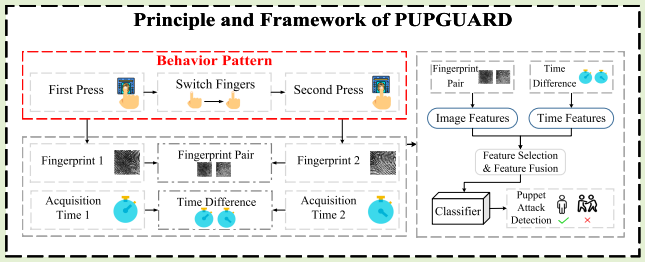
Two-Factor Authentication Approach Based on Behavior Patterns for Defeating Puppet Attacks

Wenhao Wang, Guyue Li, *Member, IEEE*, Zhiming Chu, Haobo Li, *Member, IEEE*, and Daniele Faccio

***Abstract*—Fingerprint traits are widely recognized for their unique qualities and security benefits. Despite their extensive use, fingerprint features can be vulnerable to puppet attacks, where attackers manipulate a reluctant but genuine user into completing the authentication process. Defending against such attacks is challenging due to the coexistence of a legitimate identity and an illegitimate intent. Research on countering puppet attacks is limited, and existing studies are hindered by the requirement of guiding users through the authentication process manually. In this article, we propose PUPGUARD, a solution designed to guard against puppet attacks. This method is based on user behavioral patterns, specifically, the user needs to press the capture device twice successively with different fingers during the authentication process. PUPGUARD leverages both the image features of fingerprints and the timing characteristics of the pressing intervals to establish two-factor authentication. More specifically, after extracting image features and timing characteristics, and performing feature selection on the image features, PUPGUARD fuses these two features into a 1-D feature vector, and feeds it into a one-class classifier to obtain the classification result. This two-factor authentication method emphasizes dynamic behavioral patterns during the authentication process, thereby enhancing security against puppet attacks. To assess PUPGUARD's effectiveness, we conducted experiments on datasets collected from 31 subjects, including image features and timing characteristics. Our experimental results demonstrate that PUPGUARD achieves an impressive accuracy rate of 97.87% and a remarkably low false positive rate (FPR) of 1.89%. Furthermore, we conducted comparative experiments to validate the superiority of combining image features and timing characteristics within PUPGUARD for enhancing resistance against puppet attacks.**



***Index Terms*— Behavior patterns, fingerprint, one-class classification, puppet attack detection.**

I. Introduction

FINGERPRINT traits have become increasingly popular in recent years due to their distinctiveness, reliability, universality, and security. When compared to alternative biometric authentication methods, fingerprint authentication stands out with remarkably low rates of false rejection (FRR) and false acceptance (FAR), making it a more secure option than traditional password-based authentication, which can be susceptible to theft or forgetfulness. Despite holding a substantial share of the global market and finding use in various scenarios [1], fingerprint authentication is not without its inherent flaws, including susceptibility to presentation attacks (PAs).

ISO/IEC 30 107 defines PA as "presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system" [2]. Since the proposal of PA, it has garnered considerable attention due to the minimal implementation cost associated with generating artificial fingerprints [3], and the attacker can use many common materials to complete the imitation of the victim's fingerprint, such as silicone [4], plasticine [5], and thermoplastic materials [6]. Both hardware-based and software-based methods have been proposed to improve the ability of biometric systems to resist such attacks.

However, besides detecting fake or altered biometric characteristics, PA also encompasses identifying coercion, non-conformity, and obscuration [7]. Puppet attack is an attack in which an attacker forces a legitimate victim to press a finger against a fingerprint reader for intrusion [8]. Puppet

Manuscript received 18 November 2023; revised 7 January 2024; accepted 14 January 2024. Date of publication 24 January 2024; date of current version 14 March 2024. This work was supported in part by the Natural Science Foundation of Jiangsu Province under Grant BK20211160 and in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/T021020/1 and Grant EP/X5257161/1. The associate editor coordinating the review of this article and approving it for publication was Prof. Xintao Huan. *(Corresponding author: Guyue Li.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the IEC for Clinical Research of Zhongda Hospital Affiliated to Southeast University under Approval No. 2023ZDSYLL109Y01.

Wenhao Wang, Guyue Li, and Zhiming Chu are with the School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: guyuelee@seu.edu.cn).

Haobo Li and Daniele Faccio are with the School of Physics and Astronomy, University of Glasgow, G12 8QQ Glasgow, U.K. (e-mail: haobo.li@glasgow.ac.uk; daniele.faccio@glasgow.ac.uk).

Digital Object Identifier 10.1109/JSEN.2024.3355694

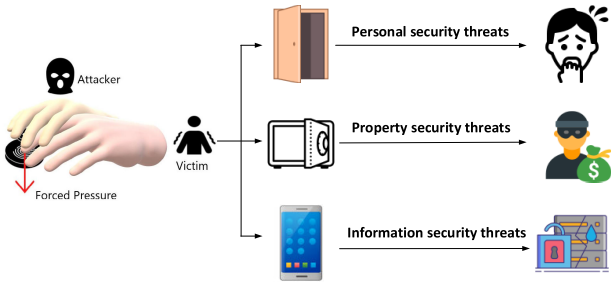


Fig. 1. Possible security risks caused by puppet attacks.

attacks often involve violence, threats, or intimidation, such as an attacker wielding a weapon to force a victim to unlock a vault with a fingerprint lock or a child forcibly pressing a parent's finger to unlock a game console. Failing to defend against puppet attacks can result in substantial financial losses and jeopardize personal safety. Hence, it is imperative to research biometric fingerprint authentication methods that can withstand puppet attacks. The schematic of the puppet attack and the security risks it may cause is shown in Fig. 1.

Unfortunately, research on puppet attacks is not as extensive as that on liveness detection, with the majority of studies in fingerprint PAs primarily concentrating on assessing whether the input fingerprint originates from a real living person or an imitation. These methods are difficult to defend against puppet attacks, because in puppet attacks, although the victim is coerced, the input fingerprint still belongs to a legitimate user. Wu et al. [8] propose the concept of puppet attack and design a detection method based on fingertip touch behavior. However, this method has certain limitations. These include potential FRR due to behavior variability and different postures, as well as the requirement for the user to hand-hold the device, which can result in failure if the device is placed stationary on a desktop.

In this article, we introduce PUPGUARD, a solution designed to defend against puppet attacks. PUPGUARD leverages user behavior patterns, specifically consecutive finger presses on the fingerprint module using different fingers, to capture intrinsic image features and timing characteristics, and subsequently implements two-factor authentication. This behavior-based approach enhances security by requiring two distinct finger presses and introducing a time gap between them, making it tougher for attackers to mimic the authentication process. Unlike traditional fingerprint authentication, which relies solely on static images, PUPGUARD focuses on dynamic behavior patterns during authentication, strengthening overall security against fingerprint PAs. We initially conduct separate preprocessing for both fingerprint images and timing characteristics. Subsequently, we employ local binary pattern (LBP), histogram of oriented gradients (HOG) techniques, and residual network (ResNet) to extract discriminative features from characterized behavioral patterns. Following this, we perform feature selection on image-based features and fuse them with time-based features to create a fused feature vector, which is finally input into a one-class classifier to obtain the classification result.

Based on our investigation, there is currently no publicly available dataset that comprehensively encompasses both image features and timing characteristics required by our PUPGUARD method. Specifically, a fingerprint pair is precisely characterized as two distinct fingerprint images acquired through consecutive double presses of the fingerprint module using different fingers during a single authentication process, serving to represent image features. The corresponding time interval between presses is utilized to represent the timing characteristics. Existing fingerprint datasets may contain unforced and coerced fingerprint images but do not directly facilitate the formation of fingerprint pairs or the generation of datasets encompassing timing attributes of behavior patterns. This limitation arises from the absence of continuous consecutive presses of the fingerprint module with differing fingers in existing datasets, which fails to reflect the characteristics of continuous pressing in behavior patterns. To address this issue, we established a database comprising 496 fingerprint pairs (992 fingerprints) and corresponding time intervals collected from 31 individuals aged between 20 and 85.

To demonstrate the necessity of our database and the superiority of using PUPGUARD, we conducted a large number of experiments. The results showed that PUPGUARD reaches highest accuracy of 97.87% and lowest false positive rate (FPR) of 1.89%, respectively. The experiment using only image features for detection and the one using only timing characteristics proved the necessity of employing both types of features to represent behavior patterns for detecting puppet attacks. Furthermore, we performed experiments involving behavioral patterns where the same finger was used for two consecutive presses to establish the importance of utilizing two different fingers. Subsequently, we conducted experiments that showed improved performance of PUPGUARD with the expansion of the training set.

The contributions of this article are summarized as follows.

1) We propose PUPGUARD, a system that leverages user behavior patterns to capture inherent image features and timing characteristics, thereby implementing a two-factor authentication method. This heightened security approach mandates two separate finger presses with a time gap between them, increasing the difficulty for potential attackers attempting to replicate the authentication process.
2) To assess the performance of PUPGUARD, we assembled a dataset of 496 fingerprint pairs (comprising 992 individual fingerprints) and their associated time intervals from 31 participants spanning ages 20–85. This dataset, obtained with Institutional Review Board (IRB) approval, effectively encapsulates the specified behavioral patterns.
3) A series of comprehensive experiments were carried out to illustrate both the essentiality and effectiveness of PUPGUARD. These experiments encompassed scenarios using solely image features, exclusively timing characteristics, and employing the same finger for both presses. Our experimental findings conclusively indicate that PUPGUARD attains an outstanding accuracy rate of 97.87% while simultaneously achieving the lowest FPR of 1.89%.

The rest of this article is organized as follows. Section II reviews related work on liveness detection and puppet attacks. Section III describes the motivation and principle for our work. In Section IV, the proposed PUPGUARD is explained in detail. The experimental results and detailed analysis are presented in Section V. Limitations of PUPGUARD are discussed in Section VI. Finally, Section VII provides a summary of this article.

