PresSafe: Barometer-based On-screen Pressure Assisted Implicit Authentication for Smartphones

Muyan Yao, *Member, IEEE,* Dan Tao*, Ruipeng Gao,
Jiangtao Wang, Sumi Helal, *Fellow, IEEE* and Shiwen Mao, *Fellow, IEEE*

***Abstract*—Graphic-pattern-based implicit authentication has been successfully exploited to elevate the security of smartphones. On-screen pressure is one of the key features in such approach since it can reveal users' touch pattern. However, state-of-the-art approaches rely on a system API to obtain on-screen pressure, which is not adequately accurate and cannot meet the demands of robust implicit authentication. To bridge this gap, we propose *PresSafe*, a novel implicit authentication system that utilizes the smartphone's built-in barometer sensor to measure pressure during the unlocking process, and to utilize the pressure data in authentication. A key technical challenge in utilizing barometer sensing, however, is to understand the user activity through measured pressure. To overcome this challenge, *PresSafe* leverages barometer data along with data from other conventional but heterogeneous ambient sensors to produce accurate and robust user activity descriptions. *PresSafe* utilizes a transfer learning based hybrid workflow to integrate user activity representation learning with a lightweight classical authentication algorithm to obtain a unified model. This approach offloads computational cost from the terminal and addresses privacy concerns. To ensure applicability of our approach despite data heterogeneity and insufficient training data, we utilize a channel-adaptive data processing mechanism. Extensive experiments utilizing more than 70,000 records from 23 volunteers in 6 different locations show that *PresSafe* achieves an FAR of 0.45 %, an FRR of 0.49 %, and an EER of 0.47 %, which clearly demonstrate its superiority over several existing solutions.**

***Index Terms*—Implicit authentication, smartphone, barometer, pressure sensing, transfer learning, heterogenous data, representation learning.**

I. Introduction

CONCERNS of data security have sparked ages of exploration of proper security schemes for smartphones. Customers are already familiar with a series of authentication schemes that can be roughly divided into biometrics based ones and token based ones. The former one requires additional hardware, and can be easily attacked [1]. Besides, environment factors, body accessories can prevent it from reading such biometrics, impacting its performance. On the other hand, the situation for the latter group of methods can even be worse [2]–[6].

Over the years, implicit authentication has been studied to elevate the security. Approaches in [7]–[9] utilize built-in IMU sensors to obtain a description of users' behavior for identity recognition. Furthermore, implicit authentication can be fused with traditional pattern unlocking approaches [10]–[15] to form two-factor authentication systems. These systems do not require costly hardware, are environment-independent, and can hardly be attacked due to its working mechanism. Since these systems are imperceptible to users while delivering a higher security, it has performance advantages over traditional solutions.

We observe that most smartphone implicit authentication approaches have one thing in common - they all involve touch pressure as one of the key features in model building and classification. It is thus interesting to examine whether the pressure feature is truly reliable as it seems in these works. To answer this question, we have conducted a preliminary and empirical study on several major mobile operating systems of smartphones. We found that, Apple has discontinued iPhone's support for "3D Touch" after 2018 [16], and in the last three years, none of the globally-launched commercial Android phones is capable of full-screen force sensing. In other words, most smartphones do not come with dedicated pressure sensing structures on their display module, limiting their ability to provide on-screen pressure detection in a precise way. As a compromise, authentication methods on these smartphones rely on an application programming interface (API) to provide a so-called "pressure" description while the user is interacting with the phone. However, this solution is based on an assumption that the harder the user presses, the bigger the touchpoint traces will be. Through our studies, we found that this assumption is not reliable and cannot be met in many real world scenarios. Our studies confirm that the API is intended for general-purpose use only, but can hardly serve high precision applications, including implicit authentication. This observation motivates to develop a more physical and accurate way to sense the on-screen pressure

M. Y. Yao and D. Tao (corresponding author) are with the School of Electronic and Information Engineering, Beijing Jiaotong University, Haidian District, Beijing, China, and the State Key Laboratory of Integrated Services Networks, Xidian University. E-mail: muyanyao@bjtu.edu.cn, dtao@bjtu.edu.cn.

R. P. Gao is with the School of Software Engineering, Beijing Jiaotong University, Haidian District, Beijing, China. E-mail: rpgao@bjtu.edu.cn.

J. T. Wang is with the Center for Intelligent Healthcare, Coventry University, CV1 5RW Coventry, United Kingdom. E-mail: jiangtao.wang@coventry.ac.uk.

S. Helal is with the CISE Department, University of Florida, Gainesville, FL 32611, United States. E-mail: helal@cise.ufl.edu.

S. Mao is with the Department of Electrical and Computer Engineering, 200 Broun Hall, Auburn University, Auburn, AL 36849-5201, United States. E-mail: smao@ieee.org.

Manuscript received April 11, 2022; revised July 20, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61872027 and Grant 62072029, Open Research Fund of the State Key Laboratory of Integrated Services Networks under Grant ISN21-16, and the Beijing Natural Science Foundation under Grant L192004. S. Mao's work is supported in part by the NSF under Grant ECCS-1923163.

DOI: 10.1109/JIOT.2022.3199657

during user's sliding, and utilize it for implicit authentication, but without the need of any additional hardware modification to the smartphone.

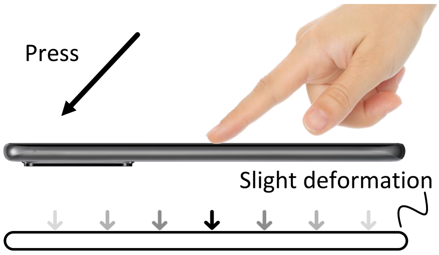


Fig. 1: Illustration of the correlation between the press behavior and the pressure variation inside smartphone.

Through our experiments, which will be elaborated later in the paper, we observe a clear correlation between the sliding behavior during the pattern unlocking process and variations of pressure inside the smartphone's chassis. We have taken advantage of this phenomenon (see Fig. 1) to form a more comprehensive description of the user's behavior on the basis of conventional motion based data along with features of on-screen pressure. By utilizing the fine-grained information in the touch events with the help of barometer data, we depict the user's behavior on the third dimension of the screen (see Fig. 2c), beyond the conventional 1D and 2D features (see Fig. 2a, and 2b), elevating the authentication performance by providing additional information that is mishandled by existing approaches.

In this paper, we propose *PresSafe*, an implicit authentication system for smartphones, which utilizes data from both the barometer and other conventional ambient sensors to produce a more robust portrait of user's behavior. However, despite the promising power of barometer sensory data, it is not straightforward to apply it to improve the overall implicit authentication performance in practice, due to the following technical challenges.

Challenge 1: Environmental impact and sensor heterogeneity. Being an environmental sensor, the barometer receives contextual information from both the user and the environment. In our case, the collaboration between the barometer and other conventional ambient sensors further multiplies the difficulty in data processing. A solution to offset the adverse effects caused by changes in the environment and the heterogeneity in sensors, while still being able to retrieve and utilize information of user's behavior from the readings, is necessary.

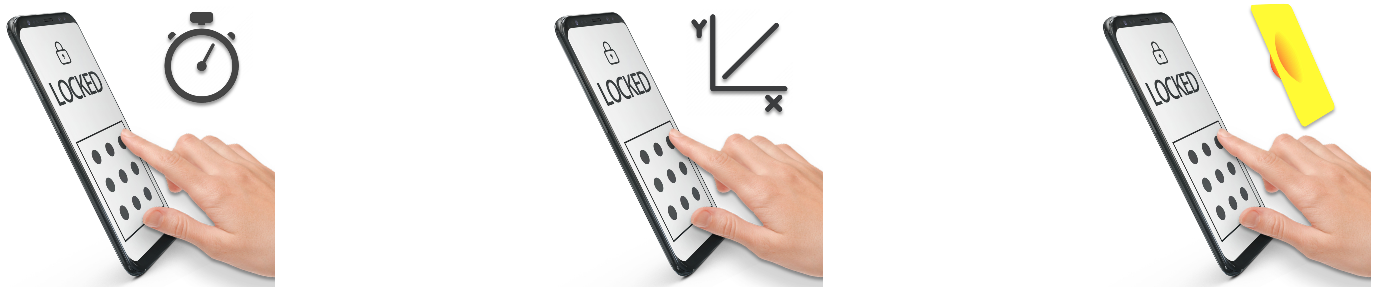
Challenge 2: Insufficient data and feature extraction. Behavior information of the user is needed to tune our model, but excessively repeating this process should be restrained to ensure good user experience. This situation leads to insufficiently collected data. Moreover, the conventional manual feature extraction approach may also result in an unrepresentative feature set. We need to address the side effects from this end and find proper ways to improve the overall performance despite insufficient data.

Challenge 3: Balance between performance, computational cost and privacy. This work is focused on mobile security while controlling the computational cost and energy consumption on terminal devices. To avoid intemperate power consumption of the tuning process, a plausible solution is to offload part of the training to remote servers. However, the involved data are highly sensitive since they contain real user's behavior information. We need to cautiously offload computational tasks from the terminal so that it never causes any compromise of user's privacy.

We incorporate a series of techniques to address these challenges, including barometer-assisted on-screen pressure sensing, transfer learning, representation learning, and channel-adaptive processing. These techniques ensure that more robust features are learned form a comprehensive user representation, requiring no dedicated sensors on screen panel but enabling an elevated overall authentication performance, even with a smaller set of sensors when compared to competing solutions. The contributions of this paper can be summarized as follows:

- Instead of using a conventional general-purpose API for on-screen pressure acquisition, which is not accurate enough for implicit authentication, this is the first work to introduce the integration of barometer and other ambient sensors during the unlocking process, to produce a more comprehensive user profile, and achieve improved robustness and accuracy.
- To address *Challenge 1*, we present a special data fusion approach. We use a fusion technique to deal with the heterogeneity in various sensors for behavior description collection, and a pre-processing algorithm to offset any potential impact cased by the changes in the environment while maintaining effective user information.
- Concerning *Challenge 2*, a 1D-CNN based representation learner is proposed. We incorporate a timestep-depthwise-convolution-based deep learning model to produce a robust user feature profile in a fashion of representation learning, so that the physical patterns shared within- and inter-sensor data are effectively utilized. We also propose an additional channel-adaptive algorithm to elevate the performance of feature extraction even when data are insufficient.
- To deal with *Challenge 3*, we design a hybrid workflow with transfer learning and computation offloading, so as to integrate user activity representation learning with a lightweight classical authentication algorithm, and offload most training overhead from terminal devices to cloud in a privacy-protected manner. In this fashion, we reduce the local resource consumption while avoiding possible performance degradation or privacy concerns.
- We develop a prototype on several off-the-shelf smartphones and conduct extensive experiments with 23 volunteers. On 6 different locations, more than 70,000 samples are recorded, based on which the performance benchmark is derived. Experimental results demonstrate that *PresSafe* achieves a False Acceptance Rate (FAR) of 0.45 %, a False Rejection Rate (FRR) of 0.49 %, and an Equal Error Rate (EER) of 0.47 %, and outperforms several baseline schemes.

The remainder of this paper is organized as follows. In



(a) 1D on-screen feature (elapsed time)

(b) 2D on-screen feature (with touchpoints traces)

(c) 3D on-screen feature (with on-screen pressure)

Fig. 2: The illustration of different dimensions of the on-screen feature.

