Memory-Augmented Autoencoder based Continuous Authentication on Smartphones with Conditional Transformer GANs

Yantao Li, Li Liu, Shaojiang Deng, Huafeng Qin, Mounim A. El-Yacoubi, and Gang Zhou, *Fellow, IEEE*

Abstract—Over the last years, sensor-based continuous authentication on mobile devices has achieved great success on personal information protection. These proposed mechanisms, however, require both legal and illegal users' data for authentication model training, which takes time and is impractical. In this paper, we present MAuGANs, a lightweight and practical Memory-Augmented Autoencoder-based continuous Authentication system on smartphones with conditional transformer Generative Adversarial Networks (GANs), where the conditional transformer GANs (CTGANs) are used for data augmentation and the memory-augmented autoencoder (MAu) is utilized to identify users. Specifically, MAuGANs exploits the smartphone built-in accelerometer and gyroscope sensors to implicitly collect users' behavioral patterns. With the normalized legitimate user's sensor data, MAuGANs uses a CTGAN composed of a conditional transformer-based generator and a conditional transformer-based discriminator to create additional training data for the MAu. Then, the MAu is trained on the augmented legitimate user's data. The trained MAu reconstructs the current user data and then calculates the reconstruction error between the reconstructed data and current user data. To carry out user authentication, MAuGANs compares the reconstruction error with a predefined authentication threshold. We evaluate the performance of MAuGANs on our dataset, where our extensive experiments demonstrate that MAuGANs reaches the best authentication performance, when comparing with the representative state-of-the-art methods, by 0.33% EER and 99.65% accuracy on 10 unseen users.

Index Terms—Continuous authentication, memory-augmented autoencoder, conditional transformer GANs, EER

1 Introduction

With the rapid advancement of mobile communication technologies, mobile devices have become indispensable and they have been playing a significant role in our daily lives. Nowadays, people are used to browsing social networks, doing online shopping and bank transactions, and storing personal sensitive information on mobile devices. For instances, with the Facebook app on mobile devices, people can view and comment on news, pictures or emotional expressions posted by others. With the Walmart app, people can buy groceries and then pay the order online with their registered credit/debit cards. People also can receive private alerts, save screenshots and take photos by using mobile devices. People's reliance on mobile devices is incredibly increasing and thus their importance is quickly emerging. Thus, it is essential to authenticate the users of the mobile devices to ensure legal usage. However, traditional authentication mechanisms, such as PINs [1] and graphical patterns, require users' active participation, suffering, as a result, from smudge attacks [2], shoulder surfing attacks [3], and password inference attacks [4]. In recent years, biometrics-based authentication has become a promising security mechanism on mobile devices, due to the uniqueness of human biometric characteristics. Biometric characteristics can be broadly categorized into: physiological biometrics and behavioral biometrics. Physiological biometrics refer to physical traits of the human body, including face, hand geometry, fingerprint, finger vein, and iris. These traits, however, not only need costly-specialized sensors, but also suffer from the hazard of spoofing attacks [5]. For example, it is indeed possible to generate a synthetic or real partial fingerprint that serendipitously matches one or more the stored templates for a significant number of users in practical fingerprint-based authentication systems [6]. Behavioral biometrics relate to human behavioral patterns, involving human motion, gesture, gait, voice, and breath. For instance, the authors of [7] authenticate users silently and transparently by exploiting the user touch behavior biometrics and leveraging the integrated sensors to capture the micro-movement of the device caused by user's screen-touch actions. In [8], the authors utilize the touching sensor to record the on-screen gesture and the inertial sensor to capture the device motion caused by the touching gesture, and then combine the unique features from the on-screen gesture and the device's motion for user authentication. The authors in [9] leverage photoplethysmography (PPG) sensors available in most wrist-worn wearable devices to simultaneously perform a password/pattern/signature au-

- *Yantao Li, Li Liu, and Shaojiang Deng are with the College of Computer Science, Chongqing University, Chongqing 400044, China. E-mail: {yantaoli,20162969,sj_deng}@cqu.edu.cn*
- *Huafeng Qin is with the School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing 400067, China. E-mail: qinhuafengfeng@163.com*
- *Mounim A. El-Yacoubi is with SAMOVAR, Telecom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France. E-mail: mounim.el_yacoubi@telecom-sudparis.eu*
- *Gang Zhou is with the Department of Computer Science, William & Mary, Williamsburg, VA, 23185, USA. E-mail: gzhou@cs.wm.edu*

Manuscript received February XX, 2023; revised Month date, 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 62072061, 61976030 and U20A20176 (Corresponding author: Huafeng Qin.)

thentication and a physiological-based authentication in a two-factor system. While these authentication systems work well for initial login on mobile devices, the security issue may raise with unattended devices after users have logged in [10].

To mitigate the issue of post initial identification, behavioral biometrics-based continuous authentication mechanisms have been proposed to implicitly and constantly verify the users throughout the usage of mobile devices. For instance, the authors of [13] propose a two-stream CNN-based continuous authentication system utilizing the one-class SVM as the classifier. In [15], the authors present a smartphone-based continuous authentication framework using a deep learning based support vector data description (DeSVDD) algorithm as the classifier. Nevertheless, most of the continuous authentication systems either exploit convolutional neural networks (CNNs) to extract discriminative features or utilize deep learning networks as classifiers. Thus, they are facing the challenges associated with insufficient sensor data and complex classifiers, which degrade the authentication accuracy and classification efficiency, respectively. Specifically, these authentication systems not only need a large amount of training data for deep feature extractors and classifiers, but they require both legitimate user's data and imposter's data for model training. There are some works addressing the challenge of insufficient training data by deep learning based data augmentation techniques [16], [17]. They have, however, to train a deep model for each source of each user's data, e.g. 44×3 independent transformer-based GANs for 44 volunteers' accelerometer, gyroscope and magnetometer sensor data in [17], which costs much training resource of time and computation. In addition, the classifier training usually requires both legitimate user's data and imposter's data in most of the existing authentication systems, which also takes extra costs.

Different from the existing works, we propose MAuGANs, a Memory-Augmented Autoencoder based continuous Authentication system on smartphones with conditional transformer GANs. In MAuGANs, the user performs, on the commercial mobile devices, widely-adopted operations, such as web browsing or text producing, that are implicitly processed without user's awareness for user continuous authentication. Specifically, MAuGANs consists of three modules, data collection and preprocessing, data reconstruction, and authentication. MAuGANs first leverages the smartphone built-in accelerometer and gyroscope sensors to implicitly collect user's behavioral patterns, which are then normalized for the designed generative adversarial network (GAN) training or used for system testing. Based on the normalized legitimate user's sensor data, MAuGANs uses, for smartphone sensor data augmentation, the designed conditional transformer GAN (CTGAN), composed of a conditional transformer-based generator and a conditional transformer-based discriminator, that can be trained on multiple users' data simultaneously. With the augmented legitimate user's data, MAuGANs trains the designed memory-augmented autoencoder (MAu) to record the most representative prototypical patterns for data reconstruction in the enrollment phase. With the current user's data, MAuGANs uses the well-trained MAu to reconstruct the data, and calculates the reconstruction error between the reconstructed data and the current user's data for authentication in the continuous authentication phase. We evaluate the overall performance of MAuGANs on our dataset, and the experimental results demonstrate that MAuGANs outperforms the representative state-of-the-art methods by achieving the lowest equal error rate and the highest authentication accuracy.

MAuGANs tackles the following challenges to provide user continuous authentication:

The first challenge is how to generate sufficient training data with high efficiency? To address this challenge, we design a conditional transformer GAN for user sensor data augmentation, that creates high-quality accelerometer and gyroscope data of users conditioned on their labels. Different from traditional GANs, the conditioned one can reduce training cost and enhance generalization.

The second challenge is how to design a lightweight classifier with only legitimate user's data? To tackle this challenge, we design a memory-augmented autoencoder trained only on the legitimate user's data. The trained MAu reconstructs the user's data based on the current user's input and then calculates the reconstruction error for classification. Unlike widely-used one-class classifiers, such as OC-SVM or isolation forest, we simply compare the reconstruction error with predefined authentication threshold to identify the user.

The key contributions of our work are summarized as follows:

- We present MAuGANs, a lightweight and practical continuous authentication system based on the memory-augmented autoencoder, leveraging the smartphone built-in accelerometer and gyroscope sensors. MAuGANs consists of three modules, namely data collection and preprocessing, data reconstruction, and authentication.
- We propose, for smartphone sensor data augmentation, conditional transformer GANs, composed of a conditional transformer-based generator and a conditional transformer-based discriminator, that can be trained on multiple users' data simultaneously, thereby significantly reducing training cost and moderately enhancing generalization.
- We design a memory-augmented autoencoder to perform the identification by harnessing the reconstruction error on current user input. The MAu trained on legitimate data is lightweight and can be directly implemented on smartphone for authentication which is instantaneous.
- We validate MAuGANs using our dataset and compare it with representative solutions. Extensive experiments demonstrate that MAuGANs outperforms the representative solutions by attaining the lowest EER and the highest accuracy.

The remainder of this work is organized as follows: Sec. 2 reviews the existing representative authentication systems. We introduce the system models consisting of the communication model, the adversary model, and an overview of MAuGANs in Sec. 3. In Sec. 4, we depict the data collection and preprocessing phases for MAuGANs. Sec. 5 details the conditional transformer GAN for training data augmentation. Sec. 6 elaborates the data reconstruction

and authentication processes. We evaluate the performance of MAuGANs in Sec. 7. Sec. 8 concludes our work.

2 Related Work

In this section, we review the existing representative approaches dedicated to sensor-based authentication systems, and data augmentation based authentication systems, respectively.

2.1 Sensor-based Authentication Systems

Smartphones are equipped with various sensors, e.g. the accelerometer, gyroscope, magnetometer, pressure, microphone, speaker, and camera, which can be exploited to capture human motion, gesture, gait, and voice for user identification. Extensive research works have exploited these built-in sensors to capture behavioral or physiological biometrics for user recognition. In [18], the authors leverage the accelerometer, magnetometer, and gyroscope on smartphones to capture users' motion patterns for user authentication, where two types of feature fusion solutions, i.e. serial feature fusion and parallel feature fusion, are used to combine the designed features for effective feature representation. The authors in [19] exploit the accelerometer and gyroscope ubiquitously built into smartphones to capture users' behavioral features for continuous authentication with a deep feature fusion technology. The authors in [20] explore users' PIN input behaviors or patterns to classify their hand posture shape characteristics by describing the fine-grained multi-path effect when users' fingers are still on screens as a second-factor authentication. In [21], the authors conduct specially-designed multi-touch gestures with a single swipe on touchscreens of smart devices to record behavioral traits and hand geometry for user authentication. The authors in [22] utilize the smartphone accelerometer to monitor person's gait patterns continually in the background in order to identify a walking individual by analyzing the recorded gait data. In [23], the authors leverage the structure-borne propagation of sounds to reckon the pressure on the device screen from the user's finger for secure PIN authentication. The authors in [24] extract target specific features with the best characterized gait patterns, and then apply a normalization algorithm to remove gait-irrelevant perturbation in signals for a WiFi-based person identification. In [25], the authors utilize the microphone on mobile devices to capture pop noises caused by users' oral airflow when saying passphrases for voice authentication. In [26], the authors exploit the acoustic signals from the enrolled device to compare with the calculated dynamic acoustic fingerprints by the login device for securing mobile two-factor authentication. The authors in [27] employ the reflected acoustic signals from vocal tracts during user speaking to perform user authentication on mobile devices. In [5], the authors leverage audio modules on smartphones from signal variations associated with hand dynamics to track the whole unlocking process of devices and then extract robust and discriminative features for user identification. The authors in [28] utilize a signal processing strategy and a pattern matching technique to capture volitional and non-volitional gaze patterns by leveraging the built-in front camera to track human eye movement for user authentication.

