The Self-Detection Method of the Puppet Attack in Biometric Fingerprinting

Guyue Li, Yiyun Ma, Wenhao Wang, Junqing Zhang and Hongyi Luo

***Abstract*—Fingerprint authentication has become a staple in securing access to personal devices and sensitive information in our daily lives, with the security level of such systems being paramount. Recent attention has been drawn to the puppet attack, a forced fingerprint unlocking scenario that exploits legitimate user fingerprints for unauthorized access. Traditional authentication methods are constrained by their reliance on additional sensors and are typically limited to static authentication scenarios, lacking versatility in dynamic or mobile contexts. In this study, we employ physical modeling to elucidate puppet attack, unraveling the distinctive stress patterns and points of application associated with forced interactions. By scrutinizing the physical alterations induced during such attacks, our investigation unveils discernible changes in the texture of fingerprints, specifically reflecting variations linked to different force patterns. Consequently, we introduce a detection system that operates without the need for external sensors, solely utilizing fingerprint images to extract texture features, thereby offering a broadly applicable solution. To address the challenge posed by the absence of puppet attack samples in existing datasets, we constructed a comprehensive database, incorporating a substantial number of puppet attack fingerprints collected from 70 volunteers aged between 20 and 75. This database facilitates a more robust detection of puppet attack. Our system demonstrates accuracy rates of 85.5%, 97.2%, 86.5%, and 78.1% across four distinct scenarios within our puppet attack database.**

***Index Terms*—Puppet attack, identification security, biometric system, support vector machine.**

I. Introduction

Automated recognition of behavioral and physiological characteristics of an individual is the core of biometric technology as the International Organization for Standardization described [1]. Fingerprint, as a kind of common and easily obtained physiological characteristic, has been utilized to reliably check the identity by analyzing the local feature extracted and increasing the efficiency of access to services. Currently, fingerprints are mainly captured by optical reflection, silicon crystal capacitive sensors, and ultrasonic scanning [2]. Among them, capacitive sensors are widely used. Specifically, the silicon sensor is employed as one polar plate of the capacitor and the finger pressed is employed as another polar plate of the capacitor. The capacitance difference between fingerprint ridges and valleys derived is used to generate the fingerprint patterns. Compared to optical reflection, a capacitive sensor is less susceptible to finger dirt and favorable to capturing more fingerprint details. Therefore, capacitive sensors-based fingerprint recognition systems are ubiquitously commercialized for intelligent door lock security, cell phone unlocking [3], and e-commerce payment [4].

However, concerns on the vulnerability of capacitive sensor based fingerprint recognition systems grow. Research shows that fingerprint authentication is vulnerable to attacks. In particular, presentation attacks employ artificial fingerprint replicas to counterfeit real fingers and spoof existing capacitive modules. Such attacks can circumvent a fingerprint recognition system security with a success rate of over 70% [5], [6]. Different from the presentation attack that passes the authentication with artificial crafts, there still exists risks for authentication systems [7]. ISO/IEC 30107-1:2016 [1] emphasizes "*involuntary reactions*" in the biometric authentication process, where the fingerprints that capacitive sensors fetched come from the clients, but the clients have no intention [8]. This kind of attack is defined as puppet attack, where an intruder violently pinches the client's finger and forces the client to unlock the verification system. Such attacks can happen in a robbery or terrorist event, which is abrupt for clients. Another possible scenario is that the client's child holds his/her fingers to unlock the phone [9]. Compared to presentation attacks, puppet attack employs fingerprints of legitimate clients rather than mimicry to pass the authentication system. Puppet attack's abruptness and authentication process legitimacy make it challenging to be detected. Existing presentation attack detection methods will not be able to resist the puppet attack [10]–[12]. Similarly, authentication systems utilizing only capacitive sensors, such as access control system of offices or labs, cars or computers activated by fingerprints, are all exposed to the potential risk of puppet attack. Therefore, it is necessary to propose a system that can defend against attack only with fingerprint images. As Fig. 1 illustrates, puppet attack may cause personal security risks, data leakage, and pecuniary losses in many scenarios. The widespread use of fingerprint authentication systems makes it more significant for system security.

The cell phone manufacturers and smart door locks in the mainstream market do not have defensive countermeasures against puppet attacks. Apple released attention aware features [13] which can intelligently check whether a client is

This work was supported by the National Natural Science Foundation of Jiangsu Province, China (No. BK20211160). The work was approved by the IEC for Clinical Research of Zhongda Hospital affiliated to Southeast University, the approval number is 2023ZDSYLL109-Y01. Manuscript received xxx; revised xxx; accepted xxx. Date of publication xxx; date of current version xxx. The review of this paper was coordinated by xxx. *(Corresponding author: Guyue Li.)*

G. Li, Y. Ma, W. Wang and H. Luo are with the School of Cyber Science and Engineering, Southeast University, Nanjing, 210096, China.

J. Zhang is with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom. (email: Junqing.Zhang@liverpool.ac.uk)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier xxx

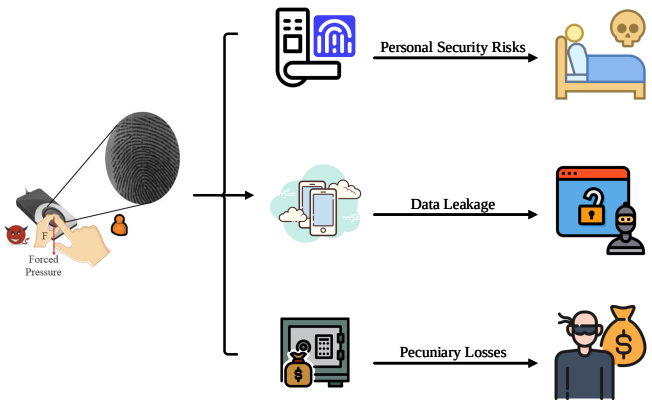


Fig. 1. Puppet attack scenarios and security risks.

paying attention to the device by capturing depth data and infrared images. It can recognize if a client's eyes are open and the attention is directed towards the device, making it hard for intruders to unlock devices without clients' knowledge. However, to the best knowledge of the authors, this is the only defensive countermeasure available in the mainstream market. Having said that, Apple's attention aware features only support phones with a 3D face recognition mechanism, which means fingerprint authentication is still unprotected. There have also been research efforts on attention aware needs. For example, the cellphone's built-in gyroscope is utilized to judge whether the pressing action is involuntary [7]. However, this method can hardly be employed on devices like smart door locks, which do not have built-in sensors other than capacitive fingerprint sensors.

The difficulty in detecting such attacks is to differentiate the subtle difference between forced fingerprints and unforced fingerprints, which are derived from resistance actions. To solve puppet attack problems, where devices are unlocked with biometric features but without clients' permissions or clients are aware of the attack but fail to resist, we first collected fingerprints and their corresponding pressure values. By observing and analyzing the data we collected, we found that pressure is not a good way for detection and would cost more burden for the system. Then, we found that texture would change under different stress distributions. Existing fingerprint datasets lack forced fingerprints. The ethical approval to collect our own dataset is needed. The ethical review response for our forced fingerprint collection was approved by the IEC for Clinical Research of Zhongda Hospital affiliated to Southeast University. The approval number is 2023ZDSYLL109-Y01, dated March 30, 2023. Approved by our Institutional Review Board, we established our own database called puppet attack database, which comprised 5,600 fingerprints from 70 volunteers and is divided into forced and unforced classes. These volunteers' ages are between 20 and 75. For forced and unforced fingerprints, we recorded images that successfully passed the authentication process. To simulate real-world authentication scenarios, the volunteers are required to unlock the devices with different pressing centers, rotation angles, and pressing pressure. As for forced conditions, the intruders are also asked to pinch clients' fingers to unlock in different positions. In the database, we can observe whether the attacked fingerprint images are different from the normal ones. By importing fingerprint samples that are attacked and training them with machine learning, we can solve the puppet attack detection problem. The contributions of this paper are summarized as follows:

- We physically model the puppet attack. By correlating the attack pattern with the physics of torque variations, thereby facilitating the detection of texture changes under duress. We demonstrates the feasibility of detecting subtle differences in attacks without the need for additional sensors.
- We introduce a novel method, SDM-PA, for defending against puppet attack. This method was validated using a dataset comprising 5,600 fingerprint images, covering individuals aged 20 to 75 to simulate real-world scenarios. Additionally, we explored the impact of varying negative sample rates on attack detection, determining the optimal ratio for effective detection.

The rest of the paper is arranged as follows. In Section II, we introduce the previous fingerprint-related datasets and research. In Section III, we present the details of the puppet attack in this paper. Section IV describes the proposed method that defends against puppet attack. We describe the testbed, experiment design and evaluation metrics in Section V. Section VI reports experimental results. Section VII represents the future work and Section VIII concludes this paper.

II. Related Work

There has been great progress of defence against fingerprint presentation attacks in the last few years. We summarize the existing fingerprint datasets and review the representative work on fingerprint attack detection.

A. Dataset

As for the fingerprint liveness dataset, LivDet2017 comprises three databases with different scanners. Among the three databases, image height is not constant in Database2 because it depends on the finger swipe way [14]. After that, the LivDet series is extended with LivDet2019, which consists of a never-seen-before material in the detection [15] and LivDet2023 [16].

