sensors

MDPI

Article

A Hybrid Deep Learning System for Real-World Mobile User Authentication Using Motion Sensors

Tiantian Zhu [1], Zhengqiu Weng [1,2], Guolang Chen [2,3,*] and Lei Fu [4]

1 College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China; ttzhu@zjut.edu.cn (T.Z.); derisweng@163.com (Z.W.)
2 Department of Information Technology, Wenzhou Polytechnics, Wenzhou 325035, China
3 School of Management, Zhejiang University, Hangzhou 310023, China
4 College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310023, China; fulei@zjut.edu.cn
* Correspondence: 2019f098@zju.edu.cn; Tel.: +86-571-8795-3851

Received: 22 May 2020; Accepted: 7 July 2020; Published: 11 July 2020

Abstract: With the popularity of smartphones and the development of hardware, mobile devices are widely used by people. To ensure availability and security, how to protect private data in mobile devices without disturbing users has become a key issue. Mobile user authentication methods based on motion sensors have been proposed by many works, but the existing methods have a series of problems such as poor de-noising ability, insufficient availability, and low coverage of feature extraction. Based on the shortcomings of existing methods, this paper proposes a hybrid deep learning system for complex real-world mobile authentication. The system includes: (1) a variational mode decomposition (VMD) based de-noising method to enhance the singular value of sensors, such as discontinuities and mutations, and increase the extraction range of the feature; (2) semi-supervised collaborative training (Tri-Training) methods to effectively deal with mislabeling problems in complex real-world situations; and (3) a combined convolutional neural network (CNN) and support vector machine (SVM) model for effective hybrid feature extraction and training. The training results under large-scale, real-world data show that the proposed system can achieve 95.01% authentication accuracy, and the effect is better than the existing frontier methods.

Keywords: mobile authentication; VMD; CNN; SVM; semi-supervised learning

1. Introduction

With the rapid development of mobile communication technology and the Internet, mobile devices came into being and entered people's lives. There are many types of mobile devices, including not only common smartphones, but also multifunctional smartwatches, fitness trackers, medical monitoring equipment, and augmented reality glasses. CCS Insight forecasts that 5G-enabled phones will reach sales of 210 million units in 2020, a tenfold increase compared with 2019. In 2024, sales of 5G phones are projected to hit 1.15 billion units, accounting for 58% of all mobile phones sold that year [1].

Mobile devices have various innovative functions, which can meet the requirements of personalized customization, thereby greatly improving the quality of user's daily lives. As the diversity of mobile device applications increases, more and more users store personal private information locally. To prevent illegal access to the private information stored on the mobile device, it is urgent to design a suitable and robust authentication mode to protect the user's privacy according to the mobile device's own hardware, software features, and application scenario characteristics. The evolution of mobile user authentication methods can be roughly divided into three stages: The first stage is represented by knowledge-based authentication methods, which requires users to explicitly enter authentication

information, such as passwords, patterns, etc. This method only verifies whether the user has correctly entered the account credentials created in advance, but cannot determine whether the user itself is trusted. In addition, since the screens of mobile devices are generally small, the experience of some user interaction interfaces for login and input is poor. For example, if the user sets a strong password containing more than 10 characters, it will take a long time for the user to enter a long password on the mobile device before unlocking or login, which leads to poor usability and unfriendly experience. In addition, previous work has shown that knowledge-based authentication methods are not only easy to be brutally cracked [2] but also vulnerable to a series of new attacks, such as stain attacks [3,4], shoulder peeping [5,6], and sensor inference [7–9]. The second stage takes fingerprint authentication and face recognition as the origin. Compared with the knowledge-based authentication mechanism, the static biometric authentication method based on fingerprint and face can achieve relatively high accuracy and better respond to multiple attacks. However, such biological information collection may cause users to worry about privacy and leakage of personal information. At the same time, the security of fingerprint authentication and face recognition has also been questioned by many cutting-edge works [10–13]. Although these two commonly used methods (knowledge-based authentication and static biometric-based authentication) have been in full swing for decades, there are a series of deficiencies in usability, security, and effectiveness. Therefore, in recent years, using motion sensors such as acceleration sensors, gyro sensors, and gravity sensors as data sources, a method of authentication based on the user's dynamic behavior, has been proposed by many researchers (e.g., gait authentication [14–31], user's usage behavior authentication [32–38], and daily life behavior authentication [39–51]). These methods have better user experience and stronger privacy protection capabilities, laying the foundation for the development of the third phase of mobile user authentication. The effective collection and analysis of motion sensor data have become a key factor that affects the authentication ability. The existing mobile user authentication methods based on motion sensor have a series of problems, such as weak feature extraction ability, poor de-noising ability, and low authentication accuracy of the owner, which need to be further improved, mainly manifested as follows:

(1) Motion sensor data collected from mobile phones in the real-world often contain noise. At present, most of the data collection of mobile authentication work is often in a relatively closed and stable experimental environment, such as a special data acquisition laboratory. There is no need to consider the impact of noise, but once deployed into the real-world, the availability is poor. Existing work based on large-scale real data [32] only filters the data lying on flat surfaces, but in the actual environment, there are many data unrelated to authentication, such as sensor signal discontinuities, mutation, etc., so the de-noising ability is limited. Previous work [52] has envisaged the use of recursive mode decomposition methods, such as empirical mode decomposition (EMD) [53] to de-noise, which adopts an adaptive method—through recursive steps the same decomposition method successively separates different modal components in the target signal. However, EMD decomposition is easily affected by singular points, such as signal discontinuities and mutations, which makes the corresponding extreme points change greatly, resulting in distortion of the calculated envelope. Instinct mode functions (IMF) components separated on this basis will be affected by singular points, causing the IMF components to contain the mode or abrupt abnormal features of adjacent components, thus introducing new noise.

(2) In a real complex environment, the sensor data collected often lacks a trusted label. This makes supervised learning [22,24,25,27,28,32,33,35,37,38,40–42,47–51] difficult to use, while unsupervised learning [31,46] has a delay in the training phase, although a semi-supervised algorithm has been proposed in frontier work [32] to improve the accuracy, the method only uses a single classifier (SVM), and its proposed initial classifier trained with some labeled samples is a weak classifier. If the previous iteration is used continuously to classify the unlabeled data, which will cause the continuous accumulation of its own errors, then it cannot guarantee that the resulting model has high accuracy.

(3) The representation ability of a feature extracted from sensor data in existing methods is weak. The data collected from the motion sensor generally appear as a time-series signal. First,

in some studies, dynamic time warping (DTW) [34] and Pearson correlation coefficient (PCC) [26] are used to directly compare the time series signal, which has some defects such as unstable signal period segmentation and fewer representative features. Second, if traditional machine learning methods are used for training, feature extraction needs to be performed initially on the original signal data. Most of the existing feature extraction methods are based on statistics and empirically extract a variety of time-domain and frequency-domain features, including average, standard deviation, maximum/minimum, etc. [22,24,25,27,28,31–33,35,37,38,40–42,46,47,51], which mainly rely on manual extraction and have limited ability to represent user behavior. Third, the existing sensor data modeling methods based on deep learning [29,30,43–45,48–50] are mostly based on the original signal data, with less consideration in signal enhancement, and a lack of deeper analysis of user behavior signal data.

In order to overcome these challenges, this paper proposes a hybrid deep learning system suitable for mobile authentication in the real-world. The system can implicitly collect motion sensor data and use VMD to de-noise and enhance the signals when users are using mobile phones, and then send them to the server for label refactoring using a cooperative semi-supervised algorithm, and then use CNN and SVM for feature extraction and model training, and finally send the trained model back to the mobile terminal for real-time authentication. Experimental results on large-scale real data show that the system can achieve satisfactory accuracy, surpassing the existing frontier work. In the actual use process, users can first use our system to perform implicit real-time user authentication when the user opens the application. If the authentication fails, further authentication methods such as fingerprints, faces, and passwords can be used to greatly improve mobile devices' availability and ease of use.

The main contributions of our works include the following aspects:

1. A heuristic signal enhancement method is proposed, which uses variational mode decomposition to decompose the signal, so as to eliminate the influence of noise and determine the optimal number of modes K by experiments.
2. The semi-supervised collaborative training method (Tri-Training) is used to de-noise, eliminating the influence of noise in the motion sensor data collected in the real-world, and providing high-quality data for the hybrid deep learning training model.
3. Using a hybrid deep learning method, the model structure includes CNN and SVM. CNN is used to extract the mixed features of each information component processed by VMD, and SVM is used for effective model training.
4. These methods are integrated to form a system. Data from training results under the large-scale real environment show that the system proposed in this paper can achieve 95.01% authentication accuracy, and the effect is better than the existing frontier methods.

The remainder of this article is organized as follows: Section 2 describes the background of our work. Section 3 covers system design in detail. Section 4 presents the overall evaluation of our system. Section 5 surveys the relevant work and Section 6 concludes our work.

2. Technical Background

2.1. Variational Mode Decomposition

VMD is not as susceptible to singular points in the signal as EMD; VMD is an adaptive and nonrecursive method that can analyze both nonstationary and nonlinear signals [54,55]. The essence of the VMD algorithm is the process of solving the variational problem. This process includes the construction and solution of the variational problem. The variational problem is expressed as follows: the original signal f can be decomposed into multiple IMF eigenmode functions, and it is assumed the bandwidth of each modal eigenfunction is limited and has different center frequencies. On this basis, the corresponding IMF eigenmode function is searched, and the cumulative sum of the bandwidth corresponding to each IMF is required to be minimized.

