Hindawi
Security and Communication Networks
Volume 2023, Article ID 3673113, 11 pages
https://doi.org/10.1155/2023/3673113

Research Article

A Robust Continuous Authentication System Using Smartphone Sensors and Wasserstein Generative Adversarial Networks

Shihong Zou,[1] **Huizhong Sun**,[1] **Guosheng Xu**,[1] **Chenyu Wang**,[1] **Xuanwen Zhang**,[2] **and Ruijie Quan**[3]

[1]*School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100786, China*
[2]*Comprehensive Research Center of Electronic Information Technology in the MIIT of China, Weihai 264200, China*
[3]*University of Technology Sydney, Sydney, Australia*

Correspondence should be addressed to Huizhong Sun; sunhuizhong@bupt.edu.cn

Received 31 January 2022; Revised 21 September 2022; Accepted 3 February 2023; Published 26 April 2023

Academic Editor: Gökhan Kul

Since the continuous authentication (CA) system based on smartphone sensors has been facing the challenge of the low-data regime under some practical scenarios, which leads to low accuracy of CA, it needs to be solved urgently. To this end, currently, the generative adversarial networks (GAN) provide a powerful method to train the result generative model that could generate very convincing verisimilar data. The framework of the GAN and its variants shed much light on improving the performance of CA. Therefore, in this article, we propose a continuous authentication system on smartphones based on a Wasserstein generative adversarial network (WGAN) for sensor data augmentation, which utilizes accelerometers, gyroscopes, and magnetometers of smartphone sensors to sense phone movements caused by user operation behavior. Specifically, based on sensor data under different user activities, the WGAN is used to create additional data in training data for data augmentation. With the augmented data, we design a convolutional neural network to learn and extract deep features from sensor data, and then use four classifiers of RF, OCSVM, DT, and KNN to train these features. Finally, we train and test on the HMOG dataset, and the results show that the EER of the authentication system is between 3.68% and 6.39% on the sensor data with a time window of 2 s.

1. Introduction

With advances in smartphones in terms of computing power and storage capacity, it has evolved into a versatile device that can meet both personal and business needs. Users often store photos, chat messages, and sensitive files on such mobile devices. However, sensitive information stored on mobile devices has the problem of information leakage. Thus, there are higher requirements for the security mechanism of user authentication. Traditional authentication mechanisms are based on knowledge-based authentication (e.g., passwords and PINs) and authentication based on physiological biometrics (e.g., voice and face patterns). This one-time login process does not guarantee that the identified user is the real user throughout the login session. If an illegal user can bypass the initial login session, this restriction may expose the device to information theft and leakage. As a result, smartphones need a continuous authentication (CA) mechanism that can protect user information throughout working hours to supplement the initial login authentication, thereby providing more comprehensive security protection.

In the CA scheme, authentication based on user behavioral biometrics has higher security and reliability, where the authentication-based motion pattern or touchscreen gestures does not require additional equipment. In particular, for the studies using motion patterns as behavioral biometrics, there are many factors that affect the acquisitions of motion sensor signals in real-world scenarios, leading to differences in user behavior modeling between the enrollment phase and the authentication phase. Smartphone built-in motion sensors are designed to detect and measure movement determined by various factors, such as human activity (e.g., walking or sitting), smartphone holding

postures, body posture, or psychological and physiological states (eg., tension or stress). To provide accurate authentication performance, most motion sensor-based studies require the subjects to operate smartphones in a fixed usage environment (eg., walking hand-held [1] and sitting on a chair [2]). These stringent requirements make the authentication system work well in a laboratory environment, whereas it is hard to make robust authentication in real-world scenarios.

As deep learning has shown its effectiveness in many fields, many deep learning methods have been gradually applied to continuous authentication research [3–6]. The scarcity of training data still endures as an obstacle to the establishment of robust deep authentication models. For instance, Tran and Choi [3] proposed two data augmentation algorithms, arbitrary time deformation (ATD) and stochastic magnitude perturbation (SMP), by observing the changes in actual gait data to solve the data scarcity problem and improve the robustness of deep gait models. Lu et al. [4] proposed an authentication framework based on a recurrent neural network to capture the unique behavioral biometrics of the user when entering a password. It can infer malicious imposters from limited training data. Amini et al. [5] extracted the time domain and frequency domain features from the smartphone's motion sensor, combined with the long short-term memory (LSTM) model, and constructed an authentication framework. The framework can identify users with 96.70% accuracy within 20 seconds. Zou et al. [6] proposed to capture human gait behavior based on motion sensors (an accelerometer and a gyroscope) and apply hybrid deep neural networks to extract gait features, making gait recognition robust.

Therefore, in this article, we propose a robust continuous authentication system using a WGAN on the smartphone for sensor data augmentation, which mainly focuses on the use of smartphones for user authentication under different activities. Different from the state-of-the-art authentication methods, the existing method usually requires the user to swipe the screen and press the keys of the mobile phone under a specified activity. Our method collects motion sensor (an accelerometer, a gyroscope, and a magnetometer) data under different activities. To obtain well authentication performance, we utilize the WGAN model to create additional data from raw sensor data under different activities for data augmentation. With the augmented data, we design a convolutional neural network to learn and extract deep features from sensor data, and then use four classifiers of RF, OCSVM, DT, and KNN to train these features. Finally, we train and test on the HMOG dataset.

The main contributions of this article are as follows:

(i) Based on sensor data under different user activities, we propose the WGAN to generate additional data in the training data for data augmentation.

(ii) With the augmented data, we design a convolutional neural network to learn and extract deep features from sensor data, then use four classifiers of RF, OCSVM, DT, and KNN to train these features, and finally test on different user activity data.

(iii) The experiments are evaluated and analyzed on the HMOG [7] dataset, and the results show that the EER of the authentication system is between 3.68% and 6.39% on the sensor data with a time window of 2 s.

The rest of this article is organized as follows: Section 2 mainly reviews the related work of reviewing the data augmentation in authentication systems and the continuous authentication based on deep learning. Section 3 presents the details of our proposed method. Section 4 describes our experimental results and analysis. Section 5 concludes our approach.

