StrokePIN: Enhancing PIN Authentication With Keystroke Dynamics for Mobile Devices

Jingyu Yao and Ding Wang

***Abstract*—Keystroke dynamics-based authentication is a promising approach to enhance the security of personal identification number (PIN)-based authentication systems for mobile devices. While its effectiveness has been extensively studied, due to the limitations on the number of samples users can provide, most research struggles with the trade-off between system performance and user experience. Additionally, little effort has been devoted to quantifying the security of PIN keystroke dynamics. In this paper, we present StrokePIN, a user-friendly and efficient authentication system that utilizes multi-modality data. Specifically, we leverage a few-shot learning technique, called Siamese Network, which enables lightweight deployment of the system without retraining. To evaluate StrokePIN, we design a set of experiments and collect two new multi-modality datasets of keystroke dynamics of 20 PINs from 116 users. These datasets are the only publicly available resources of their kind to date, and we have made them accessible online. Furthermore, we quantify the security of keystroke dynamics of PINs with entropy, revealing its security boundaries. The evaluation results show that compared to the baselines, StrokePIN achieves state-of-the-art performance with False Acceptance Rate (FAR) of 2.2% and False Rejection Rate (FRR) of 1.9% on unseen users. For unseen PINs, StrokePIN achieves an FAR of 2.6% and an FRR of 1.4%. Additionally, by dynamically updating the template library, StrokePIN can mitigate the impact of user behavior drift over time, achieving the performance with FAR of 8.3% and FRR of 0.4%. Our security analysis results indicate that keystroke dynamics can provide 4.03-5.83 bits of security against 3 to 10 online guessing attacks.**

***Index Terms*—Keystroke dynamics-based authentication, PIN-based authentication, mobile devices, implicit authentication.**

I. Introduction

THE rise of mobile services such as financial transactions and confidential information storage necessitates robust user authentication to ensure legitimate user access. Given the potential limitations of biometric authentication, devices still require personal identification numbers (PINs). Biometric authentication does not fully replace knowledge-based authentication [1]. However, PINs are vulnerable to various attacks, such as online-guessing attacks [1], [2], [3], shoulder surfing attacks [4], [5] and side-channel attacks [6], [7], [8]. For example, Bailey et al. [2] showed that online-guessing attacks can crack 10% of PINs (13% for 4-digit PINs and 7% for 6-digit PINs) within five attempts. The success rate of shoulder surfing attacks [4] is 43.8% with only one view. These attacks pose persistent threats to PIN-based authentication.

Although the blacklist [1] may provide a balance between usability and security, the rapid advancement of artificial intelligence [8], [9] is posing new threats to the fundamental security of PINs. Specifically, Shukla et al. [9] demonstrated that an attacker can recover over 70% of all characters in a password by analyzing the spatio-temporal movements of the user's hand during typing. Moreover, Berend et al. [8] indicated that an attacker can achieve an alarming 83.7% success rate within 20 tries in a single-user scenario by analyzing sensor data. The increasing potential risks further underscore the necessity for more secure authentication schemes.

Keystroke dynamics-based authentication [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], a non-intrusive verification mechanism, has gained considerable attention for improving the security of PIN-based authentication. Extensive research [26], [27], [28] indicate that the signals from sensors during user interactions with smartphones can reflect unique physiological characteristics and behavioral habits, such as the force, rhythm, and size of the touch area. Using built-in sensors, one can capture distinctive behavioral patterns in an imperceptible manner, thus preventing unauthorized access without compromising the user experience. While existing approaches have made pioneering contributions to the field, three key challenges remain that may hinder their practical deployment in real-world scenarios.

The first challenge is the time-consuming and fixed registration phase. Existing studies implicitly require users to provide sufficient samples (e.g., 85 [12] and 96 [23]) during the registration phase to build an effective model. Unfortunately, the variety of postures further increases user burden, since they need to provide sufficient templates for each posture. In addition, in real-world scenarios, some users prefer fingerprint or facial recognition for unlocking. The low frequency with which users use PINs to unlock may not be sufficient to support the system in collecting enough information within a few days.

The second challenge is the limited feature engineering. Most works in the literature [12], [13], [14], [15], [16], [17], [18], [21], [23], [24] require extracting features from each sensor's data

Received 15 October 2025; revised 9 January 2026; accepted 18 February 2026. This work was supported in part by the Natural Science Foundation of Tianjin, China under Grant 25JCJQJC00230 and in part by the National Natural Science Foundation of China under Grant 62572259. *(Corresponding author: Ding Wang.)*

The authors are with the College of Cryptology and Cyber Science, Nankai University, Tianjin 300350, China, also with the State Key Laboratory of Cryptology, Beijing 100878, China, and also with the TBI Center, NDST, and DISSec, Nankai University, Tianjin 300350, China (e-mail: jingyuyao@mail.nankai.edu.cn; wangding@nankai.edu.cn).

Digital Object Identifier 10.1109/TDSC.2026.3667918

manually and separately, which can better capture discriminative features when the sample size is limited. However, this may neglect the intrinsic relationships between data from different sensors, thereby failing to capture fine-grained behavioral patterns.

The third challenge is the dilemma between model performance and user experience. Since *adversarial data distributions are unknown a priori*, One-class classification [14], [22], [23] and distance measurement [10], [12], [13], [14], [15], [16], [17], [18], [21] are widely used in the literature (e.g., euclidean Distance [13], [15], [16], [17], [18], Manhattan Distance [14], [15], [16], [17], [18], [21], [23], Nearest Neighbor Distance [12] and One-Class K-Nearest Neighbors [22]). However, improving the performance of a model requires collecting a great quantity of samples, which may compromise user experience. In addition, several studies [24], [25] have developed binary classification models using deep learning technology. Specifically, El-Kenawy et al. [24] build authentication model based on bidirectional recurrent neural network, Kim et al. [25] designed a model using convolutional neural network and achieved 6.7% EER using 2,400 training samples and 1,600 test samples. *Collecting sufficient data during registration may remain impractical in real-world deployments*, especially in the scenario where user postures are various. Limited sample sizes may prevent models from attaining desired performance, thus compromising deployment.

In this paper, we present StrokePIN, which achieves user-friendly deployment with efficient authentication. StrokePIN has excellent scalability and can be easily deployed on smartphones without the need to retrain user-specific models. The key insight of StrokePIN is that by 1) only requiring a small number of samples during registration, 2) learning deep features from cross-modal latent representations, and 3) achieving efficient authentication with limited enrolled templates.

It is challenging to follow existing frameworks while achieving lightweight deployment and effective verification. Therefore, we design a new framework StrokePIN utilizing the Siamese Network [29], which enables the fully trained model to be deployed on the client side without the need for retraining. Specifically, StrokePIN employs a server-client architecture, where a service provider is responsible for training a user-agnostic universal feature extractor and classifier, users need to provide a small number of templates locally. StrokePIN lies in the feature extractor's ability to extract discriminative features from unseen data, thereby eliminating the need for prior access to target users' data. This significantly streamlines deployment and enhances practical utility.

With the generalization capability of authentication model, StrokePIN also supports dynamic enrollment during operation. During login, when a user enters a correct PIN in an unseen pattern, the system will automatically transition to the biometric-based authentication system, such as fingerprint or facial recognition. Following successful authentication, the PIN entry sample associated with the new pattern is recorded for future verification. This workflow significantly streamlines StrokePIN's enrollment procedure, making it capable of dynamically adapting to changes in behavioral postures.

To further enhance the authentication performance, we design a feature fusion mechanism that can automatically learn deep characteristics from cross-modal latent representations. Specifically, we design two dedicated feature extractors: one for touch sensors and another for motion sensors. The subsequent fusion module can learn cross modal features from them.

We conduct a comprehensive evaluation of StrokePIN using in-lab datasets. For unseen users, StrokePIN achieves a FAR of 0.022 and a FRR of 0.019 at the optimal threshold. For unseen PINs, it achieves a FAR of 0.026 and a FRR of 0.014 at the optimal threshold. Moreover, on the cross-session test set, StrokePIN achieves a FRR of 0.04 and a FAR of 0.083. These results demonstrate StrokePIN's excellent generalization capability. Furthermore, we quantify the security of keystroke dynamics of PINs using β-success-rate and α-guesswork metrics. The results show that keystroke dynamics can provide 4.03-5.83 bits of security when β ranges from 3 to 10. For α-guesswork metrics, keystroke dynamics can provide 3.40-4.56 bits of security when α is 0.2. Finally, we evaluate the impact of the feature fusion mechanism. The results show that this mechanism can reduce the error rates of StrokePIN by 9.4% to 37.9%.

The contributions of this paper are as follows:

- This paper presents a new authentication framework, StrokePIN, which supports efficient deployment without retraining on the client side, achieving lightweight registration. Compared to the relevant works in the literature, StrokePIN achieves the best performance.
- We collect two multi-modal datasets of keystroke dynamics for PINs. They are available online .[1] To the best of our knowledge, *they are the only publicly accessible multi-modal datasets of keystroke dynamics for PINs to date*. We expect that these datasets will provide valuable support for future research, facilitating the development and validation of more robust and sound schemes.
- Through experiments on 29,651 recordings of user keystrokes, we evaluate the impact of unseen users, unseen PINs, and temporal factors on StrokePIN. Additionally, we quantify the security of keystroke dynamics of PINs. The entropy analysis results reveal that keystroke dynamics can offer 4.03 to 5.83 bits of security against online dictionary attacks with up to 10 attempts. When evaluated under the α-guesswork metric with α=0.2, keystroke dynamics provides a security level equivalent to 3.40-4.56 bits. We have released the core code on the GitHub .[2]

II. Background and Motivation

In this section, we present the background and motivation of this paper, highlighting the necessity of our work.

A. The Dilemma of PIN-Based Authentication

Despite the rise of biometric-based authentication, such as fingerprint or facial recognition, devices still require PINs after

[1] https://github.com/Litraveler/StrokePIN.git, DOI: 10.21227/thft-4721, 10.21227/s258-3h11

[2] https://github.com/Litraveler/StrokePIN.git

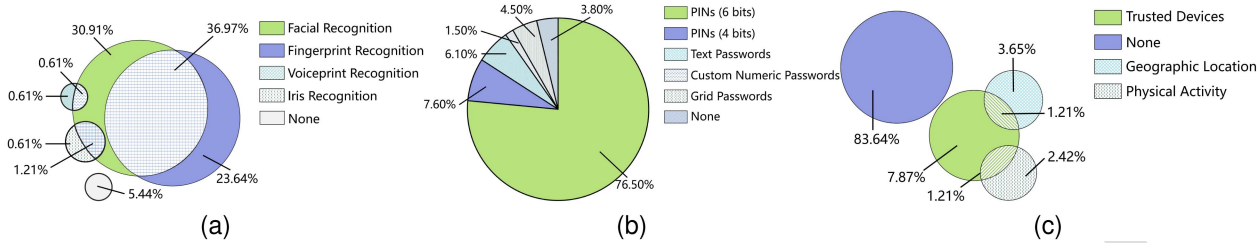

Fig. 1. Our survey results of authentication methods used by 165 users. While employing Biometric-based Authentication, 96.2% of them simultaneously configured Knowledge-based Authentication, and 76.5% opted for a 6-digit PIN. This indicates that knowledge-based authentication remains a prevalent fallback, with the 6-digit PIN being the most popular Knowledge-based Authentication scheme. Q2

restart or when the biometric recognition fails [1]. We investigate 165 users regarding their protection measures. As shown in Fig. 1(b), among them, 84.1% use PINs as the fallback option for device unlocking, where 76.5% use the 6-digit PINs and 7.6% use the 4-digit PINs. More details can be found in the Section V-C. Consistent with previous views [1], [3], [25], PINs still remain crucial for the security of mobile devices.

PIN-based authentication presents a fundamental security dilemma: while offering high availability, it suffers from inherent vulnerabilities to many attacks. Online guessing serves as a critical security benchmark, with research revealing severe predictability (over 50% coverage by top 5%–8% PINs [30]), low entropy (7.2 bits for 6-digit PINs [30]), and alarming attack success rates (6.5% within 10 guesses [1], 7% within 5 attempts [2]). Attackers leverage optimized guessing lists and leakage databases, achieving 2.9%–5.7% success rates within 10 guesses [3]. Compared to online guessing attacks, shoulder surfing poses an even greater physical security challenge. Video-based observation attacks report 10.8% single-attempt success rates (escalating to 26.5% with multiple observations) for 6-digit PINs [4], while real-world attackers demonstrate 1.9× higher effectiveness than video simulations [5].

