CNN-based Continuous Authentication on Smartphones with Conditional Wasserstein Generative Adversarial Network

Yantao Li, *Member, IEEE,* Jiaxing Luo, Shaojiang Deng, and Gang Zhou, *Senior Member, IEEE*

***Abstract*—With the widespread usage of mobile devices, the authentication mechanisms are urgently needed to identify users for information leakage prevention. In this paper, we present CAGANet, a CNN-based continuous authentication on smartphones using a conditional Wasserstein generative adversarial network (CWGAN) for data augmentation, which utilizes smartphone sensors of the accelerometer, gyroscope and magnetometer to sense phone movements incurred by user operation behaviors. Specifically, based on the preprocessed real data, CAGANet employs CWGAN to generate additional sensor data for data augmentation that are used to train the designed CNN. With the augmented data, CAGANet utilizes the trained CNN to extract deep features and then performs PCA to select appropriate representative features for different classifiers. With the CNN-extracted features, CAGANet trains four one-class classifiers of OC-SVM, LOF, IF and EE in the enrollment phase and authenticates the current user as a legitimate user or an impostor based on the trained classifiers in the authentication phase. To evaluate the performance of CAGANet, we conduct extensive experiments in terms of the efficiency of CWGAN, the effectiveness of CWGAN augmentation and the designed CNN, the accuracy on unseen users, and comparison with traditional augmentation approaches and with representative authentication methods, respectively. The experimental results show that CAGANet with the IF classifier can achieve the lowest EER of 3.64% on 2-second sampling data.**

***Index Terms*—Continuous authentication, conditional Wasserstein GAN, CNN, deep feature, equal error rate (EER).**

I. Introduction

WITH the rapid development of mobile communication technologies (e.g., the mobile Internet, the Internet of Things, and smart interconnection), mobile devices (e.g., smartphones, smartwatches, and tablets) have been widely developed and popularized, and have deeply affected people's life and work. For instances, people prefer to use mobile devices to place orders and make payments in daily life, store photos and chat messages for personal life, and send classified documents and emails for work communication. However, more and more sensitive information stored on mobile devices suffers from information leakage. Thus, there is an increasing and urgent need for security mechanisms to identify the mobile device users for protecting their personal and privacy data on devices [1], [2]. The existing authentication mechanisms can be broadly categorized into knowledge-based authentication, physiological biometrics-based authentication, and behavioral biometrics-based authentication. Specifically, the knowledge-based authentication schemes (e.g., passwords, PINs, and graphical patterns) mainly require users to input predefined knowledge, which are vulnerable to various attacks, such as smudge attack [3] and shoulder surfing attack [4]. The physiological biometrics-based mechanisms, such as fingerprints (Touch ID) [5], face patterns (Face ID) [6], and voice [7], mainly require user direct participation based on the unique physicological features, which suffer from capturing attack [8], replaying attack [9], and spoofing attack [10]. The behavioral biometrics-based approaches (e.g., touch gestures [11], gait [12], and GPS patterns [13]) can non-intrusively identify users while using the mobile devices based on the invariant behavioral features, which use built-in sensors (e.g., the accelerometer, gyroscope, and magnetometer) to sample users' invariant behavioral data for user authentication. However, one-time authentication mechanisms (e.g., knowledge-based mechanisms and physiological biometrics-based mechanisms) share a common problem that they authenticate the user only at the initial logging-in session and do not re-authenticate the user until the user logs out. This could pose a critical security weakness for mobile devices that attackers can easily access to everything on unattended mobile devices without logging out [14]. To address this problem, mobile devices must continuously monitor and authenticate the users after the initial logging-in session. The behavioral biometrics-based authentication shows a promising solution that can continuously identify the users without their cooperation during their operation on the devices.

The behavioral biometrics-based continuous authentication has been widely studied in recent years, aiming at identifying invariant features of human behaviors during different activities [32]. However, they are currently facing two challenges: limited training data and inefficient feature representation. Data collection commonly costs extensive efforts from organizers and participants including time and energy but the continuous authentication systems usually require sufficient data to train a classifier or a deep-learning based extractor. The authentication accuracy significantly lies on the feature representation ability and the systems require efficient feature extractors to learn more discriminative features to ensure high accuracy. To overcome these challenges, the state-of-

Y. Li, J. Luo and S. Deng are with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: yantaoli@cqu.edu.cn; sj_deng@cqu.edu.cn).

G. Zhou is with the Department of Computer Science, William & Mary, Williamsburg, VA 23185, USA (e-mail: gzhou@cs.wm.edu).

Manuscript received April XX, 2021; revised August XX, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62072061 and by the Fundamental Research Funds for the Central Universities under Grant 2021CDJQY-026.

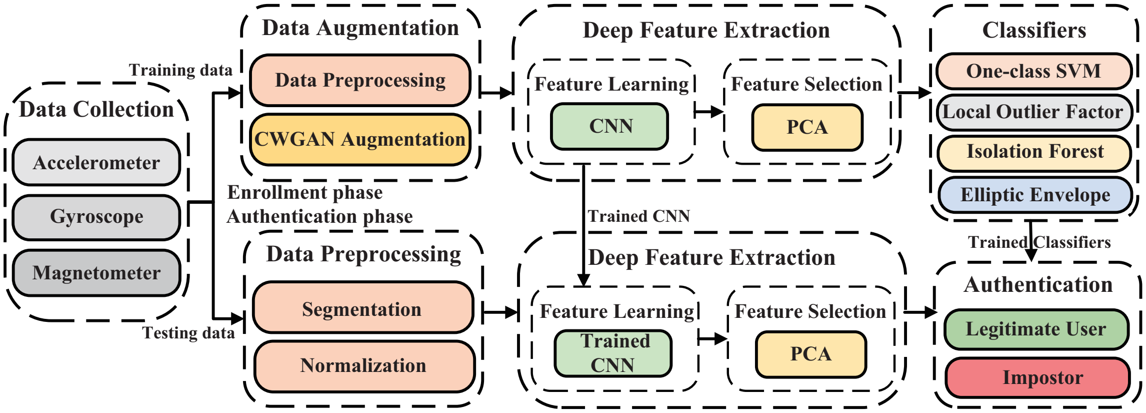

Fig. 1: Architecture of CAGANet

the-art authentication methods have been proposed. For limited training data, the data augmentation approach has been utilized in continuous authentication for creating additional training data [29], [30], [31], [32], [33], [34], such as permutation, sampling, scaling, cropping, jittering, flipping, and rotation. However, most of these augmentation approaches are primitively designed for image data augmentation, such as palmprint recognition [16] and finger-vein recognition [17]. For inefficient feature representation, deep learning approaches have been exploited into continuous authentication to extract deep features [35], [36], [37], [38], [39], [40], [41], such as DNN, CNN, and RNN. However, most of these works focus on physiological biometrics-based features, such as respiratory biometrics [37] and fingertip-touch [39].

Different from the existing works, this paper aims to provide a conditional Wasserstein generative adversarial network (CWGAN) for sensor data augmentation and specially design a convolutional neural network (CNN) based on a basic unit and a basic unit for spatial down sampling for discriminative deep feature extraction in a continuous authentication system. In this paper, we present CAGANet, a CNN-based continuous authentication system using a CWGAN for data augmentation that leverages the accelerometer, gyroscope and magnetometer on smartphones to monitor users' behavioral patterns. CAGANet is composed of two phases of the enrollment phase and the continuous authentication phase. In the enrollment phase, CAGANet learns the profile of a legitimate user where the designed CNN is trained by exploiting the CWGAN-augmented training data and the one-class classifiers are trained by CNN-extracted deep features. In the continuous authentication phase, the current user is continuously authenticated as a legitimate user or an impostor by utilizing the trained CNN and trained classifiers on the testing data. If it is classified as a legitimate user, CAGANet will allow the user to continue using the smartphone; Otherwise, it requires the user to provide the legitimate identification to continue access the smartphone.

The main contributions of this work are summarized as follows.

- We present CAGANet, a CNN-based continuous authentication system using a CWGAN for data augmentation that leverages the accelerometer, gyroscope and magnetometer on smartphones. CAGANet consists of five modules: data collection, data augmentation, deep feature extraction, classifiers, and authentication.
- We utilize CWGAN to generate additional sensor data for CNN training, and specially design a CNN based on a basic unit and a basic unit for spatial down sampling to extract representative deep features. Extensive experiments are conducted to demonstrate how CWGAN can generate high-quality sensor data and illustrate how the designed CNN can learn discriminative features, respectively.
- We evaluate the authentication performance of CAGANet on four one-class classifiers of OC-SVM, LOF, IF and EE, and the experimental results demonstrate that CAGANet outperforms other representative approaches and achieves the lowest EER of 3.64% with the IF classifier.

The rest of this paper is organized as follows. Section II presents the overview of CAGANet. In Section III, we introduce the data collection and the data preprocessing for CWGAN training and testing. Section IV details CWGAN augmentation for deep feature extraction. In Section V, we provide the CNN-based feature extraction method composed of feature learning and feature selection. We elaborate the authentication with four classifiers in Section VI. In Section VII, we describe the experimental setting and excessively evaluate the performance of CAGANet. We review the state-of-the-art on data augmentation and deep learning in Section VIII and Section IX concludes this work.

II. CAGANet Overview

We present the architecture of CAGANet, a CNN-based continuous authentication system using a conditional Wasserstein generative adversarial network (CWGAN) for data augmentation. CAGANet employs smartphone built-in sensors, including the accelerometer, gyroscope, and magnetometer, to sense phone movements incurred by user operation behaviors.