2. Related Work

In this section, we review the data augmentation in authentication systems and the continuous authentication based on deep learning.

2.1. Data Augmentation in Authentication Systems. Data augmentation is an effective method to solve the problem of data scarcity and improve the robustness of deep learning models. In this approach, the amount of training data is multiplied by introducing noise or appropriately modifying the raw data [8]. In deep learning, using more training data can reduce the overfitting and train a more robust deep model. Due to its effectiveness, data augmentation is widely used in various tasks (such as image recognition [9–11], speech recognition [12, 13], and anomaly detection).

In continuous authentication methods, data augmentation techniques have not been given enough attention in continuous authentication based on motion sensors. To the best of our knowledge, there are currently few studies utilizing data augmentation methods to augment sensor data. For instance, Li et al. [14] propose a sensor-based continuous authentication system for continuously monitoring users' behavior patterns. In this system, the rotation method is applied to the collected raw data to create additional data, thereby improving the robustness of the authentication system. Buriro et al. [15] proposed SwipeGAN: generating swipe samples for smartphone user authentication. Experimental results demonstrate the quality of the generated synthetic samples and their effectiveness in improving the accuracy of the authentication scheme. Li et al. [16] proposed a smartphone-based continuous authentication based on user behavior patterns. By utilizing the accelerometer and gyroscope in the smartphone, data augmentation techniques such as permutation, sampling, scaling, cropping, and jittering were applied to the training data to create additional data. Sitova et al. [7] adds HMOG (hand movement, direction, and grasping) features for continuous authentication of smartphones, which greatly improves the authentication performance.

2.2. Deep Learning for Continuous Authentication. Deep learning algorithms are becoming increasingly popular to improve the performance of continuous authentication systems.

Amini et al. [2] proposed a continuous authentication method based on user action behavior. The method extracted the time-domain and frequency-domain features of the user's motion behavior and then uses the LSTM algorithm to learn the time-domain and frequency-domain features to establish a continuous authentication model, which can identify users with an accuracy of 96.70% within 20 seconds. Zhang et al. [17] proposed an LSTM-based user authentication system. The system uses the mutual information model and the principal component analysis model to process data, extract features of gait data, and verify the identity of legitimate users. The results show that the average recognition accuracy of the LSTM model is about 95%. Lu et al. [4] used a unique motion pattern as a behavioral biometric when the user entered a password and used the recurrent neural network (RNN) model to learn the deep representation of the movement patterns. Experiments show that DeepAuth can well solve the security problems of resource-constrained devices. Zou et al. [6] proposed a hybrid model based on the CNN and the RNN to learn robust gait features, including time-domain and frequency-domain characteristics, to deal with the problem of user identification in complex scenes.

Existing research and results have proved the effectiveness of deep learning in user behavior identification. However, deep learning-based authentication models must be based on sufficient user behavior data. Specifically, training a deep network typically requires a large, dispersed dataset that covers a large number of data instances that may arise in practice. In this article, we propose a continuous authentication system based on WGAN-generated sensor data.

3. Proposed Method

In this section, we describe a robust continuous authentication system based on smartphone sensors and Wasserstein Generative Adversarial Networks. The built-in sensors of smartphones, including accelerometers, gyroscopes, and magnetometers, are used to perceive the user's sliding screen and click.

As shown in Figure 1, the authentication system architecture mainly includes two parts: enrollment phase and authentication phase. The raw data refer to the sensor data recorded when the user operates the smartphone with different tasks. The enrollment phase is mainly for data augmentation of the training data, CNN deep feature learning, and the construction of a training model. The user authentication phase is mainly to verify the test data and judge the legitimacy of the user.

3.1. Raw Data. In this article, we use the public HOMG dataset, which records real-time touch, sensor, and keystroke data invoked by 100 users when they interact with the smartphone (sampling frequency f = 100Hz). Data were recorded for three smartphone usage scenarios: (1) reading documents; (2) text production; and (3) navigation on a map to locate a destination. The tasks lasted 5 to 15 minutes (participants were randomly assigned), and each participant completed 24 stages (8 reading stages, 8 writing stages, and 8 map navigation stages) for a total of 2 to 6 hours of behavior feature.

Based on the above dataset, we chose three sensors on the smartphone, namely an accelerometer, a gyroscope, and a magnetometer, to monitor the user's behavior on the smartphone. The accelerometer and gyroscope are motion sensors that capture the user's coarse-grained and fine-grained motion patterns, respectively, while the magnetometer is a position sensor that determines the phone's physical location in the real frame of reference.

In the enrollment phase, we select 100 minutes of data for each user and train the data with a time window $t = 2$ of seconds, step size is 0.25 seconds, the sample size is $n = t^* f$, and f *is* the sampling frequency. In the authentication phase, we collect sensor samples, and the samples represent $<t, x, y, z>$, where t represents the time stamp and x, y, and z represent the different axis values of the accelerometer, gyroscope, and magnetometer. Each of t, x, y, and z is stored as a vector of a different sensor.

3.2. Data Preprocessing. In this section, the data preprocessing stage aims to feature extraction from the raw sensor. We select the sensor data under different tasks (Reading + Sitting, Reading + Walking, Writing + Sitting, Writing + Walking, Map + Sitting, and Map + Walking) from the raw data. According to different tasks, we read the values of the accelerometer, gyroscope, and magnetometer. In our study, we consider two different types of input data for these three sensors to evaluate our proposed model. One is the raw data input, which is used for CNN learning (see Section 3.4). The other extracts handcrafted features from the raw data, such as the mean over a sliding window, which is used on experimental benchmark data.

For the sensor data for each task, the data are then segmented according to time windows. For raw data, each input is a matrix of size: $X = \{X^{(k)}\} = d^{(k)*}2f^*T$(see Section 3.4 for detailed description), as the input of WGAN data augmentation and CNN feature extraction. The length of the feature vector depends on the sampling frequency f of the sensor and the time interval T of data acquisition.

