mmSign: mmWave-based Few-Shot Online Handwritten Signature Verification

MINGDA HAN, Shandong Normal University, China and City University of Hong Kong Shenzhen Research Institute, China
HUANQI YANG, TAO NI, DI DUAN, MENGZHE RUAN, and YONGLIANG CHEN, City University of Hong Kong, China and City University of Hong Kong Shenzhen Research Institute, China
JIA ZHANG, Shandong Normal University, China
WEITAO XU, City University of Hong Kong, China and City University of Hong Kong Shenzhen Research Institute, China

Handwritten signature verification has become one of the most important document authentication methods that are widely used in the financial, legal, and administrative sectors. Compared with offline methods based on static signature images, online handwritten signature verification methods are more reliable because of the temporary dynamic information (e.g., signing velocity, writing force, stroke order) that alleviates the risk of being forged. However, most existing online handwritten signature verification solutions are reliant on specific signing devices (e.g., customized pens or writing pads) and require extensive data collection during the registration phase, resulting in poor adaptability and applicability for new users. In this article, we propose mmSign, a millimeter wave (mmWave)–based online handwritten signature verification system, which enables accurate sensing of the user's hand movements when signing through the superior sensing capability of mmWave. mmSign extracts the time-velocity feature maps from the captured mmWave signals by the carefully designed signal processing algorithms and then exploits a transformer-based verification model for signature verification. In addition, a novel meta-learning strategy with proposed task generation and data augmentation methods is introduced in mmSign to teach the verification model to learn effectively with

The work described in this article was substantially sponsored by the project 62101471 supported by NSFC and was partially supported by the Shenzhen Research Institute, City University of Hong Kong. The work was also partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 21201420 and Project No. CityU 11201422), Shenzhen Science and Technology Funding Fundamental Research Program (Project No. 2021Szvup126), and NSF of Shandong Province (Project No. ZR2021LZH010), Chow Sang Sang Group Research Fund sponsored by Chow Sang Sang Holdings International Limited (Project No. 9229062). The work was also partially supported by CityU MFPRC grant no. 9680333, CityU SIRG grant no. 7020057, CityU APRC grant no. 9610485, CityU ARG grant no. 9667225 and CityU SRG-Fd grant no. 7005666.

Authors' addresses: M. Han, Shandong Normal University, No. 1 Daxue Road, Jinan, Shandong, 250358, China and City University of Hong Kong Shenzhen Research Institute, No. 8 Yuexing 1st Road, Shenzhen, Guangdong, 518057, China; email: mingdhan@cityu.edu.hk; H. Yang, T. Ni, D. Duan, M. Ruan, and Y. Chen, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong SAR, China and City University of Hong Kong Shenzhen Research Institute, No. 8 Yuexing 1st Road, Shenzhen, Guangdong, 518057, China; email: {huanqi.yang, taoni2-c, dduan5-c, cs.mzr, cs.ylchen}@my.cityu.edu.hk; J. Zhang (corresponding authors), Shandong Normal University, No. 1 Daxue Road, Jinan, Shandong, 250358, China; email: zhangjia@sdnu.edu.cn; W. Xu (corresponding authors), City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong SAR, China and City University of Hong Kong Shenzhen Research Institute, No. 8 Yuexing 1st Road, Shenzhen, Guangdong, 518057, China; email: weitaoxu@cityu.edu.hk.

1550-4859/2024/05-ART89 $15.00
https://doi.org/10.1145/3605945

limited samples, allowing our model to quickly adapt to new users. Extensive experiments show that mmSign is a robust, efficient, and secure handwritten signature verification system, achieving 84.07%, 87.31%, 91.12%, and 96.54% verification accuracy when 1, 3, 5, and 10 labeled signatures are available, respectively, while being resistant to common forgery attacks.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**; • **Security and privacy → Biometrics;**

Additional Key Words and Phrases: Signature verification, mmWave sensing, Meta-learning

ACM Reference format:
Mingda Han, Huanqi Yang, Tao Ni, Di Duan, Mengzhe Ruan, Yongliang Chen, Jia Zhang, and Weitao Xu. 2024. mmSign: mmWave-based Few-Shot Online Handwritten Signature Verification. *ACM Trans. Sensor Netw.* 20, 4, Article 89 (May 2024), 31 pages.
https://doi.org/10.1145/3605945

1 INTRODUCTION

Biometric-based authentication has gained popularity as a more convenient and reliable way to achieve secure authentication, which utilizes people's unique biological characteristics (e.g., fingerprints [72], irises [45], and behavioral habits [64]) for authentication. Among them, handwritten signature verification has been widely used as one of the main verification methods for paper documents in the fields of finance, law, administration, and more. Despite its widespread application, signature verification is vulnerable to forgery attacks [8, 32], which results in enormous damage. In the financial sector, for example, signature forgery of paper checks accounts for 66% of financial fraud activity in 2022 [32]. Moreover, signature forgery also happens in other fields [8, 47, 54], indicating the significance of handwritten signature verification research.

Depending on the signature acquisition approach, existing signature verification methods can be divided into two categories: offline signature verification methods [15, 25, 49] and online signature verification methods [9, 38, 40]. The offline signature verification methods use the user's static signature features (i.e., 2D image features) for verification [24]. Since the offline methods consider only the final static signature features and ignore the dynamic features during the signature execution process, they are vulnerable to being forged, whereas the online signature verification methods utilize the user's dynamic features (e.g., signing velocity, writing force, stroke order) during the signature execution process for verification [31]. Compared with offline methods, online methods can obtain additional dynamic signing information, which makes them more reliable.

Owing to its reliability, a variety of online signature verification schemes have been proposed. According to different signature acquisition methods, the existing online signature verification schemes can be divided into four categories: digital signature device-based [50], wearable device–based [40], camera-based [79], and wireless sensing–based [82] signature verification methods. While these schemes employ different signature acquisition methods, they essentially leverage the same idea that different people show distinct dynamic characteristics due to different signing habits even when signing the same name. We summarize some existing online handwritten signature verification schemes in Table 1 and find the following limitations.

(1) Low generalizability. Users may sign on different surfaces (e.g., paper documents, tablets) or with different pens (e.g., signature pens, digital pens). The digital signature device–based online handwritten signature verification systems [38, 40, 50] can only sign on/with specified digital devices (e.g., digital tablets or pens) and cannot verify signatures on paper documents. In addition, existing acoustic-based schemes [9, 17, 82] are sensitive to the relative position of the user's signature box to the acoustic sensor, which results in rapid system performance degradation when the position of the user's signature changes. **(2) Low data efficiency.** Most of the existing online

Table 1. Comparison of Online Handwritten Signature Verification Methods

Scheme	Signature Acquisition Method	Generalizability	Privacy Protection	Data Efficiency	User Experience
[38, 43, 50]	Digital Tablets or Pens	✗	✓	✗	✓
[37, 40, 51]	Wearable Devices	✓	✓	✗	✗
[79]	Camera	✓	✗	✗	✓
[9, 17, 82]	Acoustic	✗	✓	✓	✓
mmSign	mmWave	✓	✓	✓	✓

handwritten signature verification systems often require a large number of genuine signatures from new users to achieve good verification performance [16, 31, 50]. Although a large amount of data can improve system performance, extensive data collection is time-consuming and labor-intensive, which is not in line with the practical application scenario. **(3) Less privacy protection.** Camera-based solutions [79] always shoot the user's hand at a very close distance during the signing process, which may cause leakage of the user's private information such as fingerprints [48, 71]. As such, a promising handwritten authentication system needs to focus only on the signature without capturing extra sensitive information from the document. **(4) Low user experience.** Wearable device–based online signature verification systems require the user to wear a specific device (e.g., smartwatch [40, 51], data glove [37]) while signing the document, which is not convenient in practice and degrades the user experience.

The aforementioned limitations motivate us to design an online signature verification system that would meet the requirements of high generalizability, adequate privacy protection, high data efficiency, and good user experience. In this article, we propose mmSign, which leverages the superb sensing capabilities of millimeter wave (mmWave) to achieve a non-intrusive online handwritten signature verification with only a few samples. Table 1 illustrates the properties of mmSign while comparing it with other existing schemes. The basic idea is to use mmWave to sense hand movements during the user's signature execution process. Due to the different hand sizes and signature habits, such as the signing velocity and stroke order of each individual, the features obtained by mmWave radar are different even when signing the same name. Although the idea is straightforward, we need to address several non-trivial challenges.

- **Challenge 1:** The raw frequency modulated continuous wave (FMCW) signal obtained from the mmWave radar contains a lot of noises from surrounding objects and the user's body. Therefore, how to eliminate static and dynamic noises and obtain time-velocity feature maps reflecting the user's signature information is the first challenge.
- **Challenge 2:** After getting the time-velocity feature maps, we need to use them to verify the genuineness of the signature. However, even the same user has slight differences during different signature execution processes, which results in differences in the generated feature maps. Therefore, how to design a verification model to extract high-level features that are robust to changes in the feature maps but still user-specific is another challenge.
- **Challenge 3:** Most existing signature verification systems require large amounts of training data to achieve good performance. The massive signature collection for each newly registered user is impractical, and it reduces the user experience. Therefore, how to achieve a good verification performance with limited data when new users register is the third challenge.

We propose a series of approaches to tackle the above challenges in mmSign. Firstly, we design several novel signal processing methods to eliminate various noises and accurately locate hand movements for extracting time-velocity features during the signature execution process. Then, a transformer-based verification model is designed to encode the input feature maps into a high-level vector, and verify the genuineness of the signature. In order to improve the generalization ability and robustness of the designed verification model, we design three data augmentation schemes based on the variation characteristics of the mmWave signal during the signature execution process. Finally, with the help of a designed task generation strategy, a meta-learning framework is introduced in mmSign to quickly adapt to newly registered users using only a few samples.

Our contributions in this article are summarized as follows:

- We propose mmSign, the first mmWave-based non-invasive online handwritten signature verification system, which is applicable to any writing surface without any privacy leakage and is data efficient and user-friendly.
- We propose a series of signal processing methods to obtain the informative hand-signing features from the raw FMCW signals. Specifically, a sub-signal generation algorithm and a feature extraction method are designed to accurately localize the hand movements and obtain the time-velocity feature maps, respectively.
- We design a novel transformer-based verification model to verify the authenticity of the signature. Together with the proposed three data augmentation methods based on the variation characteristics of mmWave signals during the signature execution process, mmSign achieves favorable verification performance.
- We formulate the handwritten signature verification task as a meta-learning problem and design a meta-learning framework to ensure that new users can quickly adapt to our system with only a few samples. In addition, a task generation strategy is proposed to enhance the performance of meta-learning.
- We conduct a comprehensive evaluation of mmSign in multiple real-world environments using various signature pens and writing surfaces. Evaluation results demonstrate mmSign's good adaptability to new users. Security analysis is also conducted to show that mmSign is resistant to common forgery attacks.

The rest of this article is organized as follows. We briefly review the related works in Section 2. We present the design details of mmSign in Section 3. We evaluate the performance of mmSign through extensive experiments in Section 4. Then, we present the results of the user study in Section 5. Finally, in Sections 6 and 7, we discuss the remaining problems and conclude this article, respectively.

2 RELATED WORK

In this section, we briefly review the related works on handwritten signature verification, mmWave sensing, and few-shot learning in wireless sensing.

2.1 Signature Verification

Offline signature verification systems. The offline signature verification system registers the user's static signature into the system through an offline signature acquisition device (e.g., scanner, camera). When the user logs in again, the static signature used to log in is compared with the registered signature to determine whether the user is legitimate. Since the offline signature verification system represents signatures as images, the key to achieving accurate signature verification is to extract the desirable features from the signature image.

Many research efforts have been devoted to finding good handcrafted feature representations for offline signatures. Oliveira et al. [49] used graphometric features, such as the ratio of height/width, the symmetry, and the empty spaces between strokes, to examine handwriting for signature verification. Drouhard et al. [19] leveraged the directional probability density function obtained from the gradient of the signature outline to represent the directional features, which respond to the direction of the signature's strokes. With the development of deep learning, many researchers have attempted to use deep learning models to extract features directly from the raw signature image. SigNet [15] modeled the signature verification task with a convolutional Siamese network to realize offline writer-independent signature verification. Soleimani et al. [59] proposed to use Deep Multitask Metric Learning (DMML) for offline signature verification by applying skilled forgery knowledge in the feature learning process. Hafemann et al. [25] formulated the offline signature verification problem as a meta-learning problem and used extended Model Agnostic Meta Learning (MAML) to improve the classifier adaptation to new users. However, offline handwritten signatures based on static images are vulnerable to being forged.

Online signature verification systems. Online signature verification is also known as dynamic signature verification. Unlike offline systems, online signature verification systems utilize the dynamic information (e.g., signing velocity, signing pressure, and stroke order) of the signer while signing as the basis for verification. Therefore, online signature verification systems present higher reliability than conventional offline approaches.