As illustrated in Fig. 1, CAGANet architecture consists of two phases: 1) the enrollment phase, where CAGANet learns a profile of a legitimate user by utilizing the training data to train CWGAN and CNN, and 2) the continuous authentication phase, where the system authenticates users by exploiting the trained CNN and the trained four one-class classifiers on the testing data. Moreover, CAGANet is composed of five modules, including the data collection, data augmentation, deep feature extraction, classifiers, and authentication. The data collection module utilizes the three sensors to capture users' every subtle operation behaviors on their phones and sample the corresponding behavioral data instantaneously. The data augmentation module first segments and normalizes the the sampled data, and then uses CWGAN to augment them. Note that data augmentation module only performs on training data in the enrollment phase to generate additional sensor data for CNN training. The deep feature extraction module learns the CNN-based features and then selects representative features by PCA. With the selected deep features, the classifier module trains the four one-class classifiers and generates the legitimate user's profile from the training data. Based on the trained CNN and trained classifiers, the authentication module classifies the current user as a legitimate user or an impostor with the testing data. CAGANet will allow the continuous smartphone usage if it is a legitimate user; otherwise, it requires the user's identification, such as initial login inputs.

III. Data Collection and Preprocessing

In this section, we first describe how CAGANet collects the dataset and then preprocesses the collected data for both CWGAN and CNN training.

A. Data Collection

We select three smartphone built-in sensors of the accelerometer, gyroscope, and magnetometer, to sense a user's behavioral motion on the phone. The accelerometer and gyroscope are motion sensors, which capture a user's coarse-grained and fined-grained motion patterns, respectively, while the magnetometer is a position sensor, which determines the phone's physical position in the real frame of reference.

A user's operation on a smartphone triggers the data collection module starting to collect the raw sensor data from the accelerometer, gyroscope, and magnetometer, respectively, for a time period t with a sampling rate f. To collect data for CAGANet training, we developed a data collection tool for Android phones to record the real-time behavioral data while the participants operate on the phones. We recruited 100 participants (53 male and 47 female) to conduct three designed tasks on the phones with developed tools: 1) document reading, 2) text production, and 3) navigation on a map to locate a destination. These tasks lasting 5 to 15 minutes were randomly assigned once the participants logged into the developed tool, and each participant was expected to perform 24 sessions (8 reading sessions, 8 writing sessions, and 8 map navigation sessions) with totally 2 to 6 hours of behavior traits.

In the enrollment phase, we select the sensor readings of the accelerometer, gyroscope, and magnetometer from 88 participants (44 male and 44 female) with the sampling rate $f = 100Hz$ and select 100 minutes of the data for each user with a $t = 2$ or 5-second window size for training. In the authentication phase, for a time period t, n $(n = t \times f)$ samples can be collected, and each synchronized sample can be denoted as $(x_a, y_a, z_a, x_g, y_g, z_g, x_m, y_m, z_m)^T \in \mathbb{R}^9$, where x, y, z represent the three axes of a sensor, and a, g, m indicate the accelerometer, gyroscope, and magnetometer, respectively.

In the experiments, we divide the 88 participants' data into two groups, where the data of 68 participants are used for CWGAN augmentation and classifier training in the enrollment phase and the data of the rest 20 are used as the input of trained CNN and trained classifiers for CWGANet testing in the authentication phase.

B. Data Preprocessing

For a time period t, CAGANet can collect n samples of time domain data for the accelerometer, gyroscope, and magnetometer, which can be represented by a $d \times n$ matrix:

$$D^i = \begin{bmatrix} x_a^{i1} & y_a^{i1} & z_a^{i1} & x_g^{i1} & y_g^{i1} & z_g^{i1} & x_m^{i1} & y_m^{i1} & z_m^{i1} \\ x_a^{i2} & y_a^{i2} & z_a^{i2} & x_g^{i2} & y_g^{i2} & z_g^{i2} & x_m^{i2} & y_m^{i2} & z_m^{i2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ x_a^{in} & y_a^{in} & z_a^{in} & x_g^{in} & y_g^{in} & z_g^{in} & x_m^{in} & y_m^{in} & z_m^{in} \end{bmatrix}^T,$$

where $d = 9$ and $n = t \times f$. Then, all the dataset can be expressed as: $D_{enroll} = [D^1, D^2, \cdots, D^{num}]^T$, where num indicates the number of time periods.

For CWGAN training, we normalize each element d_{enroll} in D_{enroll} into $(-1, 1)$ by $d'_{enroll} = \frac{d_{enroll} - min(D_{enroll})}{max(D_{enroll}) - min(D_{enroll})}$ and $d_{cwgan} = \frac{d'_{enroll} - 0.5}{0.5}$, and obtain the normalized data D_{cwgan}, which are used for CWGAN augmentation.

For CNN training, with the augmented data, we normalize each user's data including x, y, z- axes into $(-1, 1)$ in D_{enroll}, and obtain the normalized data D_{cnn}.

IV. CWGAN Augmentation

Generative adversarial networks (GANs) proposed by I. Goodfellow *et al.* typically consist of two adversarial networks: a generator and a discriminator. The generator produces realistic-like data to confuse the discriminator, while the discriminator tries to distinguish whether a sample comes from the real or realistic-like data [15]. GANs have been exploited into biometric recognition tasks for data augmentation, such as palmprint recognition [16], and finger-vein recognition [17]; however, most of the existing GAN-based approaches focus on image augmentation. In this section, we design a conditional Wasserstein GAN (CWGAN) for smartphone sensor data augmentation. We first present the CWGAN architecture, and then detail CWGAN training procedures.

A. CWGAN

Given real sensor data x_r with distribution p_r and generated data x_g with distribution p_g, the generator G produces realistic-like x_g to confuse the discriminator D, while the discriminator D intends to distinguish whether a sample comes from x_r or x_g. The generator and discriminator are both parameterized as convolutional neural networks (CNN). The adversarial training procedure can be formulated as a minimax problem, as shown in Eq. (1):

$$\begin{aligned} \min_{\theta_G} \max_{\theta_D} & L(p_r, p_g) = \\ & \mathbb{E}_{x_r \sim p_r}[\log D(x_r)] + \mathbb{E}_{x_g \sim p_g}[\log D(x_g)], \end{aligned} \tag{1}$$

TABLE I: Generator structure

Layer	Output	#Kernel	KSize	Stride	Padding
Sensor	(ND + CN) $\times 1 \times 1$	-	-	-	
ConvTranspose2d (BN+ReLU)	$64 \times 11 \times 3$	64	(11,3)	(1,1)	(0,0)
ConvTranspose2d (BN+ReLU)	$128 \times 24 \times 3$	128	(4,3)	(2,1)	(0,1)
ConvTranspose2d (BN+ReLU)	$256 \times 49 \times 3$	256	(3,3)	(2,1)	(0,1)
ConvTranspose2d (BN+ReLU)	$512 \times 99 \times 3$	512	(3,3)	(2,1)	(0,1)
ConvTranspose2d	$3 \times 200 \times 3$	3	(4,3)	(2,1)	(0,1)
Tanh	-	-	-	-	-

TABLE II: Discriminator structure

Layer	Output	#Kernel	KSize	Stride	Padding
Sensor	$(3 + \text{CN}) \times 200 \times 3$	-	-	-	-
Conv2d (LeakyReLU)	$32 \times 200 \times 3$	32	(3,3)	(1,1)	(1,1)
Conv2d (LeakyReLU)	$64 \times 100 \times 3$	64	(3,3)	(2,1)	(1,1)
Conv2d (LeakyReLU)	$128 \times 50 \times 3$	128	(3,3)	(2,1)	(1,1)
Conv2d (LeakyReLU)	$256 \times 25 \times 3$	256	(3,3)	(2,1)	(1,1)
Conv2d (LeakyReLU)	$512 \times 12 \times 3$	512	(3,3)	(2,1)	(0,1)
Linear (LeakyReLU)	1024	-	-	-	-
Linear	1	-	-	-	-

TABLE III: CNN Architecture

Layer	Output	#Kernel	KSize	Stride	Parameter
Sensor	$3 \times 200 \times 3$	-	-	-	
Conv2d (BN+ReLU)	$24 \times 100 \times 3$	24	(3,3)	(2,1)	696
Stage 2	$48 \times 50 \times 3$	48	-	(2,1)	3912
	$48 \times 50 \times 3$	48	-	(1,1)	
Stage 3	$96 \times 25 \times 3$	96	-	(2,1)	13584
	$96 \times 25 \times 3$	96	-	(1,1)	
Stage 4	$192 \times 13 \times 3$	192	-	(2,1)	50208
	$192 \times 13 \times 3$	192	-	(1,1)	
Conv2d (BN+ReLU)	$1024 \times 13 \times 3$	1024	(1,1)	(1,1)	198656
AvgPool	$1024 \times 1 \times 1$	-	(13,3)	-	-
FC	$1 \times \text{CN}$	-	-	-	61500

where θ_G and θ_D indicate the parameters of the generator and discriminator, respectively. p_g is implicitly defined by $x_g = G(x_z)$, where x_z is initially sampled from a Gaussian noise distribution.

