



TS-GAN: Time-series GAN for Sensor-based Health Data Augmentation

ZHENYU YANG and YANTAO LI, College of Computer Science, Chongqing University, China
GANG ZHOU, Department of Computer Science, William & Mary, USA

Deep learning has achieved significant success on intelligent medical treatments, such as automatic diagnosis and analysis of medical data. To train an automatic diagnosis system with high accuracy and strong robustness in healthcare, sufficient training data are required when using deep learning-based methods. However, given that the data collected by sensors that are embedded in medical or mobile devices are inadequate, it is challenging to train an effective and efficient classification model with state-of-the-art performance. Inspired by generative adversarial networks (GANs), we propose TS-GAN, a Time-series GAN architecture based on long short-term memory (LSTM) networks for sensor-based health data augmentation, thereby improving the performance of deep learning-based classification models. TS-GAN aims to learn a generative model that creates time-series data with the same space and time dependence as the real data. Specifically, we design an LSTM-based generator for creating realistic data and an LSTM-based discriminator for determining how similar the generated data are to real data. In particular, we design a sequential-squeeze-and-excitation module in the LSTM-based discriminator to better understand space dependence of real data, and apply the gradient penalty originated from Wasserstein GANs in the training process to stabilize the optimization. We conduct comparative experiments to evaluate the performance of TS-GAN with TimeGAN, C-RNN-GAN and Conditional Wasserstein GANs through discriminator loss, maximum mean discrepancy, visualization methods and classification accuracy on health datasets of ECG_200, NonInvasiveFatalECG_Thorax1, and mHealth, respectively. The experimental results show that TS-GAN exceeds other state-of-the-art time-series GANs in almost all the evaluation metrics, and the classifier trained on synthetic datasets generated by TS-GAN achieves the highest classification accuracy of 97.50% on ECG_200, 94.12% on NonInvasiveFatalECG_Thorax1, and 98.12% on mHealth, respectively.

CCS Concepts: • Computing methodologies → Machine learning algorithms; • Human-centered computing → Human computer interaction (HCI);

Additional Key Words and Phrases: GANs, LSTM, data augmentation, SSE, time dependence, space dependence

ACM Reference format:

Zhenyu Yang, Yantao Li, and Gang Zhou. 2023. TS-GAN: Time-series GAN for Sensor-based Health Data Augmentation. *ACM Trans. Comput. Healthcare* 4, 2, Article 12 (April 2023), 21 pages.

<https://doi.org/10.1145/3583593>

12

Zhenyu Yang and Yantao Li equally contributed to this research.

This work was supported by the National Natural Science Foundation of China under Grants No. 62072061 and No. U20A20176.

Authors' addresses: Z. Yang and Y. Li (corresponding author), College of Computer Science, Chongqing University, 174 Shapingba Central Street, Chongqing, China, 400044; emails: {20172725, yantaoli}@cqu.edu.cn; G. Zhou, Department of Computer Science, William & Mary, 251 Jamestown Road, Williamsburg, VA, USA, 23185; email: gzhou@cs.wm.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2637-8051/2023/04-ART12 \$15.00

<https://doi.org/10.1145/3583593>

1 INTRODUCTION

The rapid development of AI has made it possible to use aggregated healthcare data to train powerful models that can automatically diagnose diseases as well as enable an increasingly personalized approach to medicine with maximum effectiveness in a timely manner [18, 42]. That is to say, a doctor only needs to input the patient's health data, such as **electrocardiogram (ECG)**, **electroencephalography (EEG)**, or body motion into a model, and the output of the model can diagnose the disease [28]. For example, **cardiovascular disease (CVD)** systems leverage **deep neural networks (DNNs)** to detect arrhythmia of captured ECG signal leading to cost reduction of continuous heart monitoring and thus improving the prediction quality [14], which requires large amounts of ECG data for pre-training. Based on deep learning techniques, the authors of Reference [15] propose an intelligent Parkinson detection system that is a DNN classifier trained by utilizing gait information extracted from a large amount of motion data of subjects. On the one hand, different from images and voices, it is difficult to search or collect a large amount of medical data in a short time, which is mainly caused by privacy and legal issues when sharing and reporting patient health information [40]. On the other hand, an effective and robust diagnosis model needs to be trained on large health datasets collected by various mobile sensors [35].

