operating habits in walking and stationary states, resulting in unacceptable false positives for the user experience.

The above discussions about the existing studies demonstrate two real-world challenges. First, unlike conventional point-of-entry techniques, continuous authentication modules are restricted to smartphones' performance limitations. Second, it is challenging to propose a general authentication mechanism in different usage scenarios because different sensor data patterns in different contexts represent different biometrics.

To address these challenges, we propose a continuous authentication method with behavioral biometrics features extracted from the built-in sensors. Our proposed method has the following advantages:

(1) This paper proposes a novel feature representation tactic that can offer an interpretable authenticate result

(2) Our work mines the intrinsic differences between grown-ups and minors, which can protect minors from mishandling or accessing restricted information

(3) Instead of proposing an authentication method for all usage scenarios, our proposed method can be adapted to more fine-grained scenarios

(4) Our proposed implicit technique keeps continuous authenticating in the background to remedy the point-of-entry mechanisms

(5) The proposed method only collects the training features from the built-in sensor to complete the authenticate task without requiring any extra sensitive permissions

To our best knowledge, we are the first to present representation-learning-based authentication based on three feature extraction opportunities: clicking, sliding, and the interval between operations. To reduce the load caused by feature engineering, we present a novel representation of handhold features named Multiple Channels Biological Graph (MCBG). To this end, we subdivide a touch movement into three stages: press down, finger movements, and finger up. We capture the sensor data for each contiguous finger screen operation. In the experimental stage, we analyze the impact of the used number of sensors on the authentication effect. We use convolutional neural networks (CNN) to classify MCBGs and train models in different motion states. Furthermore, the experimental results show that MCBG is effective for identifying minors. Note that we aim to provide a new feature representation idea instead of proposing an image recognition algorithm. We hope that the proposed method can be regarded as a general feature representation, even when there is a breakthrough in image recognition or the performance of smartphones in the future. In summary, our contributions to this work are as follows:

(1) We propose MCBG and design a novel continuous identity authentication method based on representation learning. MCBG is used as input of our designed lightweight model by exploiting the capabilities of convolutional neural networks (CNN).

(2) We collect usage real-world dataset of 180 participants, including sensor records in various scenarios. Furthermore, we release our dataset and feature set to the security community to validate our work and motivate further study.

(3) We evaluate the performance of the proposed method in several experiments. The recognition accuracy of the sliding authentication scenario achieves 95.43%, the recognition accuracy of the password authentication scenario achieves 93.92%, the comprehensive authentication accuracy in daily usage achieves 99.38%, and the accuracy of minor identification achieves 96%. Results show that our lightweight model is computationally demanding and energy-consuming.

The remainder of this paper is organized as follows. Section 2 discusses the relevant studies. Section 3 designs the design details of MCBG. Section 4 discusses the framework of this work. Section 5 evaluates the performance of the proposed approach. Section 6 summarizes the achievements and future work. Note that we release the full dataset and extracted feature set to researchers for reproducibility purposes.

2. Related Literature

Many endeavors have been done on mobile device user identification. The existing mobile user identification technology mainly includes identifying the user and their biological attributes, such as age, gender, and gait. We discuss the related works from two aspects according to our emphasis.

2.1. Entry Point User Authentication. The user entry-point-based authentication mainly includes password mechanism and physiological recognition. In the early 1990s, Kwon and da Vitoria Lobo [7] used facial images to predict the real age. They collected 47 high-resolution images and divided them into three groups: infants, youngsters, and elders. Their method is based on facial geometry and wrinkles analysis. The experimental result reaches 100% accuracy. However, their detection effect is highly dependent on the pixel value. Buyuk and Arslan [9] used multilingual speech data to train the feedforward DNN model. However, the classification effect is affected by the surrounding environment noise. Furthermore, fingerprint-based methods have been getting increasing attention recently. Gnanasivam and Muttan [8] collected 3,570 fingerprint images, divided these images into 5 age groups, and then used discrete wavelet transform and singular value decomposition to extract feature vectors from fingerprint images. Finally, the k-nearest neighbor algorithm (KNN) is used to classify the feature vector. Experimental results show that the accuracy of age prediction is 76.84%.

However, the credentials used for accessing smartphones are vulnerable to social engineering attacks.

In general, current point-of-entry features cannot continuously authenticate the operator's identity. The authentication effectiveness of using these features depends on multiple external environmental factors. Studies [12, 13] summarized the vulnerability of fingerprint recognition systems to attacks that have been highlighted in the biometrics literature.

2.2. Continuous User Authentication. Behavioral recognition techniques extract specific features from observed behavioral patterns of human activities such as handwriting, gestures, and keystroke dynamics. Some behavioral features also change with age or health condition.

Keystroke dynamics technology [20] analyzes a person's typing behavior and extracts keystroke characteristics, such as the press duration and the interval between two keystrokes, as a basis for distinguishing a person or a group. Syed Idrus et al. [21] used keystroke dynamics for age classification. They collected user's keystroke data in the input process and extracted four kinds of features. Experimental results show that when the user enters a specific word, the classification accuracy rate for age reaches 78%. Uzun et al. [22] distinguished underage users through physical keyboards using keystroke dynamics. However, Epp et al. [23] revealed in their research that different emotions of the same person will produce different keystroke behavior patterns. Al Maadeed and Hassaine [24] proposed a reasoning system using handwriting analysis for gender, age, and nationality. They combined the random forest classifier and the spectral regression (SR-KDA) classifier by collecting Arabic and English handwritten samples of 1,017 volunteers. The experimental result showed that the accuracy rate of the age range prediction achieves 60.62%. Although smartphones currently support handwriting screens, the scene of handwriting rarely appears in the daily use of smartphones.

Compared with handwriting, gestures such as sliding, dragging, and tapping are more common in the scenes of daily use of smartphones. Vatavu et al. [25] collected click data from 119 volunteers. They used the offset value between the user's real click position and the target position as features to classify the testers. Davarci et al. [26] further added 16 features extracted from the motion sensor through signal processing technology and adopted the k-nearest neighbor algorithm in the classification step. Experimental results show that the accuracy of this method can reach 92.5% based on collecting 30 consecutive clicks of data. However, the critical limitation of these methods is that the smartphone computational overhead restricts the authentication efficiency. As a remedy, we propose a continuous authentication system with behavioral biometrics features on smartphones. We disassemble a finger touch process into three stages to reduce system load and improve detection efficiency: press down, finger movements, and finger up. We capture the sensor data of each contiguous finger screen operation. To improve the detection accuracy, we present MCBG, a novel representation of the user's operating features. With MCBG, we identify the user's identity information during password entering and sliding to unlock. Experimental results show that our method outperforms other related solutions in efficiency and accuracy.