To obtain the bandwidth of each mode function, VMD first utilizes the Hilbert transform to convert each mode u_k into an analytical expression in a single-sided spectral domain:

$$u_k^+(t) = \left(\delta(t) + \frac{j}{\pi t}\right) * u_k(t) \tag{1}$$

After the Hilbert transformation, the frequency spectrum of each mode is shifted to the baseband and the corresponding estimated center frequency is adjusted by using an exponential tuned term. Then, the bandwidth is estimated according to the Gaussian smoothness of the demodulated signal by utilizing the squared L2-norm of the gradient [56]. Thus, the VMD process is realized by solving a constrained variational problem [57] and the model is expressed as follows:

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_{k=1}^{K} \left\| \partial \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\}, \ subject\ to \sum_{k=1}^{K} u_k(t) = f(t) \tag{2}$$

where $f(t)$ is the target signal, $\{u_k\} = \{u_1, \ldots, u_K\}$ represents the set of the decomposed modes, and $\{\omega_k\}$: $= \{\omega_1, \ldots, \omega_K\}$ represents the respective center frequencies. Then, replace the solution of the constrained variational problem with the optimal solution of the unconstrained variational problem via a quadratic penalty term and Lagrangian multipliers:

$$\mathcal{L}(\{u_k\},\{\omega_k\},\lambda) = \alpha \sum_{k=1}^{K} \left\| \partial \left[\left(\delta(t) + \tfrac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 + \left\| f(t) - \sum_{k=1}^{K} u_k(t) \right\|_2^2 + \left\langle \lambda(t), f(t) - \sum_{k=1}^{K} u_k(t) \right\rangle \tag{3}$$

To solve the original minimization problem, the alternate direction method of multipliers (ADMM) is adopted to determine the saddle point of the augmented Lagrangian in a sequence of iterative suboptimizations. By this process, all the mode functions are obtained and updated by Wiener filtering to tune the center frequency in the spectral domain:

$$\begin{aligned} \hat{u}_k^{n+1}(\omega) &= \frac{\hat{f}(\omega) - \sum_{i<k} \hat{u}_i^{n+1}(\omega) - \sum_{i>k} \hat{u}_i^{n}(\omega) + \frac{\hat{\lambda}^n(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k^n)^2} \quad (\omega > 0) \\ \omega_k^{n+1} &= \frac{\int_0^\infty \omega \left| \hat{u}_k^{n+1}(\omega) \right|^2 d\omega}{\int_0^\infty \left| \hat{u}_k^{n+1}(\omega) \right|^2 d\omega} \end{aligned} \tag{4}$$

The center frequency ${\omega_k}^n$ is calculated from the weighted center of each mode in the spectral domain; the center represents the frequency of the least squares linear regression of the instantaneous phase.

2.2. Semi-Supervised Training

2.2.1. Co-Training

Semi-supervised learning can learn from a small amount of labeled sample data and a large number of unlabeled data, thereby improving data utilization and system performance, and reducing training costs. Collaborative Training (Co-Training) has been a popular semi-supervised learning algorithm since it was proposed by Blum et al. [58], and has become a research hotspot in the field of machine learning and pattern recognition. In the process of Co-Training, it is often necessary to establish two or more classifiers for effective cooperation. In the process of two classifiers Co-Training, the new labeled data generated by a single classifier will be used as labeled data to enter the next iteration training process of another classifier.

The basic idea of Co-Training follows. Determine two fully redundant and independent feature sets F1 and F2 on the entire data set D; these two features can be classified independently and effectively. Record the labeled parts of the data set as L and the unlabeled part as U. During training, the first step is to train two classifiers C1 and C2 on F1 and F2 of L, respectively. In the second step, u samples are randomly selected from the unlabeled data set U and put into the set U′, and all elements in U′ are

labeled with C1 and C2. In the third step, m positive samples and n negative samples with the highest reliability are taken from the results of the two classifiers and put into L. In the fourth step, 2p + 2n data are selected from U and added into U'. Repeat the third and fourth steps until the number of samples reaches the required size, and the resulting L is a large number of labeled sample sets.

2.2.2. Tri-Training

In general, it is difficult for data set D to meet the requirement of having two fully redundant and independent feature sets. Zhou et al. proposed the Tri-Training strategy [59], which requires neither fully redundant and independent feature sets, nor the use of different classification algorithms. Tri-Training ensures the difference between the classifiers by training on the differential data subsets extracted from the original data set.

The basic idea of Tri-Training follows. First, the labeled data set L can be repeated by sampling (bootstrap sampling) to obtain three labeled training sets L1, L2, and L3, and then generate a classifier from each training set, recorded as C1, C2, and C3. These three classifiers are used to generate pseudolabeled samples in the form of "minority obeying majority." For example, if C1 and C2 predict a certain unlabeled sample s as a positive class and C3 predicts it as a negative class, then s is provided as a pseudolabeled positive sample to C3 for learning. Specifically, if two classifiers have the same prediction for the same unlabeled sample, the sample is considered to have higher label confidence and is added to the labeled training set of the third classifier after labeling.

However, when used in the real-world, the prediction results of the classifier may be wrong. At this time, from the perspective of "minority" classifiers, it received samples with "marked noise". Based on the classification noise process introduced by Angluin and Laird [60], Zhou et al. [59] have derived the condition in the form of "minority obeying majority". In each round of learning, we only need to judge whether the condition is true, and then we can decide whether to update the classifier based on the pseudolabel samples. Intuitively, this result shows that the negative impact of the introduced noise can be offset by the benefits of using a large number of unlabeled examples, and the accumulated marked noise under certain conditions can be compensated by using a large number of unlabeled data.

3. System Design

This section mainly introduces the design process of the system. First, we describe the overall framework of the system. Second, we introduce the data collection strategy and method. Third, we introduce how to use VMD to decompose and de-noise the signal. Fourth, we introduce the use of Tri-Training for label refactoring. Fifth, we describe the hybrid training architecture and method based on CNN and SVM, including how to conduct real-time authentication.

3.1. System Framework

The overall framework of the system is shown in Figure 1, which shows that simpler calculation tasks, such as data collection and data de-noising, are performed on the mobile phone, while tasks with relatively complex calculations, such as label refactoring and online hybrid deep learning model training, are performed on the server. In addition, in order to ensure the real-time authentication, the final training model (including CNN and SVM) will be pushed to the local for offline real-time authentication, to ensure that the smartphone can be used normally when the network signal is poor or there is no network.

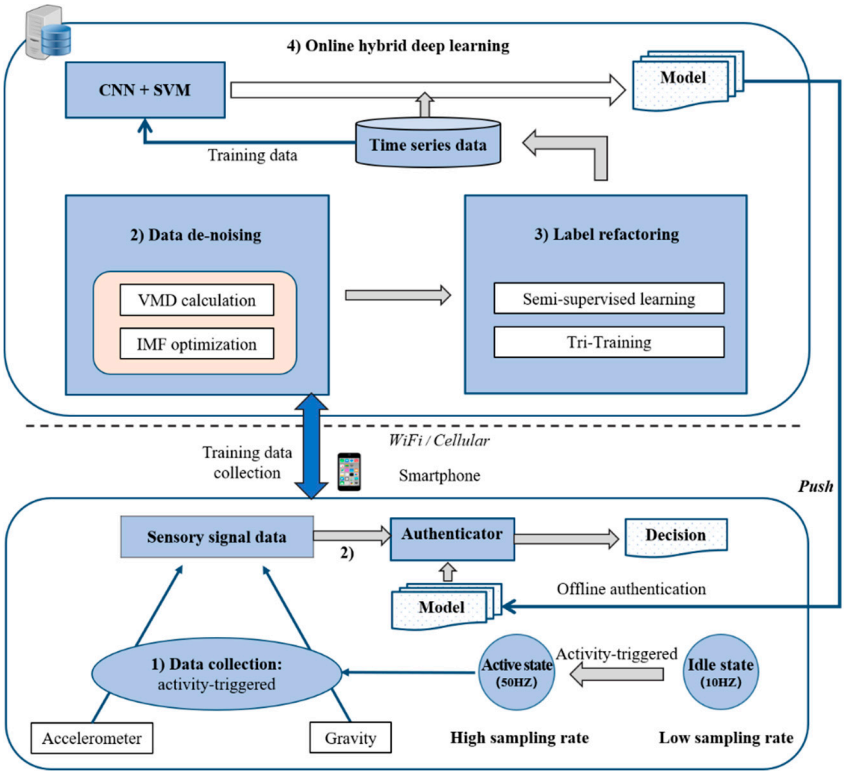

Figure 1. System architecture, including activity-triggered data collection in the client side, data de-noising, and label refactoring online learning with the enhanced semi-supervised model on the server side (convolutional neural network (CNN) and support vector machine (SVM)).

In the training phase, when the user first opens the mobile application (this behavior can represent a user's behavior patterns [32]), the acceleration and gravity sensor three-axis values of the smartphone are collected as the original time series signal data. Among them, acceleration sensors are used for authentication, and gravity sensors are used for data de-noising. Then, these data will be uploaded to the server via WiFi or cellular network for de-noising and signal enhancement. Then, the data after decomposition and de-noising will be sent to the label refactoring module for labeling operation. Eventually, the labeled data will be fed into the hybrid deep learning architecture based on CNN and SVM for model training, and the optimal model suitable for real-time authentication of the host will be obtained.

