mathematics

MDPI

Article

Enhancing Sensor-Based Mobile User Authentication in a Complex Environment by Deep Learning

Zhengqiu Weng [1,2], Shuying Wu [3,*], Qiang Wang [4] and Tiantian Zhu [2]

[1] School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou 325035, China; derisweng@163.com
[2] College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China
[3] School of Artificial Intelligence, Wenzhou Polytechnic, Wenzhou 325035, China
[4] School of Economics and Management, Wenzhou University of Technology, Wenzhou 325035, China
* Correspondence: wushuying@wzpt.edu.cn

Abstract: With the advent of smart mobile devices, end users get used to transmitting and storing their individual privacy in them, which, however, has aroused prominent security concerns inevitably. In recent years, numerous researchers have primarily proposed to utilize motion sensors to explore implicit authentication techniques. Nonetheless, for them, there are some significant challenges in real-world scenarios. For example, depending on the expert knowledge, the authentication accuracy is relatively low due to some difficulties in extracting user micro features, and noisy labels in the training phrase. To this end, this paper presents a real-time sensor-based mobile user authentication approach, ST-SVD, a semi-supervised Teacher–Student (TS) tri-training algorithm, and a system with client–server (C-S) architecture. (1) With S-transform and singular value decomposition (ST-SVD), we enhance user micro features by transforming time-series signals into 2D time-frequency images. (2) We employ a Teacher–Student Tri-Training algorithm to reduce label noise within the training sets. (3) To obtain a set of robust parameters for user authentication, we input the well-labeled samples into a CNN (convolutional neural network) model, which validates our proposed system. Experimental results on large-scale datasets show that our approach achieves authentication accuracy of 96.32%, higher than the existing state-of-the-art methods.

Keywords: mobile user authentication; deep learning; large-scale data analysis; implicit authentication; user micro feature

MSC: 68T07; 68T09

Citation: Weng, Z.; Wu, S.; Wang, Q.; Zhu, T. Enhancing Sensor-Based Mobile User Authentication in a Complex Environment by Deep Learning. *Mathematics* **2023**, *11*, 3708. https://doi.org/10.3390/math11173708

Academic Editors: Xiaofeng Xu, Jun Wu and Kaijian He

Received: 25 July 2023
Revised: 22 August 2023
Accepted: 25 August 2023
Published: 29 August 2023

1. Introduction

With the advancement of mobile communication technology and hardware updates, mobile intelligent terminal devices have become increasing popular and are widely used in people's daily lives. The mobile application market has an anticipated annual consumer spending of USD 233 billion on the Apple App Store and Google Play store in 2022–2026 [1]. According to Grand View Research, the global mobile application market is expected to grow at a compound annual growth rate (CAGR) of 13.4% from 2022 to 2030 [2]. As more and more users transmit and even store their private data in mobile devices, it becomes very important to avoid information leakage in the network attack and defense. Especially in the instance where users are part of organizations such as enterprises, governments, and national infrastructure, the information leakage caused by advanced persistent threat attacks, conducted mainly by accessing the mobile devices to collect valuable confidential information, would be a catastrophe. Therefore, in order to safeguard their privacy and security, it is urgent to design suitable and robust user authentication models based on both the application scenarios and the features of mobile devices.

The current authentication methods for mobile users can be broadly categorized into two types: knowledge-based and biometric-based. Knowledge-based authentication methods [3] require users to explicitly input information such as passwords or patterns. Although these methods are widely used due to their low cost, they have to face challenges from the perspective of usability and security [4,5]. For example, (1) they are prone to various attacks such as brute force, shoulder surfing, smudge, inference, and social engineering attacks, and (2) inputting the same password repeatedly in small dialog boxes would have impact on the user physical experience. In contrast, biometric-based authentication methods [6] can mitigate the above issues to some extent. However, frequently using the facial or fingerprint recognition may also, especially in some ungoverned scenarios, bring psychological discomfort. Therefore, both usability and security should be the priority for the authentication systems. Recently, there has been extensive research on user dynamic authentication methods based on motion sensors. These methods identify users' information with machine learning or deep learning methods to discern unique behavioral patterns through their gait/gestures, because no privacy-related permissions are required in motion sensors. Then, researchers collect a series of non-privacy motion sensor data, such as accelerometer, gravity sensor, and gyroscope sensor data (through the system-level interface any application can obtain the data). However, in real-world and complex environments, it is hard to distinguish user micro features. The major challenges are as follows:

(1) In mobile user authentication, the majority of sensory signal transformation methods rely on expert knowledge [7,8]. They lack the in-depth data mining of motion sensor signals, and they are unable to effectively learn the complex nonlinear relationships between invariant features.
(2) Label noise is widely present in the training phase, e.g., device owners lend their phones to others. Most research has paid more attention to signal denoising, but less to reducing label noise [9,10]. If a classifier is trained with incorrect labels, continuous errors can accumulate. Even if labeled samples are obtained from the target person, the classifier may still fail to authenticate the device owner.
(3) The quality of handmade features is crucial for the performance of most classifiers. However, when dealing with complex mixed motion sensor signals, relying solely on statistical features can lead to critical information loss [7,11–13]. Additionally, the feature extraction process is typically fixed by the determined algorithm, whereas iterative optimization can be used to update the parameters of the classification model. This can hinder the improvement of algorithms for sensor-based mobile user authentication.

To address the aforementioned challenges, this paper proposes an efficient sensor-based mobile authentication method under a complex environment. Our system implicitly collects motion sensor data in the interaction between users and their mobile phones and utilizes S-transform (ST) [14] and singular value decomposition (SVD) to enhance their micro features. One-dimensional time-series signals are then sent to a server for label refactoring through a Teacher–Student (TS) tri-training semi-supervised algorithm. The resulting data are then used to train a CNN (convolutional neural network) model for feature extraction and classification, which is returned to the mobile terminal (client side) for real-time authentication. Experimental results on large-scale real-world data demonstrate that our method achieves higher accuracy, compared with existing state-of-the-art methods [7,9–12]. The main contributions of our work include the following aspects:

(1) A 2D image encoding method, ST-SVD, is proposed for 1D time-series signals by using a multi-resolution analysis. This method combines S-transform (ST) and singular value decomposition (SVD) to obtain an optimal S-matrix to enhance the time-frequency characteristics of sensory signals. Then, it allows CNN to learn high-level features. Moreover, this method takes into account the spatio-temporal properties of sensory signals.

(2) A semi-supervised Teacher–Student (TS) tri-training algorithm is proposed to address label noise in real-world motion sensor datasets. This algorithm effectively eliminates the negative impact of noisy labels and provides high-quality training data for the model.
(3) Integrating the aforementioned methods, we design a system with a client–server (C-S) architecture. Experimental results on large-scale real-world datasets demonstrate that the proposed system achieves a high authentication accuracy, outperforming existing state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 illustrates the system design in detail. Section 4 shows the overall evaluation of our system. Section 5 concludes our work.