AI-enhanced attacks dramatically amplify PIN security threats by exploiting subtle physical traces. In 2017, Abdelrahman et al. [7] demonstrated that thermal traces within 30 seconds during authentication enable 78% attack success rates. Video recordings of PIN entry in public further compromise security. Spatio-temporal analysis of typing dynamics allows attackers to guess 70% of characters within 10 attempts [9], drastically reducing the number of guesses. Moreover, smartphone sensor data analysis achieves 83.7% success rates within 20 tries in single-user scenarios [8], highlighting how AI transforms residual physical evidence into critical vulnerabilities.

B. The Real-World Investigation

Unauthorized access poses a potential threat to user privacy and financial security. Reinforcing the insider threat, the 2013 survey of Muslukhov et al. [31] showed that among the participants, 12% reported negative incidents of unauthorized data access (often by acquaintances), and 9% admitted to accessing others' phones without permission. Work in 2017 by Eiband et al. [32], collecting real-life stories of shoulder surfing from 174 participants, revealed widespread awareness (97%) and frequent occurrence (21%) of such incidents, with observers capturing login credentials like PINs or passwords in 9% of cases. Shifting focus to device access, a 2019 study by Marques et al. [33] examining 102 real stories of unauthorized smartphone access found perpetrators were overwhelmingly within the victim's inner circle, intimate partners, friends, or family members. Further exploring user behavior, a 2020 study by Casimiro et al. [34] with 150 participants identified PIN reuse across different contexts (including high-value and low-value accounts) as highly prevalent. Motivated primarily by ease of memorization (55%) and convenience (23%), this practice included the concerning example of 5% reusing sensitive banking PINs for low-security voicemail.

User understanding also varies significantly. The 2021 research [35] of Bailey et al. on Signal users showed 56% comprehended PIN purposes, employing longer, complex PINs with password managers; conversely, 44% lacked understanding, typically setting short, numeric-only PINs, and 28% reused their Signal PIN elsewhere. Concrete evidence of vulnerability emerged in 2023 when Roberts et al. [36] easily unlocked 26.8% of 228 auctioned ex-police phones, which included devices with no lock screen (1.5%), common PINs/patterns (4.8%), and even one with the PIN written on its back.

C. Relevant Works

Keystroke dynamic authentication utilizes the unique typing patterns recorded by the sensors (e.g., Pressure sensor and Accelerometer) embedded in smartphones to authenticate users. Such as the variations of time intervals of input rhythm and the strength preferences of the applied force. In recent years, keystroke dynamics-based authentication [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25] is developed to enhance the security of PIN-based authentication.

Table I summarizes the details of 16 existing studies in the literature. Specifically, nine studies [10], [11], [12], [13], [14], [15], [19] investigate 4-digit PINs, while five [16], [18], [21], [23], [24] focus on 6-digit PINs. Some works [11], [12], [17], [20], [22], [25] also involve PINs of lengths including 8, 9, 10, and 16 digits. To date, only the dataset from Teh et al. [22] remains accessible. It contains timing and touch data, without motion sensor information. 15 studies utilized datasets with no more than 100 users. In this paper, we present a public dataset collected from 97 users, which includes timing, touch,

TABLE I
RELATED WORKS ON KEYSTROKE DYNAMICS FOR PIN-BASED AUTHENTICATION. †

Authors	Year	Ref	Datasets	PIN	Se	Av	N_0	Signals			Classifier			Feature Engineering	Experiments Setting				Metrics		
								S_1	S_2	S_3	C_1	C_2	C_3		N_1	N_2	N_3	N_4	FRR	FAR	EER
Kim et al.	2024	[25]	Self-collected	9-digit	6	○	17	●	●	○	○	○	●	Automatic	∼600	2400	800/800	-	5.5%	8.0%	6.7%
El-Kenawy et al.	2022	[24]	Third-party	6-digit	-	○	-	●	●	●	○	○	●	Automatic	∼20	-	-	-	9.9%	2.0%	-
Choi et al.	2021	[23]	Self-collected	6-digit	1	○	13	●	●	●	●	●	○	Artificial	∼120	96	20/64	-	-	-	10.1%
Teh et al.	2020	[22]	Self-collected	4/16-digit	1	●	150	●	●	○	●	●	○	Artificial	∼20	-	-	-	-	-	8.7%
Lee et al.	2019	[21]	Self-collected	6-digit	1	○	22	●	●	●	●	○	○	Artificial	∼100	-	-	-	12.9%	1.0%	6.9%
Nareerat et al.	2019	[20]	Self-collected	10-digit	1	○	30	●	●	○	○	●	○	Artificial	∼10	270	-	-	-	-	10.0%
Al-Jarrah et al.	2019	[19]	Self-collected	4-digit	1	○	70	●	●	○	●	○	○	Artificial	∼51	20	20/138	-	-	-	8.3%
Lee et al.	2018	[18]	Self-collected	6-digit	1	○	22	●	●	●	●	○	○	Artificial	∼100	-	-	-	-	-	7.9%
Coakley et al.	2016	[17]	Self-collected	10-digit	1	○	52	●	●	●	●	○	○	Artificial	∼43	-	1/5	-	-	-	3.9%
Roh et al.	2016	[16]	Self-collected	6-digit	15	○	15	●	●	●	●	○	○	Artificial	∼225	-	-	-	-	-	6.9%
Deng et al.	2015	[15]	Third-party	4-digit	1	○	55	●	●	●	●	○	○	Artificial	∼150	165	165	-	-	-	2.8%
Grant Ho	2014	[14]	Self-collected	4-digit	1	○	55	●	●	●	●	●	○	Artificial	∼150	60	10	-	4.4%	5.3%	-
Mendizabal et al.	2014	[13]	Third-party	4-digit	1	○	80	●	●	●	●	○	○	Artificial	∼25	-	-	3/6/9	-	-	20.0%
Zheng et al.	2014	[12]	Self-collected	4/8-digit	1	○	80	●	●	●	●	○	○	Artificial	∼125	-	-	85	-	-	3.7%
Tasia et al.	2013	[11]	Self-collected	4-8-digit	5	○	100	●	○	○	●	●	○	Artificial	∼10	-	-	5	10.0%	9.8%	10.0%
Hwang et al.	2009	[10]	Self-collected	4-digit	1	○	25	●	○	○	●	○	○	Artificial	∼35	-	30/48	5	-	-	4.0%
This work	2026	-	Self-collected	6-digit	4	●	116	●	●	●	○	○	●	Automatic	∼200	∼38,000	∼13,000	∼13,000	1.9%	2.2%	2.4%

†: This table summarizes the specific details of 17 related works. "Se" is the number of sessions during data collection. "Av" indicates the availability of datasets. "No" is the number of participants in datasets. S_1 to S_3 represent time data, touch data, and sensor data, respectively. C_1 to C_3 represent the statistical models, machine learning models and deep learning models, respectively. N_1 indicates the number of samples of each participant. N_2 to N_4 represent the number of samples in training phase, testing phase and enrollment phase, respectively. • denotes "yes", ◦ denotes "no", and "-" signifies that the relevant studies do not explicitly provide the information.

and motion sensor data. Existing works [12], [13], [14], [15], [16], [17], [18] indicate that multi-modal data can capture richer behavioral biometric information, thereby effectively enhancing model performance. Among these, 11 studies [12], [13], [14], [15], [16], [17], [18], [21], [23], [24] employed multi-modal datasets. In terms of classifiers, 13 works [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [21], [22], [23] used statistical models or one-class classifiers such as euclidean distance, Manhattan distance, and one-class support vector machines. Six studies [11], [14], [20], [22], [23] applied machine learning models including random forest and k-nearest neighbors (KNN). With recent advances in deep learning, these techniques have been adopted in biometric recognition. Two studies [24], [25] utilized deep learning approaches, such as convolutional neural networks (CNN) [25] and bidirectional recurrent neural networks (BRNN) [24].

Dataset: Keystroke dynamics-based authentication is a promising method to further enhance the security of PIN-based authentication schemes for mobile devices. However, only one dataset [22] is available so far. To facilitate this research, we collect 18,935 samples from 96 users, covering 20 types of PINs under two activity states (walking and sitting). The dataset has been made publicly available with the aim of supporting future studies. More details can be found in Section V-D.

Security Evaluation: To the best of our knowledge, no prior study has evaluated the security of keystroke dynamics specifically for PINs. In 2024, Hamme et al. [37] proposed a novel evaluation framework capable of estimating the effective key space of behavioral biometrics, which they applied to assess the security of keystroke dynamics in free-text scenarios. This paper adopts this framework to evaluate the security of keystroke dynamics for PINs. More details can be found in Section V-J.

Conflict: As shown in Fig. 1(a) and (c), 94.56% surveyed users employ biometric-based authentication (e.g., facial, fingerprint, voiceprint and iris recognition), with 16.36% additionally enabling intelligent authentication (including trusted devices, geographic location, and physical activity recognition). Due to the widespread adoption of alternative solutions, the frequency of usage of PIN-based authentication cannot support collecting rich samples on the client side. More details can be found in Section V-C. Existing works [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25] often pay less attention to the difficulty of collecting sufficient data during the registration phase in real-world scenarios, which may increase deployment difficulty and decrease user experience.

III. THREAT MODEL

Inspired by real-world investigations and the growing threat of various attacks discussed in Section II-A, we consider a potentially stronger attacker similar to [25], who can obtain mobile devices and PINs of target users. The attacker could be an experienced thief or an insider like a friend or colleague. Once the attacker successfully compromises the devices, they

can gain unauthorized access to the downstream services. In our threat model, we reference and partially adopt assumptions from prior work [28], [29] concerning attacks and defenses. In particular, we make the following assumptions:

A1. The device is used by the owner in most cases but may be occasionally accessed by the attacker.
A2. The attacker can get PINs of victims through technical or non-technical approaches.
A3. The data collected from sensors is protected by secure hardware such as TrustZone [38], to prevent possible injection attacks or data leaks.
A4. The service provider is trusted. The authentication model is trained and managed by the service provider, who is responsible for defending against backdoor or poisoning attacks [39], [40].
A5. Side-channel attacks aiming to extract user behavioral data via device sensors fall outside the scope of this work, as established techniques exist to mitigate them through data obfuscation [41].

IV. Design

We now introduce StrokePIN, a new user-friendly authentication system designed for real-world devices like smartphones. We aim to achieve the following goals:

G1: Generalization. Since the model is trained on data unrelated to target users, it must have the ability to extract features from unknown users' data and verify their identity. Moreover, the key space for a 6-digit PIN is 10^6, making it impractical for the service provider to gather data covering all PINs. Therefore, StrokePIN should also be effective in classifying categories that are not seen during training.

G2: Lightweight Registration. StrokePIN must be user-friendly and allow lightweight registration. Specifically, enrollment should complete within seconds, require no more than three samples for each posture.

G3: Robustness. StrokePIN should be robust to changes in user behavior. For example, if a user enters their PIN while walking for the first time, their behavioral pattern may not match the templates in the template pool. StrokePIN needs to update the template pool after successful biometric authentication.

We propose multiple approaches and design a new framework to achieve the three goals. For **G1**, we devise two identical CNN models as feature extractors for two types of data to learn latent representations. Rather than training a specific model for each user, our goal is to enable the model to learn highly discriminative and generalizable features from raw data, equipping it to acquire features from unseen data. We leverage triplet loss to optimize latent representations, enhancing the model's feature learning and generalization. Based on the feature extractor, we design a feature fusion mechanism using a feedforward neural network to learn intrinsic relationships between the two latent representations and capture more subtle patterns, further improving performance. The generalization capability allows addressing **G2** and **G3** trivially, avoiding extensive templates. We propose a dynamic enrollment mechanism: when a user enters the PIN with an unseen posture, it automatically updates the template pool after successful authentication.

A. Overview

As shown in Fig. 2, StrokePIN includes a service provider who is responsible for training models, and users who access the services from the service provider. Unlike existing solutions [28], [29], the data used for model training is not related to users and is collected by the service provider. Although StrokePIN increases the cost of collecting data for the service provider, it maximizes the protection of user privacy. This design directly aligns with regulations like GDPR [42], which prohibit collecting/uploading user sensor data.

Data: StrokePIN mainly employs built-in sensors on smartphones, including touch sensors and motion sensors (gravity sensor, accelerometer and gyroscope), to capture multi-modality data during PIN entry. Touch sensors can capture user physiological and behavioral information, such as touch area, pressure, and timestamp. Gravity sensors, accelerometers, and gyroscopes collectively monitor device movement by detecting real-time changes in orientation and rotation.