Different from the aforementioned authentication systems, we explore the smartphone accelerometer and gyroscope sensors to propose a memory-augmented autoencoder based continuous authentication system, which is lightweight and practical in real-time authentication.

2.2 Data Augmentation based Authentication Systems

Data augmentation is one of the commonly-used techniques in machine learning to prevent model overfitting and enhance generalization. A significant body of research utilizes data augmentation strategies to create additional data for authentication systems that have limited data. Data augmentation in the authentication can be categorized into approaches based on transformation, search, and deep learning. For transformation-based data augmentation, the authors in [30] explore five transformation-based methods, namely permutation, scaling, jittering, sampling, and cropping, to create extra data in a continuous identification system on smartphones. In [31], the authors exploit five approaches of flipping, downsampling, cropping and label expansion in both time domain and frequency domain for time-series anomaly detection. The authors in [32] exploit local averaging as a down-sampling technique and shuffling on wearable sensor data for human activity classification. In [33], the authors transpose landmark pixel coordinates of a camera to other camera poses to generate training samples for the authentication of users' acoustics and vision. For search-based data augmentation, the authors of [34], [35] utilize auto augmentation search in the search space of permutation, jittering, cropping, scaling, rotation, time-warping, and magnitude-warping on the collected data to find an optimum plan for augmenting the training data in a CNN-based continuous authentication system on smartphones. In [36], the authors exploit a modality-agnostic automated data augmentation in the latent space to fine-tune four universal data transformation operations of hard example interpolation, hard example extrapolation, Gaussian noise and difference transform to augment data for any modality in a generic way. The authors in [37] use expert and gate networks to search the optimal weights for some meta transformation-based operations to perform data augmentation for activity classification. For deep learning-based data augmentation, in [38], the authors employ a conditional Wasserstein generative adversarial network (CWGAN) for data enhancement in a continuous authentication system on smartphones. To generate extra training data, the authors of [17] utilize a transformer-based GAN, composed of a transformer-based generator and a CNN-based discriminator in a continuous authentication system on smartphones.

Although these data augmentation approaches have been used in representative authentication systems, we differ in that we propose the conditional transformer GANs for smartphone sensor data augmentation in an authentication system, which greatly reduces training cost in terms of time and computation resource.

3 System Model

In this section, we consider a continuous authentication system, that explores behavioral features extracted from motion sensors of the accelerometer and gyroscope on mobile

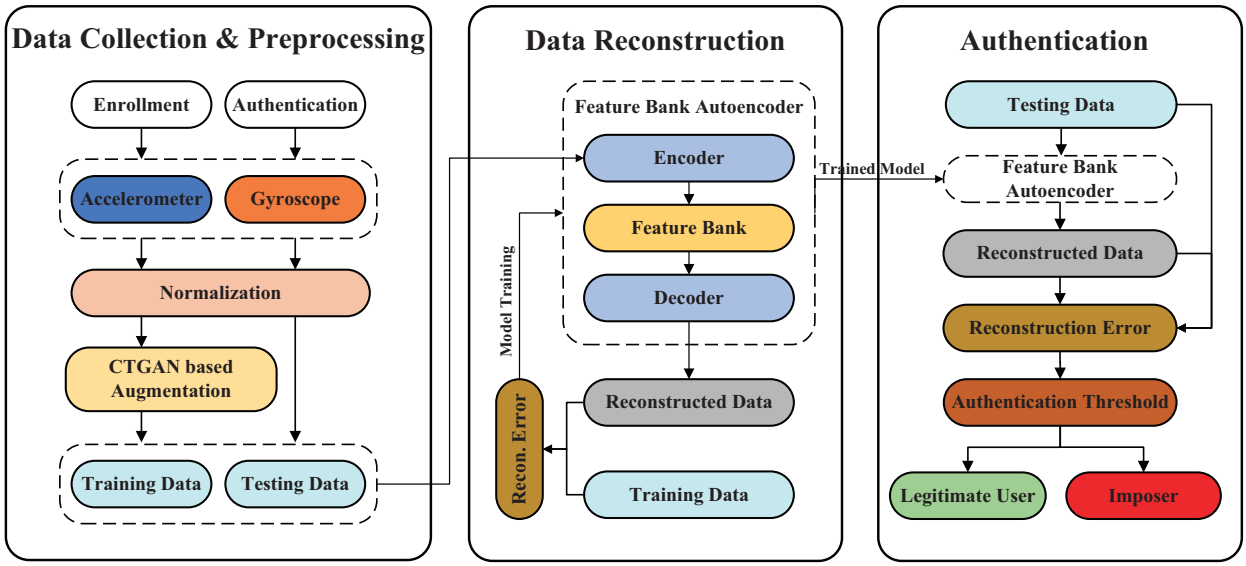


Fig. 1: Architecture of MAuGANs.

devices (e.g., smartphones, tablets, and smartwatches). The users only need to operate on the mobile devices without knowledge of direct involvement after unlocking them. We describe next the communication model and the adversary model, and then provide an overview of MAuGANs.

3.1 Communication Model

We design a continuous authentication system that involves two players: a set of users and a remote server. The mobile device acquires the user behavioral traits and transmits the legitimate user's data to the server. The remote server utilizes the legitimate user's data to train the authentication model, and then returns the trained model to the mobile device for conducting authentication. The authentication process typically consists of two phases: the enrollment phase and the authentication phase. In the enrollment phase, the mobile device captures the legitimate user's behavioral patterns and then submits the data to the remote server. The remote server exploits the conditional transformer GAN to augment the submitted legitimate user's data, then trains the memory-augmented autoencoder by using the augmented data, and finally returns the trained autoencoder to the mobile device. In the authentication phase, the two sensors begin with collecting real-time behavioral/motion data while users are operating their mobile devices. Each device is monitored by its ad hoc trained autoencoder, that carries out user authentication based on the autoencoder reconstruction error threshold.

3.2 Adversary Model

Continuous authentication is an implicit identification process for users based on extracting behavioral features by mobile devices' built-in motion sensors. However, sensor-based continuous authentication is threatened by the mimic attack. In this type of attack, an adversary first watches how a legitimate user operates his/her device to pass the authentication stage, and then practices to impersonate that user's behavior (operation way) to fool the accelerometer and gyroscope sensors.

3.3 MAuGANs Overview

We present an overview of MAuGANs, a lightweight and practical continuous authentication system on smartphones, that exploits conditional transformer generative adversarial networks (CTGANs) for data augmentation and utilizes a memory-augmented autoencoder (MAu) to identify users, as illustrated in Fig. 1. MAuGANs is composed of two ends: a smartphone and a remote server. The smartphone is dedicated to the entire continuous authentication process with the well-trained MAu. The remote server is responsible for the training of the CTGANs and MAu, which greatly reduces the computation amount on smartphones. In our work, we focus on the smartphone end for continuous authentication, which exploits the built-in sensors of the accelerometer and gyroscope, to capture users' behavioral patterns. As demonstrated in Fig. 1, MAuGANs is composed of three modules: data collection and preprocessing, data reconstruction, and authentication. The MAuGANs authentication process consists of the enrollment phase and the authentication phase.

In the enrollment phase, MAuGANs acquires, on the smartphone, the profile of a legitimate user by training the MAu on a remote server. Specifically, the smartphone captures the legitimate user's behavioral data from the built-in accelerometer and gyroscope, which are first preprocessed and then transmitted to the remote server via a secure socket protocol. With the normalized sensor data, the remote server trains the CTGAN to generate more sensor data. The latter are then combined with the collected data in a certain proportion for the MAu training. In the training process, the MAu parameters are updated recursively for effectively representing the prototypical elements of the legitimate user. The server sends the well-trained MAu of the legitimate user to his/her smartphone for user identification.

In the continuous authentication phase, based on the trained MAu with the legitimate user's data, MAuGANs identifies the current user based on the reconstruction error on his/her sensor data input. Concretely, the accelerometer

and gyroscope sensor data are collected once a user starts using the smartphone. They are then normalized, and fed to the trained MAu of the legitimate user. Next, the trained MAu reconstructs the normalized sensor data from the current user input and then calculates the reconstruction error between the reconstructed data and the current user data input. If the reconstruction error is less than a predefined authentication threshold, MAuGANs identifies the current user as a legitimate one; otherwise, the proposed system recognizes the user as an imposter, and thus the mobile device will be locked for initial login.

MAuGANs is highly adapted to the resource-constrained mobile devices, as it is lightweight thanks to its few computation demands associated with data reconstruction and comparison, and its practicability emanating from the simple implementation of the trained MAu.

4 Data Collection and data Preprocessing

In this section, we describe the data collection process and the preprocessing of the collected data.

For *data collection*, we choose smartphone built-in sensors that can effectively represent the user's behavioral characteristics. The accelerometer motion sensor captures the user coarse-grained patterns, such as hand movements. The gyroscope sensor acquires fine-grained motion traits, such as finger touch gestures. The magnetometer measures the ambient magnetic field, such as device orientation [39]. To reduce the computation and accelerate the authentication, we select the accelerometer and gyroscope sensors for MAuGANs. Once a user begins operating the smartphone, MAuGANs starts collecting the sensor data of the accelerometer and gyroscope over an authentication period t with sampling rate f_s. For each period t, MAuGANs gains n ($n = t \times f_s$) samples of sensor data, each denoted by $(x_a, y_a, z_a, x_g, y_g, z_g)^T \in \mathbb{R}^6$, where x, y, z indicate a sensor's three axes, and a, g represent the sensors of the accelerometer and gyroscope, respectively.

For *data preprocessing*, we first represent the collected sensor data of a time period t by a $r \times n$ matrix:

$$D = [D^1, D^2, \cdots, D^r]^T = \begin{bmatrix} x_a^1 & y_a^1 & z_a^1 & x_g^1 & y_g^1 & z_g^1 \\ x_a^2 & y_a^2 & z_a^2 & x_g^2 & y_g^2 & z_g^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_a^n & y_a^n & z_a^n & x_g^n & y_g^n & z_g^n \end{bmatrix}^T,$$

where $r = 6$ and $n = t \times f_s$. Then, we normalize each axis of each sensor into (0,1] by $d^i_{j_{norm}} = \frac{d^i_j}{-} min(D^i) max(D^i) - min(D^i)$, where $i = 1, 2, \cdots, r$ for each axis of the two sensors and $j = 1, 2, \cdots, n$ for the number of samples, respectively. Finally, the normalized sensor data D_{norm} are used for the training of the conditional transformer GANs in the enrollment phase.