However, existing fingerprint datasets have limitations: 1) There is no protocol of age distribution. The limited age distribution makes the method difficult to promote in large scale volunteer scenarios. 2) They do not have forced fingerprint samples, thus we can hardly extract correspondingly puppet attack features for detection.

B. Presentation Attack Defend Methods

Biometric fingerprint authentication is prone to presentation attacks, where attackers utilize artificial replicas to intrude on the system [6]. The existing defence methods can be divided into two categories: hardware method and software method [20]–[22].

TABLE I
COMPARISON WITH OTHER AUTHENTICATION SYSTEM.

Paper	Overview	Features	Classification Method
[7]	Using fingertip behaviors measured by mobile phones' acceleration and the rotation angle.	Time-domain and frequency-domain features from mobile phones' accelerations and rotation angles.	OC-SVM
[17]	Employing finger movement to authenticate clients.	First touch position, first moving direction, moving distance, etc. from multi-touch screen, accelerometer and gyroscope.	Binary SVM
[18]	Utilizing data from wearable devices to authenticate clients.	Signals from calorie burn, step counts and heart rate.	Binary SVM
[19]	Fingerprint authentication aided by knuckle images acquired by camera phtography.	Fingerprint and knuckle images.	Mask R-CNN
SDM-PA	Defensing against puppet attack during fingerprint authentication.	Single fingerprint image.	SVM

The hardware-based methods utilize physiological features [23], such as odor [24], temperature [25], pulse oximetry [26] and blood pressure [27]. However, to fetch these inherent biometric signals, external sensing devices are employed, which are always customized and auxiliary. Moreover, the time expenses of the authentication process would be longer and cause a bad user experience. Furthermore, hardware update difficulties emerge once the system is successfully attacked. In general, hardware-based methods are inflexible, and not suitable for large-scale commercial use [28].

In contrast, software-based methods analyze the salient features from fingerprint images captured by sensors in order to achieve the detection goal, which is more favorable thanks to its convenience and low cost. Extracted features from sensors can be broadly classified into the following three types:

- Anatomical features, including pore locations and pore distribution [29]. These features provide valuable information about the individuality of the fingerprint pattern.
- Physiological features, the most typical one is sweat-pores based feature [30], [31]. Subtle variations in the fingerprint pattern due to changes in moisture levels can be indicative of an individual's physiological state, thus ensuring robustness against potential attempts to deceive the biometric system.
- Texture-based features. The utilization of texture features allows the system to focus on unique texture characteristics, such as ridges and furrows, which are crucial for fingerprint recognition.

Among them, the texture feature has become one of the most widely applied features, because methods based on singular points, such as pore locations and sweat-pores, may result in wrong classification. The model performance is intrinsically limited by the state of the fingerprint skin surface, such as abrase, wrinkle, and perspiration [32]. The basic idea behind texture features-based methods is that abnormal fingerprint images have different texture distributions, despite it is indistinguishable from human eyes. Nikam [33] employed Local Binary Pattern (LBP) histograms based on the gradient for the application on fingerprint live detection for the first time, where texture details are generated by the comparison of the center pixel value and its adjacent pixels, and proves that only one image is sufficient to defend spoof attacks. Ghiani *et al.* [34] presented a method based on local phase quantization (LPQ) to discriminate the differences between live samples from fake samples since the loss of information may occur as a result of the replica fabrication process. Gragnaniello *et al.* [35] proposed the weber local descriptor (WLD), which can extract two-dimensional histogram features from square bipartite and differential exciatation. Furthermore, Gragnaniello *et al.* [36] proposed local contrast phase descriptor (LCPD) for detection tasks, combining gradient with local phase information together. Wasnik *et al.* [37] employed LBP, Histograms of Oriented Gradients (HOG) and Binarized Statistical Image Features (BSIF) on Maximum Filter Response (MFR) images and achieved a negative Presentation Classification Error Rate (BPCER) of 1.8% for print photo attacks.

C. Biometric Authentication Methods

Some research employ clients' inherent habits and unique behaviors to defend attacks. Wu *et al.* highlights puppet attack detection [7], where a client's finger is put on the sensor. To defend against the attack, the authentication system is built with clients' fingertip-touch behaviours, which are presented as acceleration and rotation angle of mobile devices. However, the behavior patterns are captured by additional sensors and require the clients to do additional actions(e.g., pattern locks, signature), which increases the burden of the authentication process. The Table. I compares in detail of the design goals, features and classification methods.

III. PUPPET ATTACK

In this section, we introduce the puppet attack in detail, including the attack model and the feasibility of pressure-based detection.

A. Modeling of Attack

As ISO/IEC 30107 describes [1], the puppet attack is that an attacker forces the involuntary client's finger on the fingerprint sensor to gain access via the fingerprint authentication system. Fig. 2 exemplifies the scenario where the client unlocks the cellphone, where (a) is the normal state, (b) is the attack state. The gray hand represents the attacker, while the red index finger belongs to the client. The attacker is gripping the hand of the client and forcing the victim to unlock the phone. F_{xy} and F_z denote the decomposition force on the XY plane and along the Z axis, respectively.

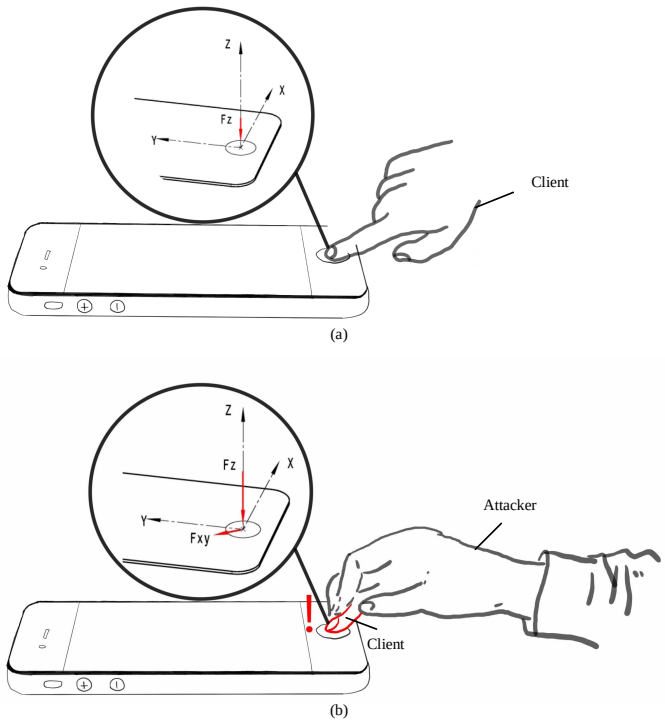


Fig. 2. Interaction force during the attack. (a) is the normal state, (b) is the attack state.

To gain a deeper understanding of the impact of the puppet attack, it is imperative to consider the concept of Von Mises stress, which is a method utilised to measure the distribution of internal mechanical stress within an object. With the stress distribution calculated by Von Mises stress analysis, we can assess the extent of the attack's impact on fingerprint images. Such analysis assists in deepening the understanding of how attackers distort fingerprint images through the application of external forces and enable unauthorized access.

As a consequence, we perform a finite element analysis (FEA), which is based on generative part structural analysis of CATIA V5. The von Mises stress, τ, is calculated as:

$$\tau = \sqrt{\frac{1}{2}\left[(\tau_x - \tau_y)^2 + (\tau_y - \tau_z)^2 + (\tau_z - \tau_x)^2\right]}, \quad (1)$$

where τ_x, τ_y and τ_z denote three orthogonal stresses.

We define θ as the tilt angle between the finger axis and the plane and build the von Mises stress analysis of the plane with the generative part structural analysis of CATIA. Fig. 3(a) shows the stress analysis under an unforced situation where F is $3N$ and θ is $90°$; the stress is concentrated in the central part of the sensor. In contrast, Fig. 3(b) illustrates the stress analysis of a forced situation where F is $30N$ and θ is $30°$; the stress is concentrated at the edge of the sensor.

We define a client C_A is forced by an intruder I_A and a client C_B is forced by an intruder I_B. Both C_A and I_B are female. Also, C_B and I_A are male. At the same time, we collect 50 fingerprint images per state, which means C_A and C_B are both collected 100 fingerprint images. The fingerprint data collection system details will be described in Section V-A.

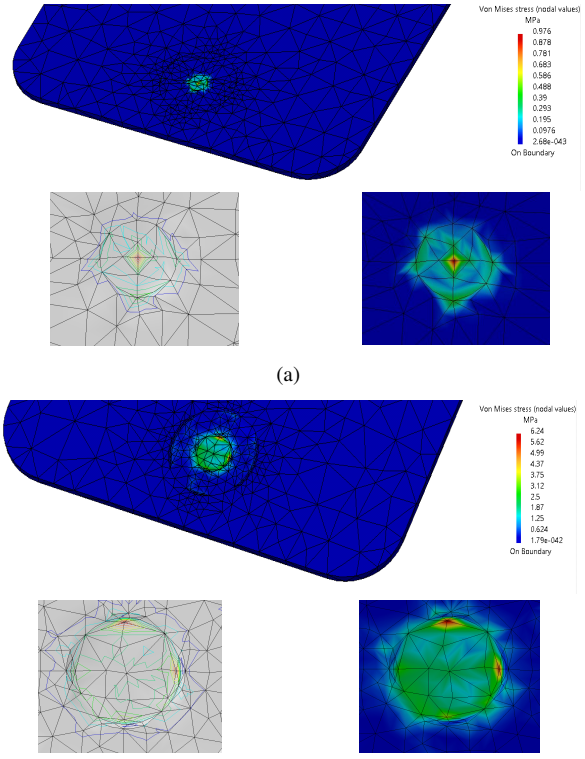


Fig. 3. Comparison of stress distributions of (a) a normal state; (b) an abnormal state.