Section II, we introduce the background and related works in the field of smartphone implicit authentication. In Section III, we use an off-the-shelf smartphone and a dynamometer for a preliminary study on the correlation between the user's unlocking action and the readings of the barometer. In Section IV, we explain the framework and workflow of the *PresSafe*, and in Section V we elucidate the details of the barometer-assisted pressure sensing technique that is dedicated to portray the behavior information during the sliding process. Later in this paper, Section VI explains the approach to perform data fusion and data augmentation. Section VII elucidates the feature extraction process and how we manage to implement and deploy a neural structure on a terminal device in a privacy and energy caring manner. Section VIII present the evaluation, and Section IX is the conclusion of the *PresSafe*.

II. Background and Related Works

A. Token-based Explicit Authentication

This type of authentication mechanisms, namely the PIN (Personal Identification Number), password or pattern unlocking, are the most traditional ways to secure a system. If the provided token matches the preset one, access will thus be granted. Widely accepted though, they only provide entry-level security protection, and hardly provide solutions to a series of classical attack scenarios, including shoulder surfing attacks [17], [18], brute-force attacks, smudge attacks [4], [19], etc.

The root cause for this defect lies in the lack of correlation in the token and the provider's identity: these schemes do not verify if the provider is legitimate, but only correctness of the token. Clearly, every person with a correct token could be granted access.

B. Biometrics-driven Explicit Authentication

Lately introduced phones usually feature a combination of different biometric scanners, e.g., fingerprint reader, 2D/3D facial recognition module, iris scanner, etc. Providing a natural way to authenticate though, their weaknesses are also obvious.

1) **Prone to be attacked.** The explicit biometric features are visible and easy to replay, causing them prone to be attacked through smudge attacks [20], [21], thermal attack [22], or replay attack [23].
2) **Biometric immutability.** Using the same set of biometrics in different systems may cause fatal security failures if one of these systems is compromised, since the biological characteristics can hardly be changed as the human body matures. The consequence of any compromise of the biometric credential would be permanent [24].
3) **Additional cost.** Behind these biometrics-driven solutions, specific hardware modules are required, causing additional cost and not being available on all kinds of smartphones.
4) **Work scenario requirements.** Due to the need to scan relevant biometrics, these solutions usually hold stringent requirements on working scenarios [25]. Specifically, due to the epidemic prevention requirements, performance of facial recognition systems is usually downgraded when a mask in on.

These limitations narrow down the scenarios in which the biometrics-driven solutions are actually applicable.

Admittedly, some recent studies have tried to update the way to extract biometric features. [2] captures the fingerprint-induced sonic effect during swipings so as to authenticate the user. [26] senses lip motion by the reflected ultrasonic signals. In [27], they use digital accessories to detect the body electric potentials. [28] relies on vibration captures by accelerometer to reconstruct the user's heartbeat signal. [29] use user's fingerprint to assist in generating private keys. In [30], a dynamic feature selection for hand-writing is developed to enhance the authentication performance. However, these solutions are still not perfect: signals used in these works are generated from biochemical effect [2], [26], [28], which are weak and are not directly measured thus needs a high signal-to-noise ratio environment; or are sent from other devices [27], which face common problems in machine-to-machine authentication, as in [31], [32]. The approaches in [29], [30] heavily rely on online servers and can hardly be performed in an offline manner.

C. Solutions of Implicit Authentication

Implicit authentication refers to the type of techniques that enable smart devices to recognize its owner not only from conventional tokens, but also by the behavior the user demonstrates, especially during the unlocking attempts.

Feng et al. [33] proposed a solution leveraging the RFID (Radio Frequency Identification) devices to detect walking pattern features and help the implicit authentication process. The studies in [34] found the speaking-induced body sound transmission from the throat and the ear canal could be used for extracting a sound conduction pattern. Wang et al. [35] trying to model biological feature from the deformation caused by in-ear wearables, and combine this feature with dynamic motions for user authentication.

The above-mentioned solutions require the user to carry additional hardware and/or perform designated operation that is beyond the usual action to operate the phone. Being novel though, they fail to provide a friendly user interface for the unlocking process, therefore may not be well received by the crowd. Based on the fact that graphic credentials can be memorized more easily [36], and around 40 % of the participants in a work [37] are more willing to use pattern unlocking as the authentication mechanism on their smartphones, researchers are also trying to develop solutions for pattern unlocking based implicit authentication.

Angulo et al. [38] used *Random Forest* to build the authentication model, requiring data not only from the owner, but also from potential attackers. Similarly, Alpar et al. [39] proposed a histogram-based technique for authentication. These schemes used multi-class classification algorithms as classifier to determine the user's identity, which makes them unsuitable for real applications: multi-class classifiers need data of all labels, i.e., the intruders and the owners, to get trained. Obviously the assumption that a device owner being able to obtain behavior information from intruders in advance, is not reasonable.

New sensors are used in the authentication process, and relevant algorithms are also developed in works that are more recent. Liu et al. [13] introduced statistics based schemes as the authentication module, and utilized accelerometer for motion detection. They also introduced feature processing including basic manual extraction and selection, before the authentication model receives the data. Ku et al. [12] noticed that the user's behavior may vary in different scenarios. They collected two sets of user data: sitting and walking, and found that the authentication performance would degrade, once the application scenario is not fixed. Wang et al. [40] proposed a context-aware module to classify the application scenario, and send the weighted user data to the corresponding authentication module. Shi et al. [41] further adopted a polygonal line weighted strategy. This method takes consistency of unlocking patterns into account, then analyzes the patterns with different grains, and enhances the information stored in the patterns' principal parts though weighting.

Intuitively, there should exist much variance in a user's pressing behavior, especially when the user attempts to draw an unlocking pattern. The prior works used, directly or indirectly, the pressure on the screen, as one of the key features in their classification mechanisms. But according to our empirical research, the main-stream off-the-shelf smartphones do not come with dedicated pressure sensing structures on the display module, resulting in their inability of precise on-screen pressure detection. Actually, the prior works used an Android built-in API to obtain a so-called "pressure" feature of touch events [42], which is not meant for high-end application, i.e., authentication. The information fetched this way is an approximation based on the size of touchpoints, making it unsuitable for authentication tasks.

III. Preliminary and Empirical Study

In this section, we first discuss the status quo of smartphone on-screen pressure sensing. Then, we conduct preliminary experiments to validate the relationship between sliding and barometer readings.

A. Status Quo of Pressure Sensing on Smartphones

When the user interacts with the phone, on-screen pressure not only reflects the user's operating habits, but also captures the touch process dynamically. Therefore, on-screen pressure has become a key feature in a number of related studies.

To verify if the pressure feature is accurate enough for authentication task, we made an investigation on the consumer electronics marker.[1] To the best of our knowledge, no matter for iOS or Android, none of the commercial phones globally launched in the last three years are equipped with a dedicated on-screen pressure sensing structure [16], [43]. Considering this, researchers in the related works used a workaround to sense the on-screen pressure.

The workaround is based on an Android API that can report approximate, so-called "pressure" detection values on devices without such physical sensors [42], but these values are highly related to the touchpoint size. As mentioned before, the relationship between the size of touchpoint and the on-screen pressure could be loose. If the touch size is directly, and only related to the on-screen pressure, then this API could work well. Otherwise, the "pressure" feature obtained in this way will not work properly in implicit authentication tasks.

We conduct a qualitative experiment using a Galaxy Note 10+, on which a customized Android application is deployed to capture the pressure resulted from user's sliding action through the Android API used in many existing studies. During the experiment, we try to stable the touching force to the same minimum level, so that the screen can sense our touch. To demonstrate if the touchpoint size will be affected by factors other than the touch pressure, we repeat this process with different parts of the finger.

From Fig. 3, it is clear that, different contact parts of the same finger resulted in noticeable differences on the touchpoint size during the sliding process. **Clearly, changing on touchpoint size has been translated different on-screen touch pressures, thus affecting the authentication performance of the studies which utilize this API.** This way of getting the pressure value is indeed dedicate-sensor-free but not sufficiently accurate for security applications. We further discuss this phenomenon in Section V-B.

[1] We focus this study within the field of consumer electronics. This is because the approach in this paper mainly focuses on solutions that can directly be applied to off-the-shelf smartphones. We understand that, through some additional hardware, such as customized wave guide layer or laser detectors, a more precise pressure sensing can be achieved. But it is also obvious that, this kind of solutions need additional hardware, thus not being within reach of the most population.

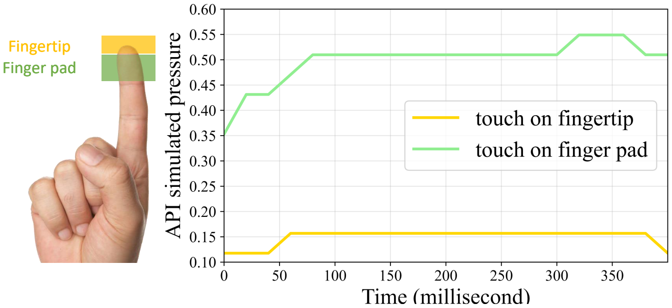


Fig. 3: The API-simulated pressure feedback while different finger parts interacting with the phone. The results are the average of ten attempts.

B. Sliding Operation and the Barometer Readings

We further conduct experiments with the same Galaxy Note 10+, but with the barometer enabled. This additional experiment is for the purpose of exploring any possible relationship between the barometer readings and the pressure on the screen. In the experiment, we use a zigzag unlocking pattern as the preset and record the barometer readings during the unlocking process. When the unlocking process starts, nine dots will be displayed on the screen (Fig. 4), and the user connects the dots sequentially with her finger to unlock the phone.

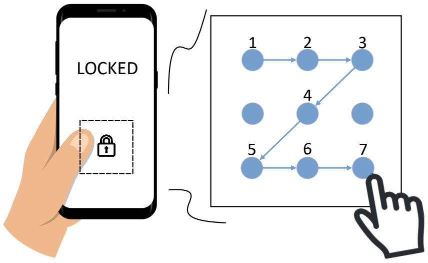


Fig. 4: An example of pattern unlocking process.

Through our experiments, we find it interesting that the feedback value of the barometer vary during the pattern unlocking process. While the user operating the phone, the exerted force causes the chassis of the phone to deform, further impacting its inner air pressure. We record this phenomenon using the barometer sensor, and link the barometer readings to the on-screen pressure in a meaningful way, thus hopefully creating a more robust presentation of user's behavior.

Although the barometer assisted on-screen pressure sensing looks promising, we first need to examine if the barometric pressure profile is adequate to distinguish different users. We hire six volunteers to repeat drawing the unlocking pattern on the phone, so as to analyze the properties of the collected data, as plotted in Fig. 5. Being overlapping partially though (User 1, 4, 5, 6), it is clear that the spread of such data is rather limited, and the time sequence essence helps them to hold significant differences throughout the record.

To check on this point, we use the following methods to resample and scale the pressure profiles. First, we conduct FFT (Fast Fourier Transform) transformation to resample all the concerned data to the same length. Then, we use the following formula to scale the numerical range of them to $[0, 1]$.

$$X_{scaled} = (X - X_{min})/(X_{max} - X_{min}) \tag{1}$$

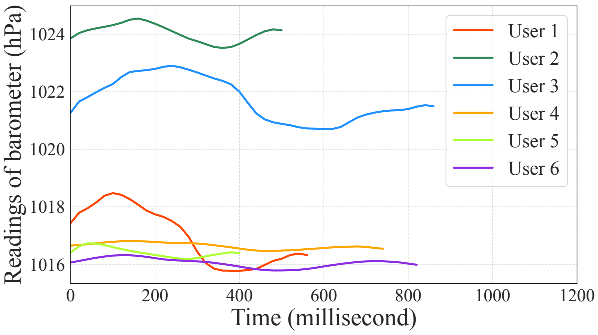


Fig. 5: On-screen pressure profiles of different users.