Based on Eq. (1), we apply Kantorovich-Rubinstein duality of Earth-Mover distance in Wasserstein GANs [18], and enforce Lipschitz constraint with gradient penalty instead of weight clipping to directly constrain the gradient norm [19]. Then, the CWGAN used for data augmentation can be expressed by Eq. (2):

$$\begin{aligned}\min_{\theta_G}\max_{\theta_D} & L(p_r, p_g, y_r) = \\ & \mathbb{E}_{x_r \sim p_r, y_r \sim p_y}[D(x_r|y_r)] - \mathbb{E}_{x_g \sim p_g, y_r \sim p_y}[D(x_g|y_g)] \\ & - \lambda \mathbb{E}_{\hat{x} \sim \hat{p}, y_r \sim p_y}[(||\nabla_{\hat{x}|y_r} D(\hat{x}|y_r)||_2 - 1)^2],\end{aligned} \tag{2}$$

where λ is a hyperparameter that controls the trade-off between original objective and gradient penalty, $\hat{x}$ indicate the data points sampled from the straight line between real distributions p_r and p_g with $\hat{x} = \alpha x_r + (1 - \alpha) x_g (\alpha \in [0, 1])$, and y_r denotes a label fed into both discriminator D and generator G. In G we concatenate p_z with p_y while in D concatenate both p_r and p_g with p_y to construct a hidden representation controlling the categories of generated data [20].

B. CWGAN Architecture

We design the architecture of the CWGAN including a generator structure and a discriminator structure, as shown in Tables I and II, respectively. As shown in Table I, the generator consists of five 2D transposed convolution layers and one Tanh active function layer. We apply the batch normalization (BN) and Rectified Linear Unit (ReLU) after each of the first four operations. In Table I, 'ND' indicates the noise dimension (noise_dim) and 'CN' represents class number for training (class_num), respectively. As shown in Table II, the discriminator comprises five 2D convolution layers, and two linear layers. We apply a leaky version of a Rectified Linear Unit (LeakyReLU) on the five 2D convolution layers, and the following one linear layer.

Based on the normalized data D_{cwgan}, we randomly separate the 88 participants' data into two groups, one group with U_{fa} ($U_{fa} = 68$ in experiments) participants' data and the other with $U_{ft} = 88 - U_{fa}$. The U_{fa} participants' data are first augmented by CWGAN and then used for the designed CNN training and validation while the U_{ft} participants' data are just extracted features by the trained CNN for classifier training. We label the U_{fa} participants by one-hot encoding and obtain a label set $l[\texttt{batchsize}, U_{fa}]$ for them. In generator G, Gaussian noise $x_z[\texttt{batchsize}, 100]$ with the standard uniform distribution is concatenated with $l[\texttt{batchsize}, U_{fa}]$ to generate an original input $x_g[\texttt{batchsize}, 100 + U_{fa}]$. The output of G is $(\texttt{batchsize}, 3 \times 200 \times 3)$, where the first '3' indicates the three sensors, '200' represents 2-second data, and the last '3' denotes the three axes of a sensor. In discriminator D, the input is the output of G $(\texttt{batchsize}, 3 \times 200 \times 3)$, and the output $512 \times 12 \times 3$ is concatenated with label $l[\texttt{batchsize}, U_{fa}]$. The output of D is the probability that the generated sample belongs to the real sample.

For CWGAN training, we train the generator once for every 20 times of the discriminator training, for total 50 times, with optimizer `RMSprop`, and learning rate 0.00005. The `batchsize` is set as 128 and the hyperparameter λ for gradient penalty is set to 10.

V. Deep Feature Extraction

In this section, we provide a CNN-based deep feature extraction method that is composed of feature learning and feature selection. Before describing the method, we elaborate the design of the CNN for feature learning.

A. CNN Design

Inspired by ShuffleNet V2 [21], we design the architecture of a CNN, as presented in Table III. The proposed CNN mainly consists of a 2D convolutional layer (Conv2d), a stack of ShuffleNet V2 units grouped into three stages (Stage 2, Stage 3, and Stage 4), another Conv2d layer, and a full connection layer (FC). We adopt BN and ReLU right after each Conv2d, and an average pooling (AvgPool) after the second Conv2d layer. In addition, Stages 2, 3, and 4 have the same building block structure which is comprised of a basic unit for spatial down sampling followed by a basic unit.

Specifically, the basic unit begins with splitting the channels C into two identical branches, where one branch remains as identity with $C/2$ channels. The other branch is a bottlenet unit with depthwise convolution (DWConv) (1×1 Conv followed by 3×3 DWConv followed by 1×1 Conv, with BN and ReLU) [22]. Then, the two branches are concatenated with C channels and divided into g_b subgroups, on which the channel shuffle operation is applied by reshaping the output channel dimension into (g_b, n), transposing and flattening it back as the basic unit outputs, where $C = g_b \times n$. The basic unit for spatial down sampling starts with two identical (C-channel) branches, where one consists of 3×3 DWConv with stride 2 followed by 1×1 Conv with BN and ReLU, and the other one is composed of a bottlenet unit with DWConv (stride $= 2$). Then, the two branches are concatenated with $2C$ channels and divided into g_d subgroups, and the channel shuffle operation is applied as the unit output.

B. Feature Learning

We utilize the designed CNN to learn and extract discriminative deep features on the normalized CWGAN-augmented data D_{cnn}. As demonstrated in Table III, there are 1800 (3 sensors $\times 2$ seconds $\times 100Hz \times 3$ axes) samples in 2 seconds. In the first Conv2d layer, there are 24 filters with the size of 3×3 and stride $= 2$. In Stage 2, we apply the basic unit for spatial down sampling with 48 filters ($2C$), stride $= 2$ and $g_d = 2$, and then the basic unit with 48 filters, stride $= 1$ and $g_b = 2$. In Stage 3, we exploit the same structure with 96 filters and use 192 filters in Stage 4. In the second Conv2d layer, there are 1024 filters with the size of 1×1. The AvgPool layer with the size of 13×3 is used to decrease the channel output dimensions and extract features. The FC layer is exploited to classify the inputs into a finite number of classes. The total CNN architecture contains 328556 parameters. The designed CNN learns 1024 deep features (AvgPool layer) for the sensors of the accelerometer, gyroscope and magnetometer.

C. Feature Selection

With the CNN-learned deep features, we utilize the principal component analysis (PCA) to select the deep features with high discriminability for classifiers. As illustrated in the CAGANet architecture (Fig. 1), we exploit four classifiers to conduct the classification with the selected deep features. According to the experiment in VII-B, PCA selects 15 deep features for one-class support vector machine, 15 for local outlier factor, 50 for isolation forest, and 150 for elliptic envelope, respectively.

VI. Authentication with Four Classifiers

With the CNN-learned and selected deep features, we utilize four one-class classifiers for training and testing: a) One-class support vector machine (OC-SVM), b) local outlier factor (LOF), c) isolation forest (IF), and d) elliptic envelope (EE). Note that we exploit the four one-class classifiers to show the generality of the CNN-learned features. Specifically, OC-SVM exploits a kernel function to map data into a high-dimensional space and considers the origin as the only sample from other classes, where the kernel function is a radial basis function. LOF measures the local deviation of the data point to its neighbors, which decides whether a data point is an outlier using the local density estimated by k-nearest neighbors based on a given distance metric. A data point can be regarded as an outlier if its local density is substantially lower than its neighbors [24]. IF detects abnormal data points by subsampling the dataset to construct iTrees and further integrate multiple iTrees into a forest to detect abnormal data. A data point can be viewed as an abnormal one if these random trees collectively produce shorter path lengths for it [25]. EE models the data as a high dimensional Gaussian distribution with possible covariances between feature dimensions and attempts to find an boundary ellipse that contains most of the data using FAST-Minimum covariance determinate to estimate the size and shape of the ellipse. A data point is classified as anomalous one if it is outside of the ellipse [26].

In the enrollment phase, the classifiers are trained by the selected deep features and CAGANet generates the legitimate user's profile from the training data. In the continuous authentication phase, the trained classifiers classify the selected testing deep features. Based on the trained classifiers and testing data, CAGANet authenticates the current user as a legitimate user or an impostor. If the user is classified as an impostor, CAGANet will require initial login inputs; Otherwise, it will allow the continuous usage of the smartphone and meanwhile continuously authenticates the user.

VII. Evaluation

In this section, we evaluate the performance of CAGANet based on the collected 88 users' dataset, where the randomly-selected 68 users are used for training and the rest 20 users (unseen users) are used for testing. To evaluate the authentication performance of CAGANet, we first introduce the experimental settings, and then conduct extensive experiments. Specifically, for the evaluation experiments, we first determine the optimal deep feature numbers for different classifiers and find the best parameter combinations for them. Then, we evaluate the quality of CWGAN-generated sensor data. Next, we investigate the authentication accuracy of CAGANet on CWGAN augmentation and the designed CNN, respectively, and then explore the accuracy on unseen users. Finally, we compare CAGANet with traditional augmentation approaches and with representative authentication methods, respectively.

A. Experimental Settings

1) CNN and classifier training: We randomly selected 68 users' data out of 88 to train the designed CNN. Based on the trained CNN, we choose the rest 20 users' data (deep features extracted by the trained CNN) to train the four classifiers of OC-SVM, LOF, IF, and EE.

For CNN training, 80% randomly-selected 68 users' data are used for training and the rest 20% for validation, with a batch size of 256. For each batch-size training data, we feed them to the designed CNN, calculate the loss utilizing cross entropy based on the output, then perform back-propagation, and finally update parameters for the learning rate. With an initial value of 0.001, we use Stochastic Gradient Descent optimizer to update the learning rate by reducing 50% if the validation loss of current epoch is higher than the previous one until the maximum 300 epochs or the validation loss remains for 10 continuing epochs.

For classifier training, we randomly select one user from the 20 users as the legitimate user and the rest 19 users as impostors. We utilize ten-fold cross-validation on each legitimate user. That is, the positive samples from the legitimate user are equally divided into 10 subsets, where 9 subsets are used as training sets and the rest one is used as a testing set. Then, the negative samples from all the impostors with the same size to positives are selected and then divided into 10 subsets, where one of them is used as the testing set. The above procedures are repeated 10 times until each subset of positive or negative samples are tested once. Finally, we repeat the ten-fold cross-validation 20 times until each of the 20 users are selected as a legitimate user once.