The most classic online signature acquisition devices are digital tablets [38] and electronic pens [13, 46, 50, 55], both of which can obtain temporal dynamic information (e.g., signing pressure, pen inclination, velocity, and acceleration) about the user's signature process through the built-in sensors, such as gyroscopes, inertial measurement units, and strain gauges. However, both of these methods require signing on a designated digital signature medium (e.g., tablet), which makes them inapplicable to situations in which users sign on paper documents in their daily lives. To solve this problem, wearable device–based online signature verification methods are proposed. For example, Levy et al. [40] leveraged the smartwatch to capture movement data (i.e., accelerometer and gyroscope measurements) from the built-in sensors during the signature execution process and trained a classifier to determine whether a query signature was genuine or forged. PPGSign [51] proposed to leverage the photoplethysmography (PPG) sensors in the wrist-worn wearable device to obtain the unique blood flow changes in the user's hand movement during the signing process to verify the authenticity of the signature. Kamel et al. [37] proposed to use the data glove to obtain information about the multiple degrees of freedom obtained for each finger and hand. They used the singular value decomposition numerical tool for signature classification and verification. Although these solutions do not require a signature medium, they do require the signer to wear specific hardware devices, which is unrealistic in practice and reduces the user experience. Yasuda et al. [79] proposed to use low-cost webcams for non-intrusive online handwritten signature verification. However, this solution raises privacy issues.

The latest works take advantage of the sensing capability of acoustic signals. ASSV [17] is the first system that uses acoustic signals transmitted and received by smartphones to realize signature verification. SilentSign [9] is another acoustic-based online signature verification system that leverages acoustic signals to measure the change in distance of the pen tip when signing and develops a phase-based distance measurement method for signature verification. However, existing acoustic-based handwritten signature verification schemes model the whole hand/pen as a single reflection point and intentionally neglect weak multi-path signals. This approach means that the final signal obtained is the result of the two moving parts (the user's hand and the pen's upper part, which will be explained in Section 3.2.3) canceling each other out, which is insufficient to accurately capture the user's hand/pen movement features during the signing process, particularly when there are significant changes in the signature position. Additionally, existing acoustic-based methods leverage the channel impulse response (CIR) phase to estimate the hand/pen moving patterns, which is heavily dependent on the target distance [70]. For instance, experiments conducted at ASSV [17] reveal that when the line-of-sight (LOS) distance between the signature position and the acoustic sensor is 6 cm, the average signature verification accuracy is 89.95%, but it rapidly drops to 29.95% when this distance is reduced to 3 cm. Similarly, although SilentSign [9] demonstrates better performance with its designed algorithms, the accuracy decreases by 20% when the relative vertical position (perpendicular to LOS) changes by 10 cm, which is not acceptable in signature verification. In this article, we use commercial mmWave radar for online handwritten signature verification. mmSign has higher accuracy and stronger anti-interference capability with the large bandwidth of mmWave compared with the above acoustic-based online signature verification methods.

2.2 mmWave Sensing

With the rapid development of wireless sensing technologies, recent works propose to leverage wireless signals, such as acoustic signals [10, 66], Wi-Fi signals [26, 67], and mmWave signals [35, 42] for various fine-grained sensing tasks. Among these wireless signals, mmWave, with its short wavelength and high frequency, can sense the tiny movements of the target more accurately.

Recently, researchers have used mmWave in various sensing tasks, such as human activity recognition [36, 57], vital sign monitoring [11, 78], audio reconstruction [30, 63], and user identification [23, 77]. In addition, many researchers use mmWave for authentication. For example, VocalPrint [41] uses mmWave signals to capture the unique characteristics of a user's vocal cord vibrations when they are speaking to achieve a secure and attack-resistant authentication. Heart-Print [68] is a commercial mmWave radar-based multi-user authentication method, which first locates and separates different users through a designed clustering algorithm, and then uses the proposed signal energy comparison method and feature extraction method for heartbeat feature extraction to achieve continuous multi-user authentication. Likewise, M-Auth [69] adopts a similar idea to leverage the user's unique breathing pattern for multi-user authentication. Moreover, mmFace [75] implements a reliable liveness detection and face authentication system that works even under the occlusion of face masks by extracting facial biometric and structural features when the mmWave signals bounce off the human face. To the best of our knowledge, mmSign is the first work that uses commodity mmWave radar to achieve online handwritten signature verification.

2.3 Few-shot Learning in Wireless Sensing

Despite the success of deep learning in various tasks [27, 56, 81], they require large amounts of data and multiple iterations for training multiple models in different scenarios. To address this problem, few-shot learning algorithms [20, 58, 61] are proposed to achieve fast domain adaptation with only a few labeled samples from different conditions.

With the prosperity that few-shot learning has brought to the computer vision area [65, 80], more and more researchers have harnessed few-shot learning methods into the implementation of wireless sensing systems. For instance, MetaSense [22] designs a task generation strategy to effectively leverage the available data and enhance the performance of meta-training. GazeGraph [39] is a cognitive context sensing system that uses the human gaze as a sensing modality, which uses the few-shot learning strategy to quickly adapt to unseen perceptual scenarios using a small number of instances. OneFi [73] is a Wi-Fi-based human gesture recognition system that enables the recognition of unseen gestures with only one (or few) labeled samples assisted by the few-shot recognition mechanism. In addition, CAUTION [64] is able to build an accurate user model for a Wi-Fi channel state information (CSI)-based human authentication system with a very limited number of CSI training samples. Inspired by these works, we design a novel meta-learning strategy to adapt our handwritten verification model to new users with a few samples.

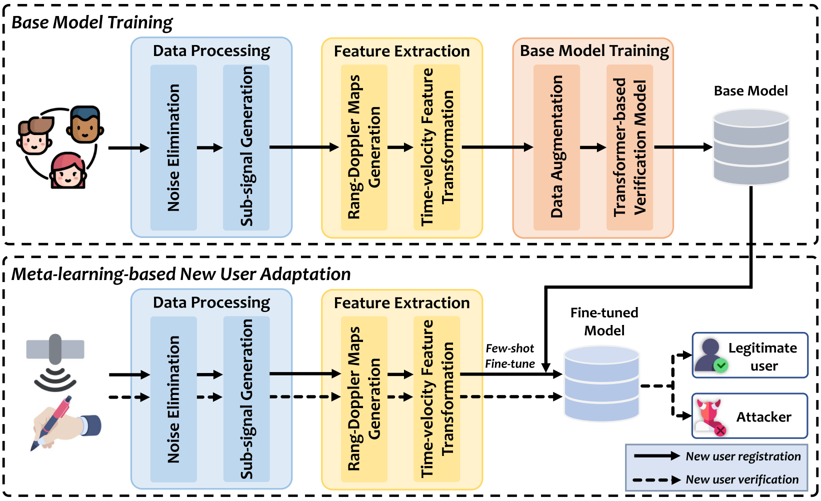


Fig. 1. System overview.

3 SYSTEM DESIGN

3.1 Overview

Figure 1 presents the overview of mmSign, an mmWave-based online handwritten signature verification system built on commodity mmWave radar. The core idea of mmSign is to make the base model learn to recognize the authenticity of a new user's handwritten signature quickly, thereby enabling fast adaptation to new users with only a few labeled samples. Specifically, there are two phases in mmSign: the base model training phase and meta-learning-based new user adaptation phase.

Base model training phase: The mmWave radar first obtains the raw FMCW data when the users sign their names. Then, the static noises are filtered from the raw FMCW data, and the sub-intermediate frequency (sub-IF) signals are obtained by the designed sub-signal generation algorithm. Next, the generated sub-IF signals are used to obtain the Range-Doppler Maps (RDMs) that respond to the hand movement through the designed RDM generation algorithm, and the generated RDMs are transformed into the time-velocity feature map. Finally, in the base model training module, the obtained feature maps are first augmented by the proposed data augmentation algorithms and then fed into the transformer-based verification model for based model training.

Meta-learning-based new user adaptation phase: When new users register their handwritten signature in mmSign, a meta-learning framework is introduced to avoid intensive data re-collection and reduce the time overhead of the model training process. In addition, a designed task generation strategy is used to provide multiple tasks in the meta-training process to improve the efficiency of meta-training. These generated tasks are leveraged to teach the base model to learn a new task (i.e., verify the genuineness of the new user's handwritten signature) quickly and update the base model with only a few signature samples.

3.2 Data Processing

3.2.1 Data Collection and Static Noise Elimination. The mmWave radar transmits the FMCW signal, that is, a chirp. The frequency of the chirp signal increases linearly with time t and can be expressed as

$$f = f_0 + St, \tag{1}$$

ALGORITHM 1: Static Noise Elimination Algorithm

```
Input  : $S$: raw IF signal matrix
         $N_F$ : frame number
         $N_I$ : IF signal number of each frame
Output : $S'$: denoised IF signal matrix
1 Initialize an empty denoised IF signal matrix $S'$
2 for $i = 0; i < N_F - 1; i = i + 1$ do
3     $\boldsymbol{n} \leftarrow \emptyset$                                  ▷ Initialize the static noise vector
4     $\boldsymbol{n} \leftarrow \frac{1}{N_I} \sum_{m=0}^{N_I-1} S(i, m, :)$        ▷ Calculate the static noise vector
5     for $j = 0; j < N_I - 1; j = j + 1$ do
6         $\boldsymbol{t} \leftarrow S(i, j, :)$                     ▷ Get the $j$-th raw IF signal in the $i$-th frame
7         $S'(i, j, :) \leftarrow \boldsymbol{t} - \boldsymbol{n}$               ▷ Calculate the denoised IF signal vector
8     end
9 end
```

where f_0 is the starting frequency and S is the frequency modulation slope. Suppose that the amplitude of the transmitted signal at time t is A; then, the transmitted sinusoidal FMCW signal $s_T(t)$ can be expressed as

$$s_T(t) = A \cos \left[2\pi \left(f_0 t + \frac{St^2}{2} \right) \right]. \tag{2}$$

When the transmitted signal encounters an obstacle (e.g., the user's hand) at a distance d, the radar will receive a delayed version of the transmitted signal $s_R(t)$, which can be expressed as

$$s_R(t) = \alpha A \cos \left[2\pi \left(f_0 (t - \tau) + \frac{S(t - \tau)^2}{2} \right) \right], \tag{3}$$

where α is the path loss, $\tau = 2d/c$ is the time delay, and c is the speed of light. Finally, the transmitted signal $s_T(t)$ is mixed with the received signal $s_R(t)$, and a low-pass filter is used to filter out the sum frequency components to obtain the IF signal:

$$s_{IF}(t) = LPF\{s_T(t) \cdot s_R(t)\} = A_{IF} \cos \left(2\pi f_{IF} t + \phi_{IF} \right), \tag{4}$$

where A_{IF} is the amplitude of the IF signal, $f_{IF} = S\tau = 2dS/c$ is known as the beat frequency, and ϕ_{IF} is the phase. Therefore, the IF signal after sampling can be expressed as

$$s_{IF}(n) = s_{IF}(t) \cdot \mu(n), \quad n = 0, 1, \ldots, N_S - 1, \tag{5}$$

where $\mu(n)$ is the unit step sequence and N_S is the number of samples per IF signal. Combining the signals from N_F radar frames, we can obtain the raw 3D matrix S of size $N_F \times N_I \times N_S$, where N_I is the number of IF signals in each frame.

In addition to the user's hand movements, mmWave radar senses static information about the user's body as well as the surrounding environment (e.g., walls, tables, and chairs), which leads to the generated feature maps containing a lot of static noises. To eliminate these static noises, we design a static noise elimination algorithm (see Algorithm 1). Specifically, the mean value of all chirps in each radar frame is leveraged to represent the static noise vector (Lines 3–4). Thus, the denoised signal can be obtained by subtracting the static noise vector from the raw signal (Lines 6–7).

3.2.2 Sub-signal Generation. After removing the static noises, we need to accurately locate the position of the user's hand to extract useful signals that can reflect the user's handwritten signature execution process. However, low-cost commercial mmWave radars cannot guarantee accurate range estimation under a low signal-to-noise ratio (SNR) based on a single IF signal.

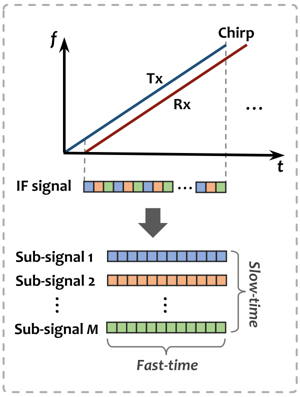


Fig. 2. Sub-signal generation.

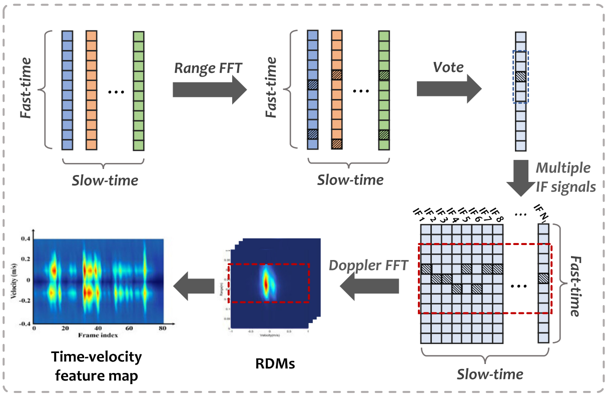


Fig. 3. Time-velocity feature map generation.