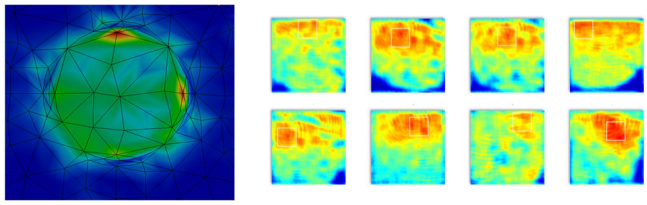


Fig. 4. The stress distribution and heat maps of forced fingerprints.

Then, we drew the heat map of fingerprint images from C_A when he was forced. Part of these images are shown in Fig. 4. We find that there is a correlation between the texture of the fingerprints and their stress distribution. As Fig. 3 and Fig. 4 show, the stress of the forced fingerprints is mainly located at the edges. At the same time, the heat maps of the fingerprint images also show the corresponding characteristics. Then we calculate the statistical results. We establish a two-dimensional rectangular coordinate for the image, with the lower left corner at (0,0) and upper right corner at (160,160). Taking only one maximum pressure center. The pressure cluster center of 100 images is at (117,138), which is similar to stress distribution results.

B. *Feasibility of Pressure-based Detection*

With the stress distribution in Section III-A, there is a clear difference between forced and unforced fingerprints. This section will validate whether forced fingerprints can be detected by pressure sensors.

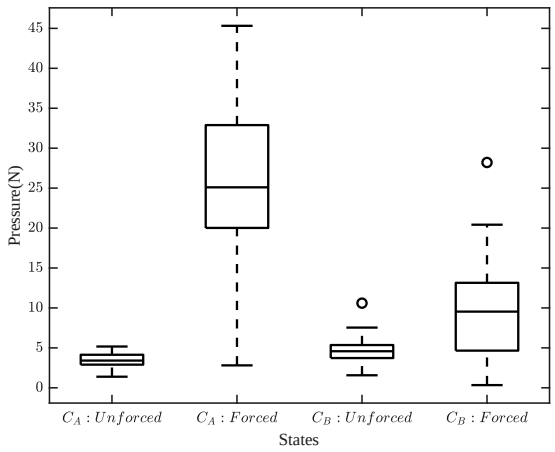


Fig. 5. Fingerprint pressure under different states.

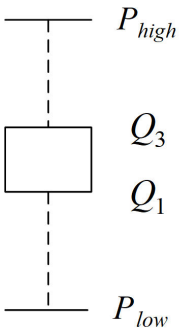


Fig. 6. Numerical metrics on boxplots.

We recorded the pressure of the fingerprints under forced state and unforced state. We collected 50 fingerprint images per state per client. As described in section III-A before, client C_A and intruder I_B are female. Client C_B and the intruder I_A are male. We draw Fig. 5, a boxplot based on clients' pressure under different states [38]. As a kind of statistical chart, boxplots clearly display the distribution of a set of data. As it illustrates, we can conclude that under a normal state, the pressing pressure between different genders is minimal, but male's pressure is a bit more than female's in general. Also, when the attacker is male, the press would face a significant increase.

In order to distinguish between fingerprints' states with the boxplot, we define the range of the normal value as

$$[P_{\text{low}}\ , P_{high}] \tag{2}$$

where P_{low} is the minimum value and P_{high} is the maximum value as Fig. 6 shows. The lower quartile, Q_1, upper quartile, Q_3, and interquartile range, IQR, are calculated as

$$\begin{aligned} P_{high} &= Q_3 + 1.5IQR, \\ P_{low} &= Q_1 - 1.5IQR, \\ IQR &= Q_3 - Q_1, \end{aligned} \tag{3}$$

respectively.

We define $Q_{1,n}$ as the Q_1 under the normal state, $Q_{1,ab}$ as the $Q1$ under the abnormal state. The pressure threshold P_{th} is defined as:

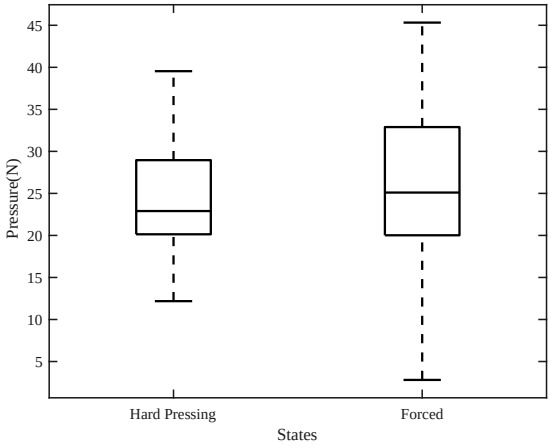


Fig. 7. Fingerprint pressure under voluntary and involuntary states.

$$P_{th} = \frac{Q_{1,ab} + Q_{3,n}}{2}. \tag{4}$$

For client C_B, the P_{th} is $5.01N$. That is, the attack that is less than $5.01N$ can hardly be differentiated. With this method, the final accuracy for client C_B is 70.7%.

As for C_A, Fig. 5 clearly shows that her fingerprint states can be easily distinguished with pressure difference. As a result, we asked her to press the fingerprint sensor in a hard way voluntarily. As Fig. 7 shows, there exists a large overlap in the IQR between the two states. There would be a high rejection rate using pressure as the discriminatory criteria.

To conclude, employing pressure as a threshold for forced fingerprint pressing attack detection can succeed in some situations. However, there still exist many drawbacks. First, this method requires individual threshold calculation, which can be very burdensome in the case of multiple clients. Besides, the pressure threshold-based discrimination method is difficult to work when the clients actively press hard. A high rejection rate would cause extra usability burdens on users. Meanwhile, the texture would change with the stress distribution. Therefore, using texture features-based image classification is more reasonable and convenient, and it is necessary to build the corresponding dataset. Details of the establishment of dataset would be introduced in Section V-A.

IV. Proposed Method

In the field of puppet attack detection, the lack of public puppet attack datasets makes it difficult to evaluate. Consequently, we first establish a database. Then, image preprocessing and feature extraction are utilized for the model training. The framework of the defending method is shown in Fig. 8. In the training stage, our research involves acquiring fingerprint images from users in both forced and unforced states. We apply the Otsu image segmentation method to effectively segment the fingerprint images into foreground and background regions, facilitating subsequent feature extraction. Following the image segmentation, we employ extraction techniques such as gray level difference statistics, histogram of oriented gradients and local binary patterns. The extracted

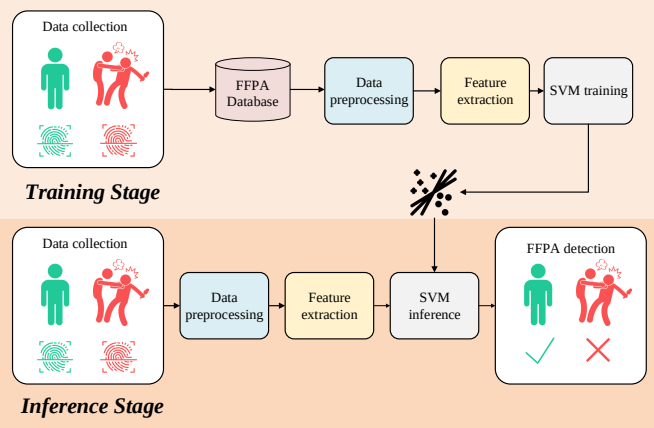


Fig. 8. Framework of the defending method.

features are encoded as vector representations and utilized as the input for the SVM classifier. By performing feature extraction and SVM training on fingerprint images in both normal and foerced states, we obtain a model for classifying fingerprint images. In the inference stage, the trained SVM model is utilized to classify new fingerprint images.

A. Data Preprocessing

An unsupervised and nonparametric method called Otsu algorithm [39] is used for image segmentation, which computes the threshold of every image since it can generate the best segmentation threshold automatically according to the image itself. It is considered to be the best algorithm for threshold selection in image segmentation, because it is simple to calculate and not affected by brightness and contrast. Therefore, it has been widely used in digital image processing.

Otsu segmentation divides the image into background and foreground according to the gray characteristics of the image. Because variance is a measure of gray distribution uniformity, the greater the inter class variance between background and foreground, the greater the difference between the two parts of the image. When part of the foreground is incorrectly divided into the background or vice versa, the difference between the two parts will become smaller. Therefore, the segmentation that maximizes the variance between classes means the minimum misclassification probability.