2) Evaluation metrics: We utilize two evaluation metrics of the accuracy and equal error rate (EER) to evaluate the effectiveness of CAGANet. We start with four basic metrics: True positive (TP) indicates that operation behaviors from legitimate users are correctly identified; True negative (TN) indicates that operation behaviors not from legitimate users are correctly declined; False positive (FP) indicates that operation behaviors not from legitimate users are incorrectly identified as legitimate; False negative (FN) indicates that operation behaviors from legitimate users are incorrectly rejected. Based on the four basic metrics, the accuracy is the percentage ratio of the total number of correct authentication against the total number of authentication, defined as: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$. EER is the point that the true acceptance rate (FAR) equals to the true rejection rate (FRR), where $FAR = \frac{FP}{FP+TN}$ and $FRR = \frac{FN}{FN+TP}$. A lower EER indicates higher authentication accuracy [27].

B. Feature Number and Classifier Parameter

To determine the optimal deep feature numbers selected by PCA for the four classifiers of the OC-SVM, LOF, IF and EE, we conduct experiments to investigate the impact of feature numbers on the classifiers. We compute the accuracy of CAGANet with different classifiers as the feature number increase from 5 to 200 over 2 seconds and 5 seconds, respectively. We tabulate the accuracy for different classifiers with varying feature numbers over 2 seconds and 5 seconds in Table IV. As shown in Table IV, for all the classifiers, the accuracy gradually increases with the growth of selected features until an optimal number and then slightly decreases for both 2 seconds and 5 seconds. Specifically, for OC-SVM and LOF classifiers, 15 deep features selected by PCA reach the highest accuracy of 97.42% (96.89%) and 97.98% (97.00%), respectively, over 2 seconds (5 seconds). For IF classifier, 50 deep features achieve the best accuracy of 97.24% (97.27%) over 2 seconds (5 seconds) while 150 deep features reach 95.91% (95.65%) for EE classifier. In particular, 2-second sampling data show a slightly better accuracy than 5-second data.

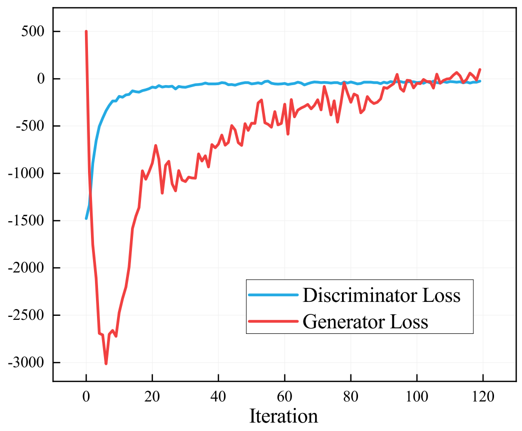

Fig. 2: Discriminator and generator loss

In addition, based on the optimal feature numbers for the classifiers, we use the grid search to find the best parameter combinations for classifiers of the OC-SVM, LOF, IF and EE. As listed in Table V, for OC-SVM classifier, we find that the radial basis function works best with $\mu = 0.01$ and $\gamma = 0.015625$ over both 2 and 5 seconds. For LOF, we utilize Manhattan distance as the Minkowski metric with the optimal parameters of $\texttt{n\_neighbors} = 20$ and $p = 1$ on 2 seconds, and $\texttt{n\_neighbors} = 10$ and $p = 2$ on 5 seconds. For IF, the optimal parameter of $\texttt{n\_estimators}$ is 100 on 2 seconds and 60 on 5 seconds. For EE, the robust location and covariance are directly computed with the FastMCD algorithm without additional treatment and the optimal parameters of $\texttt{contamination} = 0.1$ and $\texttt{assume\_centered} = \texttt{False}$ on 2 seconds, and 0.1 and $\texttt{True}$ on 5 seconds.

For 2-second or 5-second sampling data, the feature number for each classifier is the same, and the accuracy and optimal parameters are almost the same (2-second data show slightly better). Considering the time cost and memory occupancy, the following experiments are all based on 2-second sampling data.

C. Efficiency of CWGAN

We evaluate the efficiency of the designed CWGAN framework for sensor data augmentation. The quality of the CWGAN-generated sensor data can be evaluated by two indicators of discriminator loss and maximum mean discrepancy (MMD), where the CWGAN-generated sensor data are considered to have high qualities if their distributions are approximate to real distributions. Discriminator loss indicates the Earth-Mover distance between the real sensor data x_r and the generated data x_g when the network converges. That is, the CWGAN-generated data are high qualities if the discriminator loss is approximate to 0 [20]. MMD measures the distance between x_r and x_g. That is, the CWGAN-generated data are high qualities if the distance is close to 0 [28].

In the experiment, the critic value is set to 20 to ensure the discriminator is fully optimized, RMSProp optimizer is used with learning rate equal to 0.00005 and the hyperparameter λ is set to 10. The CWGAN is trained for 50 epochs with the batch size 128. The discriminator and generator loss, and

TABLE IV: Accuracy (%) for different classifiers with varying feature numbers over 2 seconds (5 seconds)

Classifier \ Number	5	15	25	35	50	100	150	200
OC-SVM	95.45 (93.89)	**97.42 (96.89)**	94.73 (96.39)	91.71 (94.79)	79.87 (91.86)	68.70 (79.24)	31.23 (68.47)	59.95 (59.26)
LOF	95.32 (92.48)	**97.98 (97.00)**	97.05 (96.77)	95.95 (96.39)	94.45 (94.78)	90.08 (89.56)	86.73 (86.66)	84.79 (85.05)
IF	89.82 (87.58)	95.37 (94.80)	96.48 (96.49)	97.08 (96.92)	**97.24 (97.27)**	96.66 (96.52)	95.42 (94.96)	93.49 (92.85)
EE	78.42 (73.20)	83.36 (76.16)	88.70 (86.30)	91.95 (90.10)	93.79 (93.05)	95.00 (94.68)	**95.91 (95.65)**	92.63 (91.50)

TABLE V: Optimal Parameter Combinations on 2 seconds (5 seconds)

Classifier	# Feature	Optimal Parameter Combination
OC-SVM	15	$\mu = 0.01$ (0.01), $\gamma = 0.015625$ (0.015625)
LOF	15	`n_neighbors` $= 20$ (10), $p = 1$ (2)
IF	50	`n_estimators` $= 100$ (60)
EE	150	`contamination` $= 0.1$ (0.1), `assume_centered` $=$ `False (True)`

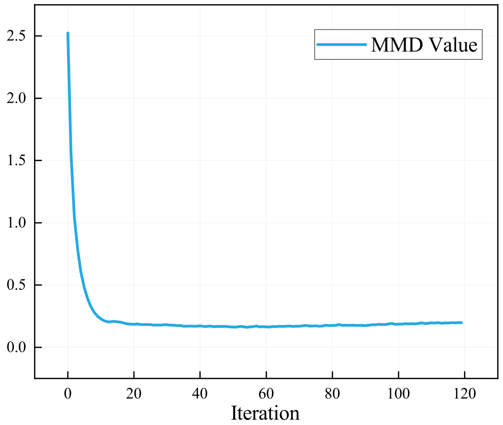

Fig. 3: MMD

MMD are illustrated in Figs. 2 and 3, respectively. Specifically, as shown in Fig. 2, the discriminator loss converges to a small value along with the increase of the training epoch. In particular, it converges to 0 when the training epoch comes to 100. The generator loss sharply decreases and then gradually converges to a small value, which indicates that it generates sensor data with high similarity to real data. As depicted in Fig. 3, the MMD has a generally similar converged tendency with the discriminator loss, which implies that adversarial-training reduces the distance between the two mapping distributions.

D. Effectiveness of CWGAN Augmentation

We investigate the authentication accuracy of CAGANet to evaluate the effectiveness of the proposed CWGAN augmentation approach on the four classifiers of the OC-SVM, LOF, IF and EE over different dataset sizes.

Fig. 4 illustrates box plots of EERs of CAGANet over different dataset sizes for the four classifiers. For the four classifiers, based on the dataset sizes varying from 40 to 400, we plot boxes of EERs with no augmentation (blue box plot) and with CWGAN augmentation approach (red box plot), as shown in Fig. 4(a) for OC-SVM classifier, Fig. 4(b) for LOF classifier, Fig. 4(c) for IF classifier, and Fig. 4(d) for EE classifier, respectively. As illustrated in Fig. 4, for all the four classifiers, the EERs with CWGAN augmentation approach significantly decrease comparing with that without data augmentation over all dataset sizes. That is, CWGAN augmentation approach can effectively improve the authentication accuracy of CAGANet. Moreover, with the increase of the dataset size, the EER gradually decreases and the decrement of the EER reduces (i.e. accuracy increment tends to saturate). This is because the CNN training reaches saturation when the training data come to sufficiency.

In addition, we tabulate the mean EER without or with CWGAN augmentation over different dataset sizes for the four classifiers in Table VI, where '|' separates the mean EERs without and with CWGAN augmentation approach for different dataset sizes. As depicted in Table VI, the OC-SVM classifier generates the highest mean EERs of 19.93% without CWGAN and 10.56% with CWGAN when dataset size is 400. Even the worst condition (OC-SVM classifier), CWGAN augmentation approach reduces by 47% in mean EER, which indicates that CWGAN augmentation exhibits certain improvement in the authentication accuracy of CAGANet. On the other hand, the IF classifier performs the best with 5.05% mean EER on no augmentation and 3.64% with CWGAN, achieving a reduction of 28% in mean EER when dataset size comes to 200. Moreover, the best mean EERs of LOF and EE classifiers reach 8.69% and 6.69% on no augmentation, and 6.87% and 5.81% with CWGAN, respectively, where the dataset size is 400.