II. Related Work

Fingerprint authentication is susceptible to PAs, as skilled individuals with inexpensive hardware and software can easily generate synthetic fingerprints, thereby increasing their chances of successfully executing such attacks [33].

Hardware-based PAD methods necessitate the inclusion of specific sensors within the fingerprint biometric system. These sensors are responsible for verifying the authenticity of signals, such as pulse oximetry [9], blood pressure [10], [11], and odor [12]. By capturing both the fingerprint and one or more of these signals, the biometric system can authenticate the user. Additionally, some hardware-based techniques involve differentiating between the electrical properties [13], [14] of living skin and counterfeit materials, as well as utilizing optical coherence tomography (OCT) [15], [16], [17], [18], [19].

Software-based methods use image processing techniques to extract image features from acquired images, combined with machine learning methods to improve defense against fingerprint spoofing attacks [34]. Specifically, software-based methods can be divided into dynamic and static methods. Dynamic techniques utilize time-varying features that require a sequence of fingerprint images or videos to extract [33]. These features identify the authenticity of fingerprints by detecting the physiological characteristics of the human body. Current mainstream methods include skin distortion-based methods [20], [21], [22] and perspiration-based methods [23], [24], [25], [26]. Unlike dynamic methods, static methods only need one image of the fingerprint. They extract the required features from the image to complete the detection of PA. Methods based on physiological or anatomical features mainly utilize perspiration [35], [36] and sweat pores on the finger surface [27], [28], [29]. Methods based on the surface coarseness [30] of the fingerprint rely on the premise that the surface of the fake fingerprint is rougher [37] to judge the authenticity of the fingerprint. Moreover, texture feature-based methods are widely employed. Coli et al. [31] use high-frequency energy to tell a finger from a fake, because a fake finger does not retain the high-frequency details of a live one. Ghiani et al. [32] proposed a method based on rotation-invariant local phase quantization, which exploits the lack of information during the fabrication of fake fingerprints and extracts the texture features of fingerprint images to reject fake fingerprints.

Unfortunately, most of the existing researches on PAs focus on liveness detection, so it is difficult for these methods to detect puppet attacks. Existing methods of defending against puppet attacks have certain flaws. Wu et al. [34] propose a detection method based on fingertip-touch behavior, but its reliance on a handheld authentication device poses challenges for application in stationary scenarios like door locks or safes where the fingerprint device remains fixed.

Unlike existing literature, our work will leverage user behavior patterns to implement a two-factor authentication method, which can perform identity verification both when the user is holding the authentication device and when the device is stationary, addressing the current research gap in usage scenarios. Table I summarizes the aforementioned methods and their common drawbacks.

III. Principle of PUPGUARD

We represent a legitimate user subject to a puppet attack as a combination of two attributes: the user's genuine identity and an illegitimate state. The simultaneous presence of these attributes complicates the defense against puppet attacks. Effectively countering such attacks hinges on the precise identification and differentiation of these attributes during user authentication. If we treat these attributes as Boolean values and view puppet attack detection as a logical "AND" relationship between them, a user is deemed legitimate only when both properties are true—signifying the possession of a legal identity and legal status.

Conventional fingerprint authentication methods ascertain the user's identity legitimacy by scrutinizing the image captured during a single press of the fingerprint module. However, these methods pose challenges in identifying the stateful attributes of puppet attacks since, even if a user falls victim to a puppet, the captured fingerprint still belongs to the legitimate user. To fortify defenses against puppet attacks, it is imperative to capture user status information utilizing data beyond the fingerprint image during the pressing action.

We are aware that when an individual's state becomes abnormal, it frequently manifests through specific behavioral patterns, such as trembling, stiffness, weakness, or the use of excessive force. In situations where a user is subjected to a puppet attack and compelled to undergo authentication against their will, the victim's response can vary from resistance due to anger, trembling due to fear, to stiffness and powerlessness due to disorientation. Consequently, in PUPGUARD, our emphasis is on analyzing the user's behavioral patterns to extract the state-related characteristics of the authentication process, facilitating the detection of puppet attacks.

Inspired by the keystroke dynamic authentication method [38], [39], [40], we explore alterations in user behavior patterns when subjected to puppet attacks compelling them to press their fingerprints. We introduce an innovative behavioral pattern to counter puppet attacks, requiring the user to consecutively press the fingerprint module using two different fingers. Deconstructing this pattern into single actions involves a first press with one finger, transitioning to another finger, and a second press with a different finger. The temporal aspect of switching fingers is crucial in this behavioral pattern, prompting us to consider both the fingerprint image and the time of finger switching for comprehensive puppet attack detection.

TABLE I
ANALYSIS OF METHODS FOR LIVENESS DETECTION AND PUPPET ATTACK DETECTION

Detectable Attacks	Method		References	(Common) Disadvantages
Presentation attack	Hardware-based	Pulse oximetry	[9]	1.Vulnerable to puppet attacks. 2.Require additional sensors.
		Blood pressure	[10], [11]	
		Odor	[12]	
		Electrical properties	[13], [14]	
		OCT	[15]–[19]	
	Software-based	Skin distortion	[20]–[22]	1.Vulnerable to puppet attacks. 2.Relatively complex image processing algorithm.
		Perspiration	[23]–[26]	
		Sweat pores	[27]–[29]	
		Surface coarsenes	[30]	
		Texture feature	[31], [32]	
Puppet attack	FINAUTH		[8]	Handheld authentication device needed, limiting applicability.

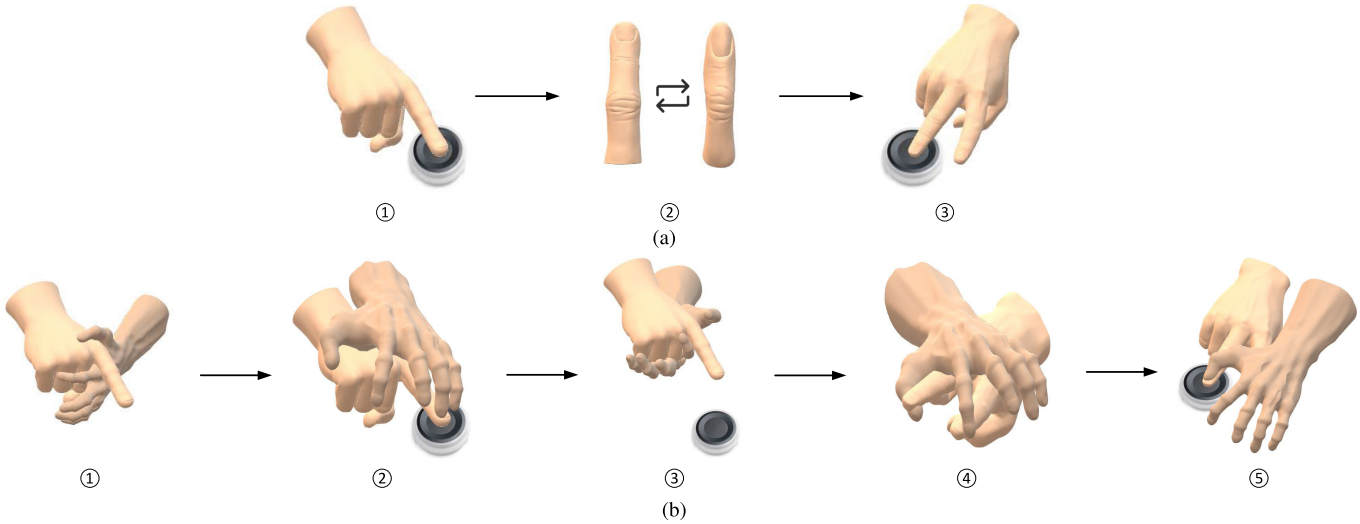


Fig. 2. Behavioral pattern analysis in normal and attacked states. (a) Normal state. (b) Attacked state.

A. Rationale Behind the Effectiveness of this Behavior Pattern

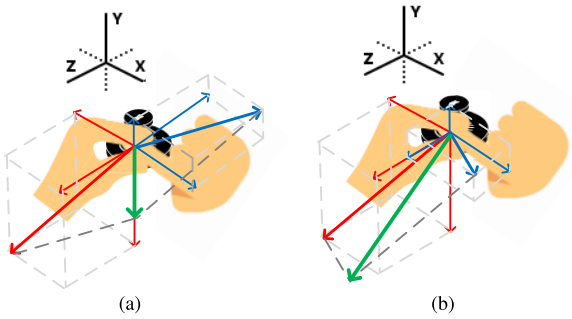
When users execute this behavior pattern under normal circumstances, they adhere to their accustomed rate, direction, and force while contacting the fingerprint sensor. The finger-switching process is conducted in a relaxed and natural manner, as illustrated in Fig. 2(a).

Conversely, when the user is coerced into performing the authentication action, as depicted in the first diagram of Fig. 2(b), the attacker forcibly elevates the victim's finger to execute the first fingerprint press. Subsequently, illustrated in the second diagram of Fig. 2(b), the attacker extends two fingers and forcefully presses the victim's index finger onto the fingerprint module. Following this, as demonstrated in the third diagram of Fig. 2(b), the attacker utilizes the palm to forcefully raise the victim's finger, which was used for the initial press. During finger switching, the victim's resistance to cooperation prompts the attacker to forcibly use their thumb to lift the victim's middle finger for the subsequent press, as shown in the fourth diagram of Fig. 2(b). Subsequently, the attacker controls the victim's hand, using his own fingers to forcibly collect the fingerprint with the victim's middle finger in the last diagram of Fig. 2(b).

Analysis of fingerprint images and finger-switching duration revealed significant differences between the normal and forced states. In the second and fifth diagrams of Fig. 2(b), due to the victim's resistance and the attacker's coercion, the obtained fingerprint image significantly deviates from the normal image, affecting the center and force of the press.