2. Related Work

Authentication methods for mobile device users. Mobile device user authentication methods are essential for protecting user data. Currently, they can primarily be classified into three types: knowledge-based methods [5], static feature-based methods [6], and dynamic behavior-based methods [7]. Knowledge-based methods require users to explicitly enter a digital password or gesture pattern to unlock the mobile device or log into an application. These methods can only verify whether a user knows the credential but cannot determine whether the user is the device owner. Furthermore, they pose certain risks, such as poor human–computer interaction experience and privacy leakage. Previous studies have shown that they can easily be broken by brute force attacks [15], smudge attacks [16], shoulder surfing attacks [17], and sensor inference [18]. In contrast, static biometric authentication methods are based on fingerprints and faces, which can achieve relatively high recognition accuracy [19–21]. However, except concerns about the user experience and privacy leakage mentioned in Section 1, recent research has shown that misusing fingerprint APIs on Android can make applications vulnerable to various attacks [19], and face recognition methods based on deep learning algorithms have been proven to be circumvented by sophisticated attackers [20,21]. Dynamic behavior-based authentication methods access data from built-in sensors in mobile devices, including environmental locations, keystroke behaviors, finger movements, etc., and they work through the combination of feature engineering and model training. These methods may call privacy-related permissions to obtain user privacy data.

Static/dynamic feature authentication methods are based on credential technology and privacy risks. Given their inherent drawbacks, numerous researchers [8,11–13,22–27] proposed dynamic user authentication based on motion sensors, which has the following limitations: First, most user authentication based on motion sensors requires a user to use their phone at a fixed location or perform specific actions (in a lab environment), which is unrealistic and results in a large number of noisy labels in complex environments (non-lab environments). Second, the credibility of the data collected in complex environments is often questionable, for example, unreliable labels are generated if the phone is used by others except the owner. To achieve both good user experience and high authentication accuracy simultaneously, this paper proposes an effective sensor-based mobile user authentication method in complex environments.

Data denoising method for sensor-based mobile user authentication. Currently, most dynamic authentication methods utilize motion sensors [8–13,22–27]. They do not consider the impact of hardware noise. Then, it is very difficult for them to handle unlabeled data (noise data) in real environments. Finally, overfitting occurs and authentication accuracy decreases. In order to address the noise, some researchers [8] proposed noise elimination algorithms to obtain an effective dataset in the data preprocessing stage, but these algorithms cannot precisely distinguish the mislabeled and training samples. To overcome it, researchers used semi-supervised methods, combining noisy data with a set of clean labels [7]. Zhu et al. [7–9] observed that flat data cannot reflect the discrepancies of different user patterns in its collection. Removing it will omit analyzing its usability.

Additionally, when collecting users' data, researchers find the data are often mislabeled, such as unreliable labels we mentioned above. This paper simultaneously considers the noise and mislabeled data during the training process, fitting the training requirements to the greatest extent to provide high-quality data.

Mobile user authentication model based on motion sensors. The existing research methods using motion sensors [8–13,22–27] continuously collect sensor data and establish corresponding models to verify users' ID. Lu et al. [25] used unsupervised learning algorithms to process unlabeled data, but this method resulted in high latency. Additionally, unsupervised clustering algorithms with parameter adjustment have high overhead, and the parameter's generalization needs to be verified. Zhu et al. [7] designed a semi-supervised online learning algorithm. It has a high level of accuracy and low latency in processing unlabeled data under relatively complex environments, but the classification they used (binary class SVM) is not applicable to time series data due to ignoring the context of user behaviors. Furthermore, most existing studies [8–13,22–27] assume the input data are sufficient, which is not considered in real complex environments. In contrast, this paper proposes the transformation of 1D signals into 2D images. Meanwhile considering the spatio-temporal characteristics of sensory signals, we extract spatio-temporal features using CNN to achieve high mobile user authentication accuracy.

3. Methodology

3.1. Overview Framework

The method proposed in this paper is shown in Figure 1, consisting of the four following steps:

1. Collecting mobile user authentication data. The sensor data are collected through human–computer interaction between users and mobile devices.
2. Executing 2D optimal S-matrix coding with singular value decomposition (SVD) and S-transform (ST). In this approach, the 1D time-series signals are transformed into 2D matrix features with ST first. Then, SVD is applied to the S-matrix to enhance user micro features.
3. Filtering mislabeled data and using Teacher–Student (TS) tri-training to correct mislabeled data during training. This step can further improve the quality of the training dataset.
4. Using a Convolutional Neural Network (CNN) model to extract features from 2D optimal S-matrix images. The trained CNN model is used to authenticate whether the user is the device owner.

3.2. Sensory Signal Collecting

With the advancement in hardware technology, chips such as the Graphics Processing Unit (GPU) and Tensor Processing Unit (TPU) are being increasingly integrated into mobile devices to meet the ever-growing computational needs. Likewise, a plethora of sensors, e.g., accelerometer, gyroscope, gravity, light, and heart rate sensor, are being embedded into smart devices as privacy-neutral sensors. Specific tasks can be achieved through these sensors perceiving user behavior-related data.

Mobile devices are often equipped with various sensors of different functions to collect user behavioral features, thus eliminating cumbersome explicit operations. The more the sensor data are used for authentication, the less the attackers who can bypass the corresponding authentication system are able to do so. However, this also increases the probability that the authentication system rejects authorized device users. Therefore, we have to make the following selection criteria: (1) The sensor needs to be universal, that is, the sensor is popular and can be embedded in various mobile devices. (2) The sensor needs to be privacy-neutral, that is, the collected information does not contain personal identity-related content. For example, cameras, microphones, and GPS can, respectively, obtain users' facial features, sound, and visited locations. All these details are what users do not want to disclose to others. Under these criteria, touch sensors will not be considered

because they not only collect users' fingerprints but also record their touch behaviors on the device. (3) The sensors need to be environment-neutral. For example, in noisy environments, the microphone is not effective, and in strong light, the usability of cameras is affected.