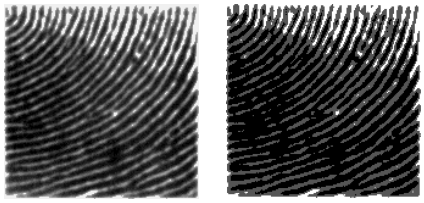
Let a $M \times N$ pixels image be represented in 256 gray levels $[0, 1, 2, \ldots, 255]$. n_i represents the number of pixels whose gray level is i. The sum of pixels in the image can be represented in

$$n = \sum_{i=0}^{255} n_i. \tag{5}$$

The probability that the gray level of a pixel is i is

$$p_i = \frac{n_i}{n}. \tag{6}$$

A threshold k, $0 < k < 255$, can be used to separate the input image into two classes, C_1 and C_2, where C_1 consists of pixels with gray values in the range $[0, k]$ and C_2 consists of pixels



(a) Original (b) Otsu

Fig. 9. Original image of unforced fingerprints and image processed by Otsu.

with gray values in the range $[k+1, 255]$. The probability of a pixel is classified in C_1 is,

$$P_1(k) = \sum_{i=0}^{k} p_i. \tag{7}$$

The probability of a pixel is classified in C_2 is

$$P_2(k) = \sum_{i=k+1}^{255} p_i. \tag{8}$$

The mean gray values of C_1 are

$$m_1(k) = \sum_{i=0}^{k} iP\left(i \mid C_1\right) = \frac{1}{P_1(k)} \sum_{i=0}^{k} ip_i. \tag{9}$$

Similarly, the mean gray values of C_2 are

$$m_2(k) = \sum_{i=k+1}^{255} iP\left(i \mid C_2\right) = \frac{1}{P_2(k)} \sum_{i=k+1}^{255} ip_i. \tag{10}$$

The average gray value of pixels with gray levels of 0 to k is

$$m_k = \sum_{i=0}^{k} ip_i. \tag{11}$$

The global mean of the image is given by

$$m_G = \sum_{i=0}^{255} ip_i = P_1(k)m_1(k) + P_2(k)m_2(k). \tag{12}$$

The interclass variance is defined as

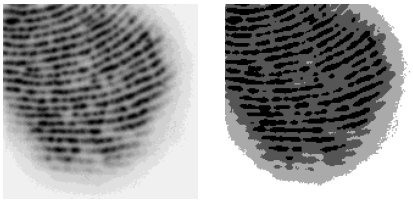
$$\begin{aligned} \sigma_B^2(k) &= P_1(k)\left(m_1 - m_G\right)^2 + P_2(k)\left(m_2 - m_G\right)^2 \\ &= P_1(k)P_2(k)\left(m_1 - m_2\right) \\ &= \frac{\left(m_G P_1(k) - m\right)^2}{P_1(k)\left(1 - P_1(k)\right)}. \end{aligned} \tag{13}$$

The bigger the mean difference between m_1 and m_2, the bigger σ_B^2, which is a measure of separability between classes. Therefore, the optimal threshold k^* is needed to maximize σ_B^2, that is:

$$k^* = \arg\max \sigma_B^2(k). \tag{14}$$

Using the optimum threshold value k^*, the image chosen can be divided into two classes. In this way, pressing area is segmented and the specific value can be calculated.

The original fingerprint images and the ones processed by Otsu of unforced and forced scenarios are shown in Fig. 9 and Fig. 10, respectively.



(a) Original (b) Otsu

Fig. 10. Original image of forced fingerprints and image processed by Otsu..

B. Feature Extraction

We proposed the utilization of three feature extraction methods, namely the gray level difference statics (GLDS), local binary patterns (LBP), and histogram of oriented gradients (HOG). The GLDS method focuses on quantifying the variations in grayscale intensities, providing valuable insights into local contrast and texture patterns. LBP excels at ridge patterns and texture patterns, which are crucial for accurate recognition. Furthermore, HOG plays a significant role in capturing the directional information of ridge patterns, including distribution and orientations.

1) Feature Extraction Based on GLDS: GLDS is an algorithm based on the estimation of the probability of pixel pairs at a given distance [40]. By calculating the gray difference histogram, the features from GLDS reflect an image's texture characteristics. The gray histogram of the texture region uses the features of contrast, mean, entropy and angular second moment to describe the texture characteristics.

Set (u, v) as a point in the image, the gray level difference between it and the point $(u + \Delta u, v + \Delta v)$ is:

$$\Delta g_{(u,v)} = g_{(u,v)} - g_{(u+\Delta u, v+\Delta v)} \tag{15}$$

where $g_{(u,v)}$ is the gray value of point (u, v). By moving (u, v) through the image, the number of each $\Delta g_{(u,v)}$ is counted as $n_{\Delta g}$. The probability of each $\Delta g_{(u,v)}$ is calculated:

$$p_{\Delta g} = \frac{n_{\Delta g}}{n_{\Delta}}, \tag{16}$$

where n_{Δ} represents the sum of gray difference in the image. Texture characteristics are closely related with p_{Δ_g}. The $p_{\Delta i}$ would change rapidly along with the variation of g in the coarse part of the image. Let L be all possible values of gray difference value. The common gray level difference based features, i.e., contrast, mean, entropy and angular second moment, are computed according to the following formulas:

$$con = \sum_{\Delta g=0}^{L} (\Delta g)^2 p_{\Delta g}, \tag{17}$$

$$mean = \frac{1}{L} \sum_{\Delta g=0}^{L} (\Delta g) p_{\Delta g}, \tag{18}$$

$$ent = -\sum_{\Delta g=0}^{L} p_{\Delta g} \log_2(p_{\Delta g}), \tag{19}$$

$$asm = \sum_{g=0}^{L} (p_{\Delta g})^2. \tag{20}$$

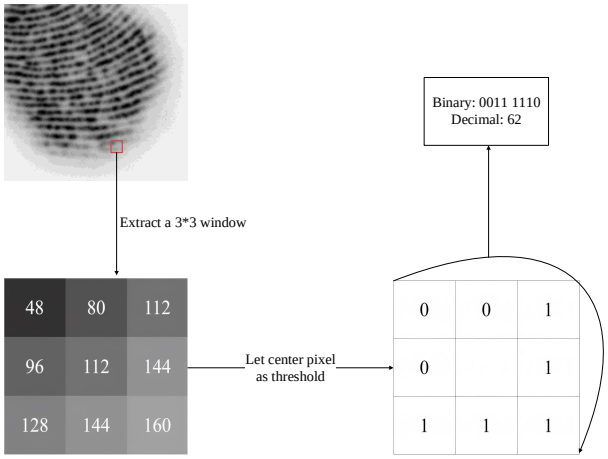


Fig. 11. LBP Feature extraction algorithm.

The contrast directly reflects the contrast of the brightness of a pixel value and its domain pixel value. If the element deviating from the diagonal has a large value, that is, the image brightness value changes rapidly, and the contrast will have a large value. Entropy is a measure of the amount of information an image and represents the non-uniformity or complexity of the texture in the image.

2) Feature Extraction Based on LBP: LBP is a kind of method used to describe the local texture features of an image. The LBP operator is a $3 * 3$ window. As shown in Fig. 11, let the center pixel as (u_c, v_c) and its gray level value is g_c. When the surrounding pixels' gray level value, g_s, is greater than the center one, it would be marked as 1. Otherwise, it would be 0. The LBP coding process can be given as

$$LBP\,(u_c, v_c) = \sum_{j=0}^{7} 2^j \cdot s(g_s - g_c), \tag{21}$$

where j is the number of adjacent pixels and $s(t)$ is a symbolic function, defined as

$$s(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{else} \end{cases}. \tag{22}$$

As a result, an ordered 8-bit binary number will be produced and used to reflect texture information. After the LBP pattern of each pixel is calculated, a histogram is built to represent the texture image.

3) Feature Extraction Based on HOG: HOG is also a method used for object detection with the information of gradient distribution and edge directions [41]. It is implemented by dividing an image into many regions called cells and computing a histogram of gradients within the cell. These histograms will finally be combined to form a feature descriptor. The formula for extracting HOG features is as follows:

$$G(u, v) = \sqrt{G_H(u, v)^2 + G_V(u, v)^2}, \tag{23}$$

where $G_H(u, v)$ and $G(u, v)$ represent the horizontal and vertical direction gradient at the pixel point (u, v) respectively.

The gradient direction at the pixel point (u, v) can be calculated as

$$\alpha(u,v) = \tan^{-1}\left(\frac{G_V(u,v)}{G_H(u,v)}\right). \tag{24}$$

Orientation bins refer to the discrete divisions of the angle of gradient directions. In the context of HOG, the range of gradient angles, typically from 0° to 180° (unsigned gradients), is divided into a predetermined number of bins. The choice of nine bins is standard, which discretizes the gradient directions into 20 degree intervals. Each bin corresponds to a range of angles, and the gradient magnitudes are accumulated in their respective bins based on the gradient direction of each pixel:

$$\text{Bin}(k) = \sum_{u,v \ \ \alpha(u,v)\in\text{Range}(k)} G(u,v), \tag{25}$$

where $\text{Bin}(k)$ represents the k-th bin in the histogram, $\text{Range}(k)$ is the range of angles that fall into the k-th bin, and the sum is taken over all pixel positions (u, v) in the cell for which the gradient direction $\alpha(u, v)$ is within the range corresponding to that bin.

Once the histograms for all cells are computed, they are normalized over larger, spatially connected regions called blocks. This normalization step is crucial for improving the invariance to changes in illumination and shadowing. The final HOG feature descriptor for the image is the concatenated vector of these normalized histograms from all of the blocks.