E. Effectiveness of CNN

In this section, we evaluate the effectiveness of CNN in terms of the efficiency of CNN architecture and that of CNN-learned features, respectively.

1) Efficiency of CNN Architecture: To assess the performance of the proposed CNN architecture, we compare the specific CNN with existing popular CNN structures, such as VGG [42], DenseNet [43], and GoogLeNet [44]. In order to adapt these CNN structures to our input of $(3, 200, 3)$, we make some modifications to VGG, DenseNet, and GooLeNet, respectively. a) VGG Net-D (16 layers): kernel size $= 3$, stride $= (2, 1)$, padding $= 1$ for all the MaxPool layers; FC-1000 is changed to FC-68 and the outputs of the last FC-4096 are used as features. b) DenseNet-121: output size $= (3 \times 3)$, stride $= (2, 1)$, padding $= 1$ for the first convolution layer; stride $= (2, 1)$ for the max pool layer; average pool $= (3, 3)$, stride $= (2, 1)$, padding $= 1$ for transition layers; global average pool $= 7 \times 3$ in the classification layer and is used as feature output; 1000D fully-connected is changed to 68D fully-connected. c) GooLeNet: kernel size $= (3 \times 3)$, stride $= (2, 1)$, padding $= 1$ in the first convolution layer; stride $= (2, 1)$ for all the max pool layers; kernel size $= (7 \times 3)$ in the avg pool layer and the outputs of this layer are used as features; output size $= 68$ in the linear layer.

Based on the three existing CNN structures (e.g. VGG, DenseNet and GoogLeNet) and the designed CNN, we com-

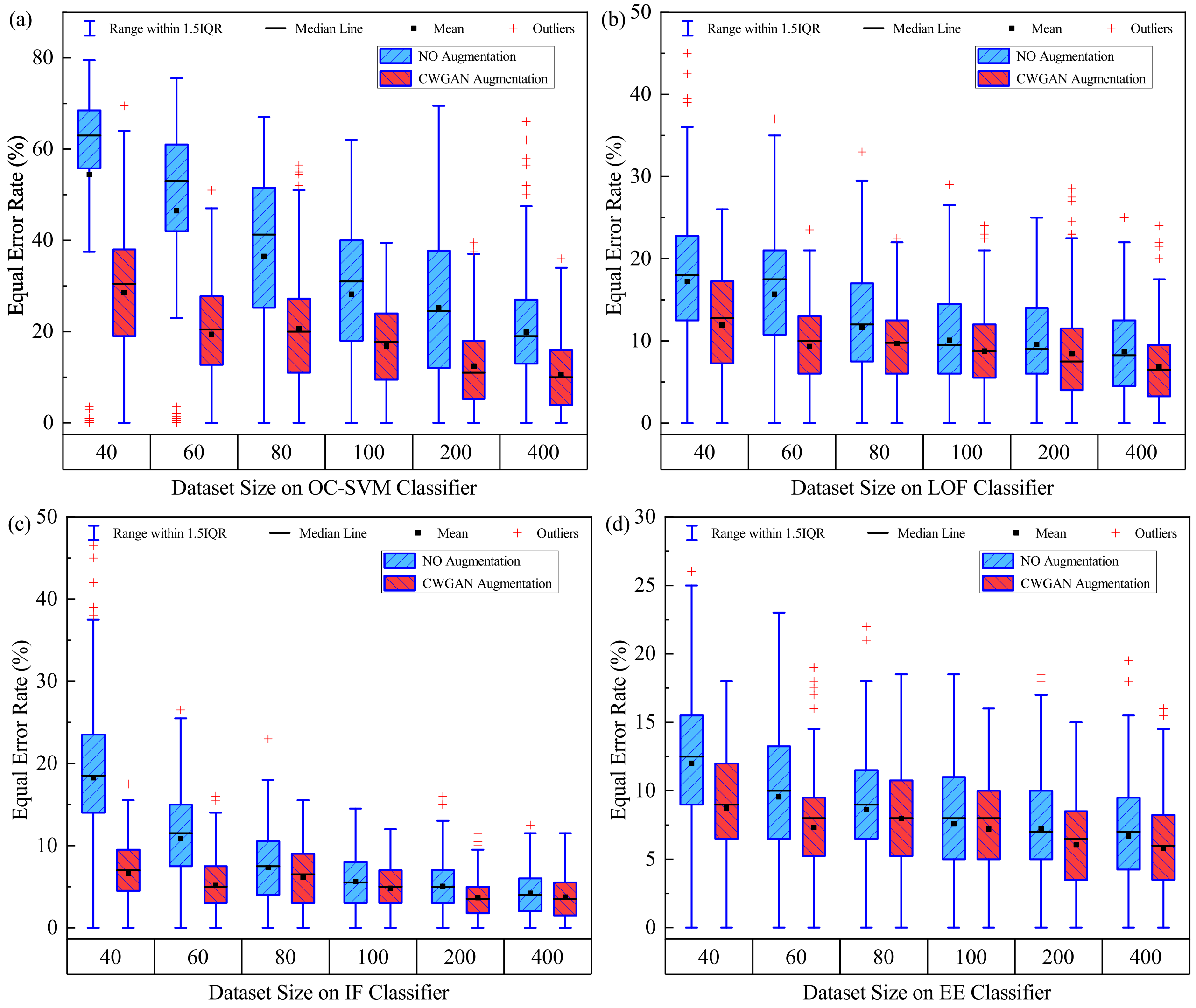

Fig. 4: EER with CWGAN Augmentation over different dataset sizes for different classifiers. (a) OC-SVM. (b) LOF. (c) IF. (d) EE.

TABLE VI: Mean EER (%) without | with CWGAN Augmentation over Different Dataset Sizes for the Four Classifiers

Classifier \ Dataset size	40	60	80	100	200	400
OC-SVM	54.47 | 28.56	46.52 | 19.43	36.50 | 20.66	28.21 | 16.87	25.17 | 12.46	19.93 | **10.56**
LOF	17.23 | 11.90	15.69 | 9.31	11.62 | 9.69	10.09 | 8.78	9.55 | 8.44	8.69 | **6.87**
IF	18.25 | 6.64	10.87 | 5.15	7.34 | 6.12	5.62 | 4.81	5.05 | **3.64**	4.20 | 3.74
EE	12.02 | 8.73	9.56 | 7.32	8.61 | 7.97	7.58 | 7.22	7.24 | 6.05	6.69 | **5.81**

pute the mean EER and parameter amount with dataset size 400 over the four classifiers in Table VII. As shown in Table VII, the designed CNN in this work generally outperforms DenseNet and GoogLeNet in both the mean EER and parameter amount. Although VGG shows the best mean EERs, it has 250 times of parameter amount over the designed CNN, which occupy much more storage space on smart devices. Therefore, in view of the accuracy and parameter amount, the designed CNN is the best network structure for CAGANet.

2) Efficiency of CNN-learned Features: We study the authentication accuracy of CAGANet to evaluate the validity of the CNN-learned features for the four classifiers of the OC-SVM, LOF, IF and EE, respectively. We compare the accuracy of CAGANet on designed features and on CNN-learned features with the same data size 400.

With the CWGAN-augmented data, the 16 designed features tabulated in Table VIII are extracted from time and frequency domains for one sensor, where the magnitude is calculated as the square root on the sum of the squares of the three axes for one sensor data. There are 48 designed features (16 features × 3 sensors) for the three sensors of the accelerometer, gyroscope and magnetometer. Then, from the 48 designed features, we utilize Fisher score to select discriminative features whose sum of Fisher scores accounted for 90% of the sum of that for all features, and use grid search to find optimal classifier parameters. We calculate the accuracy of CAGANet on selected discriminative features for the four classifiers, respectively.

TABLE VII: Mean EER (%) and Parameter on Different Network Structures for the Classifiers

Network	OC-SVM	LOF	IF	EE	Parameter	Times
VGG Net-D	3.19	4.11	2.67	5.78	75,827,332 (75M)	250
DenseNet-121	19.76	11.3	9.38	8.84	7,015,876 (7M)	23
GoogLeNet	14.05	7.165	3.72	6.26	6,333,028 (6M)	20
Designed CNN	10.56	6.87	3.74	5.81	328,556 (0.3M)	1

TABLE VIII: Designed features.