5 Conditional Transformer GAN for Data Augmentation

Generative adversarial networks (GANs) present a solution to learn deep representations without extensively-annotated training data [44], [45]. They have been widely applied in classification and regression tasks (e.g. DCGAN [46], CoGANs [47] and SimGAN [48]), image synthesis (e.g. LAPGAN [49], GAWWN [50], and DeLiGAN [51]), image-to-image translation (e.g. CycleGAN [52] and MGANs [53]), and super-resolution (e.g. SRGAN [54]). GANs are typically composed of two adversarial networks, i.e. the generator and the discriminator. The generator aims to learn the distribution of the genuine data, while the discriminator aims to correctly distinguish whether the input data come from the genuine data or the generated data. Transformer is a model architecture entirely depending on an attention mechanism to capture global dependencies between its inputs and outputs [55]. Thus, transformer-based GAN is typically composed of a transformer-based generator that progressively increases feature resolution and a multi-scale discriminator that simultaneously draws semantic contexts and low-level textures [56]. As an unsupervised learning method, the transformer-based GAN, however, suffers from issues associated with random output and excessive training. Specifically, there is no control on the modes of the data being generated, and thus the models need to be trained on each class separately. In this section, we propose a transformer GAN conditioned on class labels, conditional transformer GAN (CTGAN), that tackles the issues mentioned above. We to propose a conditional transformer GAN for smartphone sensor data augmentation. Superior to unconditional transformer GAN, the CTGAN can train multiple users' data simultaneously, and only configures the hyper-parameters once, thereby significantly reducing training cost and moderately enhancing generalization.

TABLE 1: Generator Structure

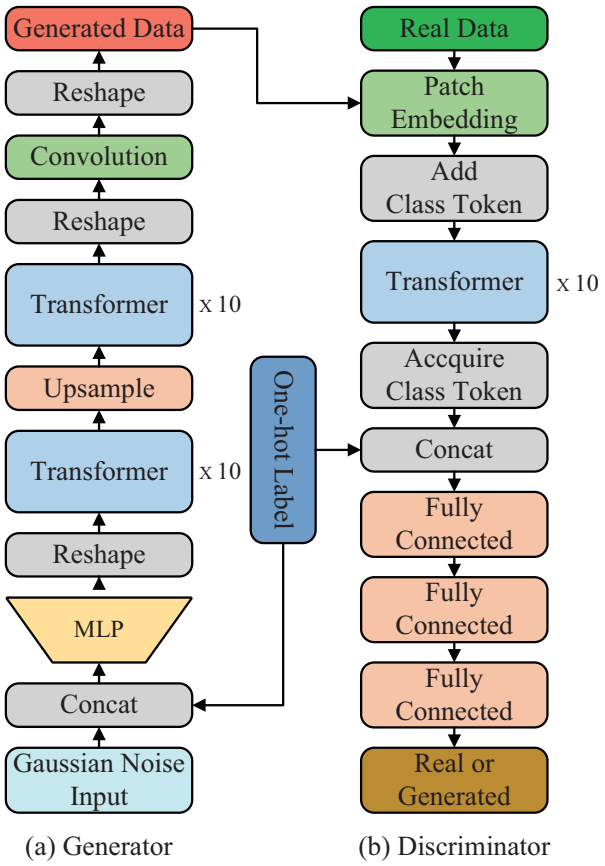
Operators	Output	Repeat	Multi-head
Random Noise + one-hot label	$100 + N$	1	-
MLP	6400	1	-
Reshape	50×128	1	-
Transformer	50×128	10	8
Upsample	200×32	1	-
Transformer	200×32	10	8
Reshape	$20 \times 10 \times 32$	1	-
Conv	$20 \times 10 \times 3$	1	-
Reshape	200×3	1	-

5.1 CTGAN

The CTGAN consists of a conditional transformer-based generator and a conditional transformer-based discriminator. The transformer encoder is composed of a multi-head self-attention module and a feedforward multiple-layer perceptron with GELU nonlinearity.

5.1.1 Generator

The generator produces genuine-like samples by learning the distribution of the real data. As shown in Table 1 and Fig. 2(a), the conditional transformer-based generator mainly consists of 2 stages of transformers with the input of the random noise concatenated with one-hot labels, where each stage stacks 10 transformer encoders with 8 heads for the multi-head attention mechanism. Before each transformer encoder, the positional encoding is inserted, due to the time-sequential nature of the sensor data, and thus the generated samples conform to the time series aspect of real samples. An upsample block composed of a reshaping module and a pixel-shuffle module is placed between the transformer



(a) Generator (b) Discriminator

Fig. 2: Architecture of CTGAN.

stages for reducing the parameter amount and fitting long time-window data. To obtain the output with the sensor data format, we first reshape the 1D sequence generated by the last transformer to a 2D feature map, then use the Conv to compress the embedding dimension, and finally reshape it back to a 1D sequence associated with sensor data format.

5.1.2 *Discriminator*

The discriminator distinguishes whether a sample is from the real data or genuine-like data. As shown in Table 2 and Fig. 2(b), the conditional transformer-based discriminator mainly consists of Conv (patch embedding), transformer, and dense layers, where the Conv layer has no activation function, the dense layer adopts a leaky version of the rectified linear unit (LeakyReLU) as the activation function, and WANG-GP is adopted as the loss function. The input sensor data are first split into patches, then projected into a high-dimension space by patch embedding, and finally a class token is appended at the beginning and the positional encoding is added as the input of the transformer layer. The transformer layer consists of stacked 10 transformer encoders with 8 heads for the multi-head attention mechanism. The class token is acquired from the output of the transformer layer, and then is combined with a one-hot label as the input of three dense layers for classification.

5.2 Data Augmentation

In the traditional field of natural language processing (NLP), the transformer typically takes a sentence as the input,

TABLE 2: Discriminator Structure

Operators	Output	Repeat	Multi-head
Sensor Data	200×3	1	-
Patch Embedding	20×128	1	-
Add cls token	21×128	1	-
Transformer	21×128	10	8
Acquire cls token	128	1	-
cls token+one-hot label	$128 + N$	1	-
Dense (LeakyReLU)	512	1	-
Dense (LeakyReLU)	512	1	-
Dense	1	1	-

where each word is represented by a fixed-length vector through embedding. In this work, we select a time period of $t = 2$ seconds with a sampling rate of $f_s = 100Hz$, and thus $n = t \times f_s = 200$ samples can be collected in the period. Then, the 2-second D_{norm} sensor data can be represented by a 1D sequence of $(200, 3)$, where 200 denotes the sequence length and 3 indicates the embedding representation of each position. Thus, the generator takes the input of the random Gaussian noise with size of 100 concatenated with N one-hot labels ($N = 70$ corresponds to 70 users) and then feeds them to a multiple-layer perceptron (MLP), which are then reshaped into a vector with length of (50×128) for the transformer encoder, where 50 indicates the time window's length and 128 represents 128-dimensional embedding for the sensor data. Between the two transformer encoders, the 1D sequence vector (50×128) is first reshaped to a 2D feature map with shape of $(10 \times 5 \times 128)$ by the upsample layer, where an upsampled feature map with shape of $(20 \times 10 \times 32)$ is obtained by using the pixelshuffle operation, and the first two elements are then combined in the embedding dimension resulting in a 1D sequence with shape of 200×32, which are then fed to the other transformer encoder. To obtain the sensor data format (three embedding dimensions corresponding to the three axes of a sensor), 1D sequence data with 32 dimensions generated by the second transformer is first reshaped to a 2D feature map with shape of $20 \times 10 \times 32$, and then the channel is compressed to 3 by the Conv layer, and finally the data is reshaped back to a 1D sequence with shape of 200×3.

Taking the real data or generated data with shape of (200×3) as the input, the discriminator utilizes a patch embedding operation to map the senor data into a high-dimensional space representation of (20×128), where the patch embedding is deployed by a convolution with kernel size of (10×10), stride of (10×10) and kernel number of 128. After that, a 128-dimensional class token is appended at the beginning of the representation to obtain a 1D sequence with shape of (21×128), and then a positional encoding with shape of (21×128) is added, which is then fed to the transformer encoders. The 128-dimensional class token in the output is acquired and then is concatenated with a N-dimensional one-hot label to obtain a 1D sequence with shape of $((128 + N) \times 1)$, which is taken by three fully-connected layers to output the real or fake prediction.

6 Data Reconstruction and Authentication

Data compression and reconstruction are significant unsupervised approaches in anomaly detection tasks. As these approaches learn a normal profile based only on the normal

data examples, they do not reconstruct well the anomaly samples when compressing and reconstructing them. The expectation then is that the reconstructed error is larger in this case, which allows identifying the anomaly samples. Real-life anomaly detection tasks, however, are challenging due to the lack of human supervision. With the development of deep learning, it is possible to use deep neural networks to fit arbitrary data distributions. Deep autoencoder (AE) is one of the deep learning models that is effective for compressing and reconstructing input data. The AE is typically composed of an encoder and a decoder, where the encoder generates a compressed encoding from the input and the decoder reconstructs the data from the encoding [57]. Specifically, the encoder compresses the high-dimensional input data (e.g. pictures, and time-series data) into a low-dimensional feature representation (encoding) by propagating the input through the encoder layers (i.e. Fully connected (FC), and Conv). The decoder restores and reconstructs the low-dimensional feature representation through the decoder layers (i.e. FC, and DeConv) to obtain an approximation of the original input. However, the authors in [58], [59] indicate that sometimes the AE can generalize so well that it can also reconstruct abnormal inputs well. To tackle this undesirable generalization issue, a memory-augmented autoencoder is proposed in our work to reduce too strong generalization [59].

6.1 Memory-Augmented Autoencoder

Inspired by [59], [60], we design a memory-augmented autoencoder, named MAu, to perform the classification. The MAu mainly consists of an encoder, a feature memory module, and a decoder, as illustrated in Fig. 3. Different from an autoencoder (AE), the MAu includes a memory module, which saves the prototypical features of trained normal samples. Thus, the decoder in the MAu conducts data reconstruction merely from the prototypical features saved in the memory, thereby addressing the generalization issue of abnormal samples. As shown in Fig. 3, the encoder module in the MAu encodes the input into a latent feature representation. Then, the memory module utilizes the encoded representation as a query to fetch the most similar features in the memory through memory addressing. Finally, by adding the stored feature representation as memory-based representation, the decoder performs the data reconstruction.

Encoder and decoder: The encoder intends to compress the input sensor data into a latent feature representation, that is used as a query to retrieve the relevant items in the feature memory. The encoder of the MAu consists of four consecutive Conv layers with the batch normalization (BN) and LeakyReLU on each layer. The decoder aims to reconstruct the input sensor data from the memory-based representation in the feature memory. The decoder of the MAu is composed of four stacking ConvTranspose layers with BN and LeakyReLU on the first three layers.

Feature memory: The feature memory stores the prototypical encoded feature representations with the input of the feature representation from the encoder, as shown in Fig. 4. The feature memory is used to store the features during training, with random values as its initial content.

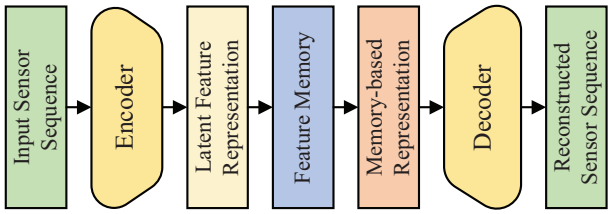


Fig. 3: Architecture of MAu.

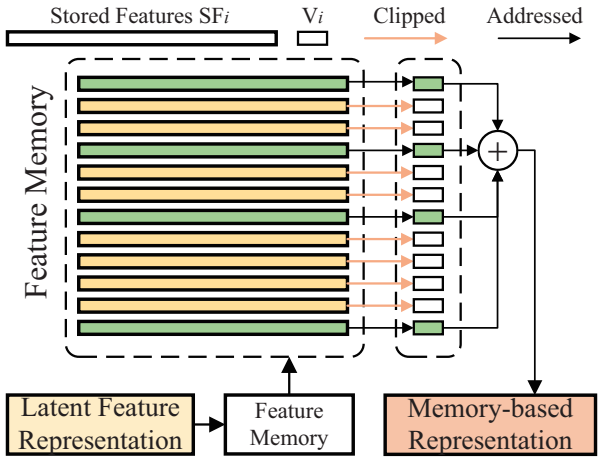


Fig. 4: Architecture of feature memory.