Besides, in Fig. 2(b), coercion, resistance, or trembling prolongs the time required for the attacker to align the victim's finger with the fingerprint module, increasing the time spent on switching fingers in the behavioral pattern. We elucidate this phenomenon by conducting an analysis of forces in two critical scenarios [41]. As illustrated in Fig. 3, the attacker's force is depicted through red arrows, the victim's force through blue arrows, and the resulting force through green arrows. At the depicted moment in Fig. 3(a), the magnitudes of forces in the x- and z-directions are equal but opposite for both the attacker and the victim. In the y-direction, however, the force applied by the victim is lesser than the downward force applied by the attacker, leading to a resultant force directing downward. Consequently, the attacker can compel the victim to engage the fingerprint acquisition module.

The above analysis leads us to the following two conclusions.



(a) (b)

Fig. 3. Force analysis in two cases. (a) Successfully press the acquisition device. (b) Slip away from the acquisition device.

1) No matter how disparate the strength difference between the victim and the attacker is, it is very difficult for the attacker to align the victim's finger to the sensor within the time interval in the normal state, because in the case of the victim struggling and the attacker forcibly controlling it, even a small change of the victim's strength can lead to a significant change of the resulting combined force.
2) Resistance movements that may occur in a victim of a puppet attack, such as moving the finger away from the sensor or rotating the finger as far as possible when forced to press, can make the resulting fingerprint image significantly different from that in the normal state, e.g., the center of the press, the angle of rotation, or the force of the press.

B. Justification for this Specific Behavior Pattern

The rationale behind defining a behavior pattern that necessitates the user to press the fingerprint module twice consecutively with two different fingers, as opposed to pressing twice with the same finger or pressing the sensor thrice with three different fingers, is twofold.

1) Experimental proof and analysis in Section V reveal the deficiencies of using the same finger to press the sensor twice, followed by utilizing image features and time features for puppet attack detection.
2) We conducted an experiment measuring the total duration of pressing the sensor thrice with three different fingers, averaging 6.1298, more than twice the average duration of our defined behavioral pattern. Considering data collection convenience and user experience, we chose the pragmatic approach of using two different fingers to press the fingerprint module twice in succession.

Therefore, we choose this specific behavior pattern for the detection of puppet attacks.

C. Framework of PUPGUARD

The framework of PUPGUARD is shown in Fig. 4. PUPGUARD utilizes user behavior patterns to capture intrinsic image features and timing characteristics, subsequently integrating a two-factor authentication mechanism. This approach bolsters security by necessitating two distinct finger presses and introducing a time gap between them, rendering it more challenging for potential attackers to replicate the authentication procedure. The proposed scheme achieves detection of puppet attacks through monitoring.

1) Pressure applied to the sensor during finger presses.
2) Time gap between the two presses.

More precisely, pressure monitoring is executed using 160×160 pixel matrices, while time interval monitoring is achieved by calculating the difference in the generation time of the two fingerprint images.

Our initial process involves the independent preprocessing of both fingerprint images and timing characteristics. Subsequently, we apply LBP, HOG, and ResNets to extract distinctive features from characterized behavioral patterns. Then, we perform feature selection on the image-based features. Following this, we merge image-based and time-based features through feature fusion, creating a fused feature vector. This vector is then fed into a one-class classifier to derive the final classification results. It is worth noting that we also experiment with decision level fusion, which will be presented in the subsequent sections.

IV. Proposed Method

The workflow of PUPGUARD can be divided into the following steps: data acquisition, data preprocessing, feature extraction and selection, feature fusion, and classification. We also try not to use feature fusion but to classify the two features separately and apply decision fusion. Therefore, in this section, we present the implementation details of the above steps one by one.

A. Data Acquisition

Since the PUPGUARD method requires experimental data derived from a specific behavioral pattern, it is not possible to directly utilize existing databases for experimental data. Here, we show the data collection and data acquisition process of PUPGUARD.

1) Fingerprint Acquisition Module: We compared a variety of fingerprint acquisition modules, and finally chose BM2166 semiconductor fingerprint module [42], because it integrates semiconductor sensor and fingerprint algorithm chip, and has the advantages of small size, low power consumption, simple interface, high module reliability, and good adaptability to wet and dry fingers. The parameters of BM2166 are shown in Table II. The fingerprint module and STM32 micro-controller together form the fingerprint acquisition system [43], as shown in Fig. 5.

The system can capture fingerprint images in various pressing situations, whether the volunteer is pressing at various angles and centers, or when the volunteer's finger is unintentionally and subtly sliding or rolling during the pressing process. Meanwhile, the system records the current time in standard format `yyyymmddHHMMSS.xxxxxx` each time it successfully captures a fingerprint image. In this format, `yyyy` represents the four-digit year, `mm` represents the two-digit month, `dd` represents the two-digit day of the month, `HH` represents the two-digit hour of the day in 24-h format, `MM` represents the two-digit minute in the hour, `SS` represents the two-digit number of seconds in the minute, and `xxxxxx` represents the six-digit number of microseconds in the second.

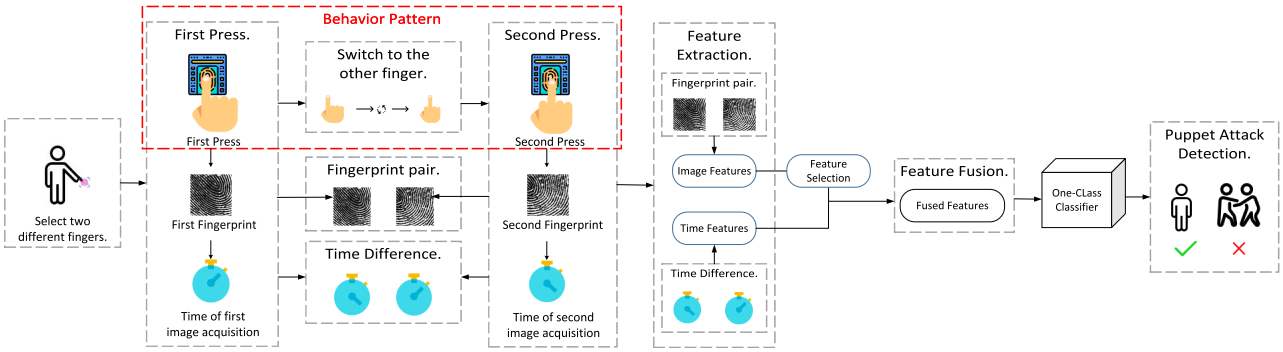


Fig. 4. Framework of the proposed PUPGUARD.

TABLE II
PARAMETERS OF THE BM2166

Parameter	Performance
Image Size	8 mm × 8 mm
Image Pixels	160 × 160
Resolution	508
Sensor Durability	>100k times
Fingerprint Image Acquisition Time	<110ms
Overall Recognition tTime	<1s
Communication Interface	UART
Communication Baud Rate	57600
Operating Temperature	−20°C to +40°C
Storage Temperature	−40°C to +70°C

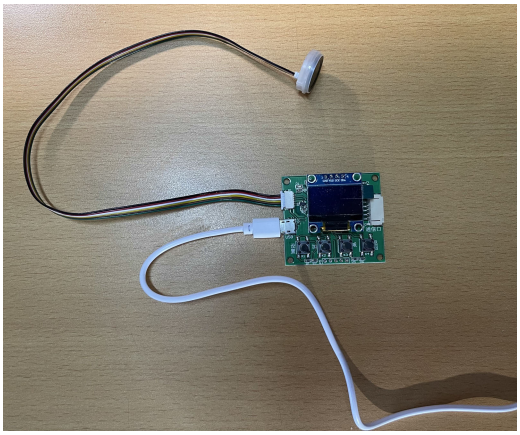


Fig. 5. Fingerprint acquisition module.

2) Acquisition Details: A successful data entry is defined as follows in accordance with the behavioral pattern: volunteers, in a relaxed and natural state, selecting two different fingers and pressing the fingerprint collection module twice in a row, with each finger pressing the module once, in a continuous and natural manner without deliberate pauses or accelerations. We ensure that all volunteers' pressing actions are considered normal, accommodating various legitimate scenarios that may occur. For instance, if after pressing the first finger, the volunteer notices dust on the second finger, they can simply wipe it off and proceed with the second pressing action. Similarly, if the volunteer encounters any other minor interruptions or adjustments during the process, they can be accommodated as long as they align with the overall requirements of the behavioral pattern.

Volunteers were asked to complete data entry using different pressing positions, including pressing with fingertips, pressing with the middle of fingers, pressing with the side of fingers, and pressing with the lower part of fingers. Since almost all volunteers are not accustomed to using their ring fingers for fingerprint pressing, only seven volunteers participated in data entry with their ring fingers of both hands, completing a total of 31 pairs of fingerprints with the ring finger. Other volunteers were asked to use their thumbs, index fingers, middle fingers, and little fingers of the left and right hands to complete the data entry.

A complete data acquisition process of the acquisition system can be summarized in the following steps: 1) the volunteer selects two different fingers; 2) the two fingers are pressed consecutively according to the requirements of a specific behavioral pattern; 3) the system sets up the two captured fingerprint images as a fingerprint pair; 4) the system records the moments of the two fingerprint acquisitions and makes the difference; and 5) the system adds the fingerprint pair and the time difference to the dataset as a set of data.

During the data entry process, the collection device was fixed on a table at a height of 1.2 m. We required volunteers to perform data collection in two postures: standing and sitting. In the sitting posture, volunteers were instructed to sit within a range of 0.2–0.5 m in front of the collection device. In the standing posture, volunteers were required to stand in front of the collection device with their arms naturally hanging down to complete the data collection. Between each successful data entry behavior, volunteers were required to completely remove their fingers from the fingerprint collection device to ensure a significant difference between each data entry behavior.

3) Data Constitution: The dataset contains only data collected from volunteers in their normal state, which means that it does not include any anomalous data collected from volunteers who are under puppet attacks. The dataset encompasses various pressing postures that users would naturally adopt, including different pressing angles and centers, as shown in Fig. 6(a). At the same time, the dataset includes various combinations of two presses with different fingers. Combinations refer to pressing two different fingers in two different orders, such as pressing the thumb first and then the index finger, or vice versa. The detailed compositional information of our collected dataset is shown in Table III.