For the fusion of features, LBP and HOG are catenated to make up a one-dimensional vector.

C. Texture Classification Based on SVM

SVM is initially proposed to solve binary classification problems [42]. However, it is also found to be useful for nonlinearly separable cases. SVM can convert the classification problem to a quadratic programming problem comparing to other machine learning algorithms. The local minimum point problem can be avoided as the result of global optimal solution, comparing to neural networks.

Suppose training dataset $\{(x_t, y_t)\}_{t=1}^{N}$ where $x \in R^d$ represent the input, and $y \in \{1, -1\}$ is the training label that represents the output of the classification. To separate the two samples correctly, a hyperplane, $w^T x + b = 0$, is needed to find. Constrained by the Lagrange function, the optimal classification function is obtained by

$$\begin{aligned} f(x) &= \text{sgn}((w \cdot x) + b) \\ &= \text{sgn}\left(\sum_{t=1}^{N_v} (a_t y_t \phi(x) \cdot \phi(SV_t) + b)\right), \end{aligned} \tag{26}$$

where N_v is the number of the support vector obtained from training stage, a_t is the Lagrange coefficient, x is the input sample to be classified, SV_i is the training samples chosen as the support vectors in training steps, y_t is the class label of SV_t, w is the normal vector of the hyperplane, b is the hyperplane offset, $\phi(x)$ is the feature map function. For a nonlinear problem, the kernel function is used to transform the inner product of the high dimensional space into the inner product kernel function of the original space:

$$f(x) = \text{sgn}\left(\sum_{t=1}^{N_v} a_t y_t \mathcal{K}(x, SV_t) + b\right) \tag{27}$$

where $\mathcal{K}(\cdot, \cdot)$ is the kernel function.

Kernel function is the kernel principle and implementation of SVM in achieving nonlinear algorithm without the increment of complexity. For kernel function $\mathcal{K}(v_1, v_2)$, there are four common forms in most cases: linear kernel function, polynomial kernel function, radial basis function (RBF), and sigmoid kernel function.

In this article, the polynomial kernel function and RBF are applied. The formula of the polynomial kernel function is

$$\mathcal{K}(v_1, v_2) = [(v_1 \cdot v_2) + 1]^q, q > 0. \tag{28}$$

And the formula of RBF is

$$\mathcal{K}(v_1, v_2) = \exp\left(-\|v_1 - v_2\|^2\right)/2\sigma^2\right). \tag{29}$$

In (28) and (29), v_1 and v_2 is the input of the kernel function. In this research, v_1 represents the the number of support vectors in the SVM model, v_2 refers to the fingerprints to be classified. q and σ is the SVM model parameters obtained in the training stage.

Identifying the forced fingerprints and unforced fingerprints is the goal of the article. The texture features mentioned above are chosen as the inputs of the SVM model.

V. Testbed and Dataset Construction

A. Testbed

We employed the BM2166 capacitive sensor and STM32F407ZET6 microcontroller [43] as the fingerprint extraction platform, shown in Fig. 12. The platform has moderated imaging velocity, which meets the demand of capturing abnormal fingerprint details. When a finger rolls or slides on the capacitive sensor, sliding traces would be shown on the image. BM2166 capacitive fingerprint extraction module is used to fetch the fingerprint images.

The size of the collected fingerprint image is $8mm * 8mm$, the image pixel size is $160 * 160$, the resolution (DPI) is 508, the working temperature is from $-20^\circ C$ to $+40^\circ C$ and the working relative humidity is from 40% to 85%.

B. Data Acquisition

The dataset consists of a total of 5,600 fingerprint samples from 70 volunteers during three weeks. These volunteers' ages range from 20 to 75 and all have different pressing habits. Some of them get used to unlocking with the sides of the finger, while others prefer the center. Also, the roughness of the finger surface varies from volunteer to volunteer.

We take four kinds of scenarios into consideration.

- Scenario S1 - The Complete Dataset: This scenario encompasses the entire collection of fingerprint samples, providing a baseline for our system's performance across all types of puppet attack. It serves as a comprehensive set to gauge the overall effectiveness of the detection algorithm.

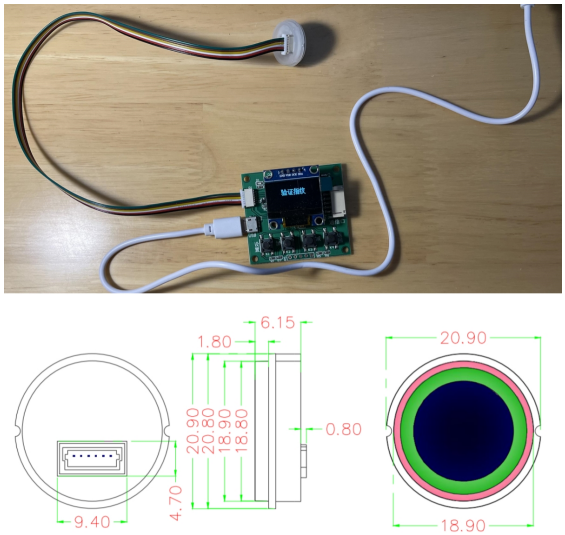


Fig. 12. Fingerprint extraction system and BM2166 drawing.

- Scenario S2 - Direct Pressing: In this scenario, fingerprints are collected from volunteers pressing their fingers directly down on the sensor. This represents the most common and straightforward use case, where the authentication process is assumed to be under duress.
- Scenario S3 - Angled Pressing: Fingerprints in this scenario are obtained when volunteers press their fingers at a 45-degree angle to the left or right. This simulates a situation where an attacker might not be able to exert pressure uniformly, thereby creating a more complex pattern to analyze and detect.
- Scenario S4 - Side Pressing: This scenario involves fingerprints taken from the side of the volunteers' fingers. This simulates an attack where the attacker uses the side of the victim's finger to avoid detection, presenting a subtler and possibly more challenging form of forced authentication to recognize.

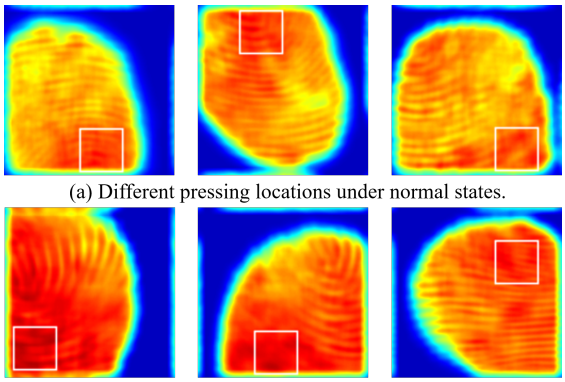
By dividing the dataset into these scenarios, we aim to create a nuanced understanding of how different forced pressing techniques might affect the authentication process. This approach allows us to tailor the detection algorithm for specific types of attacks and to ensure that our system is capable of identifying forced authentications with high precision across a range of different conditions.

The number of images under different scenarios is illustrated in Table II. There are two states of the fingerprints: normal state and abnormal state. Considering there are many forced forms in real life, we set three degrees of freedom as follows:

- Location of the Pressing Center. No matter the client is forced to press the fingerprint or not, there must exist an area of the biggest pressure. We let the volunteers press their fingers on different areas of the sensor, which are shown in Fig. 13.
- Rotation Angle. In a real-world scenario, clients can unlock devices with different pressing angles, e.g., by pressing straight or rolling with an angle. As is illustrated in Fig 14, the volunteers were asked to unlock devices with many directions, which led to different flow directions of the fingerprints.
- Pressure. In most instances, the pressure would be extremely high when the client is under danger. However, the pressure might not be that high if the client try to resist the intruder. In order to simulate more conditions in real life, the volunteers are asked to imitate the scenario above and we use a histogram to visualize it. The histogram under different states are illustrated in Fig. 15.

TABLE II
NUMBER OF IMAGES UNDER DIFFERENT SCENARIOS.

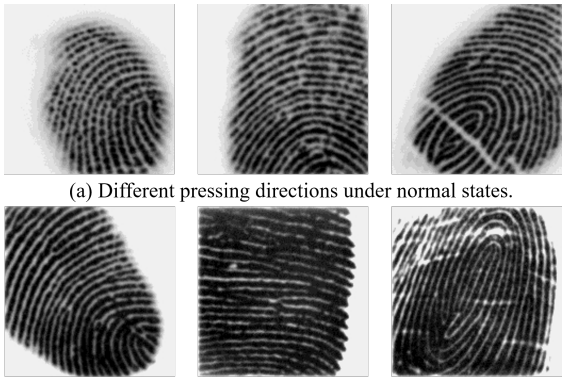
Scenario	Abnormal	Normal
Scenario S1 - All Unlock	5,600	5,600
Scenario S2 - Straight Unlock	1,400	1,400
Scenario S3 - 45 Degree Unlock	1,400	1,400
Scenario S4 - Side Unlock	2,800	2,800



(a) Different pressing locations under normal states.

(b) Different pressing locations under abnormal states.

Fig. 13. Location of the pressing center.



(a) Different pressing directions under normal states.

(b) Different pressing directions under abnormal states.

Fig. 14. Rotation angle of fingerprints.