The latent features (strips) in the feature memory will be clipped (orange strips) or addressed (green strips) to vectors (green bars), which are then computed to memory-based representation according to the following addressing method.

In order to obtain address vector V, we exploit the cosine similarity to compute the similarity between the latent feature representation LR from the decoder and each of the stored feature SF in the feature memory by Eq. (1) and then use the $softmax()$ function to normalize the similarity by Eq. (2):

$$CS(LR, SF_i) = \frac{LR \times SF_i^{\intercal}}{\|LR\|\|SF_i\|}, \tag{1}$$

$$V_i = softmax(CS(LR, SF_i)), \tag{2}$$

where SF_i is the ith stored feature in the feature memory, and V_i represents the ith value of the address vector V.

In order to further weaken the reconstruction ability of abnormal samples, we apply the clip operation on the address vector as in Eq. (3):

$$V_i = \begin{cases} V_i, & V_i > \lambda \\ 0, & otherwise \end{cases} \tag{3}$$

where $i \in [1, memcap]$, $memcap$ indicates the capacity of the feature memory (a hyper-parameter), and $\lambda = \frac{1}{memcap}$ is a threshold. Eq. (3) indicates V_i is set to itself if the similarity is greater than the threshold λ; otherwise $V_i = 0$.

Memory-based representation: The memory-based representation MR output by the feature memory can be expressed as Eq. (4):

$$MR = \sum_{i=1}^{memcap} (SF_i \times V_i) \tag{4}$$

Based on the memory-based representation, the decoder outputs the reconstructed sensor sequence, which are compared with the input sensor sequence to measure the reconstruction error. We use the reconstruction error as the criterion for classification, by deciding the user to be an impostor if this error is greater than the predefined threshold.

6.2 Data reconstruction:

We design the MAu architecture for data reconstruction, as illustrated in Table 3. Given the accelerometer and gyroscope sensor data of $2 \times 200 \times 3$ as the input, the first Convolution layer with 32 kernels, kernel size of 3×1, padding of 1×0 and stride 1×1 generates a $32 \times 200 \times 3$ output. In order to increase channels from 32 to 128, three consecutive Convolution layers with kernel size of 3×3, padding of 1×0, stride of 2×1, and kernels of $64, 128, 128$, respectively, are applied and $128 \times 25 \times 3$ of latent feature representation is obtained. We apply the BN and LeakyReLU on each convolution layer. In our work, the dimension of the stored feature is set to 128. Then, the latent feature representation is reshaped to 128×75 and then fed to feature memory. After memory reconstruction, memory-based representation of 128×75 is obtained and then reshaped back to $128 \times 25 \times 3$ for the input of the decoder. The memory-based representation of $128 \times 25 \times 3$ is fed to four layers of ConvTranspose with kernel size of 3×1, padding of 1×0, stride of 2×1 (1×1 for the last layer), and kernels of $128, 64, 32, 2$, respectively, to reconstruct data of $2 \times 200 \times 3$. The second dimension of the kernel size and the stride in the encoder and decoder are set to 1, so that they reconstruct the data of a single axis (x, y, z) of a sensor, thereby extracting and reconstructing features of different axes. We apply the BN and LeakyReLU on the first three ConvTranspose layers.

6.3 Authentication

The reconstruction error is the difference between the reconstructed sensor sequence output by the decoder and the input sensor sequence to the encoder. If the error is larger than the predefined threshold, the user is recognized as an imposter. The reconstruction error is calculated by the mean square error (MSE) in Eq. (5):

$$MSE = \frac{1}{i \times j \times k} \sum_{i=1}^{2} \sum_{j=1}^{200} \sum_{k=1}^{3} (O_{ijk} - R_{ijk})^2 \tag{5}$$

where O indicates the input sensor data, R represents the reconstructed sensor data by the MAu, and i, j, k denote data dimensions.

In real-time applications, the MSE of 2-second sensor data is computed and then compared with a predefined authentication threshold, which is assigned according to the best result on the cross-validation tests. If the MSE is less than the threshold, the user is identified as a legitimate user; otherwise, the user is classified as an imposter and the phone will be locked immediately.

7 Performance Evaluation

We, in this section, evaluate the performance of MAuGANs based on the collected 70 subjects' dataset. For each subject, we randomly select 80% data as the training data and the rest 20% as the testing data. After training, we test the trained MAu model 70 times for each subject, on the datasets consisting of the real testing data (20% of the subject's data) as positive data and randomly-selected data from other subjects as negative data in a proportion of $2:1$. This is because the legitimate user has more chances to access the device than imposters. For the predefined authentication threshold, we add different numbers of legitimate users to the testing data in order to show that the system can correctly distinguish between legitimate users and imposters. During the test, we assign the threshold by the value when the system obtains a balanced EER. In practical, due to the absence of imposters' data, the average value of legitimate users' data reconstruction errors can be used as the threshold. To evaluate the performance of MAuGANs, we describe the experimental setup and then detail the extensive experiments. For the evaluation experiments, we start with the performance of MAuGANs. Then, we evaluate the efficiency of CTGANs, the effectiveness of data augmentation, and the efficiency of the MAu, and next analyze the security of MAuGANs. Finally, we compare MAuGANs with the representative state-of-the-art methods.

7.1 Experimental Setup

In this section, we describe our experimental setup including the dataset collection, the training of the CTGAN and MAu, and the evaluation metrics.

7.1.1 Dataset

To collect the experimental data, we developed an Android phone-based data collection tool to acquire users' behavioral patterns when they interact with their phones [61]. For our dataset collection, we recruited 100 volunteers including 53 males and 47 females to interact with the experimental phones, with the IRB approval from William & Mary. To obtain high-quality data, we instructed subjects to execute three interactive tasks: document reading, text producing, and map navigating for locating a destination, which largely encompassed the behaviors of users using their mobile devices. Once the subjects signed in the data collection tool, one of the three tasks was automatically assigned. For each session, the subject either sat or walked to finish the tasks, which lasted about 5 to 15 minutes. The subjects performed 24 sessions (8 reading sessions, 8 writing sessions, and 8 map navigating sessions) in the same way as an ordinary user would do, producing 2 to 6 hours of behavior traits in total. The sensor readings of the accelerometer and gyroscope with a sampling rate $f = 100$ Hz were stored as CSV files on the experimental devices.

In our experiments, we select 70 subjects' data from the collected accelerometer and gyroscope sensor data on the experimental phones, where the first 100-minute data of each subject (with a size of the 2-second time window) were chosen as our experimental dataset.

TABLE 3: MAu Architecture

Operators	Output	# Kernel	KSize	Padding	Stride
Sensor Data	$2 \times 200 \times 3$	–	–	–	–
Conv+BN+LeakyReLu	$32 \times 200 \times 3$	32	$(3, 1)$	$(1, 0)$	$(1, 1)$
Conv+BN+LeakyReLu	$64 \times 100 \times 3$	64	$(3, 1)$	$(1, 0)$	$(2, 1)$
Conv+BN+LeakyReLu	$128 \times 50 \times 3$	128	$(3, 1)$	$(1, 0)$	$(2, 1)$
Conv+BN+LeakyReLu	$128 \times 25 \times 3$	128	$(3, 1)$	$(1, 0)$	$(2, 1)$
Feature Memory	$128 \times 25 \times 3$	–	–	–	–
ConvTranspose+BN+LeakyReLu	$128 \times 50 \times 3$	128	$(3, 1)$	$(1, 0)$	$(2, 1)$
ConvTranspose+BN+LeakyReLu	$64 \times 100 \times 3$	64	$(3, 1)$	$(1, 0)$	$(2, 1)$
ConvTranspose+BN+LeakyReLu	$32 \times 200 \times 3$	32	$(3, 1)$	$(1, 0)$	$(2, 1)$
ConvTranspose	$2 \times 200 \times 3$	2	$(3, 1)$	$(1, 0)$	$(1, 1)$

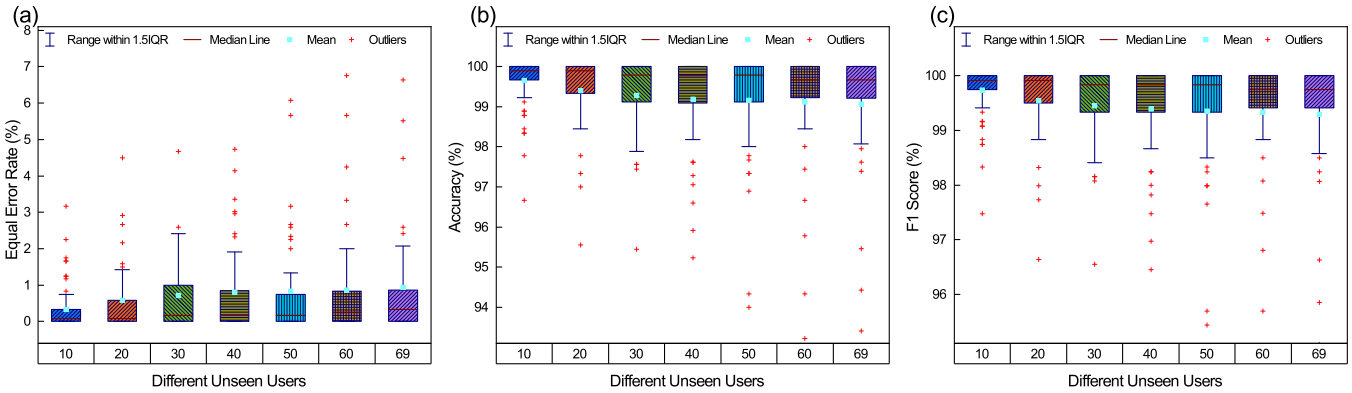


Fig. 5: EER, accuracy, and F1 score for MAuGANs on varying number of unseen users.

TABLE 4: EER, FAR, FRR, Accuracy, and F1 score (%) with SD on Different Number of Unseen Users

Unseen User	10	20	30	40	50	60	69
EER (SD)	0.33 (0.60)	0.58 (1.40)	0.71 (1.45)	0.80 (1.66)	0.83 (1.81)	0.86 (1.99)	0.93 (2.15)
FAR (SD)	0.25 (0.60)	0.52 (1.44)	0.70 (1.50)	0.77 (1.68)	0.78 (1.84)	0.80 (2.00)	0.89 (2.18)
FRR (SD)	0.40 (0.62)	0.64 (1.39)	0.73 (1.42)	0.84 (1.66)	0.88 (1.81)	0.92 (1.99)	0.97 (2.13)
Accuracy (SD)	99.65 (0.60)	99.40 (1.40)	99.28 (1.44)	99.19 (1.67)	99.15 (1.81)	99.12 (1.99)	99.06 (2.14)
F1 score (SD)	99.73 (0.45)	99.54 (1.07)	99.46 (1.10)	99.40 (1.25)	99.36 (1.40)	99.33 (1.54)	99.30 (1.63)

7.1.2 Training

Conditional transformer GAN training: we train two independent CTGANs for the accelerometer and gyroscope, respectively, only on the corresponding 70 subjects' data associated with their one-hot labels, with batch size of 128. For each CTGAN, with a batch of 100-dimensional Gaussian noise and one-hot labels of 70 subjects as the input, the generator creates sensor data associated with their one-hot labels. With the generated data and real data as the inputs, the discriminator seeks to distinguish them from each other and only the classified true data associated with the user true label can be recognized as true. In the CTGAN training, for both the generator and discriminator, we apply the WGAN-GP loss function and Adam optimizer to update the learning rate for 200 epochs with an initial value of 0.000001 and a weight decay of 0.00001. We train the generator once for every 2 epochs of the discriminator training. After the CTGANs are well trained, only the generators remain for accelerometer and gyroscope sensor data augmentation, respectively.