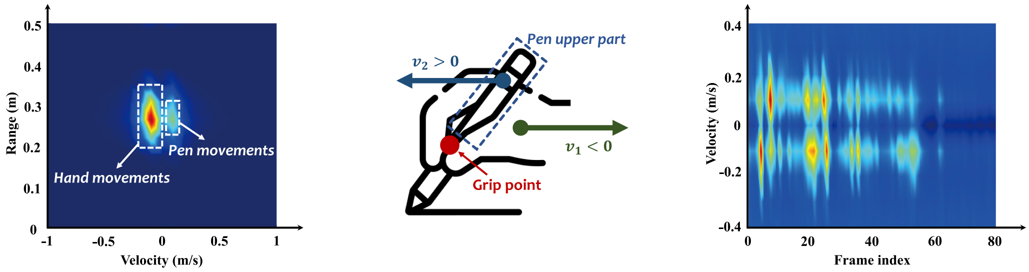
mmVib [35] proposes a sub-signal generation method to achieve robust range estimation, which uses a sliding window to separate the IF signal into different sub-signals according to different starting frequencies. Although this method provides multiple observations (sub-signals) at the same time, the bandwidth of each sub-signal becomes $1/M$ of the original IF signal (M is the number of generated sub-signals). Because the range resolution of mmWave radar is proportional to the bandwidth of the mmWave signal, this method results in range resolution reduction for the individual sub-signal. Therefore, as shown in Figure 2, we design a sub-signal generation algorithm that can use the full bandwidth information of the original IF signal and does not lead to a decrease in range resolution. Specifically, for each IF signal, we generate multiple sub-signals by

$$s_i(n) = s_{IF}(n) \sum_{j=0}^{W-1} \delta[n-(jM+i-1)], \quad i = 1, 2, \ldots, M, \tag{6}$$

where $s_i(n)$ represents the i-th sub-signal, M is the number of sub-chirps, and $W = \lfloor N_S/M \rfloor$ is the length of the sub-signal. These sub-IF signals can be considered to be transmitted at the same time. Therefore, these sub-signals will be used for cross-referencing with each other. Compared with the method in mmVib, our method obtains each sub-signal using the full bandwidth information and therefore does not sacrifice range resolution.

3.2.3 Range-Doppler Map Generation. After obtaining multiple sub-signals, we need to extract accurate hand movement information from them. We first apply the Fast Fourier Transform (i.e., range FFT) on each sub-signal to get the range information. As shown in Figure 3, since there are other moving objects in front of the radar besides the user's hand (e.g., the user's torso and other pedestrians or moving objects in the environment), the range FFT will generate multiple peaks at different IF frequencies. We use the range bin where the first peak is located as the position of the user's hand since the user's hand is the closest moving object to the radar. For the i-th chirp, assuming that the range bin corresponding to the first peak of the k-th sub-signal after Range-FFT is p_{ik}, then we can obtain the accuracy range bin p_i by majority voting. Since the user's hand will occupy multiple range bins, we locate the hand position through a window. Assuming that the window size is L, the range bin of the user's hand in the i-th IF signal is $[p_i - \frac{L}{2}, p_i + \frac{L}{2}]$. By repeating the above operation for each IF signal, we can obtain the range of interest (red dashed box in Figure 3) by the following expression:

$$\boldsymbol{H} = \left\{ n \in \boldsymbol{N} : p_{min} - \frac{L}{2} \leq n \leq p_{max} + \frac{L}{2} \right\}, \tag{7}$$



(a) Range-Doppler map.

(b) Different velocity components during the signing.

(c) Time-velocity feature map.

Fig. 4. Feature extraction.

where p_{min} and p_{max} are the minimum and maximum values of all range bins in the frame, respectively. Then, we set the values outside the region of interest to zero to exclude other dynamic noises. We perform another FFT (i.e., Doppler FFT) on multiple IF signals in the slow-time dimension to obtain the velocity variation information during the signature process. Through the above steps, we can get the RDMs that reflect the user's signature execution process.

Figure 4(a) shows one RDM frame generated by the user during the signature execution process. Our obtained RDM usually contains two velocity components of equal magnitude and opposite direction. This is because the pen is divided into two parts, the upper part and the lower part, with the grip point (red dot in Figure 4(b)) as the center during the user's signature process. As shown in Figure 4(b), the velocity direction of the upper part is opposite to the hand movements direction, and the velocity direction of the lower part is the same as the hand movements direction. The intensity of each component of the RDM depends on the radar cross-section (RCS). Since the effective reflective area of the upper part of the pen is smaller than that of the hand, the intensity generated by the pen movement is smaller than that of the hand movement.

3.2.4 Time-velocity Feature Transformation. During the signature execution process, the change in distance from the user's hand to the mmWave radar is extremely small, which cannot be accurately sensed by the mmWave radar. Therefore, we use velocity change information during the signature execution process as the signature verification feature.

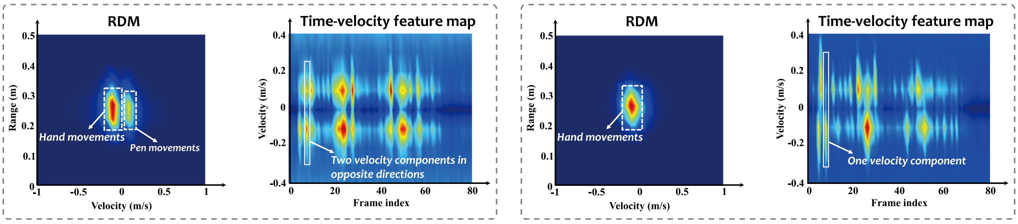
To obtain the velocity change information during the signature execution process, we use the following equation to transform the RDM of all frames into a 2D time-velocity feature map:

$$V_{(n,i)} = \frac{\sum_{j=1}^{N_R} [\boldsymbol{RDM}_{(n,i,j)} \cdot B_j]}{N_R}, \quad i \in [1, N_D], j \in [1, N_R], \tag{8}$$

where N_R is the number of Range FFT, N_D is the number of Doppler FFT, B_j is the range bin index, and $\boldsymbol{RDM}_{(n,i,j)}$ represents the value corresponding to Doppler bin i and range bin j in the n-th RDM frame. Figure 4(c) shows the time-velocity feature map we finally obtained, which reflects the velocity variation of the user's hand and pen during the signature execution process.

3.3 Data Augmentation

To improve the performance of the base model using limited data, we propose three data augmentation methods based on the variation characters of the time-velocity feature maps obtained from the mmWave signal during the signature execution process. The basic idea of data augmentation techniques is to synthesize new data by transforming existing labeled training samples so that the neural network model can learn a broader range of intra-class variations. By observing the



(a) Longer pen.

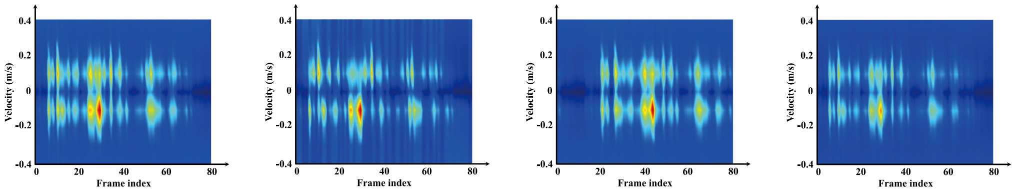
(b) Shorter pen.

Fig. 5. Feature maps obtained by different types of pens.

generated time-velocity feature maps, we find three common types of variations. The first type is the intensity variation of a specific component in the feature map due to different pen types and pen grip positions. The second type is the variation of reaction time and the signing speed. The third type is the magnitude variation of the user's hand movements when signing. Based on these observations, we design three data augmentation methods that can be efficiently implemented to augment the training set.

3.3.1 Augmenting Data with Velocity Transformation. As mentioned in Section 3.2.3, the pen's upper part generates a velocity component in the opposite direction of the hand movement during the signing process, whose intensity depends on the RCS of the pen's upper part. During the signature execution process, the RCS varies depending on the type of pen and the grip position, which results in different time-velocity feature maps obtained by the same user when signing with different types of pens. To investigate the impact of the pen's RCS, we use pens with different lengths to sign on the same surface. Specifically, we use a longer pen (14 cm) and a shorter pen (7 cm) to generate the corresponding time-velocity feature maps. As shown in Figure 5(a), due to the large RCS of the longer pen's upper part, the mmWave radar senses two velocity components with opposite directions. Therefore, two velocity components with opposite directions exist in the same frame of the final generated time-velocity feature map. However, the RCS of the shorter pen's upper part is very small (or equal to zero), and the mmWave radar cannot sense the velocity component caused by the pen's upper part. Therefore, only the velocity component induced by the hand movements exists in the obtained time-velocity feature map as shown in Figure 5(b). With the above analysis, we augment the data by changing the intensity of the pen movement velocity component. Specifically, for each RDM in Section 3.2.3, we divide it into two parts: the positive velocity part and the negative velocity part. We keep the velocity part generated by the hand movements fixed and multiply the other velocity part by a decay factor α within $[0, 1]$.

3.3.2 Augmenting Data with Time Transformation. We find that the following two factors have a significant impact on the extracted feature maps. **(1) Reaction time.** The user's reaction time when signing is inconsistent; thus, the start time of valid signature information may have different offsets. These temporal offsets can be achieved by translating the feature maps in the horizontal direction. Specifically, for the obtained time-velocity feature matrix, we first determine the signature start time t_1 and the end time t_2 by thresholding and then cyclically shift the elements in the time-velocity feature matrix by P elements, where P is less than the smaller of t_1 and t_2 to prevent the temporal features of the signature from being interrupted. **(2) Signing time.** The different signing speeds of the user can result in different signature execution times. These two factors can be changed by transforming the time-velocity feature map in the time-dimension for data augmentation. The difference in signature execution time can be achieved by stretching or compressing the original time-velocity feature. Specifically, we first compress or stretch the data between t_1 and



(a) Original data. (b) Velocity transformation. (c) Time transformation. (d) Magnitude transformation.

Fig. 6. Data augmentation. Based on the effect of different velocity, time, and amplitude variations on the mmWave signals during the signature execution process, we propose three data augmentation methods to improve the performance of the base model training.

t_2 by downsampling or interpolating to β times the original data, and then interpolate or downsample the data outside the range of $[t_1, t_2]$ to make sure that the length of the augmented data is the same as the original data, where β is the compress/stretch factor within $[-0.7, 1.3]$.

3.3.3 Augmenting Data with Magnitude Transformation. We find that the intensity of the time-velocity feature maps generated during the user's signature changes due to the magnitude of the user's hand movements. Generally, the greater the magnitude of the user's signature movements, the darker the color of the feature map *and vice versa*. Therefore, we simulate the effect of the user's different hand movement magnitudes during the signature execution process by changing the color range of the time-velocity feature map. Specifically, we first divide the entire feature map into eight segments equally by frame (ten frames per segment). Then, four of these segments are randomly selected for intensity transformation. For the selected segment, suppose that the color range is $[a, b]$; we can change the intensity of the feature map by adjusting the color range to $[a, (1 + \gamma)b]$, where γ is the transformation factor within $[-0.3, 0.3]$.

Figure 6 shows the time-velocity feature maps obtained using the above data augmentation methods. Note that the label of augmented data is the same as the original data, and each method has a parameter to adjust the level of data augmentation. All of these methods can be easily applied in the meta-training phase and generalized in the training dataset to accommodate the negative impact of signature inconsistencies and user-specific problems. The performances of the data augmentation will be evaluated in Section 4.4.

3.4 Signature Verification

After obtaining the time-velocity feature map of the signing process, we need to verify the authenticity of the signature. The time-velocity feature map is essentially a sequence of data, as it represents temporal information such as the velocity and the magnitude changes of the hand movements over time during the signature execution process. In mmSign, we design a transformer-based verification model to derive a high-level representation of the input time-velocity feature map and obtain the accurate signature verification result.

The architecture of our verification model is illustrated in Figure 7. The obtained time-velocity feature map is first transformed into multiple linear vectors with time information by patch embedding and time embedding. Then, these linear vectors are used as the input of the transformer encoder to obtain long-term dependencies among all the time patches. The multi-head self-attention (MSA) [62] is leveraged to serve as the primary primitive of the encoder, which reduces the dependence on external information and is superior in capturing the internal correlation of sequential data or features. After processing with the transformer encoder, we obtain a high-level representation of the input time-velocity feature map. Finally, we use a multi-layer perceptron (MLP) to

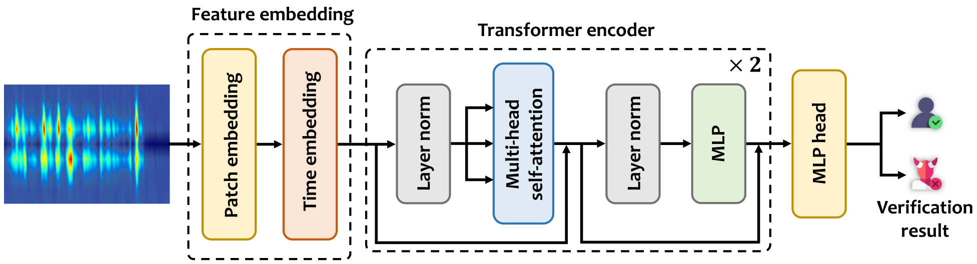


Fig. 7. Verification model. The obtained time-velocity feature map is first partitioned into multiple patches and appended with temporal information in the feature embedding block. Then, the time patches are fed into the transformer encoder to obtain a high-level representation. Finally, the verification result can be obtained by the MLP head.

perform binary classification and determine whether the input signature is generated by a legitimate user. We present the details of each module below.