3. Design and Implementation of MCBG

Continuous authentication technologies aim to examine the underlying difference between the device owner and guests regarding their gesture behaviors when operating smartphones. To this end, we utilize sensor-based features in smartphone usage. Most of the touch traces of a user tend to follow a similar pattern, which is different from other users [11, 18].

To reduce the load caused by feature engineering, we propose MCBG as a novel feature representation tactic, which includes a key-in feature graph, interval feature graph, and sliding feature graph according to actual application scenarios. We divide a complete screen touch process into three steps: down, move, and up. During a complete screen touch, we collect the information gathered by the built-in sensors and store it in a time-stamped file. The variable definitions that existed in this section are shown in Table 1.

3.1. Key-In Feature Graph. The key-in feature graph (KFG) applies to all short-term screen click actions such as typing. In this case, the user's finger trajectory is scattered on the screen. so features such as slide speed and slide distance are not considered. In the actual use of smartphones, there is a deviation between the target position and the actual position of a finger touch, implying multiple user physical characteristics (such as gender, age, and health status). For example, when an adult user clicks a button, the distribution of click traces is more concentrated than a minor intuitively. Therefore, the coordinates of a user's multiple clicks on the same target are distributed around the click target (tx, ty) and are farther away from other targets.

Therefore, we determine the click targets and their corresponding actual positions by the CFDP algorithm [27]. We then confirm the target position (tx, ty) and calculate their offset. Figure 1 shows a partial cluster result of an underage participant.

We randomly selected ten users' click records (the dataset is discussed in Section 5) by clicking on the "a" and "p" on the virtual keyboard, of which the distribution is shown in Figure 2. We connect each user's points to display their touch habits for clarity. According to our observation, the number of records generated by typing or sliding operations does not exceed 300. To improve the data storage efficiency, we set the size of the feature graph to $300 * 40$ pixels. In the pixels of each row of the feature graph, we use 40 bits to store each record, where the first bit represents record type, and the last 39 bits are sensor records represented by binary. We then approximate the data to 8 decimal places.

In this way, we create a $300 * 40$ all-zero matrix in the generating process of a key-in feature graph with the data in Table 2 and finally save the matrix as a grayscale image. Note that energy represents the sum of squares of the sensor on

Table 1: Variable definition of MCBG.

Variable	Definition
x_i, y_i	The ith actual touch's position
tx_i, ty_i	The ith target touch's position
t_i	The ith touch's timestamp
p_i	The ith touch's pressure
size_i	The ith touch's contact area
touchmajor_i	The ith touch's long axis of the touch area
touchminor_i	The ith touch's short axis of the touch area
acc, gyr, ori	Returned by the acceleration, gyroscope, and orientation sensor, including 3 floating numbers of 3 axes
down, move, up	Respectively denote pressing, moving, and lifting actions in the process of touching screens
s_i	The ith touch record
$\widehat{s}$	Average of s
n	Total number of s
std^s	Standard deviation of s
Skewness	$(\sum_1^n (s_t - \widehat{s})^s / n * \text{std}^s)$

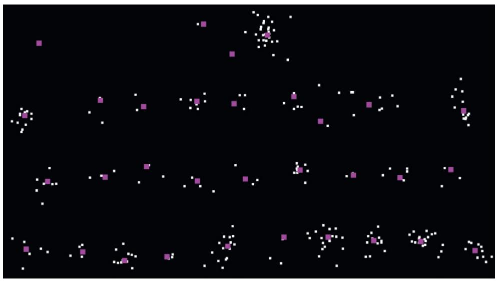
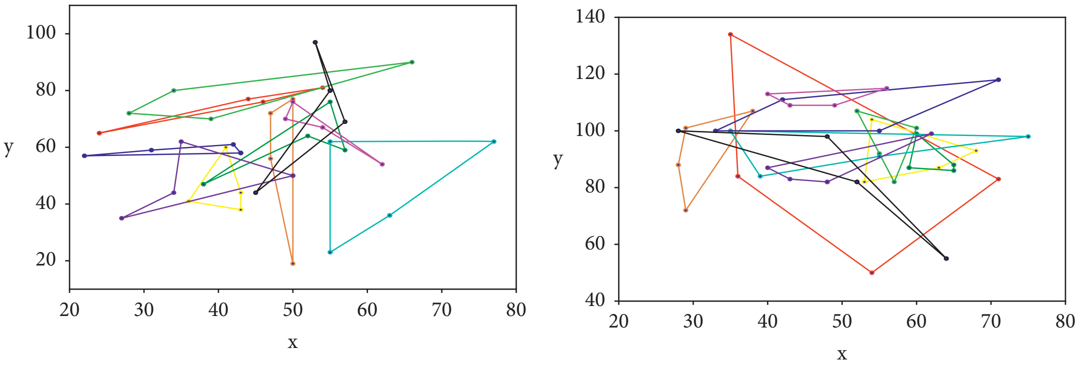


Figure 1: Cluster result of click positions. The white dots represent the actual clicks, and the purple dots denote the click targets determined by the algorithm.



(a)

(b)

Figure 2: Button position distribution of 10 user data. Take the lower-left corner of the screen as the coordinates (0, 0). (a) Position distribution of "a" and (b) position distribution of "p."

three axes. In this way, we constructed a key-in feature graph. Figure 3 shows the result of five random users clicking the button "p." The white part corresponds to binary 0, and the black part is 1. The picture from top to bottom is down, move, and up data, calculated data, and sensor data. It can be seen intuitively from the figure that the operating habits of different users are entirely different.

3.2. Interval Feature Graph. The interval feature graph (IFG) is used to record the handheld features of the user during continuous input or other operations. When the user continuously operates the screen, the time features of two consecutive touches can be represented by Figure 4.

For the analysis of two adjacent touches, we selected time and space-related features to construct an interval feature graph, which is shown in Table 3. Because the feature values in Table 3 are generally small, to save the calculation cost, we write these features into a 31 ∗ 24 matrix in turn, where the first bit of each row of pixels represents record type, and the remaining 23 bits are used to store data.

Therefore, the time threshold is set to 1,300 milliseconds, and only operations within the time range will be generated interval graphs. Figure 5 shows the interval between

Table 2: Features used in key-in feature graph.