In the real-time authentication phase, the motion sensor data collected will be locally de-noised and the signal enhanced, and the resulting output signal data will be sent to the fully trained CNN architecture to extract features, and the optimized SVM model will be used for real-time authentication.

3.2. Data Collection

Data collection is the first step in mobile authentication. What data to collect and how to collect data are the key factors that affect the final authentication accuracy. Previous research work [32] has proposed a method for real-time mobile authentication without user consciousness, which mentions how to effectively collect motion sensor data representing the user's habits in the process of using mobile phones. Inspired by [32], this paper collects data when the user opens the mobile application.

We develop an Android-based application and start the background service to monitor the application running in the foreground in real-time. When the following two requirements are met, data collection will be started. First, the screen of the smartphone is on. Second, the application in the foreground is switched. These two conditions indicate that the user is opening a certain application and using the mobile phone, which was proved to be a valid user mode for mobile authentication

in [32]. In the actual collection process, the data collection time lasts 3 s. If the user uses the mobile phone for less than 3 s (the mobile phone screen is off during the collection process), the collection is terminated.

When judging the application of the foreground, Android provides two APIs: getRunningAppProcesses() and getRunningTasks(), to retrieve the current application running in the foreground. However, starting with Android 5.0, those APIs are deprecated. To preserve the portability of our system on the fragmented Android devices, we invoke the system command-line tool "ps" and implement a parser to map the ID of a running application (i.e., pid) to the application name on Android 5.0 and Android 6.0. Since Android 7.0, Android has locked down the permissions of/proc, and we cannot get the running process via ps. Instead, the list of running apps can be alternatively fetched by using UsageStatsManager [61] on Android 7.0+. Our implementation allows us to intercept the active applications properly on all the existing versions of Android. The summary of our implementation is shown in Table 1.

Table 1. Android version and corresponding method for obtaining foreground applications.

SDK Version (V)	Our Methods
V < 21 (Android 5.0-)	GetRunningTasks (permission.GET_TASKS)
21 ≤ V ≤ 23 (Android 5.0 and Android 6.0)	Ps or UsageStatsManager
V ≥ 24 (Android 7.0 +)	UsageStatsManager (android.permission.PACKAGE_USAGE_STATS)

In order to save power, the services running in the Android background will collect and judge data under different frequencies, which can be divided into the idle state and the active state. Generally, the background data sampling frequency is idle, which is 10 Hz. When these two data collection conditions are met at the same time, the data collection frequency will jump from the idle state to the active state, and the sampling rate is 50 Hz.

3.3. Data De-Noising

Data de-noising includes two aspects. One is to remove sensor data (invalid data) that is irrelevant to the user's mobile phone habits. The other is to eliminate the effects of singular values, such as signal discontinuities and mutations in motion sensors.

The first type is invalid data that cannot represent the users' usage behavior, and this part of the data is not related to our authentication. When collecting data in actual scenarios, it is usually impossible to guarantee that the user always holds the mobile phone in daily use. For example, users can place the device on a desktop for interactive operation. In this case, even if the two collection requirements are met (the mobile phone screen is on and a new application is running in the foreground), the data collected still cannot effectively reflect the differences in usage patterns between different users. By asking 20 volunteers to pick up the phone and place it on a fixed surface, we get the boundaries of the gravity sensor readings on three dimensions by minimizing the errors of device placement prediction; if the readings of gravity sensor meet $-1.5 < X_{gr}(k) < 1.5$, $-1.5 < Y_{gr}(k) < 1.5$, and $9 < |Z_{gr}(k)| < 10$ simultaneously, we regard it as invalid data and remove it. Here, $X_{gr}(k)$, $Y_{gr}(k)$, and $Z_{gr}(k)$ represent values of the three axes of the gravity sensor x, y, and z at time k.

In the second type, for the recognition of different usage patterns and authentication of different mobile users, it is essential to utilize a classifier to analyze the signal obtained from the current sensors. However, it is difficult to apply the time-series signal directly for common classification methods since it may contain noise in the real-world scenario.

In order to carry out subsequent effective feature extraction (including statistical feature extraction during the label refactoring stage and CNN feature extraction during hybrid deep learning training), we need to de-noise and enhance the data first. VMD is utilized to decompose the signal obtained into several IMFs for potential feature extraction, as described in previous work [54]. The submode decomposed by VMD contains a specific spectrum, which can accurately trace the signal changes. Thus, signal decomposition can effectively eliminate the impact of noise and separate useful components in

high-level modes. In VMD, the decomposition results depend on the mode number, which is defined as K. Another parameter α represents the data-fidelity constraint, which influences the tightness of the band-limits. It is also important in the decomposition. Parameter α is selected as 2500, based on the experimental trial. The value of K will be given through detailed experiments described in Section 4.

3.4. Label Refactoring

The motion sensor data collected from the smartphone in the real-world may not be the data used by oneself. For example: In a class, Bob may lend his mobile phone to Alice. If it happens to be in the training stage of Bob's model, Alice's usage habits will become a mislabeling effect and enter the classifier, and ultimately affect Bob's model accuracy. In this regard, after obtaining de-noising and enhanced signal data, we use Tri-Training for label refactoring to eliminate the impact of other (mislabeled) data on the Bob model as much as possible.

We use three different classifiers (decision tree, logistic regression, naive Bayes) to carry out label refactoring comprehensively. The specific steps follow:

(1) Feature extraction. Since the inputs of these three classifiers cannot be original time series data, it is necessary to segment the IMFs signal first and extract the features of each segment. During segmentation, we use a sliding window to slice the data into 0.2 s, and 50% overlap each slice front and back, as shown in Figure 2, (the front and back windows are shown by the dotted lines in the figure have a 0.1 s overlap).

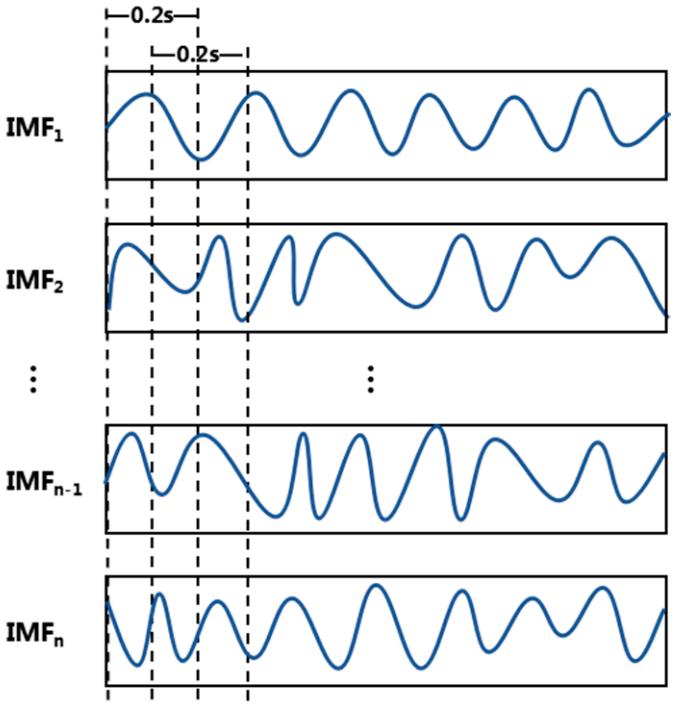

Figure 2. Schematic diagram of the sliding window (instinct mode functions (IMF)).

After that, we integrated the previous work [22,24,25,27,28,31–33,35,40–42,46] and selected nine types of features, including RMS, standard deviation, variance, range, skewness, kurtosis, mean, average deviation, and permutation entropy. We denote the ith value of the feature vector as $\mathcal{F}_i = \{F_{1i}, F_{2i}, \ldots, F_{pi}\}$, which includes P features, with feature extraction of 0.2 s data in each sliding window. All the features and the corresponding expressions are listed in Table 2. For each IMF, we get 29 feature vectors ($3/0.1 - 1 = 29$) because we utilize the sliding window design with 50% overlap.

Table 2. Statistical parameters for feature extraction.

No.	Statistical Feature	Expression
1	RMS	$\sqrt{\frac{1}{K}\sum_{k=1}^{K}[x(k)]^2}$
2	Mean	$\sum_{k=1}^{K}[x(k)]/K$
3	Standard Deviation	$\sqrt{\sum_{k=1}^{K}[x(k)-\overline{x}]^2/(K-1)}$
4	Variance	$\sum_{k=1}^{K}[x(k)-\overline{x}]^2/K$
5	Range	$\max(x(k))-\min(x(k))$
6	Kurtosis	$\sum_{k=1}^{K}[(x(k)-\overline{x})/\sigma]^4/K$
7	Skewness	$\sum_{k=1}^{K}[(x(k)-\overline{x})/\sigma]^3/K$
8	Average Deviation	$\sum_{k=1}^{K}\left\|x(k)-\overline{x}\right\|/K$
9	Permutation Entropy	$-\sum_{i=1}^{m!}p(\pi_i)\log(p(\pi_i))$

(2) Tri-Training. In the Tri-Training phase, the original data set D (10-day data set, which will be explained in Section 4) is divided into three parts: labeled data set L, potential mislabeled data U, and verification data set V. We assume that the person who uses the mobile phone at the beginning is himself, so we put the data of the first two days into L, and the data in the middle six days may contain other people's mislabels and put them into U. In the last two days, the data representing the user's own test data are put into V.