3.4.1 Feature Embedding. The input time-velocity feature map first needs to be converted by the feature-embedding block into time patches, which are served as the input of the transformer encoder. Specifically, the feature-embedding block contains two layers: the patch embedding layer and the time embedding layer. The input of our model is the time-velocity feature map $\boldsymbol{m} \in \mathbb{R}^{H \times W \times C}$, where H, W, C are the height, width, and channel number of the time-velocity feature map, respectively. As mentioned above, the time-velocity feature map is essentially a temporal sequence and does not contain spatial information compared with the image input of the traditional vision transformer [18]. Therefore, we only process in the time dimension (i.e., width) and divide it into time patches $\boldsymbol{m_p} \in \mathbb{R}^{N \times (H \cdot T \cdot C)}$ by patch-embedding layer, where T is the width (time duration) of each patch and $N = W/T$ is the number of patches. We then flatten the time patches and map them to D dimensions using a linear projection with a parameter matrix $\boldsymbol{W} \in \mathbb{R}^{(H \cdot T \cdot C) \times D}$. A special classification token $\boldsymbol{m}_{cls} \in \mathbb{R}^D$ is attached to the beginning of embedded time patches to represent the meaning of the entire sequence [14]. The attention mechanism processes all the input patches in parallel, which means that the temporal information in the original feature map is lost. Therefore, a time-embedding layer is used to add temporal information for each time patch. Specifically, we add $\boldsymbol{E}_t = (\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_N)$ to each patch to retain the absolute temporal information, where $\boldsymbol{t}_i \in \mathbb{R}^D$. In summary, the flattened time patches after the feature embedding block can be expressed as

$$\boldsymbol{R} = \left[\boldsymbol{m}_{cls}; \boldsymbol{m}_p^1 \boldsymbol{W}; \boldsymbol{m}_p^2 \boldsymbol{W}; \ldots; \boldsymbol{m}_p^N \boldsymbol{W}\right] + \boldsymbol{E}_t = [\boldsymbol{r}_0; \boldsymbol{r}_1; \ldots; \boldsymbol{r}_N], \tag{9}$$

where $\boldsymbol{r}_i \in \mathbb{R}^D$ is the i-th time patch.

3.4.2 Transformer Encoder. The flattened time patches are then fed into a transformer encoder, which consists of alternating layers of MSA and MLP, with the layer normalization [5] connected by residual structures between each layer.

Multi-head self-attention block. The structure of the MSA block is shown in Figure 8. Multi-head attention extends the model's ability to focus on the different time duration of the input time-velocity feature map by jointly attending information from different representation subspaces. We apply multi-head attention with h heads, where the self-attention function is calculated h times. Given the flattened time patches $\boldsymbol{R}$ obtained by feature embedding, the trainable query matrix $\boldsymbol{W}^Q \in \mathbb{R}^{D \times D}$, key matrix $\boldsymbol{W}^K \in \mathbb{R}^{D \times D}$, and value matrix $\boldsymbol{W}^V \in \mathbb{R}^{D \times D}$ are first multiplied with the time patches $\boldsymbol{R}$ to obtain the query matrix $\boldsymbol{Q}$, key matrix $\boldsymbol{K}$, and value matrix $\boldsymbol{V}$. Then, the query

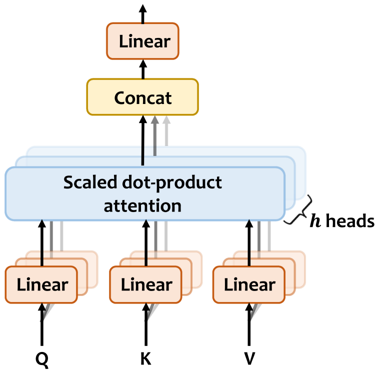


Fig. 8. Multi-head self-attention (MSA).

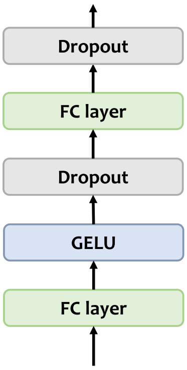


Fig. 9. Multi-layer perceptron (MLP).

matrix, key matrix, and value matrix are linearly projected h times with different, learned linear projections $\boldsymbol{W}_i^Q$, $\boldsymbol{W}_i^K$, and $\boldsymbol{W}_i^V \in \mathbb{R}^{D\times\frac{D}{h}}$ $(1 \le i \le h)$ to generate linear projected queries, keys, and values:

$$\boldsymbol{Q}_i = \boldsymbol{Q}\boldsymbol{W}_i^Q, \quad \boldsymbol{K}_i = \boldsymbol{K}\boldsymbol{W}_i^K, \quad \boldsymbol{V}_i = \boldsymbol{V}\boldsymbol{W}_i^V. \tag{10}$$

Next, the attention of each head is calculated for each group of $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ by the following equation:

$$\boldsymbol{head}_i = \text{Attention}(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) = \text{softmax}\left(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^\top}{\sqrt{D/h}}\right)\boldsymbol{V}_i. \tag{11}$$

By concatenating the output sequence $\boldsymbol{head}_i$ of each head, we can obtain the final output of the multi-head self-attention:

$$\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\boldsymbol{head}_1, \boldsymbol{head}_2, \ldots, \boldsymbol{head}_h)\boldsymbol{W}^O, \tag{12}$$

where $\boldsymbol{W}^O \in \mathbb{R}^{D\times D}$ is the linear projection matrix.

Multi-layer perceptron block. The output of the MSA block is fed to the MLP block after layer normalization. The MLP block is shown in Figure 9, which contains two fully connected (FC) layers, two dropout layers, and one GELU layer. The output of MLP is residually connected with the output of the MSA to obtain the transformer encoder output.

3.4.3 Model Outputs. After the transformer encoder, a high-level presentation $\boldsymbol{Z} \in \mathbb{R}^{(N+1)\times D}$ of the input time-velocity feature map is inferred. Note that we append a special classification token $\boldsymbol{m}_{cls}$ to the embedded time patches in the feature embedding block, which is used to represent the meaning of the entire input sequence (see Section 3.4.1). Therefore, we use the output of the transformer encoder $\boldsymbol{z}_0$ corresponding to the classification token as the input of the MLP head. The MLP head contains a fully connected layer to get the final verification result.

The transformer architecture presents a novel self-attention mechanism, which enhances its ability to capture global temporal features in the context of signature processing. Despite the input data being presented in image format, it inherently contains temporal information related to variations in the user's signature velocity and the RCS intensity caused by different signature postures at different time periods. The self-attention mechanism adeptly captures the internal relationships within this information, enabling the transformer structure to extract comprehensive time-dependent sequence features during the signature execution process. In addition, instead of directly adopting the patch embedding and position embedding methods in the traditional vision transformer ViT [18], we design a new feature-embedding scheme for the unique temporal nature of the mmWave feature maps we obtain, so as not to destroy the complete timing information

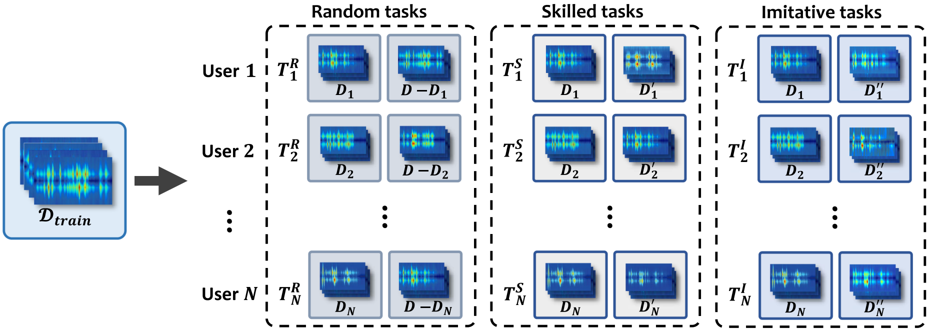


Fig. 10. Task generation. We propose a task generation strategy based on forged signature types, which utilizes random forged signatures and genuine signatures to form random tasks, skilled forged signatures and genuine signatures to form imitation tasks, and imitative forged signatures and genuine signatures to form imitative tasks, thereby improving the base model's ability to resist forgeries when new users adapt.

of particular time periods, as described in Section 3.4.1. Finally, using multiple heads ensures that the model can extract user-specific features from several different subspaces. These design choices ensure that the verification model extracts features with high accuracy and robustness.

3.5 Meta-learning-Based New User Adaptation

Although the proposed verification model can achieve good verification performance, like most existing handwritten signature verification schemes, it requires extensive signature collection during the new user registration phase, which is unrealistic and user-unfriendly. Therefore, how to reduce user signature collection efforts in the registration phase while maintaining strong system performance is an urgent problem to be solved. In mmSign, to reduce the data collection efforts, a meta-learning strategy is introduced to enable new users to quickly adapt to the verification system using only a few training samples.

3.5.1 Problem Formulation. Meta-learning [28] adopts the "learn to learn" concept and uses prior knowledge to generalize to new tasks with limited training samples rapidly. In mmSign, we formulate the problem as follows. We consider the signature verification task for each user as a meta-learning task. We have a source dataset $\mathcal{D}_{train}$, which is leveraged to generate the task set $\mathcal{T}$. Each task $T_i \in \mathcal{T}$ contains a support set S_{T_i} and a query set Q_{T_i}, which do not intersect ($S_{T_i} \cap Q_{T_i} = \varnothing$). Since each task $\mathcal{T}_i$ contains two types of signatures (i.e., genuine signatures and forged signatures), our problem is a 2-way K-shot problem, where K is the number of genuine or forged signatures in the support set and query set of each task. Therefore, the objective of our meta-learning scheme is to use the task set $\mathcal{T}$ generated by the source dataset $\mathcal{D}_{train}$ to train the base model to learn how to quickly adapt to the signature verification tasks for new users using only K labeled samples.

3.5.2 Task Generation. Existing meta-learning methods use random sampling from the large available dataset to generate multiple tasks for based model training [21, 58], which is inefficient in the field of signature verification, where the signature data is very limited. Therefore, how to effectively leverage the limited signature data to generate meta-learning tasks applicable to our signature verification problem is a unique challenge.

In the sector of signature verification, two types of traditional forgery attacks exist: random forgery attacks, where the attacker does not know the user's signature and uses a random signature

ALGORITHM 2: Meta-training Algorithm

```
   Input  : $\mathcal{D}_{train}$: source dataset for meta-training
            $\alpha$, $\beta$: learning rates
   Output : $\theta$: meta-learned weights of the base model
1  Randomly initialize $\theta$
2  while not done do
3      Generate three kinds of tasks using $\mathcal{D}_{train}$ to build the task set $\mathcal{T}$      ▷ See Section 3.5.2
4      foreach $T_i \in \mathcal{T}$ do
5          $S_{T_i} \leftarrow K$ support samples from $T_i$
6          $Q_{T_i} \leftarrow K$ query samples from $T_i$      ▷ $S_{T_i} \cap Q_{T_i} = \emptyset$
7          Evaluate $\nabla_\theta \mathcal{L}_{T_i}(f_\theta)$ with $S_{T_i}$
8          $\theta'_{T_i} \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{T_i}(f_\theta)$      ▷ Calculate task-specific parameters
9          Evaluate $\nabla_\theta \mathcal{L}_{T_i}(f_{\theta'})$ with $Q_{T_i}$
10     end
11     Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i \in \mathcal{T}} \mathcal{L}_{T_i}(f_{\theta'})$      ▷ Meta-update
12 end
```

instead; and skilled forgery attacks, where the attacker has access to the user's signature and performs an imitation [25, 31, 33]. In addition to traditional forgery attacks, our mmWave online signature verification system can be subject to a new type of attack, which we call an imitative forgery attack. In imitative forgery attacks, the attacker can obtain information about the user's signature execution process and tries to fool the system by imitating the user's signing process. Considering these three different types of forgery attacks, we design a task generation scheme to improve the resistance of the verification model to forgeries when new users adapt, instead of randomly generating tasks using the source dataset. Specifically, as shown in Figure 10, we generate three different types of tasks for each user u: random tasks T_u^R, which consist of the user's genuine data D_u and other user's data $D - D_u$ (i.e., random forgery signatures); skilled tasks T_u^S, which consist of the user's genuine data D_u and skilled data D'_u (i.e., skilled forgery signatures); and imitative tasks T_u^I, which consist of the user's genuine data D_u and imitative data D''_u (i.e., imitative forgery signatures). It is worth noting that since each user (or task) contains many genuine signatures and forgery signatures, each user can generate multiple random tasks, skilled tasks, and imitative tasks in the meta-training phase. These three task sets form our final task set $\mathcal{T}$. The effectiveness of our proposed task generation scheme will be evaluated in Section 4.5.