The process data are presented in Fig. 6. We use two vertical red dashed lines to indicate the time interval when different records reach their first peak, and two blue lines for the first minimum. It can be observed that those data reflect the touch, press, and sliding events of the user, and are diversified in patterns of rhythm, variation, and distribution. Obviously, the profiles are essentially the portrayal of the user's operating behavior, and the time rhythms contained in these data still help our model differentiate them even after scaling the data range.

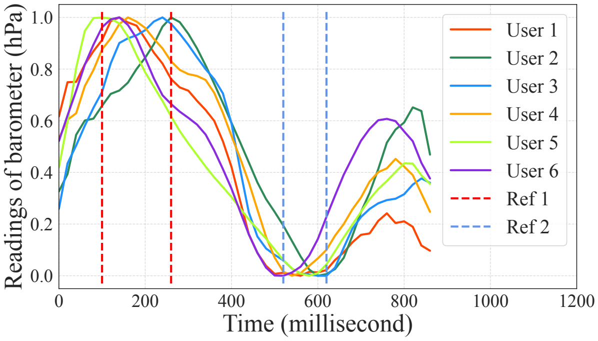


Fig. 6: Processed on-screen pressure profiles.

We then use distribution analysis and Euclidean distance to further validate this point. Table I demonstrates the numerical spread of each single user's pressure profile. We observe the mean value of the pressure profile of the six volunteers varies in a relatively large area, while the variance of it stays low and stable.

TABLE I: Properties of Data Spread in the On-screen Pressure Profile

Property	Mean[2]	Variance
Mean	1016.75	0.47
Variance	0.058	0.0036

[2] This column presents the mean of the corresponding property, i.e., the mean of each volunteer's mean value. Similar for the variance column.

On the other hand, we first pre-process the collected data as described above, and calculate the mean recording of five pressure profiles from user 1 as $\boldsymbol{p}_{ref}$. Then we use another three pressure profiles of user 1, as denoted by $\boldsymbol{p}_{pos1}$ to $\boldsymbol{p}_{pos3}$. Besides, the pressure profiles from other users are also used for comparison, as denoted by $\boldsymbol{p}_{neg1}$ to $\boldsymbol{p}_{neg3}$. The calculated Euclidean distance of these profiles are presented in Table II.

TABLE II: Euclidean Distances Between Collected Profiles

Sample	Euclidean distance to $\boldsymbol{p}_{ref}$	Comparison
$\boldsymbol{p}_{pos1}$	0.49	**+ 0.00 %**
$\boldsymbol{p}_{pos2}$	0.52	**+ 7.41 %**
$\boldsymbol{p}_{pos3}$	0.55	**+ 13.79 %**
$\boldsymbol{p}_{neg1}$	1.71	**+ 71.52 %**
$\boldsymbol{p}_{neg2}$	1.86	**+ 73.89 %**
$\boldsymbol{p}_{neg3}$	3.05	**+ 84.04 %**

From the Euclidean results, we observe the pressure profiles under user 1 show a similar distribution under the time sequence, while the data from different users are discriminative. The distribution also proofs that, for a single user, her profile data span is narrow and limited, which ensures that the data will not be mixed up altogether; when the data are mixed up, the barometric on-screen pressure profile is adequately discriminative among different users. These conclusions also apply to the data we collected from other users.

IV. The Proposed System

In this section, we present the framework of our proposed authentication scheme, *PresSafe*, in detail. Fig. 7 illustrates an overview of the system, which consists of two main parts: the training stage and the guarding stage.

A. The Training Stage

Before being used as an authentication system, *PresSafe* requires the device owner's behavior information for preprocessing and model training. The main steps are as follows:

1) Pre-training. Since the local computational and energy resource could both be limited for the phones, we partially offload the training process to the cloud by pre-training a representation learner remotely, in order to ensure privacy of such sensitive user data. In our designed workflow, the training process does not involve any real data from the device owner, but uses a combination of desensitized data from volunteers. In real applications, this part can be done in advance, so this model is pre-installed and distributed with the devices.

2) Local fine-tuning. When initializing the smartphone's security system, the device owner will be required to enter her behavior information by simply operating the preferred graphic unlocking pattern for a few times. In this process, a set of sensors will be invoked to capture continuous readings.

When the initiating process is done, the generated behavior description will be processed by the data augmentation module, simulating the owner's unlocking behavior in different scenarios. Then the augmented descriptions will be reshaped to form a three-dimensional array to fit the timestep-depthwise-convolution-based structure, i.e., the feature extractor. After a slight local fine-tuning of the feature extractor, the top few layers will be discarded and the remaining layers will function as the representation learner. The representation learner can thus generate a user representation, which will be used for the training of the authentication module.

B. The Guarding Stage

In this stage, the *PresSafe* is ready to guard the smartphone and ensure the safety of data stored on it.

1) Submit the unlocking attempt. The user will first be guided to finish the pattern, but for only once. During this process, a set of sensors are used to generate the behavior description. After this, the description will be translated into a user representation through the tuned representation learner.

2) Authentication. To validate the user's identity, our algorithm will examine the outputs from both the explicit and implicit module. Only when the explicit unlocking pattern drawn by the user and the implicit representation both match the template, access shall be granted. Through this two-factor implicit authentication approach, we can enhance the device's security.

V. Barometer-assisted Pressure Sensing

A. Analysis of the Barometer Readings

In Section III-B, we have qualitatively discussed the relation between the barometer readings and the sliding operation. Results demonstrate that the barometer readings can be used to capture the deformation caused by the user's operations and thus being able to represent the pressure profile. For a further exploration, we provide an analysis on the barometer readings.

First, to examine if the barometer readings actually capture the pressure on the screen, we use a digital dynamometer to measure the force applied to the frontal panel of the smartphone as in Fig. 8a. The phone is placed on a horizontal and rigid plane, since we expect no potential deformation from the holding surface to affect our experiment results. On the other hand, the dynamometer is hand-held, and its detector directly contacts with the center of the smartphone's display panel. The detector is used to simulate how the finger interacts with the phone. The values both from the dynamometer and the barometer are recorded and presented in Fig. 8b.

In this figure, the horizontal axis represents the readings from the dynamometer and the vertical axis stands for the variation of barometer readings. To reveal the correlation between these two features, we use first-order linear regression to produce a trend line. It can be observed that the variation of the built-in barometer readings and the dynamometer readings exhibit a linear correlation.

Considering the correlation between these two events, we design an adaptive pressure sensing approach without using extra sensors. Every time when the user interacts with the phone, there will be a pressure being applied to the screen, which causes a slight deformation of it. That is, the display module is sunken inward, leading to a fluctuation of the internal air pressure inside the chassis. The value reported

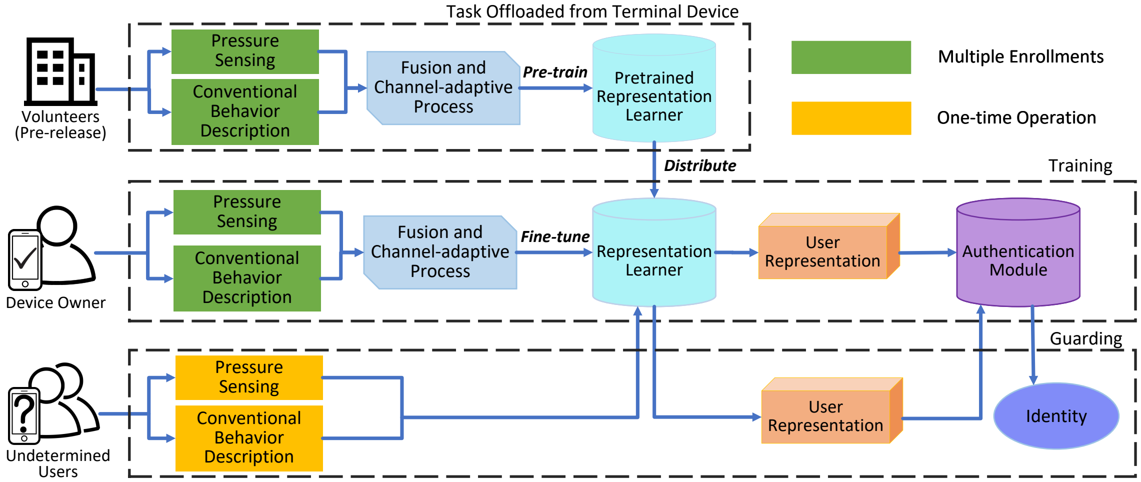
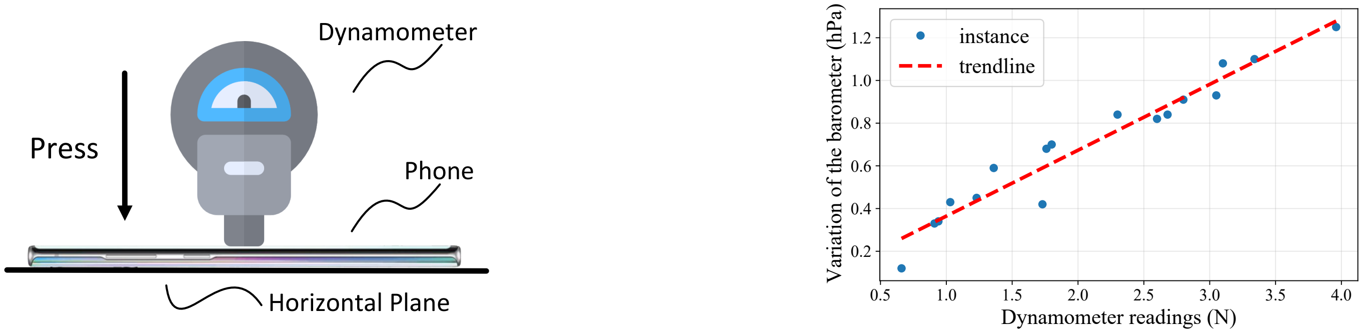


Fig. 7: The framework of PresSafe.



(a) Illustration of the experiment using the dynamometer and smartphone

(b) Correlation between readings from the barometer and the dynamometer

Fig. 8: The experiment with the dynamometer and smartphone.

by the built-in barometer will then fluctuate accordingly. By analyzing the trend of the series of readings reported by barometer, it is now possible to sense the pressure being applied to the screen.

B. Sliding Pressure Sensing

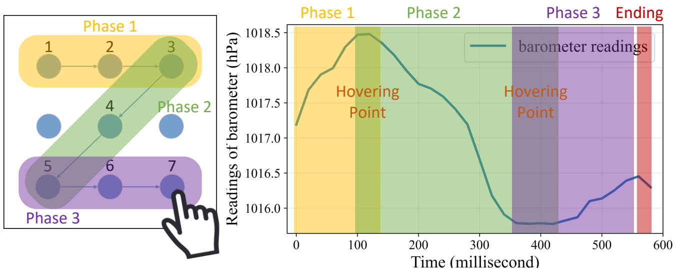


Fig. 9: The process of drawing pattern Z and the corresponding trend of barometer readings.

Considering that the pattern unlocking is a sliding process, we further conduct experiments to assess the correlation between on-screen sliding operations and the variation of barometer readings. Fig. 9 shows the process of completing the Z pattern.[3] The pressure resulted from different sliding phase of the pattern drawing process can be captured by the built-in barometer.

Overview of the pattern. According to the dynamic characteristics, which can be captured by ambient IMU sensors along with the sliding pressure, the entire unlocking process can be dissected into the following three phases. Phase 1 contains dots 1-3, which is the beginning part of the sliding. The swipe direction is very simple, namely from left to right. Phase 2 contains dots 3-5, and this phase is between two turning points, i.e., dot 3 and dot 5. Since the finger needs to turn its direction at the turning points, the time consumed at these two dots are supposed to be relatively longer, and the pressure does not show a large variation. Thus, we expect two obvious hovering point located at the beginning and ending of phase 2 in the barometer readings. Phase 3 is much like Phase 1, but the ending part will be different, resulting in an additional transient response.