For handcrafted features, we calculate the min, max, mean, mean square, variance, standard deviation, mean squared, root-mean-square (RMS), mean-squared-error (MSE), peak, mean-square- frequency (MSF), root-mean-square-frequency (RMSF), and root-mean-square-error (RMSE) over the window. These features are computationally easier to implement and have been shown to be effective for CA [18]. Then, each window is a vector of size: number of extracted features × number of sensor channels.

Finally, to reduce the effect of noise, we normalize the raw data and handcrafted features and map the interval ($x_{\min}$ and $x_{\max}$) to the unit scale (0 and 1).

$$x = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{1}$$

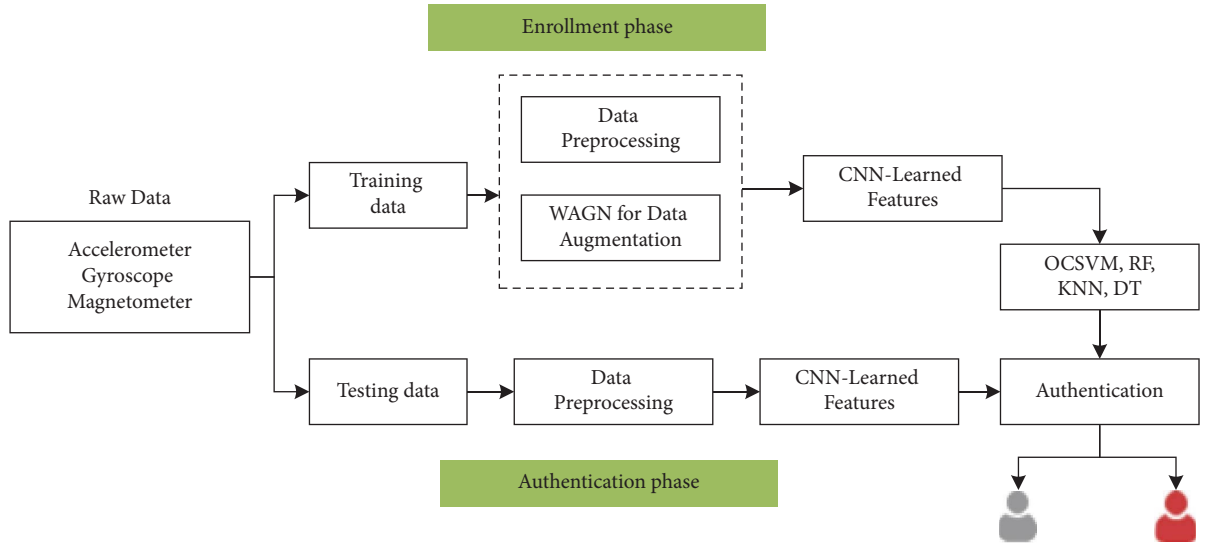

Figure 1: The architecture of continuous authentication.

where $x_{\min}$ min and $x_{\max}$ are the minimum and maximum values in the feature vector.

3.3. Data Augmentation. The GAN [19] is a deep learning model, which consists of a generator and a discriminator. The discriminator is used to assist in training a generator that can accurately learn the distribution characteristics of the raw data. The generator continues to evolve to generate synthetic data that is closer to the real data, while the discriminator also continues to evolve to improve its ability to discriminate between real and fake data. In an ideal state, both of them reach a dynamic equilibrium and optimize their performance through adversarial game training: the generator learns the distribution properties of real data accurately by receiving random noise (usually from uniform or normal distribution) and generates data that can be faked, while the discriminator distinguishes real data from generated data [20]. In this section, we design a WGAN based on the GAN for data augmentation in a continuous authentication system.

3.3.1. WGAN. The GAN training process consists of two adversarial networks: the generator network G maps noise to the input space, and the discriminator network D distinguishes between real and generated data. In this article, we input real sensor data x for the following two-player minimax game with equation (2):

$$\min \max L(D,G) = E_{x\sim p_r}[\log D(x)] + E_{x\sim p_g}[\log(1-D(x))], \tag{2}$$

where p_r is the distribution of real data, and p_g is the generative data distribution implicitly defined by $x = G(z)$ and $z \sim p(z)$, for which z is sampled from Gaussian distribution.

Based on equation (2), to effectively address the problem of mode collapse, we measure the distance between real samples and generated samples by Wasserstein distance instead [21]. The Wasserstein distance is defined as follows:

$$W(p_r, p_g) = \inf_{\gamma \in \prod(p_r, p_g)} E[\|x-y\|], \tag{3}$$

where $\prod(p_r, p_g)$ denotes the entire set of feasible joint distributions $\gamma(x, y)$ of the true data distribution p_r and generative data distribution p_g. However, equation (3) is highly intractable. We use the Kantorovich–Rubinstein [21] duality to reconstruct the Wasserstein distance as follows:

$$W(p_r, p_g) = \sup_{\|D\|L \le 1} E_{x\sim p_r}[D(x)] - E_{x\sim p_g}[D(x)], \tag{4}$$

where D is the set of Lipschitz continuous functions subject to this constraint (as in equation (5)) and sup is the least upper bound.

$$|D(x_1) - D(x_2)| \le |x_1 - x_2|. \tag{5}$$

The objective function between the generator and the discriminator of the WGAN is as follows:

$$\min_G \max_D L(D,G) = E_{x\sim p_r}[D(x)] + E_{x\sim p_g}[D(G(\tilde{x}))]. \tag{6}$$

To perform the discriminator of the WGAN, we utilize parameters to clip the weights of the discriminator. c is a controlled hyperparameter [22], and the weights of the discriminator must be in a certain range $(-c, c)$.

3.3.2. WGAN for Different Activity Data. In this section, we randomly separate the 88 participants' data into two parts. 60 participants' data are first augmented by the WGAN and then used for the designed CNN training and validation, while 28 participants' data are just extracted features by the trained CNN for classifier training. We use 60 participants' data as the input of the WGAN.