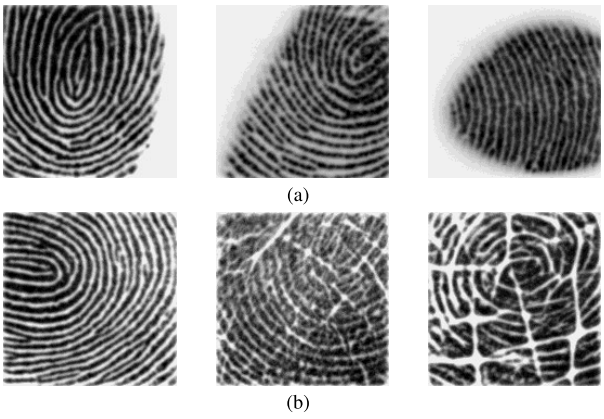


Fig. 6. Sample fingerprints in the dataset. (a) Different pressing gestures. (b) Different degrees of fingerprint wear.

TABLE III
DETAILED INFORMATION OF THE DATASET

Combination	Fingerprint pairs	Standing/Sitting
Thumb+Index Finger	124	50%/50%
Thumb+Middle Finger	124	50%/50%
Thumb+Ring Finger	10	50%/50%
Thumb+Little Finger	31	48%/52%
Index Finger+Middle Finger	124	50%/50%
Index Finger+Ring Finger	10	50%/50%
Index Finger+Little Finger	31	48%/52%
Middle Finger+Ring Finger	11	45%/55%
Middle Finger+Little Finger	31	48%/52%

A total of 31 participants were involved in the data collection process, comprising 12 females and 19 males. Their ages ranged from 20 to 85 years, with nine participants falling within the 20–30 age range, six participants within the 30–45 age range, six participants within the 45–50 age range, seven participants within the 50–56 age range, and three participants within the 56–85 age range. The larger age range ensures that the dataset encompasses the condition of fingerprint wear in all age groups, as shown in Fig. 6(b).

B. Data Preprocessing

The preprocessing of experimental data is divided into two parts: preprocessing of fingerprint images and preprocessing of timing characteristics. For timing characteristics, we standardize them. For fingerprint images, we utilize two different preprocessing methods, one using the classical image segmentation algorithm and the other based on resizing, cropping, and normalization.

1) Image Preprocessing Based on Otsu: For fingerprint image segmentation, we employ the Otsu method. Otsu's thresholding algorithm finds an optimal threshold value to separate image foreground and background based on grayscale variance [44]. This robust technique handles varying lighting, contrast, and noise levels in image processing tasks. Utilizing this optimal threshold achieves image segmentation. To visualize, Fig. 7 contrasts the original and Otsu processed images. This preprocessing approach is labeled *Prepro1*.

2) Image Preprocessing Based on Resizing, Cropping, and Normalization: The images in our training dataset undergo a series of preprocessing steps to prepare them for analysis. Initially, these images are subjected to resizing and center cropping to achieve uniformity in size, ensuring that they can be effectively processed by our model. Subsequently, we convert the images into PyTorch tensors, as this format is compatible with our chosen model architecture.

Fig. 7. Comparison of fingerprint images before and after using Otsu. (a) Original fingerprint image. (b) Fingerprint image using Otsu.

Once the images are transformed into tensors, we take an essential step in the preprocessing pipeline, which involves normalizing the pixel values. This normalization process is crucial for achieving standardized data representation throughout the subsequent processing stages. By scaling the pixel values appropriately, we bring the images to a common scale and remove any potential biases in the data.

The combination of resizing, center cropping, converting to tensors, and pixel value normalization forms a critical foundation for the success of our model during training. These preprocessing steps allow the model to effectively learn and extract meaningful features from the images, leading to better performance and generalization on unseen data. This preprocessing approach is labeled *Prepro2*.

3) Timing Characteristics Standardization: For timing characteristics standardization, we utilize the formula

$$t^* = \frac{t - \mu}{\sigma} \tag{1}$$

where μ and σ are the mean and standard deviation of the timing characteristics, respectively. Numerically, μ is equal to 3.0645, and σ is equal to 0.0055. Using this formula, we can transform the raw data point t into a deviation relative to μ, and then divide it by σ to ensure that the standardized data has a unit variance.

C. Feature Extraction and Feature Selection

In this section, feature extraction and feature selection is discussed. Since timing characteristics is 1-D data, normalized timing data is directly used as timing characteristics. For the preprocessed fingerprint images, we use and compare two different features, i.e., LBP- and HOG-based features, and ResNet-based features. To select the best feature combinations as well as reduce the feature dimensions, we also perform feature selection on the image features.

1) LBP- and HOG-Based Features: The LBP algorithm, which was first proposed by Ojala et al. [45] for texture classification, is a widely used texture descriptor in computer vision applications. The basic idea of LBP is to compare each pixel in an image with its surrounding neighbors. For each pixel, a binary code is generated based on whether the surrounding pixel has a higher or lower intensity value than the center pixel. These binary codes are then concatenated to form a unique pattern for that local neighborhood.

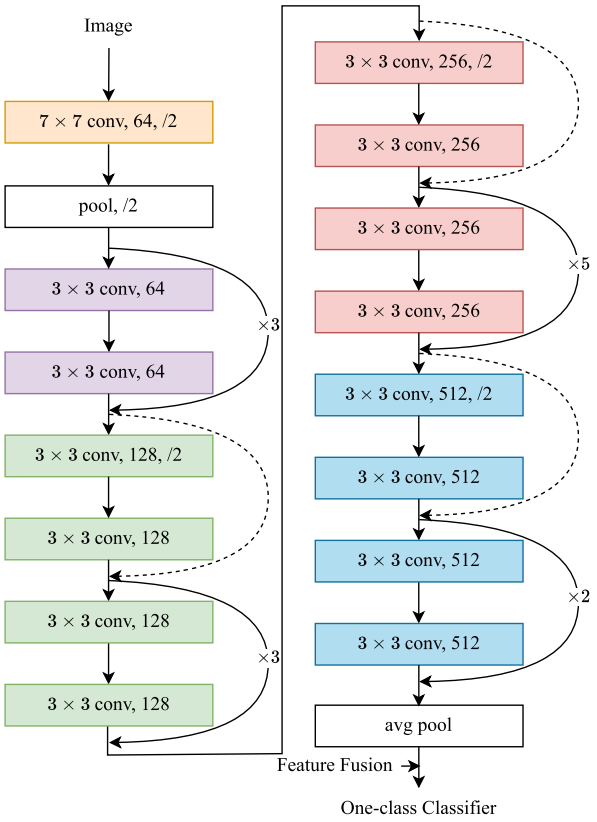


Fig. 8. Framework of ResNet34-based feature extractor.

The HOG algorithm works by analyzing the gradient orientations of small image patches and constructing histograms of these orientations [46]. These histograms are then normalized and concatenated to form a feature vector that represents the image.

2) ResNet-Based Features: ResNet, short for Residual Network, is a deep convolutional neural network architecture proposed by He et al. [47]. It utilizes residual blocks, employing "skip connections" to pass residual information, effectively tackling the vanishing gradient problem in deep networks. ResNet allows the construction of exceptionally deep networks and achieves outstanding performance in computer vision tasks.

To leverage ResNet for extracting image features, we modified the original architecture by retaining the convolutional layers and pooling layers responsible for learning hierarchical spatial features, while discarding the fully connected layers used for classification in the original network. This alteration facilitates the extraction of higher level, semantically rich feature representations from the input images, which can be utilized for puppet attack detection. For instance, the framework of using the ResNet34 to extract features for subsequent classification is shown in Fig. 8.

3) Feature Selection on Image Features: After extracting the image features using the method described above, the image features are still high dimensional compared to the 1-D timing characteristics. To select the best feature combinations as well as reduce the feature dimensions, we perform feature selection on the image features, and in our experiments we employ principal component analysis (PCA).

TABLE IV
CHANGES IN DIMENSIONALITY OF FEATURE VECTORS AFTER PCA

Features	Dimension before PCA	Dimension after PCA
LBP	51200	481
HOG	25922	450
ResNet34-based	1024	267
ResNet50-based	4096	277
ResNet101-based	4096	298
ResNet152-based	4096	285

PCA is a popular data analysis technique for handling high-dimensional datasets. It achieves dimensionality reduction by linearly transforming data into a new coordinate system while retaining as much information as possible in lower dimensions, thereby enhancing data interpretability. It identifies principal components, with the first principal component being the direction that maximizes the variance of the projected data, and subsequent principal components being orthogonal to the previous ones while also maximizing the variance of the projected data. The changes in the dimensions of the feature vectors before and after the use of PCA in our approach are shown in Table IV.

After feature selection and feature dimensionality reduction, the image features will complete feature fusion with 1-D timing characteristics, as will be described below.

D. Feature Fusion and Decision Fusion

In our defined behavior pattern, timing characteristics are represented as 1-D, while image features belong to high-dimensional space. Therefore, we employ two fusion methods to deal with these two features. The first method is feature fusion, where we fuse the two features to form a 1-D feature vector, and this fused feature vector can characterize the behavioral patterns more effectively. The second method is decision level fusion, where we use two classifiers, as will be described in Section IV-E, to process image features and timing characteristics separately, and then the outputs of the two classifiers are fused to obtain the final classification results.

1) Feature Concatenation: We concatenate image features and timing characteristics into a single larger feature vector, and then use this merged vector for prediction.

2) Feature Cross: We intersect image features with timing characteristics to generate newly combined features. In particular, we multiply each element of the image features with the timing characteristics to create a new feature vector. This method is suitable when there is some correlation between image and time features.

E. Decision Fusion

In addition to employing feature fusion, we explore the utilization of two distinct classifiers to independently process the distinct feature types within PUPGUARD. Subsequently, a decision fusion mechanism is applied to amalgamate the classification outcomes of the two classifiers, yielding the ultimate detection results. We have explored two methods of

decision fusion: simple majority voting [48] and weighted majority voting [49].