3.5.3 Meta-training. With the generated tasks in Section 3.5.2, we train the base model via meta-learning. Specifically, mmSign employs model-agnostic meta-learning (MAML) [21] to update the base model parameters. MAML can be applied to any gradient descent-based deep neural network with only a few gradient steps needed for model parameters updating. MAML assumes the existence of initial parameters that can be transferred to new tasks with only a few shots, and it performs initial parameter training with the goal of making the trained parameters adaptive to changes in different tasks.

For the data in each task T_i, we divide it into a support set S_{T_i} and a query set Q_{T_i}, each of which has K samples (K-shot). The base model training process is illustrated in Algorithm 2. For each task, we evaluate $\nabla_\theta \mathcal{L}_{T_i}(f_\theta)$ with K samples in S_{T_i} (Line 7). Then, the adapted task-specific parameters can be calculated as

$$\theta'_{T_i} = \theta - \alpha \nabla_\theta \mathcal{L}_{T_i}(f_\theta), \tag{13}$$

which is called the inner loop update, where $\nabla_\theta \mathcal{L}_{T_i}(f_\theta)$ is the cross-entropy loss in the task and is defined as

$$\mathcal{L}_{T_i}(f_\theta) = \sum_{(x_j, y_j) \in S_{T_i}} y_j \log f_\theta(x_j) + (1 - y_j) \log f_\theta(1 - x_j). \tag{14}$$

Then, the meta-objective function is defined as

$$\arg\min_\theta \sum_{T_i \in \mathcal{T}} \mathcal{L}_{T_i}(f_{\theta'}), \tag{15}$$

which is designed to find parameters θ that can minimize the sum of all the task losses, and each task loss is evaluated by Q_{T_i} (Line 9). Finally, stochastic gradient descent (SGD) is leveraged to minimize the meta-objective function and obtains the parameters θ:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i \in \mathcal{T}} \mathcal{L}_{T_i}(f_{\theta'}), \tag{16}$$

which is called the outer loop update. The base model with favorable initial parameters θ can be obtained after this process.

3.5.4 Model Adaptation. After the base model obtained by the above meta-training process, we fine-tune the base model using the new user's data, which contains only K-shot. The model adaptation process can be expressed as

$$\theta_u = \theta - \alpha \nabla_\theta \mathcal{L}_u(f_\theta), \tag{17}$$

where θ_u is the parameters of the new user's fine-tuned model.

4 EVALUATION

In this section, we first introduce the experimental setup and the data collection of mmSign. Then, we conduct a thorough experiment to demonstrate the performance of mmSign and its ability to withstand forgery attacks. Finally, we deploy mmSign on a Raspberry Pi to test its energy and time consumption for signature verification.

4.1 Experimental Setup

4.1.1 Implementation. As shown in Figure 11, we use a commercial FMCW radar AWR1642[1] and real-time data-capture adapter DCA1000EVM[2] for raw data collection. The default frame rate of mmWave radar is 10 FPS, and the number of chirp loops is 255. In addition, to verify the robustness of mmSign, we conduct experiments using four pens with different materials and lengths, and three different signature surfaces, as will be described in Section 4.8. The proposed verification model is trained offline on a desktop PC with an Intel i7-10700 CPU, 64 GB RAM, and RTX 3080 GPU. Keras 2.6.0 [12] with TensorFlow 2.6.0 [1] backend is used for model construction and training.

To evaluate the performance of mmSign, we consider the verification accuracy, which is widely adopted in the previous handwritten signature verification systems [16, 31]. Verification accuracy indicates the fraction of correctly classified signatures to the total number of signatures. A higher verification accuracy indicates that the system has better usability to correctly distinguish between genuine signatures and forged signatures. In addition, we report the false rejection rate (FRR) of our system, which represents the percentage of authentic signatures that are erroneously classified as forgeries by mmSign. A lower FRR signifies the system's improved capacity to correctly identify

[1] https://www.ti.com/product/AWR1642.
[2] https://www.ti.com/tool/DCA1000EVM.

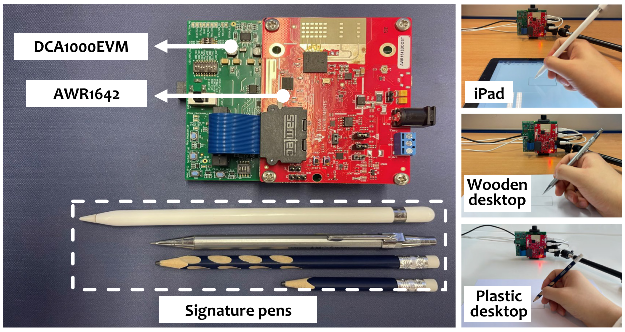


Fig. 11. Experimental devices. Four types of signature pens and three different signature surfaces are used to evaluate our system.

authentic signatures and to decrease the number of false rejections, which is crucial for ensuring a seamless and dependable user experience.

4.1.2 Attack Scenarios. We consider three types of forgery attacks performed by attackers: random forgery attacks, skilled forgery attacks, and imitative forgery attacks.

- Random forgery attacks. The attacker does not know any information about the user, and the attacker tries to use a random signature to fool the verification system.
- Skilled forgery attacks. The attacker knows the user's signature, but not the user's signature execution process, and the attacker tries to fool the verification system by imitating the user's signature.
- Imitative forgery attacks. The attacker observes the whole signature execution process when the user performs a signature through secretly filmed videos or shoulder-surfing. The attacker tries to fool the verification system by imitating the user's signature execution process.

The first two attacks are common attacks in the signature verification sector, whereas the third attack is specifically against our mmWave signature verification system. We will evaluate these three attacks in Section 4.9.

4.2 Data Collection

We recruited 30 volunteers, 18 males and 12 females, of different ages and hand sizes for data collection.[3] All volunteers were informed about how mmSign works, and each volunteer performed five signatures to familiarize oneself with mmSign before the official data collection. Before conducting the data collection, the volunteers signed the consent forms that clearly stated the purpose, procedure, and data usage of the study. We conducted experiments using an Apple pencil (Pen 1 in Table 2) and an iPad, with the signature box (4 cm by 2 cm) located directly in front of the mmWave radar at a distance of 30 cm in the office. The signatures we collected are in English, and we also evaluate the impact of different signature languages on our system in Section 4.7. The data collection is divided into the following two parts. **(1) Genuine signatures collection:** Each volunteer was required to provide 50 signature samples as genuine signatures. During the user's signature execution process, we used video to record the process for imitative forged signature collection. Therefore, we have 50 genuine signatures per volunteer. **(2) Forged signatures collection:** For

[3] Ethical approval has been granted by the corresponding organization (No. H002254).

each target volunteer, we randomly selected ten volunteers from the remaining 29 volunteers for the forged signature collection. As mentioned in Section 4.1.2, there are three types of forged signatures, which are random forgery, skilled forgery, and imitative forgery. We randomly chose five genuine signatures from each volunteer as random forgeries. For skilled forgeries, we asked each volunteer to imitate the signature of the target user five times. Finally, we asked the ten volunteers to watch the target user's signature video and imitate the signature execution process of the target user five times as imitative forgeries. As a result, we have 50 random forged signatures, 50 skilled forged signatures, and 50 imitative forged signatures for each target user.

The above raw data is augmented by our three data augmentation methods introduced in Section 3.3 to obtain our source dataset $\mathcal{D}_{train}$. Specifically, for each data augmentation method, we augment the original data twice using two different parameters. Therefore, for each volunteer, we have $50 \times 7 = 350$ genuine signatures, $50 \times 7 = 350$ random forged signatures, $50 \times 7 = 350$ skilled forged signatures, and $50 \times 7 = 350$ imitative forged signatures. These data will be used as the source dataset to generate tasks for meta-learning using our proposed task generation method in Section 3.5.2.

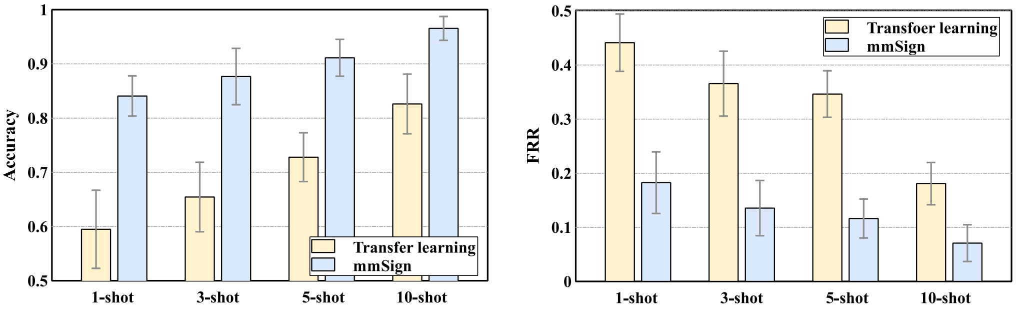
4.3 Overall Performance

We use the collected dataset to evaluate the performance of mmSign. The leave-one-volunteer-out training method is leveraged to evaluate the performance of mmSign. Specifically, we iteratively select one volunteer as the newly registered user and use the remaining 29 volunteers' data as the source data to train the base model. We report the average verification accuracy of all volunteers. Thus, we can assess whether our system is user-independent, that is, whether it works for newly registered users. Meanwhile, to evaluate the performance of our meta-learning framework, we compare mmSign with the transfer learning. For transfer learning, we train the base model with all data from 29 volunteers and fine-tune the base model with the new user's data. Since the random forged signatures in our dataset are composed of the genuine signatures of other users, we only use skilled forgeries as forged signatures during the base model training phase to avoid label conflicts in transfer learning.

The comparative performance of mmSign and transfer learning is depicted in Figure 12. When implementing transfer learning, the verification accuracy in scenarios of one-shot, three-shot, five-shot, and ten-shot stands at 59.49%, 65.44%, 72.79%, and 82.61%, correspondingly, whereas mmSign attains a verification accuracy of 84.07%, 87.31%, 91.12%, and 96.54% in the one-shot, three-shot, five-shot, and ten-shot settings, respectively. Additionally, as evidenced in Figure 12(b), the FRR of transfer learning is higher than that of mmSign by 25.87%, 23.20%, 20.88%, and 11.30% for the one-shot, three-shot, five-shot, and ten-shot scenarios, respectively. This disparity in performance can be attributed to the employment of a novel meta-learning approach by mmSign, which facilitates the acquisition of knowledge from multiple tasks within the task space, in contrast to transfer learning that only optimizes a single task. Consequently, mmSign exhibits a more efficient adaptation to new users with limited data.

4.4 Impact of Data Augmentation

In this experiment, we evaluate whether the proposed three data augmentation methods can enhance the performance of the base model during the meta-training process and thus improve the verification accuracy of newly registered users. We consider the following five different scenarios: (1) without data augmentation (w/o), (2) data augmentation with velocity transformation (w/v), (3) data augmentation with time transformation (w/t), (4) data augmentation with magnitude transformation (w/m), and (5) data augmentation with the above three methods (w/a).



(a) Accuracy.

(b) False rejection rate.

Fig. 12. Overall performance.

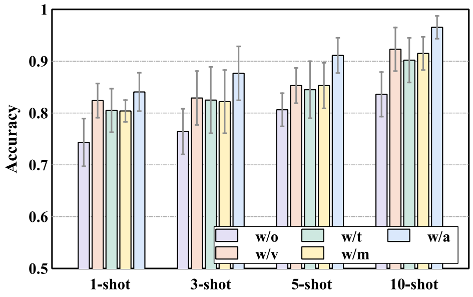


Fig. 13. Impact of data augmentation.

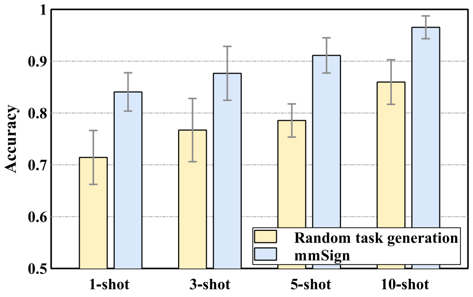


Fig. 14. Impact of task generation.

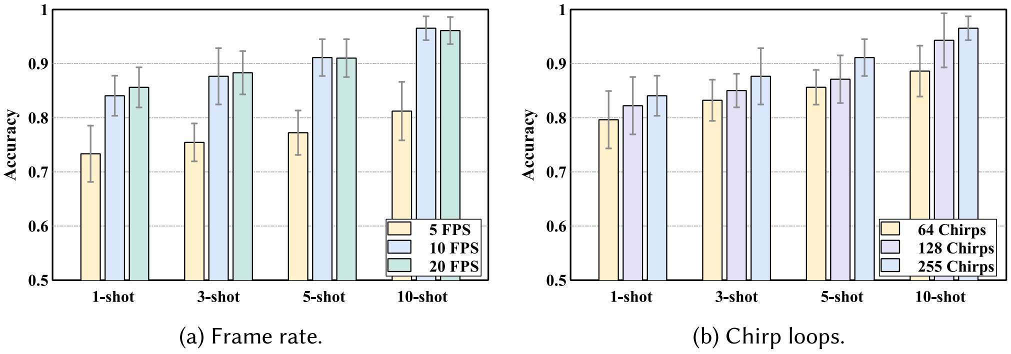
The evaluation results are shown in Figure 13. When no data augmentation strategy (i.e., w/o) is introduced, the accuracy of one-shot, three-shot, five-shot, and ten-shot is 74.33%, 76.42%, 80.63%, and 83.61%, respectively. The experimental results show that using only one data augmentation strategy (i.e., w/v, w/t, and w/m) or all three data augmentation strategies (i.e., w/a) can significantly enhance the performance of our system. The highest accuracy is achieved when all three data augmentation strategies are used simultaneously because the augmented source dataset is larger, allowing better coverage of different real-world scenarios and thus improving the learning ability of the base model.