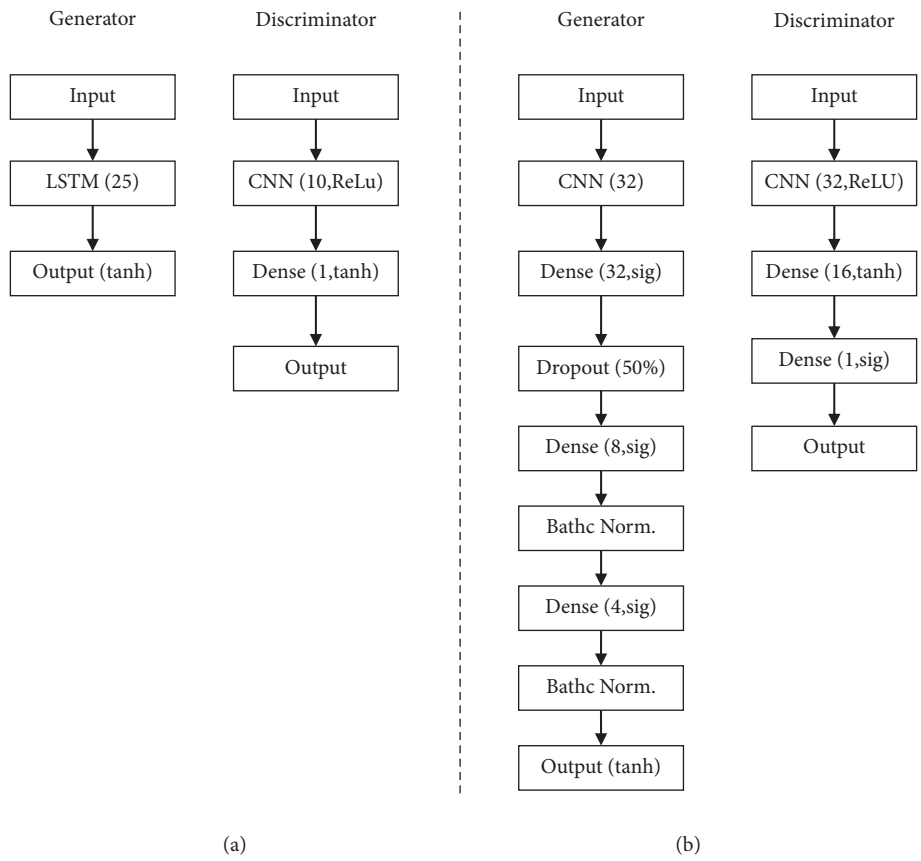

Figure 2: The WGAN for different activity data. (a) Is the WGAN for walking? (b) Is the WGAN for sitting?

According to the task types under different user activities, we choose different neural network modules (LSTM and CNN) to build a generative adversarial network model. One is for user actions while walking (see Figure 2(a)), and the other is for user actions while sitting (see Figure 2(b)).

Figure 2(a) shows a generator based on a LSTM layer and uses a Tanh activation on its output. The generator's responsibility is to generate data from the noise data that has a similar structure to the real sensor data.

The discriminator has a CNN layer using 32 filters with ReLU activation function and a dense layer with Tanh activation function. The output layer has a single neuron without an activation function. The discriminator's responsibility is to predict if its input is real or not based on its Wasserstein distance.

Figure 2(b) shows a generator based on a CNN with 32 filters. The model utilized a dense layer and a sigmoid activation function. We applied dropout with a rate of 50% and a dense layer that used the sigmoid activation function. We then added a batch normalization layer and a dense layer, which applied the sigmoid activation function. We again applied the batch normalization layer. The output layer of the generator was dense with the Tanh as an activation function.

The discriminator used the CNN of 32 kernels with ReLU activation and a dense layer with Tanh activation function. We also added a dense layer of one unit with a sigmoid activation function. The output layer is a further dense layer of one neuron but without an activation function.

3.4. CNN for Feature Extraction. For augmented sensor data and real sensor data, based on DeepSense [23], this article designs a CNN architecture to learn and extract deep features, which consists of two parts: (1) a separate volume for each input sensor tensor $X^{(k)}$ product network and (2) k separate convolutional network outputs combined convolution, where $k = 3$.

The model transfers the input $(X^{(k)} \in d^{(k)} * 2f * T)$, as shown in Figure 3, where $d^{(k)}$ represents the value corresponding to the k sensor signal, k is the number of sensor types, T collects the time for each sensor data, and f is the sampling frequency.

Since the structure of a single convolutional network for different sensors is the same, this article focuses on a single convolutional subnet of the input tensor $X^{(k)}$.

3.4.1. CNN for Single Sensor Data. Each sensor sample data contains multiple time windows, and this article will process it layer by layer in T dimension (one window at a time). Each $d^{(k)} * 2f$ window slice through the convolutional neural network component consists of three stages, as shown in Figure 4:

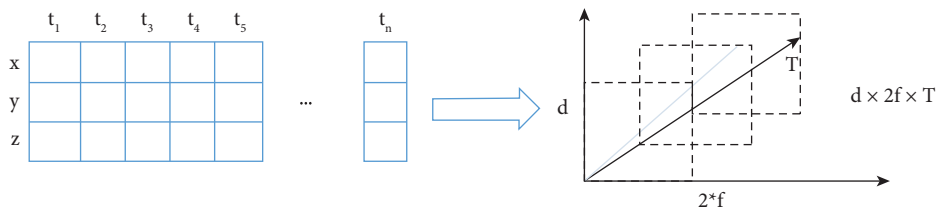

Figure 3: Sensors segment and measure data.