Training Phase: The training phase of StrokePIN is conducted by a service provider. The high-level pipeline is illustrated in Fig. 2. It consists of three modules: the feature extraction module, the feature fusion module, and the classifier module. The feature extraction module consists of two sub-modules: one for extracting touch features and the other for extracting motion features. The last two modules are combined as a deep neural network model for feature fusion and identification. During the training phase, a service provider first builds a feature extraction module, which maps a pair of sensor data to two latent representation vectors. The feature extraction module is used as the base model for training subsequent modules. Second, the feature fusion module is trained to further extract a cross-modality representation vector with the output of the feature extraction module. The classifier module is then trained to determine whether two representation vectors from the feature fusion module belong to the same user.

Registration Phase: As shown in Fig. 2, StrokePIN supports two registration modes: (1) Explicit Registration, where behavioral patterns are captured during initial PIN setup; and (2) Implicit Registration, which is triggered when a login attempt with the correct PIN fails due to behavioral variations (e.g., posture changes like sitting during registration versus walking during login). In implicit registration, StrokePIN initiates a biometric authentication fallback (e.g., face or fingerprint). Upon successful verification, the new behavioral template is dynamically added to the template pool. This mechanism not only guarantees that all behavioral data originates from the legitimate user but also enhances the robustness of system. Critically, beyond leveraging biometric-based authentication to guarantee legitimacy, StrokePIN runs transparently, requiring no additional user effort.

Login Phase: During the login phase, StrokePIN performs two-factor authentication, verifying both the PIN correctness and consistency of behavioral patterns. Specifically, StrokePIN

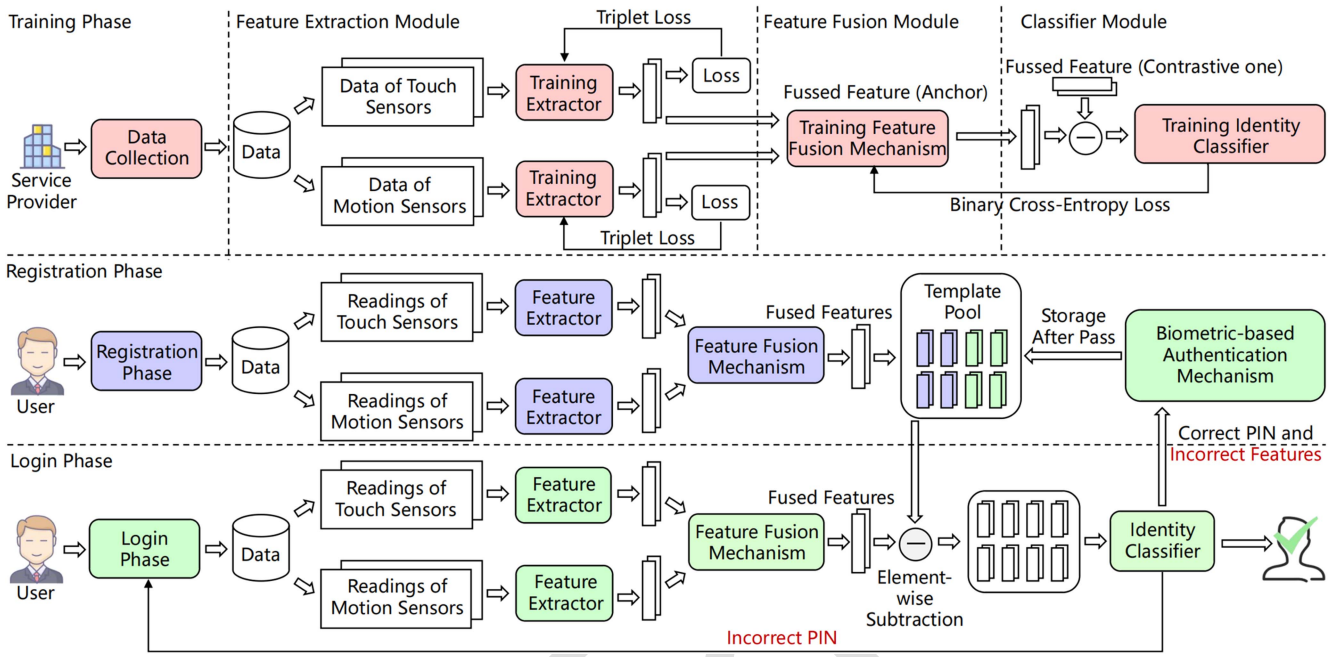

Fig. 2. The illustration of the module structure. StrokePIN comprises a service provider and users. The classifier is trained and maintained by the service provider. By decoupling model training and deployment at the data level, users can deploy the model without exposing their privacy-sensitive data. Meanwhile, StrokePIN can achieve efficient deployment, requiring no more than three samples for each posture.

leverages the well-trained model to detect potential adversaries who may have correct PINs but exhibit abnormal behavioral patterns. These are conducted on users' devices and do not upload any data to the service provider. Specifically, given the sensor readings of a new input, StrokePIN extracts its latent representation vector and compares it against all templates in the pool using identity classifier. If the classifier's confidence score falls below the threshold, StrokePIN considers the input abnormal and the device may be compromised. Then, a more stringent authentication method such as fingerprint recognition will be invoked.

B. Data Preprocessing

During a single PIN entry session, multiple touch events are triggered. Each touch event corresponds to a contact point characterized by eight attributes: timestamp t [3], X-coordinate x [4], Y-coordinate y [5], major diameter d [6], minor diameter n [7], orientation o [8], pressure level p [9], and contact area a [10]. Thus, a complete PIN entry yields a multivariate time-series represented as a two-dimensional matrix T with eight columns (corresponding to eight features),where each row encapsulates the sensor readings of an individual touch event.

$$T = \begin{bmatrix} t_1 & x_1 & y_1 & d_1 & n_1 & o_1 & p_1 & a_1 \\ t_2 & x_2 & y_2 & d_2 & n_2 & o_2 & p_2 & a_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_m & x_m & y_m & d_m & n_m & o_m & p_m & a_m \end{bmatrix}$$

For motion sensors, we utilize triaxial data from gravity sensors [11], accelerometers [12], and gyroscopes [13] to capture subtle behavioral characteristics during PIN entry. We chose them because they have been proven to be effective and are commonly applied to some related fields [43], [44]. Each data from sensors are represented as (x, y, z) vectors corresponding to their three orthogonal axes. For each PIN entry session, we organize the sensor data into multivariate time-series matrices structured as follows. Each row in matrix M represents a time-indexed data vector comprising nine raw channels: the triaxial readings (x, y, z) of three motion sensors.

$$M = \begin{bmatrix} Grax_1 & Gray_1 & Graz_1 & Accx_1 & \dots & Gyrz_1 \\ Grax_2 & Gray_2 & Graz_2 & Accx_2 & \dots & Gyrz_2 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ Grax_m & Gray_m & Graz_m & Accx_m & \dots & Gyrz_m \end{bmatrix}$$

Data Normalization: Before authentication, Min-Max normalization is applied to perform a linear transformation on the raw

[3] `System.nanoTime()`
[4] `event.getX()`
[5] `event.getY()`
[6] `event.getAxisValue(MotionEvent.AXIS_TOUCH_MAJOR)`
[7] `event.getAxisValue(MotionEvent.AXIS_TOUCH_MINOR)`
[8] `event.getAxisValue(MotionEvent.AXIS_ORIENTATION)`
[9] `event.getPressure()`
[10] `event.getSize()`

[11] `Sensor.TYPE_GRAVITY`
[12] `Sensor.TYPE_ACCELEROMETER`
[13] `Sensor.TYPE_GYROSCOPE`

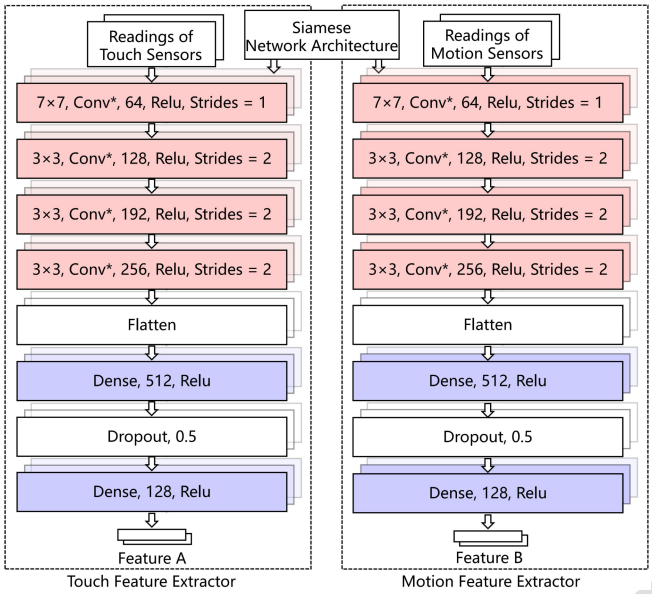

Fig. 3. Design details of the feature extraction module, which employs CNN and multilayer perceptron technologies. With Siamese Network framework, the feature extractors can extract discriminative features directly from limited samples of each class, laying the foundation for subsequent feature fusion mechanism.

data, as shown in (1). n denotes the total number of data points of all PIN entry sessions. x iss the feature value.

$$\overline{x_i} = \frac{x_i - \min_{1\le j\le n} x_j}{\max_{1\le j\le n} x_j - \min_{1\le j\le n} x_j} \tag{1}$$

All feature columns in matrices T and M, except the timestamp column in T, can be directly normalized using (1). For the timestamp column in T, where each unit is derived from `System.nanoTime()`, applying normalization directly would distort the temporal relationships between PIN entry events. To address this, we first baseline each PIN entry session by setting its initial timestamp to zero. Subsequent timestamps within the session are then set as offsets relative to this baseline. Following this session-relative transformation, 1 is applied to the adjusted timestamps.

C. Feature Extraction Module

The goal of feature extraction module is to convert the raw data from sensors to latent representation vectors, where the noise is eliminated. Rather than extracting user-specific characteristics, our focus is on deriving user-independent features. The feature extraction module clusters the representation vectors of the same user while pushing those of different users far away.

Design: At a high level, we primarily employ convolutional neural networks as main backbone to construct the Siamese Networks. As illustrated in Fig. 3, the feature extraction module comprises two carefully design dedicated sub-modules: one processing touch sensor data and the other handling motion sensor data. Each sub-module employs a Siamese Network to derive discriminative feature representations from raw sensor data. This design inherently optimizes the model to discriminate between genuine users and impostors. Specifically, representation vectors

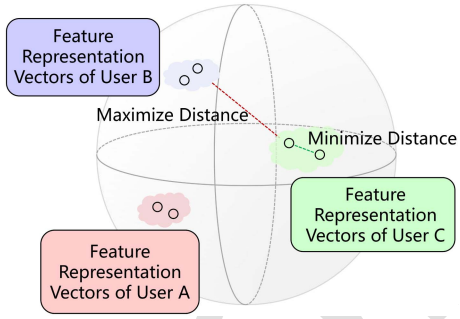

Fig. 4. The illustration of feature representation space. Each feature vector is mapped onto a unit sphere. Effective training pulls feature vectors of the same user closer while pushing different users' vectors apart, thus creating distinct effective regions per user.

from the same user exhibit significantly closer proximity in the latent space, while those from distinct users diverge markedly. The Siamese Networks can learn generalizable features even when only a handful of samples per class are available.

For each sub-module, we use four convolutional layers with an increasing number of filters. The first convolutional layer employs 64 filters with a kernel size of 7 and a stride of 1. Subsequent layers increase filter depth (128, 192, 256 filters) while reducing temporal resolution, using kernel sizes of 3 and strides of 2 for efficient hierarchical representation learning. Batch normalization is applied after each convolution to stabilize training and reduce the number of training epochs. Then, the output of the convolutional block is flattened and processed by a fully connected layer with 512 units. A dropout layer is then applied for regularization before projecting the features into a discriminative 128-dimensional embedding space via the final dense output layer.