4.5 Impact of Task Generation Method

We evaluate the effectiveness of the proposed task generation approach described in Section 3.5.2. We use random task generation from the source dataset as the baseline, which has been widely used in recent meta-learning methods [21, 58]. Specifically, the data in each task is selected randomly from the source data, without considering the type of forged signature. As shown in Figure 14, after using the proposed task generation scheme, the verification accuracy of one-shot, three-shot, five-shot, and ten-shot is improved from 71.43%, 76.71%, 78.56%, and 85.98% to 84.07%, 87.31%, 91.12%, and 96.54%, respectively. In comparison with random task generation, dividing the tasks into random tasks, skilled tasks, and imitative tasks can better improve the base model's ability to verify different kinds of forged signatures.

4.6 Impact of Radar Configuration Parameters

In this subsection, we evaluated the impact of different radar configuration parameters on the experimental results.



(a) Frame rate. (b) Chirp loops.

Fig. 15. Impact of radar configuration parameters.

Frame rate. We evaluate the impact of the mmWave radar frame rate. The frame rate of mmWave radar is set to 5 FPS, 10 FPS, and 20 FPS. As shown in Figure 15(a), when the mmWave radar frame rate changes from 5 FPS to 20 FPS, the verification accuracy under one-shot, three-shot, five-shot, and ten-shot settings increases from 73.35%, 75.45%, 77.24%, and 81.23% to 85.62%, 88.32%, 91.02%, and 96.11%, respectively. Higher frame rates provide more detailed samples of the signature execution process, but also require more computational resources for data processing. Since signing is a relatively fast process, 5 FPS is not enough to get the complete dynamic information, which results in the worst verification performance. Experimental results show that the difference between 20 FPS and 10 FPS in the ten-shot case is less than 0.5%, which is relatively small. Therefore, 10 FPS is sufficient to accurately capture the velocity change when signing.

Chirp loops. The impact of mmWave radar velocity resolution on the system is evaluated in this experiment. The velocity resolution is reflected in the number of chirp loops in each radar frame. As shown in Figure 15(b), with the increase of chirp loops from 64 to 128, the accuracy in the one-shot, three-shot, five-shot, and ten-shot settings increases from 79.64%, 83.23%, 85.63%, and 88.62% to 82.23%, 85.03%, 87.12%, and 94.31%, respectively. With the increase of chirp loops from 128 to 255, the accuracy in the above four settings increases from 82.23%, 85.03%, 87.12%, and 94.31% to 84.07%, 87.31%, 91.12%, and 96.54%, respectively. The improvement in accuracy can be explained as follows. The mmWave radar velocity resolution can be expressed as $\Delta v = \lambda/(2MT_c)$, where λ is the wavelength, M is the number of chirp loops, and T_c is the chirp period. When the chirp period T_c is fixed, the velocity resolution increases as the number of chirp loops increases, which means a better ability to distinguish velocity changes during the signature execution process.

4.7 Adaptability to Different Signature Types

Since the handwritten signatures of different users vary in language and complexity, in this section, we evaluate the adaptability of our system to different types of signatures.

Signature language. To assess the adaptability of our system to different language types, we test the adaptability of our system to Chinese signatures based on the base model trained by the source dataset. We recruit five additional volunteers and collect their Chinese signature data following the steps in Section 4.2. The average experimental results of the five volunteers are shown in Figure 16(a). The verification accuracy of Chinese signatures in one-shot, three-shot, five-shot, and ten-shot cases are 80.30%, 83.17%, 86.95%, and 91.68%, respectively, which is slightly lower than the verification accuracy of English signatures. The velocity-time feature map responds to variations in radial velocity during the signing, but not tangential velocity; Chinese signatures contain more lateral (tangential) strokes compared with English signatures, which results in less information obtained by mmWave radar when performing Chinese signatures than English signatures.

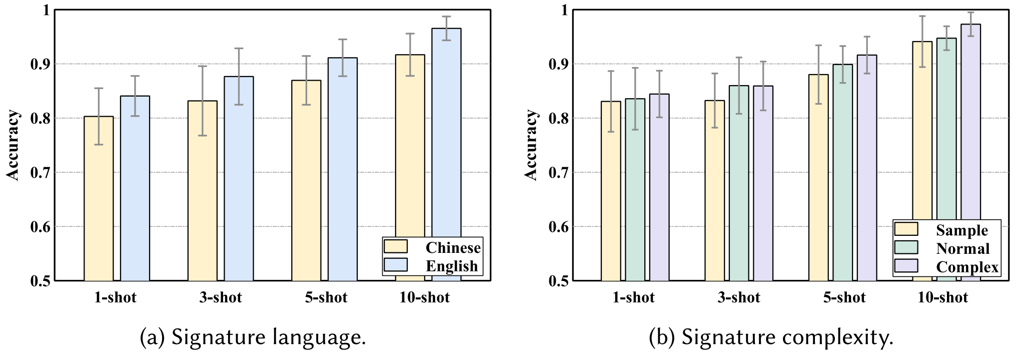


Fig. 16. Adaptability to different signature types.

Signature complexity. We evaluate the adaptability of our system to signatures of different complexity. Based on the number of letters in the name, we divide all samples into three categories, which are simple (less than 7 letters), normal (7–11 letters), and complex (more than 11 letters). For each category, we randomly selected three volunteers whose signatures met the criteria for verification. The verification results are shown in Figure 16(b). As we can see, when the signature complexity is increased from simple to complex, the verification accuracy increases by 1.4%, 2.7%, 3.6%, and 3.2% in the one-shot, three-shot, five-shot, and ten-shot cases, respectively. The verification accuracy of our system is higher in the case of high signature complexity because signatures with high complexity contain much richer dynamic information.

4.8 Adaptability to Different Scenarios

Users may perform signature verification in different scenarios. Thus, in this section, we evaluate the adaptability of mmSign to different real-world scenarios after new user registration is completed, which include different signature sizes, different relative positions of radar and signature box, different deployment environments, different signature pens, and different signature surfaces. Note that the base model is trained using the source dataset collected in Section 4.2, with the default scenario of size 2, position 0, office, pen 1, and iPad. We collect data from the other five new users in different scenarios and fine-tune the base model with the data from the default scenario. Then, the fine-tuned model is evaluated with data from other scenarios.

Signature size. Different signature scenarios may have different requirements for the size of the signature. Thus, we verify the adaptability of mmSign to different signature sizes in this experiment. The size of the signature box is divided into three types: size 1 (2 cm by 1 cm), size 2 (4 cm by 2 cm), and size 3 (8 cm by 4 cm). We use the signatures of size 2 for new user training, and then use the signatures of size 1 or size 3 for testing. The average verification accuracy corresponding to the three different sizes of the five new users is shown in Figure 18(a). Under size 1 and size 3, the verification accuracy is not significantly different from the default size (i.e., size 2), which demonstrates the robustness of our system to signature size.

Relative positions of radar and signature box. The adaptability of the system to the relative position of mmWave radar and the signature box is evaluated. As shown in Figure 17, we move the signature box from its default position (P0) to the four other positions (P1–P4), and the horizontal and vertical distance between the centers of two adjacent signature boxes is 15 cm. The signatures used for fine-tuning the base model are collected at P0, whereas the signatures used to test the fine-tuned model are collected at the other positions. The average results of the five users at different positions are shown in Figure 18(b). As we can see, the verification accuracy at P3 and P4 remains

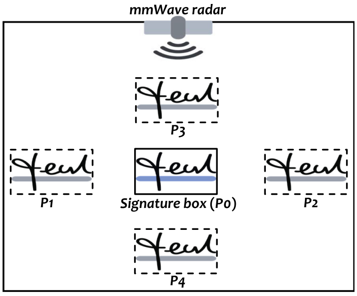


Fig. 17. Signature box positions.

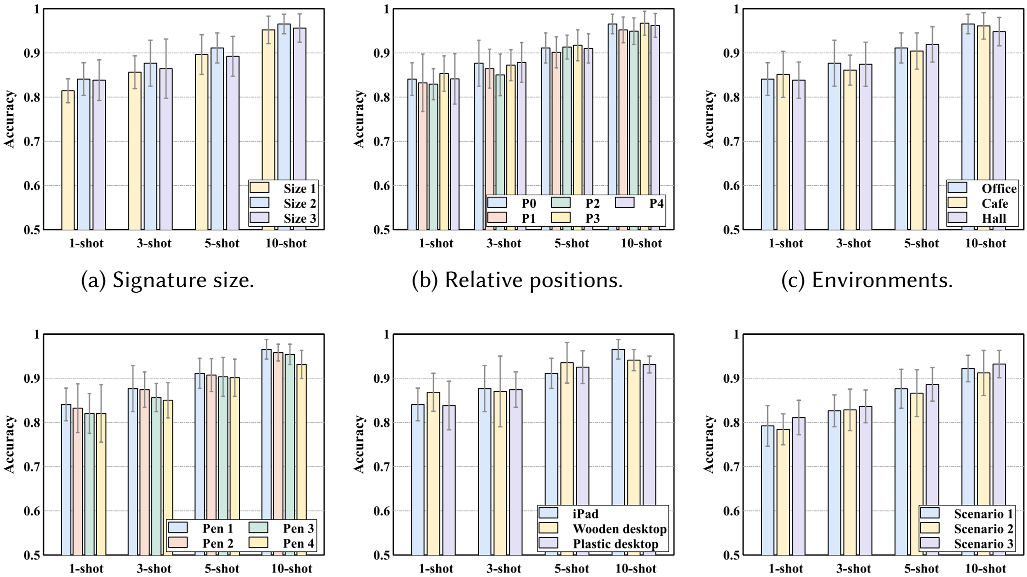
Table 2. Different Types of Pens for Signature

	Length (cm)	Diameter (cm)	Material
Pen 1	17.6	0.9	Plastic
Pen 2	14.0	0.8	Metal
Pen 3	14.0	0.7	Wood
Pen 4	7.0	0.7	Wood

similar to that at P0, which is due to the fact that the mmWave radar can accurately sense the radial velocity changes of the hand movements independent of the radial distance. Despite exhibiting the poorest performance at P1 and P2, mmSign demonstrates a mean verification accuracy decrease of less than 1.5% in comparison with P0. This phenomenon can be attributed to the fact that the mmWave radar is capable of detecting a significantly reduced radial velocity component despite the signature's location being positioned obliquely relative to the radar's direct line of sight. Thus, the radar is still able to capture the radial component associated with the signature's velocity. Additionally, as discussed in Section 3.2.3, our system obtains a feature map that reflects both the velocity variations of the user's hand and the pen's upper part, providing richer information for signature verification compared with existing acoustic-based solutions. Consequently, mmSign is more robust to changes in signing position than acoustic-based solutions.

Environments. Since signature verification may occur in various environments, we conduct experiments in different environments to verify the robustness of mmSign. We chose the office, cafe, and school hall for this experiment. There are people walking around during the data collection. The signatures collected in the cafe and school hall are used to test the fine-tuned model that is trained using data collected in the office. As shown in Figure 18(c), the verification accuracy in the cafe and school hall remains at a similar level compared with the accuracy in the office, where signatures from the same environment are used in both the training and testing phases. This is because our signal processing algorithms filter out other environmental interference, ensuring that the obtained time-velocity feature map contains only the information of the user's signature execution process.

Signature pens. As described in Section 3.2.3, the obtained time-velocity feature map contains the velocity component due to the opposite direction of the end part of the signature pen. In addition, signature pens of different materials may also affect the verification results due to their different reflection intensities on the mmWave radar. To verify the impact of different signature pens on our system, we choose four signature pens with various lengths and materials for our experiments, as shown in Table 2. The signatures obtained with Pen 1 are leveraged to fine-tune the base model, and the fine-tuned model is tested with signatures obtained using the other pens. Note that all signatures are performed on A4 paper on a wooden desktop, and since Pen 1 is an electronic pen, the process does not actually produce a signature image. The experimental results are shown in Figure 18(d), from which we observe that the accuracy of using a longer pen (i.e., Pen 2 and Pen 3) for signature is slightly higher than that of a shorter pen (i.e., Pen 4). This is because the time-velocity feature map obtained by using a longer pen contains more information, as introduced in Section 3.3.1. The accuracy of using the metal pen (i.e., Pen 2) for signature is slightly higher than that of using the wooden pen (i.e., Pen 3) due to the metal pen's larger RCS.