Feature	Explanation
Mean	Mean value of the magnitudes of sensor readings
Median	Median value of the magnitudes of sensor readings
SD	Standard deviation of the magnitudes of sensor readings
Maximum	Maximum value of the magnitudes of sensor readings
Minimum	Minimum value of the magnitudes of sensor readings
Range	Difference between the maximum and minimum values
Kurtosis	Width of peak of the magnitudes of sensor readings
Skewness	Orientation of peak of the magnitudes of sensor readings
Quartiles	25%, 50%, 75% quartiles of magnitudes of sensor readings
Energy	Intensity of the magnitudes of sensor readings
Entropy	Dispersion of spectral distribution of the magnitudes
P1	Amplitude of the 1st highest peak of the magnitudes
P2F	Frequency of the 2nd highest peak of the magnitudes
P2	Amplitude of the 2nd highest peak of the magnitudes

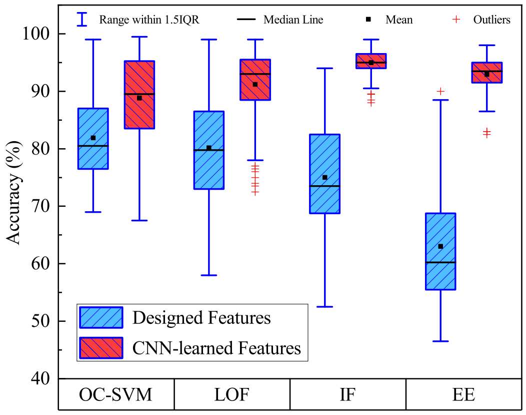

Fig. 5: Accuracy Comparison on Classifiers with Different Features

Based on the designed features and on CNN-learned features, the accuracy of CAGANet on the four classifiers is plotted in Fig. 5 and tabulated in Table IX. As illustrated in Fig. 5, the proposed CAGANet (red box plot) significantly outperforms the one based on designed features (blue box plot) over the four classifiers. In particular, the CAGANet on CNN-learned features with the IF classifier shows the best accuracy. Moreover, as depicted in Table IX, the proposed CAGANet with the IF classifier achieves 95.00% accuracy and receives 19.98% improvement comparing to that on designed features. Although the EE classifier with CNN-learnined features reaches 92.94% accuracy, it obtains the highest improvement of 29.92% compared to that with designed features.

F. Accuracy on Unseen Users

We explore the authentication accuracy of CAGANet on unseen users to evaluate the performance of pre-trained 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. We set $m = 28$ and unseen users as $20, 30, 40, 50, 60$, respectively.

Fig. 6 depicts the box plots of the EER of CAGANet on different number of unseen users for the classifiers of the OC-SVM, LOF, IF and EE, respectively. As illustrated in Fig. 6, EERs slightly fluctuate with the increase of the unseen users, but they are low and stable overall for the four classifiers, which indicate the high efficiency and strong robustness of the designed CNN. Moreover, Table X describes the mean EER with SD for CAGANet on different number of unseen users for the four classifiers. As listed in Table X, the mean EERs are all below 4.48% (40 unseen users for EE classifier) and the SDs are all less than 4.15% (60 unseen users for OC-SVM classifier). Specifically, when select 28 users to train the designed CNN and the rest 60 unseen users to train and test classifiers, LOF, IF and EE classifiers achieve the lowest mean EERs of 3.83%, 1.70% and 3.26%, respectively. However, with 40 unseen users, OC-SVM classifier reaches the best accuracy of 3.39% mean EER. Among the four classifiers, IF classifier shows the best accuracy of 1.70% mean EER.

G. Comparison with Traditional Augmentation Approaches

To illustrate the superiority of the proposed CWGAN augmentation approach, we compare the authentication accuracy of CAGANet with three traditional augmentation approaches of permutation, scaling and flipping over the four classifiers. Specifically, permutation randomly perturbs the temporal location of within-window events that relates to the act of arranging all the elements of a dataset into some sequence or order. Scaling introduces window-wise multiplicative noise (a scaling factor $\theta \in (0.9, 1.1)$) to the training data that increases robustness against noise. Flipping symmetrically flips the training data by blocks in rows.

To conduct the experiment, we use 68 users with dataset size 200 per user to train the designed CNN and exploit the trained CNN to extract features from the rest 20 users. For each classifier, we calculate the mean EER of CAGANet with different augmentation approaches: no augmentation, CWGAN augmentation, permutation augmentation, scaling augmentation, and flipping augmentation. We tabulate the mean EERs with SD of CAGANet on on augmentation and on augmentation approaches of CWGAN, permutation, scaling and flipping, respectively, in Table XI. As illustrated in Table XI, CWGAN augmentation greatly improves the accuracy of CAGANet and shows the best accuracy with the mean EERs of 12.46%, 8.44% and 3.64% over classifiers of OC-SVM, LOF and IF, respectively. Even though CAGANet with permutation augmentation over EE classifier shows a lower EER (5.92%), the proposed CAGANet performs a slightly higher EER (6.05%), but exhibits a lower SD of 3.66% comparing with the 3.96% SD. In addition, we can conclude that CWGAN augmentation approach outperforms data aug-

TABLE IX: Accuracy (%) with SD on Different Features for the Four Classifiers

Feature \ Classifier	OC-SVM	LOF	IF	EE
Designed features	81.89 (6.92)	80.19 (9.48)	75.02 (8.72)	63.02 (10.06)
CNN-Learned features	88.83 (7.47)	91.21 (5.72)	**95.00** (2.03)	92.94 (2.81)

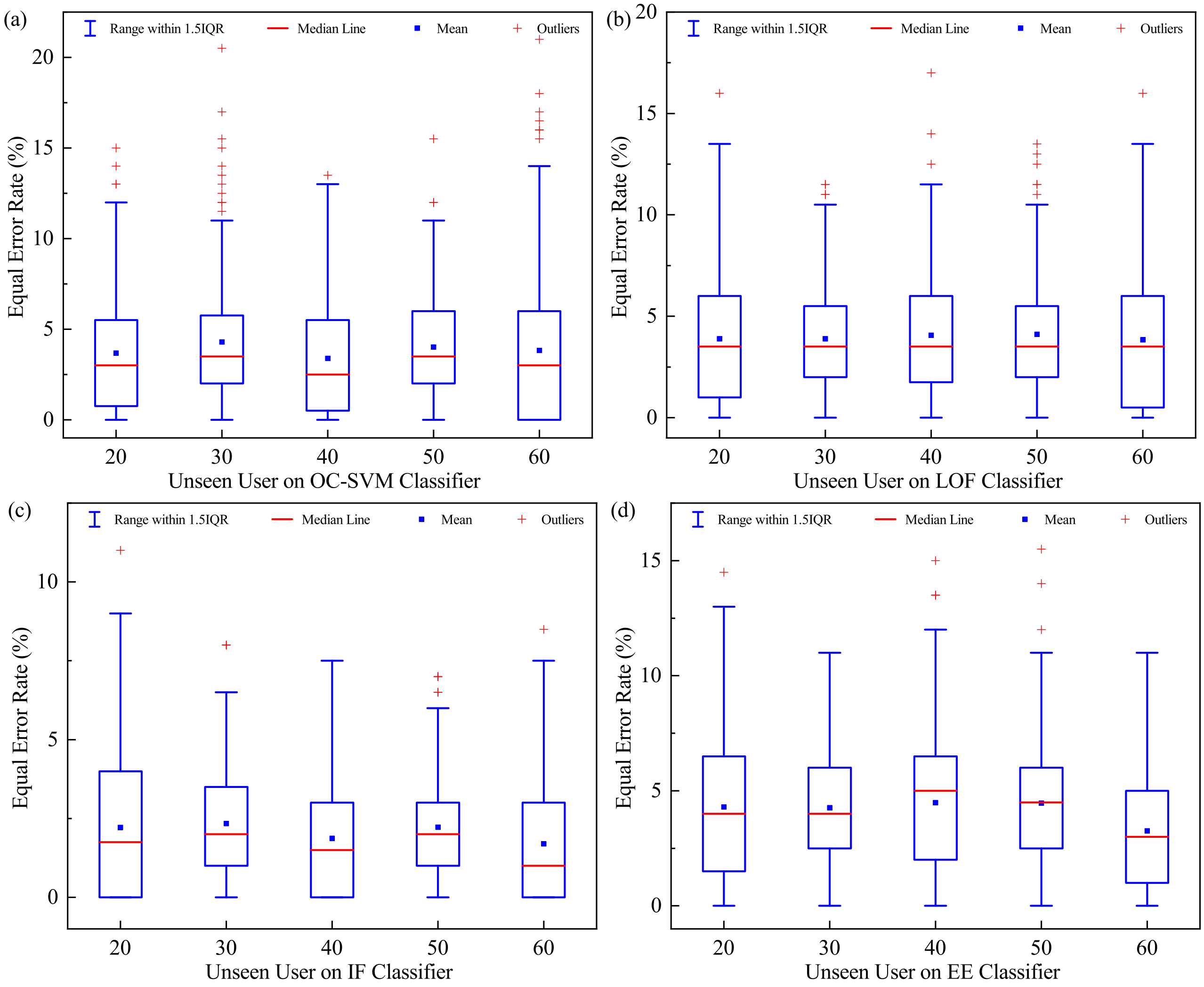

Fig. 6: EER on different number of unseen users for different classifiers. (a) OC-SVM. (b) LOF. (c) IF. (d) EE.

TABLE X: Mean EER (%) with SD on Different Number of Unseen Users for the Four Classifiers

Classifier \ Unseen Users	20	30	40	50	60
OC-SVM	3.68 (3.40)	4.29 (3.52)	**3.39** (3.17)	4.01 (2.86)	3.83 (4.15)
LOF	3.88 (3.34)	3.89 (2.64)	4.06 (3.12)	4.11 (2.74)	**3.83** (3.37)
IF	2.22 (2.15)	2.34 (1.94)	1.87 (1.80)	2.22 (1.64)	**1.70** (1.77)
EE	4.30 (3.32)	4.27 (2.65)	4.48 (3.25)	4.46 (2.70)	**3.26** (2.65)

mentation approaches of permutation, scaling, and flipping on sensor data.

H. Comparison with representative Authentication methods

To demonstrate the difference between CAGANet and the state-of-the-art approaches, we first qualitatively analyze the difference between CAGANet and data augmentation-based authentication methods, and then compare it with continuous authentication approaches on our dataset.