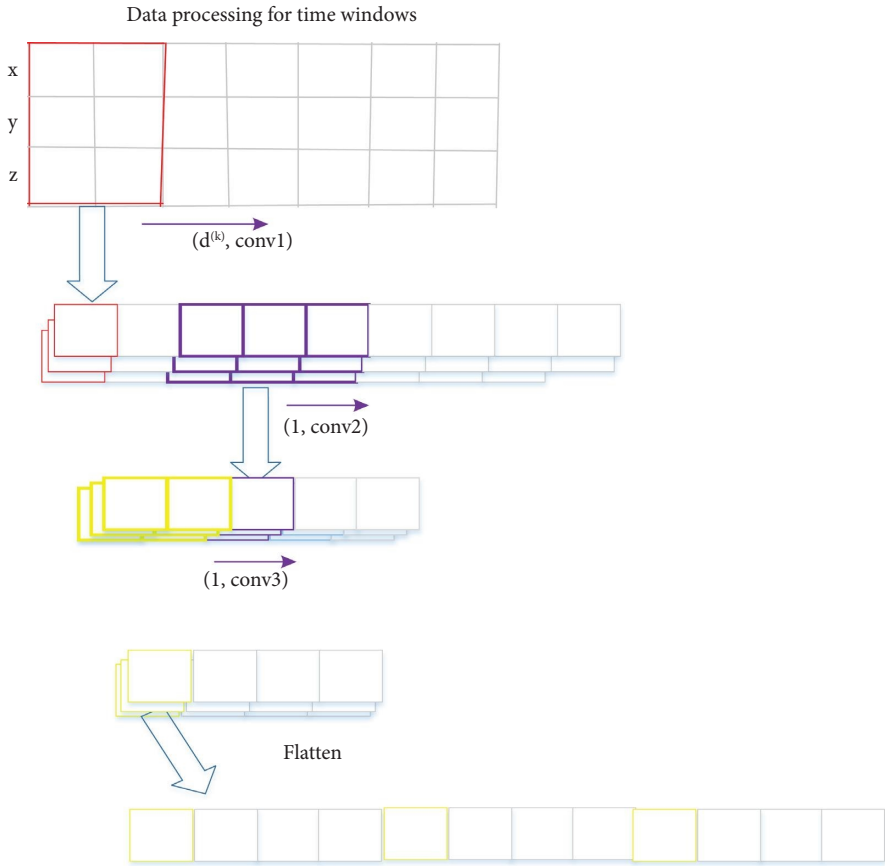

Figure 4: The CNN for single sensor data.

We use 2D convolutional filters to capture interactions between dimensions in the local frequency domain. The output is then passed through a 1D convolutional filter layer to capture high-level relations. The output of the last filter layer is flattened to produce sensor feature vectors.

3.4.2. Integrate Multisensor Data. The procedure given above is followed for each K sensor. We now have K sensor features, which we can pack into a matrix with K rows.

The sensor feature matrix is then passed through a second convolutional neural network component with the same structure as subsection 3.4.1. That is, the 2D convolutional filter layer is followed by two 1D vector layers. Finally, we flatten the output of the last filter into the integrated sensor feature vector. The window width t is appended at the end of this vector.

A time window now has the combined sensor feature vector, and we need to repeat the above process for all T windows.

3.5. Classification Algorithm. With the CNN-learned deep features, we adopt an oversampling strategy to deal with the data imbalance problem [24], then use four classifiers of RF, OCSVM, DT, and KNN to train these features, and finally test them on different user activity data.

4. Experimental Results and Analysis

In this section, to evaluate the authentication performance of the WGAN, we introduce the experimental setup and then conduct extensive experiments. Next, we evaluate the effectiveness of the WGAN, analyze and discuss the authentication accuracy of the WAGN on data augmentation and CNN-learned features, explore the authentication accuracy of unseen users, and finally compare with existing authentication methods.

4.1. Experimental Settings

(1) CNN and Classifier Training: For CNN training, 80% randomly-selected 60 users' data are used for training and the rest 20% for validation. For classifier training, we randomly select one user from the 28 users as a legitimate user and the remaining 27 users as impostors. We perform ten-fold cross validation on every legitimate user. That is, the positive samples of legal users are divided into 10 subsets on average, of which 9 subsets are used as training sets, and the rest are used as test sets. Then, the negative samples from all the impostors with the same size to positives are selected and then divided into ten subsets, where one of them is used as the testing set. The above process is repeated 10 times until each positive or negative sample subset is tested once. Finally, we repeat the 10-fold cross validation 28 times until each of the 28 users is selected as a legitimate user once.

(2) Evaluation Metrics: We evaluate the effectiveness of the method using two evaluation metrics: accuracy and EER. The accuracy is the percentage ratio of the total number of correct authentication against the total number of authentication. EER is the point that the true acceptance rate (FAR) equals the true rejection rate (FRR). The lower the EER, the higher the authentication accuracy [25].

4.2. Efficiency of the WGAN. We design the WGAN that can learn the details of raw sensor data for different activities (operation while sitting or walking). As shown in Figure 4, the red lines represent the loss of generating sensor data in the process of iterative training, and the blue line represents the loss of identifying true and false data in the process of iterative training. With the number of iterations increases, the discriminator and generator gradually converge to a small value. In particular, when the iteration period reaches 3000, the discriminator and generator loss converges to 0, indicating that the generated sensor data are highly similar to the real data.

Through continuous training of the WGAN model, the ability of the generator to capture real sensor data distribution during human-computer interaction is improved. From Figure 5, we can see that the data generated when standing on the mobile phone are more stable than the data generated when walking.

From the generator's point of view, different WGAN models can achieve stable results. The test results meet our goal of the generator being able to learn the true distribution of real sensor data and output synthetic sensor data that is close to real sensor data.

4.3. Effectiveness of WGAN Augmentation. In this section, we evaluate the authentication performance on real and augmented data. These data are all extracted by the CNN. As shown in Table 1, we use four classifiers to validate the mean EER from different data sizes.

We can see from Table 1 that the combination of real data and augmented data improves the authentication performance. For example, when the data size is 75% of the training data, the OCSVM classifier has the highest mean EER. The mean EER is reduced by 48% after the WGAN augmentation method is used, and the mean EER is 3.68%, it performs the best performance. For other classifiers, as the size of the data set increases, the authentication performance is better. In particular, the mean EER is generally reduced after the data augmentation method is used.

4.4. Effectiveness of the CNN. In this section, to evaluate the effectiveness of the features learned by the CNN, we choose four classifiers to test on handcrafted features (See Reference [18] for features extracted) and CNN-learned features, and the accuracy of our proposed method on the four classifiers is shown in Figure 6.