Phase 1. The barometer readings during this phase demonstrate a trend similar to our analysis. It can be inferred that during phase 1, when the finger moves from the left side of

[3] The participant grips the phone with his right hand, and the unlocking process is also accomplished with the thumb of the same hand in this experiment.

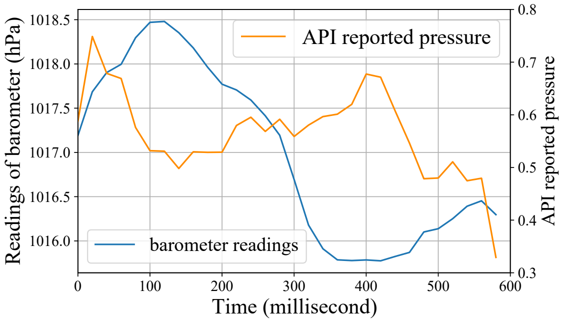


Fig. 10: A comparison of the barometer-based and software-based pressure sensing techniques.

the screen to the right side, the screen generally becomes more bent, resulting in a more significant pressure. The air inside the chassis of the phone also becomes more compressed, causing the barometer to report higher readings. On dot 3, the finger has to turn its direction to enter phase 2. Then there comes the first hovering point in the barometer value.

Phase 2. On phase 2, the finger is sliding from the upper right to the lower left of the area. Since the user performs the unlocking attempt with the same hand gripping the phone, the thumb is stretched as it moves away. Therefore, the force applied by the finger gets weaker, and the depression of the screen becomes slowly recovered and the compression of the inside air is gradually eased. Thus, the pressure sensed by the barometer will become attenuated. The decreasing values of the barometer support this conjecture. At the end of this phase, the finger comes across dot 5, where it is going to alter its direction again. Therefore, there will be another hovering point.

Phase 3. Phase 3 is the final stage of this pattern. The finger moves from left to right, so the pressure on the display panel is similar to that in phase 1. There will be an increasing air pressure inside the smartphone, resulting escalated barometer values.

Interestingly, there is an additional transient response from the barometer, when the finger leaves the screen. This process is shown in Fig. 9. When the pattern is finished, since the force applied by the finger disappears, the compression of the chassis recovers, and there is a brief drop in the value of the barometer. After a short period, the air pressure returns to the normal level.

Comparison between API simulated and barometer assisted on-screen pressure sensing. To demonstrate the limitation of the traditional software simulated sliding pressure sensing approach, we collect both barometer readings and the pressure applied to the smartphone screen, which is based on the software simulations during the unlocking process of a Z pattern, in parallel. The results are plotted in Fig. 10.

Previously we have proved the effectiveness of the barometer based sliding pressure sensing approach, as the graph from this scheme accurately captures the actual activity, and reveals the variation of the force of the user. On the other hand, the pressure simulated by the software shows a completely different trend. This approach simulates pressure values on the screen panel by the size of the touch point [42], so the results could be easily affected by the contacting part and angle of the finger at different moments.

From the above observations, it is safe to say that *PresSafe*, which captures the sliding pressure through the variation of barometer readings, is a highly promising solution balancing availability and precision. This solution avoids any involvement of dedicated hardware nor additional modifications to smartphones, thus ensuring the possibly maximum group of customers being able to benefit from this technology. On the other hand, it also mitigates the improper pressure detection method that is common in related works, by providing finer pressure sensing of the sliding operations.

VI. Data Fusion and Channel-adaptive Process

A. Conventional Behavior Description

For an implicit authentication system, the fundamental is to get a natural behavioral description during the user's unlocking process, which requires not only to sample information from the traditional interactive interface, i.e., the screen, but also to unlock the potential and explore more possibilities of ambient sensors.

In this regard, there are several effective sensors that are capable of generating the descriptions, e.g., the gyroscope, the accelerometer, etc. Through our analysis of the collected data, we find that the unique features of different user's unlocking behavior can also be found in the drawing rhythm, finger movement path, and the pressure applied to the screen while finger sliding on it. By utilizing these ambient sensors, along with the exerted pressure sensed by barometer, we can construct the user's profile from multiple dimensions and with different levels of granularity.

To conclude, the behavior description contains the following information:

1) Plane information. All the behavior descriptions generated in the smartphone screen space are referred to as plane information, i.e., the x and y pixel coordinates. This kind of information reveals the touch preference of the user.

2) Motion information. Since the movement of the user's hand has a strong relation with the motion status of the device, we adopt the data from relevant sensors to represent such activities, namely the shake, tile, rotation, roll, pitch, and yaw. Then, the corresponding biometric properties, especially the ones related to the users' hands, including the flexibility or swipe preference, are considered.

3) Pressure information. As discussed in Sec. V, the hold, squeeze, or touch behavior can result in a force directly exerted to the chassis of the smartphone. This kind of behavior can cause a variation of the barometer readings, which can be recorded and further utilized. We take this feature into account to mitigate the limitations of conventional API simulated touch force.

B. Fusion: Combating Sensor Heterogeneity

Since the integration of data both from the barometer and other ambient sensors is desired, the heterogeneity in the sensory data should be addressed. First, the sampling processes of the sensors that are invoked during the unlocking process are not synchronized. The sampling rate of the barometer is much lower than that of the IMU sensors, with a gap of up to 3 to 5 times. Second, the sensory data from different sensors have different physical meanings and different ranges.

Sensory data, as fetched from different sensors and being stored in different feature channels in the behavior description, are represented as a sequence of values

$$\boldsymbol{X} = (\boldsymbol{X}_1^{d_1}, \boldsymbol{X}_2^{d_2}, \boldsymbol{X}_3^{d_3}, \cdots, \boldsymbol{X}_T^{d_T}), \tag{2}$$

where $\boldsymbol{X_i^d} \in R^d$, for $i = 1, 2, \cdots, |T|$, and d is the dimensionality of the sensor data (e.g., $d = 3, i = 500$ in case of a 3-axis gyroscope reporting 500 samples); T is the total amount of collected samples, i.e., the count of timesteps. After concatenating all the sensors' recordings, the behavior description $\boldsymbol{B}$ is constructed.

Considering that the sampling rate of each channel (i.e., from different sensors) is constant but are different from each other, we need to ensure the sequential signals share the same length. Since the signals possess their own physical connotations, if we use regression to fit them, it will be hard to determine the proper order of polynomial terms in the regression process. Instead, we choose to deploy an FFT (Fast Fourier Transform) based approach to synchronize and align samples in different channels.

Specifically, we resample the records in different channels to synchronize their time stamps. The goal of this process is described as follows.

$$\min \sum_{c=0}^{n} \sum_{m=0}^{d} (\boldsymbol{\mathscr{E}}(\boldsymbol{X}^c[m]) - S) \tag{3}$$

where n is the total number of sensors; S is the sync factor, which will be explained later in this part; $\boldsymbol{\mathscr{E}}$ is the action of acquiring the number of timesteps of the chosen dimension, whose feedback matches the previously defined T. For instance, $\boldsymbol{\mathscr{E}}(\boldsymbol{X}^5[1])$ represents second dimension of the 6_{th} sensor recordings' maximum timestep index (index both starting at *0* for n and d). The goal in this step is to minimize the accumulated difference between the number of timesteps of different sensor's data, and the sync factor S.

To solve problem 3, we need to find the sync factor S in accordance with the T in different sensor's different dimensions. The calculation of S is defined as follows.

$$S = \boldsymbol{\mathscr{P}}\{\boldsymbol{\mathscr{E}}(\boldsymbol{X}^c[m])\} \tag{4}$$

where $c = 0, 1, \cdots n$; $m = 0, 1, \cdots d$; $\boldsymbol{\mathscr{P}}$ means the 0.8 $percentile$, which is an empirical setting.

Let $\boldsymbol{X}^c[m]$ be the signal to be up-sampled. Considering T might be greater than, equal to, or less than S, Algorithm 1 is used to process the signal.

Through our experiments in Section V-A, we observe the linear correlation between variation of the barometer readings and the sliding pressure. In the meantime, barometer, being an environment sensor, can also receive contextual information from the background that can cause ambiguities. During our experiments, we find that, first, even when the smartphone is placed still and not being pressed, the barometer readings still vary with time. According to our observations, its readings from a static smartphone could change up to 3hPa in half an hour. We conjecture that this might be caused by the change of atmospheric pressure. Second, the barometer reading also changes as the thermal status of the smartphone shifts. The room temperature is $13°C$, and we use an infrared thermometer to measure the temperature of the phone, which is $10°C$ initially. After 5 minutes, its temperature rises to $21°C$, which causes a variation in the internal air pressure up to 2hPa.

Algorithm 1 Process to align the signals.

```
Input: sync factor, $S$; raw sensory data, $\boldsymbol{X}^c[m]$;
Output: aligned sensory data, $\boldsymbol{X}_{al}^c[m]$;
while $\sum_{c=0}^{n}\sum_{m=0}^{d}(\boldsymbol{\mathscr{E}}(\boldsymbol{X}^c[m]) - S) \neq 0$ do
    for $m = 0, 1, \cdots, d$ do
        for $c = 0, 1, \cdots, n$ do
            $\boldsymbol{x}_{raw} \leftarrow \boldsymbol{X}^c[m]$;
            $T \leftarrow \boldsymbol{\mathscr{E}}(\boldsymbol{X}^c[m])$;
            if $T > S$ then
                $\boldsymbol{X}_{al}^c[m] \leftarrow downsample(\boldsymbol{x}_{raw}, S)$;
            else if $T = S$ then
                $\boldsymbol{X}_{al}^c[m] \leftarrow \boldsymbol{x}_{raw}$;
            else if $T < S$ then
                $\boldsymbol{x}_p \leftarrow zero\text{-}padding(\boldsymbol{x}_{raw}, S - T)$;
                $\boldsymbol{X}(e^{j\omega}) \leftarrow FFT(\boldsymbol{x}_p, S)$;
                $\boldsymbol{X}_{al}^c[m] \leftarrow IFFT(\boldsymbol{X}(e^{j\omega}), S)$;
            end if
            return $\boldsymbol{X}_{al}^c[m]$;
        end for
    end for
end while
```

The above problems can be mitigated by setting a limit to the time period of one single enrollment process. Pattern enrollment for one single attempt cannot take more than 2 minutes, while the normal time consumption of it is usually less than 20 seconds. Recordings longer than this limit will be truncated. We propose Algorithm 2 to mitigate the offset caused by the changes in atmospheric pressure and smartphone temperature.

Algorithm 2 Process to mitigate the offset in barometer readings.

```
Input: aligned sensor data, $\boldsymbol{X}_{al}^{baro}[0]$;
Output: processed barometer data, $\boldsymbol{X}_{al}^{baro}[0]$;
$\boldsymbol{X}_{al}^{baro}[0] \leftarrow detrend(\boldsymbol{X}_{al}^{baro}[0])$;
$\boldsymbol{x}_{init} \leftarrow \boldsymbol{X}_{al}^{baro}[0][0]$;
$T \leftarrow \boldsymbol{\mathscr{E}}(\boldsymbol{X}_{al}^{baro}[0])$;
for $t = 0, 1, \cdots T$ do
    $\boldsymbol{X}_{al\_eo}^{baro}[0][t] \leftarrow (\boldsymbol{X}_{al}^{baro}[0][t] - \boldsymbol{x}_{init})$;
end for
$\boldsymbol{X}_{al}^{baro}[0] \leftarrow \boldsymbol{X}_{al\_eo}^{baro}[0]$;
return $\boldsymbol{X}_{al}^{baro}[0]$
```

In short, detrending and differentiation are applied to the barometer sensory data $\boldsymbol{X}_{al}^{baro}[0]$ (*[0]* because barometer readings are one-dimensional data), so that the trend in the

TABLE III: Range of Data in Different Channels of the Behavior Description

Channel [4]	Minimum	Maximum
tpX (pixels)	71.72	1411.88
tpY (pixels)	257.77	1440.45
baro (hPa)	1012.37	1026.12
gyrX ($^\circ/s$)	- 2.68	1.71
gyrY ($^\circ/s$)	- 4.72	2.62
accX (m/s^2)	- 8.87	- 0.66
accY (m/s^2)	- 0.27	9.63
accZ (m/s^2)	1.67	8.47

sequence, if any, is first removed. In addition, the variation of barometer readings are kept for further analysis, so that the effects caused by varied starting point of atmospheric pressure are offset.