Memory-Augmented Autoencoder training: We train the MAu only for the legitimate user for real-time authentication in practice. We combine the randomly-selected 80% data of the legitimate user with the corresponding CTGAN-augmented data in the proportion of 1 : 2 as the training data for the MAu and the rest 20% with are utilized for testing, with a batch size of 1,024. We feed each batch of user training data to the corresponding MAu, calculate the reconstruction error on the input and output of each epoch, then perform back-propagation, and finally update the parameters for the learning rate. We also apply Adam optimizer to update the learning rate with the initial value of 0.0001 for 800 epochs. The MAu has approximately 20M parameters and the average training time of the MAu is about 320s (about 0.4s per epoch on a 3070 GPU and 800 epochs in total), which makes the training of our MAu effortless.

7.1.3 Metrics

We employ the FAR (false acceptance rate), FRR (false rejection rate), EER (equal error rate), accuracy, and F1 score, as our evaluation metrics. The latter are widely used in continuous authentication systems to comprehensively assess the performance. FAR indicates the percentage of the number of imposters' samples that are falsely identified as legitimate samples w.r.t the total number of imposters' samples [62]. The FRR represents the ratio between the number of falsely rejected legitimate samples and the number of all legitimate samples [63]. The EER is the intersection value when the FAR equals the FRR [8]. It provides a good trade-off as a high FAR might lead to the admission of more imposters, while a high FRR might cause the rejection of more legiti-

mate users. The accuracy measures the likelihood that the system accepts legitimate users and rejects imposters [25]. F1 score is the harmonic mean of the precision and recall; its best and worst scores are 1 and 0 respectively [17].

7.2 Overall Performance of MAuGANs

We evaluate the overall performance of our system MAuGANs in terms of EER, FAR, FRR, accuracy, and F1 score, by varying the number of unseen users. To conduct the experiment, we first randomly select one subject from 70 subjects as a legitimate user, where 80% data of the user are used to train MAuGANs. Then, we randomly select n subjects as imposters (unseen users) from the remaining 69 subjects. The n-imposters' data are combined with the rest 20% legitimate user's in the proportion of $1:2$ to validate MAuGANs, with $n = 10, 20, ..., 60, 69$. We calculate the corresponding metrics until each subject is selected once as the legitimate user. Fig. 5 depicts the box plots of the EER, accuracy, and F1 score for MAuGANs for a varying number of unseen users. As illustrated in Fig. 5, the EER, accuracy, and F1 score show a relative stable and high-precision trend for all the different unseen users. Specifically, with the increase of the number of the unseen users, the mean EER slightly grows under 1% (Fig. 5(a)), the accuracy slowly decreases over 99% (Fig. 5(b)), and the F1 score slightly drops above 99% (Fig. 5(c)), which indicate MAuGANs has high-precision authentication performance. For further analysis, Table 4 lists the EER, FAR, FRR, accuracy and F1 score with SD (standard deviation) for MAuGANs on different numbers of unseen users. As tabulated in Table 4, with the growing number of unseen users, EER, FAR, and FRR slightly increase, while the accuracy and F1 score slowly decrease. MAuGANs achieves the best performance of 0.33% EER, 0.25% FAR, 0.40% FRR, 99.65% accuracy on 10 unseen users, and 99.73% F1 score, and attains the worst results of 0.93% EER, 0.89% FAR, 0.97% FRR, 99.06% accuracy, and 99.30% F1 score on 69 unseen users. From Fig. 5 and Table 4, we can conclude that MAuGANs achieves high-precision and effectiveness in verifying legitimate users with less than 1% EER and more than 99% accuracy for different numbers of unseen users.

7.3 Efficiency of Conditional Transformer GANs

To assess the efficiency of the proposed CTGANs, we explore three metrics to measure the quality of the created data: the discriminator loss, MMD (maximum mean discrepancy), and t-SNE (t-distributed stochastic neighbor embedding). The discriminator loss measures the Earth-Mover distance between the genuine sensor samples and the created samples until the network converges; the higher the quality of the generated samples, the closer the loss is to 0 [64]. The discriminator losses of the generated data of accelerometer and gyroscope sensors are visualized in Fig. 6. As illustrated in Fig. 6, with the growth of the training epochs, the discriminator loss of the generated accelerometer sensor data first sharply drops, then increases immediately close to 0, and then slightly drops and gradually grows to 0 until 140 epochs. The discriminator loss of gyroscope slightly oscillates around a small value and reaches 0 until 70 epochs. Thus, the discriminator losses converging to 0 indicate that the generated accelerometer and gyroscope sensor data show similar distributions to their real data, and thus have high quality.

The MMD calculates the distance between the genuine sensor sample distribution and the created sensor sample distribution; the higher the quality of the created samples, the closer the MMD is to 0 [65]. Based on the trained CTGAN, we compute the MMD of 50 created samples and 50 genuine samples at one epoch until 200 epochs, for the accelerometer and gyroscope, respectively, as illustrated in Fig. 7. As shown, the MMD of the accelerometer sensor data decreases slowly until 100 epochs, then sharply drops, and finally slightly trends towards a very small value (approximately 0) around 125 epochs while that of the gyroscope data oscillates around 100 and falls to a small value around 125 epochs, with the increase of the epochs. MMDs close to 0 mean that the created sensor samples show high quality.

t-SNE maps the high dimensions of the created data samples non-linearly into 2D, which can be visualized [66]. We generate the same amount of data as the real data's amount for the accelerometer and gyroscope sensors, respectively, and then randomly select 500 samples for each sensor from these data. Then, we visualize the distributions of the 500 generated samples and 500 real samples of the two sensors by t-SNE in Fig. 8. Blue star and orange star indicate the genuine and created samples of the accelerometer, while red triangle and purple triangle represent the genuine and created samples of the gyroscope. As depicted in Fig. 8, the sensor data are clearly divided by colors into two clusters associated with the two sensors. In each cluster, most of the genuine data and created data overlap, which illustrates the created sensor data show a quite close distribution to the real data. Therefore, we can conclude that the proposed CTGANs generate high-quality sensor data for MAuGANs.

7.4 Effectiveness of Data Augmentation

We investigate the effectiveness of the proposed CTGANs-based data augmentation by comparing MAuGANs with CTGANs and MAuGANs without CTGANs on different data sizes. To conduct the experiment, we randomly select one out of 70 subjects as a legitimate user and the remaining 69 as imposters, in order to train and test MAuGANs until each subject is selected once as the legitimate user; we choose the worst settings of 69 unseen users from Table 4. With the data size growing from 100 to 500, we illustrate the EER, accuracy, and F1 score for MAuGANs with CTGAN (orange box) and MAuGANs without the augmentation (blue box) in Fig. 9. As illustrated in Fig. 9, MAuGANs with data augmentation overall outperforms that without augmentation. The performance slightly fluctuates with the increase of the data sizes, and the CTGAN shows the best performance with the data size of 200. Compared to that without data augmentation, MAuGANs always achieves lower EER (Fig. 9(a)), higher accuracy (Fig. 9(b)), and higher F1 score (Fig. 9(c)), which indicate the effectiveness of the proposed CTGANs on MAuGANs. In addition, we list the EER, FAR, FRR, accuracy and F1 score with or without data augmentation approaches over different data sizes in the first two rows of Table 5. As depicted, MAuGANs with CTGAN data augmentation reaches the best EER of 2.56%,

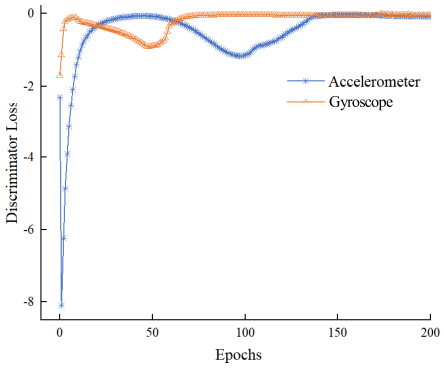


Fig. 6: Discriminator loss.

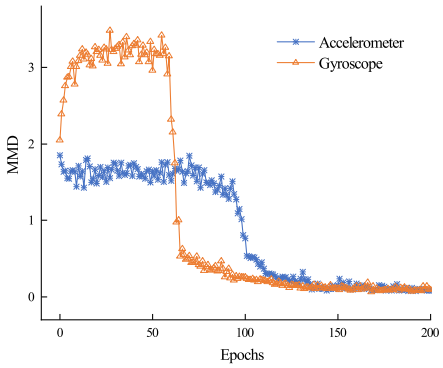


Fig. 7: MMD.

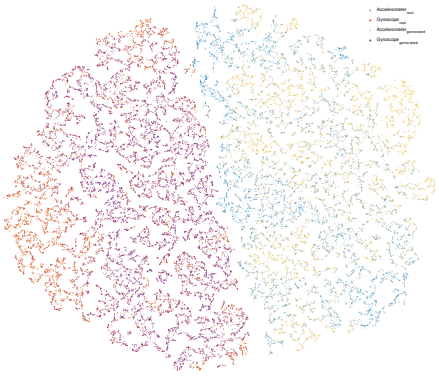


Fig. 8: Distribution of genuine and created samples by t-SNE.

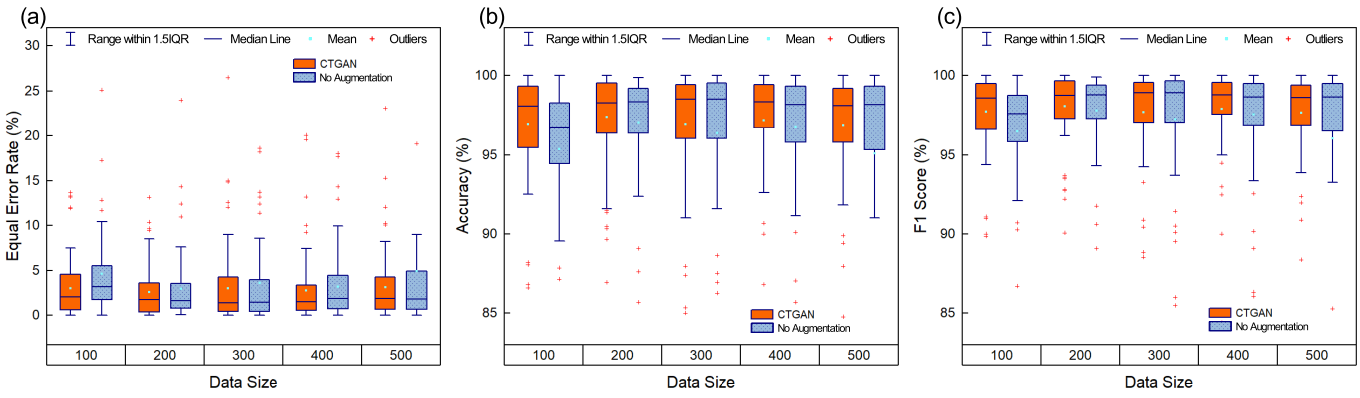


Fig. 9: EER, accuracy, and F1 score of MAuGANs with or without CTGAN over different data sizes.