1) Simple Majority Voting: We adopt a unanimous voting approach to accomplish decision fusion [50]. Two distinct one-class classifiers, denoted as C_{time} and C_{img}, are trained for the recognition of timing and image features, respectively. Upon receiving input samples, C_{time} processes time features, while C_{img} processes image features. The decisions rendered by these classifiers are independent and are represented as d_{time} and d_{img} for timing and image classification, respectively. The final decision classifies a sample as positive only if both d_{time} and d_{img} are positive; otherwise, the final decision categorizes the sample as an anomaly.

2) Weighted Majority Voting: C_{time} and C_{img} may not perform equally well and therefore, employing unanimous voting may not be optimal. In this case, the appropriate solution is to weight each classifier based on its performance. We employ a weighted majority voting approach introduced in [49].

In this approach, the iterative process involves updating weights for each instance within the validation set. Initially, all weights are uniformly set to 1. The weights of the classifiers that correctly predict class label of an instance are incremented by the ratio of the number of incorrectly predicting classifiers to the whole number of classifiers, as outlined in the following:

$$w_{ij} = \begin{cases} w_{i-1,j} + \alpha_i, & \text{if the } j\text{th classifier makes a correct prediction for the } i\text{th instance} \\ w_{i-1,j}, & \text{if the } j\text{th classifier makes an incorrect prediction for the } i\text{th instance} \end{cases} \tag{2}$$

where w_{ij} denotes the weight of jth classifier as the operation realized on ith instance. The alteration in weight, denoted as α_i, is computed as $\alpha_i = Y_i/n$, where Y_i represents the count of erroneous predictions for the ith instance and n is the number of classifiers.

When all instances in the validation set are processed through C_{time} and C_{img} once, the resultant values are stored as weights, denoted as w_{time} and w_{img}, respectively. These weights are subsequently employed as the voting influence for each classifier in predicting class labels for instances within the test set. In the ultimate decision-making process, the sum of all weighted votes for each class is calculated. The class with the highest weighted vote is designated as the predicted class for the instance to be classified, as outlined in (3), where $d_{t,c} = 1$, if classifier t decides for class c, and $d_{t,c} = 0$ otherwise

$$\max_{c \in \{\text{normal,outlier}\}} \sum_{t \in \{\text{img,time}\}} w_t d_{t,c}. \tag{3}$$

F. Detection Based on One-class Classifiers

Since our dataset contains only legitimate user data and no outlier data, this is a one-class classification problem. Therefore, we use the following three models to detect puppet attacks: 1) one-class support vector machine (OC-SVM); 2) isolation forest (IF); and 3) local outlier factor (LOF).

1) One-Class Support Vector Machine: OC-SVM is a type of support vector machine algorithm that is used for novelty detection. The goal of one-class SVM is to learn a decision boundary that separates the normal data points from the outliers. The algorithm takes a single class of input data, typically representing the normal class, and learns a decision boundary that maximizes the margin around the normal data points [51]. This margin is defined as the distance between the decision boundary and the closest data point from the normal class.

2) Local Outlier Factor: LOF is based on the concept of local density, determined by considering k nearest neighbors and their distances [52]. By comparing the local density of an object with that of its neighbors, regions with similar density can be identified, along with points that have significantly lower density than their neighbors, classifying them as outliers. The local density is estimated by the typical distance at which a point can be "reached" from its neighbors. The definition of "reachability distance" used in LOF is an additional measure to produce more stable clustering results.

3) Isolation Forest: IF is a popular anomaly detection algorithm introduced by Liu et al. [53]. It efficiently identifies outliers in large-scale datasets by creating random binary trees and measuring the isolation of anomalies based on their shorter path lengths from the root. Its non-parametric nature, computational efficiency, and effectiveness in high-dimensional data have made it widely utilized in various domains, including cybersecurity, fraud detection, and fault diagnosis.

V. Experiments and Analyses

A. Experimental Preparation and Evaluation Indexes

To evaluate the performance of PIPGUARD, we create a test set that contains 94 fingerprint pairs (188 fingerprint images) and corresponding time difference data, including 41 positive samples and 53 negative samples. Abnormal behavior is defined as any instance or combination of the following behaviors during the data collection process: 1) forcefully pressing the fingerprint module with a single finger; 2) forcefully pressing the fingerprint module with both fingers simultaneously; and 3) exhibiting an unusually prolonged or shortened time difference between the two finger presses.

We collected the test set by involving different combinations of male victims and male attackers, female victims and male attackers, male victims and female attackers, and female victims and female attackers. When collecting negative samples, victims adopt two postures: standing and sitting. The attacker's actions mainly include using two fingers to pinch the victim's fingers to complete the pressing, using one finger placed above the victim's fingers to complete the pressing, dragging the back of the victim's fingers to complete the pressing, and supporting the victim's palm with the palm of the hand, using the thumb to press the victim's fingers to complete the pressing. In both postures, negative samples are collected under the following scenarios.

1) The victim simulates deep unconsciousness, and the attacker has complete control over the victim to complete the collection. The victim naturally hangs

TABLE V
EXPERIMENTAL RESULTS OF PUPGUARD

Features	One-class Classifier	Feature Fusion	Accuracy	FPR	Recall	Precision	F1-score
ResNet34-based	OC-SVM	Feature Cross	93.62%	3.77%	90.24%	94.87%	0.92
		Feature Concatenation	65.96%	52.83%	90.24%	56.92%	0.70
	IF	Feature Cross	93.62%	3.77%	90.24%	94.87%	0.92
		Feature Concatenation	79.79%	32.08%	95.12%	69.64%	0.80
	LOF	Feature Cross	93.62%	9.43%	97.56%	88.89%	0.93
		Feature Concatenation	52.13%	84.91%	100.00%	47.67%	0.65
ResNet50-based	**OC-SVM**	**Feature Cross**	**97.87%**	**1.89%**	**97.56%**	**97.56%**	**0.98**
		Feature Concatenation	68.09%	56.60%	100.00%	57.75%	0.73
	IF	Feature Cross	93.62%	7.55%	95.12%	90.70%	0.93
		Feature Concatenation	63.83%	54.72%	87.80%	55.38%	0.68
	LOF	Feature Cross	88.30%	18.87%	97.56%	80.00%	0.88
		Feature Concatenation	48.94%	88.68%	97.56%	45.98%	0.63
ResNet101-based	OC-SVM	Feature Cross	89.36%	5.66%	82.93%	91.89%	0.87
		Feature Concatenation	65.96%	49.06%	85.37%	57.38%	0.69
	IF	Feature Cross	94.68%	3.77%	92.68%	95.00%	0.94
		Feature Concatenation	76.60%	37.74%	95.12%	66.10%	0.78
	LOF	Feature Cross	96.81%	5.66%	100.00%	93.18%	0.96
		Feature Concatenation	53.19%	83.02%	100.00%	48.24%	0.65
ResNet152-based	OC-SVM	Feature Cross	90.43%	3.77%	82.93%	94.44%	0.88
		Feature Concatenation	61.70%	54.72%	82.93%	53.97%	0.65
	IF	Feature Cross	93.62%	7.55%	95.12%	90.70%	0.93
		Feature Concatenation	73.40%	43.40%	95.12%	62.90%	0.76
	LOF	Feature Cross	94.68%	7.55%	97.56%	90.91%	0.94
		Feature Concatenation	48.94%	90.57%	100.00%	46.07%	0.63
LBP	OC-SVM	Feature Cross	88.29%	15.09%	92.68%	82.61%	0.87
		Feature Concatenation	43.62%	96.23%	95.12%	43.33%	0.60
	IF	Feature Cross	45.74%	94.34%	97.56%	44.44%	0.61
		Feature Concatenation	44.68%	96.23%	97.56%	43.96%	0.61
	LOF	Feature Cross	79.79%	33.96%	97.56%	68.97%	0.81
		Feature Concatenation	44.68%	96.23%	97.56%	43.96%	0.61
HOG	OC-SVM	Feature Cross	84.04%	26.42%	97.56%	74.07%	0.84
		Feature Concatenation	42.55%	100.00%	97.56%	43.01%	0.59
	IF	Feature Cross	61.70%	67.92%	100.00%	53.25%	0.69
		Feature Concatenation	42.55%	100.00%	97.56%	43.01%	0.60
	LOF	Feature Cross	69.15%	54.72%	100.00%	58.57%	0.74
		Feature Concatenation	43.62%	100.00%	100.00%	43.62%	0.61

down the upper arm, forearm, and palm without any resistance.

2) The victim is awake, and the collection process has not started when under the control of the attacker. Resistance is attempted during the controlled collection process. The resistance measures taken by the victim mainly include retracting fingers, rotating fingers, forcefully lifting fingers, and shaking fingers.
3) The victim is awake and has already started the normal collection process. Just before completing the normal authentication, the attacker suddenly controls the victim's hand, rapidly completing the remaining collection process. In this collection scenario, the victim simulates a lack of awareness of the attack, i.e., after being controlled, the victim rapidly completes the remaining collection process without resistance.

It should be noted that this test set is only a subset of all puppet attack scenarios because we cannot collect other types of puppet attack data, such as uncontrollable trembling or genuine unconsciousness due to violence or weapons.

All samples within the test set undergo identical preprocessing procedures as those applied to the training set. In particular, the fingerprint images are either processed through Otsu preprocessing or undergo resizing, cropping, and standardization. The timing features are standardized using (1), with the mean and standard deviation of the timing features for all samples in the test set calculated as 4.7085 and 3.1637, respectively. Furthermore, for the negative samples within the test set, the mean and standard deviation of the time features are computed as 5.9677 and 3.7669, respectively. Upon comparing the mean and standard deviation of the time characteristics for all positive samples as presented in Section IV-B3, a notable disparity is evident. The mean and standard deviation of the time characteristics for abnormal samples are substantially larger. This discrepancy arises from the challenges posed by the victim's struggle and resistance, making it arduous for the attacker to predict and control the victim. Consequently, there is an escalation in the duration and intra-class differences of the temporal features across all abnormal samples.