We perform bootstrap sampling on the labeled data set L to obtain three labeled training sets Lc, Ll, and Ln, each set contains one-third of the total L. Then, we generate a classifier from each training set, denoted as Cd, Cl, and Cn. These three classifiers are used to generate pseudolabeled samples in the form of "minority obeying majority". For example, if Cd and Cl predict a certain unlabeled sample s as a positive class, and Cn predicts it as a negative class, then s is provided as a pseudolabeled positive sample to Cn for learning. Specifically, if two classifiers have the same prediction for the same unlabeled sample, the sample is considered to have higher label confidence and is added to the labeled training set of the third classifier after labeling. At the same time, we use the conditions required by the "minority obeying majority" [59] to eliminate the classification error. Due to a large number of samples in this paper, the negative effects of the noise can be offset to some extent [59]. Finally, we take the intersection of Lc, Ll, and Ln as the final labeled training data set L'.

Here, we choose the binary classification, which can better represent the different mobile device usage patterns of the device owner and others. In the process of using decision tree, logistic regression, and naive Bayes classifiers, we need to obtain a certain proportion of other people's data as a counterexample. In this paper, we use the strategy of strategic sampling in [32] to extract the most representative counterexample data samples from the massive set of others' data.

3.5. Hybrid Training Based on CNN and SVM

The hybrid training based on CNN and SVM mainly includes two parts. The first part uses CNN architecture for feature extraction. The second part uses SVM to classify the extracted features.

The structure of CNN is shown in Figure 3. The structure and description of each layer follow:

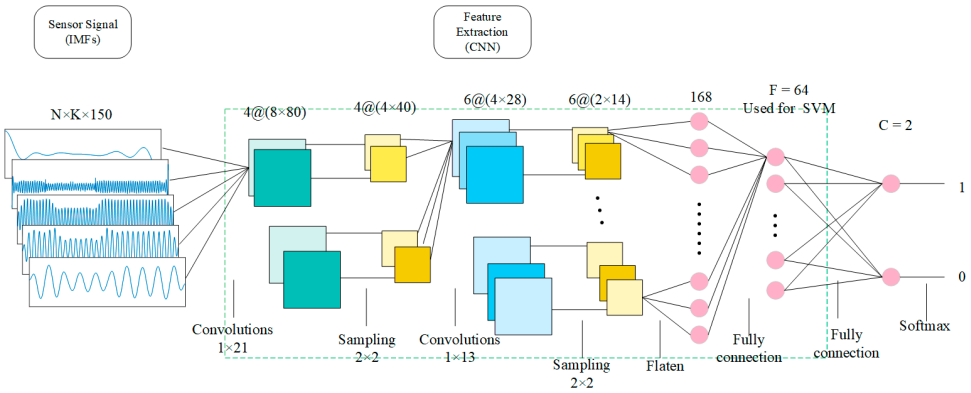

Figure 3. CNN extraction structure (used to extract features).

Input layer. The IMF signal obtained by VMD decomposition and label refactoring will be used as the input of CNN. Since the final output is label 1 (representing owner) or label 0 (representing others), CNN's input must contain both owner's data and others' data, where K represents the mode number determined by VMD and N represents the IMF number (including owner and others).

Convolutional layer. The parameters of this layer are obtained by tuning. The convolutional layer implements a one-dimensional kernel (1×21 samples and 1×13), performing filtering of the input and processing each input vector separately. The activation functions are Relu.

Sampling layer (pooling layer). The parameters of this layer are obtained by tuning. Maximum pooling is applied to the output of convolutional layers to further reduce their dimensionality and increase the spatial invariance of features.

Tiled layer. The neuron is tiled to connect to the next fully connected layer.

Fully connected layer. The parameters of this layer are obtained by tuning. In the fully connected layer, each output neuron of the last layer is connected to all input neurons of this layer (weights are not shared).

Softmax layer. The classification results of the training stage were obtained.

After training the parameters of the CNN network, each time a set of data representing a user's behavior ($K \times 150$) is input, 64 feature values can be obtained for SVM training.

After the training, the CNN network parameters and SVM model will be pushed to the mobile phone. During the authentication phase, every three seconds of the user's behavior patterns are collected, the invalid data will be filtered by the de-noising method, and the VMD is used to eliminate the effects of noise. Then it is fed into the CNN architecture to obtain a feature vector composed of 64 features. Finally, it is sent to the SVM model representing the owner for real-time authentication. By setting the classification threshold θ, if the output is greater than θ, it is the owner; otherwise, it is others.

4. Experimental Evaluations

4.1. Data Set

In this paper, we have two data sets. The first data set is used to evaluate the overall performance of our system, and the second data set is used for verifying how many training samples are enough for model training. For the first data set, we collected large-scale unlabeled data in a real unsupervised environment, which comes directly from an Internet company with millions of users in China. For ethical considerations, all participants were actively informed of the purpose of data collection, and given the option to opt-in or opt-out. In the end, we collected data from 1513 users using 10 days of raw data. In this paper, a total of 283,133,354 pieces of raw sensor data were collected. After data preprocessing, 283,006,659 pieces were found to be valid, and the average number of effective records per user was 187,050 (about 37,409 samples, $187{,}050/50/0.1 - 1 = 37{,}409$).

In the label refactoring phase, decision tree, logistic regression, and naive Bayes classifiers are used to divide the labeled data set into a training set and a test set during training, and the ratio is 4:1. The ratio of positive samples to negative samples is 1:5, and the data volume of positive samples and negative samples in the test stage is also extracted according to the same ratio distribution.

In the hybrid training phase based on CNN and SVM, we divide the labeled data set L′ into a training set and a test set after Tri-Training, with a ratio of 4:1. The ratio of positive samples to negative samples is 1:5, and the data volume of positive samples and negative samples in the test stage is also extracted according to the same ratio distribution.

For the second data set, we experimented with 106 people based on the data collected within one week from another large Internet company with labels, and the average number of effective samples per user ranges from 1024 to 20,132, depending on the frequency of the usage.

For all the data sets, the data collection frequency is 50 Hz, and each data collection lasts 3 s. In order to uniquely mark each user, we will acquire the international mobile equipment identity (IMEI) while collecting data and use IMEI to distinguish different users on the server side.

4.2. Evaluation Index

In terms of accuracy, we define the following evaluation matrix. True positive (TP), the owner is accurately marked. False positive (FP), others are marked as owner. True negative (TN), others are accurately marked. False negative (FN), the owner is marked as others. In terms of classification accuracy, the following indicators are used in this chapter:

$$\text{Powner} = \text{TP}/(\text{TP} + \text{FP})$$

$$\text{Rowner} = \text{TPR} = \text{TP}/(\text{TP} + \text{FP})$$

$$\text{Fowner} = 2 * \text{Powner} * \text{Rowner}/(\text{Powner} + \text{Rowner})$$

$$\text{Pother} = \text{TN}/(\text{TN} + \text{FN})$$

$$\text{Rother} = \text{TN}/(\text{TN} + \text{FP})$$

$$\text{Fother} = 2 * \text{Pother} * \text{Rother}/(\text{Pother} + \text{Rother})$$

$$\text{FPR} = \text{FP}/(\text{TN} + \text{FP})$$

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{FN} + \text{TN})$$

Powner, Rowner, and Fowner represent the precision, recall, and F1-score of user owners, respectively. Pother, Rother, and Fother represent the precision, recall, and F1-score of other users, respectively. In addition, we use the receiver operating characteristic (ROC) curve to represent the overall authentication performance of the system. The ROC curve shows the relationship between the true positive rate (TPR) and false positive rate (FPR) at different classification thresholds θ. In particular, the area under curve (AUC) is used to represent the area enclosed by the ROC curve and the coordinate axis.

4.3. VMD Effect Analysis

To deploy VMD for data de-noising and signal enhancement, the most important thing is to calculate the decomposed number K via experiment. Figure 4a–d represent a representative acceleration sample of different decomposition results of VMD when K = 3, 4, 5, and 6, respectively. According to the experience of the existing work [54], we use the permutation entropy of each IMF to judge the quality of the decomposition results. We set the threshold as 0.6 as [54] did. It can be observed that the sensor signals in Figure 4a–c are under-decomposed. Although the decomposition results are useful, they are still far from the target threshold. When K = 6, the first five IMFs are valid for our classification, while the last one can be considered as noise, which totally meets the requirements for

data de-noising and signal enhancement. Finally, for each sensor signal, we choose K = 6 and select the first five IMFs as valid ones.