(a) Signature size. (b) Relative positions. (c) Environments.

(d) Signature pens. (e) Signature surfaces. (f) Random scenarios.

Fig. 18. Adaptability to different scenarios.

Signature surfaces. We evaluate the performance of our system on different signature surfaces, which simulates the actual situation of signing on different materials (e.g., paper, tablet). We choose three different surfaces: an iPad (default), A4 paper on a wooden desktop, and A4 paper on a plastic desktop. We perform three tests using Pen 1, which is an electronic pen and does not produce signature images. We train the model using signatures collected from the iPad and test it with signatures collected on other surfaces. The experimental results are shown in Figure 18(e). As we can see, the signature surface has a minimal impact on the verification accuracy of mmSign due to the system's reliance on the hand movements of the signer during the signature execution process, which is independent of the signature surface.

Random scenarios. In order to assess the adaptability of our system in an entirely unfamiliar scenario characterized by varying signature sizes, relative signature positions, signature pen types, and signature surface types, a random selection of scenarios is employed for experimental purposes. Specifically, three scenarios are identified and designated as follows: Scenario 1 with size 1, P4, cafe, pen 2, and plastic desktop; Scenario 2 with size 3, P1, hall, pen 4, and wooden desktop; and Scenario 3 with size 1, P3, hall, pen 3, and wooden desktop. The evaluation results are shown in Figure 18(f). Our evaluation indicates that simultaneous changes to multiple parameters have a notable impact on the accuracy of signature verification, in contrast to changes to a single parameter. However, even in an entirely unfamiliar scenario, the average accuracies of mmSign in one-shot, three-shot, five-shot, and ten-shot cases are 79.56%, 82.91%, 87.62%, and 92.21%, respectively. This outcome is commendable given the relatively minor time overheads (i.e., less than one and a half minutes for ten signatures) required for new user registration.

4.9 Security Analysis

In this section, we assess the resistance of mmSign to the three forgery attacks mentioned in Section 4.1.2.

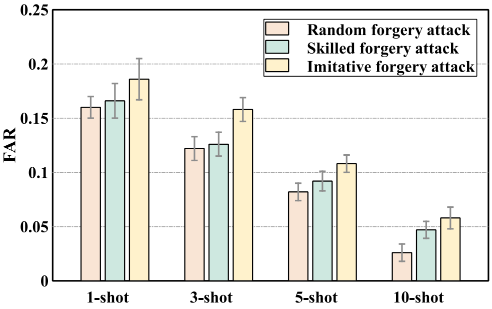


Fig. 19. Security analysis.

Table 3. System Overhead

Stage \ Performance	Computation time (s)	Energy consumption (mJ)
Static noise elimination	0.563	4.075
Sub-signal Generation	0.821	5.913
RDM generation	2.135	15.623
Time-velocity feature map generation	1.507	11.038
Signature verification	0.990	7.810
Total	**6.016**	**44.459**

We recruit ten volunteers as newly registered legitimate users and five volunteers as attackers to conduct forgery attacks. Each legitimate user performs ten signatures as genuine signatures, and the signing process is recorded on video for the attackers to imitate. For each target user, we randomly select two other users from the group of nine remaining users and gather three types of forged signatures from them, as described in Section 4.2. Therefore, we have ten samples for each type of forged signature. These signatures are used for base model fine-tuning. During the evaluation phase, the five attackers forge three different types of signatures for each legitimate user, with ten instances of each type of forgery. We report the false accept rate (FAR) of mmSign as a metric to assess the effectiveness of our system in resisting three kinds of forgery attacks.

As shown in Figure 19, for one-shot, three-shot, five-shot, and ten-shot cases, the FARs of mmSign are 16.0%, 12.2%, 8.2%, and 2.6% for random forgeries; 16.6%, 12.6%, 9.2%, and 4.7% for skilled forgeries; and 18.6%, 15.8%, 10.8%, and 5.8% for imitative forgeries, respectively. This is because the attackers can only imitate coarse-grained information such as hand movements and stroke order, and cannot imitate fine-grained information such as hand size, stroke interval, and signing velocity, which are also reflected in the obtained feature map and extracted as high-level features by the designed verification model as the basis for verification. To sum up, mmSign is resilient to common forgery attacks.

4.10 System Overhead

We train the model offline on the desktop PC and deploy it to the Raspberry Pi 4B to test the system overhead. We use a power monitor to evaluate the verification time and the energy consumption required for mmSign to verify a signature. The computation time and energy consumption of different stages are presented in Table 3. We can see the static noise elimination time, the sub-signal generation time, the RDM generation time, the time-velocity feature map generation time, and the verification time are 0.563 s, 0.821 s, 2.135 s, 1.507 s, and 0.990 s, respectively. Notably, the generation of RDMs necessitates performing multiple matrix FFT operations on all IF signals for every radar frame, which is the primary contributor to time overhead. Additionally, since the verification model requires image data as input, converting all RDM frames into a single time-velocity feature map and saving it as an image also incurs a relatively long processing time. The longer processing time is also the main reason for the larger energy consumption. However, the entire handwritten signature verification process is accomplished in approximately 6 s after receiving the raw data, and the total energy consumption is less than 45 mJ, demonstrating the efficiency and speed of mmSign.

5 USER STUDY

5.1 Recruitment and Design

To investigate the usability of mmSign, we further recruit 90 subjects (48 females and 42 males whose ages range from 15 to 59) to participate in the user study. It should be noted that these individuals are not involved in the previous studies. These individuals are not aware of any method we develop to prevent bias. Instead, they are told to evaluate the usability of mmSign by answering multiple questions. We present the approach of mmSign after requesting the consent of each subject to sign a consent form. Each subject then makes three signing attempts.

Following that, each participant evaluates the mmSign by responding to six questions that investigate usability across the following six aspects: ubiquity, security, privacy, efficiency, accuracy, and user-friendliness. The six questions are listed as follows: (1) I think the application of the verification method is ubiquitous; (2) I think the verification method is secure; (3) I think the verification method is privacy-preserving; (4) I think the verification method is efficient; (5) I think the verification method is accurate; and (6) I think the verification method is user-friendly. The responses range from 1 to 10 on a scale of strongly disagree to strongly agree for each item.

5.2 User Study Results

Figure 20 shows the statistical results of this user study. We can observe that mmSign achieves an average satisfaction score of over seven on all questions and close to nine on the three aspects of security, efficiency, and accuracy. Specifically, the average scores for ubiquity, security, privacy, efficiency, accuracy, and user-friendliness are 7.65 ± 2.26, 8.52 ± 1.89, 7.63 ± 2.39, 8.48 ± 1.84, 8.46 ± 1.14, and 7.52 ± 2.29, respectively. People's understanding of mmWave radar varies due to their differing levels of knowledge about this technology. Therefore, some subjects have questioned the ubiquity and privacy of mmWave radar. In addition, many subjects have not used an automatic signature verification system. Thus, even though mmSign only requires a small number of signatures for registration, some subjects still think it is time-consuming and labor-intensive, leading to poor user-friendliness. These findings suggest that there may be some limitations of the mmSign for certain users, particularly those with less familiarity with mmWave radar or automatic signature verification systems, which contribute to the more dispersed scores on questions (1), (3), and (6) in the user study. However, most users express satisfaction with mmSign in terms of security, accuracy, and efficiency after experiencing it, resulting in the scores for these questions being more concentrated. To wrap up, the overall scores of this user study indicate that users believe mmSign has good usability.

6 LIMITATION AND FUTURE WORK

Efficiency of the meta-learning module. In mmSign, the meta-learning-based new user adaptation module adopts the MAML [34] as the training algorithm to enable fast-adaptive few-shot learning. However, the MAML training process requires the calculation of a higher-order

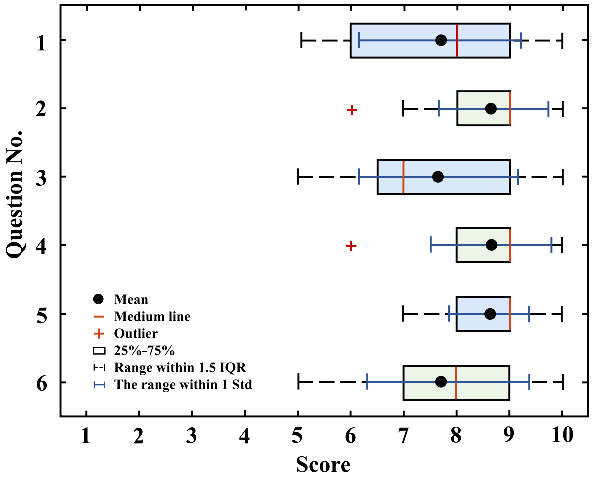


Fig. 20. User study results. 90 subjects are recruited to participate in a user study of mmSign, which has six questions on ubiquity, security, privacy, efficiency, accuracy, and user-friendliness. For each question, the responses span a scale from 1 (strongly disagree) to 10 (strongly agree).

derivative of the gradient, which can lead to high computational overhead [52]. In the future, we will use multi-step loss optimization for MAML [4], the implicit MAML algorithm [52], and task-adaptive MAML [6] to improve the efficiency of base model training for new user adaptation. In addition, the initial network model trained by using the MAML mechanism may be biased towards a subset of tasks generated in the meta-training stage and may lack the ability to generalize to new task domains. To alleviate this situation, we will use the task-independent meta-learning [34] algorithm to improve the generalization ability of the model.

Lack of forgery signatures. The implementation of new user registration in mmSign requires only a few training samples. However, in real-world scenarios, obtaining negative samples in the form of forged signatures poses a challenge, as the new user may only have access to their genuine signatures. Three types of forgery signatures exist in the system: random forgery signatures, skilled forgery signatures, and imitative forgery signatures. While we can replace random forgery signatures with genuine signatures of other users, the current stage restricts us from addressing skilled and imitative forgery signatures, and we rely on other users to perform imitation. Several approaches have been proposed to address the issue of binary classification problems that involve only one class of samples. These approaches include support vector domain description (SVDD) [53], PU learning [7], and generative adversarial network (GAN)–based methods [2, 29]. Nevertheless, the integration of these methods with few-shot learning poses a challenge in achieving accurate signature verification using a limited number of labeled samples, which will be further explored in our future work.

Robustness of mmSign in the long term. The design principle of mmSign is that when the user signs the same user name, the behaviors of different users are distinct, such that mmSign can distinguish among people from the collected mmWave sensory data. In mmSign, the specific-designed signal processing algorithm and the transformer-based meta-learning model make the handwritten signature verification resistant to adversarial attacks. However, a recent study [74] indicates that the behavior of individuals may change slightly over time, which poses a non-trivial challenge to mmSign. Therefore, the core of the solution to this challenge is how to make mmSign adaptable to the gradual changes in handwriting styles of different users while maintaining the accuracy of the verification. In the future, we will use domain adaptation methods [44, 76] and lifelong learning methods [3, 60] to enable mmSign to be adaptable to the gradual changes in handwriting styles.

7 CONCLUSION

In this article, we present an mmWave-based online handwritten signature verification system, mmSign, by extracting unique behavioral characteristics of handwritten signatures using commercial mmWave radar. Particularly, mmSign designs a series of novel signal processing algorithms to eliminate various noises and extract features from the raw signals during the signature extraction process. In addition, a meta-learning mechanism is introduced in mmSign to improve the adaptation performance of the transformer-based verification model for new users. Extensive evaluations in different real-world environments using various signing pens and surfaces demonstrate that mmSign achieves an average verification accuracy of 84.07%, 87.31%, 91.12%, and 96.54% in the one-shot, three-shot, five-shot, and ten-shot settings, respectively, while also effectively resisting common forgery attacks. To the best of our knowledge, mmSign is the first work to utilize mmWave signals for online handwritten signature verification, offering a new approach to the development of secure and reliable signature verification.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: A system for large-scale machine learning. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. 265–283.

[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P. Breckon. 2019. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Proceedings of Asian Conference on Computer Vision (ACCV)*. 622–637.

[3] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3366–3375.

[4] Antreas Antoniou, Harri Edwards, and Amos Storkey. 2019. How to train your MAML. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[6] Sungyong Baik, Janghoon Choi, Heewon Kim, Dohee Cho, Jaesik Min, and Kyoung Mu Lee. 2021. Meta-learning with task-adaptive loss function for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9465–9474.

[7] Jessa Bekker and Jesse Davis. 2020. Learning from positive and unlabeled data: A survey. *Machine Learning* 109 (2020), 719–760.

[8] Parkside Chambers. 2019. *Court of Appeal Clarifies the Burden of Proof for Forgery in Civil Cases.* Retrieved November 15, 2022 from http://www.parksidechambers.com.hk/court-of-appeal-clarifies-the-burden-of-proof-for-forgery-in-civil-cases.

[9] Mengqi Chen, Jiawei Lin, Yongpan Zou, Rukhsana Ruby, and Kaishun Wu. 2020. SilentSign: Device-free handwritten signature verification through acoustic sensing. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 1–10.

[10] Yongliang Chen, Tao Ni, Weitao Xu, and Tao Gu. 2022. SwipePass: Acoustic-based second-factor user authentication for smartphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–25.