Type	Variable	Label
Raw data	p, size, touchmajor, touchminor	Down, move, up
	x, y	
	acc_i, gyr_i, ori_i	acc, gyr, ori
Calculated data	$x_{i+1}-x_i$, $y_{i+1}-y_i$, $t_{i+1}-t_i$	Down, move, up
	Skewness $(t_{i+1}-t_i)$	
	max (p_i), max $(\mathrm{touchmajor}_i)$	
	max (size_i), max $(\mathrm{touchminor}_i)$	
	min (p_i), min $(\mathrm{touchmajor}_i)$	
	min (size_i), min $(\mathrm{touchminor}_i)$	
	Skewness (p_i), Skewness $(\mathrm{touchmajor}_i)$	
	Skewness (size_i), Skewness $(\mathrm{touchminor}_i)$	
	$tx_{i+1}-tx_i$, $ty_{i+1}-ty_i$	
	$\mathrm{acc}_{i+1}-\mathrm{acc}_i$, $\mathrm{gyr}_{i+1}-\mathrm{gyr}_i$, $\mathrm{ori}_{i+1}-\mathrm{ori}_i$	acc, gyr, ori
	Skewness $(\mathrm{acc}_i, \mathrm{gyr}_i, \mathrm{ori}_i)$	
	energy $(\mathrm{acc}_i, \mathrm{gyr}_i, \mathrm{ori}_i)$	

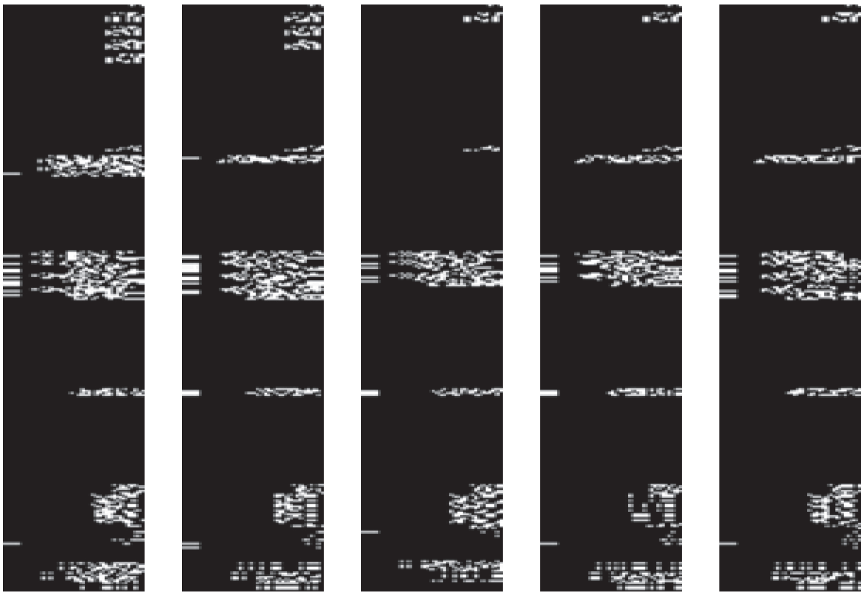


Figure 3: The KFG of five users clicking "p" (partial).

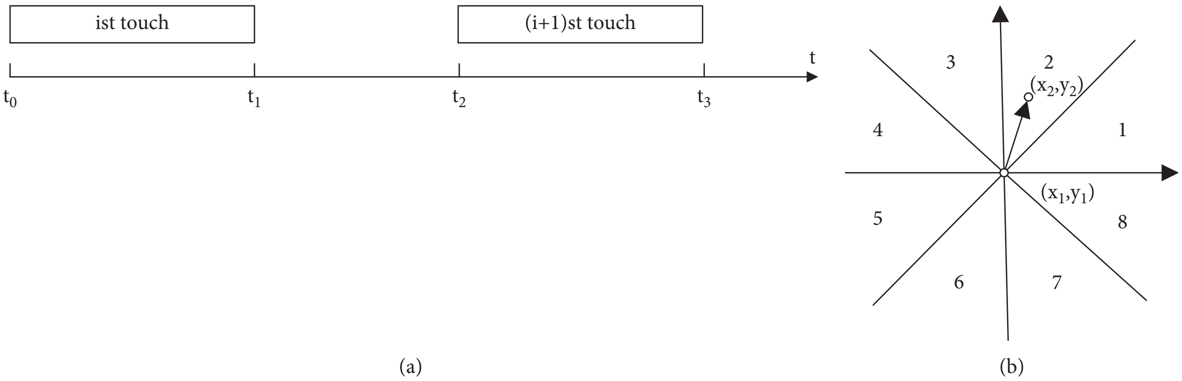


Figure 4: Schematic graph of two adjacent touch events. (a) Time relationship between two adjacent touch events; (b) spatial relationship between two adjacent touch events.

Table 3: Features used in interval feature graph.

Type	Variable	Label
Raw data	x_1, y_1, x_2, y_2	Down, move, up
Calculated data	d_1, d_2, d_3	
	$(d_2/d_1), (d_3/d_2), (d_3/d_1)$	
	$\arctan(d_3/d_1)$	
	$(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}/t_2 - t_0)$	
	$(\mathrm{acc}_2 - \mathrm{acc}_1), (\mathrm{gyr}_2 - \mathrm{gyr}_1), (\mathrm{ori}_2 - \mathrm{ori}_1)$	acc, gyr, ori

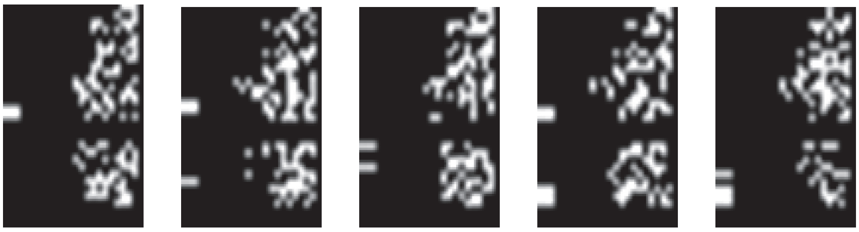


Figure 5: Interval feature graphs constructed during continuously input passwords by five random users (partial).

inputting "i" and "p" when five random users continue to input passwords.

3.3. Sliding Feature Graph. The goal of the sliding feature graph (SFG) is designed to characterize the user's single sliding feature. The main scene is the nine-square grid sliding to unlock the smartphones. After the user slides the unlock pattern once, no matter whether the screen is successfully unlocked or not, the following operation will be irrelevant to this operation. Therefore, there is no need to generate an IFG corresponding to the sliding gesture.

SFG can be regarded as an extended version of KFG. Therefore, the SFG adopts the features and structure of the key-in graph. The difference is that the size of the sliding feature graph is set to 1700 $*$ 40 for all gesture data. Figure 6 shows an SFG generated when a user slides to unlock the screen according to the pattern "S."

Besides, we add the LDP size feature at the end of the sliding feature graph. As shown in Figure 7, LDP refers to the point on the sliding track with the shortest straight line distance from the starting points and endpoints. LDP size refers to the touch area of LDP.

Figure 6: Sliding feature graph constructed during the process of a user sliding the screen to unlock (partial).

4. Design and Implementation of Continuous Authentication

In this section, we present the implementation of our framework. This system has a C/S architecture consisting of two main parts: the Android client (installed on Android devices) that extracts the MCBG and the PC server that conducts data processing to train deep-learning-based models. The server completes the main computational task to minimize the impact on system performance.