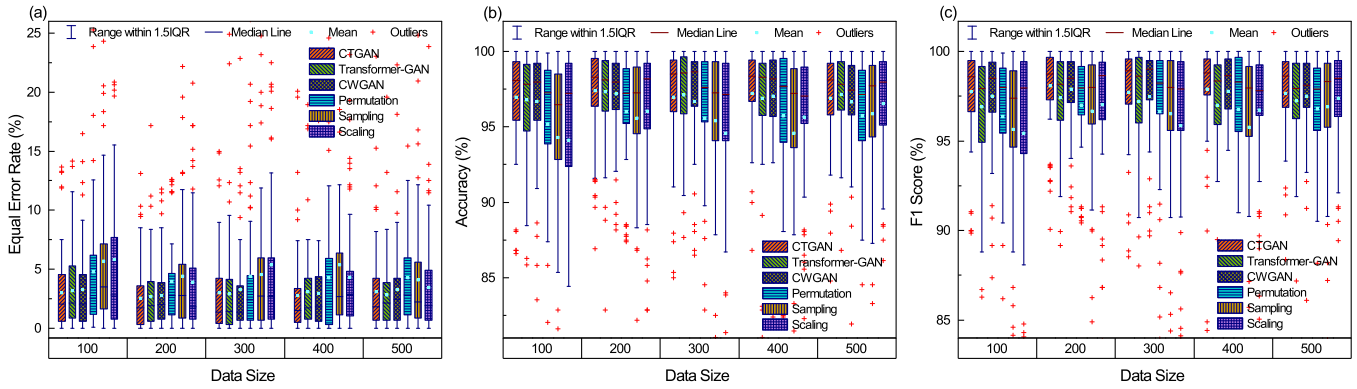


Fig. 10: EER, Accuracy, and F1 score with different data augmentation methods over different data sizes.

FAR of 2.44%, FRR of 2.67%, accuracy of 97.40%, and F1 score of 98.07% on data size of 200, which show margin improvements of 0.36% on EER, 0.40% on FAR, 0.33% on FRR, 0.35% on accuracy, and 0.99% on F1 score, compared to no data augmentation.

To further verify the effectiveness of CTGANs-based data augmentation approach, we compare our CTGAN with representative data augmentation approaches, such as deep learning based methods – transformer-GAN [17] and CWGAN [38], and geometric transformation based methods – permutation, scaling, and sampling. Based on the setting of 69 unseen users, we conduct the same experiment as MAuGANs evaluation in Section 7.2, by replacing the CTGAN in MAuGANs with the aforementioned representative data augmentation approaches. With the data size growing from 100 to 500, we plot boxes of the EER, accuracy, and F1 score with the representative data augmentation approaches, as illustrated in Fig. 10. As shown, the performances of all the data augmentation approaches vary with the increase of the data sizes. In particular, the proposed CTGAN outperforms all the representative data augmentation approaches in terms of EER, accuracy and F1 score. We note also that deep learning-based methods overall show better performance than geometric transformation-based methods (except scaling on data size of 500). Moreover, we list the corresponding EER, FAR, FRR, accuracy, and F1 score with

TABLE 5: EER, FAR, FRR, Accuracy and F1 Score (%) (SD) With Different Data Augmentation Methods Over Different Data Sizes

Approach	Metric	100	200	300	400	500
No Augmentation	EER	4.59 (5.74)	2.92 (3.85)	3.59 (5.72)	3.19 (3.91)	4.89 (11.02)
	FAR	4.46 (5.76)	2.84 (3.91)	3.54 (5.79)	3.07 (3.94)	4.80 (11.05)
	FRR	4.72 (5.74)	3.00 (3.80)	3.64 (5.67)	3.30 (3.89)	4.97 (11.00)
	Accuracy	95.36 (5.74)	97.05 (3.83)	96.39 (5.70)	96.77 (3.90)	95.08 (11.01)
	F1 Score	96.48 (4.58)	97.80 (2.95)	97.26 (4.50)	97.58 (2.98)	96.05 (9.92)
CTGAN	EER	3.02 (3.38)	2.56 (2.92)	3.00 (4.38)	2.76 (3.84)	3.07 (3.91)
	FAR	2.99 (3.51)	2.44 (3.00)	2.92 (4.49)	2.69 (3.91)	3.01 (4.01)
	FRR	3.05 (3.27)	2.67 (2.87)	3.08 (4.30)	2.84 (3.80)	3.16 (3.83)
	Accuracy	96.97 (3.34)	97.40 (2.90)	96.97 (4.35)	97.20 (3.83)	96.88 (3.88)
	F1 Score	97.74 (2.52)	98.07 (2.18)	97.72 (3.36)	97.91 (2.94)	97.67 (2.97)
Transformer-GAN [17]	EER	3.20 (3.22)	2.67 (3.00)	2.89 (4.39)	3.09 (3.84)	2.84 (3.17)
	FAR	3.20 (3.31)	2.57 (3.08)	2.81 (4.45)	3.05 (3.90)	2.73 (3.27)
	FRR	3.19 (3.16)	2.76 (2.97)	2.97 (4.35)	3.13 (3.80)	2.94 (3.13)
	Accuracy	96.80 (3.22)	97.33 (3.00)	97.11 (4.38)	96.91 (3.84)	97.16 (3.18)
	F1 Score	96.90 (3.12)	97.41 (2.91)	97.19 (4.27)	97.00 (3.73)	97.25 (3.09)
CWGAN [38]	EER	3.27 (3.50)	2.77 (2.79)	3.30 (4.87)	2.94 (4.09)	3.30 (3.87)
	FAR	3.14 (3.60)	2.68 (2.87)	3.23 (4.91)	2.92 (4.13)	3.20 (3.93)
	FRR	3.40 (3.46)	2.86 (2.77)	3.37 (4.84)	2.97 (4.05)	3.40 (3.82)
	Accuracy	96.67 (3.48)	97.20 (2.78)	96.67 (4.86)	97.05 (4.07)	96.67 (3.85)
	F1 Score	97.52 (2.64)	97.91 (2.10)	97.49 (3.78)	97.78 (3.12)	97.50 (2.96)
Permutation	EER	4.77 (6.13)	3.98 (4.57)	4.40 (5.92)	4.26 (5.39)	4.27 (4.51)
	FAR	4.67 (6.23)	3.90 (4.65)	4.31 (5.99)	4.19 (5.47)	4.28 (4.58)
	FRR	4.86 (6.04)	4.05 (4.52)	4.48 (5.88)	4.32 (5.32)	4.26 (4.45)
	Accuracy	95.20 (6.09)	96.00 (4.55)	95.57 (5.90)	95.72 (5.36)	95.73 (4.49)
	F1 Score	96.36 (4.81)	96.99 (3.54)	96.64 (4.60)	96.77 (4.14)	96.80 (3.46)
Scaling	EER	5.84 (8.50)	3.93 (4.65)	5.39 (7.03)	4.33 (4.90)	3.44 (4.25)
	FAR	5.70 (8.49)	3.87 (4.69)	5.32 (7.07)	4.26 (4.97)	3.39 (4.33)
	FRR	5.98 (8.50)	4.00 (4.63)	5.46 (6.99)	4.40 (4.86)	3.50 (4.20)
	Accuracy	94.11 (8.49)	96.04 (4.64)	94.58 (7.01)	95.64 (4.89)	96.53 (4.23)
	F1 Score	95.45 (6.93)	97.02 (3.56)	95.86 (5.50)	96.72 (3.76)	97.40 (3.24)
Sampling	EER	5.68 (6.71)	4.44 (5.42)	4.56 (5.61)	5.39 (9.32)	4.11 (4.93)
	FAR	5.59 (6.76)	4.46 (5.53)	4.48 (5.60)	5.32 (9.40)	4.09 (5.01)
	FRR	5.77 (6.67)	4.41 (5.33)	4.63 (5.63)	5.47 (9.25)	4.14 (4.86)
	Accuracy	94.29 (6.70)	95.57 (5.38)	95.41 (5.62)	94.58 (9.29)	95.88 (4.90)
	F1 Score	95.65 (5.33)	96.66 (4.18)	96.53 (4.35)	95.76 (8.11)	96.90 (3.78)

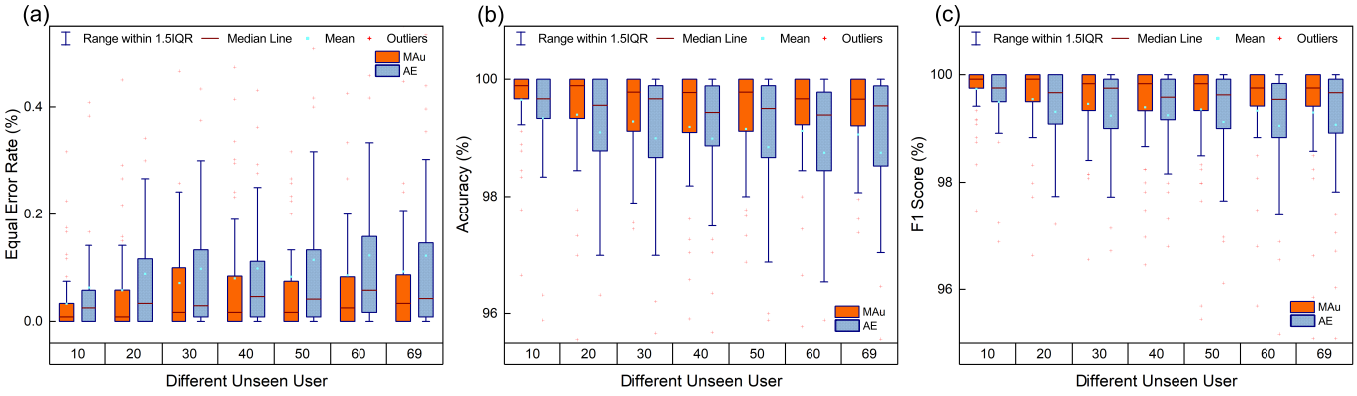


Fig. 11: EER, Accuracy, and F1 Score of MuGANs with MAu or AE over Different Unseen Users.