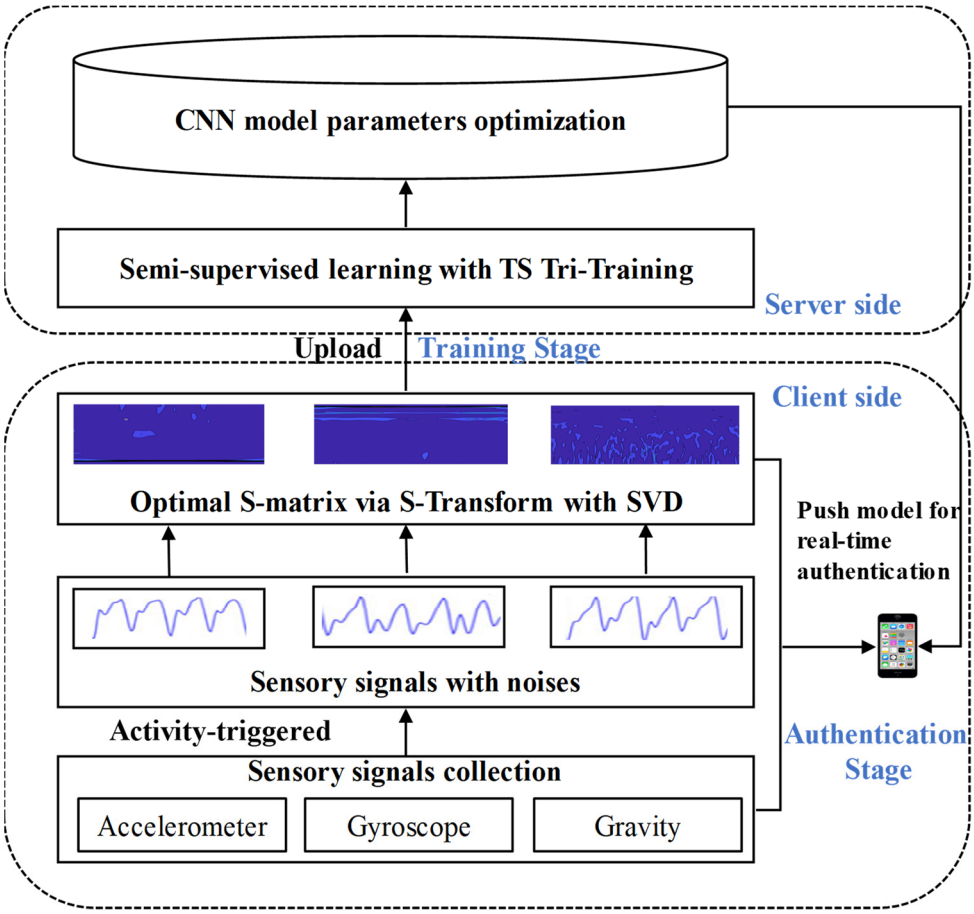

Figure 1. Sensor-based mobile user authentication process.

To effectively balance the relation among security (i.e., the system's underlying API interface can be safely used on non-rooted smartphones), privacy (i.e., the data collection does not require any privacy-related permission), and usability (i.e., any third-party application can be called to achieve device-level protection), we collect the raw data with accelerometer sensors, gyroscope sensors, and gravity sensors and conduct a large number of experimental comparisons to eventually achieve user privacy-neutral trustworthy authentication. Although the gravity sensor is a software-based sensor that utilizes accelerometer and gyroscope sensor data for processing, it can record the absolute position of the user when they use the mobile device, which is regarded as an important feature of user authentication.

We implicitly collect the above three types of motion sensor data when users use their mobile phone. Firstly, when a user opens a mobile phone application (representing a user behavior feature), we collect the smartphone's accelerometer, gyroscope, and gravity sensor X, Y, and Z axes' values as raw time sequence signal data. Inspired by [8], the following two scenarios can trigger the data collection: (1) the screen of the smartphone is lit up and (2) the foreground application of the smartphone is switched. This can greatly improve the security of mobile devices because the authentication starts no matter when users operate their mobile devices or open a particular application. In the real world, when the application starts, the ideal duration of data collection is generally 2–4 s, and its frequency is 50 Hz. Setting the collection duration t ($2\text{ s} \leq t \leq 4\text{ s}$) in advance, we obtain effective

user behaviors. If the time when users use their phone is less than t (in second), that is, the screen is turned off in the collection, the collection is terminated. In this paper, we set t to 3 s.

3.3. Sensory Signal with 2D Coding Conversion

Numerous advanced techniques [11–19] for signal processing, including time-frequency diagrams and histograms, have been developed to transform 1D time-series signals into 2D matrix features. S-transform (ST) is used to create a time-frequency representation of signals. ST outperforms the Short-time Fourier Transform (STFT) since the window in ST is adaptively wider in the time domain at lower frequencies and narrower at higher frequencies, and it provides excellent frequency localization at low frequencies and good time localization at higher frequencies. Inspired by [14,28], ST is introduced as a means of 2D signal encoding. This enables deep learning models to extract complex time-frequency features and achieve a high accuracy classification for different users.

Using a localized Gaussian window, both movable and scalable, we implement ST from the time domain to two-dimensional frequency translation domains and then to Fourier frequency domains. Then, we obtain the amplitude-frequency-time spectrum and phase-frequency-time spectrum, which are useful for defining local spectral characteristics of a sensory signal. The phase information incorporation in ST makes it an excellent candidate for sensor-based mobile user authentication. The continuous ST is defined in Equation (1), where the function h(t) is a continuous wavelet transform (CWT) with a specific mother wavelet multiplied by the phase factor. The mother wavelet is a waveform primarily used as a basis for the CWT. It provides the oscillatory properties necessary to analyze signals with varying frequency content at different time scales. Here, we use Morlet wavelet as the mother wavelet.

$$S(\tau, f) = \int_{-\infty}^{+\infty} h(t) \frac{|f|}{\sqrt{2\pi}} e^{-\frac{(\tau-t)^2 f^2}{2}} e^{-i2\pi f t} dt \tag{1}$$

Since the signal is obtained in discrete time series with a sampling interval T, let f correspond to n/NT and τ correspond to jT, the discrete ST can be represented as in Equation (2).

$$\begin{gathered} S[jT, \tfrac{n}{NT}] = \sum_{m=0}^{M-1} H[\tfrac{m+n}{NT}] e^{-\frac{2\pi^2 m^2}{n^2}} e^{\frac{i2\pi mj}{N}}, n \neq 0 \\ S[jT, 0] = \sum_{m=0}^{M-1} H[\tfrac{m}{NT}], n = 0 \end{gathered} \tag{2}$$

Here, n ranges from 0 to $N-1$, where N represents the sampling number. The range of j is from 0 to 1. ST and produces a complex matrix known as the S-matrix, with rows representing time and columns representing frequency. Put it another way, each column represents a local spectrum at a specific time. Frequency-time contours are then visually obtained from the S-matrix. It is important to note that the original elements of the S-matrix are complex exponentials. To visually represent frequency-time amplitude, we have to calculate and process the magnitude of each S-matrix element.