As presented in Figure 5, we can see that the authentication EER of the CNN-learned features (yellow box) is significantly better than that based on handcrafted features (purple box). In particular, based on CNN-learned features with the OCSVM classifier shows the best EER. In addition, as shown in Table 2, the mean ERR of the OCSVM classifier based on CNN-learned features proposed in this article reaches 3.86%, which is 6.61% lower than statistical features.

We compared the results of handcrafted features and CNN features without any data augmentation (DA) on OCSVM, as shown in Table 3.

In Figure 7 and Table 3, we can see that when the sampling time is fixed, the authentication accuracy of CNN-learned features (DA) and hand-crafted features (DA) is generally higher. When there is no data augmentation, CNN-learned features are compared with hand-crafted features, and the accuracy of authentication based on CNN-learned features is improved.

4.5. Authentication Performance under Different Activities. To verify the robustness of the authentication system, we test data on different user activities. Different new activity data and old activity data are divided according to the task type. We use the authentication model trained from historical user activity data to deal with new activity data. The authentication results are as follows:

From Table 4, we can see the mean EER (%) of different user activities for different classifiers. Our proposed method has the smallest mean ERR of 3.87% on the OCSVM classifier. The mean EER of the RF classifier on Sitting + Reading and Walking + Map is 4.32% and 7.1%, respectively, and the mean EER of the OCSVM classifier on Sitting +Map, Walking + Reading, and Walking + Writing is 6.39%, 4.58%, and 5.14%, respectively. In addition, we can see from Figure 8 that the EER values in different user activities are generally low and stable, which indicates that the designed CNN has high efficiency and strong robustness.

4.6. Accuracy for Unseen Users. To evaluate unseen users, we explore the authentication accuracy of the WGAN on

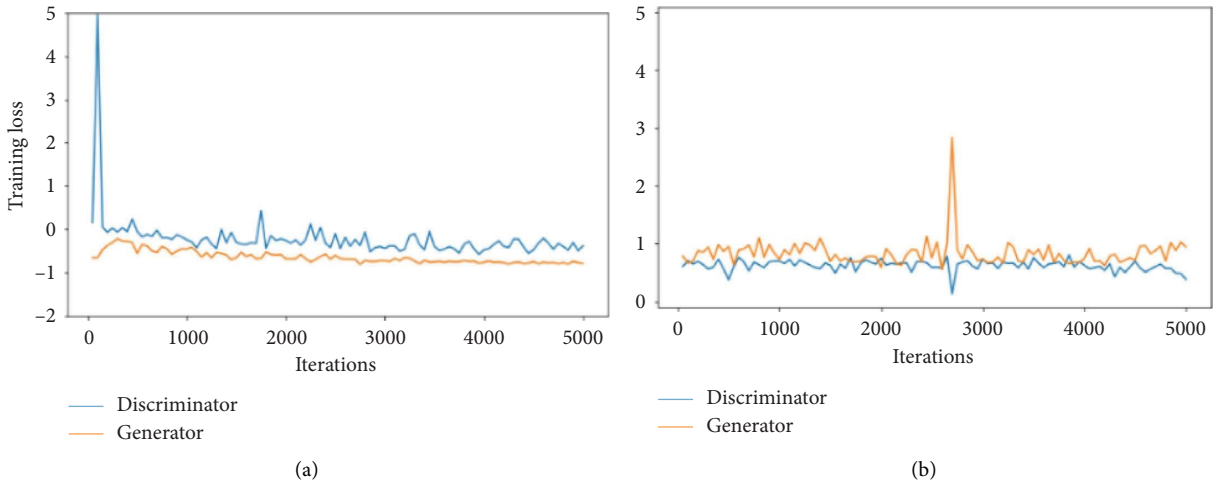

Figure 5: (a) Discriminator and generator loss for walking. (b) Discriminator and generator loss for sitting.

Table 1: Mean EER (%) of real data and augmented data for the four classifiers at different data sizes.

Classifiers	Real data				Real data + Augmented data			
	25%	50%	75%	100%	25%	50%	75%	100%
KNN	21.28	15.54	13.65	12.78	12.18	11.69	9.08	8.57
OCSVM	22.18	13.58	9.87	10.56	25.13	11.83	**3.68**	4.02
RF	19.25	10.58	9.67	8.86	7.89	6.94	5.48	4.84
DT	15.89	12.89	9.68	7.73	9.86	7.52	6.83	5.94

The mean EER is reduced by 48% after the WGAN augmentation method is used, and the mean EER is 3.68%, it performs the best performance.

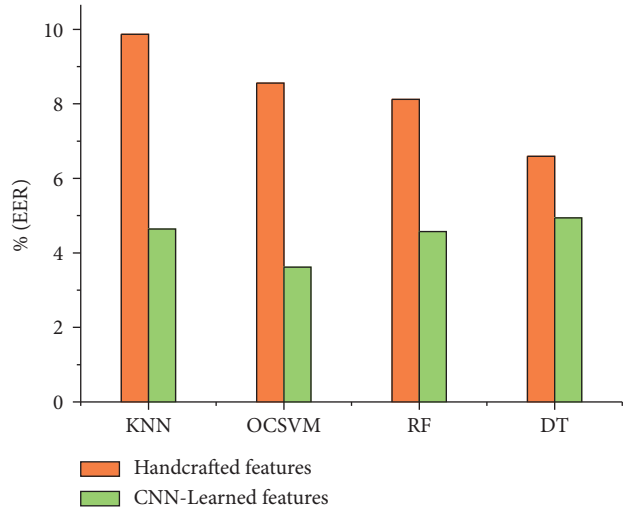

Figure 6: Mean ERR comparison on classifiers with different features.

unseen users to evaluate the performance of the pretrained CNN on unseen users. To conduct the evaluation, we randomly select m users for CNN training and choose some users from the rest (88 – m) for classifier training and testing.

Table 2: Mean ERR (%) of the four classifiers on different features.

Features	KNN	OCSVM	RF	DT
Handcrafted features	9.87	10.57	9.35	11.45
CNN-learned features	6.56	3.86	6.94	9.34

Table 3: Accuracy (%) of handcrafted features and CNN-learned feature without any data augmentation.