TABLE 6: EER, FAR, FRR, Accuracy, and F1 score (%) (SD) on MAu or AE Over Different Unseen Users

Approach	Metric	10	20	30	40	50	60	69
MAu	EER	0.33 (0.60)	0.58 (1.40)	0.71 (1.45)	0.80 (1.66)	0.83 (1.81)	0.86 (1.99)	0.93 (2.15)
	FAR	0.25 (0.60)	0.52 (1.44)	0.70 (1.50)	0.77 (1.68)	0.78 (1.84)	0.80 (2.00)	0.89 (2.18)
	FRR	0.40 (0.62)	0.64 (1.39)	0.73 (1.42)	0.84 (1.66)	0.88 (1.81)	0.92 (1.99)	0.97 (2.13)
	Accuracy	99.65 (0.60)	99.40 (1.40)	99.28 (1.44)	99.19 (1.67)	99.15 (1.81)	99.12 (1.99)	99.06 (2.14)
	F1 Score	99.73 (0.45)	99.54 (1.07)	99.46 (1.10)	99.40 (1.25)	99.36 (1.40)	99.33 (1.54)	99.30 (1.63)
AE	EER	0.63 (1.58)	0.89 (1.96)	0.98 (1.93)	0.99 (1.95)	1.15 (2.09)	1.22 (2.36)	1.22 (2.47)
	FAR	0.53 (1.61)	0.82 (2.00)	0.89 (1.95)	0.91 (1.95)	1.11 (2.17)	1.16 (2.37)	1.16 (2.51)
	FRR	0.73 (1.56)	0.95 (1.95)	1.06 (1.92)	1.06 (1.97)	1.18 (2.03)	1.29 (2.35)	1.30 (2.45)
	Accuracy	99.33 (1.57)	99.09 (1.95)	99.00 (1.92)	98.99 (1.96)	98.84 (2.07)	98.75 (2.35)	98.75 (2.47)
	F1 Score	99.49 (1.21)	99.31 (1.52)	99.24 (1.48)	99.24 (1.49)	99.12 (1.60)	99.05 (1.84)	99.05 (1.89)

data size growing from 100 to 500 in Table 5, respectively. As described in Table 5, for the performance of the EER, Transformer-GAN, CWGAN, and permutation gain their best EERs of 2.67%, 2.77% and 3.98% on data size of 200, respectively; Scaling and sampling achieve their best EERs of 3.44% and 4.11% on data size of 500. In comparison, the CTGAN obtains the lowest EER of 2.56% on data size of 200, by margins of 0.11% (2.67%, Transformer-GAN on data size of 200) at least. For the performance of the accuracy, Transformer-GAN, CWGAN, and permutation obtain their best accuracy of 97.33%, 97.20% and 96.00% on data size of 200, respectively; Scaling and sampling achieve their best accuracy of 96.53% and 95.88% on data size of 500. In comparison, the CTGAN receives the highest EER of 97.40% on data size of 200, by margins of 0.07% (97.33%, Transformer-GAN on data size of 200) at least.

7.5 Efficiency of Memory-Augmented Autoencoder

We examine the efficiency of the MAu by comparing the proposed MAu with the typical autoencoder (AE) [57]. We conduct the same experiment as MAuGANs evaluation in Section 7.2 by replacing the MAu with the AE. The corresponding results of the EER, accuracy and F1 score of MAuGANs with the MAu or AE over different unseen users are illustrated in Fig. 11. As demonstrated in Fig. 11, the MAu in MAuGANs overall outperforms the AE by showing lower EER, higher accuracy and higher F1 score, respectively. Specifically, the EERs slowly grow below 0.10% and 0.20% (Fig. 11(a)), accuracy gradually decreases over 99% and 98% (Fig. 11(b)), and F1 score slowly reduces above 99% and 98% (Fig. 11(c)) for the MAu and AE, respectively, as the number of unseen users increases. Moreover, the EER, FAR, FRR, accuracy and F1 score on the MAu or AE over different unseen users are tabulated in Table 6. As listed in Table 6, the performance of both the MAu and AE in MAuGANs gradually degrade with the growth of the unseen users, but the MAu is always superior to the AE. For the EER, the MAu achieves the best EER of 0.33% on 10 unseen users and the worst 0.93% on 69 unseen users, while the AE receives the best EER 0.63% on 10 unseen users and the worst 1.22% on 69 unseen users, with improvement margins of 0.30% and 0.29%, respectively. For the accuracy, the MAu reaches the best 99.65% on 10 unseen users and the worst 99.06% on 69 unseen users, while the AE obtains 99.33% and 98.75%, respectively, with improvement margins of 0.32% and 0.31%. For the F1 score, the MAu gains the best 99.73% and the worst 99.30%, while the AE obtains 99.49% and 99.05%, with improvement margins of 0.24% and 0.25%, respectively. In summary, the MAu is more effective and suitable than the AE for reconstructing user data in MAuGANs.

7.6 Security analysis

We investigate the performance of MAuGANs in user authentication against mimic attack, where an adversary is expected to try its best to impersonate the legitimate user's behavioral patterns. For the mimic attack, we first select one out of 70 subjects as the legitimate user and then inject the random Gaussian noise ranging from 0 to 0.1, which imitates a very tiny sensor vibration affecting the legitimate user's data. Next, we use half legitimate user's original data with half impersonated data as the test data to verify the corresponding legitimate user's MAu. We repeat the above experiments (70 times) until each of the 70 subjects is selected as the legitimate user once. The performance of MAuGANs in defending against the mimic attack is tabulated in Table 7. As shown, MAuGANs achieves 3.28% EER, 3.09% FAR, 3.47% FRR, 96.72% accuracy and 96.72% F1 score. That is to say, the unique accelerometer and gyroscope sensor data are still hard to imitate by an adversary with similar behavioral patterns. Note that we did not train a user to mimic the legitimate user's behavioral patterns, as the data with small amount of added noise are closer to the real data.

TABLE 7: Performance (%) (SD) of MAuGANs in Defending Against the Mimic Attack

EER	FAR	FRR	Accuracy	F1 score
3.28 (3.65)	3.09 (3.78)	3.47 (3.56)	96.72 (3.65)	96.72 (3.65)

7.7 Comparison with representative methods

We show the superiority of the proposed MAuGANs by comparing MAuGANs with the state-of-the-art authentication methods with data augmentation, namely SensorAuth [30], RobustTAD [31], FRDA-HAR [32], EchoPrint [33], CAuSe [34], MODALS [36], Expert–gatingCNN [37], CAGANet [38], and ADFFDA [17], as listed in Table 8. As shown in Table 8, we display the methods, data source, data augmentation approaches, and accuracy for these state-of-the-art data augmentation approaches. Concretely, SensorAuth utilizes five transformation methods, namely permutation, sampling, scaling, jittering, and cropping, to generate acccelerometer and gyroscope sensor data and gains a 19.04% EER with data size of 100 on the OC-SVM classifier [30]. RobustTAD uses flipping and downsampling in the time domain, and magnitude and phase in the frequency domain on the U-Net-DeWA dataset with data size of 100 to reach a F1 score of 87.25% [31]. FRDA-HAR exploits the local averaging as a downsampling technique to create accelerometer and gyroscope sensor data for human activity classification and reaches a 88.14% accuracy with a deep LSTM classifier and batch size of 8000 [32]. EchoPrint explores the projection matrix rotation imitating different camera poses to augment face images and obtains 81.78% balanced accuracy (BAC) with vision features [33]. CAuSe utilizes an AAS-based optimal strategy for the accelerometer, gyroscope and magnetometer data augmentation, and achieves an accuracy of 91.12% and EER of 5.68% on the LOF classifier with data size of 100 [34]. MODALS uses four universal data transformation operations, namely hard example interpolation, hard example extrapolation, Gaussian noise and difference transform, to augment HAR data, and reaches a 91.87% accuracy on the MLP classifier with 256 units [36]. Expert–gatingCNN exploits expert networks and gate networks to search for optimal weights for four-corner cropping and center cropping along with flipping to realize data augmentation for activity classification. It reaches a 76% accuracy on the gating classifier with 224×224 video flips [37]. CAGANet utilizes a conditional Wasserstein GAN

TABLE 8: Comparison With Data Augmentation Based Authentication Approaches

Method	Data Source	Data Augmentation Approach	Accuracy (%)
SensorAuth [30]	Acc., Gyr.	permutation, sampling, scaling, cropping, jittering	EER: 19.04 (OC-SVM, 100)
RobustTAD [31]	Time-series	Flipping, downsampling, magnitude, phase	F1: 87.25 (U-Net-DeWA, 100)
FRDA-HAR [32]	Acc., Gyr.	Downsampling	Acc: 88.14 (DLSTM, 8000)
EchoPrint [33]	Face image	Rotation	BAC: 81.78 (vision features)
CAuSe [34]	Acc., Gyr., Mag.	AAS-based optimal strategy	Acc: 91.12; EER: 5.68 (LOF, 100)
MODALS [36]	Acc., Gyr.	interpolation, extrapolation, Gaussian, transform	Acc: 91.87 (MLP, 256)
Expert–gatingCNN [37]	Video	cropping, flipping	Acc: 76.00 (gating, 224×224)
CAGANet [38]	Acc., Gyr., Mag.	CWGAN	Acc: 90.08; EER: 8.78 (LOF, 100)
ADFFDA [17]	Acc., Gyr., Mag.	Transformer-GAN	EER: 1.62 (OC-SVM, 700)
MAuGANs	Acc., Gyr.	CTGAN	Acc: 99.06; EER: 0.93 (69 unseen users)

(CWGAN) to create acccelerometer, gyroscope, and magnetometer data and reaches an accuracy of 90.08% and an EER of 8.78% with data size of 100 on the LOF classifier [38]. ADFFDA uses a transformer-based GAN to augment acccelerometer, gyroscope, and magnetometer data and obtains an EER of 1.62% on the OC-SVM classifier with a dataset of size 700 [17].

Comparing to the state-of-the-art data augmentation-based authentication methods, MAuGANs utilizes the conditional transformer GAN to augment accelerometer and gyroscope sensor data and achieves the best accuracy of 99.06% and EER of 0.93% with 69 unseen users (the worst case).

8 Conclusion

In this paper, we have presented a lightweight and practical continuous authentication system, named MAuGANs, based on the CTGAN and MAu, leveraging the smartphone accelerometer and gyroscope built-in sensors. MAuGANs uses the CTGAN to augment sensor data for the authentication model that is trained on multiple users' data simultaneously. To identify users, MAuGANs exploits the MAu which is trained only on the legitimate user's data. In MAuGANs, the user performs widely-adopted operations on the commercial mobile devices, which are implicitly processed without user's awareness, for his/her continuous authentication. We validate the performance of MAuGANs on our dataset, and the extensive experiments demonstrate that MAuGANs outperforms the representative state-of-the-art approaches. Nonetheless, although MAuGANs achieves superior performance, it highly relies on user behaviors during a short enrollment phase, making it incapable of dealing with an ever-changing user pattern. To address this issue, we can make the enrollment phase longer to allow the behavior converges to a stable state, or repeat the enrollment phase periodically. In addition, our data were collected on predefined tasks and proper experimental devices, which limits the applicability of MAuGANs. To address this limitation, we plan to collect comprehensive data on more common tasks (e.g., application usage, interactive game playing, or web browsing) and different types of devices (e.g., tablets, smartwatches, or smartphones) under different experimental conditions (e.g., standing, sitting or walking; indoor or outdoor). Furthermore, we plan to train a user to mimic the legitimate user's behavior to further enhance our threat model.

References

[1] D. Wang, Q. Gu, X. Huang and P. Wang, "Understanding human-chosen pins: Characteristics, distribution and security," In *Proc. ACM on Asia Conf. Comput. Commu. Secu. (ASIA CCS)*, 2017, pp. 372–385.

[2] 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 technol. (WOOT)*, 2010, pp. 1–7.

[3] 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)*. 2006, pp. 177–184.

[4] M. Zhou, Q. Wang, J. Yang, Q. Li, F. Xiao, Z. Wang, and X. Chen, "PatternListener: Cracking Android Pattern Lock Using Acoustic Signals," In *Proc. 2018 ACM SIGSAC Conf. Comput. Commu. Secur. (CCS)*. 2018, pp. 1775–1787.

[5] Y. Chen, T. Ni, W. Xu and T. Gu, "SwipePass: Acoustic-based Second-factor User Authentication for Smartphones," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* vol. 6, no. 3, Article 106 (September 2022), 25 pages.

[6] A. Roy, N. Memon and A. Ross, "Masterprint: Exploring the vulnerability of partial fingerprint-based authentication systems," *IEEE Trans. Inf. Forensics Secur.* vol. 12, no. 9, pp. 2013–2025, 2017.

[7] C. Bo, L. Zhang, X.-Y. Li, Q. Huang and Y. Wang, "SilentSense: silent user identification via touch and movement behavioral biometrics," In *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2013, pp. 187–190.