Due to the non-stationary and non-Gaussian characteristics of sensory signals, the SVD algorithm is utilized to denoise signals and enhance user micro features. An $m \times m$ matrix A can be formulated as: $A = U \sum V^T$, where $U = [u_1, u_2, \ldots, u_m]$, and $V = [v_1, v_2, \ldots, v_m]$ are orthogonal matrices. The column vectors of matrices U and V refer to the orthonormal eigenvectors of matrices AA^{T} and $A^{\mathrm{T}}A$, respectively. The diagonal matrix $\sum$ arranges the singular values in descending order, which is designated as $\sum = [diag(\sigma_1, \sigma_2, \ldots, \sigma_l), 0]$, where $l = \min(m, n)$ and $\sigma_1 \geq \sigma_2 \geq, \ldots, \geq \sigma_l$.

From the aforementioned analysis, it is evident that how to select a suitable SV order is imperative for SVD-based denoising. Therefore, to attain SVD adaptive decomposition,

this paper adopts the maximum SV mean method to determine a reasonable SV order. The SV mean is defined by Equation (3) below:

$$Z_i = \frac{\alpha_{i-1} - \alpha_{i+1}}{2} \tag{3}$$

where Z_i is the i-th SV mean. $k = \arg(maxZ_i)$ is the best SV order. α_i represents the i-th SV in the sorted descending sequence of singular values obtained from the SVD algorithm for a given signal. The maximum SV mean can originate from the uncorrelation of the fault component α_{i-1} and noise disturbance component α_{i+1} in the signal. Thus, the maximum mean value can be regarded as a demarcation point between the fault component and noise disturbance component. Ultimately, the S-matrix A is reconstructed and represented as B if we only preserve the first k SVs.

3.4. Label Correction with TS Tri-Training

In previous research, training data labels are often assumed to be perfectly tagged without any noise. However, in reality, a vast majority of labeled samples are imperfect due to various factors in the training phase. Moreover, manual labeling incurs significant time and labor costs. To address this, we use an automatic and general algorithm named Teacher–Student (TS) tri-training to mitigate label noise. TS tri-training ensures classifier differentiation by training differential data subsets extracted from the original dataset. As a result, the algorithm can recognize and correct noisy labeled samples.

TS tri-training involves the following steps: First, the original dataset I (a 10-day dataset, as will be described in Section 4) is partitioned into three sub-sets; a labeled dataset L, which may contain mislabeled data, an unlabeled dataset U, and a verification dataset V. Then, we assume that when a user starts using the mobile phone in the initial period, we regard the user as the owner. Accordingly, the data in the first two days belongs to L, while the data in the next six days—which may have been mislabeled—is placed in U. Finally, the data in the last two days, representing the user's own test data, are put in V.

Through Bootstrap Sampling on dataset L, we obtain three labeled training subsets Lc, Ll, and Ln, of which each is 1/3 of the total L. Then, three classifiers, Cd, Cl, and Cn, stem from the above three subsets, respectively. Pseudo-labeled samples in the form of "minority obeying majority" [29] are then generated by using these classifiers. Specifically, if Cd and Cl predict an unlabeled sample s is positive with a probability greater than the confidence coefficient τt, and Cn predicts it is negative with a probability less than τs, then Cd and Cl are regarded as teachers, and Cn as a student. Finally, as a pseudo-labeled positive sample, s is provided to Cn for learning. If the two teacher classifiers make the same prediction for the same unlabeled sample, this sample is considered as one with a higher label confidence and then added to the labeled training set of the third classifier after being labeled. The "minority obeying majority" [29] aims to eliminate classification errors. Though noise exists within the labeling process, it can be offset to some extent given the large number of samples utilized in this study. The intersection of Lc, Ll, and Ln is then taken as the final labeled training dataset L′. We choose binary classification because it can better distinguish whether the device is used by its owner. According to the tri-training based image classification methods [30], classifiers of Decision Tree, Support Vector Machine, and K-Nearest Neighbors require a certain proportion of counterexamples. Strategic sampling [8] is adopted to extract the most representative counterexample data samples from the massive data of other people. We set $\tau t = 0.8$ and $\tau s = 0.2$ through fine-tuning.

3.5. Convolutional Neural Networks Construction

In order to achieve sensor-based mobile user authentication, we use a Convolutional Neural Network (CNN) structure that comprises an input layer, two convolutional layers (CL1, CL2), followed by max pooling layers (MP1, MP2), two fully connected layers (FCL), and an output layer as a classifier. A detailed account is presented in Table 1.

Table 1. Structure of CNN model.

Layer	Input Shape	Structure	Output Shape
Input	9×150	-	$9 \times 100 \times 150$
CL1	$9 \times 100 \times 150$	$20@(2 \times 3)$	$20@(24 \times 148)$
MP1	$20@(24 \times 148)$	2×4	$20@(12 \times 37)$
CL2	$50@(12 \times 37)$	$50@(3 \times 8)$	$50@(10 \times 30)$
MP2	$50@(10 \times 30)$	2×6	$50@(5 \times 5)$
FC	750×1	256×1	256×1
Output	256×1	2×1	2×1

Input Layer. We collect nine time signals (i.e., X, Y, and Z axes' values from accelerometer, gyroscope, and gravity sensor as raw time sequence signal data) as the input (9×150). Then, an ST/SVD matrix is created (totally $9 \times 100 \times 150$). As mentioned in Section 3.3, the optimal S-matrix is obtained and used as an input for the CNN model. The input layer, unlike traditional RGB images, is solely connected with a matrix derived from 2D encoding features. Moreover, it should be noted that the original elements of the S-matrix are complex exponential. Thus, computing the magnitude of each element is necessary for further processing the visualization of frequency-time amplitude.

Convolutional Layers. Two-dimensional convolutional operations are carried out across these layers, where local features are transformed into global ones. Their main purpose is to achieve weight sharing, and then improve the efficiency and feasibility of the model. In this paper, the stride is set to 1. The sizes of the convolutional kernels in the Convolutional Layer1 (CL1) and the Convolutional Layer2 (CL2) are 2×3 and 3×8, respectively. The number of kernels in CL1 is 20, while in CL2, it is 50. The activation function is ReLU.

Max-Pooling Layers. Max-pooling layers down-sample vital and invariant information. These layers aim to reduce training time and avoid overfitting. The sizes of the Max-Pooling Layer 1 (MP1) and the Max-Pooling Layer 2 (MP2) are 2×4 and 2×6, respectively.

Fully Connected Layers. These layers aim to flatten the learned features into a single vector. The size of the Fully Connected Layer (FCL) is 256×1.

Output Layer. The output layer serves as a classification vector via a softmax function (loss function), and the optimizer is Adam.

4. Experimental Evaluations

This section mainly focuses on testing the performance and accuracy of the mobile user authentication method based on feature enhancement and semi-supervised learning. In the experimental environment, we use a server based on Ubuntu 20.04, equipped with an Intel(R) Intel Xeon E5 CPU and 128 GB memory.