Training Strategy: Let $\overline{T}$ denotes the matrix of normalized touch sensor readings, $\overline{M}$ represents the matrix of normalized motion sensor readings. The feature extraction process can be formalized as 2 and 3, where f_1 and f_2 denote the touch and motion feature extractors, F_1 and F_2 denote the representation vectors.

$$F_1 = f_1(\overline{T}) \tag{2}$$

$$F_2 = f_2(\overline{M}) \tag{3}$$

We employ the triplet loss [29] to train our feature extractor. This loss function can simultaneously learn the distances between both positive and negative sample pairs. It directly optimizes the relative relationships within the feature representation space, enabling the learning of discriminative features without requiring extensive class labels. Fig. 4 illustrates potential distributions of feature representation vectors for three distinct users within the overall feature space. The objective of the triplet loss is to ensure that embedding vectors representing samples from the same user are positioned in close proximity within the feature space, while vectors representing samples from different users are pushed far apart from each other. The triplet loss function is formulated as:

$$\mathcal{L}_{f_i} = \max\left(d(f_i(a), f_i(p)) - d(f_i(a), f_i(n)) + \alpha, 0\right) \tag{4}$$

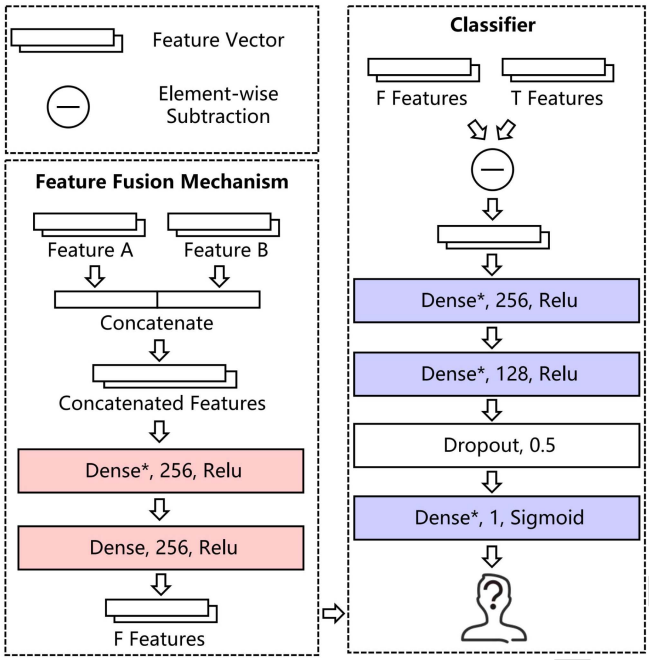

Fig. 5. The illustration of the feature fusion module and classifier module. With the input of two feature vectors extracting from two modalities, the purpose of the feature fusion module is to extract more discriminative features from multi-modal vectors. Then, the classifier will authenticate users with these features.

where $f_i(\cdot), i \in \{1, 2\}$ denotes the touch and motion feature extractors, a represents the anchor sample, p denotes the positive sample (belonging to the same user as a), n denotes the negative sample. The margin hyperparameter α enforces a minimum separation distance between positive and negative pairs, preventing the model from collapsing to trivial solutions. The function $d(\cdot, \cdot)$ computes the L2 distance of two feature representation vectors. With well-trained feature extractors, we can obtain the feature representation vectors of raw data.

D. Feature Fusion Module and Classifier Module

The goal of the feature fusion module is to learn more discriminative features from the representation vectors of different modalities, thereby enhancing StrokePIN performance. This module projects the dual-source feature vectors into an expanded joint latent space via nonlinear high-dimensional mapping, generating fused representations with greater discriminative power. Subsequently, the classifier is trained to determine the probability that any two distinct samples belong to the same user, using the element-wise subtraction of their fused representation vectors as input. The model architecture is shown in Fig. 5, where we design the feature fusion mechanism $\mathcal{F}$ for two types of feature vectors. Specifically, we leverage two fully-connected layers as the architecture of the feature fusion mechanism that fuses two vectors into a higher dimensional representations. Besides the feature fusion mechanism, we introduce a classifier $\mathcal{C}$ that first stacks two fully-connected layers with 256 and 128 hidden units, each followed by batch normalization and ReLU activation. A dropout layer (p = 0.5) is applied after the second layer, and the final fully-connected layer outputs a single logit with sigmoid activation to yield the probability $\hat{y}$ that the two samples originate from the same user. The feature fusion mechanism and classifier are combined as a deep neural network model, which is trained using binary cross-entropy loss. The loss function is formulated as follows:

$$\mathcal{L}(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \quad (5)$$

where the label $y \in \{0, 1\}$ equals 1 when the two input samples are from the same user, and 0 otherwise. It is worth noting that the parameters of the well-trained feature extractors need to be fixed when training the feature fusion mechanism and the classifier.

V. Evaluation

In this section, we conduct comprehensive experiments to evaluate the effectiveness of StrokePIN. Our experiments aim to answer the following questions:

- *RQ1:* How is StrokePIN's performance on unseen users?
- *RQ2:* How is StrokePIN's performance on unseen PINs?
- *RQ3:* How does StrokePIN perform on different PINs?
- *RQ4:* Compared to PIN-based authentication systems, how much additional entropy does StrokePIN provide?
- *RQ5:* How time affects strokePIN's performance?
- *RQ6:* Does feature fusion mechanism enhance StrokePIN's performance?
- *RQ7:* How is single modality performance?

A. Implementation

All the experiments are conducted on a server running Ubuntu 24.04.2 LTS, equipped with 16 GB of memory, an Intel Core i5-14400F CPU (10 cores / 16 threads) and NVIDIA GeForce RTX 4060. We use Tensorflow 2.12.0 to implement the neural networks. The feature extractors are trained in isolation, whereas the feature fusion model and classifier are updated simultaneously with the results obtained from feature extractors. We train the model using the triplet loss and binary cross-entropy loss. The former is defined by ourselves and the latter is provided by the sub-module `tensorflow.keras.losses`.

We employ the Grid Search [28] to find the optimal hyperparameter. Specifically, we set the optimizer to `Adam`, the learning rate to 1e-3, the batch size to 64, and the margin α to 0.01, 20 local training epochs.

We use five smartphones of the same brand to measure inference cost. The smartphones are `Huawei P30 pro`. Using the same brand can eliminate hardware variability and guarantee the reproducibility of experimental results.

B. Evaluation Metrics

To comprehensively evaluate StrokePIN, we use five different metrics in the evaluation: False Acceptance Rate (FAR), False Rejection Rate (FRR), Equal Error Rate (EER), True Positive Rate (TPR), and Area Under ROC Curve (AUC) [28]. These evaluation metrics are based on the number of correctly (TP) and incorrectly (FN) classified benign authentication events as

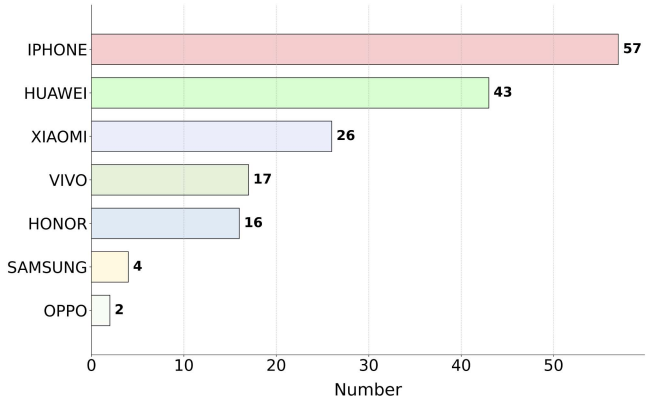

Fig. 6. This figure presents the distribution of mobile phones used.

well as the number of correctly detected (TN) and unrecognized (FP) attack attempts [29]. More details of them are as follows:

FAR: FAR represents the proportion of attack attempts incorrectly classified as benign events and is defined as

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{6}$$

FRR: FRR represents the proportion of benign events incorrectly rejected and is defined as

$$\mathrm{FRR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} \tag{7}$$

EER: EER is the error rate at the threshold where FAR and FRR are equal. It represents the trade-off between FAR and FRR across different thresholds.

TPR(Recall): TPR represents the proportion of benign events correctly identified and accepted and is defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{8}$$

ROC: ROC is the area under the ROC curve, which plots the TPR against the FPR at various classification thresholds. The AUC quantifies the overall ability of the system to discriminate between benign authentication events and attack attempts across all possible thresholds, where a higher AUC indicates better performance (1.0 represents perfect discrimination).

C. Lock-Screen Methods Survey

We investigate the screen unlocking methods of 165 users across three dimensions: biometric authentication, knowledge-based authentication, and behavioral identification. The questionnaire content is provided below.

Q1. Phone Brand and Model _____

Q2. Biometric-based Authentication methods in use
- ☐ Fingerprint Recognition
- ☐ Face Recognition
- ☐ Iris Recognition
- ☐ Voiceprint Recognition
- ☐ None

Q3. Knowledge-based Authentication methods in use
- ☐ 4-digit PIN
- ☐ 6-digit PIN
- ☐ Custom Numeric Password
- ☐ Alphanumeric Password (supports letters, numbers, and symbols)
- ☐ Grid Password
- ☐ None

Q4. Intelligent Identification
- ☐ Geographic Unlock
- ☐ Trusted Device Unlock
- ☐ Physical Activity Recognition Unlock
- ☐ None

This study surveyed 165 undergraduate and postgraduate students. As shown in Fig. 6, the mobile phone brands used include OPPO, Samsung Galaxy, HONOR, VIVO, XIAOMI, Huawei, and iPhone.

D. Data Collection

To support our study, we design an Android application that collects touch and sensor data as users perform PIN entries. Touch data sampling frequency is constrained by the sampling frequency of the smartphone hardware. In this paper, the touch data sampling frequency is approximately 120 Hz. For motion sensors, we adopted the lowest-power sampling rate, `SensorManager.SENSOR_DELAY_NORMAL`, which corresponds to a sampling frequency of 5 Hz.

We require participants to complete the PIN entry task. The number of PINs and the maximum number of correct inputs are optimized to equalize the completion time and the number of collected samples. This task is designed to be performed with the phone held in vertical orientation, consistent with users' typical screen-unlocking posture. Device rotation is disabled to maintain consistent interface orientation throughout the experiment. The home page displays the number of remaining PINs and the current progress of the experiment.

Task Design: The goal of the PINs entry task is to gather touch and motion data by simulating how users tend to unlock their phones. Participants are required to input PINs according to on-screen prompts. Prior to the task, participants are instructed to adopt their habitual phone-unlocking posture (e.g., single-handed or two-handed grip) mirroring real-world usage. Upon PIN entry, the callback initiates a background thread to log keystrokes and record sensor readings simultaneously.

To investigate the impact of different postures, we conduct experiments under two common postures (walking and sitting). Participants are required to maintain their preferred natural input pattern consistently throughout experiments. The PINs are selected based on usage patterns identified by Wang et al. [30], for example, one-digit repetition (e.g., 111111), keypad patterns (e.g., 147258), sequential sequences (e.g., 123456), and triple repetitions (e.g., 136136). Specifically, we strategically select 20 PINs, with the chosen PINs and their corresponding structural patterns presented in Table III. For each PIN, participants are required to perform seven correct entries. Wrong entries will be discarded as ~~invalid~~ data.

TABLE II
STATISTICAL DATA OF THE TWO COLLECTED DATASETS

Dataset	Sessions	Experiment Duration	Number of participants	Size	Size (cleaning)	Input Duration (mean)	Input Duration (sitting)	Input Duration (walking)
D_1	1	1 day	97	18,935	16,996	1.29 s	1.34 s	1.24 s
D_2	4	2 weeks	19	14,061	12,655	1.12 s	1.14 s	1.10 s

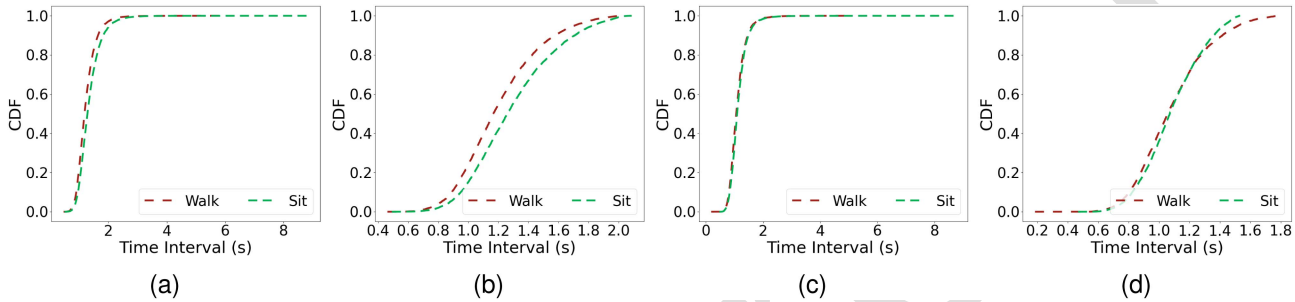

Fig. 7. During the experiment, the initial samples of participants may deviate substantially from their genuine input patterns because they are unfamiliar with the PINs. We mitigate this issue by cleaning the data to remove anomalous trailing observations, thereby enhancing the reliability of results. Fig. 7(a) and (b) respectively show the CDF plots of D_1 before and after data cleaning, with the horizontal axis representing the input time interval. In contrast, Fig. 7(c) and (d) respectively display the CDF plots of D_2 before and after data cleaning.