Our framework is divided into four modules: data extraction module, information processing module, MCBG generation module, and identity authentication module. Note that the data extraction module is implemented in the Android client, whereas the remaining modules belong to the PC server. Figure 8 shows the workflow of the proposed method.

4.1. Data Collection Module. The data that generates MCBG is captured by the built-in sensors provided by the Android system. The Android system provides real-time dump information of device events to corresponding sensor services or applications by event pool. Most Android devices have built-in sensors that can provide highly accurate raw data. These sensors are used to locate or measure the users' movements on a three-dimensional coordinate axis or to sense changes in the external environment. Eventually, each event information is broadcast in data packets to applications or services with reading permissions.

The Android platform currently supports three types of sensors: dynamic, environmental, and position sensors. Besides, user gesture features are directly obtained from the hardware event pool through the *getevent-lt* command provided by the Android system [28]. According to our statistics, an average of 45 data packets are generated for each operation in users' daily use. We obtain the touch coordinates, contact area, timestamp, and other data. Since most Android devices do not directly provide hardware support for screen pressure, we use the method provided in kernel [29] to obtain touch pressure. Android fits the area where the finger touches the screen into an ellipse. We get *ABS_MT_TOUCH_MAJOR* and *ABS_MT_TOUCH_MINOR* by *getevent* command, which, respectively, represent the long axis of the contact surface and the long axis of the fingertip. As the pressing force increases, the fingers deform, and *ABS_MT_WIDTH_MAJOR* increases. Therefore, formula (1) can be used to estimate the magnitude of the pressing force p.

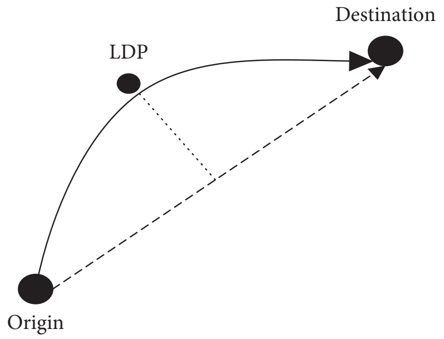


Figure 7: LDP schematic diagram.

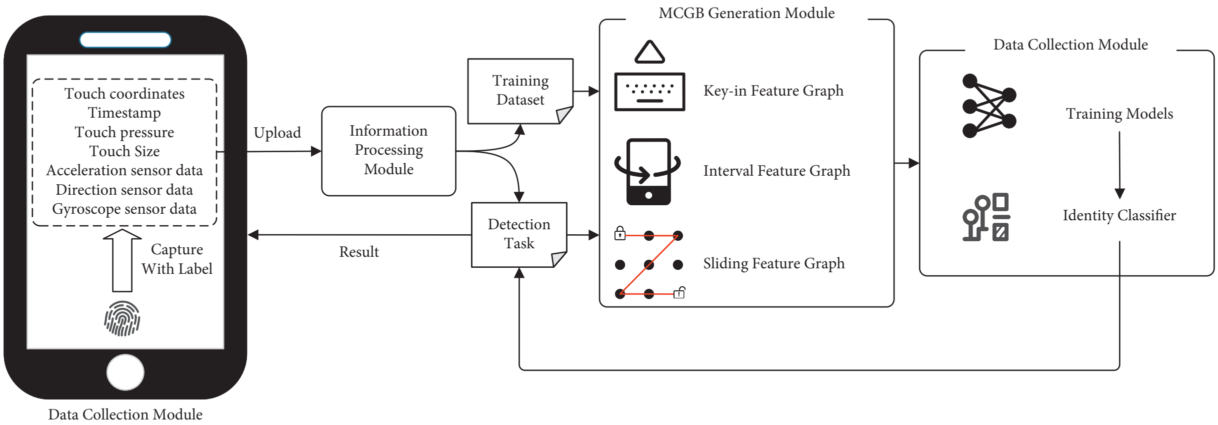


Figure 8: Continuous authentication flowchart.

$$p = \frac{\text{ABS\_MT\_WIDTH\_MINOR}}{\text{ABS\_MT\_WIDTH\_}MAJOR}. \tag{1}$$

Next, we disassemble a finger touch process into three stages: down, move, and up. Our primary purpose is to identify the user's identity information based on the sensor data during the touch process. Therefore, we only capture the data collected from the downstage to the upstage and the sensor data during the two-finger touch during the continuous operations. This strategy can significantly reduce system load and improve detection efficiency. Figure 9 shows the data captured by a user attempt to unlock the screen, which includes one down record, five move records, one lift record, and nine sensor change records. Finally, the data collection module sends the collected information to the server and waits for the detection result.

4.2. Information Processing Module. The collected sensor data requires a preprocessing stage for inevitable noise handling and temporal alignment for sequence generation. The information processing module is designed to relieve the computational burden of smartphones. The data processing tasks are migrated from the mobile client to the server. This module first receives the original data from the mobile client and removes the incomplete data, such as sliding gesture records that do not start with "down" or end with "up." Then the module divides the data into training and test dataset according to task type.

4.3. MCBG Generation Module. The main task of the feature map generation module is to further convert the standardized data into binary and write it into the image file. This module mainly solves three real-world problems:

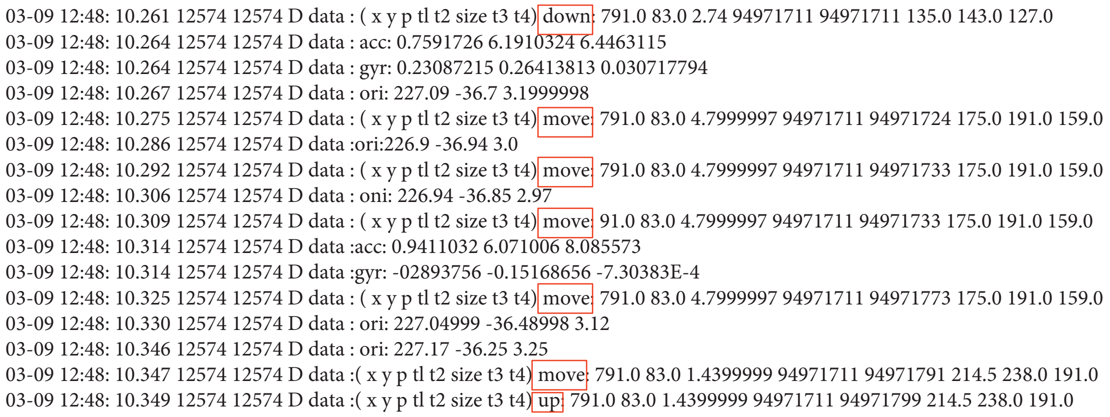
(1) The user's identity authentication when entering passwords or typing text, corresponding to the key-in feature graph (discussed in Section 3.1)

(2) The user's identity authentication during the continuous screen operation, corresponding to the interval feature graph (discussed in Section 3.2)

(3) The user's identity authentication in the process of sliding unlocks, corresponding to the sliding feature graph (discussed in Section 3.3)

The generation strategies of the three feature maps are discussed in detail in Section 3. Finally, these maps are handled by the identity recognition module.