4.1. Dataset

The dataset consists of two parts. The first part (dataset I) is from a well-known domestic Internet company, with 1513 volunteers aged from 20 to 60. With the users' permission, the dynamic data are collected as authentication data when they use their phones in complex environments. The data are from different users in an unsupervised manner and contain a large number of potential noisy labels. In the two scenarios triggering the data collection in Section 3.2, the built-in acceleration sensor, the gyroscope sensor, and the gravity sensor of the mobile devices capture relevant sensor signals at a rate of 50 Hz. The collection lasts for 10 days, and we obtain a total of 283,006,659 pieces of valid datum, with an average of 187,050 pieces per user. Dataset I is randomly divided into three groups. For each volunteer, 60% data are used for training, 20% are for validation, and 20% are for testing (to ensure the generalizability of the experimental results, all tests are cross-validated 10 times). The second part (dataset II) comes from 20 volunteers in our laboratory, aged between 20 and 60 in a supervised environment. The collection lasts for

10 days, and a total of 600,000 pieces of valid datum are collected. Dataset II is mainly used to test the label denoising algorithm.

During the label correction, classifiers such as Decision Tree, Support Vector Machine, and *K*-Nearest Neighbors are utilized to partition the labeled dataset into training and test sets at a ratio of 4:1 in the model training. The ratio of positive to negative samples is kept at 1:5, which is the same with that in the testing.

In the hybrid CNN-based training phase, the labeled dataset L' is divided into the training and test sets, at a ratio of 4:1 as well. As in the label correction, the ratio of positive to negative samples is maintained at 1:5 as well.

For all datasets, data collection occurs at a frequency of 50 Hz, and each data collection lasts for 3 s. We use the International Mobile Equipment Identity (IMEI) to identify each user's ID and distinguish them on the server side.

4.2. Evaluation Metrics

We employ the following four metrics to evaluate the efficacy and accuracy of the model: True Positive (TP), which denotes the number of owners who have been identified correctly as the owners; False Positive (FP), which signifies the number of non-owners who have been misidentified as the owners; True Negative (TN), which indicates the number of non-owners who have been identified correctly as the non-owners; and False Negative (FN), which represents the number of owners who have been misidentified as the non-owners. In terms of classification accuracy, the following indicators are listed:

The authentication rate of the owner (sensitivity) is:

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

The authentication rate of non-owner (specificity) is:

$$TNR = \frac{TN}{FP + TN} \tag{5}$$

The total accuracy is:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

4.3. Performance of ST-SVD on Sensory Signals

To assess the efficiency of ST-SVD, we compared traditional ST and generalized ST [31]. We analyzed sensory signals from acceleration sensors of two different users (randomly selected from our dataset) to illustrate the superiority of ST-SVD. Figures 2 and 3 compare the original sensory signals, the matrix obtained from adopting generalized ST, the matrix obtained from adopting traditional ST, and the matrix obtained from adopting ST-SVD (our proposed method). Time-frequency diagrams obtained from generalized ST are poor in extracting features, whereas traditional ST is sensitive to background noise. ST-SVD depicts clearer user patterns excellently, revealing the sensitivity of sensor-based mobile user authentication. ST-SVD accurately captures users' subtle behavioral differences (i.e., user micro features).

From Figures 2 and 3, it can be concluded that ST-SVD exhibits energy concentration superior to traditional S-transform and Generalized S-transform (the X-axis represents the second t(s)). Additionally, ST-SVD obtains a more distinct essential characteristic, location, in user usage patterns. ST extracts multi-resolution features of sensory signals, and SVD plays a significant role in S-matrix reconstruction, which enhances user micro features in time-frequency diagrams.

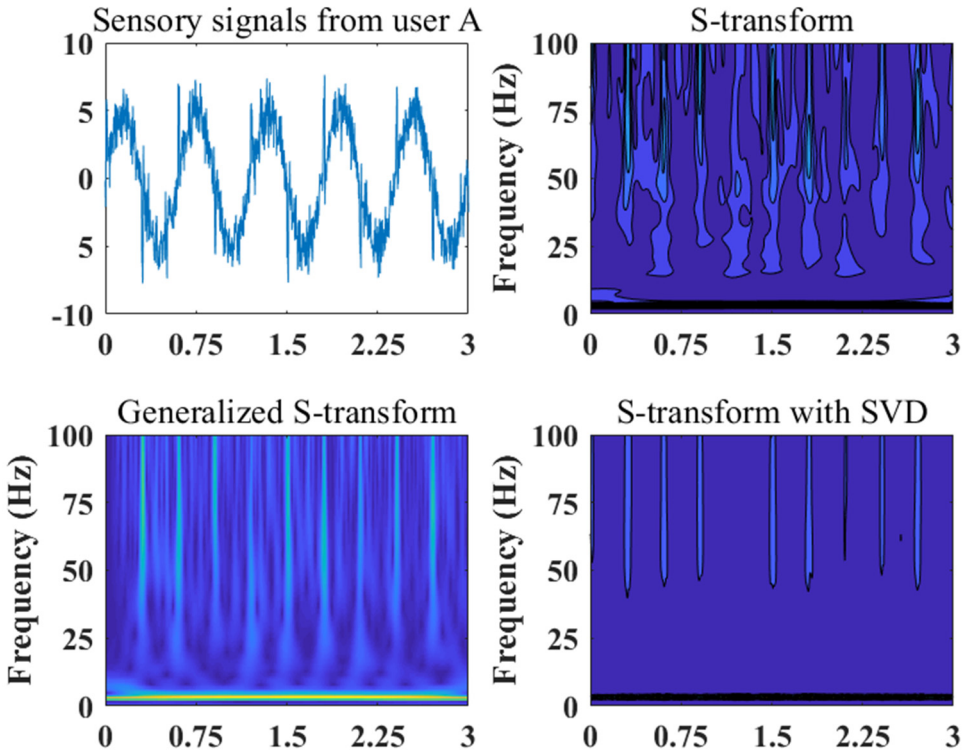

Figure 2. Comparison of different signal analysis results on User A.