We measure the performance of our proposed method with accuracy, FPR, recall, precision, and $F1$-score. Accuracy is the proportion of correct predictions, recall is the probability of correctly predicting positive samples, precision refers to the proportion of correct predictions among all predicted positive samples, FPR is the probability of predicting an abnormal data as normal, and $F1$-score is the harmonic mean of precision and recall.

In practical scenarios, the tolerance for rejection surpasses that for exposure to illicit intrusions. Consequently, in evaluating the efficacy of PUPGUARD, emphasis should be placed on accuracy and FPR.

B. Performance of PUPGUARD

Table V presents the experimental results of the PUPGUARD method under different conditions. It is worth noting that the preprocessing method, Prepro2, mentioned earlier, is only combined with ResNet-based features, while Prepro1 is only combined with LBP- and HOG-based features.

Four types of deep learning-based features are evaluated using three classifiers, along with two feature fusion methods. LBP- and HOG-based features are evaluated with the same classifiers. It is noteworthy that regardless of which of the above feature extraction methods is used, we perform feature selection and dimensionality reduction on the extracted image features.

The methods using LBP- or HOG-based features for detecting puppet attacks demonstrate poor performance. Regardless of the one-class classifier or feature fusion method employed, the best achieved performance is only 88.29% accuracy and 15.09% FPR. These results are insufficient for effective security defense.

In contrast, employing ResNet-based features significantly improves performance. Specifically, using ResNet50-based features, OC-SVM, and feature cross-fusion, PUPGUARD achieves the highest accuracy of 97.87% and an FPR of 1.89%.

Furthermore, under the premise of using ResNet features, feature cross-fusion outperforms feature concatenation noticeably. This can be attributed to our defined behavior patterns having 1-D timing characteristics, while image features exist in a high-dimensional space.

If solely employing feature concatenation to construct fused feature vectors, certain limitations and challenges arise. A significant limitation is the dimensionality mismatch between timing and image features, potentially leading to suboptimal performance by not fully utilizing their complementary information. Additionally, differences in feature scales could result in biased performance, favoring one feature type over others during the learning process.

In contrast, employing the feature cross-fusion method creates a more integrated and informative representation. Leveraging the inherent relationships between different feature types and their complementary strengths leads to improved performance and more accurate detection of puppet attacks. Moreover, feature cross-fusion mitigates dimensionality mismatch issues and ensures a more efficient and effective use of the combined feature set in the learning process.

TABLE VI
EXPERIMENTAL RESULTS SOLELY BASED ON IMAGE FEATURES

Features	Classifier	Accuracy	FPR
ResNet50	OC-SVM	62.77%	66.04%
	IF	64.89%	58.49%
	LOF	46.81%	92.45%
LBP	OC-SVM	43.62%	98.11%
	IF	43.62%	98.11%
	LOF	43.62%	98.11%
HOG	OC-SVM	42.55%	100.00%
	IF	43.62%	100.00%
	LOF	43.62%	100.00%

TABLE VII
EXPERIMENTAL RESULTS SOLELY BASED ON TIMING CHARACTERISTICS

Features	Classifier	Accuracy	FPR
Timing	OC-SVM	88.29%	11.32%
	IF	89.36%	13.21%
	LOF	89.36%	13.21%

C. Detection Solely Based on Image Features

The purpose of this experiment is to demonstrate the necessity of using both image features and timing characteristics in the PUPGUARD method to characterize our defined behavior patterns, in other words, to demonstrate the superiority of combining timing characteristics to detect puppet attacks. Using only image features means that image features do not need to be fused with timing characteristics but are directly fed into a one-class classifier.

The performance of this experiment is shown in Table VI. The general performance of this experiment is suboptimal, with the highest achievable accuracy falling below 70%, and the FPR is unacceptably high. This may be attributed to the following reasons: when coerced, the victim will make different degrees of resistance. When the victim's resistance is robust, despite a significantly prolonged time interval between the two presses, the force applied to the fingerprint collection module may remain normal or even insufficient due to the resistance. In other words, in this case, the image features are normal but the timing characteristics is abnormal. If only the image features are used for puppet attack detection, there will be a high error rate and FPR.

D. Detection Solely Based on Timing Characteristics

The purpose of this experiment is to demonstrate the necessity of using both image features and timing characteristics in the PUPGUARD method to characterize our defined behavior patterns, in other words, to demonstrate the superiority of combining image features to detect puppet attacks. In this experiment, the input feature vector is only the timing characteristics, that is, the input is only 1-D features. The performance of this experiment is shown in Table VII.

The performance of this experiment is better than the experiment using only image features, but there is still a large performance difference compared to the method that uses both features for detection. This method also has obvious disadvantages, resulting in mediocre performance. Contrary to

TABLE VIII
EXPERIMENTAL RESULTS WITH UNANIMOUS VOTING

C_{img}	C_{time}	Accuracy	FPR
OC-SVM	OC-SVM	94.68%	0.00%
OC-SVM	IF	95.74%	1.89%
OC-SVM	LOF	95.74%	1.89%
IF	OC-SVM	92.55%	1.89%
IF	IF	92.55%	3.77%
IF	LOF	92.55%	3.77%
LOF	OC-SVM	90.43%	5.66%
LOF	IF	91.49%	7.55%
LOF	LOF	91.49%	7.55%

TABLE IX
EXPERIMENT RESULT WITH WEIGHTED MAJORITY VOTING

C_{img}	C_{time}	w_{img}	w_{time}	Accuracy	FPR
OC-SVM	OC-SVM	2.0	5.5	88.29%	13.21%
OC-SVM	IF	1.0	5.5	89.36%	13.21%
OC-SVM	LOF	1.0	5.5	89.36%	13.21%
IF	OC-SVM	2.0	1.0	68.09%	50.94%
IF	IF	1.0	1.0	94.12%	15.38%
IF	LOF	1.0	1.0	94.64%	15.00%
LOF	OC-SVM	2.0	1.5	46.81%	92.45%
LOF	IF	1.0	1.5	89.36%	13.21%
LOF	LOF	1.0	1.5	89.36%	13.21%

what was described in Section V-C, in this case, the attacker may have such a large power gap to the victim that the victim has to perform two quick presses. In this case, the time interval between pressings may be within the normal range, but the two pressing speeds are too fast and the force is too strong, resulting in excessive grayscale of the fingerprint image, severe deviation of the pressing center, or serious dragging marks in the pressing image. In other words, in this case, the image is abnormal but the timing characteristics is normal. If only the timing characteristics is used for detection, it will lead to huge risks.

E. Performance Using Unanimous Voting

Instead of feature fusion, we explore the utilization of decision fusion. We first employ a unanimous voting approach for decision fusion, as described in Section IV-E. Based on the experimental results of the above two experiments, we use ResNet50-based features as image features.

The experimental results using unanimous voting are shown in Table VIII. It can be observed that the overall performance using unanimous voting is commendable. It can be noted that FPR achievable with decision fusion is generally exceedingly low, even reaching 0.00% at one point. This is due to the fact that the final decision classifies a sample as positive only if both d_{time} and d_{img} are positive; otherwise, the final decision categorizes the sample as an anomaly. However, it can be seen that the accuracy of this method is not as good as the method of feature cross used in PUPGUARD, which is due to the fact that the method of unanimous voting produces too many false negative (FN) values.

F. Performance Using Weighted Majority Voting

Besides unanimous voting, we also employ the utilization of weighted majority voting. Initially, a validation group comprising 1/11 of the training dataset is randomly chosen for deployment in the weighting process. The initialization of weights commenced with all values set to 1. During each iteration involving instances from the validation set, only the weights of classifiers that make accurate predictions are incremented. The incrementation of weights is determined by the ratio of the number of classifiers with incorrect predictions to the total number of classifiers ($n = 2$). The experimental results using weighted majority voting are shown in Table IX. w_{img} and w_{time} represent the final weights assigned to image features and time features, respectively. These weights are employed as the voting influence for each classifier in predicting class labels.

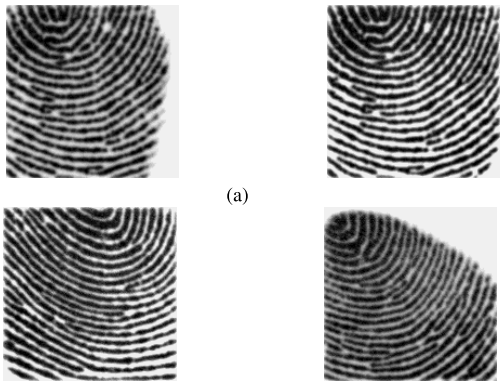


Fig. 9. Comparison of fingerprint pairs using the same finger and different fingers. (a) Two fingerprints pressed twice using the same finger. (b) Two fingerprints pressed in succession by two different fingers.

Observations reveal that when the voting influence of time features is greater than or equal to the voting influence of image features, both accuracy and FPR perform well. However, when the voting influence of time features is less than that of image features, both accuracy and FPR deteriorate significantly. In comparison to methods employing unanimous voting, this approach tends to exhibit a higher FPR, making it unsuitable for defense scenarios that prioritize high accuracy and low FPR.

G. Detection With Same Finger Pressed Twice

The purpose of this experiment is to demonstrate the necessity of using two different fingers in PUPGUARD. Specifically, in constructing the dataset, volunteers were asked to use the same finger to press twice, with the same requirements as described in Section IV. To complete this experiment, we invited the same volunteers as those who created the dataset described in Section IV, and each person completed two presses using the thumb, finger, middle finger, ring finger, and little finger, respectively, collecting a total of 282 fingerprint pairs and time interval data as the training dataset. At the same time, we also created a test dataset using the method described in Section V-A, which includes 50 fingerprint pairs and time interval data.