4.4. Identity Recognition Module. Typically, the assigned task is to identify potential associations between user identity and their biometrics generated during use. The convolutional neural network is the best candidate algorithm for this task



```
03-09 12:48: 10.261 12574 12574 D data : ( x y p tl t2 size t3 t4) down: 791.0 83.0 2.74 94971711 94971711 135.0 143.0 127.0
03-09 12:48: 10.264 12574 12574 D data : acc: 0.7591726 6.1910324 6.4463115
03-09 12:48: 10.264 12574 12574 D data : gyr: 0.23087215 0.26413813 0.030717794
03-09 12:48: 10.267 12574 12574 D data : ori: 227.09 -36.7 3.1999998
03-09 12:48: 10.275 12574 12574 D data : ( x y p tl t2 size t3 t4) move: 791.0 83.0 4.7999997 94971711 94971724 175.0 191.0 159.0
03-09 12:48: 10.286 12574 12574 D data :ori:226.9 -36.94 3.0
03-09 12:48: 10.292 12574 12574 D data : ( x y p tl t2 size t3 t4) move: 791.0 83.0 4.7999997 94971711 94971733 175.0 191.0 159.0
03-09 12:48: 10.306 12574 12574 D data : oni: 226.94 -36.85 2.97
03-09 12:48: 10.309 12574 12574 D data : ( x y p tl t2 size t3 t4) move: 91.0 83.0 4.7999997 94971711 94971733 175.0 191.0 159.0
03-09 12:48: 10.314 12574 12574 D data :acc: 0.9411032 6.071006 8.085573
03-09 12:48: 10.314 12574 12574 D data :gyr: -02893756 -0.15168656 -7.30383E-4
03-09 12:48: 10.325 12574 12574 D data : ( x y p tl t2 size t3 t4) move: 791.0 83.0 4.7999997 94971711 94971773 175.0 191.0 159.0
03-09 12:48: 10.330 12574 12574 D data : ori: 227.04999 -36.48998 3.12
03-09 12:48: 10.346 12574 12574 D data : ori: 227.17 -36.25 3.25
03-09 12:48: 10.347 12574 12574 D data :( x y p tl t2 size t3 t4) move: 791.0 83.0 1.4399999 94971711 94971791 214.5 238.0 191.0
03-09 12:48: 10.349 12574 12574 D data :( x y p tl t2 size t3 t4) up: 791.0 83.0 1.4399999 94971711 94971799 214.5 238.0 191.0
```

Figure 9: Data collected from an attempt to unlock the screen.

to capture these potential behavioral patterns from a sequence of sensor data [30].

The data training module continuously receives MCBGs of different training tasks and stores them in files. After training, we update the existing classifiers. We use MCBG as input for the corresponding classifier, get the recognition result, and feed it to the mobile terminal without additional computing tasks.

4.5. Time Complexity. The method proposed in this paper is mainly time-consuming in the MCBG extraction stage and recognition stage. Because the construction of each feature map is generated row by row, the generation time is only related to the constant matrix size. The matrix sizes of KFG, IFG, and SFG are $300 * 40$, $31 * 24$, and $1700 * 40$, respectively. The training algorithm mainly determines the time complexity of the recognition phase. We adopt CNN as the recognition algorithm (discussed in Section 5).

Theoretically, the time complexity of CNN is defined as $O(\sum_{l=1}^{D} M_l^2 \cdot K_l^2 \cdot C_{l-1} \cdot C_l)$, where M is the size of the feature map, K represents the size of the convolution kernel, and C_{in} and C_{out} represent the number of input/output channels. Besides, M is determined by four parameters: input size X, convolution kernel size K, Padding, and Stride. Therefore, the overall time complexity of CNN is the accumulation of the time complexity of all convolutional layers. When the hyperparameters and the input matrix size are fixed, the overall complexity is constant. We evaluate the detection efficiency and resource consumption in Section 5.

5. Evaluation Results

In this section, we first introduce the experimental setup of evaluation and address four research questions:

RQ 1: How does the sensor selection affect the authentication effects?

RQ 2: Which machine-learning-based algorithm is appropriate for continuous authentication?

RQ 3: How much resource consumption does the proposed method impose?

RQ 4: Can MCBG be used to identify the minors?

RQ 5: How does the user's movement affect MCBG?

5.1. Experimental Setup and Dataset. All experiments are run on an Intel(R) Xeon(R) CPU X5650 @ 2.67 GHz with four clusters of 128 GB memory. Furthermore, the experiment smartphones are loaded with Kirin 980 processors with 8 GB of memory. The battery's capacity is 3650 mAh. The screens' resolution is $2340 * 1080$ pixels. The metrics in this section are Recall, TNR, and accuracy, representing the accuracy rate in the positive examples, the accuracy rate in the negative examples, and the accuracy rate in the entire test set.

The experimental data is gained from 180 participants (130 adults and 50 minors). Figure 10 shows the age distribution of the 50 underage participants. The participants were taken in turn as legitimate users for multiple experiments. In each set of experiments, one participant was regarded as a legitimate user, and the others were regarded as illegal holders.

Besides, we decided to collect data ourselves because our feature extraction method is different from other methods. The current public dataset cannot support our construction of the MCBG graph. Therefore, we collect usage data from 180 participants, including operating data records for various scenarios. To simulate the daily scenarios, we arrange these participants to perform the following three steps on the given smartphones: (1) use the system's default virtual keyboard, and repeatedly type "*The quick brown fox jumps over the lazy dog*," (2) input 6-digit presupposed passwords in a designated input box, and (3) unlock the screen with the "N-," "Z-," "X-," "L-," "S-," and "V"-shaped sliding gestures. The MCBGs collected in the three operations from the same participant are labeled as a set of daily operations. To avoid a large number of repeated operations of image data authenticity in a short time, each participant completed data collection within seven days. In this way, we obtained the 19,179 KFGs, 17,461 IFGs, and 17,978 SFGs.