VI. Experimental Evaluation

A. Experiment Design

We use four texture feature extraction methods and two SVM kernel functions to get the optimized algorithm for solving the puppet attack classification problem. GLDS, LBP, HOG, and LBP-HOG-fusion features are used. As for the

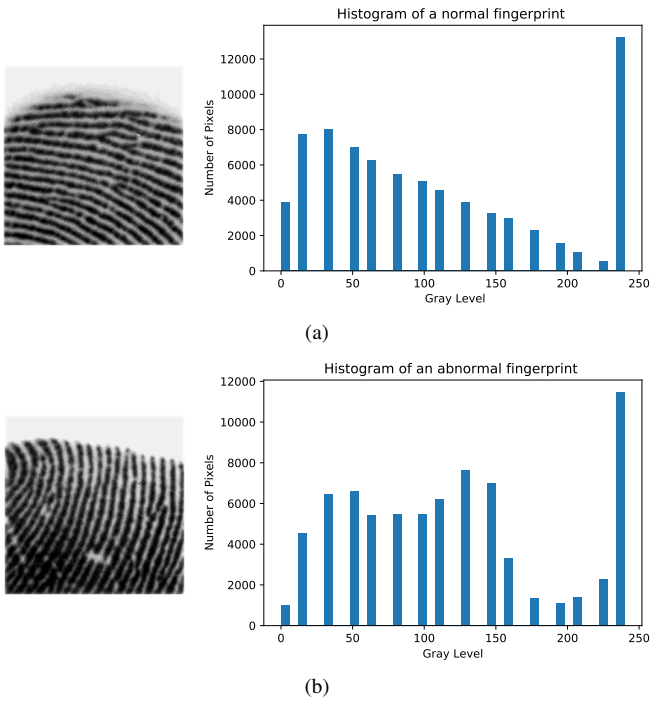


Fig. 15. Histogram under different states. (a) Histogram of a normal fingerprint under 106N pressure. (b) Histogram of an abnormal fingerprint under 106N pressure.

fusion of LBP and HOG, the concatenation of two kinds of features extracted is calculated as the input.

The puppet attack dataset is comprised of forced fingerprints and unforced fingerprints. Then, the identification performance of the texture features is evaluated with two different kernel functions: the polynomial kernel function and RBF. Classifiers are trained to classify the texture features into two classes: normal fingerprints and abnormal fingerprints. The SVM is investigated using 10-fold cross-validation in order to find the best parameters. By reading the test set samples' feature vectors, the trained classifier analyzes the test set and outputs the corresponding prediction results. The classification accuracy is thus obtained by comparing prediction labels and actual test labels.

B. Evaluation Metrics

Faced with a classification problem, the actual category and prediction category of the samples will produce different combinations. For a binary classification problem, results are divided into positive and negative classes. In the actual classification, there are four situations:

- True Positive (TP): Classifying a normal fingerprint correctly.
- False Positive (FP): Classifying an abnormal fingerprint as a normal fingerprint.
- True Negative (TN): Classifying an abnormal fingerprint correctly.
- False Negative (FN): Classifying a normal fingerprint as an abnormal fingerprint.

The performance evaluation indexes in this research include false positive rate (FPR), recall, precision, F1-score and accuracy. FPR represents the probability that the attack samples incorrectly classified as positive ones, defined as

$$\text{FPR} = \frac{FP}{FP+TN}. \tag{30}$$

Recall is the probability of negative samples being predicted correctly, given as

$$\text{Recall} = \frac{TP}{TP+FN}. \tag{31}$$

Precision is the ratio of negative samples predicted correctly among all samples which are predicted as negative,

$$\text{Precision} = \frac{TP}{TP+FP}. \tag{32}$$

Ideally, there should be high recall and precision values as well as low FPR values at the same time, which may not be achieved in most cases. Therefore, F1-score is introduced to consider the harmonic value of precision and recall comprehensively, defined as

$$\text{F1-score} = \frac{2}{\left(\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}\right)}. \tag{33}$$

The accuracy rate is the most commonly used metric to evaluate the closeness of the measurement value of a quantity to its true value. In this study, it gives the percentage of the samples classified correctly, defined as

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}. \tag{34}$$

Making choices according to specific situations is necessary. In general search situations, improving the accuracy under the condition of ensuring the recall rate is needed. While in other cases, precision is much more important, In a real biometric identification scenario, our goal is to avoid intruders, which means there should be as few abnormal fingerprints identified as normal as possible. On the mathematical aspect, low FPR, high accuracy, and high precision are much more needed. If possible, a high recall and F1-score are also needed.

C. Analysis of the Defend Methods

We take four kinds of scenarios into consideration, compromising of straight unlock (S2), 45-degree unlock (S3), side unlock (S4) and all of the above (S1). Table III, Table IV, Table V, and Table VI show the forced fingerprint classification accuracy, FPR, recall, precision, and F1-score on puppet attack dataset under scenario S1, scenario S2, scenario S3 and scenario S4, with features of GLDS, LBP, HOG and the fusion of LBP and HOG respectively. The table results are organized according to kernel function and features extracted. Results are shown after 10-fold cross-validation.

TABLE III
RESULTS FOR SCENARIO S1 - ALL UNLOCK

Features	Kernel Function	Accuracy	FPR	Recall	Precision	F1-Score
GLDS	Polynomial	68.5	28.8	65.2	65.3	65.2
	RBF	65.5	30.2	60.7	63.8	62.2
LBP	Polynomial	80.9	21.1	82.9	77.7	80.2
	RBF	69.1	47.5	85.7	64.3	73.4
HOG	Polynomial	78.7	24.0	81.5	77.2	83.9
	RBF	66.8	37.1	70.8	65.6	68.1
LBP-HOG	Polynomial	85.5	7.7	78.7	91.1	84.4
	RBF	72.9	8.5	54.2	86.5	66.7

TABLE IV
RESULTS FOR SCENARIO S2 - STRAIGHT UNLOCK

Features	Kernel Function	Accuracy	FPR	Recall	Precision	F1-Score
GLDS	Polynomial	74.6	28.3	77.6	73.3	75.4
	RBF	72.5	29.9	74.9	78.6	73.3
LBP	Polynomial	97.1	1.3	95.6	98.7	97.1
	RBF	86.9	12.9	86.7	87.1	86.9
HOG	Polynomial	95.9	1.4	93.1	98.5	95.7
	RBF	80.2	13.3	73.7	84.7	78.8
LBP-HOG	Polynomial	97.2	1.0	95.4	99.0	97.2
	RBF	86.7	1.2	85.4	87.7	86.5

TABLE V
RESULTS FOR SCENARIO S3 - 45-DEGREE UNLOCK

Features	Kernel Function	Accuracy	FPR	Recall	Precision	F1-Score
GLDS	Polynomial	69.8	32.0	72.4	69.4	70.9
	RBF	62.4	34.7	59.6	63.2	61.3
LBP	Polynomial	86.6	14.3	87.4	86.0	86.7
	RBF	71.7	37.6	81.0	68.3	74.1
HOG	Polynomial	83.9	17.3	85.1	83.1	84.1
	RBF	67.4	36.4	71.3	66.2	68.7
LBP-HOG	Polynomial	86.5	13.9	86.9	86.2	86.5
	RBF	71.7	38.9	82.3	67.9	74.4

TABLE VI
RESULTS FOR SCENARIO S4 - SIDE UNLOCK

Features	Kernel Function	Accuracy	FPR	Recall	Precision	F1-Score
GLDS	Polynomial	61.4	41.5	64.3	60.8	62.5
	RBF	59.8	41.2	60.8	59.6	60.2
LBP	Polynomial	76.6	24.5	77.7	76.0	76.8
	RBF	64.1	45.3	73.4	61.9	67.2
HOG	Polynomial	75.8	25.4	77.1	75.1	76.1
	RBF	61.9	35.6	59.4	62.6	61.0
LBP-HOG	Polynomial	78.1	22.2	78.5	77.9	78.2
	RBF	65.2	42.9	73.4	63.1	67.9

1) Scenario S1 - All Unlock: From Table III, among the features extracted, the FPR of LBP-HOG performs well in general, which is 7.7% with polynomial kernel function and 8.5% with RBF. Also, LBP-HOG with polynomial kernel function achieves the highest accuracy in scenario S1, with 85.5%. As a result, it delivers the optimal performance over these methods. In contrast, GLDS can hardly function properly on puppet attack dataset, only achieving accuracies of 68.5% and 65.5%. In other words, GLDS can hardly meet this case's basic binary classification requirements.

2) Scenario S2 - Straight Unlock: In scenario S2, LBP-HOG with both kernel functions has a similar performance to LBP with both kernel functions. The former one delivers good performance of 97.2% and 86.7% accuracy and the latter one is of 97.1% and 86.9% accuracy. However, the LBP-HOG with polynomial kernel function delivers optimal performance of 1.0% FPR. This improves the LBP with the same kernel function's performance of 1.3% FPR, which is an improvement ratio of 29%.

3) Scenario S3 - 45 Degree Unlock: Similar with scenario S2 and S1, LBP-HOG with polynomial kernel function still holds the lowest FPR, which is 13.9%. However, its accuracy, 86.5%, is lower than LBP with polynomial kernel function, which is 86.6%.