[11] Zhe Chen, Tianyue Zheng, Chao Cai, and Jun Luo. 2021. MoVi-Fi: Motion-robust vital signs waveform recovery via deep interpreted RF sensing. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*. 392–405.

[12] Francois Chollet et al. 2015. *Keras.* Retrieved November 11, 2022 from https://github.com/fchollet/keras.

[13] Hewitt D. Crane and John S. Ostrem. 1983. Automatic signature verification using a three-axis force-sensitive pen. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13, 3 (1983), 329–337.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*. 4171–4186.

[15] Sounak Dey, Anjan Dutta, J Ignacio Toledo, Suman K. Ghosh, Josep Lladós, and Umapada Pal. 2017. SigNet: Convolutional Siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131* (2017).

[16] Moises Diaz, Miguel A. Ferrer, Donato Impedovo, Muhammad Imran Malik, Giuseppe Pirlo, and Réjean Plamondon. 2019. A perspective analysis of handwritten signature technology. *ACM Computing Surveys* 51, 6 (2019), 1–39.

[17] Feng Ding, Dong Wang, Qian Zhang, and Run Zhao. 2019. ASSV: Handwritten signature verification using acoustic signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–22.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[19] J.-P. Drouhard, Robert Sabourin, and Mario Godbout. 1996. A neural network approach to off-line signature verification using directional PDF. *Pattern Recognition* 29, 3 (1996), 415–424.

[20] Li Feifei, Robert Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 4 (2006), 594–611.

[21] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1126–1135.

[22] Taesik Gong, Yeonsu Kim, Jinwoo Shin, and Sung-Ju Lee. 2019. MetaSense: Few-shot adaptation to untrained conditions in deep mobile sensing. In *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys)*. 110–123.

[23] Tianbo Gu, Zheng Fang, Zhicheng Yang, Pengfei Hu, and Prasant Mohapatra. 2019. mmSense: Multi-person detection and identification via mmWave sensing. In *Proceedings of the ACM Workshop on Millimeter-Wave Networks and Sensing Systems (mmNets)*. 45–50.

[24] Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. 2017. Offline handwritten signature verification–Literature review. In *Proceedings of the International Conference on Image Processing Theory, Tools and Applications (IPTA)*. 1–8.

[25] Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. 2019. Meta-learning for fast classifier adaptation to new users of signature verification systems. *IEEE Transactions on Information Forensics and Security* 15 (2019), 1735–1745.

[26] Mingda Han, Linlin Guo, Jia Zhang, Hui Ji, Zihan Diao, and Jiande Sun. 2022. WiID: Precise WiFi-based person identification via bio-electromagnetic information. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*. 1105–1112.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[28] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2021. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2021), 5149–5169.

[29] Ming Hou, Brahim Chaib-Draa, Chao Li, and Qibin Zhao. 2017. Generative adversarial positive-unlabelled learning. *arXiv preprint arXiv:1711.08054* (2017).

[30] Pengfei Hu, Yifan Ma, Panneer Selvam Santhalingam, Parth H. Pathak, and Xiuzhen Cheng. 2022. MILLIEAR: Millimeter-wave acoustic eavesdropping with unconstrained vocabulary. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*. 11–20.

[31] Donato Impedovo and Giuseppe Pirlo. 2008. Automatic signature verification: The state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 5 (2008), 609–635.

[32] J. P. Morgan. 2022. *2022 AFP Payments Fraud and Control Report.* Retrieved November 15, 2022 from https://www.afponline.org/publications-data-tools/reports/survey-research-economic-data/Details/payments-fraud.

[33] Anil K. Jain, Friederike D. Griess, and Scott D. Connell. 2002. On-line signature verification. *Pattern Recognition* 35, 12 (2002), 2963–2972.

[34] Muhammad Abdullah Jamal and Guojun Qi. 2019. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11719–11727.

[35] Chengkun Jiang, Junchen Guo, Yuan He, Meng Jin, Shuai Li, and Yunhao Liu. 2020. mmVib: Micrometer-level vibration measurement with mmWave radar. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*. 1–13.

[36] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, et al. 2018. Towards environment independent device free human activity recognition. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*. 289–304.

[37] Nidal S. Kamel, Shohel Sayeed, and Grant A. Ellis. 2008. Glove-based approach to online signature verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 6 (2008), 1109–1113.

[38] Alisher Kholmatov and Berrin Yanikoglu. 2005. Identity authentication using improved online signature verification method. *Pattern Recognition Letters* 26, 15 (2005), 2400–2408.

[39] Guohao Lan, Bailey Heit, Tim Scargill, and Maria Gorlatova. 2020. GazeGraph: Graph-based few-shot cognitive context sensing from human visual behavior. In *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys)*. 422–435.

[40] Alona Levy, Ben Nassi, Yuval Elovici, and Erez Shmueli. 2018. Handwritten signature verification using wrist-worn devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (2018), 1–26.

[41] Huining Li, Chenhan Xu, Aditya Singh Rathore, Zhengxiong Li, Hanbin Zhang, Chen Song, Kun Wang, Lu Su, Feng Lin, Kui Ren, et al. 2020. VocalPrint: Exploring a resilient and secure voice authentication via mmWave biometric interrogation. In *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys)*. 312–325.

[42] Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, Bing Wang, Changhao Chen, John A. Stankovic, Niki Trigoni, and Andrew Markham. 2020. See through smoke: Robust indoor mapping with low-cost mmWave radar. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*. 14–27.

[43] Marcos Martinez-Diaz, Julian Fierrez, Ram P. Krish, and Javier Galbally. 2014. Mobile signature verification: Feature robustness and performance comparison. *IET Biometrics* 3, 4 (2014), 267–277.

[44] Yu Mitsuzumi, Go Irie, Daiki Ikami, and Takashi Shibata. 2021. Generalized domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1084–1093.

[45] Kenrick Mock, Bogdan Hoanca, Justin Weaver, and Mikal Milton. 2012. Real-time continuous iris recognition for authentication using an eye tracker. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*. 1007–1009.

[46] Daigo Muramatsu and Takashi Matsumoto. 2007. Effectiveness of pen pressure, azimuth, and altitude features for online signature verification. In *Proceedings of the International Conference on Biometrics (ICB)*. 503–512.

[47] CBS News. 2016. *Slamming "Bizarre" Law, Judge Rules on Florida's Vote-by-mail Ballots.* Retrieved November 15, 2022 from https://www.cbsnews.com/news/florida-voters-can-fix-vote-by-mail-ballot-federal-judge-rules.

[48] CSO Security News. 2018. *Busted! Cops Use Fingerprint Pulled from a WhatsApp Photo to ID Drug Dealer.* Retrieved November 19, 2022 from https://www.csoonline.com/article/3268837/busted-cops-use-fingerprint-pulled-from-a-whatsapp-photo-to-id-drug-dealer.html.

[49] Luiz S. Oliveira, Edson Justino, Cinthia Freitas, and Robert Sabourin. 2005. The graphology applied to signature verification. In *Proceedings of the Conference of the International Graphonomics Society (IGS)*. 286–290.

[50] Felix Ott, Mohamad Wehbi, Tim Hamann, Jens Barth, Björn Eskofier, and Christopher Mutschler. 2020. The OnHW dataset: Online handwriting recognition from IMU-enhanced ballpoint pens with machine learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–20.

[51] A. B. M. Mohaimenur Rahman, Yetong Cao, Xinliang Wei, Pu Wang, Fan Li, and Yu Wang. 2022. PPGSign: Handwritten signature authentication using wearable PPG sensor. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*. 2721–2726.

[52] Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. *Advances in Neural Information Processing Systems* 32 (2019).

[53] Carolina Sanchez-Hernandez, Doreen S. Boyd, and Giles M. Foody. 2007. One-class classification for mapping a specific land-cover class: SVDD classification of fenland. *IEEE Transactions on Geoscience and Remote Sensing* 45, 4 (2007), 1061–1073.

[54] M. C. Schoeman-Malan. 2015. Fraud and forgery of the testator's will or signature: The flight from formalities to no formalities. *Journal of South African Law/Tydskrif vir die Suid-Afrikaanse Reg* 2015, 1 (2015), 125–152.

[55] Hiroki Shimizu, Satoshi Kiyono, Takenori Motoki, and Wei Gao. 2004. An electrical pen for signature verification using a two-dimensional optical angle sensor. *Sensors and Actuators A: Physical* 111, 2-3 (2004), 216–221.

[56] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[57] Akash Deep Singh, Sandeep Singh Sandha, Luis Garcia, and Mani Srivastava. 2019. RadHAR: Human activity recognition from point clouds generated through a millimeter-wave radar. In *Proceedings of the ACM Workshop on Millimeter-Wave Networks and Sensing Systems (mmNets)*. 51–56.

[58] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems* 30 (2017).

[59] Amir Soleimani, Babak N. Araabi, and Kazim Fouladi. 2016. Deep multitask metric learning for offline signature verification. *Pattern Recognition Letters* 80 (2016), 84–90.

[60] Gan Sun, Yang Cong, and Xiaowei Xu. 2018. ". In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 32.

[61] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1199–1208.

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).

[63] Chao Wang, Feng Lin, Tiantian Liu, Ziwei Liu, Yijie Shen, Zhongjie Ba, Li Lu, Wenyao Xu, and Kui Ren. 2022. mmPhone: Acoustic eavesdropping on loudspeakers via mmWave-characterized piezoelectric effect. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*. 820–829.

[64] Dazhuo Wang, Jianfei Yang, Wei Cui, Lihua Xie, and Sumei Sun. 2022. CAUTION: A robust WiFi-based human authentication system via few-shot open-set gait recognition. *IEEE Internet of Things Journal* 9, 18 (2022), 17323–17333.

[65] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. 2019. Few-shot video-to-video synthesis. *arXiv preprint arXiv:1910.12713* (2019).

[66] Wei Wang, Alex X. Liu, and Ke Sun. 2016. Device-free gesture tracking using acoustic signals. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*. 82–94.

[67] Xuyu Wang, Chao Yang, and Shiwen Mao. 2017. PhaseBeat: Exploiting CSI phase data for vital sign monitoring with commodity WiFi devices. In *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*. 1230–1239.

[68] Yao Wang, Tao Gu, Tom H. Luan, Minjie Lyu, and Yue Li. 2022. HeartPrint: Exploring a heartbeat-based multiuser authentication with single mmWave radar. *IEEE Internet of Things Journal* 9, 24 (2022), 25324–25336.

[69] Yao Wang, Tao Gu, Tom H. Luan, and Yong Yu. 2022. Your breath doesn't lie: Multi-user authentication by sensing respiration using mmWave radar. In *Proceedings of the IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 64–72.

[70] Yanwen Wang, Jiaxing Shen, and Yuanqing Zheng. 2020. Push the limit of acoustic gesture recognition. *IEEE Transactions on Mobile Computing* 21, 5 (2020), 1798–1811.

[71] Michael Whyte. 2015. *Fingerprint Identifications from Images and Video Recovered from Cell Phones.* Retrieved November 19, 2022 from http://www.in-the-loop.net.au/fp-identifications-from-digital-media.

[72] Cong Wu, Kun He, Jing Chen, Ziming Zhao, and Ruiying Du. 2020. Liveness is not enough: Enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks. In *Proceedings of the USENIX Security Symposium (USENIX Security)*. 2219–2236.

[73] Rui Xiao, Jianwei Liu, Jinsong Han, and Kui Ren. 2021. OneFi: One-shot recognition for unseen gesture via COTS WiFi. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 206–219.

[74] Weitao Xu, Yiran Shen, Yongtuo Zhang, Neil Bergmann, and Wen Hu. 2017. Gait-watch: A context-aware authentication system for smart watch based on gait recognition. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. 59–70.

[75] Weiye Xu, Wenfan Song, Jianwei Liu, Yajie Liu, Xin Cui, Yuanqing Zheng, Jinsong Han, Xinhuai Wang, and Kui Ren. 2022. Mask does not matter: Anti-spoofing face authentication using mmWave without on-site registration. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*. 310–323.

[76] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. 2021. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 8978–8987.

[77] Xin Yang, Jian Liu, Yingying Chen, Xiaonan Guo, and Yucheng Xie. 2020. MU-ID: Multi-user identification through gaits using millimeter wave radios. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*. 2589–2598.

[78] Zhicheng Yang, Parth H. Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra. 2017. Vital sign and sleep monitoring using millimeter wave. *ACM Transactions on Sensor Networks* 13, 2 (2017), 1–32.

[79] Kumiko Yasuda, Daigo Muramatsu, Satoshi Shirato, and Takashi Matsumoto. 2010. Visual-based online signature verification using features extracted from video. *Journal of Network and Computer Applications* 33, 3 (2010), 333–341.

[80] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. 2020. DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12203–12213.

[81] Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* (2015).

[82] Run Zhao, Dong Wang, Qian Zhang, Xueyi Jin, and Ke Liu. 2021. Smartphone-based handwritten signature verification using acoustic signals. *Proceedings of the ACM on Human-Computer Interaction* 5, ISS (2021), 1–26.

Received 27 December 2022; revised 23 April 2023; accepted 20 June 2023