TABLE III
20 PINS USED IN EXPERIMENTS

PIN	Pattern †	PIN	Pattern
194012	YYYYMM	201412	YYYYMM
400101	YYMMDD	141231	YYMMDD
194011	YYYYMD	201499	YYYYMD
121940	MMYYYY	122914	MMYYYY
010140	MMDDYY	123114	MMDDYY
111940	MDYYYY	992914	MDYYYY
111111	One Digit Repeated	147258	Numpad Patterns
123456	Sequential Numbers	585520	Chinese Elements
121212	Couplets Repeated	112233	Double Sequential
136136	Triple Repeated	111222	Triple Sequential

†: To ensure representativeness, these PINs are selected following common real-world patterns, such as single-digit repetitions, sequential numbers, repeated couplets, and triple repetitions [30]. Following these commonly used patterns can improve the referential value, representativeness, and reliability of subsequent experimental results.

E. Ethical Consideration

We ensure that our research design, research objectives comply with local laws, regulations, and general ethical principles. Prior to the initiation of the experiment, we fully disclose the relevant risks and potential benefits to the participants, and clearly inform them that the collected data will be used for experimental research purposes. The collected datasets will undergo anonymization before storage. Both datasets have been made public on GitHub and IEEE DataPort, and all data have been anonymized to ensure that no association is established with the participants' personal information.

F. Dataset

Two datasets are collected in this study, with their detailed specifications summarized in Table II. One dataset D_1 is derived from a single session involving 97 users, yielding an initial sample size of 18,935. After data preprocessing, 16,996 valid samples are retained. The average input duration of this dataset is 1.29 seconds, with the average duration of PIN entry being 1.34 seconds under sitting condition and 1.24 seconds under walking condition. The other dataset D_2 is collected from 19 users who completed four sessions over a two-week period, with an interval of 3-4 days between consecutive sessions. The total number of raw samples is 14,061, and 12,655 valid samples are obtained after data cleaning. The average input duration of D_2 is 1.12 seconds, where average input durations under sitting and walking conditions are 1.14 seconds and 1.10 seconds, respectively.

As shown in Fig. 7(a) and (c), we find that 0.55% of the samples in D_1 have a time interval exceeding 3 seconds (up to 8.83 seconds), whereas only 0.25% of the samples in D_2 exhibit a time interval longer than 3 seconds (up to 8.73 seconds). This likely occurs because participants may be unfamiliar with PINs and type cautiously during initial attempts. Therefore, we remove all entries taking longer than 3 seconds in D_1 and D_2. Additionally, we exclude the first two samples of each PIN entered by each user to mitigate potential behavioral inconsistencies caused by unfamiliarity with PINs.

Fig. 7(b) and (b) present the Cumulative Distribution Function (CDF) of time intervals after data cleaning, the processed data exhibit a distribution within the 0-2 s window. Given that D_2 contains multi-session data, we select D_2 to conduct an analysis of RQ5. For the remaining six research questions (excluding RQ5), the larger-scale database D_1 is employed for investigation. For six research questions (excluding RQ5), all the samples of participants are divided into training, validation, and test sets by the ratio of 3:1:1. The users in training, validation, and test sets are mutually exclusive. Samples from the same user cannot appear in more than one set. For RQ5, D_2 is partitioned into training set and test set at a ratio of 7:3, with validation set

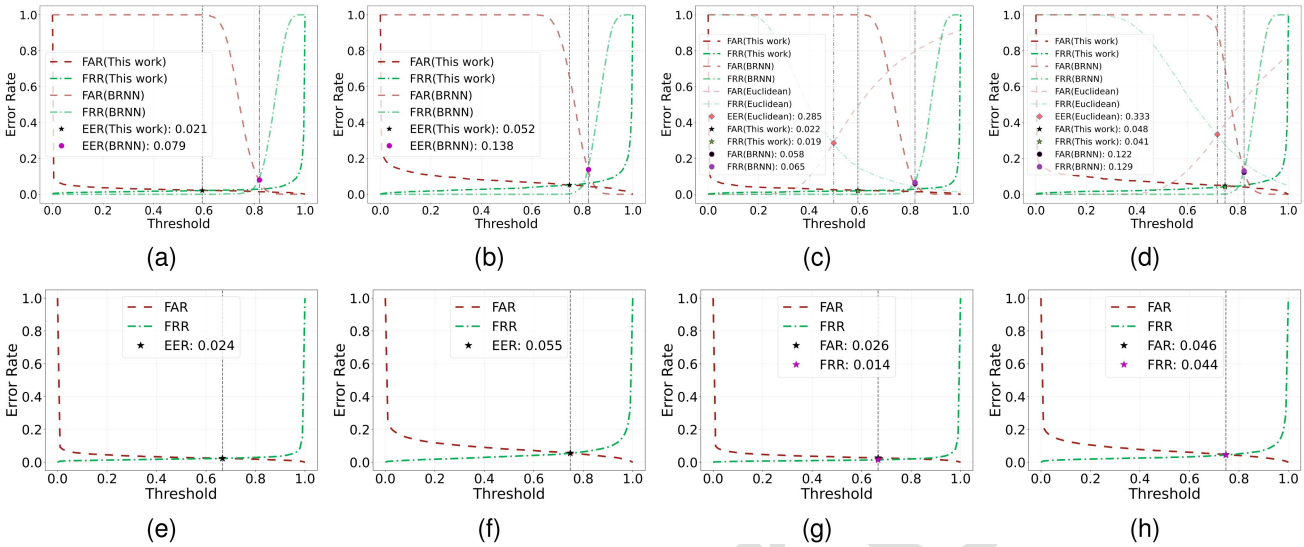

Fig. 8. Performance evaluation of StrokePIN on unseen users Fig. 8(a)–(d) and PINs Fig. 8(e)–(h). The FAR (red) and FRR (green) against varying thresholds (x-axis) demonstrate StrokePIN's robustness and reliability in real-world deployment without prior access to target user data or adversarial samples. Low error rates on optimal thresholds demonstrate StrokePIN's effectiveness for practical implementation.

accounting for 30% of training set. In addition, D_1 is split into training set and validation set at a 7:3 ratio. Dataset Partitioning follows the principle **P4** suggested by Georgiev et al. [45], as we cannot obtain samples of users or adversaries in advance.

G. RQ1: Unseen Users

In this section, we assess StrokePIN's performance on unseen users, comparing it with BRNN [24] and euclidean Distance [13], [15], [16], [17], [18] across various metrics. First, the optimal threshold is determined by calculating the FAR, FRR, and EER on the validation set. This threshold is subsequently applied to the test set to compute FAR and FRR. Finally, the ROC curve is plotted.

FAR, FRR and EER: The result of the first part is shown in Fig. 8(a)–(d). The x-axis represents different thresholds and the y-axis represents the error rate. The red line represents the FAR and the green line represents the FRR. We mark the optimal threshold (chosen by the validation set) with a vertical dashed line. The intersection points of the vertical dashed line and the curve are marked with asterisks. The result of the validation set for the two postures is visualized in Fig. 8(a) (sitting) and (b) (walking). The result of the test set for the two postures are visualized in Fig. 8(c) (sitting) (d) (walking). The validation set identifies the threshold of the EER, where FAR equals FRR (intersection of red/green curves). This threshold simultaneously minimizes both error rates. As Fig. 8(a) and (b) show, for StrokePIN, the EERs are 0.021 and 0.052, respectively. For BRNN, the EERs are 0.079 and 0.138, respectively. As EER represents the optimal threshold to balance declined benign attempts and accepted attack attempts, a lower EER of StrokePIN represents higher effectiveness for authentication.

For the results on the test set in Fig. 8(c) and (d), we report the FAR and FRR based on the optimal thresholds. For StrokePIN on the optimal threshold in Fig. 8(c), FAR is 0.022 and FRR is 0.019. For StrokePIN on the optimal threshold in Fig. 8(d), FAR is 0.048 and FRR is 0.041. For BRNN on the optimal threshold in Fig. 8(c), FAR is 0.058 and FRR is 0.065. For BRNN on the optimal threshold in Fig. 8(d), FAR is 0.122 and FRR is 0.129. For euclidean Distance, the EER is 0.285 in Fig. 8(c), and 0.285 in Fig. 8(d). In the authentication service, FAR is more important than FRR, because a false rejection only introduces an additional authentication step, while a false acceptance leads to malicious access and may cause severe security accidents [28]. The FAR of StrokePIN is over 2.63× lower than the other two baselines, which means StrokePIN provides a more reliable authentication service. The FRR of StrokePIN is over 2.95× lower than baselines, which means it can effectively reduce the probability of identifying the user as an attacker.

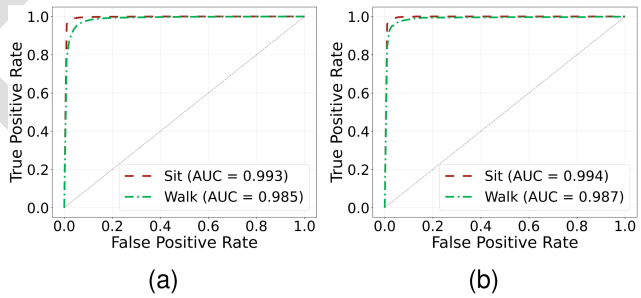

Fig. 9. (a) present the ROC curves of StrokePIN on unseen users. (b) present the ROC curves of StrokePIN on unseen PINs.

ROC and AUC: Fig. 9(a) shows the ROC scores on the test dataset. In Fig. 9(a), we also plot a naive baseline to represent the result of random guess (AUC is 0.50). For StrokePIN, the AUCs are 0.993 and 0.985. The AUC of StrokePIN on the sitting test set is higher than that on the walking test set, indicating that StrokePIN can more accurately detect malicious access when the phone is unlocked while the user remains seated.

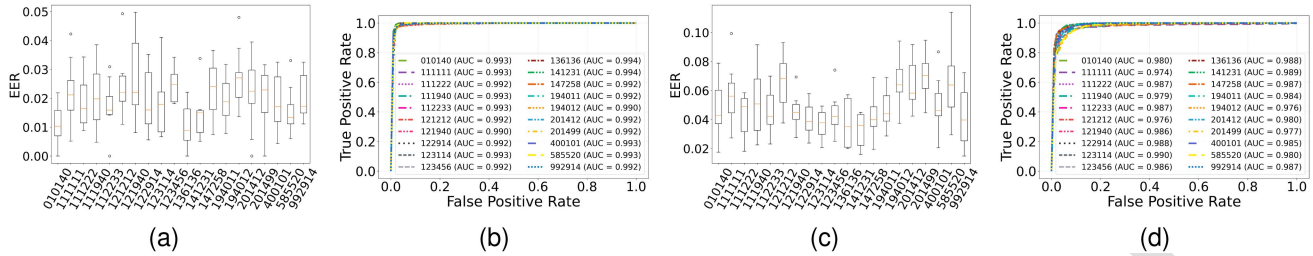

Fig. 10. (a) and (b) use box-and-whisker plots to depict the EER distributions of 20 PIN types on the sitting and walking test sets, respectively, while (b) and (d) report the ROC-based detection rates for malicious access attempts. Overall, StrokePIN achieves lower error rates on sitting condition than under the walking condition. Surprisingly, the trivial PIN "123456" attains the best performance, indicating that even simple PINs can have competitive performance. With users' preference for memorable and patterned PINs, This finding demonstrates the practical value of the StrokePIN.

Discussion: As a summary, StrokePIN can achieve over 2.63× lower FAR and 2.95× lower FRR, highlighting its well generalization capability. In addition, the results show that StrokePIN performs better on the sitting-posture test set compared to the walking-posture test set, suggesting that walking introduces additional motion-related sensor noise. We will elaborate on this point in Section ??. It is worth noting that each user in test sets has only five or fewer samples, demonstrating StrokePIN's ability to authenticate users reliably from limited samples. *This allows StrokePIN to be deployed with lightweight enrollment phase without sacrificing performance*.

TABLE IV
EERs OF DIFFERENT PINs ACROSS TWO POSTURES

PIN	sit	walk	PIN	sit	walk
010140	0.011	0.046	111111	0.022	0.056
111222	0.019	0.043	111940	0.021	0.052
112233	0.017	0.045	121212	0.024	0.066
121940	0.027	0.045	122914	0.019	0.040
123114	0.018	0.037	123456	0.025	0.044
136136	0.011	0.037	141231	0.014	0.033
147258	0.024	0.041	194011	0.019	0.046
194012	0.027	0.066	201412	0.022	0.063
201499	0.021	0.070	400101	0.019	0.051
585520	0.015	0.068	992914	0.021	0.042

H. RQ2: Unseen PINs

For 6-digit PINs, the size of key space is 10^6. Acquiring samples of all possible PINs may be expensive for service providers. In this section, our objective is to evaluate the generalization performance of StrokePIN on PINs that are not in the training set. The samples from 20 distinct PINs were partitioned at a 2:1 ratio. Specifically, samples from 13 PINs are allocated to the training set, and samples from the remaining 7 PINs are allocated to the test set.