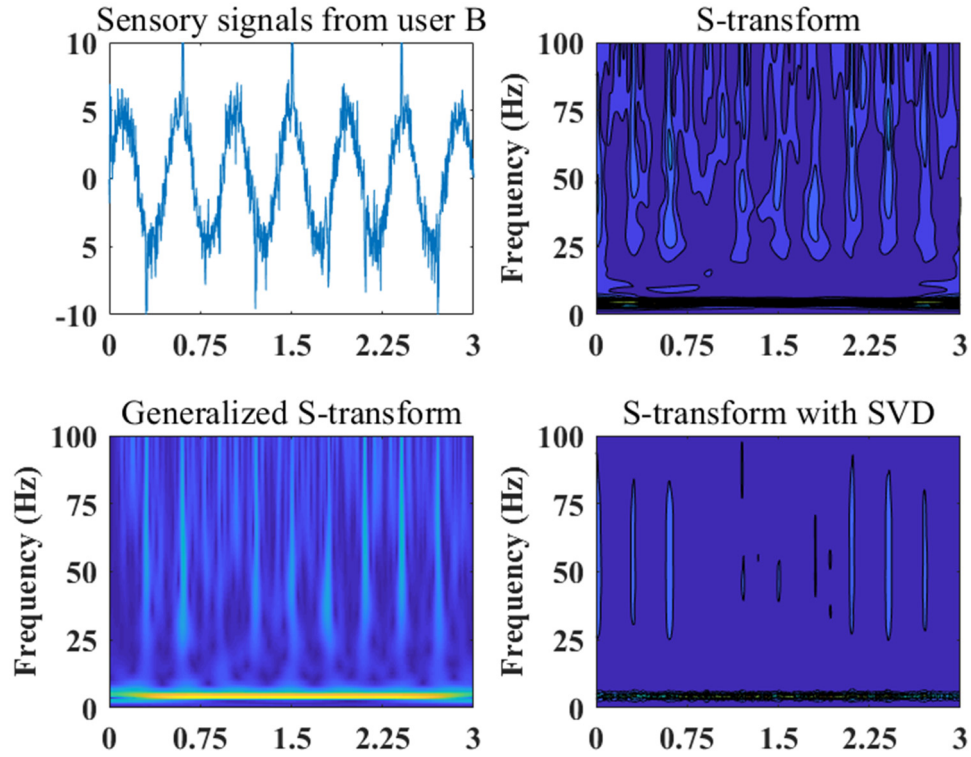

Figure 3. Comparison of different signal analysis results on User B.

4.4. Overall Accuracy on Large-Scale Dataset

Figure 4 displays the classifying performance of training, validation, and test data on dataset I with overall accuracy. It shows that all of them have a consistent and increasing convergence trend with the increase of epochs (the X-axis represents the epoch).

Notably, the accuracy of the test dataset reaches 96.32% on the 36th epoch. As mentioned, Convolutional Neural Networks (CNNs) are inspired by biological feedforward Artificial Neural Networks (ANNs) and closely resemble the mammalian visual cortex. This enables classifiers to achieve high-precision diagnoses through learning specific patterns. The average training latency for 1513 users is approximately 4825 s.

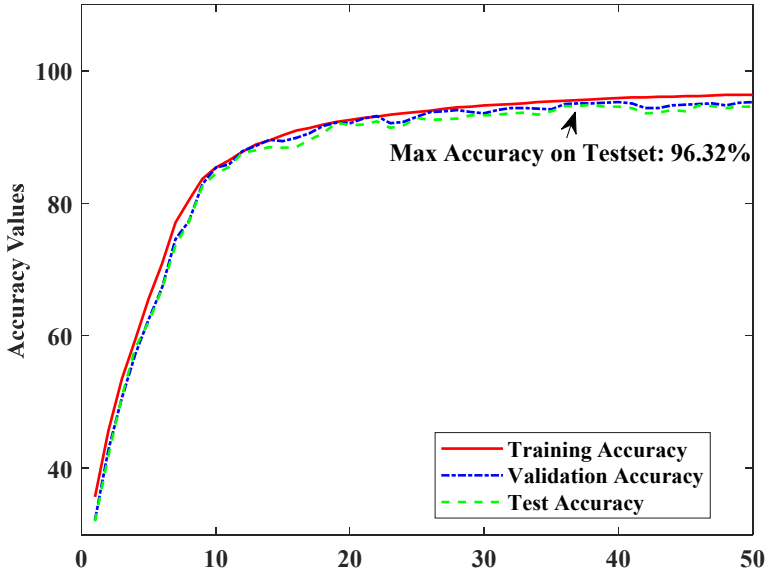

Figure 4. Comparison of different signal analysis results on dataset I.

Comparing TPR, TNR, accuracy of the method with and that without tri-training, Table 2 presents the average classification results of 10 cross-validations on 1513 users. "With TS tri-training" means the noisy labels have been removed, and vice versa.

Table 2. Average classification results with tri-training and without tri-training.

Method/Index	TPR	TNR	Accuracy	F1-Score
With TS tri-training	74.56%	98.13%	96.32%	0.9514
Without TS tri-training	72.22%	92.68%	92.89%	0.9159

As shown in Table 2, we can see that with TS tri-training, the average TPR, TNR, and accuracy increase from 72.22%, 92.68%, and 92.89% to 74.56%, 98.13%, and 96.32%, respectively, that is, eliminating noisy labels can significantly improve the final authentication. In practice, it is common that the number of positive samples (interactions from the authorized device owner) is much larger than the number of negative samples. This is because the authorized device owner primarily interacts with the device. However, a significant number of negative samples can accurately depict the usage pattern of the non-owners. This can prevent brute-force attacks from bypassing the authentication. To this end, in the training stage, the ratio of the number of samples by the owner to that by the non-owner is 1:5 to record the non-owners' behavior [7,9]. In Table 2 we find that the trained classifier can accurately represent mobile phone usage patterns of non-owners (TNR) but may miss some of the owners' usage patterns (TPR). As TN is often more critical than TP, this paper proposes to increase TN when we configure the authentication parameters. In practice, we can adjust the classification threshold for different purposes. For example, we can set a high threshold for a high security application to prevent owners' mobile devices from malicious use.

In addition, this paper employs large-scale datasets to evaluate the proposed classification method. Table 3 shows the accuracy comparison of our proposed method and other

state-of-the-art methods [7,9–12,26,27]. We find our ST-SVD+CNN achieves higher accuracy. It is noteworthy that the binary-class SVM we use for training is similar to CNN+SVM (binary-class) [10]. However, our ST-SVD+CNN and CNN+SVM (binary-class) [10] outperform SVM (binary-class) [7] in feature extraction, because the latter manually extracts signs. Additionally, combining ST-SVD with CNN leads to better results compared with only using CNN, as shown in [10]. This is because transforming time-series signals into 2D time-frequency images through ST-SVD is more appropriate for CNN-based feature extraction.

Table 3. Comparison of our work and other related work.