To address the shortage of training data, geometric-based data augmentation methods and reinforcement learning-based data augmentation approaches have been investigated. The geometric augmentation methods are based on original images attached to some simple transformation strategies or combinations, such as translation, twisting, shearing, scaling, and rotation [13]. The reinforcement learning-based data augmentation approaches automatically search the best transformation strategies [9, 10, 26, 41]. Indeed, data augmentation methods can also be used to generate medical images, such as CT, MR, and SPECT. However, the effect of applying augmentation methods for images directly to time-series health data is limited, and thus it is a challenge to find an effective data augmentation approach for time-series health data. In recent years, **generative adversarial networks (GANs)** have been proved to be effective and promising strategies that create data similar to the real data [1, 20]. Typical GANs usually consist of two parts, i.e., a generator and a discriminator. The generator creates realistic data to confuse the discriminator, while the discriminator attempts to distinguish whether the data originates from the real or generated distribution. The training process adopts an adversarial strategy: the parameters of the discriminator are fixed when optimizing a generator, and then the parameters of the generator are fixed when optimizing the discriminator, and finally the generator, which can create fake data with almost the same distribution as real data is obtained [1, 20]. The GAN architecture is derived from a two-player game, and the players are the generator G and the discriminator D , each of which incurs a cost: $J^{(G)}(\theta^{(G)}\theta^{(D)})$ for G and $J^{(D)}(\theta^{(G)}\theta^{(D)})$ for D . During training, each player attempts to minimize its own cost [21], which corresponds to a zero-sum game process. After many rounds of this two-player game, the ideal result of both players is to converge to a Nash equilibrium [16]. In the field of computer vision, GANs have been applied to image data augmentation to produce a large number of images that are not easy to obtain, and the performance of image classifiers has been significantly improved as a result of using the data generated as additional training data [43, 50].

Different from image augmentation, we focus on strengthening the discriminator to design GANs for time-series data augmentation. In this article, we propose a **Time-series GAN (TS-GAN)** architecture to create more sensor-based health data for deep learning-based classification models by leveraging sensors embedded in medical or mobile devices to capture vital features. TS-GAN mainly contains two fundamental parts, i.e., a generator and a discriminator, both of which are based on **long short-term memory (LSTM)** networks. LSTM networks are capable of memorizing and capturing time dependence, which implies the relationship between the current moment and the past moment of each dimension [49], allowing TS-GAN to learn a generative model that creates time-series data with the same time dependence as the real data. As for capturing space dependence, which implies the relationship between dimensions at each time step, an module with attention mechanism similar to a **squeeze and excitation (SE)** module is applied [25]. Specifically, inspired by the SE module in image classification, we allow the discriminator of TS-GAN to contain a **sequential-squeeze-and-excitation (SSE)** module

with a 1D convolution to capture human vital features at every time step, namely, the space dependence. Assuming a time-series segmentation $x \subseteq \mathbb{R}^{L \times M}$ where L is the length and M is the dimension, 1D convolution enlarges it to $f \subseteq \mathbb{R}^{L \times D}$ where D is larger than M , and then the SSE module takes f as an input and outputs an attention vector $a \subseteq \mathbb{R}^D$, which is multiplied by f . Thus, TS-GAN decouples time dependence and space dependence, which is achieved by LSTM networks and a strengthened discriminator containing SSE module. In addition, classical GANs have been reported to suffer from mode collapse where data generated from GANs are not as diverse as that sampled from the real distribution [24, 30], and thus our TS-GAN adopts the gradient penalty originated from WGAN-GP [23] to avoid mode collapse. We evaluate TS-GAN against three state-of-the-art methods, i.e., TimeGAN, C-RNN-GAN, and CWGAN in terms of the discriminator loss, MMD, **principal component analysis (PCA)**, **t-distributed stochastic neighbor embedding (t-SNE)**, sequence diagrams, and classification accuracy, on three health datasets, i.e., ECG_200 [11], NonInvasiveFatalECG_Thorax1 [11], and mHealth [5], respectively. Specifically, based on the three health datasets, we qualitatively compare the generated data quality of TS-GAN with the other three GANs by discriminator loss, MMD, and visualization in terms of PCA, t-SNE and sequence diagrams after training, and we quantitatively compare the effectiveness of the generated data of TS-GAN with the other GANs by using augmented training sets to train a series of binary LSTM classifiers (two layers) in terms of accuracy, recall, precision, and F_1 score, respectively. The comparison results indicate that the generated data by TS-GAN show the best performance among other GANs.

The main contributions of this work are summarized as follows:

- We propose TS-GAN, a Time-series GAN architecture based on LSTM networks for sensor-based health data augmentation to improve the performance of deep learning-based classification models.
- We design an LSTM-based generator for creating realistic data, and an LSTM-based discriminator, which contains an SSE module with 1D convolution for determining the similarity between generated time-series and the real data. To the best of our knowledge, the use of SSE modules in GANs for generating health data has never been studied.
- We evaluate the performance of TS-GAN in terms of the discriminator loss, MMD, PCA, t-SNE, sequence diagrams, and classification accuracy on ECG_200, NonInvasiveFatalECG_Thorax1, and mHealth datasets, respectively, and the experimental results show that TS-GAN can improve the performance of deep learning-based classification models than other GANs such as TimeGAN, C-RNN-GAN, and CWGAN.