C. Channel-adaptive Data Processing

The amount of training data is always critical for deep learning to deliver a satisfactory performance. Prior works on implicit authentication [13], [38], [39] require participants to repeat data enrollment process multiple times. A long data enrollment process undoubtedly causes poor user experience, which urges authentication solutions to take the data acquisition cost into account.

To ensure good performance under insufficient data, data augmentation technique is needed to model the differences in the same user's behavior in various scenarios. Our intuition is to use additive noise, e.g., Gaussian noise, to simulate user's behavior variations in different scenarios.

Table III shows data sampled from various sensors during the unlocking process. We notice that the range of gyroscope readings are very narrow, with an absolute amplitude less than 5. However, as a result of the atmospheric pressure, the raw barometer readings are distributed around 1010 mPa. In addition, the touch positions may vary with the device's screen resolutions. Thus, their values span over a rather wide range, which are distinct from other channels.

Considering the various data ranges, we propose a channel-adaptive data augmentation mechanism based on jitter signal, which is a sequence of artificial signal based on Gaussian noise. Before being applied to each channel, the noise signals are scaled in advance with the corresponding scaling factor S, so that the augmented behavior descriptions hardly lose their original information. The details of this process are provided in Algorithm 4.

Let $mean(x)$ and $std(x)$ represent the mean and standard deviation of the input data. Operator $\mathscr{F}$ measures the fluctuations in a sequence $\boldsymbol{x}(n)$ as:

$$\mathscr{F}(n) = \ln(\boldsymbol{x}(n)) - \ln(\boldsymbol{x}(n-1)). \tag{5}$$

Define the fluctuation weighter $\mathscr{W}$, which combines a variation of counting function and activation function, so as to convert the counter result to a probability-like numerical output using the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$.

Algorithm 3 The algorithm of fluctuation weighter $\mathscr{W}$.

```
Input: sequence data, $\boldsymbol{x}$;
Output: output value, $\mathscr{W}(\boldsymbol{x})$;
$counter \leftarrow 0$;
$T \leftarrow \mathscr{E}(\boldsymbol{x})$;
for $n = 1, 2, \cdots T$ do
    $\boldsymbol{v}(n) \leftarrow \mathscr{F}(\boldsymbol{x}(n))$;
end for
for $n = 1, 2, \cdots T - 1$ do
    if $(sgn(\boldsymbol{v}(n)) \times sgn(\boldsymbol{v}(n+1))) < 0$ then
        $counter \leftarrow counter + 1$;
    end if
end for
$\mathscr{W}(\boldsymbol{x}) \leftarrow \sigma(counter)$;
return $\mathscr{W}(\boldsymbol{x})$
```

Algorithm 4 The process of channel-adaptive augmentation.

```
Input: aligned sensor data, $\boldsymbol{X}^c_{al}[m]$; jitter signal, $\boldsymbol{J}$;
Output: augmented sensor data, $\boldsymbol{X}^c_{aug}[m]$; scale factor
    $\boldsymbol{U}^c[m]$;
if $\boldsymbol{U}^c[m]$ is void then
    $\boldsymbol{M}^c[m] \leftarrow mean(\boldsymbol{X}^c_{al}[m])$;
    $\boldsymbol{V}^c[m] \leftarrow std(\boldsymbol{X}^c_{al}[m])$;
    $\boldsymbol{U}^c[m] \leftarrow \sigma(\frac{\sum_{m=0}^{d}(1-10^{-\frac{\boldsymbol{V}^c[m]}{\boldsymbol{M}^c[m]}})}{10\times d})$;
end if
for $m = 0, 1, \cdots d$ do
    for $c = 0, 1, \cdots n$ do
        $\boldsymbol{X}^c_{aug}[m] \leftarrow \boldsymbol{X}^c_{al}[m] \times \boldsymbol{J} \times \boldsymbol{U}^c[m] \times \mathscr{W}(\boldsymbol{X}^c_{al}[m])$;
    end for
end for
return $\boldsymbol{X}^c_{aug}[m]$
```

Algorithm 4 presents the procedure of only one augmentation process. This process can be performed for multiple times in order to get a proper amount of data. In conclusion, this augmentation technique assures that we can synthesize much more behavior descriptions that can be used to simulate the user's action in different scenarios without losing or overwhelming the raw information contained in some channel whose original value is small.

VII. Hybrid Workflow and Authentication

To implement an implicit authentication system, one of the fundamental problems is to acquire a proper description of the user. However, to implement this through a conventional approach needs domain knowledge and is time-consuming. Since we are dealing with privacy-sensitive data on mobile devices, the balance between energy, convenience, privacy also need to be carefully considered. In this situation, we propose a hybrid workflow that is based on transfer learning, so that

[4] tpX represents the X axis of the touchpoint, baro represents barometer, gyrX represents the X component of the gyroscope, and accX represents the X component of the accelerometer.

user activity representation learning can be integrated with a lightweight classical authentication algorithm.

A. The Representation Learner

Most existing works, e.g., [12], [38], [39], required to collect data of both the device owner and intruder(s) in advance, for training the multi-class classifier. This requirement diminishes the value of authentication in real world applications. Some recent studies [13], [40], [41] applied conventional binary classifiers to address this problem. But this approach relies heavily on manually extracted features from the raw data. Conventional algorithms also have their own drawbacks when compared to deep learning, though the complex and heavy computational tasks of deep learning make the training process of deep learning models unsuitable for local execution, especially when the terminal is a smartphone. Remote execution, however, may hurt the security and privacy of user data.

To deal with this dilemma, i.e., *Challenge 3*, we propose a hybrid model with a dedicated workflow. Instead of putting all the authentication on a traditional binary classifier or on a deep learning based model, we integrate transfer learning and representation learning together with traditional machine learning techniques, so as to partially offload the computational task from the terminal while addressing privacy concerns, and taking advantage of both the conventional and light machine learning algorithms and deep learning models. The structure of our proposed hybrid model is depicted in Fig. 11.

First, the representation learner is used to bridge the raw user data and the cascaded classifier algorithm. It takes the collected data and outputs the corresponding refined user representations. With the help of user representation, we can portray the collected user's behavior information with a unifying concept, and transform the diversified data collected from a variety of sensors under this concept, optimizing the feature extraction process and authentication performance automatically.

To implement this vision, we propose a timestep-depthwise-convolution-based structure. We first categorized the input data as screen-related information, accelerometer-related information and gyroscope-related information. Then each one of these groups will first go through the timestep process unit, in which the data are processed by a time attention module and a series of 1D convolution on timesteps. Then the output from the three categories are concatenated, so that the consolidated data can further be processed in depthwise. The dimension of the consolidated data is first reduced by a bottleneck structure to reduce the parameter amount, then the data are processed by different attention and 1D convolutional module sets in depthwise. The outputs are added pointwisely to form a compressed channel information, and then it is again added by the concatenated timestep information. This process is expected to gather useful features not only from the view of timesteps, but also in a depthwise manner.

The output of this model is set as multi-categorized with softmax as the activation function, and a categorical cross-entropy as the loss function. This setting is only used during the model training process: once the parameters are ready, the several top layers are removed (denoted by $\mathscr{R}$), and the output from the remaining layers are presented as the user representation, which can be used by the authentication module.

B. The Hybrid Workflow and the Implementation of Authentication

After the generation of user representation, a lightweight classical authentication module is then cascaded to further process the information and perform identity recognition. With the help of binary classifiers, we convert the classification to an anomaly detection task, avoiding the need of requiring imposter's data in advance. The chosen algorithm is the One-Class Support Vector Machine (OCSVM). It builds the classification model based on the existing legitimate records in order to classify new instances as either legitimate or abnormal.

Then we design a customized hybrid workflow to balance of the energy, convenience, and privacy. The workflow is shown as Algorithm 5 and 6. The action of removing several top layers of a model is denoted by $\mathscr{R}$. With this approach, we are able to enhance data security and privacy to the possible highest level, without the need to worry about computational costs or privacy compromises.

Algorithm 5 Pretraining of the representation learner.

```
Input: (from volunteers) augmented sensor data, $\boldsymbol{X}_{aug}$;
Output: pretrained representation learner, $\mathscr{M}_{pre}$;
                                  ▷ This process happens pre-shipment.
while $\mathscr{M}_{pre}$ not converged do
    Train $\mathscr{M}_{pre}$ with $\boldsymbol{X}_{aug}$ on servers;
end while
return $\mathscr{M}_{pre}$
```

Algorithm 6 The authentication workflow on terminal device.

```
Input: (from owner) augmented sensor data, $\boldsymbol{X}_{aug}$; (from
    undetermined user in an unlocking attempt) aligned
    sensor data, $\boldsymbol{X}_{al}$; pretrained representation learner, $\mathscr{M}_{pre}$;
Output: representation extractor, $\mathscr{M}_{ex}$; user representation,
    $\boldsymbol{X}_{re}$; trained OCSVM model, $\mathscr{M}_{au}$; user identity, $L$;
                                  ▷ When the registration is launched.
while $iteration < threshold$ do
    Fine-tune $\mathscr{M}_{pre}$ with $\boldsymbol{X}_{aug}$ locally by a few iterations;
end while
$\mathscr{M}_{ex} \leftarrow \mathscr{R}(\mathscr{M}_{pre})$;
$\boldsymbol{X}_{re} \leftarrow \mathscr{M}_{ex}(\boldsymbol{X}_{aug})$;
Train $\mathscr{M}_{au}$ with $\boldsymbol{X}_{re}$ locally;
                                  ▷ When an unlocking attempt is launched.
$\boldsymbol{X}_{re} \leftarrow \mathscr{M}_{ex}(\boldsymbol{X}_{al})$;
$L \leftarrow \mathscr{M}_{au}(\boldsymbol{X}_{re})$;
return $L$
```

VIII. Evaluation

A. Experiment Settings

Patterns. The pattern unlocking tech was first introduced by Google [44] to enhance the security level of Android