Classifier	TPR	TNR	Accuracy	F1-Score
ST-SVD+CNN (Ours)	**74.56%**	**98.13%**	**96.32%**	**0.9514**
LSTM [9]	74.35%	97.60%	95.58%	0.9420
CNN+SVM(binary-class) [10]	74.26%	97.31%	95.01%	0.9375
SVM(binary-class) [7]	73.59%	96.42%	94.67%	0.9304
HMM [12]	71.74%	91.98%	90.54%	0.8896
Random forest [27]	72.38%	93.95%	92.36%	0.9132
DTW [26]	66.18%	89.60%	86.49%	0.8472
SVM(one-class) [11]	66.25%	89.64%	86.51%	0.8490

4.5. Performance on Noisy Labels Elimination

To validate the effectiveness of the proposed TS tri-training algorithm, we conducted experiments on dataset II with different noisy ratios. Specifically, we added noise to dataset II, which is a clean labeled dataset. The dataset contains 600,000 data samples, with a 4:1 split for training and testing. The ratio of the noisy labeled training set is set to 5%, 15%, 25%, 35%, and 45%, which means there are 24,000, 72,000, 120,000, 168,000, and 216,000 noisy labeled samples used for training, respectively.

We applied TS tri-training to reduce the number of noisy labels. Table 4 presents the performance of eliminating noisy labels. The experimental results show that the original training noisy ratio of 5%, 15%, 25%, 35%, and 45% is reduced to 0.35%, 0.68%, 1.82%, 4.53%, and 18.22%, respectively. It indicates that after TS tri-training, the remaining number of noisy labels in the training set is 1682, 3264, 7836, 21,744, and 87,456, respectively, and the accuracy reaches to above 94% at the noisy label ratio of 5%, 15%, 25%, and 35%, respectively, but not at the ratio of 45%. The reason is that the dataset at the noisy ratio of 45% is almost equivalent to a randomly labeled one, and it is unsuitable for training. Nevertheless, even under such difficult scenario, TS tri-training still shows an improvement.

Table 4. The overall accuracy before/after noisy labels elimination.

Before TS Tri-Training					
# of Training Set	480,000	480,000	480,000	480,000	480,000
Noisy Label Ratios	5%	15%	25%	35%	45%
False Labels in Training Set	24,000	72,000	120,000	168,000	216,000
After TS tri-training					
# of Training Set	480,000	480,000	480,000	480,000	480,000
Noisy Label Ratios	0.35%	0.68%	1.82%	4.53%	18.22%
False Labels in Training Set	1682	3264	7836	21,744	87,456
User Authentication Result(Before TS tri-training)					
Accuracy	93.85%	84.25%	76.37%	68.30%	53.09%
User Authentication Result (After TS tri-training)					
Accuracy	96.28%	96.12%	95.25%	94.03%	81.42%

4.6. Evaluation of Computational Cost

The experiment was conducted on a server equipped with an Intel Xeon E5 CPU, GeForce RTX 3090 Ti, and 128 G memory running Ubuntu 20.04. The average training latency for our CNN model was 30.54 s per training procedure with GPU acceleration. For TS tri-training semi-supervised learning, the average latency of each training procedure was 99.28 s. It should be noted that once the model is trained, it can be deployed to the client side for real-time verification. Therefore, clients are more concerned about their cost.

On the client side, Android is equipped with TensorFlow to perform authentication. To monitor the battery consumption, CPU usage, and memory usage, we employed the well-known Android performance testing tool, Emmagee [31]. Table 5 summarizes the obtained results.

Table 5. The overhead on three different devices.

Phone Model	Battery Consumption (mAh)	Data Collection		Authentication	
		CPU (%)	Memory (MB)	CPU (%)	Memory (MB)
Samsung S20	105.22/4000	1.20	11.05	6.38	69.38
Vivo Xplay 6	110.50/4080	1.22	11.12	6.21	65.26
M18	119.25/3400	1.25	11.09	6.46	72.64

To evaluate battery consumption, we asked a participant to use the client app for three hours, which includes the time for both data collection and offline authentication. Our application required less than 0.4% of battery in one hour. During offline authentication, CPU and memory usage on three different smartphones were slightly higher than those in data collection and ST-SVD. This is because in the real-time authentication, additional model tasks inevitably occur.

We also evaluated the latency of offline authentication when performing data collection, ST-SVD, and decision procedures 100 times on Samsung S20. The results are presented in Table 6, which shows that the entire process can be completed within 3230.60 ms. The latency introduced by steps other than data collection is negligible. Overall, the overhead on the client side fully met the requirement of real-world scenarios.

Table 6. Client Authentication Time on Samsung S20.

Procedure	Average Time (ms)
Data collection	3003.12
ST-SVD	199.58
Authentication	27.90
Overall	3230.60

4.7. Anti-Attack Capability Assessment

The security of sensor-based gait authentication has consistently been a concern of both the industry and academic communities. Previous studies [8,10] investigated its resilience to attacks, particularly mimicry attacks where an attacker observes a user's usage manner and imitates their authentic gestures and actions.

To launch a mimicry attack, we selected 20 individuals from dataset II and selected one individual as the victim in turns. First, we trained a classifier for the authorized device owner by fingerprinting their usage manner (each victim had 9240 samples for the model training). We then asked the remaining 19 individuals to imitate the victim's pattern one-by-one (100 times for each person, with each participant generating 30 samples per run). These samples were checked against the classifier, and we calculated the percentage of samples correctly labeled as other users. Our method was able to thwart imitation attacks with a probability of over 99.20%, even higher than the TNR (98.13% in Table 2) on the large

dataset. This may be attributed to the fact that our ST-SVD and CNN models represented the contextual content (e.g., the owner users' inherent usage patterns) of time series data. In this way it is difficult to bypass via imitation. Similar results have been reported in [7,9].

5. Conclusions

In this paper, we proposed a real-time authentication method for mobile users in complex environments, i.e., ST-SVD. To extract subtle user features, deal with label noise, and improve the authentication accuracy in real world, we combined S-transform and singular value decomposition to transform time-series signals into 2D time-frequency images, utilized T-S tri-training to reduce label noise in the training phrase, and inputted the well-labeled samples into a CNN model to obtain a set of robust parameters for user authentication finally. Moreover, we validated the effectiveness and the high tolerance for label noise of our system through large-scale real-world data. It can meet the requirements of generality, efficiency, and usability jointly in mobile user authentication. Our future work will consider reducing the size of the dataset and improving the generalization of authentication.