4) Scenario S4 - Side Unlock: As scenario S4 shows, all methods perform not so well comparing with S1, S2 and S3, since the highest accuracy only reaches 78.1% and the lowest FPR represents 22.2%, both derives from LBP-HOG with polynomial kernel function. LBP with polynomial kernel function has similar performance, of which accuracy is 76.6% and FPR is 24.5%.

5) Discussion: As presented in Tables III, IV, V and VI, the polynomial kernel function always performs better than RBF in every scenario. In most cases, with the same kernel function, LBP performs better than HOG, and the fusion of LBP and HOG performs best. GLDS performs the worst.

In four scenarios, straight unlock reaches the highest performance, which means when the database consists of strict standard fingerprints, the offset caused by force action can be easily detected. Similarly, when the database consists of fingerprints that are under the side unlock circumstance, the difference between forced and unforced fingerprints is harder to detect. As a result, methods in scenario S4, side unlock, perform worst compared with three other scenarios.

In all four scenarios, SVM with polynomial kernel function achieves higher accuracy than RBF in terms of all features, with a gain of about 10%.

To identify a model configuration that would not only yield high accuracy but also maintain an acceptable error rate when applied to our dataset, referred to as S1, we evaluated a range of models to discern their capacity to manage the inherent trade-off between True Positive Rates (TPR) and False Positive Rates (FPR). Our preliminary assessments indicated that three models—LBP-HOG with Polynomial Kernel, LBP with Polynomial Kernel, and LBP with RBF Kernel—stood out in terms of their performance metrics.

We opted to focus on these models for a comparative analysis because they demonstrated a promising balance between sensitivity and specificity. As Fig. 16 shows, the LBP-HOG with Polynomial Kernel configuration demonstrated the highest TPR across the lowest spectrum of FPR values, suggesting a robust capability in distinguishing between classes with minimal errors. This model was selected for its superior balance between sensitivity and specificity and is particularly suited for scenarios where precision is critical. The LBP with Polynomial Kernel configuration, while exhibiting a marginally lower TPR, was chosen for its relatively high performance and demonstrates a viable option in environments where a slightly higher error rate can be accommodated. The LBP with RBF Kernel was included in the comparison as it presents a competitive alternative, offering a modest trade-off in TPR for a reduced FPR in specific threshold settings,

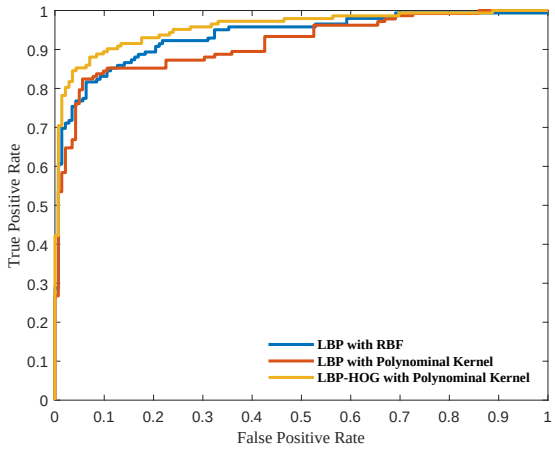


Fig. 16. Comparison of the ROC curve.

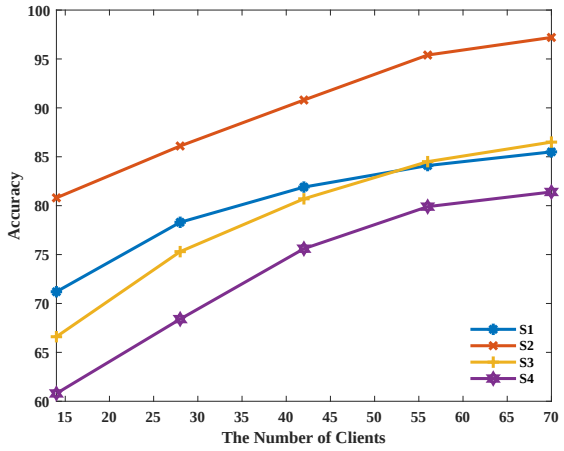


Fig. 17. Performance of the LBP-HOG-Polynomial method under different numbers of clients.

which might be desirable in certain practical applications. The LBP-HOG with Polynomial Kernel model exhibits the highest AUC, indicating its overall superior performance. The other two models present slightly lower AUC values but are still competitive, making them acceptable for use cases with different error tolerances.

In conclusion, our comparative analysis suggests that the LBP-HOG with Polynomial Kernel model is the most capable in handling lower error rates efficiently, thereby being the recommended choice for applications where accuracy is of the utmost importance. The other two models provide alternatives when a balance between different types of errors is required, allowing for flexibility based on the specific needs and error rate tolerances of the application.

D. Analysis of Puppet Attack Dataset

1) Comparison of Dataset Scale: In order to explore if the size of the dataset and client numbers of the dataset would impact the recognition method performance and limit the verification of application scenarios, we utilized the data of the first 20%, 40%, 60%, 80%, and 100% of the full dataset. The number of clients is 14, 28, 42, 56, and 70 respectively. We applied the LBP-HOG feature and training SVM with the polynomial kernel function (LBP-HOG-Polynomial method) and utilized 10-fold cross-validation.

Fig. 17 shows the detection performance of the basic method under different numbers of clients. As the number of clients increases, the forced fingerprint detection is gradually improved. Take S1 as an example, it is noteworthy that the detection improvement is gradually decreasing, which is 7.1% (from 78.3% to 71.2%), 3.6% (from 81.9% to 78.3%), 2.2%(from 84.1% to 81.9%) and 1.4% (from 85.5% to 84.1%). The improvement decreasement shows the challenge and difficulty of the detection. However, positive correlation exists between the amount of training data and the detection performance. In other words, the improvement shows the necessity of making large-scale datasets.

2) Comparison of Category Proportion: Section VI-D1 shows detection accuracy is sensitive to the size of the dataset. Besides, different quantities of subjects and training sets could have an impact on the detection performance. In this section, we carry out the experiment to quantify the relationship between training-validation proportion and detection performance.

TABLE VII
PERFORMANCE UNDER DIFFERENT TRAINING SET SIZES

Training Set Size	Accuracy
1	68.2
2	77.6
3	86.2
4	88.1
5	89.5
6	90.9
7	92.3
8	93.2
9	93.3

Since Fig. 17 illustrates that the LBP-HOG-Polynomial method performs best in the whole dataset under scenario S2, we utilize it it as the basic method, and decrease the training data to show how the detection performance changes. For each experiment, we selected 1 to 9 images as the training set, with the remaining images forming the test set.

As demonstrated in Table. VII, with the enlargement of the training set size, the performance of detection system improves. Specifically, when the number of images from every 10 images is between 1 and 3, the system performance effectively improves. This improvement is particularly pronounced when the training set size expands from including only 1 image per 10 to 3 images per 10, suggesting a threshold effect where having too few examples per class significantly hampers the model's learning capability.

This threshold effect is indicative of the model's need for a minimum amount of data to adequately learn the distinguishing features of forced and unforced fingerprint presses. Below this threshold, the model may not have sufficient examples to learn from, resulting in a lower performance. As more images are added to the training set, the model has more examples to learn the variability inherent in the data, leading to an improved ability to generalize to new, unseen data.

3) Comparison of Negative Sample Rates: To investigate the influence of negative sample rates on the classification

performance of our model, we designed an experiment varying the proportion of negative samples to positive samples in our training dataset. We employed a stratified split to ensure a consistent distribution of samples, allocating 20% of the data for testing purposes, thus preserving the original distribution of classes. For the training set, we systematically modified the negative sample rate from 0% to 100%, in increments of 10%. At each interval, we trained the LBP-HOG with Polynomial Kernel model. The model's accuracy, precision, and recall were computed against the unchanged test set to assess performance. As Fig. 18 shows, the model exhibitsthe highest accuracy at a negative sample rate of approximately 40%, where it demonstrates a well-rounded ability to identify both positive and negative instances correctly. Precision, while relatively stable, does show some variance, with a slight drop as the negative sample rate increases, implying the model's conservative stance in predicting negative instances under imbalance. An important observation is that beyond the 60% negative sample rate threshold, the model's accuracy starts to align with random chance, suggesting that the classifier's ability to distinguish between classes diminishes. This indicates a practical upper limit for the negative sample rate in training datasets for maintaining effective classification in a binary context. The balance achieved near the 40% negative sample rate underscores the model's capability to accurately classify instances without a significant bias toward either class.

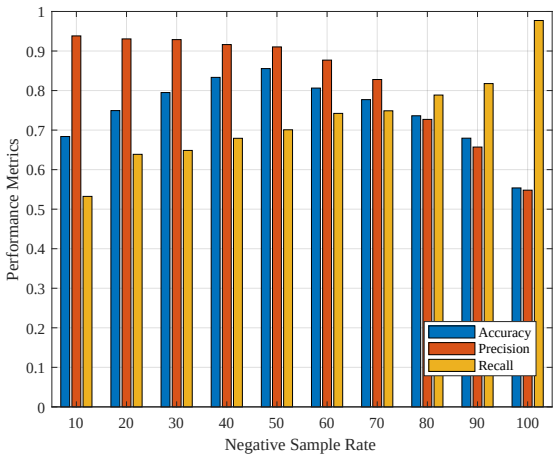


Fig. 18. Model performance at different sample rates.