5.2. Evaluation of Approach Capability

Experiment 1. (RQ 1): While smartphone sensors enhance the user experience, they also accelerate the consumption of

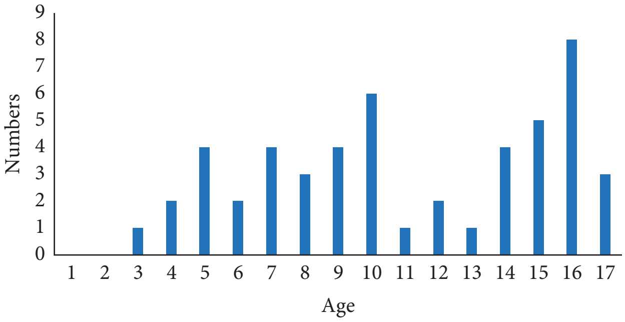


Figure 10: Age distribution of underage participants.

battery power. This experiment investigates the effects of including different sensory data on authentication performance.

Following the research of Cao [31], gyroscope sensor, acceleration sensor, and direction sensor could achieve the best authentication effect, which also fits the scenario discussed in our work. In this set of experiments, we separately evaluate the following eight sensor data selection cases: (1) no sensor; (2) only acceleration sensor; (3) only direction sensor; (4) only gyroscope sensor; (5) both acceleration and direction sensor; (6) both acceleration and gyroscope sensor; (7) both direction and gyroscope sensor; (8) all the three sensors. Note that this set of experiments is trained on the raw dataset after the corresponding rows where the sensor is not selected are removed. We train a model for each participant to distinguish himself from others in all eight cases. In the test stage, we treat each participant as the device's owner in turn and the data generated by others as the illegal data of the device. In this way, we verify that the model can correctly distinguish device owners from intruders and calculate the final FRR and FAR.

Because CNN can directly use graphical data as input, it does not require manual image preprocessing, additional feature extraction, and other complex operations. Image processing has reached almost human level with its unique fine-grained feature extraction method. Therefore, this experiment uses CNN for verification by default. Note that, in Experiment 2, we further confirmed the performance of CNN on our dataset. Figure 11 shows the comparison of the eight cases. Results show surprisingly low FAR and FRR when using data of three sensors, with a slight increase in cases 3, 5, and 7.

Experiment 2. (RQ 2): In this set of experiments, we compare the performance of several machine learning algorithms on our collected dataset. To evaluate the feature representation of MCBG, we experiment with each kind of MCBG as an independent graph without considering the correlation. Besides, we reprocess the features discussed in Section 3 into vectors as input to adapt the representation of other machine learning algorithms. We use the grid search method [32] to select the optimal value of each parameter for each model. During the test stage, to evaluate the performance of each MCBG type, we test the collected KFGs, IFGs, and SFGs separately after participants operate the default three sets of operations. Finally, each model gets an optimal value combination of parameters, and the comparison results of different algorithms are presented in Table 4.

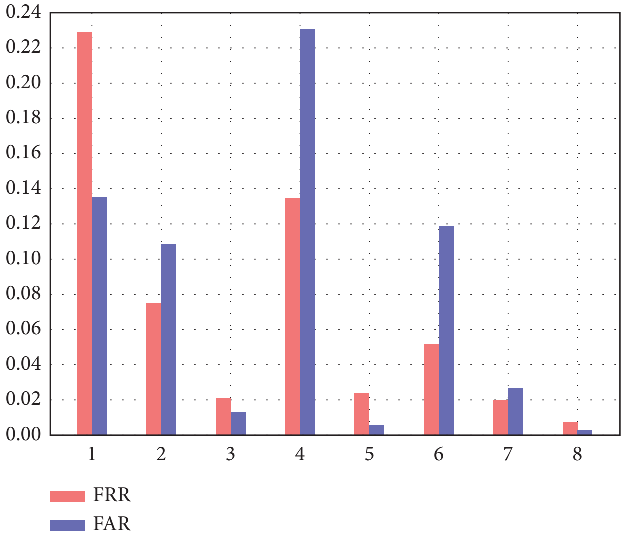


Figure 11: Comparison of eight cases.

The experimental results show that our proposed MCBG is better than traditional feature representation because the graphic features combined with the local connectivity mechanism of CNN can mine potential relevance among multiple actions. CNN can effectively extract features in images. It is more suitable for processing image classification problems than other machine learning algorithms. Therefore, we select CNN as the default classifier. Since the KFGs are generated by password typing operations, the experimental result shows that the accuracy rate in the password typing scenario achieves 93.92%. Similarly, the accuracy rate in the sliding unlock scene achieves 95.43%.

We finally test the comprehensive effect of MCBG in daily authentication scenarios. We add KFG, IFG, and SFG to the test set simultaneously and collect data generated during user operations in the background continuously. We perform model classification every 0.5 seconds. The final authentication accuracy reaches 99.38%. Table 5 shows the comparison between related research and our method.

Experiment 3. (RQ 3): Considering the user experience in real scenarios, the feature extraction program is deployed on the smartphone client, and the feature information is sent to the server to complete the training process. The feature extraction service works as a background service to reduce battery consumption and only begins when the CPU occupancy rate is below 80%. After training, the classification models are stored on the remote server. The features are sent to the server to update these models every 2 hours. Note that the above rules are just compromised strategies with mobile limited device resources. In theory, a higher model update frequency can result in higher model accuracy.

To evaluate the resource consumption of our proposed mechanism, we experimented with comparing the resource consumption by three smartphones with the same initial

Table 4: Comparison of authentication accuracy of machine learning algorithms with single MCBG.

Algorithm	KFG (%)	IFG (%)	SFG (%)
Support vector machine	84.72	66.32	46.99
Logistic regression	85.69	66.09	95.22
Naïve Bayes	85.51	64.22	62.37
C4.5 decision tree	89.51	63.30	86.69
Gradient boosting decision tree	92.70	69.29	93.97
Convolutional neural networks	93.92	92.95	95.43

Table 5: Comparison with related research using the description reported in their papers.

Study	Dataset	Sensor	Algorithm	Accuracy (%)	Scene
Ehatisham [33]	10	acc, gyr, mag	SVM	97.95	Gait
Amini [34]	47	acc, gyr	LSTM	96.70	20 s window
Chao [31]	102	acc, gyr, ori, mag	HMM	94	200 touch actions
Anusas [35]	25	acc, touch	RF	97.90	Input
Abuhamad [11]	84	acc, gyr, mag	LSTM	98.00	Daily use
Jain [36]	104	touch, acc	Hausdorff distance	97.95	Gesture
Our work	180	acc, gyr, ori, touch	CNN	99.38	Daily usage

Table 6: Resource consumption comparison results of different experiment devices.