Author Contributions: Conceptualization, Z.W. and S.W.; methodology, T.Z.; software, Q.W.; validation, Z.W., S.W. and T.Z.; investigation, Z.W.; resources, Z.W.; data curation, Q.W.; writing—original draft preparation, Z.W.; writing—review and editing, Z.W. and S.W.; visualization, Q.W.; supervision, S.W.; project administration, Z.W.; funding acquisition, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Wenzhou Key Scientific and Technological Projects (No. ZG2020031) and supported by Key Research and Development Projects in Zhejiang Province (No. 2021C01117).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Store Intelligence Data Digest Report. 2022. Available online: https://sensortower.com/resources (accessed on 6 July 2023).
2. Mobile Application Market Size, Share & Trends Report. 2030. Available online: https://www.grandviewresearch.com/industry-analysis/mobile-application-market (accessed on 6 July 2023).
3. Alhakami, H.; Alhrbi, S. Knowledge based Authentication Techniques and Challenges. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 1–6. [CrossRef]
4. Dee, T.; Richardson, I.; Tyagi, A. Continuous nonintrusive mobile device soft keyboard biometric authentication. *Cryptography* **2022**, *6*, 14. [CrossRef]
5. Bošnjak, L.; Brumen, B. Examining security and usability aspects of knowledge-based authentication methods. In Proceedings of the 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, Croatia, 20–24 May 2019.
6. Ray-Dowling, A.; Hou, D.; Schuckers, S. Stationary mobile behavioral biometrics: A survey. *Comput. Secur.* **2023**, *128*, 103184. [CrossRef]
7. Zhu, T.; Qu, Z.; Xu, H.; Zhang, J.; Shao, Z.; Chen, Y.; Yang, J. RiskCog: Unobtrusive real-time user authentication on mobile devices in the wild. *IEEE Trans. Mob. Comput.* **2019**, *19*, 466–483. [CrossRef]
8. Ren, Y.; Chen, Y.; Chuah, M.C.; Yang, J. User verification leveraging gait recognition for smartphone enabled mobile healthcare systems. *IEEE Trans. Mob. Comput.* **2015**, *14*, 1961–1974. [CrossRef]
9. Zhu, T.; Weng, Z.; Song, Q.; Chen, Y.; Liu, Q.; Chen, Y.; Chen, T. Espialcog: General, efficient and robust mobile user implicit authentication in noisy environment. *IEEE Trans. Mob. Comput.* **2020**, *21*, 555–572. [CrossRef]
10. Zhu, T.; Weng, Z.; Chen, G.; Fu, L. A hybrid deep learning system for real-world mobile user authentication using motion sensors. *Sensors* **2020**, *20*, 3876. [CrossRef] [PubMed]
11. Sitová, Z.; Šeděnka, J.; Yang, Q.; Peng, G.; Zhou, G.; Gasti, P.; Balagani, K.S. HMOG: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Trans. Info. Forensics Secur.* **2015**, *11*, 877–892. [CrossRef]
12. Shen, C.; Li, Y.; Chen, Y.; Guan, X.; Maxion, R.A. Performance analysis of multi-motion sensor behavior for active smartphone authentication. *IEEE Trans. Info. Forensics Secur.* **2017**, *13*, 48–62. [CrossRef]
13. Lee, W.H.; Lee, R.B. Multi-sensor authentication to improve smartphone security. In Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP), Angers, France, 9–11 February 2015; pp. 1–11.
14. Stockwell, R.G.; Mansinha, L.; Lowe, R.P. Localization of the complex spectrum: The S transform. *IEEE Trans. Sig. Proc.* **1996**, *44*, 998–1001. [CrossRef]

15. Zoebisch, F.; Vielhauer, C. A test tool to support brute-force online and offline signature forgery tests on mobile devices. In Proceedings of the 2003 International Conference on Multimedia and Expo(ICME), Baltimore, MD, USA, 6–9 July 2003; p. III-225.
16. Aviv, A.J.; Gibson, K.; Mossop, E.; Blaze, M.; Smith, J.M. Smudge attacks on smartphone touch screens. In Proceedings of the 4th USENIX Conference on Offensive Technologies, Berkeley, CA, USA, 23–25 June 2010; pp. 1–7.
17. Zakaria, N.H.; Griffiths, D.; Brostoff, S.; Yan, J. Shoulder surfing defence for recall-based graphical passwords. In Proceedings of the Seventh Symposium on Usable Privacy and Security, Washington, DC, USA, 20–22 July 2011; pp. 1–12.
18. Xu, Z.; Bai, K.; Zhu, S. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks, New York, NY, USA, 16–18 April 2012; pp. 113–124.
19. Bianchi, A.; Fratantonio, Y.; Machiry, A.; Kruegel, C.; Vigna, G.; Chung, S.P.H.; Lee, W. Broken fingers: On the usage of the fingerprint API in android. In Proceedings of the NDSS, San Diego, CA, USA, 18–21 February 2018; pp. 1–15.
20. Sharif, M.; Bhagavatula, S.; Bauer, L.; Reiter, M.K. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In Proceedings of the 2016 ACM Sigsac Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 1–12.
21. Goswami, G.; Ratha, N.; Agarwal, A.; Singh, R.; Vatsa, M. Unravelling robustness of deep learning based face recognition against adversarial attacks. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 1–8.
22. Kwapisz, J.R.; Weiss, G.M.; Moore, S.A. Cell phone-based biometric identification. In Proceedings of the 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems, Washington, DC, USA, 27–29 September 2010; pp. 1–7.
23. Ho, C.C.; Eswaran, C.; Ng, K.W.; Leow, J.Y. An unobtrusive android person verification using accelerometer based gait. In Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia, Bali, Indonesia, 28–30 November 2012; pp. 271–274.
24. Zhu, J.; Wu, P.; Wang, X.; Zhang, J. Sensec: Mobile security through passive sensing. In Proceedings of the 2013 International Conference on Computing, Networking and Communications, San Diego, CA, USA, 28–31 January 2013; pp. 1128–1133.
25. Lu, H.; Huang, J.; Saha, T.; Nachman, L. Unobtrusive gait verification for mobile phones. In Proceedings of the 2014 ACM International Symposium on Wearable Computers, Seattle, WA, USA, 13–17 September 2014; pp. 91–98.
26. Lee, W.H.; Liu, X.; Shen, Y.; Jin, H.; Lee, R.B. Secure pick up: Implicit authentication when you start using the smartphone. In Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, Indianapolis, IN, USA, 21–23 June 2017; 2017; pp. 67–78.
27. Buriro, A.B.; Crispo, B.; Zhauniarovich, Y. Please hold on: Unobtrusive user authentication using smartphone's built-in sensors. In Proceedings of the 2017 IEEE International Conference on Identity, Security and Behavior Analysis, New Delhi, India, 22–24 February 2017; pp. 1–8.
28. Amirou, A.; Amirou, Y.; Ould-Abdeslam, D. S-Transform with a Compact Support Kernel and Classification Models Based Power Quality Recognition. *J. Electr. Eng. Technol.* **2022**, *17*, 2061–2070. [CrossRef]
29. Zhou, Z.H.; Li, M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 1529–1541. [CrossRef]
30. Wang, S.; Guo, Y.; Hua, W.; Liu, X.; Song, G.; Hou, B.; Jiao, L. Semi-supervised PolSAR image classification based on improved tri-training with a minimum spanning tree. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8583–8597. [CrossRef]
31. Emmagee. 2018. Available online: https://github.com/NetEase/Emmagee (accessed on 6 July 2023).