TABLE X
EXPERIMENTAL RESULTS WITH SAME FINGER PRESSING TWICE

Features	Classifier	Feature Fusion	Accuracy	FPR
ResNet50	OC-SVM	Cross	76.00%	33.33%
		Concatenation	76.00%	33.33%
	IF	Cross	88.00%	25.00%
		Concatenation	80.00%	41.67%
	LOF	Cross	68.00%	66.67%
		Concatenation	60.00%	83.33%
LBP	OC-SVM	Cross	60.00%	83.33%
		Concatenation	52.00%	100.00%
	IF	Cross	60.00%	83.33%
		Concatenation	52.00%	100.00%
	LOF	Cross	60.00%	83.33%
		Concatenation	52.00%	100.00%
HOG	OC-SVM	Cross	56.00%	83.33%
		Concatenation	48.00%	100.00%
	IF	Cross	60.00%	83.33%
		Concatenation	52.00%	100.00%
	LOF	Cross	60.00%	83.33%
		Concatenation	52.00%	100.00%

It can be seen that this method has very obvious flaws, namely a high FPR and low accuracy. The reason for this is related to the way the pressings are done. When the user needs to press two different fingers in succession, there must be a finger-switching action, which will cause significant changes in the angle, press center, and press intensity of the two presses. In this experiment, the user only needs to press the same finger twice in a row, and almost all users only lift their finger slightly after the first press to complete the second press, which will result in the fingerprint images of the two presses being extremely similar. Fig. 9(a) shows two fingerprints pressed twice using the same finger, while Fig. 9(b) shows two fingerprints pressed in succession by two different fingers. It can be clearly seen from Fig. 9(a) that the two fingerprints are almost the same. Therefore, in this case, the data in the dataset cannot include all the pressed fingerprints under normal conditions. In other words, when the input positive samples are too limited, the hyperplane output by the model deviates greatly from the actual hyperplane, resulting in lower accuracy, lower precision, and higher FPR. Moreover, from a practical point of view, this verification method will reduce the attack difficulty of the attacker, because the attacker does not need to force the victim to switch fingers, but only needs to forcibly lift the victim's finger and then press the fingerprint module. The performance of this experiment is shown in Table X.

H. Effect of Dataset Size on PUPGUARD Performance

The previous experiments have already demonstrated that using ResNet50 features and feature cross outperforms other methods. Therefore, when exploring the impact of the dataset size on PUPGUARD, we will only focus on using ResNet50 features and feature cross.

To explore the effect of training dataset size on detection performance, we use 20%, 40%, 60%, 80%, and 100% of the training dataset for training, respectively. Figs. 10 and 11 show the impact of different dataset sizes on accuracy and FPR.

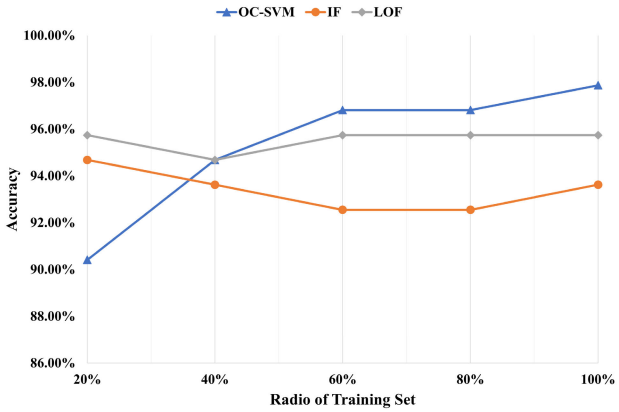


Fig. 10. Accuracy of PUPGUARD at different dataset sizes.

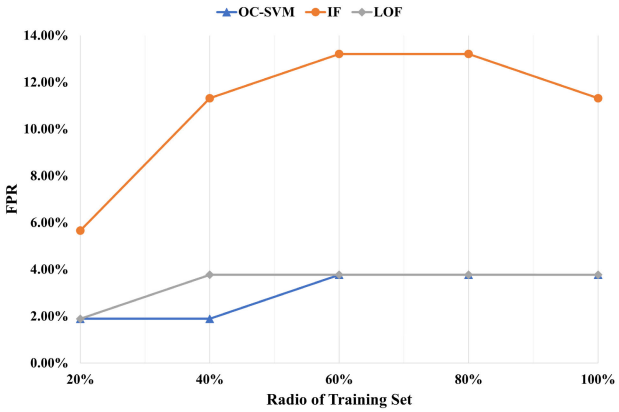


Fig. 11. FPR of PUPGUARD at different dataset sizes.

It can be observed that as the size of the training set increases, the detection accuracy and FPR of PUPGUARD gradually stabilize. This phenomenon arises from the fact that as the training set size expands, the classifier becomes more adept at capturing the data's characteristics, resulting in enhanced detection accuracy. Consistent detection accuracy and a stable FPR may indicate that the classifier has converged to a relatively steady state, suggesting minimal performance fluctuations with larger training set sizes. In fact, the accuracy of OC-SVM method steadily improves. Therefore, we can draw the conclusion that the detection performance of PUPGUARD does improve as the training set increases.

VI. LIMITATIONS OF PUPGUARD

A. User Adoption and Usability

Requiring users to follow a specific sequence of actions, such as pressing the fingerprint module twice with distinct fingers, might result in resistance or confusion among users. The added steps could potentially lead to a decline in user adoption due to increased complexity, affecting the overall usability and user experience of the authentication process. One possible solution is to add feedback mechanisms to ensure that users know whether they have performed an action in the right way by providing real-time feedback.

B. Implementation and Technical Constraints

Implementing a behavior-based authentication approach like PUPGUARD might require adjustments to hardware, software,

and user interfaces. The identification system needs to be able to recognize a complete series of authentication actions as a single authentication attempt, rather than multiple. Adapting existing authentication systems or developing new ones to incorporate dynamic behavior patterns can introduce technical challenges, compatibility issues, and potential vulnerabilities that must be carefully addressed to ensure the method's reliability and security.

VII. Conclusion

In this article, we present PUPGUARD, a solution crafted to provide protection against puppet attacks. PUPGUARD harnesses user behavior patterns, particularly the sequence of pressing the fingerprint module with different fingers, to capture inherent image features and timing characteristics. By adopting this two-factor authentication approach, we fortify security against puppet attacks, prioritizing the observation of dynamic behavior patterns throughout the authentication process. The requirement for two separate finger presses introduces an extra layer of security, with the time gap between these presses increasing the complexity for potential attackers. This comprehensive approach enhances security against fingerprint PAs.

To evaluate the effectiveness of PUPGUARD, we performed experiments using datasets gathered from 31 subjects, encompassing both image features and timing characteristics. These data collection procedures were carried out with the approval of the IRB. The results of our experiments clearly illustrate PUPGUARD's exceptional performance, achieving the highest accuracy at 97.87% and the lowest FPR at 1.89%, respectively. Additionally, we conducted comparative experiments to affirm the advantage of incorporating both image features and timing characteristics into PUPGUARD, thereby reinforcing its resistance against puppet attacks.

References

[1] F. Liu et al., "A flexible touch-based fingerprint acquisition device and a benchmark database using optical coherence tomography," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 6518–6529, Sep. 2020.

[2] ISO. (2016). *ISO/IEC 30107-1:2016*. [Online]. Available: https://www.iso.org/standard/53227.html

[3] S. Marrone, R. Casula, G. Orrù, G. L. Marcialis, and C. Sansone, "Fingerprint adversarial presentation attack in the physical domain," in *Pattern Recognition*. Cham, Switzerland: Springer, 2021, pp. 530–543.

[4] M. Espinoza and C. Champod, "Risk evaluation for spoofing against a sensor supplied with liveness detection," *Forensic Sci. Int.*, vol. 204, nos. 1–3, pp. 162–168, Jan. 2011.

[5] A. Wiehe, T. Søndrol, O. K. Olsen, and F. Skarderud, "Attacking fingerprint sensors," NISlab, Gjøvik Univ. College, Gjøvik, Norway, Tech. Rep., 200, 2004.

[6] M. Espinoza, C. Champod, and P. Margot, "Vulnerabilities of fingerprint reader to fake fingerprints attacks," *Forensic Sci. Int.*, vol. 204, nos. 1–3, pp. 41–49, Jan. 2011.

[7] C. Sousedik and C. Busch, "Presentation attack detection methods for fingerprint recognition systems: A survey," *IET Biometrics*, vol. 3, no. 4, pp. 219–233, Dec. 2014.

[8] C. Wu, K. He, J. Chen, Z. Zhao, and R. Du, "Liveness is not enough: Enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks," in *Proc. 29th USENIX Secur. Symp.* Berkeley, CA, USA: USENIX Association, 2020, pp. 2219–2236.

[9] P. Venkata Reddy, A. Kumar, S. M. K. Rahman, and T. Singh Mundra, "A new antispoofing approach for biometric devices," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 4, pp. 328–337, Dec. 2008.

[10] M. Drahansky, R. Notzel, and W. Funk, "Liveness detection based on fine movements of the fingertip surface," in *Proc. IEEE Inf. Assurance Workshop*, Jun. 2006, pp. 42–47.

[11] P. D. Lapsley, J. A. Lee, D. F. Pare Jr., and N. Hoffman, "Anti-fraud biometric scanner that accurately detects blood flow," U.S. Patent 5 737 439, Apr. 7, 1998.

[12] D. Baldisserra, A. Franco, D. Maio, and D. Maltoni, "Fake fingerprint detection by odor analysis," in *Advances in Biometrics*. Cham, Switzerland: Springer, 2005, pp. 265–272.

[13] O. G. Martinsen, S. Clausen, J. B. Nysaether, and S. Grimnes, "Utilizing characteristic electrical properties of the epidermal skin layers to detect fake fingers in biometric fingerprint systems—A pilot study," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 5, pp. 891–894, May 2007.

[14] T. Shimamura et al., "Impedance-sensing circuit techniques for integration of a fraud detection function into a capacitive fingerprint sensor," *IEEE Sensors J.*, vol. 12, no. 5, pp. 1393–1401, May 2012.

[15] Y. Cheng and K. V. Larin, "Artificial fingerprint recognition by using optical coherence tomography with autocorrelation analysis," *Appl. Opt.*, vol. 45, no. 36, p. 9238, 2006.