FAR, FRR, and EER: The results are shown in Fig. 8(e)–(h). The x-axis represents different thresholds and the y-axis represents the error rate. Results for the validation set are shown in Fig. 8(e) (sitting posture) and Fig. 8(f) (walking posture). Similarly, test set results are presented in Fig. 8(g) (sitting) and Fig. 8(h) (walking).

As Fig. 8(e) and (f) show, the EERs are 0.024 and 0.055 respectively. In Fig. 8(g) and (h), with the optimal thresholds, StrokePIN obtains 0.026 FAR and 0.014 FRR on sitting-posture test set, 0.046 FAR and 0.044 FRR on walking-posture test set.

ROC and AUC: Fig. 9(b) shows the ROC curves to detect malicious access on the test dataset. As shown in the figure, the AUCs are 0.994 on sitting test set and 0.987 on walking test set. This indicates that StrokePIN can achieve better performance on the sitting test set than on the walking test set.

Discussion: StrokePIN does not exhibit performance degradation on unknown PINs. Conversely, StrokePIN presents robust performance. This demonstrates its generalization capability on unseen PINs, which can effectively help service providers reduce training costs, indicating the practicality of StrokePIN.

I. RQ3: Performance on Different PINs

In PIN-based authentication systems, PINs exhibit varying strength levels. Specifically, some patterns (e.g., sequential numbers like "123456" or triple repeats like "136136" [30]) are more frequently adopted by users. However, our analysis reveals that there is no correlation between the strength of PINs and the distinctiveness of behavioral biometric features. Even a simple PIN can have competitive performance.

Fig. 10(a) and (c) present box plots of EER distributions for 20 distinct PIN types on sitting and walking test sets, respectively. The Kruskal-Wallis test reveals statistically significant differences in EER distributions across PINs (sitting: $p = 0.00356 < 0.05$; walking: $p = 2.46 \times 10^{-6} < 0.05$). As quantified in Table IV, for sitting-posture test set, "123456" and "010140" achieve the lowest mean EER (0.011), while "121940" and "194012" get the highest (0.027). For walking posture test set, "141231" attains the best performance (mean EER=0.033), contrasting with "201499" (mean EER=0.070). We observe that simple PINs (e.g., "123456" and "136136") can yield competitive EERs. This pivotal finding confirms that StrokePIN can substantially mitigate the security-usability trade-off inherent in PIN-based authentication.

Fig. 10(b) and (d) show the ROC scores of StrokePIN on the test datasets. As shown in the figure, the AUCs are 0.990-0.994 on sitting test set and 0.974-0.989 on walking test set. The observed performance variation across PINs remains within acceptable bounds, demonstrating StrokePIN's robustness in handling diverse PINs.

Discussion: These results contradict the intuition that keystroke dynamics complexity scales with PIN sophistication.

TABLE V
ENTROPY ANALYSIS OF STROKEPIN ON SITTING-POSTURE TEST SETS †

Metrics	PIN									
	010140	111111	111222	111940	112233	121212	121940	122914	123114	123456
$\tilde{\lambda}_3$	4.77	4.91	4.64	4.79	4.58	4.92	4.53	4.38	4.64	4.74
$\tilde{\lambda}_6$	5.45	5.67	4.92	5.12	5.28	5.63	4.70	4.87	4.90	5.13
$\tilde{\lambda}_{10}$	6.18	6.40	5.62	5.71	6.02	6.37	5.12	5.60	5.53	5.79
$\tilde{G}_{0.2}$	4.39	3.86	5.40	5.27	4.39	4.47	5.36	4.50	5.23	5.09
	136136	141231	147258	194011	194012	201412	201499	400101	585520	992914
$\tilde{\lambda}_3$	4.26	4.69	4.54	4.94	4.48	4.17	4.34	4.94	4.71	4.44
$\tilde{\lambda}_6$	4.71	5.56	4.86	5.59	4.98	4.88	4.98	5.79	4.87	5.00
$\tilde{\lambda}_{10}$	5.35	6.30	5.46	6.33	5.69	5.62	5.72	6.53	5.45	5.74
$\tilde{G}_{0.2}$	4.15	3.50	4.89	4.44	4.37	4.08	4.01	3.94	5.74	4.16

†: For each PIN, the attacker exploits the held database to select the most advantageous feature-vector sequence and launch an attack. The table reports the security evaluation results for 20 PINs obtained on the sitting-posture test set.

Notably, simple PINs can achieve surprising performance. This finding is practically advantageous since users inherently prefer creating simple, patterned, and memorable PINs [46], thereby enhancing the real-world viability of keystroke dynamics authentication.

J. RQ4: Entropy Analysis

To assess the security strength of authentication methods (e.g., passwords [47] and PINs [30]), entropy is commonly used as a measure of the underlying randomness's uncertainty [48]. In the context of biometric systems, the classic entropy model overlooks the inherent non-uniformity of biometric feature distributions and the nature of fuzzy matching. To address these limitations, hamme et al. [37] proposed a novel evaluation framework capable of estimating the effective key space of biometrics and providing metrics that account for non-uniformity within the latent space. We consider this evaluation framework in assessing the security of StrokePIN.

β-success-rate, α-work factor, and α-guesswork are partial guessing metrics used in this paper.

β-success-rate [49] is the fraction of accounts that can be broken by an adversary limited to β guesses:

$$\lambda_\beta(\chi) = \sum_{i=1}^{\beta} p_i \tag{9}$$

α-work-factor [50] defines the number of guesses required to achieve a desired success rate α:

$$\mu_\alpha(\chi) - \min\left\{ j | \sum_{i=1}^{j} p_i \geq \alpha \right\} \tag{10}$$

α-guesswork is the expected number of guesses to break a fraction α accounts:

$$G_\alpha(\chi) = (1 - \lambda_{\mu_\alpha})\mu_\alpha + \sum_{i=1}^{\mu_\alpha} p_i \cdot i \tag{11}$$

The quantization of guessing metrics are as follows [51]:

$$\tilde{\lambda}_\beta(\chi) = \log\left(\frac{\beta}{\lambda_\beta(\chi)}\right) \tag{12}$$

$$\tilde{G}_\alpha(\chi) = \log\left(\frac{2G_\alpha(\chi)}{\lambda_{\mu_\alpha}} - 1\right) + \log\left(\frac{1}{2 - \lambda_{\mu_\alpha}}\right) \tag{13}$$

In this paper, we exclusively consider the wolf attack [37], as the master key attack necessitates adversaries possessing more powerful capabilities, such as compromising trusted zones. In wolf attacks, adversaries aim to maximize effectiveness by extracting optimal biometric templates from target databases instead of generating artificial samples. The guessing sequence $p_1, p_2, \ldots, p_k$ is computed with fuzzy matching technology.

Users are split in a 2:1 ratio, with the attacker holding the first two-thirds of samples to launch attacks, and the remaining one-third serving as targets. Tables V and VI present StrokePIN's entropy values for all PINs under two postures. The results show significant posture-dependent variations: $\tilde{\lambda}_3$ ranges from 4.17–4.94 (sitting) to 3.56–4.81 (walking), $\tilde{\lambda}_6$ from 4.70–5.79 (sitting) to 3.88–5.03 (walking), and $\tilde{\lambda}_{10}$ from 5.12–6.53 (sitting) to 4.19–5.76 (walking). For $\tilde{G}_{\aleph}$, $\tilde{G}_{0.2}$ spans 3.50–5.74 (sitting) versus 2.39–5.62 (walking). Compared to $\tilde{\lambda}_3$, $\tilde{\lambda}_6$, and $\tilde{\lambda}_{10}$, $\tilde{G}_{0.5}$ exhibits greater inter-PIN variability, likely due to limited samples and resulting distribution differences. However, the results in [37] imply that the results from limited sample sizes also hold reference value. Thus, despite the limited data, the conclusions drawn here remain informative.

Table VII shows the average entropy of several authentication schemes. Rows 1–3 correspond to prior work on keystroke dynamics [37] of text and conventional 4- and 6-digit PINs [30], while rows 4–5 present the results of this paper. As the goal of StrokePIN is to further enhance the security of PIN-based authentication, the key concern is the magnitude of the security it delivers. Specifically, when the number of online dictionary guesses allowed by the adversary is limited to 3, 6, and 10, the keystroke dynamics of PINs can provide security levels of 4.62 bits, 5.14 bits, and 5.83 bits in a seated posture, and 4.03 bits, 4.35 bits, and 4.72 bits in a walking state. For $\tilde{G}_{0.2}$, the keystroke

TABLE VI
ENTROPY ANALYSIS OF STROKEPIN ON WALKING-POSTURE TEST SETS †

Metrics	PIN									
	010140	**111111**	111222	111940	112233	121212	121940	122914	123114	123456
$\tilde{\lambda}_3$	3.67	4.10	4.25	3.91	4.11	4.22	4.20	4.81	4.01	3.57
$\tilde{\lambda}_6$	4.07	4.44	4.57	4.33	4.36	4.56	4.48	4.99	4.40	3.92
$\tilde{\lambda}_{10}$	4.40	5.01	5.11	4.77	4.61	5.07	4.64	5.42	4.60	4.33
$\tilde{G}_{0.2}$	2.74	3.61	4.29	3.24	3.96	4.31	4.31	5.62	3.74	2.39
	136136	141231	147258	194011	194012	201412	201499	400101	585520	992914
$\tilde{\lambda}_3$	4.20	3.59	4.69	4.49	3.56	3.64	3.93	3.84	3.83	3.88
$\tilde{\lambda}_6$	4.54	4.05	5.03	4.64	3.88	3.99	4.30	4.18	4.03	4.28
$\tilde{\lambda}_{10}$	4.94	4.41	5.76	4.88	4.19	4.26	4.58	4.59	4.26	4.52
$\tilde{G}_{0.2}$	4.30	2.39	4.93	4.71	2.40	2.57	3.26	3.09	3.25	3.26

†: The table reports the security evaluation results for 20 PINs obtained on the walking-posture test set. Compared with its performance on the sitting-posture test set, StrokePIN reports slightly lower accuracy on the walking-posture test set, likely because the motion artifacts introduced while walking mask the subtle yet discriminative motion signals transmitted during PIN entry.

TABLE VII
ENTROPY ANALYSIS OF DIFFERENT AUTHENTICATION SCHEMES

Source \ Metrics	Numbers	$\tilde{\lambda}_3$	$\tilde{\lambda}_6$	$\tilde{\lambda}_{10}$	$\tilde{G}_{0.2}$
Keystrokes [37]	10000	4.69	4.84	5.02	5.28
4-digit [30]	NA	4.81	5.27	/	8.41
6-digit [30]	NA	4.35	5.19	/	13.21
StrokePIN (sit)	97	4.62	5.14	5.83	4.56
StrokePIN (walk)	97	4.03	4.35	4.72	3.40

dynamics of PINs provide security levels of 4.56 bits and 3.40 bits under the two behavioral states.

Discussion: For $\tilde{G}_{0.2}$, the security provided by StrokePIN is lower than that of the 6-digit PIN. This result indicates that the overlap degree of the behavioral feature distributions relied on by StrokePIN is higher compared with that of the 6-digit PIN. However, it is noted that the security evaluation of StrokePIN presented in Table VII of this paper is based on a key premise: the adversary has obtained the user's PIN code. As a two-factor authentication mechanism integrating PIN code authentication and behavioral pattern authentication, the theoretical security upper bound of StrokePIN, assuming the two factors are independent, is the product of the security provided by each factor. Nevertheless, due to the limited number of PINs corresponding to the behavioral pattern data collected in this study, we cannot accurately quantify the security upper bound of StrokePIN in the more practical scenario where the adversary neither knows the user's PIN nor masters their behavioral patterns. All the security evaluation results of StrokePIN shown in Table VII are obtained under the assumption that "the adversary has knowledge of the user's PIN". The results demonstrate that keystroke dynamics can help enhance the security of PINs, which highlights StrokePIN's potential for broadening the security boundaries of traditional PIN-based authentication systems.

K. RQ5: Effect of Time

Existing research [52], [53] indicates that behavioral biometrics may drift over time. Data collected within a short period may not well capture all of the user's behavioral patterns. Existing studies [52], [53] present that even models trained on cross-session data may face performance degradation when tested on samples from different sessions. Therefore, investigating the impact of temporal factors on the performance of StrokePIN holds significant academic and practical value.