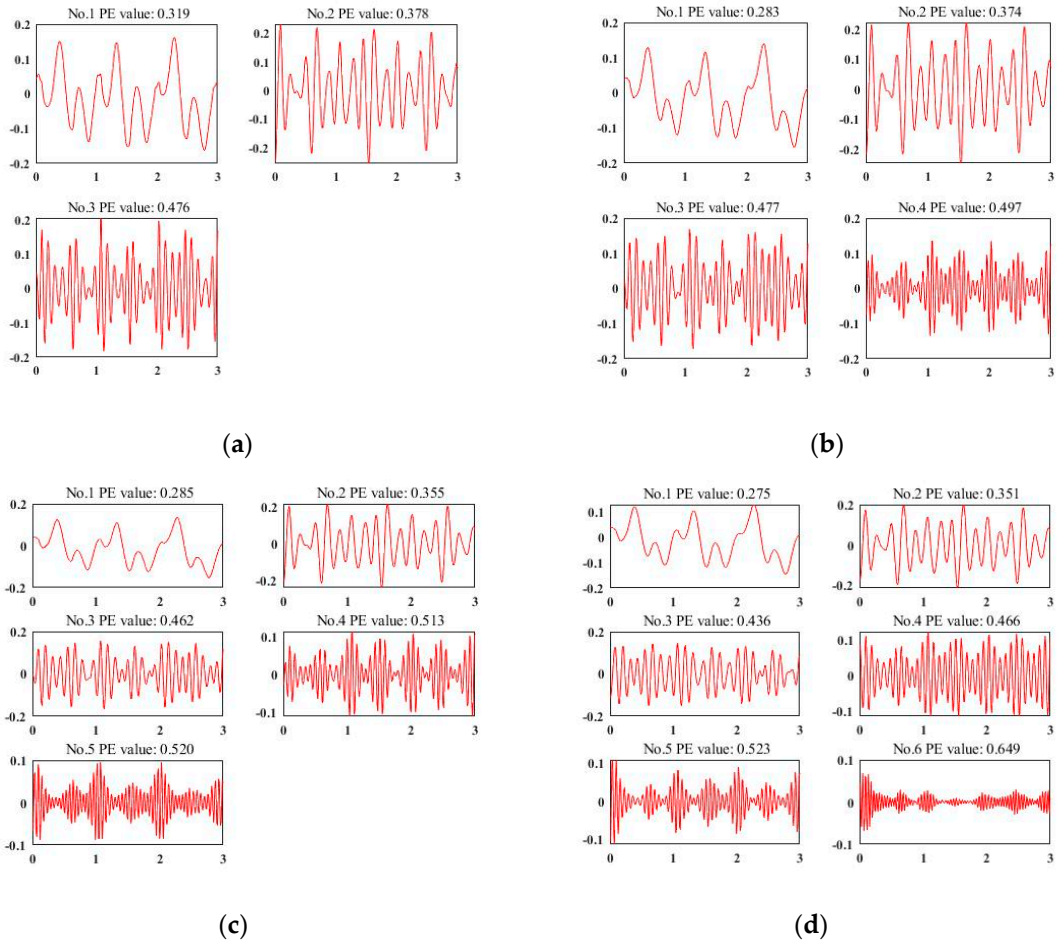

Figure 4. Sensor signal decomposition results by variational mode decomposition (VMD). (**a**) K = 3 by VMD; (**b**) K = 4 by VMD; (**c**) K = 5 by VMD; (**d**) K = 6 by VMD.

We also compare our VMD method to another famous signal decomposition method [62], called wavelet packet decomposition (WPD). Figure 5a,b presents the decomposition results by WPD when the decomposition level is 2 and 3, respectively. When K = 2, only the first decomposition result is valid (the value of permutation entropy is less than the threshold 0.6). Also, when K = 3, the valid decomposition result is less than that of VMD. VMD can perform better even if the decomposed number K is not suitable for decomposition.

4.4. Model Tuning

In order to optimize the final SVM model, we conduct the grid search to find the optimal SVM training parameters. In SVM, the two main parameters that need to be determined are C and γ, where the C is the penalty coefficient. If C becomes larger, it means that the punishment becomes larger, resulting in a situation where the model is not flexible enough and the generalization ability becomes weak. Conversely, if it becomes smaller, the punishment strength will become smaller, and the model is prone to underfitting. Another parameter γ determines the distribution of the data mapped to the new feature space. Similar to parameter C, γ that is too large will cause the model not to learn the content of the vector, but only remember the support vector itself, which leads to the reduction of generalization ability and prone to overfitting. If γ is too small, the data distribution in the new feature space will be too smooth, resulting in underfitting. In this paper, the classification threshold

θ is set to 0.5, the SVM convergence coefficient is fixed at 0.01, and the parameter γ is fixed at 0.01. The changes in the parameters and the final classification accuracy of the model are shown in Figure 6. The classification effect is the best when the value of the parameter C is 0.6. Similarly, when the SVM convergence coefficient is fixed at 0.01 and the parameter C is fixed at 0.6, the changes in the parameters and the final classification accuracy of the model are shown in Figure 6. It can be seen from the figure that when the value of the parameter γ is 0.06, the classification effect is the best. In summary, C = 0.6 and γ = 0.06 are used in this paper.

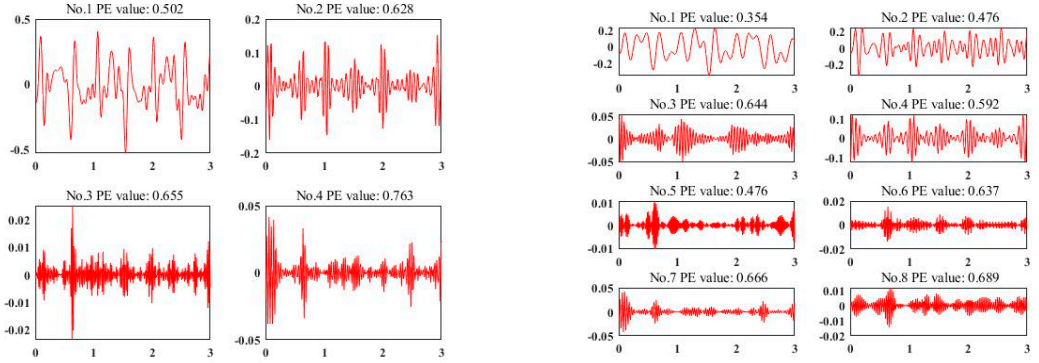

Figure 5. Sensor signal decomposition results by wavelet packet decomposition (WPD). (**a**) The decomposition level is 2 by WPD; (**b**) the decomposition level is 3 by WPD.

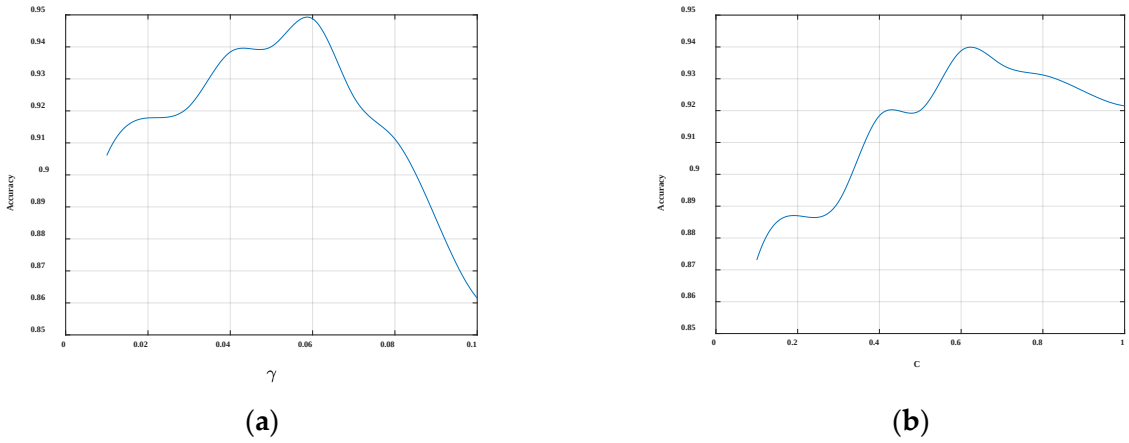

Figure 6. (**a**) The relationship between the parameter γ and the final classification accuracy of the model. (**b**) The relationship between the parameter C and the final classification accuracy of the model.

4.5. Overall Accuracy

Under the optimal parameters, we did 10 independent repeated experiments, and the final average values of TP, FP, TN, and FN are shown in Table 3.

As can be seen from Table 3, after the label refactoring, most of the mislabeling were corrected, and the final authentication results were significantly improved. By deploying VDM, the average Powner, Rowner, Fowner, Pother, Rother, and Fother increased from 91.04%, 72.10%, 80.47%, 70.15%, 92.33%, and 79.73%, to 95.02%, 74.32%, 83.40%, 76.28%, 97.01%, and 85.41%, respectively. We also evaluate the system accuracy comparison with and without using VMD de-noising to justify the effectiveness of VMD. The results show that the IMFs decomposed by VMD can accurately trace the signal changes. Our system (labeled with VMD) reaches a final classification accuracy value of 95.01%.

Table 3. The average classification results of 10 independent repeated experiments on 1513 users, including a comparison of accuracy with Tri-Training (labeled) and without Tri-Training (unlabeled).