VII. Future Work

1) Dataset Expansion: While our dataset covers a broad age range, further analysis could benefit from considering demographic factors such as age, gender, and geographic distribution more closely. Future efforts could be directed towards achieving a demographically balanced dataset to facilitate the development of more personalized detection algorithms. We encourage future researchers to replicate our experimental setup and data collection process. We have provided a detailed description of the hardware and environmental settings used in our study, enabling others to gather similar datasets independently.
2) Theoretical Aspects: Despite our rigorous efforts to ensure robustness in detecting forced fingerprint pressing attacks, the challenge of relying on single images and a limited feature set remains. Future theoretical work could explore the expansion of the feature set for enhancing detection capabilities. This may involve investigating additional biometric markers that indicate forced pressure and the application of more sophisticated anomaly detection algorithms. We encourage researchers to utilize our method to replicate our findings and extend them. By applying our detection methods to their datasets, researchers can facilitate subsequent comparisons and validations [44].
3) Network Architecture Exploration: With Support Vector Machines providing a solid baseline, future exploration might consider the integration of advanced neural network architectures. The aim would be to assess whether deep learning models, with their advanced feature representation abilities, can outperform traditional methods in this context.
4) Achieving high accuracy with limited data is a common challenge in biometric security. Future research can investigate methods to improve model generalization from small sample sizes, possibly through techniques such as few-shot learning, synthetic data generation, and transfer learning. This line of work is crucial for practical applications where data collection is challenging or privacy concerns limit the availability of large datasets.

VIII. Conclusion

In this paper, we emphasize the forced fingerprint attack and propose the attack dataset puppet attack. The proposed puppet attack dataset consists of 5,600 images from 70 volunteers under forced and unforced states. We believe that this dataset will increase the attention on this kind of attack and promote the development of finger-based attack detection. In addition, a computer-aided identification for HOG, LBP, and GLDS feature discrimination of forced and unforced fingerprints based on BM2166 capacitance fingerprint platform images is proposed. According to the experimental results, the LBP-HOG fusion feature on SVM with polynomial kernel function has a good ability to distinguish forced and unforced fingerprints and is expected to provide an auxiliary authentication basis for fingerprint verification.

References

[1] "International organization for standardization," ISO, 10 2023. [Online]. Available: https://www.iso.org
[2] F. Liu, G. Liu, Q. Zhao, and L. Shen, "Robust and high-security fingerprint recognition system using optical coherence tomography," *Neurocomputing*, vol. 402, pp. 14–28, Aug. 2020.
[3] "Touch id system," Apple Support, 2023. [Online]. Available: https://support.apple.com/en-us/HT204587
[4] "Topic: Mobile payments worldwide," Statista, 2023. [Online]. Available: https://www.statista.com/topics/4872/mobile-payments-worldwide
[5] B. Biggio, "Security evaluation of biometric authentication systems under real spoofing attacks," *IET Biometrics*, vol. 1, no. 1, pp. 11–24, Mar. 2012.
[6] 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.

[7] C. Wu, K. He, J. Chen, Z. Zhao, and R. Du, "Toward robust detection of puppet attacks via characterizing fingertip-touch behaviors," *IEEE Trans. Depend. Sec. Comput.*, Sep. 2021.

[8] 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," *Proc ACM Interact Mob Wearable Ubiquitous Technol.*, vol. 5, no. 1, pp. 1–30, Mar. 2021.

[9] A. Ng, "Child uses sleeping mom's fingerprints to buy pokemon gifts," CNET, 2023. [Online]. Available: https://www.cnet.com/culture/child-uses-sleeping-moms-fingerprints-to-buy-pokemon-gifts/

[10] "Face id, touch id, no id, pins and pragmatic security," Troy Hunt, 2023. [Online]. Available: https://www.troyhunt.com/face-id-touch-id-pins-no-id-and-pragmatic-security

[11] K. Waddell, "Police can force you to use your fingerprint to unlock your phone," The Atlantic, 05 2023. [Online]. Available: https://www.theatlantic.com/technology/archive/2016/05

[12] "This child used her sleeping mother's fingerprint to buy 250 worth of pokemon gear," Complex, 2023. [Online]. Available: https://www.complex.com/life/

[13] "Turn attention aware features on or off on your iphone or ipad pro," Apple Support, 2023. [Online]. Available: https://support.apple.com/en-us/HT208245

[14] V. Mura, G. Orrù, R. Casula, A. Sibiriu, G. Loi, P. Tuveri, L. Ghiani, and G. L. Marcialis, "Livdet 2017 fingerprint liveness detection competition 2017," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 297–302.

[15] G. Orrù, R. Casula, P. Tuveri, C. Bazzoni, G. Dessalvi, M. Micheletto, L. Ghiani, and G. L. Marcialis, "Livdet in action - fingerprint liveness detection competition 2019," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2019, pp. 1–6.

[16] "Livdet - liveness detection competitions," livdet.org, 2023. [Online]. Available: http://livdet.org/

[17] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2013, pp. 57–59.

[18] S. Vhaduri and C. Poellabauer, "Multi-modal biometric-based implicit authentication of wearable device users," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 12, pp. 3116–3125, Apr. 2019.

[19] Z. Zhou and A. Kumar, "Finger-knuckle assisted slap fingerprint identification for higher security and convenience," *IEEE Trans. Inf. Forensics Secur.*, Sep. 2023.

[20] P. Coli, G. L. Marcialis, and F. Roli, "Vitality detection from fingerprint images: a critical survey," in *International Conference on Biometrics*. Springer, 2007, pp. 722–731.

[21] F. Liu, Z. Kong, H. Liu, W. Zhang, and L. Shen, "Fingerprint presentation attack detection by channel-wise feature denoising," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2963–2976, Aug. 2022.

[22] F. Liu, H. Liu, W. Zhang, G. Liu, and L. Shen, "One-class fingerprint presentation attack detection using auto-encoder network," *IEEE Trans. Image Process.*, vol. 30, pp. 2394–2407, Jan. 2021.

[23] T. Chugh and A. K. Jain, "Fingerprint spoof detector generalization," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 42–55, Apr. 2020.

[24] D. Baldisserra, A. Franco, D. Maio, and D. Maltoni, "Fake fingerprint detection by odor analysis," in *Advances in Biometrics: International Conference, ICB 2006, Hong Kong, China, January 5-7, 2006. Proceedings*. Springer, 2005, pp. 265–272.

[25] M. Sandstrom, "Liveness detection in fingerprint recognition systems," 2004.

[26] P. V. Reddy, A. Kumar, S. Rahman, and T. S. Mundra, "A new antispoofing approach for biometric devices," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 4, pp. 328–337, Nov. 2008.

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

[28] S. A. Schuckers, "Spoofing and anti-spoofing measures," *Information Security technical report*, vol. 7, no. 4, pp. 56–62, Dec. 2002.

[29] S. Schuckers and P. Johnson, "Fingerprint pore analysis for liveness detection," 2017, uS Patent 9,818,020.

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

[31] Y. Zhang, X. Li, H. Wang, R. Wang, P. Chen, and R. Liang, "Sweat gland extraction from optical coherence tomography using convolutional neural network," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–16, Nov. 2022.

[32] J. J. Engelsma and A. K. Jain, "Generalizing fingerprint spoof detector: Learning a one-class classifier," in *Proc. Int. Conf. Biometrics (ICB)*. IEEE, Jan. 2019, pp. 1–8.

[33] S. B. Nikam and S. Agarwal, "Texture and wavelet-based spoof fingerprint detection for fingerprint biometric systems," in *2008 First International Conference on Emerging Trends in Engineering and Technology*, 2008, pp. 675–680.

[34] L. Ghiani, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection by local phase quantization," in *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*. IEEE, 2012, pp. 537–540.

[35] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "Fingerprint liveness detection based on weber local image descriptor," in *2013 IEEE workshop on biometric measurements and systems for security and medical applications*. IEEE, 2013, pp. 46–50.

[36] ——, "Local contrast phase descriptor for fingerprint liveness detection," vol. 48, no. 4, Apr. 2015.

[37] P. Wasnik, R. Ramachandra, K. Raja, and C. Busch, "Presentation attack detection for smartphone based fingerphoto recognition using second order local structures," in *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2018, pp. 241–246.

[38] M. Frigge, D. C. Hoaglin, and B. Iglewicz, "Some implementations of the boxplot," *The American Statistician*, vol. 43, no. 1, pp. 50–54, 1989.

[39] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[40] A. Baraldi and F. Panniggiani, "An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters," *IEEE Trans. Geosci. Remote Sens.*, vol. 33, no. 2, pp. 293–304, Mar. 1995.

[41] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[42] V. N. Vapnik, *The Nature of Statistical Learning Theory*. The nature of statistical learning theory, 1995.

[43] "Stm32f407ze datasheet," STMICROELECTRONICS, 2023. [Online]. Available: https://www.alldatasheet.com/datasheet-pdf/pdf/505003/STMICROELECTRONICS/STM32F407ZE.html

[44] Y. Ma, H. Luo, and W. Wang, "The-self-detection-method-of-the-puppet-attack-in-biometric-fingerprinting," Github, 2023. [Online]. Available: https://github.com/Lynnon/The-Self-Detection-Method-of-the-Puppet-Attack-in-Biometric-Fingerprinting