The objective of this section is to investigate the impact of temporal factors on the performance of StrokePIN. To this end, we collect data across four experimental sessions from 19 participants over a two-week period, with an interval of 3-4 days between consecutive sessions. The experimental results are illustrated in Fig. 11(a) to (h). Specifically, for Fig. 11(b) and (d), we adopt the data from the first session as the baseline template to evaluate the performance of StrokePIN on the data from the subsequent three sessions respectively. For Fig. 11(f) and (h), the template library is dynamically updated as the experimental sessions progress, where the template corresponding to each test session is derived from the data of its immediately preceding session.

In the validation set corresponding to the sitting posture scenario, as illustrated in Fig. 11(a) and (e), the corresponding Equal Error Rates (EERs) are 0.053 and 0.060, respectively. Based on the thresholds corresponding to the above EERs, the False Acceptance Rates (FARs) and False Rejection Rates (FRRs) (see Fig. 11(b)) on the test sets of the 2nd, 3 rd and 4th sessions are as follows: 0.143 and 0.029, 0.134 and 0.025, 0.131 and 0.050, in sequence. Similarly, the FARs and FRRs (see Fig. 11(f)) on the test sets of the 2nd, 3 rd and 4th sessions are 0.130 and 0.023, 0.136 and 0.030, 0.117 and 0.032, respectively. In the validation set corresponding to the walking scenario, as illustrated in Fig. 11(c) and (g), the EERs are 0.081 and 0.091, respectively. With the corresponding thresholds, the FARs and FRRs (see Fig. 11(b)) on the test sets of the 2nd, 3 rd and 4th sessions are listed in order as: 0.130 and 0.003, 0.123 and 0.014, 0.125 and 0.017. In contrast, as illustrated in Fig. 11(h), the FARs and FRRs on the above-mentioned three session test sets are 0.083 and 0.004, 0.085 and 0.017, 0.086 and 0.031, respectively.

Discussion: Compared to the results on the within-session test set (see Section V-G), higher error rates are reported on the cross-session test set (see Fig. 11(a) to (h)), which demonstrates that user behavior patterns vary across different

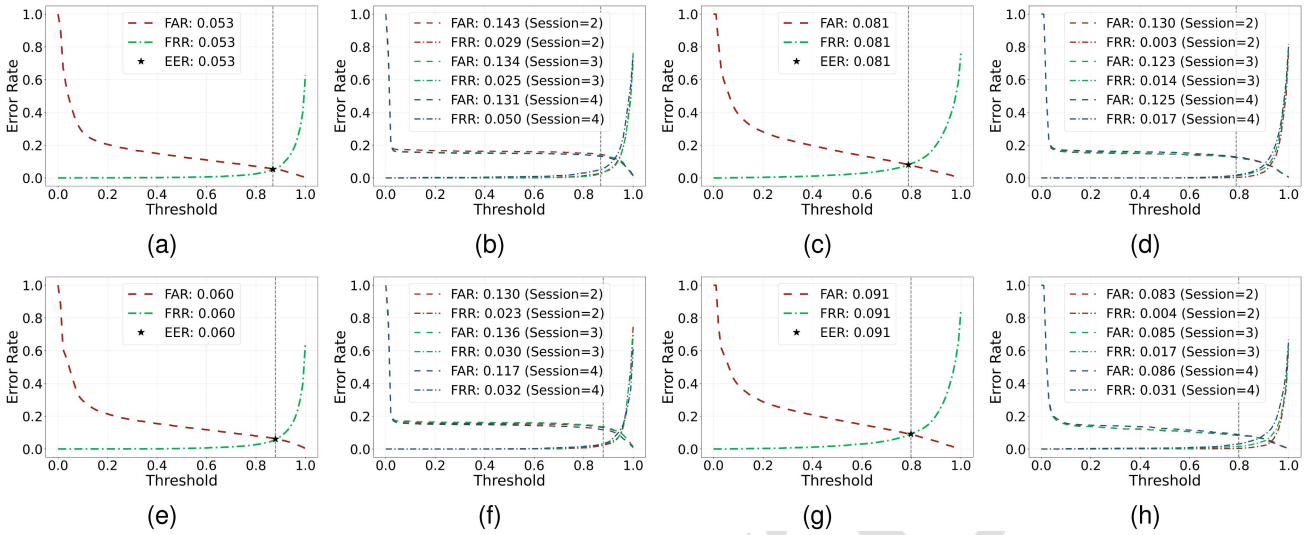

Fig. 11. (a)–(h) illustrate the impact of temporal factors on users' behavioral patterns. In (b)–(d), the data from the first session is set as the baseline template. In (f) and (h), the samples from the session immediately preceding the test session are adopted as the template samples. The results indicate that users' behavioral patterns shift over time, yet StrokePIN can mitigate the adverse effects caused by such behavioral drifts via dynamic template library updates.

time periods. Compared with the sitting condition, StrokePIN yields higher error rates on the validation set corresponding to the walking condition, consistent with the conclusions presented in Sections V-G and V-H. This is because both the training and validation sets include data from D_1 and D_2, and D_1 contains more users, which may exert a greater influence on the results. However, on the test set, StrokePIN performs better under the walking condition. Since the test set contains only data from D_2, this indicates that user behavior patterns during walking are more stable across different time periods, whereas patterns during sitting exhibit greater variability. This may be attributed to the fact that users adopt a wider range of postures while seated.

In Fig. 11(b), (d), (f), and (h), as the sessions increases, FAR shows a downward trend, while FRR rises. EER remains relatively stable with a slight upward drift, most notably in Fig. 11(h). Compared with the error rates shown in Fig. 11(b) and (d), lower error rates are reported on the dynamically updated template test set (see Fig. 11(f) and (h)), indicating that behavior patterns are more similar between consecutive sessions and that timely template updating helps improve the performance of StrokePIN. In summary, user behavior patterns shift over time, and updating the template library in a timely manner can mitigate the associated negative effects.

L. RQ6: Effect of Feature Fusion Mechanism

The goal of the feature fusion mechanism is to learn deeper features from different modalities. To assess its effectiveness, we benchmark the model's performance against StrokePIN by ablating the feature fusion mechanism. The results are presented in Fig. 12(a)–(d). The x-axis represents different thresholds and the y-axis represents the error rate. The red line represents the FAR and the green line represents the FRR. The semi-transparent line represents the performance of the model without feature fusion. Intersection points between the vertical dashed line and the curves are marked with asterisks for the baseline model and pentagons for the ablated model.

As shown in Fig. 12(a) and (c), the baseline model consistently outperforms the ablated variant across both sitting (EER: 0.027 vs. 0.029) and walking (EER: 0.050 vs. 0.064) postures. At optimal thresholds (Fig. 12(b) and (d)), the baseline achieves superior operational metrics. For sitting posture, it reduces FAR by 11.5% (0.023 vs. 0.026) and FRR by 37.9% (0.018 vs. 0.029). For walking posture, it decreases FAR by 28.2% (0.051 vs. 0.071) and FRR by 9.4% (0.048 vs. 0.053). Shapiro-Wilk tests [54] on 10-trial EER differences confirm their normality (sitting: $p = 0.62680$; walking: $p = 0.80849$). Subsequent paired one-sided t-tests demonstrate statistically significant EER improvements for the baseline model (sitting: $p = 0.00018$; walking: $p = 0.00091$), validating the effectiveness of the feature fusion mechanism.

Discussion: The efficacy of the feature fusion mechanism demonstrates that extracting cross-modal features delivers significant value, enabling further reductions in authentication error rates and enhancing model robustness. Unlike prior works, the end-to-end, fully-automated, and self-optimizing feature engineering pipeline in StrokePIN eliminates crafted feature engineering while achieving outstanding performance.

M. RQ7: Effective of Single Modality

In this section, to investigate why StrokePIN performs better on the sitting-posture data set than on the walking-posture data set, we conduct an ablation study to evaluate the contribution of each modality under both behavioral postures. Fig. 12(e) and (f) present touch modality performance on validation and test sets. The x-axis denotes threshold values, while the y-axis indicates error rate. Solid and semi-transparent lines represent performance on sitting-posture and walking-posture datasets, respectively. As shown in Fig. 12(f), StrokePIN achieves lower

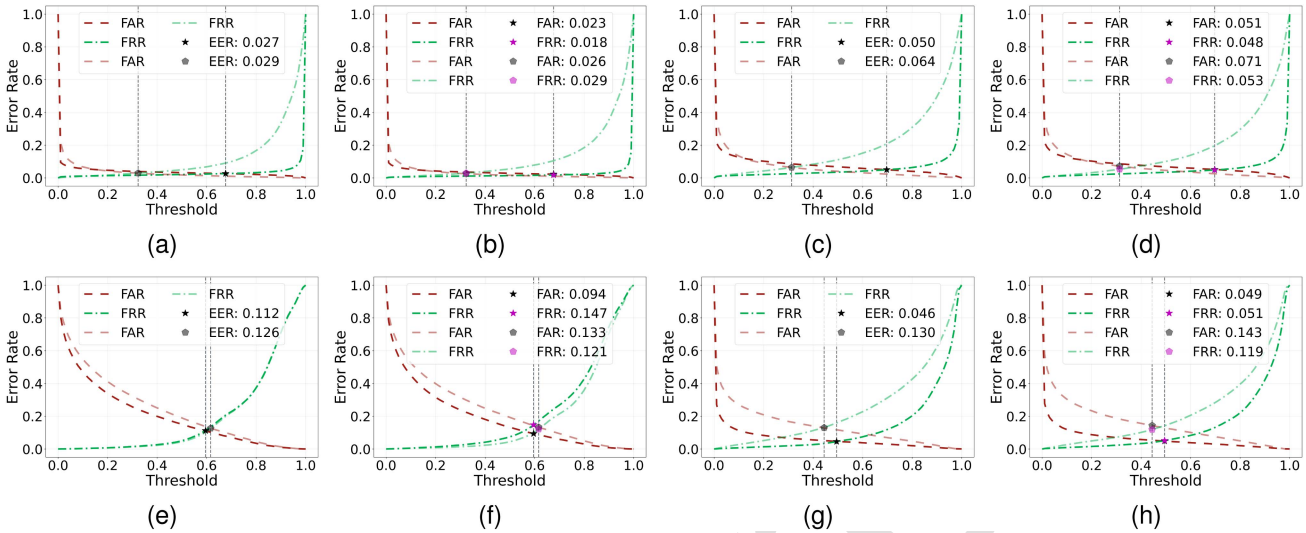

Fig. 12. (a)–(d) show the performance of the feature fusion mechanism. (e)–(h) show the performance of two different modalities. As depicted in (b) and (d), the system demonstrated significant improvements in both error rates.For sitting posture (b), FAR decreased by 11.5% (from 0.026 to 0.023) and FRR decreased by 37.9% (from 0.029 to 0.018).For walking posture (d), FAR decreased by 28.2% (from 0.071 to 0.051) and FRR decreased by 9.4% (from 0.053 to 0.048), demonstrating the effectiveness of the feature fusion mechanism. (e) and (f) present touch modality performance on validation and test sets. (g) and (h) illustrate motion modality performance on validation and test sets. StrokePIN's superior performance in sitting posture is primarily due to its motion modality. keystrokes and subtle wrist movements provide highly discriminative, user-specific features. In contrast, gait data during walking offers less distinctive characteristics and can introduce noise, reducing accuracy.

EER on the sitting-posture validation set than on the walking-posture validation set (0.112 vs. 0.126). However, this advantage reverses on the test set at the optimal thresholds. StrokePIN does not exhibit significant differences in the touch modality across the two postures. Fig. 12(g) and (h) illustrate the performance of StrokePIN on the motion modality. StrokePIN consistently outperforms on sitting-posture data sets. Specifically, on validation sets, EER decreases by 64.6% (0.046 vs. 0.130). At optimal thresholds, FAR reduces by 65.7% (0.049 vs. 0.143) and FRR by 57.1% (0.051 vs. 0.119). Consistently, StrokePIN exhibits superior performance in the motion modality.

Discussion: In summary, StrokePIN's performance advantage in sitting-posture primarily stems from its motion modality. Motion sensor data generated by keystrokes and subtle wrist rotations during PIN entry when sitting provides highly discriminative features. Although gait information during walking also offers valuable characteristics, it appears less distinctive than user-specific input patterns and may introduce noise that reduces authentication accuracy.

VI. Limitations

Like other empirical studies [25], [28], [55], this research also has some limitations that should be acknowledged when reporting the findings and generalizing the conclusions. These limitations primarily stem from sample characteristics, data collection conditions, as well as the influence of time, health status, and emotional states.