Categories	Powner	Rowner	Fowner	Pother	Rother	Fother
Unlabeled without VMD	89.08%	70.35%	78.61%	68.42%	90.57%	77.95%
Labeled without VMD	93.35%	72.50%	81.61%	72.91%	94.89%	82.46%
Unlabeled with VMD	91.04%	72.10%	80.47%	70.15%	92.33%	79.73%
Labeled with VMD	95.02%	74.32%	83.40%	76.28%	97.01%	85.41%

As mentioned above, the ratio of positive samples to negative samples in the training set is 1:5, which shows that the trained classifier can accurately represent the mobile phone usage patterns of nonowner users, but may lose some patterns of the owner user. Considering that FP (that is, a malicious user can normally use the mobile phone of the owner user) in the authentication system is often more critical than FN (the owner user is mistaken for a malicious user), this paper tends to reduce the FP when configuring the authentication parameters. In practice, the classification threshold can be adjusted. For example, when the user operates an application with higher security, the threshold can be increased, which can improve the detection rate of malicious use of the owner's mobile phone against others.

In order to obtain the ROC curve, we set the threshold value θ from 0 to 1, using a step size of 0.01. Under different classification thresholds θ, the average values of TPR and FPR of all users are calculated, and the resulting ROC curve is shown in Figure 7. The resulting AUC value is 0.9340. From a security point of view, an increase in FPR will cause a greater potential threat to the owner user. From the usability point of view, a reduction in TPR will lead to a reduction in user experience. Due to the high AUC, the system has sufficient optimization space to balance the security and availability in actual use.

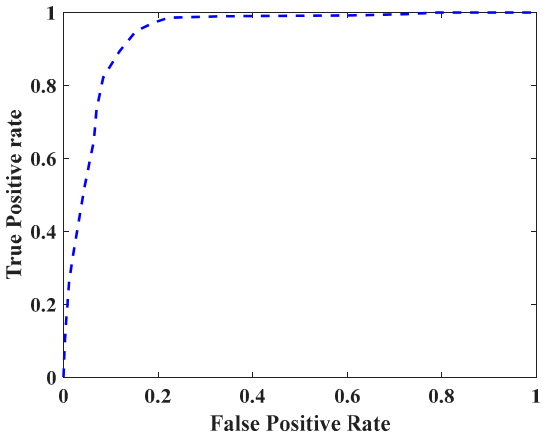

Figure 7. Receiver operating characteristic (ROC) curve of the system.

This paper investigates the most advanced research on motion sensor-based mobile authentication in the past five years and uses large-scale data sets to test its proposed classification methods. Table 4 shows a comparison of the accuracy for the five different methods in the existing work and the methods employed in this paper.

Table 4. Comparison of our work with other related work.

Study	Classifier	Accuracy
Our work	CNN + SVM (binary-class)	95.01%
Zhu et al. (2019) [32]	SVM (binary-class)	94.67%
Shen et al. (2018) [42]	HMM	90.54%
Buriro et al. (2017) [35]	Random forest	92.36%
Lee et al. (2017) [34]	DTW	86.49%
Zdeňka (2016) [41]	SVM (one-class)	86.51%

Hidden Markov Model (HMM), Dynamic Time Warping (DTW).

As can be seen from the table above, the accuracy of the authentication system in this paper is higher than other methods [32,34,35,41,42]. Among them, we also use the binary-class SVM for training, and the reason why the method proposed in this paper is better than [32] is that CNN is used for feature extraction, which has a wider coverage and more representative of the user's patterns than that of the manually extracted signs in [32]. Interestingly, the one-class SVM performs worse than the binary-class SVM used in this article. The main reason is that the characteristics of different users' mobile phone behaviors may be similar in multidimensional space, and the boundary of this approximate behavior cannot be well represented by one-class SVM.

Also, it is necessary to evaluate the training sample size's impact on system accuracy since it is possible that not every user would have enough training sample to begin with. We experimented with 106 people based on the data collected within seven days. The classification model is constructed for each individual in the first five days, and the average number of effective samples per user ranges from 1024 to 20,132, depending on the frequency of the usage. The data from the last two days are used for testing. The accuracy for each model is evaluated in Figure 8, and it shows that the accuracy has a strong relation to the sample size of positive instances. When the sample from the authorized users is insufficient, less than 4000, the performance of our classification is less satisfied with low accuracy. However, it improves drastically when the sample size increases. We also notice that once the sample size exceeds a threshold of 4000, accuracy will not improve by simply adding more samples. Since we have 37,409 data samples per user in the 1513 data set, the sample size ($37409 \times 0.2 = 7482 > 4000$) is quite sufficient in our experiment.

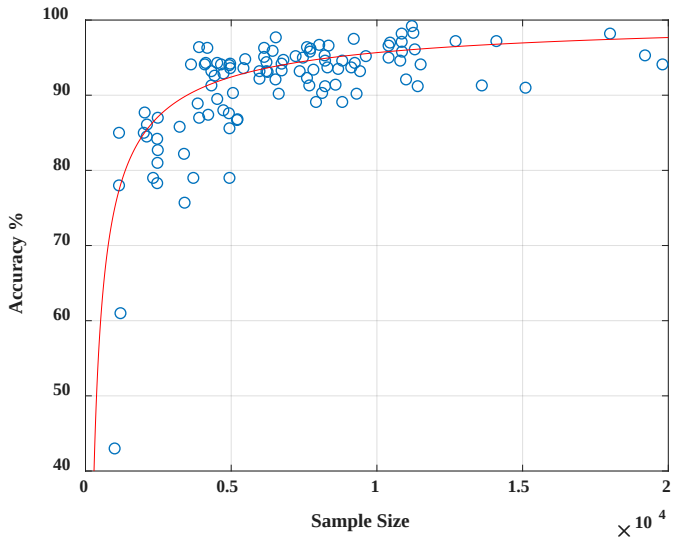

Figure 8. Sample size vs. accuracy showing that classification accuracy improves drastically when the sample size (0–4000) increases.

4.6. Anti-Attack Ability

In this section, we mainly discuss the impact of imitation attacks on our system in order to verify the robustness of our system in real-time mobile user authentication. Imitation attack refers to an attacker bypassing the authentication mechanism by observing the equipment usage pattern of the owner user and imitating the behavior of the owner user.

We invited 20 participants to conduct related experiments. First, select a model that one participant trains to represent mobile owner behavior patterns. The remaining 19 participants call the authentication module 100 times by observing the way the first participant uses the mobile phone and imitating his holding gesture. Among them, 29 data samples are generated for each imitation, and these samples are sent to the model for verification. After calculation and averaging, the probability that the mobile authentication system can successfully resist imitation attacks is 99.90%.

4.7. Overhead

In order to meet the real-time authentication, we conducted real-time detection of energy consumption tests, including mobile phone battery consumption, CPU usage, memory usage, and real-time authentication delay.

On the client side, we use the open source tool Emmage [63] to test the battery consumption, CPU usage, and memory usage of the mobile phone while the application is running. Table 5 shows the test results. In terms of battery consumption, this paper allows a participant to continue using the authentication service on a smartphone for three hours, including data collection and offline real-time authentication. On three different Xiaomi phones (MI 8), the battery consumption per hour is about 1% of the total power of the mobile phone, because users do not frequently open or switch applications under normal circumstances.

Table 5. Test results of client battery consumption.

Phone No.	Battery Consumption (mAh)	Data Collection		Authentication	
		CPU (%)	Memory (MB)	CPU (%)	Memory (MB)
Xiaomi No.1	146.58/3000	1.28	14.02	9.34	22.52
Xiaomi No.2	151.01/3000	1.31	14.02	9.58	22.68
Xiaomi No.3	149.31/3000	1.29	14.03	9.40	23.08

In addition, we also test the time-consuming authentication of the client. The following tasks are carried out on the above three smartphones (MI 8): data collection, data de-noising, feature extraction, and authentication, repeated 1000 times. The average time consumption of each stage is listed in Table 6. It can be seen that the entire process is completed in 3201.08 ms. Data collection takes up most of the time, and the time for data preprocessing, feature extraction, and authentication can be completed in about 20 ms, which meets the real-time authentication needs in real scenarios.

Table 6. Time spent on client authentication.

Procedure	Times (ms)
Data collection	3037.20
Data de-noising	17.31
Feature extraction	135.05
Authentication	11.52
Overall	3201.08

5. Related Work

In recent years, there has been much work [14–51] using the motion sensor for mobile authentication. However, the existing methods are more or less flawed. In this section, we will explain the existing

methods and highlight the innovation of our work from the aspects of de-noising ability, availability, and feature extraction coverage.

De-noising ability. Most of the previous user authentication studies [22–26,33–42,47–51] considered motion sensors ideally error-free during data collection and they had never taken the noise impact of the hardware into account, which would lead to the difficulty in fitting the model and thus affecting prediction accuracy. Existing work based on large-scale real data [32] only filters the data lying on a flat surface, but in the actual environment, there are many data that are not related to authentication, such as sensor signal discontinuities and mutations, so the de-noising ability is limited. [52] envisaged the use of recursive modal decomposition methods such as EMD [53] de-noising, which adopts the adaptive method by recursion of the same decomposition method, successively separating different modal components in the target signal. However, EMD is easily affected by singular points such as signal discontinuities and mutations, which makes the corresponding extreme points change greatly, resulting in distortion of the calculated envelope. The IMF components separated on this basis will be affected by singular points, causing the IMF components to contain the modal or abrupt abnormal features of adjacent components, thus introducing new noise. Different from the above work, this paper uses the VMD algorithm for effectively de-noising, and through experimental analysis determines the optimal number of modes.