Device	CPU usage (%)	Memory (%)	Battery consumption (%)	Flow consumption (MB)
MCBG client	51	34	24	57
Regular client	45	27	21	13
Idle client	23	12.50	11	14

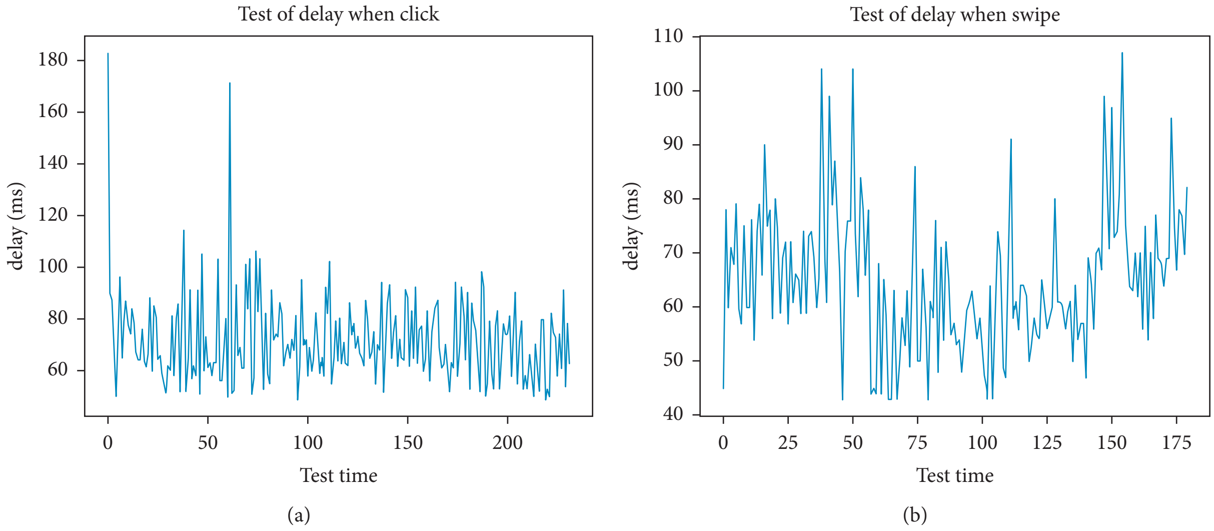


Figure 12: Clicking operation delay (a). Paddling operation delay (b).

configurations, which are, respectively, labeled as MCBG client, regular client, and idle client. Then, the program proposed in this paper is deployed on the MCBG client. To ensure that the usage of MCBG client and regular client is consistent, we test each device to slide and unlock the screen 1, 000 times and record the average memory and CPU rate each time and the total additional traffic and the total battery consumption ratio. The resource consumption comparison results are shown in Table 6.

Due to space limitations, Figure 12 only shows the delay of each tap and swipe gesture when the three sensors are collected. The average data acquisition delay time is 40–70 milliseconds, close to the results of the other seven cases. The average authentication time of our method is 1.9 seconds. The experimental results prove that the method proposed in this paper brings a minute amount of additional flow, power, and delay consumption.

Experiment 4. (RQ 4): This set of experiments aims to evaluate the effectiveness of the proposed method for detecting minors. We collected 4,800 KFGs, 3,500 IFGs, and 4,980 SFGs of underage participants in the data collection

Table 7: Comparison with previous studies using accuracy reported in their paper of minor identification.

Gestures	Our work (%)	Nguyen et al. [37] (%)	Li et al. [38] (%)
Slide	97	96	84
Click	99	89	91

Table 8: Performance evaluation of MCBG under moving status data.

Dataset	IFG accuracy (%)	SFG accuracy (%)	Total MCBG accuracy
Moving status dataset	92.72	81.63	97.05
Total dataset	88.54	85.72	96.33%

step. In this set of experiments, we randomly selected 4,500 KFGs, 3,800 IFGs, and 5,000 SFGs collected from adults for uniform label distribution. We retrained the CNN model to classify the adult and minor labels qualitatively. Table 7 shows the comparative experimental results for sliding and clicking gestures of minor identification.

The experimental results show that the accuracy of the click gesture is close to the slide gesture. Furthermore, the detection accuracy of minors is higher than the authentication accuracy of Experiment 2, implying that the muscles of minors are not fully developed, resulting in more easily distinguishable biometrics. For instance, minors often perform redundant operations when operating a smartphone.

Experiment 5. (RQ 5): The user's daily use of smartphones is often in a nonstationary state. The bumps of the device lead to changes in the user's biological information. In this experiment, participants were kept on a 4 km/h treadmill and repeated all device operations. In Table 8, we compare the performance of MCBG under the moving state dataset and the total dataset (including moving and stationary states). The experimental results show that the accuracy decreases after considering the moving state, which implies that the sensors produce more irrelevant disturbances during the moving state, which affects the results.

6. Conclusion and Future Work

Nowadays, smartphones have become crucial for our daily life tasks and are used as mediums for sharing and storing sensitive information. However, the traditional mobile device protection mechanism cannot balance user experience and continuous authentication. We design and implement a continuous authentication method based on handheld biometrics features to bridge this gap. We divide the smartphone usage scenarios into more fine-grained cases, including the operation interval features. To this end, we present MCBG, a novel feature representation of the user's operating features. We model a touch action into three stages: press down, finger movements, and finger up. We capture the sensor data of each contiguous touch operation to reduce system load. In the experimental stage, we analyze the impact of the used sensor number on the authentication effect. Finally, we use CNN to classify MCBGs and train models in different motion states. Our feature representation method can also effectively distinguish whether the user is a minor or not. Besides, we release all the scripts and samples to the security community for further study by other researchers.

Our limitations are twofold. Firstly, our dataset needs to be expanded for more fine-grained analysis regarding data size and the different kinds of volunteers, such as country and occupation. Secondly, due to the multifactor influence on sensors, the classification effect is instability. Noise reduction algorithms can be introduced in future research.

Data Availability

The authors release the full dataset and extracted feature set to researchers for reproducibility purposes on their website: https://github.com/Zurich1994/MCBG.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under 2020B0101360001, the National Key Research and Development Program of China under Grant 2020YFB1406902, the Shenzhen Science and Technology Research and Development Foundation under no. JCYJ20190806143418198, the National Natural Science Foundation of China (NSFC) (61872110), the Fundamental Research Funds for the Central Universities (Grant no. HIT.OCEF.2021007), and the Peng Cheng Laboratory Project (PCL2021A02).

References

[1] Y. Wang, Q. Wang, X. Chen et al., "Containerguard: a real-time attack detection system in container-based big data platform," *IEEE Transactions on Industrial Informatics*, vol. 18, 2020.

[2] H. Meinedo and I. Trancoso, "Age and gender classification using fusion of acoustic and prosodic features," in *Proceedings of the Eleventh Annual Conference of the International Speech Communication Association*, Makuhari, Chiba, Japan, September 2010.

[3] S. M. Mirhassani, A. Zourmand, and H.-N. Ting, "Age estimation based on children's voice: a fuzzy-based decision fusion strategy," *The Scientific World Journal*, vol. 2014, Article ID 534064, 9 pages, 2014.