First, we compare CAGANet to six data augmentation-based authentication methods of MSMDGAN+CNN [29], PHY-Layer Auth [30], EchoPrint [31], SensorAuth [32], SensorCA [33] and HMOG [34], as listed in Table XII. As illustrated in Table XII, we show the data source, data augmentation approaches, and results for all the methods with data augmentation. On the one hand, for image augmentation, MSMDGAN+CNN utilizes a multi-scale and multi-direction generative adversarial network for a single palm-vein image augmentation and reaches an accuracy of 95.40% on dataset A collected at different sessions [29]. Multiuser physical layer authentication (PHY-Layer Auth) exploits three different data augmentation algorithms of the averaging data augmentation

(ADA), exponential ADA, and stochastic weight ADA for channel impulse response to achieve an authentication rate of 93.70% after nine epochs when authenticating 41 mobile nodes [30]. EchoPrint uses the projection matrix rotation imitating different camera poses to augment new face images and obtains 81.78% balanced accuracy (BAC) with vision features [31]. On the other hand, for sensor data augmentation, SensorAuth explores five data augmentation approaches of permutation, sampling, scaling, cropping, and jittering to create additional acccelerometer and gyroscope data and achieves an EER of 6.29% with dataset size 200 by combining the five approaches [32]. SensorCA applies matrix rotation on accelerometer, gyroscope and magnetometer data to reach an EER of 3.70% on the SVM-RBF classifier [33]. HMOG augments HMOG features with tap characteristics (e.g. tap duration and contact size) to obtain 7.16% EER for walking and 10.05% EER for sitting [34]. Different from aforementioned image and sensor data augmentation approaches, CAGANet utilizes CWGAN to augment sensor data of accelerometer, gyroscope and magnetometer for the designed CNN training and achieves the lowest EER of 3.64% on the IF classifier with dataset size 200.

Then, we compare CAGANet with six continuous authentication approaches of SensorAuth [32], SensorCA [33], HMOG [34], MSAuth [45], HMMAuth [46] and Multi-Motion [47] on our dataset, as tabulated in Table XIII. As demonstrated in Table XIII, we list sensors, classifiers and the corresponding results of FAR and FRR for all the related approaches. Based on our dataset, the authentication performance of the six continuous authentication approaches can be compared on an equal basis. Specifically, CAGANet utilizes the IF classifier on the sensor data of the accelerometer, gyroscope and magnetometer to reach an average FAR of 2.94% and a FRR of 6.67% (with data size 200), which show the best performance among the other continuous authentication approaches. Although Multi-Motion applies the descriptive and intensive classifier on sensor data of the accelerometer, gyroscope, magnetometer and orientation, it reaches an average FAR of 5.13% and a FRR 6.74%, with margins of 2.19% and 0.07% for the FAR and FRR, respectively, compared with CAGANet. Note that the table just provides preliminary comparative results and each approach has its own advantages and disadvantages under different conditions.

Based on Tables XII and XIII, we are among the first to use a CWGAN for data augmentation and utilize a designed CNN to extract deep features in CAGANet, and CAGANet achieves the best results of 2.94% FAR, 6.67% FRR and 3.64% EER, comparing with SensorAuth [32], SensorCA [33], HMOG [34].

VIII. Related Work

In this section, we review the state-of-the-art on the data augmentation in authentication systems and the deep learning in authentication systems, respectively.

A. Data Augmentation in Authentication Systems

In [29], the authors proposed a single-sample-per-person palm-vein identification approach, where only a single sample per class was enrolled in the gallery set for training, consisting of a multi-scale and multi-direction generative adversarial network for data augmentation and a convolutional neural network for palm-vein identification. The authors in [30] proposed a data augmented multiuser PHY-layer authentication scheme to enhance the security of mobile-edge computing system, an emergent architecture in the Internet of Things (IoT), where three data augmentation algorithms were proposed to speed up the establishment of the authentication model and improved the authentication success rate. In [31], the authors designed a data augmentation scheme for generating “synthesized” training samples for a two-factor authentication system using acoustics and vision on smartphones, which reduced false negatives significantly with limited training sample size, thus saving the user efforts in new profile registration. The authors in [32] presented a smartphone-based continuous authentication of users based on their behavioral patterns, by leveraging the accelerometer and gyroscope ubiquitously built into smartphones, where they utilized five data augmentation approaches of permutation, sampling, scaling, cropping and jittering to create additional data by applying them on training data. In [33], the authors proposed a sensor-based continuous authentication system for continuously monitoring users' behavior patterns, where they exploited a data augmentation approach of rotation to create additional data by applying it on the collected raw data improving the robustness of the proposed system. The authors in [34] proposed behavioral biometric features for continuous authentication of smartphone users by augmenting HMOG (hand movement, orientation, and grasp) features with tap characteristics, which considerably improved authentication performance.

Different from these representative data augmentation approaches in recognition systems, we utilize a CWGAN composed of a generator and a discriminator to generate additional smartphone sensor data as the data augmentation approach in the proposed continuous authentication system. In addition, CAGANet is an early work for GAN-based sensor data augmentation, since most of the augmentation approaches (e.g. permutation, sampling, scaling, cropping, jittering, and rotation) are commonly used for image augmentation.

B. Deep Learning in Authentication Systems

The authors in [35] presented a multi-device continuous authentication architecture, deployed in the MEC and cloud infrastructures, that utilized machine learning and deep learning techniques to authenticate users according to their behaviour. In [36], the authors proposed a deep-learning-based active authentication approach that exploited sensors in consumer-grade smartphones to authenticate a user, which was based on deep learning to identify user distinct behavior from the embedded sensors with and without the user's interaction with the smartphone. The authors in [37] proposed a continuous user verification system, using the widely deployed WiFi infrastructure to capture the unique physiological characteristics rooted in user's respiratory motions, by developing a deep neural network (DNN) model leveraging the extracted respiration features. In [38], the authors presented a continual

TABLE XI: Mean EER (%) with SD on Different Augmentation Approaches for the Four Classifiers

Approach \ Classifier	OC-SVM	LOF	IF	EE
No Augmentation	25.17 (16.72)	9.55 (6.08)	5.05 (3.49)	7.24 (4.17)
CWGAN	**12.46** (9.64)	**8.44** (6.18)	**3.64** (2.71)	6.05 (3.66)
Permutation	21.02 (14.82)	8.45 (6.20)	4.37 (3.17)	**5.92** (3.96)
Scaling	29.66 (17.24)	9.76 (6.21)	5.33 (4.07)	7.32 (4.37)
Flipping	28.00 (17.40)	9.27 (6.20)	3.85 (2.85)	6.11 (4.08)

TABLE XII: Qualitative Comparison to Data Augmentation-based Authentication Methods

Method	Data Source	Data Augmentation Approach	Result
MSMDGAN+CNN [29]	Palm-vein image	Multi-scale and multi-direction GAN	Accuracy: 95.40% (dataset A)
PHY-Layer Auth [30]	Channel impulse response	Averaging DA, Exponential ADA, Stochastic weight ADA	Accuracy: 93.70% (41 nodes)
EchoPrint [31]	Face image	Rotation	BAC: 81.78% (vision features)
SensorAuth [32]	Acc., Gyr.	Permutation, sampling, scaling, cropping, jittering	EER: 6.29% (dataset size 200)
SensorCA [33]	Acc., Gyr., Mag.	Rotation	EER: 3.70% (SVM-RBF)
HMOG [34]	Acc., Gyr., Mag., Touch	HMOG with tap characteristics	EER: 7.16% (walk), 10.05% (sit)
CAGANet	Acc., Gyr., Mag.	CWGAN	EER: 3.64% (IF, 200)

TABLE XIII: Comparison with Continuous Authentication Approaches on Our Dataset

Approach	Sensor	Classifier	Result	
			FAR (SD) %	FRR (SD) %
SensorAuth [32]	Acc., Gyr.	OC-SVM	7.65 (4.59)	9.01 (5.05)
SensorCA [33]	Acc., Gyr., Mag.	SVM-RBF	3.16 (1.57)	7.35 (2.52)
HMOG [34]	Acc., Gyr., Mag.	Scaled Manhattan	12.93 (6.57)	15.67 (7.24)
MSAuth [45]	Acc., Mag., Ori.	SVM	8.07 (4.54)	9.97 (4.93)
HMMAuth [46]	Acc., Gyr.	HMM	10.12 (5.97)	12.58 (6.28)
Multi-Motion [47]	Acc., Gyr., Mag., Ori.	Descriptive and intensive	5.13 (3.01)	6.74 (3.58)
CAGANet	Acc., Gyr., Mag.	IF	2.94 (3.38)	6.67 (2.89)

learning framework for behavioral-based user authentication, combining deep learning models with online learning models to achieve learning on the fly, thereby preventing a severe drop in the accuracy between sessions (over time). The authors in [39] utilized a CNN-based feature learning to extract the intrinsic fingertip-touch characteristics for modeling users' movements in legitimate authentications to defend against all presentation attacks. In [40], the authors used a Siamese convolutional neural network to learn the signatures of the motion patterns from users and achieved a competitive verification accuracy up to 97.8%. The authors in [41] presented an in-situ authentication framework that leveraged the unique motion patterns when users entering passwords as behavioural biometrics, which used a deep recurrent neural network to capture the subtle motion signatures during password input, and employed a novel loss function to learn deep feature representations that were robust to noise, unseen passwords, and malicious imposters even with limited training data.

Although these deep features have been used in these excellent recognition systems, we differ in that we specially design a CNN based on a basic unit and a basic unit for spatial down sampling to extract discriminative deep features in a continuous authentication system. We compare the designed CNN with three popular CNN structures of VGG, DenseNet, and GoogLeNet to show that the proposed CNN outperforms the three based on the accuracy and parameter amount.