- *Sample characteristics:* All 97 participants in this study are students aged between 17 and 24, which may limit the generalizability of the findings to other demographic groups such as middle-aged and elderly individuals.
- *Data collection conditions:* All data were collected in a controlled environment. In real-world scenarios, however, user behavior states and intrinsic behavioral variations are considerably more diverse. Therefore, a key direction for future research is to collect data in real-world scenarios and design more effective authentication model.
- *Postures:* In our experiments, we found that some users changed their unlocking postures under different behavioral states. For example, they switched from two-handed to one-handed operation. In real-world scenarios, users' unlocking postures are even more diverse. Such diverse unlocking postures may exert a negative impact on model performance. Collecting relevant data and investigating diverse unlocking postures will be a key focus of future research.
- *Health status and emotional states:* User's health or emotional states may influence their input behavior [28]. How such factors affect model performance remains an problem for further study.

Overall, future research could perform more fine-grained annotation on the data, including information such as the surrounding environment and user postures. This would enable feature extractor to achieve more precise modeling of user behavior across different application scenarios.

VII. Conclusion

In this paper, we propose StrokePIN, an enhanced PIN-based authentication system leveraging keystroke dynamics to verify both the correctness of PINs and the legitimacy of input patterns. Additionally, we provide two new multi-modal datasets of 29,651 keystroke dynamics collected during PIN entry. To

the best of our knowledge, they are the only publicly available multi-modal datasets of their kind. The experimental results demonstrate that StrokePIN outperforms its counterparts, achieving state-of-the-art performance. Furthermore, StrokePIN shows robust performance on both unseen users and unseen PINs, which validates its generalization capability well. Furthermore, StrokePIN's capability to dynamically update the template library effectively mitigates the negative impact caused by shifts in behavioral patterns over time. Security analysis demonstrates that StrokePIN can effectively enhance the security of traditional PIN-based authentication schemes. This work provides important insights into the security of keystroke dynamics of PINs, which would further advance research in this field.

REFERENCES

[1] P. Markert et al., "This pin can be easily guessed: Analyzing the security of smartphone unlock PINs," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 286–303.

[2] D. V. Bailey et al., ""Someone definitely used 0000": Strategies, performance, and user perception of novice smartphone-unlock pin-guessers," in *Proc. Eur. Symp. Usable Secur.*, 2023, pp. 158–174.

[3] E. Korkes et al., ""I'm going to try her birthday": Investigating how friends guess each other's smartphone unlock pins in the lab," in *Proc. Eur. Symp. Usable Secur.*, 2024, pp. 220–234.

[4] A. J. Aviv et al., "Towards baselines for shoulder surfing on mobile authentication," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf.*, 2017, pp. 486–498.

[5] A. J. Aviv, F. Wolf, and R. Kuber, "Comparing video based shoulder surfing with live simulation," in *Proc. 34th Annu. Comput. Secur. Appl. Conf.*, 2018, pp. 453–466.

[6] M. Cardaioli et al., "Your PIN sounds good! on the feasibility of PIN inference through audio leakage," 2019, *arXiv:1905.08742*.

[7] Y. Abdelrahman et al., "Stay cool! understanding thermal attacks on mobile-based user authentication," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2017, pp. 3751–3763.

[8] D. Berend, S. Bhasin, and B. Jungk, "There goes your pin: Exploiting smartphone sensor fusion under single and cross user setting," in *Proc. 13th Int. Conf. Availability, Rel. Secur.*, 2018, pp. 1–10.

[9] D. Shukla and V. V. Phoha, "Stealing passwords by observing hands movement," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 12, pp. 3086–3101, Dec. 2019.

[10] S. Hwang, S. Cho, and S. Park, "Keystroke dynamics-based authentication for mobile devices," *Comput. Secur.*, vol. 28, no. 1/2, pp. 85–93, 2009.

[11] C. Tasia et al., "Two novel biometric features in keystroke dynamics authentication systems for touch screen devices," *Secur. Commun. Netw.*, vol. 7, no. 4, pp. 750–758, 2014.

[12] N. Zheng et al., "You are how you touch: Ustrengther verification on strengthenmartphones via tapping behaviors," in *Proc. IEEE 22nd Int. Conf. Netw. Protoc.*, 2014, pp. 221–232.

[13] I. de Mendizabal-Vázquez et al., "Supervised classification methods applied to keystroke dynamics through mobile devices," in *Proc. Int. Carnahan Conf. Secur. Technol.*, 2014, pp. 1–6.

[14] G. Ho, "TapDynamics: Strengthening user authentication on mobile phones with keystroke dynamics," *Technicalreport, Stanford Univ.*, vol. 73, pp. 74–78, 2014.

[15] Y. Deng and Y. Zhong, "Keystroke dynamics advances for mobile devices using deep neural network, 2015, pp. 59–70.

[16] J. Roh, Sung-Hun, and S. Kim, "Keystroke dynamics for authentication in smartphone," in *Proc. Int. Conf. Inf. Commun. Technol. Convergence*, 2016, pp. 1155–1159.

[17] M. J. Coakley, J. V. Monaco, and C. C. Tappert, "Keystroke biometric studies with short numeric input on smartphones," in *Proc. IEEE 8th Int. Conf. Biometrics Theory, Appl. Syst.*, 2016, pp. 1–6.

[18] H. Lee et al., "Understanding keystroke dynamics for smartphone users authentication and keystroke dynamics on smartphones built-in motion sensors," *Secur. Commun. Netw.*, vol. 2018, pp. 1–10, 2018.

[19] M. M. Al-Jarrah, G. M. Khalaf, and S. Amin, "Pin authentication using multi-model anomaly detection in keystroke dynamics," in *Proc. IEEE 2nd Int. Conf. Signal Process. Inf. Secur.*, 2019, pp. 1–4.

[20] N. Benjapatanamongkol and P. Bhattarakosol, "A preliminary study of finger area and keystroke dynamics using numeric keypad with random numbers on android phones," in *Proc. 23rd Int. Comput. Sci. Eng. Conf.*, 2019, pp. 30–34.

[21] H. Lee et al., "A parameterized model to select discriminating features on keystroke dynamics authentication on smartphones," *Pervasive Mobile Comput.*, vol. 54, pp. 45–57, 2019.

[22] P. S. Teh et al., "Strengthen user authentication on mobile devices by using user's touch dynamics pattern," *Ambient Intell. Humanized Comput.*, vol. 11, pp. 4019–4039, 2020.

[23] M. Choi et al., "Keystroke dynamics-based authentication using unique keypad," *Sensors*, vol. 21, no. 6, pp. 2242–2260, 2021.

[24] E.-S. M. El-Kenawy et al., "Meta-heuristic optimization and keystroke dynamics for authentication of smartphone users," *Mathematics*, vol. 10, no. 16, 2022, Art. no. 2912.

[25] Y. Kim, N. Kwon, and D. Shin, "KDPrint: Passive authentication using keystroke dynamics-to-image encoding via standardization," ~~CoRR, vol. abs/2405.01080, 2024.~~

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

[27] C. Wu et al., "Liveness is not enough: Enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks," in *Proc. 29th USENIX Conf. Secur. Symp.*, 2020, pp. 2219–2236.

[28] Y. Cai et al., "FAMOS: Robust privacy-preserving authentication on payment apps via federated multi-modal contrastive learning," in *Proc. 33rd USENIX Conf. Secur. Symp.*, 2024, pp. 289–306.

[29] H. Fereidooni et al., "Authentisense: A scalable behavioral biometrics authentication scheme using few-shot learning for mobile platforms," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2023.

[30] D. Wang et al., "Understanding human-chosen pins: Characteristics, distribution and security," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 372–385.

[31] I. Muslukhov et al., "Know your enemy: The risk of unauthorized access in smartphones by insiders," in *Proc. 15th Int. Conf. Hum.-Comput. Interact. Mobile Devices Serv.*, 2013, pp. 271–280.

[32] M. Eiband et al., "Understanding shoulder surfing in the wild: Stories from users and observers," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2017, pp. 4254–4265.

[33] D. Marques et al., "Vulnerability & blame: Making sense of unauthorized access to smartphones," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–13.

[34] M. Casimiro et al., "A quest for inspiration: How users create and reuse pins," *Who Are You*, pp. 1–7, 2020.

[35] D. V. Bailey, P. Markert, and A. J. Aviv, ""I have no idea what they're trying to accomplish:" Enthusiastic and casual signal users' understanding of signal pins," in *Proc. 17th USENIX Conf. Usable Privacy Secur.*, 2021, pp. 417–436.

[36] R. Roberts et al., "Blue is the new black (market): Privacy leaks and re-victimization from police-auctioned cellphones," in *Proc. IEEE Symp. Secur. Privacy*, 2023, pp. 3332–3336.

[37] T. Van Hamme et al., "A novel evaluation framework for biometric security: Assessing guessing difficulty as a metric," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 8369–8384, 2024.

[38] S. Pinto and N. Santos, "Demystifying arm trustzone: A comprehensive survey," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–36, 2019.

[39] L. Sun et al., "Leap: Trustzone based developer-friendly TEE for intelligent mobile apps," *Trans. Mobile Comput.*, vol. 22, no. 12, pp. 7138–7155, 2022.

[40] X. Gong et al., "Redeem myself: Purifying backdoors in deep learning models using self attention distillation," in *Proc. IEEE Symp. Secur. Privacy*, 2023, pp. 755–772.

[41] A. Das, N. Borisov, and M. Caesar, "Tracking mobile web users through motion sensors: Attacks and defenses," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2016.

[42] F. D. Protection, "General data protection regulation (GDPR)," *Intersoft Consulting*, vol. 24, no. 1, 2018.

[43] Z. Sitová et al., "HMOG: New behavioral biometric features for continuous authentication of smartphone users," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 5, 2015, pp. 877–892, May 2016.

[44] Z. Shen, S. Li, X. Zhao, and J. Zou, "MMAuth: A continuous authentication framework on smartphones using multiple modalities," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1450–1465, 2022.

[45] M. Georgiev et al., "Common evaluation pitfalls in touch-based authentication systems," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2022, pp. 1049–1063.

[46] T. Yang and D. Wang, "RankGuess: Password guessing using adversarial ranking," in *Proc. IEEE Symp. Secur. Privacy*, 2025, pp. 682–700.
[47] D. Wang et al., "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.
[48] T. M. Cover, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1999.
[49] S. Boztas, "Entropies, guessing, and cryptography," Dept. Math., RMIT Univ., Tech. Rep., vol. 6, pp. 2–3, 1999.
[50] J. O. Pliam, "On the incomparability of entropy and marginal guesswork in brute-force attacks," in *Proc. Springer INDOCRYPT*, 2000, pp. 67–79.
[51] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proc. IEEE Symp. Secur. Privacy*, 2012, pp. 538–552.
[52] M. Frank et al., "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 1, pp. 136–148, Jan. 2013.
[53] J. Fierrez et al., "Benchmarking touchscreen biometrics for mobile authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, pp. 2720–2733, Nov. 2018.
[54] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality complete samples," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.
[55] M. Hu et al., "Behavioral biometrics-based continuous authentication using a lightweight latent representation masked one-class autoencoder," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 3, pp. 1908–1923, May/Jun. 2025.

Jingyu Yao received the MS degree in software engineering from the School of Computer Science, Shaanxi Normal University, in Xi'an, China. He is currently working toward the PhD degree with the College of Cyber Science, Nankai University, Tianjin, China. His research interests include applied cryptography, implicit authentication, and risk-based authentication. Contact information: jingyuyao@mail.nankai.edu.cn.

Ding Wang received the PhD degree in information security from Peking University, in 2017, and was supported by the "Boya Postdoctoral Fellowship" in Peking University from 2017 to 2019. Currently, he is a full professor with Nankai University. As the first author (or corresponding author), he has published more than 120 papers at venues like IEEE S&P, ACM CCS, NDSS, USENIX Security, IEEE TDSC and IEEE TIFS. His research has been reported by over 200 media like Daily Mail, Forbes, IEEE Spectrum and Communications of the ACM, appeared in the Elsevier 2017 "Article Selection Celebrating Computer Science Research in China", and resulted in the revision of the authentication guideline NIST SP800-63. He has been involved in the community as a PC Chair/TPC member for more than 60 international conferences such as NDSS 2023-2025, ACM CCS 2022, PETS 2022-2024, ACSAC 2020-2024, RAID 2024/2023, ACM AsiaCCS 2025/2022/2021, FC 2026/2025, IFIP SEC 2018-2021, ICICS 2018-2024, SPNCE 2020-2022. He has received the "ACM China Outstanding Doctoral Dissertation Award", the Best Paper Award at INSCRYPT 2018, the Outstanding Youth Award of China Association for Cryptologic Research, the Young Scientist Nomination Award for Powerful Nation, and the First Prize of Natural Science Award of Ministry of Education. His research interests focus on passwords and cryptographic protocols.