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

[9] Y. Cao, F. Li, Q. Zhang, S. Yang and Y. Wang, "Towards Nonintrusive and Secure Mobile Two-Factor Authentication on Wearables," *IEEE Trans. Mob. Comput.*, vol. 22, no. 5, pp. 3046-3061, May 2023.

[10] S. M. Buddhacharya and N. Awale, "CNN-based Continuous Authentication of Smartphones Using Mobile Sensors," *Int. J. Innov. Res. Adv. Eng.*, vol. 9, no. 8, pp. 361-369, Aug. 2022.

[11] R. Kumar, V. Phoha, A. Serwadda, "Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns," In *BTAS*, 1–8, 2016.

[12] G. Peng, G. Zhou, D. T. Nguyen, X. Qi, Q. Yang and S. Wang, "Continuous authentication with touch behavioral biometrics and voice on wearable glasses," *IEEE Trans. Hum. Mach. Syst.*, vol. 47, no. 3, pp. 404–416, 2017.

[13] Y. Li, H. Hu, Z. Zhu and G. Zhou, "SCANet: Sensor-based Continuous Authentication with Two-stream Convolutional Neural Networks," *ACM Trans. Sen. Netw.*, vol. 16, no. 3, Article 29, 27 pages, Jul. 2020.

[14] H. C. Volaka, G. Alptekin, O. E. Basar, M. Isbilen, and O. D. Incel, "Towards Continuous Authentication on Mobile Phones using Deep Learning Models," In *Proc. Comput. Sci.*, vol. 155, pp. 177–184, Jan. 2019.

[15] Z. Shen, S. Li, X. Zhao and J. Zou, "MMAuth: A Continuous Authentication Framework on Smartphones Using Multiple Modalities," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1450-1465, 2022.

[16] Z. Yang, Y. Li, and G. Zhou, "TS-GAN: Time Series GAN for Sensor-based Health Data Augmentation," *ACM Trans. Comput. Healthcare*, vol. 4, no. 2, pp. 1-21, Apr. 2023.

[17] Y. Li, L. Liu, H. Qin, S. Deng, M. A. El-Yacoubi and G. Zhou, "Adaptive Deep Feature Fusion for Continuous Authentication with Data Augmentation," *IEEE Trans. Mob. Comput.*, 2022, doi: 10.1109/TMC.2022.3186614.

[18] Y. Li, B. Zou, S. Deng and G. Zhou, "Using Feature Fusion Strategies in Continuous Authentication on Smartphones," *IEEE Internet Comput.*, vol. 24, no. 2, pp. 49-56, 1 Mar.-Apr. 2020.

[19] Y. Li, P. Tao, S. Deng, G. Zhou, "DeFFusion: CNN-based Continuous Authentication Using Deep Feature Fusion," *ACM Trans. Sens. Netw.*, vol. 18, no. 2, Article 18, 20 pages, Oct. 2021.

[20] H. Chen, F. Li, W. Du, S. Yang, M. Conn, and Y. Wang, "Listen to Your Fingers: User Authentication Based on Geometry Biometrics of Touch Gesture," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 3, Article 75 (September 2020), 23 pages.

[21] Y. Song, Z. Cai and Z. -L. Zhang, "Multi-touch Authentication Using Hand Geometry and Behavioral Information," *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 357-372.

[22] M. Muaaz and R. Mayrhofer, "Smartphone-based gait recognition: From authentication to imitation," *IEEE Trans. Mob. Comput.*, vol. 16, no. 11. pp. 3209–3221, 2017.

[23] M. Zhou et al., "PressPIN: Enabling Secure PIN Authentication on Mobile Devices via Structure-Borne Sounds," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1228-1242, 1 March-April 2023.

[24] Y. Zhang, Y. Zheng, G. Zhang, K. Qian, C. Qian and Z. Yang, "GaitSense: Towards Ubiquitous Gait-Based Human Identification with Wi-Fi," *ACM Trans. Sen. Netw.* vol. 18, no. 1, Article 1, 24 pages, Sep. 2021.

[25] P. Jiang et al., "Securing Liveness Detection for Voice Authentication via Pop Noises," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1702-1718, 1 March-April 2023.

[26] D. Liu et al., "SoundID: Securing Mobile Two-Factor Authentication via Acoustic Signals," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1687-1701, 1 March-April 2023.

[27] L. Lu, J. Yu, Y. Chen, and Y. Wang, "VocalLock: Sensing Vocal Tract for Passphrase-Independent User Authentication Leveraging Acoustic Signals on Smartphones," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 2, Article 51 (June 2020), 24 pages.

[28] C. Song, A. Wang, K. Ren and W. Xu, "EyeVeri: A secure and usable approach for smartphone user authentication," in *Proc. 35th Annu. IEEE Int. Conf. Compu. Commu. (INFOCOM)*, 2016, pp. 1-9.

[29] Z. Sitová et al., "HMOG: New behavioral biometric features for continuous authentication of smartphone users," *IEEE Trans. Inf. Forensics Security*, vol. 11, pp. 877–892, 2016.

[30] Y. Li, H. Hu and G. Zhou, "Using Data Augmentation in Continuous Authentication on Smartphones," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 628-640, Feb. 2019.

[31] J. Gao, X. Song, Q. Wen, P. Wang, L. Sun and H. Xu, "RobustTAD: Robust Time Series Anomaly Detection via Decomposition and Convolutional Neural Networks," in *Proc. ACM SIGKDD Workshop Mining and Learning (KDD-MiLeTS)*, Feb. 2020, pp. 1-9.

[32] S.E. Odongo and D.S. Han, "Feature Representation and Data Augmentation for Human Activity Classification Based on Wearable IMU Sensor Data Using a Deep LSTM Neural Network" *Sensors*, vol. 18, no. 9, article no. 2892, pp. 1-26.

[33] 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. Netw. (MobiCom)*, Oct. 2018, pp. 321–336.

[34] S. Deng, J. Luo, Y. Li, "CNN-Based Continuous Authentication on Smartphones with Auto Augmentation Search," in *Proc. Int. Conf. Info. Comm. Secu. (ICICS)*, Nov. 2021, pp. 169-186.

[35] Y. Li, J. Luo, S. Deng, G. Zhou, "SearchAuth: Neural Architecture Search based Continuous Authentication Using Auto Augmentation Search," *ACM Trans. Sen. Netw.*, 2023, https://doi.org/10.1145/3599727.

[36] T.-H. Cheung and D.-Y. Yeung, "Modals: Modality-agnostic automated data augmentation in the latent space," In *Int. Conf. Learning Representations (ICLR)*. Apr. 2020, pp. 1-18.

[37] N. Yudistira and T. Kurita, "Gated spatio and temporal Convolutional neural network for activity recognition: towards gated multimodal deep learning," *J. Image Video Proc.* vol. 2017, article no. 85, pp. 1-12.

[38] Y. Li, J. Luo, S. Deng and G. Zhou, "CNN-based Continuous Authentication on Smartphones with Conditional Wasserstein Generative Adversarial Network," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5447-5460, 2022.

[39] Y. Ku, L. H. Park, S. Shin and T. Kwon, "Draw It As Shown: Behavioral Pattern Lock for Mobile User Authentication," *IEEE Access*, vol. 7, pp. 69363-69378, 2019.

[40] S. Dieleman, K. W. Willett and J. Dambre, "Rotation-invariant convolutional neural networks for galaxy morphology prediction," *Mon. Notices Royal Astron. Soc.*, vol. 450, no.2, pp. 1441-1459, 2015.

[41] F. Zhan, H. Zhu, S. Lu, "Spatial Fusion GAN for Image Synthesis," In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3653–3662.

[42] Y. Jiang et al., "EnlightenGAN: Deep Light Enhancement Without Paired Supervision," *IEEE Trans. Image Process.*, vol. 30, pp. 2340-2349, 2021.

[43] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5505–5514.

[44] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neur. Info. Process. Sys. (NIPS)*, Dec. 2014, pp. 2672–2680.

[45] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, "Generative Adversarial Networks: An Overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53-65, 2018.

[46] A. Radford, L. Metz, S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. 4th Int. Conf. Learning Representations (ICLR)*, May 2016, pp. 1-16.

[47] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. 29th Int. Conf. Neur. Info. Process. Sys. (NIPS)*, 2016, pp. 469–477.

[48] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang and R. Webb, "Learning from Simulated and Unsupervised Images through Adversarial Training," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 2242-2251.

[49] E. L. Denton, S. Chintala, R. Fergus et al., "Deep generative image models using a laplacian pyramid of adversarial networks," in *Proc. 28th Int. Conf. Neur. Info. Process. Sys. (NIPS)*, 2015, pp. 1486–1494.

[50] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Proc. 29th Int. Conf. Neur. Info. Process. Sys. (NIPS)*, 2016, pp. 217–225.

[51] S. Gurumurthy, R. K. Sarvadevabhatla, and V. B. Radhakrishnan, "DeLiGAN: Generative adversarial networks for diverse and limited data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 166-174.

[52] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-toimage translation using cycle-consistent adversarial networks," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2223-2232.

[53] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Euro. Conf. Comput. Vis. (ECCV)*, 2016, pp. 702–716.

[54] C. Ledig, et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 105-114.

[55] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," in *Proc. 31st Int. Conf. Neur. Info. Porcess. Sys. (NIPS)*, Dec. 2017, pp. 1-11.

[56] Y. Jiang, S. Chang, Z. Wang, "TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up," in *Proc. 35st Int. Conf. Neur. Info. Porcess. Sys. (NIPS)*, pp. 1–14, 2021.

[57] D. P. Kingma, M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[58] A. Van Den Oord, N. Kalchbrenner, L. Espeholt, et al. "Conditional image generation with pixelcnn decoders," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4790–4798, 2016.

[59] D. Gong, L. Liu, V. Le, et al., "Memorizing normality to detect anomaly: Memory-Augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, pp. 1705-1714, 2019.

[60] H. Gao, B. Qiu, R. J. Duran Barroso, W. Hussain, Y. Xu and X. Wang, "TSMAE: A Novel Anomaly Detection Approach for Internet of Things Time Series Data Using Memory-Augmented Autoencoder," *IEEE Trans. Netw. Sci. Eng.*, 2022, doi: 10.1109/TNSE.2022.3163144

[61] Q. Yang et al., "A multimodal data set for evaluating continuous authentication performance in smartphones," in *Proc. 12th ACM Conf. Embedded Netw. Sen. Syst. (SenSys)*, Nov. 2014, pp. 358–359.

[62] X. Chang, C. Peng, G. Xing, T. Hao and G. Zhou, "Isleep: A smartphone system for unobtrusive sleep quality monitoring," *ACM Trans. Sen. Netw.*, vol. 16, no. 3, July 2020.

[63] M. Abuhamad, A. Abusnaina, D. Nyang and D. Mohaisen, "Sensor-based continuous authentication of smartphones' users

using behavioral biometrics: A contemporary survey," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 65–84, 2021.
[64] Y. Luo, B.-L. Lu, "EEG Data Augmentation for Emotion Recognition Using a Conditional Wasserstein GAN," in *2018 40th Ann. Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC)*, Jun. 2018, pp. 2535-2538.
[65] A. Gretton, K. M. Borgwardt, M. Rasch, B. Scholkopf, A. J. Smola, "A Kernel Method for the Two-Sample-Problem," in *Proc. 19th Int. Conf. Neural Info. Porcess. Sys. (NIPS)*, Dec. 2006, pp. 513–520.
[66] L. V. D. Maaten, G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.