IX. Conclusion

In this paper, we propose a CNN-based continuous authentication system CAGANet using a conditional Wasserstein GAN, leveraging the accelerometer, gyroscope and magnetometer on smartphones. CAGANet utilizes a conditional Wasserstein GAN for sensor data augmentation and specially designs a CNN for discriminative deep feature extraction. When a user performs operations on the smartphone, CAGANet can collect accelerometer, gyroscope and magnetometer data, preprocess and augment them, then use the designed CNN to extract features and PCA to select discriminative features, and finally utilize classifiers of OC-SVM, LOF, IF and EE to conduct user authentication. We evaluate the performance of CAGANet with extensive experiments and the experimental results show that CAGANet can authenticate users efficiently with higher accuracy comparing with the existing works.

References

[1] J. B. F. Sequeiros, F. T. Chimuco, M. G. Samaila, M. M. Freire, and P. R. M. Inácio, "Attack and System Modeling Applied to IoT, Cloud, and Mobile Ecosystems: Embedding Security by Design," *ACM Computing Surveys*,vol. 53, no. 2, Article 25, Jun. 2020, 32 pages.

[2] S. I. Imtiaz et al., "DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network," *Future Generation Computer Systems*, vol. 115, Feb. 2021, pp. 844-856.

[3] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge Attacks on Smartphone Touch Screens," in *Proc. 4th USENIX conf. Offensive technologies (Woot)*, vol. 10, Aug. 2010, pp. 1–7.

[4] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, "Design and evaluation of a shoulder-surfing resistant graphical password scheme," in *Proc. working conf. Advanced visual interfaces (AVI)*, May 2006, pp. 177–184.

[5] X. Zhang, Y. Yin, L. Xie, H. Zhang, Z. Ge and S. Lu, "TouchID: User Authentication on Mobile Devices via Inertial-Touch Gesture Analysis," in *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, Vol. 4, No. 4, Article 162, Dec. 2020.

[6] J. Im, S. Jeon, and M. Lee, "Practical Privacy-Preserving Face Authentication for Smartphones Secure Against Malicious Clients," *IEEE Transactions on Information Forensics and Security*, vol. 15, Jan. 2020, pp. 2386-2401.

[7] Q. Wang et al., "VoicePop: A Pop Noise based Anti-spoofing System for Voice Authentication on Smartphones," *IEEE Conf. Computer Communications (INFOCOM)*, Paris, France, 2019, pp. 2062-2070.

[8] E. N. Uchenna, O. O. Raphael, and A. T. Leonard, "Overview of Technologies and Fingerprint Scanner Used for Biometric Capturing," *Innovation*, vol. 1, no. 1, pp. 1-5.

[9] C. Lin, Z. Liao, P. Zhou, J. Hu, and B. Ni, "Live Face Verification with Multiple Instantialized Local Homographic Parameterization," in Proc. 27th Int. Joint Con. Artificial Intelligence (IJCAI), Stockholm, Sweden, Jul. 2018, pp. 814-820.

[10] A. Khodabakhsh, A. Mohammadi, C. Demiroglu, "Spoofing voice verification systems with statistical speech synthesis using limited adaptation data," *Computer Speech & Language*, vol. 42, pp. 20-37, Mar. 2017.

[11] B. Zou and Y. Li, "Touch-based smartphone authentication using import vector domain description," in *Proc. 29th Annu. IEEE Int. Conf. Appl. Specific Syst. Archit. Process. (ASAP)*, Milan, Italy, Jul. 2018, pp. 85–88.

[12] C. Nickel, T. Wirtl, and C. Busch, "Authentication of smartphone users based on the way they walk using k-NN algorithm," in *Proc. 8th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process. (IIH-MSP)*, Piraeus, Greece, Jul. 2012, pp. 16–20.

[13] S. Buthpitiya, Y. Zhang, A. K. Dey, and M. Griss, "n-gram geo-trace modeling," in *Proc. Int. Conf. Pervasive Comput.*, San Francisco, CA, USA, Jun. 2011, pp. 97–114.

[14] K. Niinuma, U. Park, and A. K. Jain, "Soft Biometric Traits for Continuous User Authentication, " *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4 pp. 771–780, (2010).

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 27th Int. Connf. Neur. Info. Process. Sys. (NIPS)*, Montreal, Canada, Dec. 2014, pp. 2672–2680.

[16] G. Wang, W. Kang, Q. Wu, Z. Wang, and J. Gao, "Generative Adversarial Network (GAN) Based Data Augmentation for Palmprint Recognition," in *Proc. 2018 Digital Image Computing: Techniques and Applications (DICTA)*, Dec. 2018, Canberra, Australia, pp. 1–7.

[17] J. Zhang, Z. Lu, M. Li, and H. Wu,"Gan-based image augmentation for finger-vein biometric recognition," *IEEE Access*, vol. 7, pp. 183118-183132, 2019.

[18] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," arXiv preprint arXiv:1701.07875, 2017.

[19] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in NIPS'17, 2017, pp. 5769–5779.

[20] Y. Luo, and B.-L. Lu, "EEG Data Augmentation for Emotion Recognition Using a Conditional Wasserstein GAN," in *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Jul. 2018, pp. 2535-2538.

[21] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," in *Proc. European Conf. Computer Vision (ECCV)*, Sep. 2018, pp. 116-131.

[22] X. Zhang, X. Zhou, M. Lin and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in *2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 6848-6856.

[23] C. Wu, K. He, J. Chen, Z. Zhao and Ruiying Du, "Liveness is Not Enough: Enhancing Fingerprint Authentication with Behavioral Biometrics to Defeat Puppet Attacks," in *29th USENIX Security Symposium (USENIX Security)*, Aug. 2020, pp. 2219-2236.

[24] M. M. Breunig, H.-P. Kriegel, R. T. Ng and J. Sander, "LOF: identifying density-based local outliers," in *Prof. 2000 ACM SIGMOD*, Jun. 2000, pp. 93–104.

[25] F. T. Liu, K. M. Ting, and Z.-H Zhou, "Isolation Forest," in *Prof. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413-422.

[26] P. J. Rousseeuw, K. V. Driessen, "A Fast Algorithm for the Minimum Covariance Determinant Estimator," *Technometrics*, vol. 41, no. 3, pp. 212-223, 2012.

[27] F. H. Al-Naji, R. Zagrouba, "A survey on continuous authentication methods in Internet of Things environment," *Computer Communications*, vol. 163, pp. 109-133, 2020.

[28] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. 19th Int. Conf. Neural Information Processing Systems (NIPS)*, Dec. 2006, pp. 513–520.

[29] H. Qin, M. A. El-Yacoubi, Y. Li, and C. Liu, "Multi-Scale and Multi-Direction GAN for CNN-Based Single Palm-Vein Identification," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2652-2666, 2021.

[30] R. Liao et al., "Multiuser Physical Layer Authentication in Internet of Things With Data Augmentation," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2077-2088, 2020.

[31] B. Zhou, J. Lohokare, R. Gao, and F. Ye, "EchoPrint: Two-factor Authentication using Acoustics and Vision on Smartphones," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Net. (MobiCom)*, Oct. 2018, pp. 321–336.

[32] 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.

[33] Y. Li, H. Hu, G. Zhou, and S. Deng, "Sensor-based continuous authentication using cost-effective kernel ridge regression," *IEEE Access*, vol. 6,pp. 32554–32565, 2018.

[34] Z. Sitová 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.

[35] P. M. S. Sánchez, L. F. Maimó, A. H. Celdrán, and G. M. Pérez, "AuthCODE: A privacy-preserving and multi-device continuous authentication architecture based on machine and deep learning," *Computers & Security*, vol. 103, pp. 1-14, 2021.

[36] M. Abuhamad, T. Abuhmed, D. Mohaisen, and D. Nyang, "AUToSen:Deep-Learning-Based Implicit Continuous Authentication Using Smartphone Sensors," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5008-5020, 2020.

[37] J. Liu, Y. Chen, Y. Dong, Y. Wang, T. Zhao, and Y.-D. Yao, "Continuous user verification via respiratory biometrics," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 1-10.

[38] J. Chauhan, Y.D. Kwon, P. Hui, and C. Mascolo, "ContAuth: Continual Learning Framework for Behavioral-based User Authentication," in *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol*, vol. 4, no. 4, Article 122, Dec. 2020, 23 pages.

[39] C. Wu, K. He, J. Chen, Z. Zhao, and R. Du, "Liveness is Not Enough: Enhancing Fingerprint Authentication with Behavioral Biometrics to Defeat Puppet Attacks," in *Proc. 29th USENIX Security Symposium (USENIX Security)*, Aug. 2020, pp. 2219-2236.

[40] M. P. Centeno, Y. Guan, and A. V. Moorsel, "Mobile based continuous authentication using deep features," in *Proc. 2nd Int. Workshop Embedded and Mobile Deep Learning (EMDL)*, Jun. 2018, pp. 19–24.

[41] C. X. Lu, et al., "DeepAuth: In-situ authentication for smartwatches via deeply learned behavioural biometrics," in *Proc. Int. Symposium Wearable Computers (ISWC)*, pp. 204–207.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015, pp. 1–14.

[43] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1–12.

[44] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[45] W. Lee, R. B. Lee, "Multi-sensor authentication to improve smartphone security," in *2015 Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2015, pp. 1-11.

[46] A. Roy, T. Halevi, N. Memon, "An HMM-based multi-sensor approach for continuous mobile authentication," in *2015 IEEE Military Communications Conference (MILCOM)*, 2015, pp. 1311-1316.

[47] C. Shen, Y. Li, Y. Chen, X. Guan, R. A. Maxion, "Performance Analysis of Multi-Motion Sensor Behavior for Active Smartphone Authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 48-62, 2018.