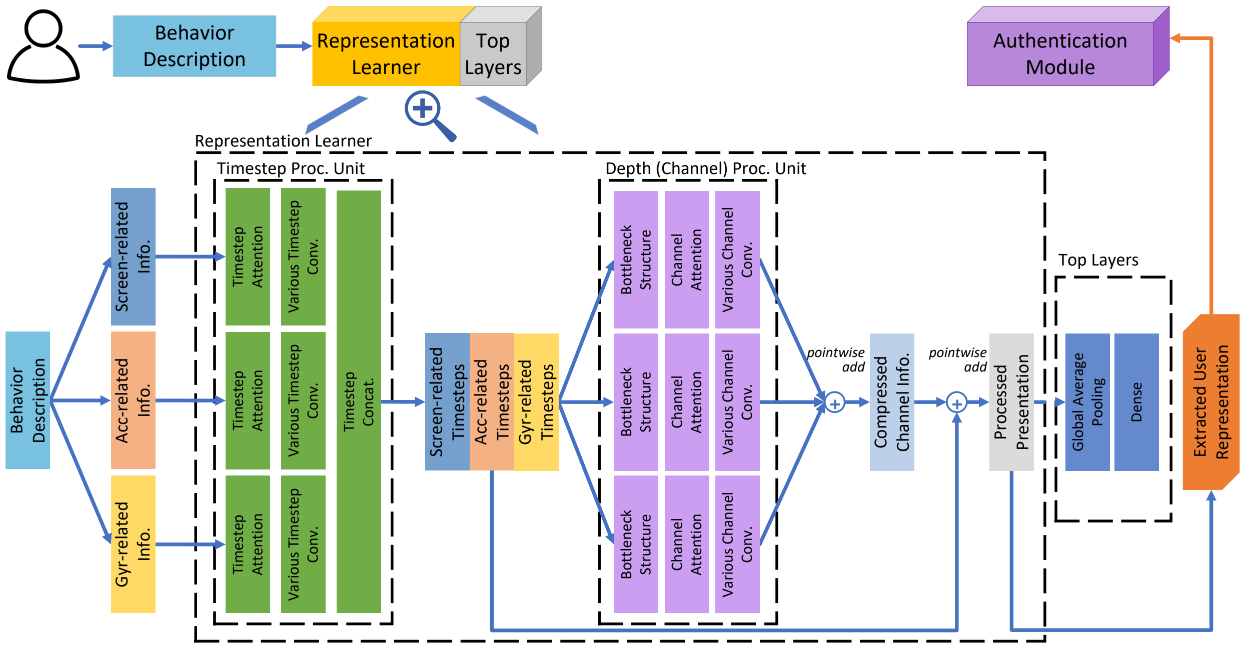


Fig. 11: The structure of the hybrid model.

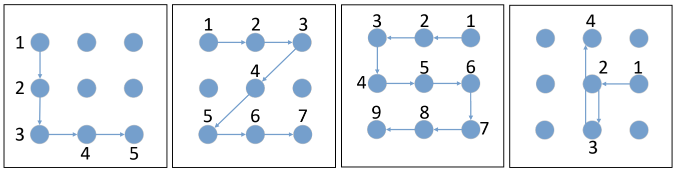


Fig. 12: The patterns that are used in our experiments.

TABLE IV: Description of the Participants

Specification	Property	Quantity
Gender	Male Female	14 9
Age	20-22 23-25 25-27	3 17 3
Interact and grip the phone with	The same hand Differnet hands	8 15
Acquisition time	7:00 - 11:00 11:00 - 15:00 15:00 - 19:00 19:00 - 23:00	7 5 6 5
On-site altitude[5]	50m - 100m 100m - 200m 200m - 450m	12 7 4

smartphones in 2008, and recent studies [41] have proved that the following perspectives on pattern unlocking.

1) The longer of the unlocking pattern, the higher the authentication accuracy.

2) Much information about the behavior features is stored in the turning corners of an unlocking pattern, while the line segments barely contribute to authentication.

Based on these analyses, we designed four designated unlocking patterns for the experiments in this section, which are illustrated in Fig. 12.

Performance Metrics. The following three metrics are used to evaluate the performance of *PresSafe*.

- False Acceptance Rate (FAR) reveals the percentage of instances in which illegal users are incorrectly accepted by the authentication system.
- False Rejection Rate (FRR) reveals the percentage of instances in which requests from the actual device owners are falsely declined.
- Equal Error Rate (EER) is defined as the point on the receiver operating characteristic curve where FAR equals FRR. The lower the EER, the more effective the system.

Dataset. To simulate implicit authentication process, we deployed a dedicated Android application on a Galaxy Note 10+ to collect behavior descriptions from 23 users. Demographics and other properties of the participants are shown in Table IV. All participants use their smartphones as an everyday necessity, and more than three quarters of them are now using or once used pattern unlocking to secure data.

During data collection, participants are asked to draw designated unlocking patterns on a Galaxy Note 10+. For each pattern, they repeat the drawing for no less than 35 times to form the dataset. The way to grip the smartphone is not specified, and they are recommended to use their most favorite gesture for drawing. In the end, a total of 77,022 behavior description samples are collected from 812 unlocking attempts for each unlocking pattern. After the process of adaptive data augmentation, the amount of behavior description data samples has been increased by 175 times, to 13,478,850.

Settings. We split the full dataset into a training set and a validation set with a ratio of 6:4. Then we duplicate the behavior description in the training set, and remove the data from one random user. The rest can be seen as the data from volunteers, as depicted in Fig. 7, which is used to pre-train the

[5] *On-site altitude refers to the altitude of the place of the corresponding data are collected.*

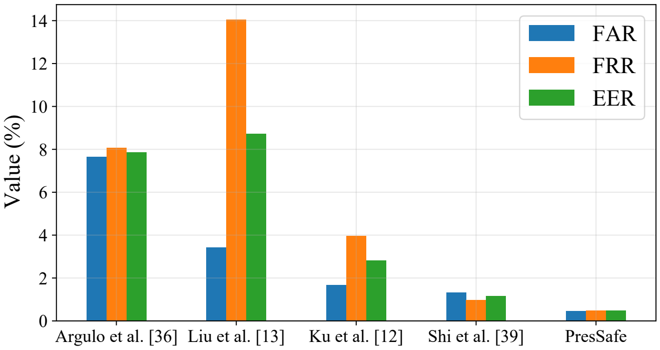


Fig. 13: Performance comparison results.

representation learner after the processing techniques proposed in this paper. For the training part, we also process the full training set using the fusion and channel-adaptive techniques, and then utilize these data to fine-tune the representation learner. Once the representation learner is ready, its top layers are then removed as in Fig. 11, and the remaining structure generates the corresponding user representation, which is then taken by the cascaded authentication module for identity recognition.

To test the performance of our proposed solution, the validation set is sent to the representation learner directly to produce user representation, and then the authentication module will determine the identity of the corresponding user. Note that the authentication module is an anomaly detector, and it only takes and generates two kinds of labels, i.e., the positive one (1), which is taken as the legitimate user, and the negative one (-1), which is taken as intruders. This way, proper measurements to deal with the user labels in the dataset are necessary. For each round while producing the dataset for pre-training the representation learner, the randomly removed user is marked as the legitimate user, and the rest as intruders. While being sent to the authentication module, the user labels in the relevant dataset are first converted into the binary form, so that the authentication module can use such data. To get a more representative result, we enumerate this process, and the final output score is the mean value to avoid outliers.

B. Overall Performance Comparison

Several pattern-unlcoking-based implicit authentication schemes are chosen to conduct performance comparison. We present the details of existing approaches in Table V, including the data source, classifier, and the reproduced performance. A clear view of the performance comparison is in Fig. 13.

TABLE VI: On-screen Pressure Sensing Technique Comparison

Pressure Resource	EER	EER Comparison
Barometer-assisted	1.80 %	**+ 0.00 %**
API approach	3.78 %	**+ 109.44 %**

Due to limited deployed sensors and data processing methods, the performance of [13], [38] is poorer. The EER of these two approaches are both higher than 7.5 %, indicating that they are not the ideal choice in terms of authentication. The introduction of gyroscope and magnetometer in [12] helps to reduce EER, but the training process in this work involve both the owner and intruder data. This presumption is not reasonable and obviously restricts its application in the real-world. Ref. [41] leverages a content-aware module and a pattern segmentation process, and reduces EER to 1.15 %. But the pressure sensing technique in Ref. [41] relies on software simulation and can hardly represent the actual situation. The best result in this comparison is achieved by our solution, *PresSafe*. Although the ambient sensors in our work are more limited when compared with the existing solutions, PresSafe achieves an EER of 0.47 %, which is much lower than that of the baseline schemes. The performance comparison proves that with the help of the barometer assisted pressure sensing technique and a series of related approaches, we can elevate the authentication performance to a higher level.

C. Ablation Study of On-screen Pressure Sensing Technique

To compare the performance of our barometric solution and the API method, we conduct an ablation study. In this experiment, the authentication is performed, only using the on-screen pressure profile, which is captured by the barometer or the API separately.

Table VI presents the results of the experiment, where increased value is marked as positive(+). Note that lower EER indicates a better performance. Compared with our proposed on-screen pressure sensing technique, EER of the conventional API approach is 109.44 % higher than that of the *PresSafe*. The *PresSafe* still achieves an EER of 1.80 % even when only relying on the barometer-based on-screen pressure. This result validates the barometer-based on-screen pressure profile, and supports our finding in previous empirical experiments. Based on the experiment results, we conclude that the barometer-assisted on-screen pressure profile sensing technique does enhance the overall authentication performance.

D. Validation of Sensor Selection

Implicit authentication heavily relies on the user's behavior data to make decisions. To generate proper behavior descriptions, we deployed a set of ambient sensors so that the user's actions during the pattern unlocking process can be recorded at different dimensions. We validate those features involved in our system by discarding each specific feature.

The combinations are set as follows:

[6] Acc - Accelerometer, Baro - Barometer, Gyr - Gyroscope, Mag - Magnetometer, RV - Rotation Vector Sensor, Touch - Touch screen related information.

[7] Data needed for user to train the authentication module.

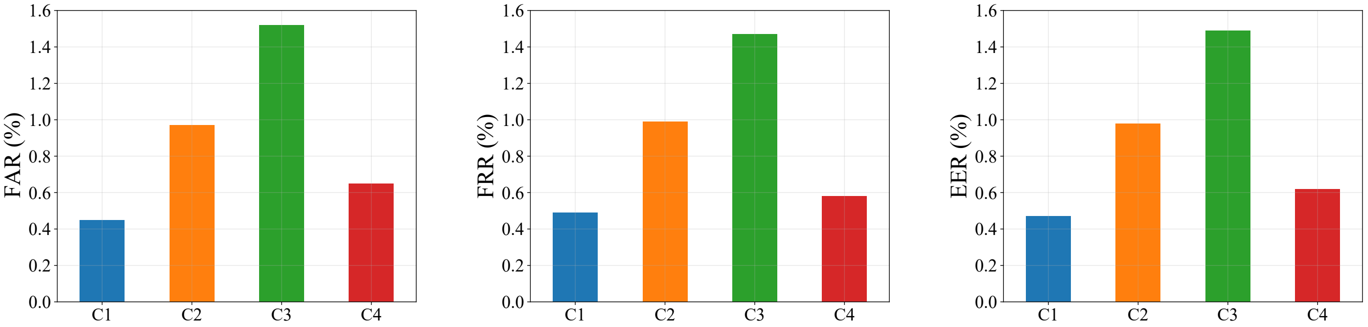
[8] The classifier used for authentication. Some work may involve similar algorithms for other purposes.

[9] The FAR, FRR and EER of [12], [13], [38], [41] are reproduced results, which are based on the dataset collected by us.

[10] The presented FAR, FRR and EER in this table are the mean value of the results on four preset patterns (Fig. 12) in this experiment.

TABLE V: Performance Comparison Results

Approach	Sensors Involved[6]	Source of Data[7]	Classifier[8]	FAR	FRR	EER[9][10]
Angulo et al. [38]	Touch	Intruder & Device Owner	Random Forest	7.64 %	8.08 %	7.86 %
Liu et al. [13]	Touch, Acc	Device Owner	Statistical Classifier	3.42 %	14.05 %	8.73 %
Ku et al. [12]	Touch, Acc, Gyr, Mag	Intruder & Device Owner	Gaussian Nave Bayes	1.68 %	3.95 %	2.82 %
Shi et al. [41]	Touch, Acc, Gyr, RV	Device Owner	One-Class Support Vector Machine	1.32 %	0.98 %	1.15 %
PresSafe	**Touch, Acc, Baro, Gyr**	**Device Owner**	**Hybrid Model**	**0.45 %**	**0.49 %**	**0.47 %**



(a) FAR

(b) FRR

(c) EER

Fig. 14: Validation of feature combination, sub-metrics. C1 - C4 represent feature combination 1 - combination 4.