Features	10 s	20 s	30 s	40 s	50 s	60 s
Handcrafted features	93.35	95.47	96.89	98.7	99.12	99.35
CNN-learned features	98.12	99.25	99.36	99.45	99.58	99.52
Handcrafted features (DA)	99.15	99.25	99.45	99.62	99.72	99.84
CNN-learned features (DA)	99.35	99.46	99.56	99.72	99.88	99.98

We set $m = 40$ and unseen users as 10, 20, 30, and 40, respectively. Then, we use the trained authentication model of the 4 classifiers for testing, and the authentication results are shown in Table 5 as follows:

From Table 5 we obtain such a result. With unseen users increasing, our authentication system keeps more than 90% accuracy on 4 classifiers. Specifically, when $n = 20$, in these four classifiers, the OCSVM classification shows the best accuracy of 98.72%. The KNN classifier shows the best accuracy of 94.54%, the RF classifier shows the best accuracy of 97.96%, and the DT classifier shows the best accuracy of 95.27%. The test on unseen user data shows that the system is robust.

4.7. Comparing with Other Existing Methods. To verify the differences between the proposed method and state-of-the-art methods, we analyze the differences with other authentication methods. As shown in Table 6, this table shows the existing methods, data sources, and authentication results of the methods to which they belong.

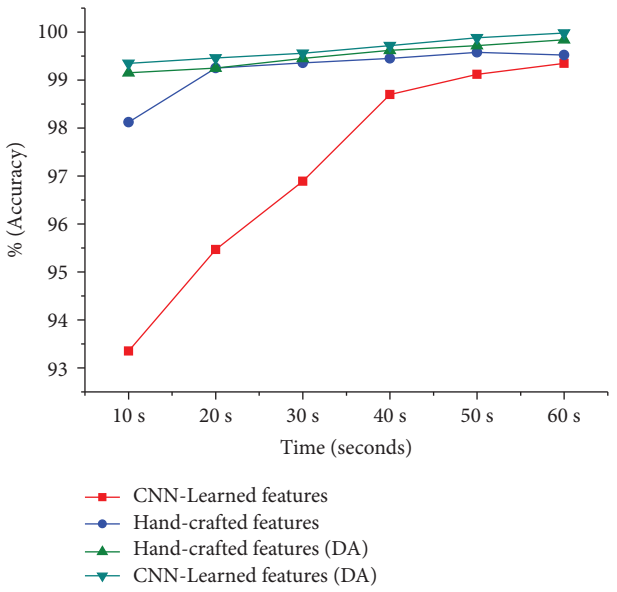

Figure 7: Accuracy (%) of handcrafted features and CNN-learned features without any data augmentation.

Table 4: Mean EER (%) on different user activities for different classifiers.

Activities	RF	OCSVM	KNN	DT
Sitting + reading	**4.32**	5.52	6.52	5.52
Sitting + writing	9.87	**3.87**	4.99	4.43
Sitting + map	6.77	**6.39**	6.64	6.93
Walking + reading	9.78	**4.58**	10.52	13.06
Walking + writing	10.78	**5.14**	11.52	9.49
Walking + map	**7.10**	6.36	13.38	10.48

Our proposed method has the smallest mean ERR of 3.87% on the OCSVM classifier. The mean EER of the RF classifier on sitting + reading and walking + map is 4.32% and 7.1%, respectively, and the mean EER of the OCSVM classifier on sitting+map, walking + reading, and walking + writing is 6.39%, 4.58%, and 5.14%, respectively.

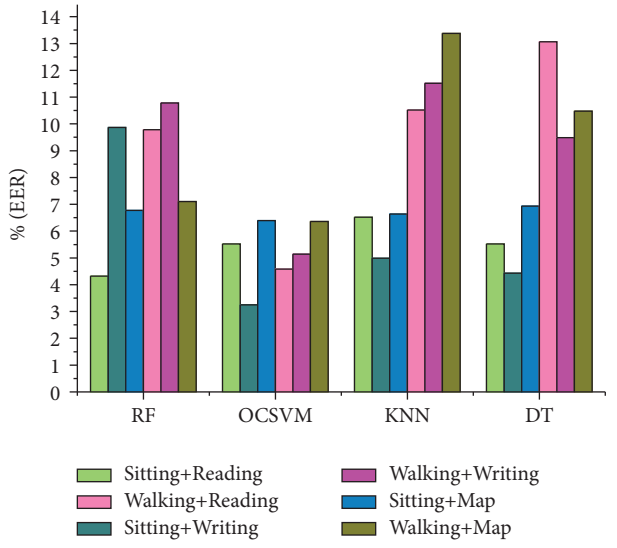

Figure 8: EER on different user activities for different classifiers.

Table 5: Accuracy (%) for unseen users on OCSVM.

Classifier\Unseen users	10	20	30	40
KNN	92.47	93.47	96.45	92.54
OCSVM	96.83	**98.72**	98.11	97.17
RF	92.28	97.17	96.39	97.27
DT	92.16	93.75	94.53	95.63

When n = 20, in these four classifiers, the OCSVM classification shows the best accuracy of 98.72%.

Table 6: Comparing with other existing methods.

Methods	Data sources	Results
SensorCA [14]	Accelerometer, gyroscope, and magnetometer	EER: 3.70% (SVM-RBF)
SensorAuth [16]	Accelerometer and gyroscope	EER: 6.29%
CWGAN [26]	Accelerometer, gyroscope, and magnetometer	EER: 3.72%
Our method	Accelerometer, gyroscope, and magnetometer	EER: 3.68% (OCSVM)
HMOG [7]	Accelerometer, gyroscope, and magnetometer	EER: 7.16% (walking) and 10.05% (sitting)

In Table 6, SensorCA [14] applied matrix rotation for accelerometer, gyroscope, and magnetometer data, resulting in an EER of 3.70% for the SVM-RBF classifier. SensorAuth [16] explored five data augmentation methods of permutation, sampling, scaling, cropping, and jittering to create additional accelerometer and gyroscope data and achieved an EER of 6.29% with a dataset size of 200 by combining the five methods. The CWGAN [26] achieved an optimal EER of 3.72% with a data volume of 700, but it did not take into account the impact of behavioural changes. Based on the HMOG dataset, [7] an EER of 7.16% is obtained for walking and an EER of 10.05% is obtained for sitting. The EER of the proposed method on different user activity data is between 3.68% and 6.39%. Note that this table only provides preliminary comparison results, and each method has its advantages and disadvantages under different conditions.