[16] A. Bossen, R. Lehmann, and C. Meier, "Internal fingerprint identification with optical coherence tomography," *IEEE Photon. Technol. Lett.*, vol. 22, no. 7, pp. 507–509, Feb. 2, 2010.

[17] Y. Cheng and K. V. Larin, "In vivo two- and three-dimensional imaging of artificial and real fingerprints with optical coherence tomography," *IEEE Photon. Technol. Lett.*, vol. 19, no. 20, pp. 1634–1636, Sep. 24, 2007.

[18] M.-R. Nasiri-Avanaki, A. Meadway, A. Bradu, R. M. Khoshki, A. Hojjatoleslami, and A. G. Podoleanu, "Anti-spoof reliable biometry of fingerprints using en-face optical coherence tomography," *Opt. Photon. J.*, vol. 1, no. 3, pp. 91–96, 2011.

[19] G. Liu and Z. Chen, "Capturing the vital vascular fingerprint with optical coherence tomography," *Appl. Opt.*, vol. 52, no. 22, p. 5473, 2013.

[20] A. Antonelli, R. Cappelli, D. Maio, and D. Maltoni, "Fake finger detection by skin distortion analysis," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 3, pp. 360–373, Sep. 2006.

[21] Y. Zhang, J. Tian, X. Chen, X. Yang, and P. Shi, "Fake finger detection based on thin-plate spline distortion model," in *Advances in Biometrics*. Cham, Switzerland: Springer, 2007, pp. 742–749.

[22] J. Jia, L. Cai, K. Zhang, and D. Chen, "A new approach to fake finger detection based on skin elasticity analysis," in *Advances in Biometrics*. Cham, Switzerland: Springer, 2007, pp. 309–318.

[23] R. Derakhshani, S. A. C. Schuckers, L. A. Hornak, and L. O'Gorman, "Determination of vitality from a non-invasive biomedical measurement for use in fingerprint scanners," *Pattern Recognit.*, vol. 36, no. 2, pp. 383–396, Feb. 2003.

[24] A. Abhyankar and S. Schuckers, "Integrating a wavelet based perspiration liveness check with fingerprint recognition," *Pattern Recognit.*, vol. 42, no. 3, pp. 452–464, Mar. 2009.

[25] G. L. Marcialis, F. Roli, and A. Tidu, "Analysis of fingerprint pores for vitality detection," in *Proc. 20th Int. Conf. Pattern Recognit.*, İstanbul, Turkey, Aug. 2010, pp. 1289–1292.

[26] S. Memon, N. Manivannan, and W. Balachandran, "Active pore detection for liveness in fingerprint identification system," in *Proc. 19th Telecommun. Forum (TELFOR)*, Nov. 2011, pp. 619–622.

[27] M. Espinoza and C. Champod, "Using the number of pores on fingerprint images to detect spoofing attacks," in *Proc. Int. Conf. Hand-Based Biometrics*, Nov. 2011, pp. 1–5.

[28] E. Marasco and C. Sansone, "Combining perspiration- and morphology-based static features for fingerprint liveness detection," *Pattern Recognit. Lett.*, vol. 33, no. 9, pp. 1148–1156, Jul. 2012.

[29] H. Choi, R. Kang, K. Choi, and J. Kim, "Aliveness detection of fingerprints using multiple static features," *Int. J. Comput. Inf. Eng.*, vol. 1, no. 4, pp. 893–898, 2007.

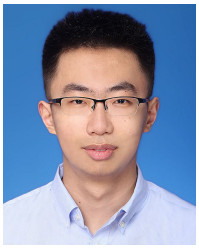
[30] L. F. A. Pereira, H. N. B. Pinheiro, G. D. C. Cavalcanti, and T. I. Ren, "Spatial surface coarseness analysis: Technique for fingerprint spoof detection," *Electron. Lett.*, vol. 49, no. 4, pp. 260–261, Feb. 2013.

[31] P. Coli, G. Luca Marcialis, and F. Roli, "Power spectrum-based fingerprint vitality detection," in *Proc. IEEE Workshop Autom. Identificat. Adv. Technol.*, Jun. 2007, pp. 169–173.

[32] L. Ghiani, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection by local phase quantization," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Hangzhou, China, Nov. 2012, pp. 537–540.

[33] K. Karampidis, M. Rousouliotis, E. Linardos, and E. Kavallieratou, "A comprehensive survey of fingerprint presentation attack detection," *J. Surveill., Secur. Saf.*, vol. 10, pp. 117–161, Jan. 2021.

[34] C. Wu, K. He, J. Chen, Z. Zhao, and R. Du, "Toward robust detection of puppet attacks via characterizing fingertip-touch behaviors," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 6, pp. 4002–4018, Nov. 2022.
[35] B. Tan and S. Schuckers, "Spoofing protection for fingerprint scanner by fusing ridge signal and valley noise," *Pattern Recognit.*, vol. 43, no. 8, pp. 2845–2857, Aug. 2010.
[36] P. Johnson and S. Schuckers, "Fingerprint pore characteristics for liveness detection," in *Proc. Int. Conf. Biometrics Special Interest Group (BIOSIG)*, Sep. 2014, pp. 1–8.
[37] Y. S. Moon, J. S. Chen, K. C. Chan, K. So, and K. C. Woo, "Wavelet based fingerprint liveness detection," *Electron. Lett.*, vol. 41, no. 20, p. 1112, 2005.
[38] T. Chang, C. Tsai, W. Tsai, C. Peng, and H. Wu, "A changeable personal identification number-based keystroke dynamics authentication system on smart phones," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 2674–2685, Oct. 2016.
[39] Y. Chen, J. Sun, R. Zhang, and Y. Zhang, "Your song your way: Rhythm-based two-factor authentication for multi-touch mobile devices," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2686–2694.
[40] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones," in *Proc. NDSS*, vol. 56, 2013, pp. 57–59.
[41] *Three-Dimensional Force System. JoVE*. Accessed: 5, Jan. 2024. [Online]. Available: https://www.jove.com/science-education/14234/three-dimensional-force-system
[42] (2023). *AS608 Fingerprint Reader Sensor Module With Cable*. [Online]. Available: https://www.rajguruelectronics.com/Product/254/AS608
[43] (2023). *STM32f407ZE Datasheet*. [Online]. Available: https://www.alldatasheet.com/datasheet-pdf/pdf/505003/STMICROELECTRONICS/STM32F407ZE.html
[44] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
[45] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proc. 12th Int. Conf. Pattern Recognit.*, vol. 1, Jun. 1994, pp. 582–585.
[46] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[48] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.
[49] A. Dogan and D. Birant, "A weighted majority voting ensemble approach for classification," in *Proc. 4th Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2019, pp. 1–6.
[50] U. Mangai, S. Samanta, S. Das, and P. Chowdhury, "A survey of decision fusion and feature fusion strategies for pattern classification," *IETE Tech. Rev.*, vol. 27, no. 4, p. 293, 2010.
[51] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, Jul. 1999, pp. 1–7.
[52] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.
[53] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 413–422.



Wenhao Wang received the B.S. degree in cyber science and engineering from Southeast University, Nanjing, China, in 2023, where he is currently pursuing the M.S. degree with the School of Cyber Science and Engineering.

His research interests include biometrics and physical-layer security.

Guyue Li (Member, IEEE) received the B.S. degree in information science and technology and the Ph.D. degree in information security from Southeast University, Nanjing, China, in 2011 and 2017, respectively.

From June 2014 to August 2014, she was a Visiting Student with the Department of Electrical Engineering, Tampere University of Technology, Tampere, Finland. She is currently an Associate Professor with the School of Cyber Science and Engineering, Southeast University, and a Visiting Scholar with the Tampere University of Technology, and Université Gustave Eiffel (ESIEE PARIS), Noisy-le-Grand, France. Her current research interests include wireless network attacks, physical-layer security solutions for 5G and 6G, secret key generation, radio frequency fingerprints, and reconfigurable intelligent surfaces.

Dr. Li was a recipient of the Young Scientist Awarded by the International Union of Radio Science (URSI), the Youth Science and Technology Prize of Jiangsu Cyber Security Association, and the A-Level Zhishan Scholar of Southeast University. She has been the Workshop Co-Chair of IEEE Conference on Vehicular Technology (VTC) from 2021 to 2022. She is currently serving as an Editor for IEEE COMMUNICATION LETTERS and an Associate Editor for *EURASIP Journal on Wireless Communications and Networking*.

Zhiming Chu received the B.S. degree from the Wuhan University of Technology, Wuhan, China, in 2022. He is currently pursuing the M.Sc. degree with the School of Cyber Science and Engineering, Southeast University, Nanjing, China.

His current research interests include physical layer security in wireless communications and wireless sensing.

Haobo Li (Member, IEEE) received the B.Eng. degree in electrical and electronic engineering from the University of Northumbria, Newcastle upon Tyne, U.K., in 2015, the M.Sc. degree in electrical and electronic engineering from the Communication and Signal Processing, University of Newcastle, Newcastle, NSW, Australia, in 2016, and the Ph.D. degree in electrical and electronic engineering from the University of Glasgow, Glasgow, U.K., 2021.

He is currently a Research Associate at the School of Physics and Astronomy, University of Glasgow. His research interests include radar and optical sensing for healthcare applications, multimodal sensing and sensor fusion algorithms, computational imaging, and applied machine learning.

Daniele Faccio received the bachelor's degree in physics from the University of Milano, Milan, Italy, in 1997, and the Ph.D. degree in photonics from the University of Nice Sophia-Antipolis, Nice, France, in 2007.

He was an Assistant Professor with the University of Insubria, Varese, Italy, from 2004 to 2010, an Associate Professor from 2010 to 2012, and then a Full Professor from 2012 to 2018 with Heriot-Watt University, Edinburgh, U.K. He is currently a Professor in quantum technologies with the University of Glasgow, Glasgow, U.K. His research interests include classical optics and quantum photonics, with interest in fundamental quantum physics and applications in healthcare.