Usability. In a complex real environment, the sensor data collected often lacks a trusted label, which makes the supervised learning algorithm [22,24,25,27,28,32,33,35,37,38,40–42,47–51] difficult to use, while the unsupervised method [31,46] has a delay in the training phase. For example, Zhu et al. [33] simultaneously used continuous time-domain data collected by mobile device acceleration sensors, gyroscope sensors, and magnetic sensors to model owner features, and the recognition rate of the owner users reached 75%. However, the test cycle of this method needs to last for 24 h, which makes real-time detection impossible in the real-world. At the same time, this method cannot handle unlabeled data in complex environments. Although [11] proposed an unsupervised learning algorithm to correspond to unlabeled data, the parameter adjustment of the unsupervised clustering algorithm needs to pay a great price, and the generalization ability of the parameter needs to be verified. Although frontier work by [32] proposed a semi-supervised algorithm to improve accuracy, the method used only uses a single classifier (SVM), and its proposed initial classifier trained with some labeled samples is a weak classifier. If the previous iteration is used continuously to classify the unlabeled data, which will cause the continuous accumulation of its own errors, then it cannot guarantee that the resulting model has high accuracy. Different from the above method, this paper adopts a cooperative semi-supervised algorithm (Tri-Training) for label refactoring, so that a large number of false labels are corrected and the final classification accuracy is greatly improved.

Feature extraction coverage. The existing methods have a weak ability to represent features extracted from sensor data. The data collected from the motion sensor generally appears as a time-series signal. First, there are some studies that use dynamic time warping, Pearson coefficients, and other methods to compare time-series signals directly, which have some defects, such as unstable signal period segmentation and fewer representative features. Second, if traditional machine learning is used for training, feature extraction needs to be first performed on the original signal data. Most of the existing feature extraction methods are based on statistics, and empirically extract a variety of time-domain and frequency-domain features, including average, standard deviation, maximum/minimum, etc. [22,24,25,27,28,31–33,35,37,38,40–42,46,47,51]; these type of features mainly rely on manual extraction and have limited ability to represent user behavior. Third, the existing sensor data modeling methods based on deep learning [29,30,43–45,48–50] are mostly based on the original signal data, with less consideration for signal enhancement, and lack of deeper analysis of user behavior signal data. In this paper, VMD is used for signal decomposition, which not only eliminates the influence of noise but also enhances the signal effectively, so that the input features of the model have a wider coverage.

6. Conclusions

Mobile authentication is a hot topic currently, and implicit real-time authentication based on motion sensors has developed rapidly in recent years. Existing motion sensor-based authentication methods have certain deficiencies in de-noising ability, usability, and feature extraction coverage. This paper proposes a heuristic signal enhancement method that uses variational mode decomposition (VMD) to decompose the signal to eliminate the effect of noise; The semi-supervised collaborative training method (Tri-Training) is used to de-noise, so as to eliminate the influence of noise in the motion sensor data collected in the real-world. At the same time, a hybrid depth learning method is used, which uses CNN for high coverage feature extraction, and SVM for effective model training. The training results under large-scale, real-world data show that the system proposed in this paper can achieve an accuracy of 95.01%, and the effect is better than the existing frontier methods.

Author Contributions: Each author contributed extensively to the preparation of this manuscript. T.Z. and Z.W. designed the experiment; T.Z. and L.F. performed the experiments; G.C. and T.Z. analyzed the data; and T.Z. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is sponsored by Wenzhou scientific research projects for underdeveloped areas (WenRenSheFa [2020] 61(No.5), WenRenSheFa [2019] 55(No.17)), supported by the Teacher Professional Development Project for domestic visiting scholars in universities of Zhejiang Provincial Department of Education in 2019 (FX2019070), sponsored by Major scientific research projects of Wenzhou Polytechnics(No.WZY2020001), supported by Zhejiang Natural Science Foundation and Qingshan Lake Science and Technology City Foundation (No: LQY19F020001), sponsored by Zhejiang Province "the 13th Five-Year Plan" for Collaborative Education Project between Industry and University (The E-Commerce "1+X" curriculum design based on university and industry co-cultivation).

Acknowledgments: We would like to thank sensors editors and reviewers for the review efforts.

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

References

1. Mobile Phone Market Forecast. Available online: https://www.ccsinsight.com/press/company-news/mobile-phone-market-will-slip-13-in-2020-but-regain-strength-in-2021/ (accessed on 2 April 2020).
2. Brute Forcing an Android Phone. Available online: https://hackaday.com/2013/11/10/brute-forcing-an-android-phone/ (accessed on 10 November 2013).
3. Aviv, A.J.; Gibson, K.L.; Mossop, E.; Blaze, M.; Smith, J.M. Smudge attacks on smartphone touch screens. *Woot* **2010**, *10*, 1–7.
4. Researchers use Smudge Attack, Identify Android Passcodes 68 Percent of the Time. Available online: https://www.zdnet.com/article/researchers-use-smudge-attack-identify-android-passcodes-68-percent-of-the-time/ (accessed on 16 August 2010).
5. 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-SOUPS'11, Pittsburgh, PA, USA, 20–22 July 2011. [CrossRef]
6. Google Can Catch Shoulder Surfers Peeking at Your Phone. Available online: https://www.digitaltrends.com/mobile/google-shoulder-surfers-smartphone-alert/ (accessed on 28 November 2017).
7. 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-WISEC'12, Tucson, AZ, USA, 1–3 April 2012. [CrossRef]
8. Lee, W.; Ortiz, J.; Ko, B.; Lee, R. Inferring Smartphone Users' Handwritten Patterns by using Motion Sensors. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, Funchal, Madeira, Portugal, 22–24 January 2018. [CrossRef]
9. Shen, C.; Pei, S.; Yu, T.; Guan, X. On motion sensors as source for user input inference in smartphones. In Proceedings of the IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2015), Hong Kong, China, 23–25 March 2015; IEEE: Piscataway, NJ, USA, 2015. [CrossRef]
10. 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 Network and Distributed System Security Symposium, San Diego, CA, USA, 18–21 February 2018. [CrossRef]

11. 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 Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
12. Sharif, M.; Bhagavatula, S.; Bauer, L.; Reiter, M.K. Accessorize to a Crime. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016. [CrossRef]
13. Bontrager, P.; Roy, A.; Togelius, J.; Memon, N.; Ross, A. DeepMasterPrints: Generating MasterPrints for Dictionary Attacks via Latent Variable Evolution*. In Proceedings of the 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS), Redondo Beach, CA, USA, 22–25 October 2018; IEEE: Piscataway, NJ, USA, 2018. [CrossRef]
14. Gafurov, D. A survey of biometric gait recognition: Approaches, security and challenges. In Proceedings of the Annual Norwegian Computer Science Conference, Bergen, Norway, November 2007; pp. 19–31. Available online: https://www.researchgate.net/profile/Davrondzhon_Gafurov/publication/228577046_A_survey_of_biometric_gait_recognition_Approaches_security_and_challenges/links/00b49528e834aa68eb000000/A-survey-of-biometric-gait-recognition-Approaches-security-and-challenges.pdf (accessed on 11 July 2020).
15. Mantyjarvi, J.; Lindholm, M.; Vildjiounaite, E.; Makela, S.; Ailisto, H.A. Identifying users of portable devices from gait pattern with accelerometers. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, USA, 23 March 2005; Volume 2, p. 973. [CrossRef]
16. Liu, R.; Zhou, J.; Liu, M.; Hou, X. A wearable acceleration sensor system for gait recognition. In Proceedings of the 2007 2nd IEEE Conference on Industrial Electronics and Applications, Harbin, China, 23–25 May 2007; pp. 2654–2659. [CrossRef]
17. Gafurov, D.; Snekkenes, E.; Bours, P. Improved gait recognition performance using cycle matching. In Proceedings of the 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, Perth, WA, Australia, 20–23 April 2010; pp. 836–841. [CrossRef]
18. Nickel, C.; Wirtl, T.; Busch, C. Authentication of Smartphone Users Based on the Way They Walk Using k-NN Algorithm. In Proceedings of the 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Piraeus, Greece, 18–20 July 2012; IEEE: Piscataway, NJ, USA, 2012. [CrossRef]
19. Derawi, M.O.; Bours, P.; Holien, K. Improved Cycle Detection for Accelerometer Based Gait Authentication. In Proceedings of the 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Darmstadt, Germany, 15–17 October 2010; IEEE: Piscataway, NJ, USA, 2010. [CrossRef]
20. Gafurov, D.; Snekkenes, E.; Bours, P. Gait authentication and identification using wearable accelerometer sensor. In Proceedings of the 2007 IEEE Workshop on Automatic Identification Advanced Technologies, Alghero, Italy, 7–8 June 2007; pp. 220–225. [CrossRef]
21. Hoang, T.; Choi, D.; Nguyen, T. Gait authentication on mobile phone using biometric cryptosystem and fuzzy commitment scheme. *Int. J. Inf. Secur.* **2015**, *14*, 549–560. [CrossRef]
22. 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-ISWC'14, Seattle, WA, USA, 13–17 September 2014. [CrossRef]
23. Derawi, M.O.; Nickel, C.; Bours, P.; Busch, C. Unobtrusive User-Authentication on Mobile Phones Using Biometric Gait Recognition. In Proceedings of the 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Darmstadt, Germany, 15–17 October 2010; IEEE: Piscataway, NJ, USA, 2010. [CrossRef]
24. 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 (BTAS), Washington, DC, USA, 27–29 September 2010; IEEE: Piscataway, NJ, USA, 2010. [CrossRef]
25. Ho, C.C.; Eswaran, C.; Ng, K.; Leow, J. An unobtrusive Android person verification using accelerometer based gait. In Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia-MoMM'12, Bali, Indonesia, 3–5 December 2012. [CrossRef]
26. 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]