1) Combination 1 (baseline): barometer, gyroscope, accelerometer. This combination is the baseline that is supposed to present the overall performance of our work.

2) Combination 2: gyroscope, accelerometer.

3) Combination 3: barometer, gyroscope.

4) Combination 4: barometer, accelerometer.

The above readings are sampled from the built-in sensors of the smartphone used for experiment. All readings are fetched under the *SENSOR_DELAY_GAME* mode, in our case[11]the sampling runs under a delay of 20-25 ms. Readings from the barometer, accelerometer, and gyroscope are floating-point values, and the readings from accelerometer is an array, which is consisted of three channels, namely the accX, accY, accZ. Similarly, the readings from gyroscope are also consisted of three axes. The results are displayed in Fig. 14. Not surprisingly, *Combination 1* reached the best performance among all the presets. We also observe that:

First, a sensor set without readings from barometer results in a drastic deterioration in authentication performance on pattern L and T. This is mainly because the removal of barometer readings in the behavior description results in the loss of sliding pressure information. The comparison proves the importance of the pressure sensing technique in our solution.

Second, removing accelerometer or gyroscope individually causes the loss of recordings of the tile and movement status. We can observe a drop in the overall performance.

Third, the performance in *Combination 2* can still match the comparison studies. We believe this performance proves the feasibility of our workflow, since the feature extractor generates proper user representations from raw data.

TABLE VII: Binary Classifier Comparison

Classifier	FAR	FRR	EER[12]	EER Comparison [13]
OCSVM	0.45 %	0.49 %	**0.47 %**	**+ 0.00 %**
IF	2.35 %	2.58 %	**2.47 %**	**+ 425.53 %**
LOF	0.94 %	1.79 %	**1.37 %**	**+ 191.49 %**

E. Authentication Module Comparison

We conducted a comparison between three most commonly used binary classifiers, namely the One-Class Support Vector Machine (OCSVM), Isolation Forest (IF), and Local Outlier Factor (LOF). These three binary classifiers can function as the authentication module mainly because their training does not involve data from the intruders, thus meeting our workflow's requirement.

Table VII shows the results of this comparison. From the table, it is clear that OCSVM outperforms the other two classifiers, which is consistent with the finding in related works [40], [41]. Based on this observation, we choose OCSVM as the authentication module in the *PresSafe*.

F. Ablation Study on Channel-adaptive Data Process Module

From our earlier observation, the data augmentation process is supposed to enhance the generalization and overall performance of this system by introducing jitter-added signals to

[11] Even set to the same delay method, the sampling rate is still device-specific.

[12] The presented FAR, FRR and EER in this table are the mean value of the results on four preset patterns (Fig. 12) in this experiment.

[13] Increase in value is marked as positive(+). However, please note that the lower EER, the better performance.

TABLE VIII: Validation of Data Augmentation

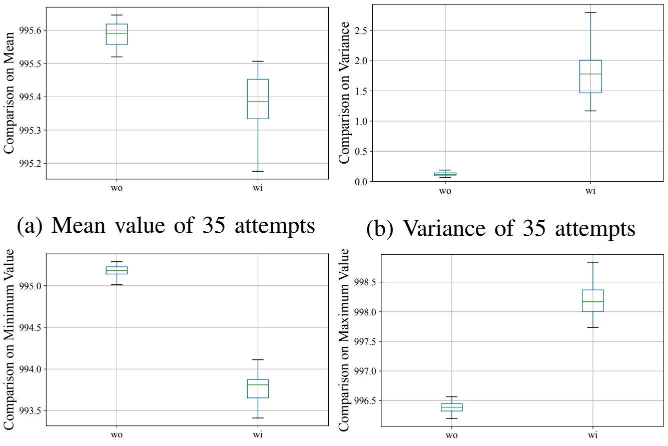
Data Augmentation	FAR	FRR	EER
W/	0.45 %	0.49 %	**0.47 %**
W/o	2.86 %	3.09 %	**2.97 %**

enhance behavior descriptions. Further, considering the properties of different channels, signals to be added will be scaled with a scaling factor, thus being channel adaptive. To validate this process, we conduct a performance comparison with or without data augmentation. Fig. 15 is the sub-metric figure illustrating the performance, while the values are presented in Table VIII. The presented FAR, FRR and EER here are the mean value of the results on four preset patterns (Fig. 12) in this experiment.

From the results, it is clear that data augmentation helps to increase the authentication performance. After the augmentation scheme, FAR is decreased by 84.27 %, FRR by 84.14 %, and EER by 84.18 %, which indicates that the data augmentation process enhances the system's overall performance at a considerable scale.

G. Generalizability of PresSafe

To verify the generalizability of our proposed solution, we further conduct extensive experiments using different phone models or in different situations. First, we design a comparison study to verify if wearing gloves can cause any impacts on *PresSafe*. The volunteers wear wool gloves with regular thickness, and repeat the data collection application for 35 times on the same Galaxy Note 10+, which is used in most of the experiments throughout this paper. An analysis of the collected data is presented in Fig. 16.



(a) Mean value of 35 attempts (b) Variance of 35 attempts

(c) Minimum value of 35 attempts(d) Maximum value of 35 attempts

Fig. 16: Box-plot on comparison of the collected data with or without gloves.

Wearing gloves causes a larger contact area on the screen, thus reducing the mean value of the recordings since the force is distracted. The variance of the four properties on glove-reading data appears more significant. Range of the glove-wearing data is also larger. The observations suggest that the glove-wearing unlocking behavior is a relatively more unsteady, dynamic process.

Despite the above findings, we notice that the pressure sensing mechanism still functions well. To explicitly present the data trend, we use a scaler to process the raw data. The scaler operation follows Eq. 1 to convert the data column-wisely, and the processed data are presented in Fig. 17.

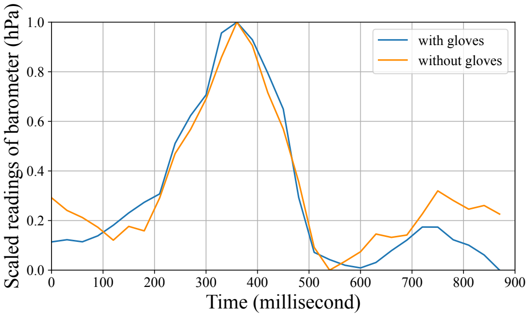


Fig. 17: Processed barometer readings using the scaler.

It can be seen that the normalized data demonstrate a similar trend compared to the case of without gloves. The rationale behind the barometer-aided on-screen pressure sensing still holds true, no matter what the use-case scenarios is. Furthermore, solutions to properly mitigate the effects of varied scenarios have already been discussed in [41]. Through our experiment and analysis, we conclude that, wearing gloves does not introduce an obvious negative impact on our proposed authentication approach *PresSafe*.

Second, we use three other types of phones, i.e., Galaxy S8, Galaxy S21, and Xperia 1, to test if the relationship between the barometric readings and the press events holds true for different types of smartphones. Fig. 18 demonstrates that, the barometric relationship underpinning our proposed approach, namely *PresSafe*, does holds for these three phones. Albeit the reactions are not exactly the same, we can still infer press events from the barometric readings, and further construct the implicit authentication process.

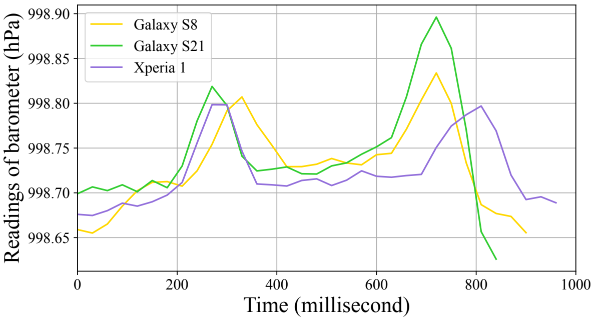
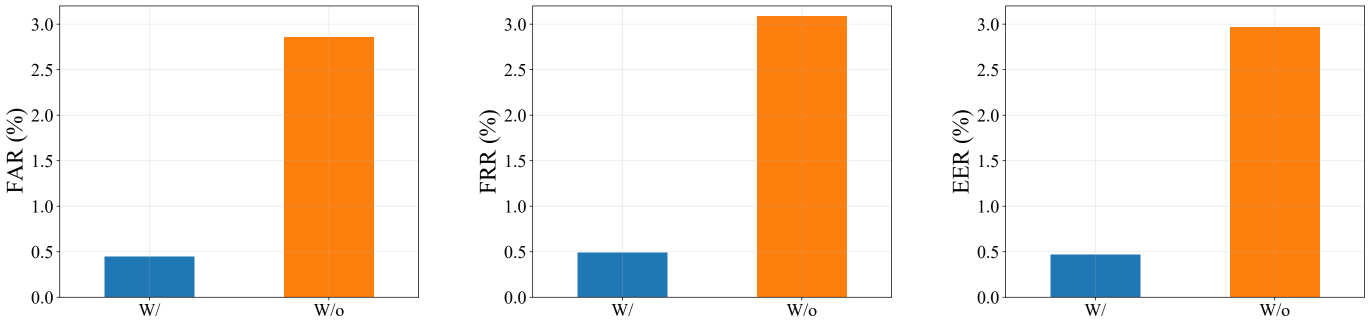


Fig. 18: Barometric readings from different types of phones.

Considering the fact that the above phones are all the Ingress Protection type, i.e., these phones are waterproof models, we further verify if the previously discussed relationship can also be observed with non-waterproof models. To do this, we remove the SIM card tray from the Galaxy Note 10+ and perform the similar data enrollment process. It can be seen



(a) FAR

(b) FRR

(c) EER

Fig. 15: Validation of data augmentation, sub-metrics.

that, removing SIM card tray results in disintegration of the sealed chassis, so the phone is not airtight anymore. Fig. 19 shows that barometer readings can no longer reflect the user's pressing and sliding actions when the phone is not airtight. Based on the experiment, it is safe to conclude that, for the models that are not built airtight, the relationship between user's actions and the barometric readings can hardly exist.

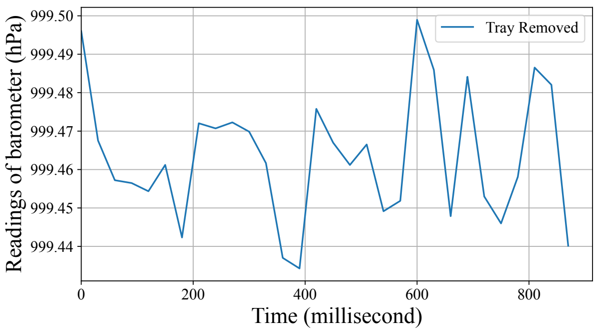


Fig. 19: Barometric readings with the SIM card tray removed.

IX. Conclusion

Mobile devices today hold unprecedented amounts of sensitive information, urging heightened security with advanced authentication approaches while balancing effectiveness and convenience/user experience. The on-screen pressure sensing commonly used in the majority of existing solutions lacks accuracy as it relies on an approximate system API which has been shown inadequate. In this paper, we proposed *PresSafe* to utilize barometer readings to better model and measure on-screen pressure portrait. We presented a set of experiments to validate our approach and quantify its advantages, utilizing 70,000 records collected from 23 volunteers in 6 different locations. Evaluation results showed that *PresSafe* achieved an FAR of 0.45 %, an FRR of 0.49 %, and an EER of 0.47 %, which represented a superior performance when compared with several existing solutions.