[4] C. Basaran, H. J. Yoon, H. K. Ra, S. H. Son, T. Park, and J. Ko, "Classifying children with 3d depth cameras for enabling

children's safety applications," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 343–347, Seattle, Washington, September 2014.

[5] Microsoft, "How Old Do I Look!," 2022, https://www.microsoft.com/zh-cn/p/how-old-do-i-look/9mw6bvjqjwzx.

[6] H. Weda and M. Barbieri, "Automatic children detection in digital images," in *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo*, pp. 1687–1690, IEEE, Beijing, China, July 2007.

[7] Y. H. Kwon and N. da Vitoria Lobo, "Locating facial features for age classification,"vol. 2055, pp. 62–72, in *Proceedings of the Intelligent robots and computer vision XII: Algorithms and techniques*, vol. 2055, pp. 62–72, International Society for Optics and Photonics, Boston, MA, USA, August 1993.

[8] P. Gnanasivam and D. S. Muttan, "Estimation of age through fingerprints using wavelet transform and singular value decomposition," *International Journal of Biometric and Bioinformatics*, vol. 6, no. 2, pp. 58–67, 2012.

[9] O. Büyük and L. M. Arslan, "An Investigation of Multi-Language Age Classification from Voice," in *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2019)*, pp. 85–92, Setúbal, Portugal, January 2019.

[10] H. Li, L. Shen, Y. Wang, J. Feng, H. Tan, and Z. Li, "Risk measurement method of collusion privilege escalation attacks for android apps based on feature weight and behavior determination," *Security and Communication Networks*, vol. 2021, Article ID 8814844, 2021.

[11] M. Abuhamad, T. Abuhmed, D. Mohaisen, and D. Nyang, "Autosen: deep-learning-based implicit continuous authentication using smartphone sensors," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5008–5020, 2020.

[12] E. Marasco and A. Ross, "A survey on antispoofing schemes for fingerprint recognition systems," *ACM Computing Surveys*, vol. 47, no. 2, pp. 1–36, 2014.

[13] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," *Woot*, vol. 10, pp. 1–7, 2010.

[14] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. 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, pp. 136–148, 2012.

[15] Z. Sitová, J. Šeděnka, Q. Yang et al., "Hmog: new behavioral biometric features for continuous authentication of smartphone users," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 877–892, 2015.

[16] M. E. Crosby and C. S. Ikehara, "Continuous identity authentication using multimodal physiological sensors,"-vol. 5404, pp. 393–400, in *Proceedings of the Intelligent robots and computer vision XII: Algorithms and techniques*, vol. 5404, International Society for Optics and Photonics, Boston, MA, USA, August 2004.

[17] C. Bo, L. Zhang, T. Jung, J. Han, X.-Y. Li, and Y. Wang, "Continuous user identification via touch and movement behavioral biometrics," in *Proceedings of the 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, IEEE, Austin, TX, USA, December 2014.

[18] X. Zhao, T. Feng, W. Shi, and I. A. Kakadiaris, "Mobile user authentication using statistical touch dynamics images," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1780–1789, 2014.

[19] T. Feng, Z. Liu, K.-A. Kwon et al., "Continuous mobile authentication using touchscreen gestures," in *Proceedings of the 2012 IEEE Conference on Technologies for homeland Security (HST)*, pp. 451–456, IEEE, Waltham, MA, November 2012.

[20] J. Ilonen, "Keystroke Dynamics," *Advanced Topics In Information Processing–Lecture*, pp. 03-04, 2003.

[21] S. Z. Syed Idrus, E. Cherrier, C. Rosenberger, and P. Bours, "Soft biometrics for keystroke dynamics: profiling individuals while typing passwords," *Computers & Security*, vol. 45, pp. 147–155, 2014.

[22] Y. Uzun, K. Bicakci, and Y. Uzunay, "Could we distinguish child users from adults using keystroke dynamics?," 2015, https://arxiv.org/abs/1511.05672.

[23] C. Epp, M. Lippold, and R. L. Mandryk, "Identifying emotional states using keystroke dynamics," in *Proceedings of the Sigchi Conference on Human Factors in Computing Systems*, pp. 715–724, Vancouver, BC, Canada, May 2011.

[24] S. Al Maadeed and A. Hassaine, "Automatic prediction of age, gender, and nationality in offline handwriting," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, pp. 1–10, 2014.

[25] R.-D. Vatavu, L. Anthony, and Q. Brown, "Child or adult? Inferring smartphone users' age group from touch measurements alone," in *IFIP Conference on Human-Computer Interaction*, pp. 1–9, Springer, Manhattan, NY, USA, 2015.

[26] E. Davarci, B. Soysal, I. Erguler, S. O. Aydin, O. Dincer, and E. Anarim, "Age group detection using smartphone motion sensors," in *Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 2201–2205, IEEE, Kos, Greece, August 2017.

[27] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[28] Android, "Documentation for App Developers," 2022, https://developer.android.google.cn/docs.

[29] D. F. Smith, A. Wiliem, and B. C. Lovell, "Binary watermarks: a practical method to address face recognition replay attacks on consumer mobile devices," in *Proceedings of the IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2015)*, pp. 1–6, IEEE, Hong Kong, China, March 2015.

[30] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 806–813, Columbus, OH, USA, March 2014.

[31] C. Shen, Y. Li, Y. Chen, X. Guan, and R. A. Maxion, "Performance analysis of multi-motion sensor behavior for active smartphone authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 48–62, 2017.

[32] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999.

[33] M. Ehatisham-ul-Haq, M. Awais Azam, U. Naeem, Y. Amin, and J. Loo, "Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing," *Journal of Network and Computer Applications*, vol. 109, pp. 24–35, 2018.

[34] S. Amini, V. Noroozi, A. Pande, S. Gupte, P. S. Yu, and C. Kanich, "Deepauth: a framework for continuous user re-authentication in mobile apps," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 2027–2035, Torino, Italy, October 2018.

[35] T. Anusas-Amornkul, "Strengthening password authentication using keystroke dynamics and smartphone sensors," in

Proceedings of the 9th International Conference on Information Communication and Management, pp. 70–74, Prague, Czech Republic, August 2019.

[36] A. Jain and V. Kanhangad, "Exploring orientation and accelerometer sensor data for personal authentication in smartphones using touchscreen gestures," *Pattern Recognition Letters*, vol. 68, pp. 351–360, 2015.

[37] T. Nguyen, A. Roy, and N. Memon, "Kid on the phone! toward automatic detection of children on mobile devices," *Computers & Security*, vol. 84, pp. 334–348, 2019.

[38] X. Li, S. Malebary, X. Qu, X. Ji, Y. Cheng, and W. Xu, "icare: automatic and user-friendly child identification on smartphones," in *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*, pp. 43–48, Tempe, Arizona, USA, February 2018.