27. Nickel, C.; Brandt, H.; Busch, C. Classification of acceleration data for biometric gait recognition on mobile devices. *BIOSIG 2011–Proc. Biom. Spec. Interest Group* **2011**. [CrossRef]
28. Hoang, T.; Choi, D.; Vo, V.; Nguyen, A.; Nguyen, T. A lightweight gait authentication on mobile phone regardless of installation error. In *Security and Privacy Protection in Information Processing Systems. SEC 2013. IFIP Advances in Information and Communication Technology*; Janczewski, L.J., Wolfe, H.B., Shenoi, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 83–101. [CrossRef]
29. Zou, Q.; Wang, Y.; Wang, Q.; Zhao, Y.; Li, Q. Deep Learning-Based Gait Recognition Using Smartphones in the Wild. *IEEE Trans. Inf. Forensics Sec.* **2020**, *15*, 3197–3212. [CrossRef]
30. Gadaleta, M.; Rossi, M. Idnet: Smartphone-based gait recognition with convolutional neural networks. *Pattern Recogn.* **2018**, *74*, 25–37. [CrossRef]
31. Cola, G.; Avvenuti, M.; Vecchio, A.; Yang, G.; Lo, B. An unsupervised approach for gait-based authentication. In Proceedings of the 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN), Cambridge, MA, USA, 9–12 June 2015; pp. 1–6. [CrossRef]
32. Zhu, T.; Qu, Z.; Xu, H.; Zhang, J.; Shao, Z.; Chen, Y.; Prabhakar, S.; Yang, J. RiskCog: Unobtrusive real-time user authentication on mobile devices in the wild. *IEEE Trans. Mob. Comput.* **2019**, *19*, 466–483. [CrossRef]
33. Jiang, Z.; Pang, W.; Xiao, W.; Zhang, J. SenSec: Mobile security through passive sensing. In Proceedings of the 2013 International Conference on Computing, Networking and Communications (ICNC), San Diego, CA, USA, 28–31 January 2013; IEEE: Piscataway, NJ, USA, 2013. [CrossRef]
34. Lee, W.; Liu, X.; Shen, Y.; Jin, H.; Lee, R.B. Secure Pick Up. In Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies-SACMAT '17 Abstracts, Indianapolis, IN, USA, 21–23 June 2017. [CrossRef]
35. Buriro, A.; 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 (ISBA), New Delhi, India, 22–24 February 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–8. [CrossRef]
36. Feng, T.; Liu, Z.; Kwon, K.; Shi, W.; Carbunar, B.; Jiang, Y.; Nguyen, N. Continuous mobile authentication using touchscreen gestures. In Proceedings of the 2012 IEEE Conference on Technologies for Homeland Security (HST), Waltham, MA, USA, 13–15 November 2012; IEEE: Piscataway, NJ, USA, 2012. [CrossRef]
37. Gascon, H.; Uellenbeck, S.; Wolf, C.; Rieck, K. Continuous authentication on mobile devices by analysis of typing motion behavior. In Proceedings of the Sicherheit 2014–Sicherheit, Schutz und Zuverlässigkeit, Vienna, Austria, 19–21 March 2014. [CrossRef]
38. Giuffrida, C.; Majdanik, K.; Conti, M.; Bos, H. I Sensed It Was You: Authenticating Mobile Users with Sensor-Enhanced Keystroke Dynamics. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Egham, UK, 10–11 July 2014; pp. 92–111. [CrossRef]
39. Kayacik, H.G.; Just, M.; Baillie, L.; Aspinall, D.; Micallef, N. Data driven authentication: On the effectiveness of user behaviour modelling with mobile device sensors. *arXiv* **2014**, arXiv:1410.7743. [CrossRef]
40. Lee, W.; Lee, R.B. Multi-sensor Authentication to Improve Smartphone Security. In Proceedings of the 1st International Conference on Information Systems Security and Privacy, Angers, France, 9–11 February 2015. [CrossRef]
41. 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. Inf. Forensics Sec.* **2015**, *11*, 877–892. [CrossRef]
42. Shen, C.; Li, Y.; Chen, Y.; Guan, X.; Maxion, R.A. Performance analysis of multi-motion sensor behavior for active smartphone authentication. *IEEE Trans. Inf. Forensics Sec.* **2017**, *13*, 48–62. [CrossRef]
43. Ronao, C.A.; Cho, S. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst. Appl.* **2016**, *59*, 235–244. [CrossRef]
44. Yang, J.; Nguyen, M.N.; San, P.P.; Li, X.L.; Krishnaswamy, S. Deep convolutional neural networks on multichannel time series for human activity recognition. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015. [CrossRef]
45. Ordóñez, F.; Roggen, D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* **2016**, *16*, 115. [CrossRef] [PubMed]
46. Shila, D.M.; Srivastava, K.; O'Neill, P.; Reddy, K.; Sritapan, V. A multi-faceted approach to user authentication for mobile devices—Using human movement, usage, and location patterns. In Proceedings of the 2016 IEEE Symposium on Technologies for Homeland Security (HST), Waltham, MA, USA, 10–11 May 2016; pp. 1–6. [CrossRef]

47. Ehatisham-ul-Haq, M.; Awais Azam, M.; Naeem, U.; Amin, Y.; Loo, J. Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing. *J. Netw. Comput. Appl.* **2018**, *109*, 24–35. [CrossRef]
48. Volaka, H.C.; Alptekin, G.; Basar, O.E.; Isbilen, M.; Incel, O.D. Towards Continuous Authentication on Mobile Phones using Deep Learning Models. *Procedia Comput. Sci.* **2019**, *155*, 177–184. [CrossRef]
49. Centeno, M.P.; Guan, Y.; van Moorsel, A. Mobile Based Continuous Authentication Using Deep Features. In Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning-EMDL'18, Munich, Germany, 15 June 2018. [CrossRef]
50. Abuhamad, M.; Abuhmed, T.; Mohaisen, D.; Nyang, D. AUToSen: Deep-Learning-Based Implicit Continuous Authentication Using Smartphone Sensors. *IEEE Internet Things* **2020**, *7*, 5008–5020. [CrossRef]
51. Lee, W.; Lee, R.B. Implicit Smartphone User Authentication with Sensors and Contextual Machine Learning. In Proceedings of the 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Denver, CO, USA, 26–29 June 2017; IEEE: Piscataway, NJ, USA, 2017. [CrossRef]
52. Choi, H.; Lee, B.; Yoon, S. Biometric Authentication Using Noisy Electrocardiograms Acquired by Mobile Sensors. *IEEE Access* **2016**, *4*, 1266–1273. [CrossRef]
53. Li, H.; Zhang, Y.; Zheng, H. Hilbert-Huang transform and marginal spectrum for detection and diagnosis of localized defects in roller bearings. *J. Mech. Sci. Technol.* **2009**, *23*, 291–301. [CrossRef]
54. Fu, L.; Zhu, T.T.; Pan, G.; Chen, S.; Zhong, Q.; Wei, Y. Power Quality Disturbance Recognition Using VMD-Based Feature Extraction and Heuristic Feature Selection. *Appl. Sci.* **2019**, *9*, 4901. [CrossRef]
55. Dragomiretskiy, K.; Zosso, D. Variational mode decomposition. *IEEE Trans. Signal Process.* **2014**, *62*, 531–544. [CrossRef]
56. Fu, L.; Zhu, T.T.; Zhu, K.; Yang, Y.L. Condition Monitoring for the Roller Bearings of Wind Turbines under Variable Working Conditions Based on the Fisher Score and Permutation Entropy. *Energies* **2019**, *12*, 3085. [CrossRef]
57. Wang, Y.X.; Liu, F.Y.; Jiang, Z.S.; He, S.L.; Mo, Q.Y. Complex variational mode decomposition for signal processing applications. *Mech. Syst. Signal Proc.* **2018**, *86*, 75–85. [CrossRef]
58. Blum, A.; Mitchell, T. Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory-COLT' 98, Madison, WI, USA, 24–26 July 1998. [CrossRef]
59. Zhou, Z.-H.; Li, M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 1529–1541. [CrossRef]
60. Decatur, S.E.; Gennaro, R. On learning from noisy and incomplete examples. In Proceedings of the Eighth Annual Conference on Computational Learning Theory-COLT'95, Santa Cruz, CA, USA, 5–8 July 1995. [CrossRef]
61. Android UsageStatsManager. Available online: https://developer.android.com/reference/android/app/usage/UsageStatsManager.html (accessed on 2 May 2020).
62. Bhuiyan, S.M.A.; Khan, J.; Murphy, G. WPD for Detecting Disturbances in Presence of Noise in Smart Grid for PQ Monitoring. *IEEE Trans. Ind. Appl.* **2018**, *54*, 702–711. [CrossRef]
63. Emmagee. Available online: https://github.com/NetEase/Emmagee (accessed on 2 May 2020).