5. Conclusions and Future Work

In this article, we propose a robust continuous authentication system using a WGAN on smartphones for sensor data augmentation, which mainly focuses on the use of smartphones for user authentication under different activities. Different from the state-of-the-art authentication methods, the existing method usually requires the user to swipe the screen and press the keys of the mobile phone under a specified activity. Our method collects motion sensor (an accelerometer, a gyroscope, and a magnetometer) data under different activities. To obtain good authentication performance, we utilize the WGAN model to create

additional data from raw sensor data under different activities for data augmentation. With the augmented data, we design a convolutional neural network to learn and extract deep features from sensor data, then use four classifiers of RF, OCSVM, DT, and KNN to train these features, and finally test on different user activity data. The experiments are evaluated and analyzed on the HMOG dataset, and the results show that the EER of the authentication system is between 3.68% and 6.39% on the sensor data with a time window of 2 s.

However, since the accuracy for continual learning on the WGAN still has room for improvement, in the future, we will look into new techniques to generate high quality of samples that best represent the real data while also encouraging the diversity.

Data Availability

The research data supporting the results of this study are available at https://www.cs.wm.edu/~qyang/hmog.html.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant no. 62102042, the National Key Research and Development Program of China under grant no. 2021YFB3101500, and the National and the Key Scientific and Technological Innovation Projects in Shandong Province of China under grant no. 2019JZZY010110.

References

[1] C. Shen, Y. Zhang, X. Guan, and R. A. Maxion, "Performance analysis of touch-interaction behavior for active smartphone authentication," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 498–513, Mar 2016.

[2] S. Amini, V. Noroozi, A. Pande, S. Gupte, P. S. Yu, and C. Kanich, "DeepAuth: a framework for continuous user re-authentication in mobile apps," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 2027–2035, Torino, Italy, October, 2018.

[3] L. Tran and D. Choi, "Data augmentation for inertial sensor-based gait deep neural network," *IEEE Access*, vol. 8, Article ID 12364, 2020.

[4] C. X. Lu, D. Bowen, Z. Peijun et al., "Deepauth: in-situ authentication for smartwatches via deeply learned behavioural biometrics," in *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, Singapore, October, 2018.

[5] S. Amini, N. Vahid, P. Amit, G. Satyajit, S. Y. Philip, and K. Chris, "Deepauth: a framework for continuous user re-authentication in mobile apps," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Torino, Italy, October, 2018.

[6] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, "Deep learning-based gait recognition using smartphones in the wild," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3197–3212, 2020.

[7] Z. Sitova, J. Sedenka, Q. Yang et al., "HMOG: new behavioral biometric features for continuous authentication of smartphone users," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 877–892, 2016.

[8] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, 2019.

[9] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 843–852, Venice, Italy, October, 2017.

[10] J. Wang and L. Perez, "The effectiveness of data augmentation in image classification using deep learning," 2017, https://arxiv.org/abs/1712.04621.

[11] C. N. Vasconcelos and B. N. Vasconcelos, "Convolutional neural network committees for melanoma classification with classical and expert knowledge based image transforms data augmentation," vol. 1, 2017, https://arxiv.org/abs/1702.07025.

[12] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Trans. Audio Speech Lang. Process*, vol. 23, no. 9, pp. 1469–1477, 2015.

[13] Z. Tüske, P. Golik, D. Nolden, R. Schlüter, and H. Ney, "Data augmentation feature combination and multilingual neural networks to improve ASR and KWS performance for low-resource languages," in *Proceedings of the 15th Annual Conference of the International Speech Communication Association*, Dublin, Ireland, September, 2014.

[14] Y. Li, H. Hu, G. Zhou, and S. Deng, "Sensor-based continuous authentication using cost-effective kernel ridge regression," *IEEE Access*, vol. 6, Article ID 32554, 2018.

[15] A. Buriro, F. Ricci, and B. Crispo, "SWIPEGAN: swiping data augmentation using generative adversarial networks for smartphone user authentication," in *Proceedings of the 3rd ACM workshop on wireless security and machine learning*, pp. 85–90, Abu Dhabi, UAE, June, 2021.

[16] Y. Li, H. Hu, and G. Zhou, "Using data augmentation in continuous authentication on smartphones," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 628–640, 2019.

[17] H. Zhang, J. Liu, K. Li, H. Tan, and G. Wang, "Gait learning based authentication for intelligent things," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4450–4459, 2020.

[18] M. Arif, M. Bilal, A. Kattan, and S. I. Ahamed, "Better physical activity classification using smartphone acceleration sensor," *Journal of Medical Systems*, vol. 38, no. 9, p. 95, 2014.

[19] I. Goodfellow, P.-A. Jean, M. Mehdi et al., "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems*, Montreal, Canada, 2014.

[20] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2017, https://arxiv.org/abs/1711.04340.

[21] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," 2017, https://arxiv.org/abs/1701.04862.

[22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, https://arxiv.org/abs/1701.07875.

[23] S. Yao, S. Hu, Y. Zhao, Z. Aston, and A. Tarek, "Deepsense: a unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th international conference on world wide web*, pp. 351–360, Perth, Australia, April, 2017.

[24] W. H. Lee and R. B. Lee, "Multi-sensor authentication to improve smartphone security," in *Proceedings of the 2015 International conference on information systems security and privacy (ICISSP)*, pp. 1–11, IEEE, Angers, France, February, 2015.

[25] D. Shi, D. Tao, J. Wang et al., "Fine-grained and context-aware behavioral biometrics for pattern lock on smartphones," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 1–30, 2021.

[26] Y. Li, J. Luo, S. Deng, and G. Zhou, "CNN-based continuous authentication on smartphones with conditional Wasserstein generative adversarial network," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5447–5460, 2022.