Our work can be extended in several ways. First, we could adopt additional approaches including Generative Adversarial Networks and other models for data augmentation. This will be examined in our future work. Also, the devices used for validating our approach are relatively limited in capabilities and resources. In future work we will include a broader range of commercial smartphones from budget to Pro models. Third, the features of user's behavior, specifically the sliding and press inclination of the fingers, usually remain stable and constant. However, we admit that for a comparatively long period of time, i.e., over the years, there remains possibilities that some specific events may cause this behavior to vary. In this case, we could add some mechanisms to slowly adapt the representation learner and the authentication module on regular basis.

References

[1] Z. Zhou, D. Tang, X. Wang, W. Han, X. Liu, and K. Zhang, "Invisible mask: Practical attacks on face recognition with infrared," Mar. 2018.

[2] A. S. Rathore, W. Zhu, A. Daiyan, C. Xu, K. Wang, F. Lin, K. Ren, and W. Xu, "Sonicprint: A generally adoptable and secure fingerprint biometrics in smart devices," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys 20. New York, NY, USA: Association for Computing Machinery, Jun. 2020, p. 121134. [Online]. Available: https://doi.org/10.1145/3386901.3388939

[3] I. Echizen and T. Ogane, "Biometricjammer: Use of pseudo fingerprint to prevent fingerprint extraction from camera images without inconveniencing users," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2018, pp. 2825–2831.

[4] I. Goicoechea-Telleria, A. Garcia-Peral, A. Husseis, and R. Sanchez-Reillo, "Presentation attack detection evaluation on mobile devices: Simplest approach for capturing and lifting a latent fingerprint," in *2018 International Carnahan Conference on Security Technology (ICCST)*, Oct. 2018, pp. 1–5.

[5] G. Ye, Z. Tang, D. Fang, X. Chen, W. Wolff, A. J. Aviv, and Z. Wang, "A video-based attack for android pattern lock," *ACM Trans. Priv. Secur.*, vol. 21, no. 4, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3230740

[6] M. Zhou, Q. Wang, J. Yang, Q. Li, F. Xiao, Z. Wang, and X. Chen, "Patternlistener: Cracking android pattern lock using acoustic signals," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS 18. New York, NY, USA: Association for Computing Machinery, Oct. 2018, p. 17751787. [Online]. Available: https://doi.org/10.1145/3243734.3243777

[7] H. Khan, U. Hengartner, and D. Vogel, "Targeted mimicry attacks on touch input based implicit authentication schemes," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys 16. New York, NY, USA: Association for Computing Machinery, Jun. 2016, p. 387398. [Online]. Available: https://doi.org/10.1145/2906388.2906404

[8] Y. Li, H. Hu, and G. Zhou, "Using data augmentation in continuous authentication on smartphones," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 628–640, Jun. 2019.

[9] A. Mahfouz, T. M. Mahmoud, and A. S. Eldin, "A survey on behavioral biometric authentication on smartphones," *Journal of Information Security & Applications*, vol. 37, no. dec., pp. 28–37, Oct. 2017.

[10] S. M. Ganesh, P. Vijayakumar, and L. J. Deborah, "A secure gesture based authentication scheme to unlock the smartphones," in *2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*, Feb. 2017, pp. 153–158.

[11] D. Izumoto and Y. Yamazaki, "Security enhancement for touch panel based user authentication on smartphones," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Mar. 2019, pp. 218–223.

[12] Y. Ku, L. H. Park, S. Shin, and T. Kwon, "Draw it as shown: Behavioral pattern lock for mobile user authentication," *IEEE Access*, vol. 7, pp. 69 363–69 378, May 2019.

[13] C. L. Liu, C. J. Tsai, T. Y. Chang, W. J. Tsai, and P. K. Zhong, "Implementing multiple biometric features for a recall-based graphical keystroke dynamics authentication system on a smart phone," *Journal of Network and Computer Applications*, vol. 53, pp. 128–139, Mar. 2015.

[14] W. Meng, W. Li, D. S. Wong, and J. Zhou, "Tmguard: A touch movement-based security mechanism for screen unlock patterns on smartphones," in *Applied Cryptography and Network Security*, M. Manulis, A.-R. Sadeghi, and S. Schneider, Eds., vol. 9696. Cham: Springer International Publishing, Jun. 2016, pp. 629–647.

[15] M. W. Abo El-Soud, T. Gaber, F. AlFayez, and M. M. Eltoukhy, "Implicit authentication method for smartphone users based on rank aggregation and random forest," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 273–283, Feb. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110016820303902

[16] A. O'Hara. (2019, Oct.) Editorial: Apple's removal of 3d touch is a backwards step for 'pro' iphones. [Online]. Available: https://appleinsider.com/articles/19/10/02/editorial-apples-removal-of-3d-touch-is-a-backwards-step-for-pro-iphones

[17] H. Khan, U. Hengartner, and D. Vogel, *Evaluating Attack and Defense Strategies for Smartphone PIN Shoulder Surfing*. New York, NY, USA: Association for Computing Machinery, Apr. 2018, p. 110. [Online]. Available: https://doi.org/10.1145/3173574.3173738

[18] M. Lee, "Security notions and advanced method for human shoulder-surfing resistant pin-entry," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 695–708, April 2014.

[19] S. Cha, S. Kwag, H. Kim, and J. H. Huh, "Boosting the guessing attack performance on android lock patterns with smudge attacks," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS 17. New York, NY, USA: Association for Computing Machinery, Apr. 2017, p. 313326. [Online]. Available: https://doi.org/10.1145/3052973.3052989

[20] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the 4th USENIX Conference on Offensive Technologies*, ser. WOOT'10. USA: USENIX Association, Aug. 2010, p. 17.

[21] H. Lee, S. Kim, and T. Kwon, "Here is your fingerprint! actual risk versus user perception of latent fingerprints and smudges remaining on smartphones," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, ser. ACSAC 2017. New York, NY, USA: Association for Computing Machinery, Dec. 2017, p. 512527. [Online]. Available: https://doi.org/10.1145/3134600.3134643

[22] Y. Abdelrahman, M. Khamis, S. Schneegass, and F. Alt, "Stay cool! understanding thermal attacks on mobile-based user authentication," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. New York, NY, USA: Association for Computing Machinery, May 2017, p. 37513763. [Online]. Available: https://doi.org/10.1145/3025453.3025461

[23] L. Zhang, S. Tan, J. Yang, and Y. Chen, "Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2016, pp. 1080–1091.

[24] M. R. Zafar and M. Ali Shah, "Fingerprint authentication and security risks in smart devices," in *2016 22nd International Conference on Automation and Computing (ICAC)*. IEEE, Sep. 2016, pp. 548–553.

[25] S. Hosseini, "Fingerprint vulnerability: A survey," in *2018 4th International Conference on Web Research (ICWR)*, Apr. 2018, pp. 70–77.

[26] J. Tan, X. Wang, C.-T. Nguyen, and Y. Shi, "Silentkey: A new authentication framework through ultrasonic-based lip reading," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, Mar. 2018. [Online]. Available: https://doi.org/10.1145/3191768

[27] Z. Yan, Q. Song, R. Tan, Y. Li, and A. W. K. Kong, "Towards touch-to-access device authentication using induced body electric potentials," in *The 25th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '19. New York, NY, USA: Association for Computing Machinery, Aug. 2019. [Online]. Available: https://doi.org/10.1145/3300061.3300118

[28] L. Wang, K. Huang, K. Sun, W. Wang, C. Tian, L. Xie, and Q. Gu, "Unlock with your heart: Heartbeat-based authentication on commercial mobile phones," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 3, Sep. 2018. [Online]. Available: https://doi.org/10.1145/3264950

[29] C. Yang, J. Zhang, J. Guo, Y. Zheng, L. Yang, and J. Ma, "Fingerprint protected password authentication protocol," *Security and Communication Networks*, vol. 2019, 2019.

[30] L. Yang, Y. Cheng, X. Wang, and Q. Liu, "Online handwritten signature verification using feature weighting algorithm relief," *Soft Computing*, vol. 22, no. 23, pp. 7811–7823, 2018.

[31] R. Mayrhofer and S. Sigg, "Adversary models for mobile device authentication," *ACM Comput. Surv.*, vol. 54, no. 9, Oct. 2021. [Online]. Available: https://doi.org/10.1145/3477601

[32] Q. Jiang, J. Ni, J. Ma, L. Yang, and X. Shen, "Integrated authentication and key agreement framework for vehicular cloud computing," *IEEE Network*, vol. 32, no. 3, pp. 28–35, 2018.

[33] C. Feng, J. Xiong, L. Chang, F. Wang, J. Wang, and D. Fang, "Rf-identity: Non-intrusive person identification based on commodity rfid devices," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 1, Mar. 2021. [Online]. Available: https://doi.org/10.1145/3448101

[34] Y. Gao, Y. Jin, J. Chauhan, S. Choi, J. Li, and Z. Jin, "Voice in ear: Spoofing-resistant and passphrase-independent body sound authentication," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 1, Mar. 2021. [Online]. Available: https://doi.org/10.1145/3448113

[35] Z. Wang, S. Tan, L. Zhang, Y. Ren, Z. Wang, and J. Yang, "Eardynamic: An ear canal deformation based continuous user authentication using in-ear wearables," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 1, Mar. 2021. [Online]. Available: https://doi.org/10.1145/3448098

[36] P. Andriotis, G. Oikonomou, A. Mylonas, and T. Tryfonas, "A study on usability and security features of the android pattern lock screen," *Information & Computer Security*, vol. 24, no. 1, pp. 53–72, Mar. 2016.

[37] H. Xu, Y. Zhou, and M. R. Lyu, "Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones," in *10th Symposium On Usable Privacy and Security (SOUPS 2014)*. Menlo Park, CA: USENIX Association, Jul. 2014, pp. 187–198. [Online]. Available: https://www.usenix.org/conference/soups2014/proceedings/presentation/xu

[38] J. Angulo and E. Wästlund, "Exploring touch-screen biometrics for user identification on smart phones," in *Privacy and Identity Management for Life*, J. Camenisch, B. Crispo, S. Fischer-Hübner, R. Leenes, and G. Russello, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 130–143.

[39] O. Alpar, "Intelligent biometric pattern password authentication systems for touchscreens," *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6286–6294, Oct. 2015.

[40] R. Wang and D. Tao, "Context-aware implicit authentication of smartphone users based on multi-sensor behavior," *IEEE Access*, vol. 7, pp. 119 654–119 667, Aug. 2019.

[41] D. Shi, D. Tao, J. Wang, M. Yao, Z. Wang, H. Chen, and S. Helal, "Fine-grained and context-aware behavioral biometrics for pattern lock on smartphones," *Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies (IMWUT)*, vol. 5, no. 1, pp. 1–30, Mar. 2021.

[42] AOSP. (2020, Sep.) Touch devices : Android open source project. Android Open Source Project. [Online]. Available: https://source.android.com/devices/input/touch-devices

[43] S. Aquino. (2020, Sep.) Apple's removal of force touch in watchos 7 makes apple watch a less accessible device. [Online]. Available: https://www.forbes.com/sites/stevenaquino/2020/09/29/apples-removal-of-force-touch-in-watchos-7-makes-apple-watch-a-less-accessible-device

[44] D. Johansson, J. T. Andersson, S. M. Thorsander, and E. Tseng, "Pattern-based mobile device unlocking," Dec. 2014, a United States Patent 8,904,479.