The remainder of this article is organized as follows: we summarize the related works in Section 2, and give the preliminary knowledge in Section 3. We detail the proposed TS-GAN in Section 4, and evaluate the performance of TS-GAN in Section 5. Section 6 concludes as well as discusses this work.

2 RELATED WORK

Thanks to the success of **artificial neural networks (ANNs)**, time-series including gait data [7, 29], EEG signal [34, 36], ECG data [46, 48], and other biosignals are increasingly used for efficient medical analysis and automatic diagnosis. However, a robust ANN needs to be trained on large datasets, and there are limited sources of medical time series. For example, the 2018 **University of California Riverside (UCR) Time-series Archive** [11] is one of the largest repositories of time-series datasets, and in a total of 128 datasets, only 12 datasets have more than 1,000 training samples, which demonstrates there is a need for time series.

The basic idea of solving the problem of lack of time series is to generate a synthetic dataset, i.e., data augmentation, which introduces possibilities to obtain diverse data while maintaining correct labels. So far, most data augmentation methods for time series are based on transformations of the original data. For example, adding random noise [17, 32], slicing or cropping, scaling [32, 44], random warping in the time dimension [17, 44], and frequency warping [27]. Transforming method is a direct transformation technology of time series, which is widely used because of its simplicity and efficiency. However, due to the high requirements for the accuracy of medical data, the healthcare field prefers to use some model-based augmentation methods [2, 3].

There are some recent studies establishing search-based models. For example, the authors of Reference [8] propose a MODALS model to search optimal data transformation combinations in the latent space by using evolutionary strategies, which allows it to work on multimodal data, and experiments show the outstanding performance of the classifier, which uses augmented time series generated by MODALS. In Reference [12], the authors adopt seven different time domain augmentation, i.e., rotation, jittering, scaling, permutation, magnitude-warping, time-warping, and cropping for sensor-based time series to form the searching space. After training by policy gradient and Monte Carlo sampling, the optimal augmentation combinations with a high probability to achieve great classification accuracy can be determined adaptively for each input. The authors in Reference [19] propose two sample-adaptive automatic weighting frameworks for data augmentation, where the first learns weights to represent the contribution of the augmented samples to the loss, and the second method chooses a subset of transformations based on the ranking of the predicted training loss. Similarly, one study uses expert networks and gate networks to search for optimal weights for several meta transformation operations to achieve data augmentation in time-series classification tasks [39].

Unlike applying transformation policies to time series by various algorithms, GAN, a type of deep generative model, directly generates new data to enlarge the training set. The authors in Reference [47] decouple time and space dependence by designing TimeGAN with a generator, a discriminator, and other two autoencoder networks. The inputs to the framework are two elements, a static characteristic and a time characteristic. The generator accepts tuples of static and temporal random eigenvectors extracted from known distributions. The real code and synthetic latent code are used to calculate the supervised loss element of the network. The discriminator receives tuples of real and synthetic latent code and classifies them as real or synthetic. The autoencoder is to preliminarily reduce the time-series space into a latent space where the generator creates the composite latent code similar to the real data, and at every time step it generates the same latent code as the real time-series data, which ensures the time dependence of generated data. It can be seen that all the operations of TimeGAN are performed in a low-dimension latent space, and simple autoencoders cannot easily capture features of multivariable time series, which may result in poor generated data. In Reference [38], the authors design a C-RNN-GAN where the generator and discriminator are based on bi-LSTM to ensure that the generated data are time dependent and similar to real data in each time step (space dependence); the discriminator loss, however, is hard to be reduced to a small value due to the complex combined tasks. The authors of Reference [37] use **conditional Wasserstein GANs (CWGAN)** to generate EEG data, and they use conditional label with time-series data as the input for training a generator, which can produce EEG data under a label of a given class. During training, they adopt the gradient penalty used in WGAN to stabilize the training process, but they treat the time-series data as images ignoring the time dependence among time steps, which cannot create correct multivariate time series with the coherent time relationship. In Reference [33], the authors explore a Transformer-based GAN composed of a Transformer-based generator and a CNN-based discriminator to generate additional training data for CNN training.

Comparing with aforementioned models, we propose a concise TS-GAN model to generate synthetic health datasets including ECG_200, NonInvasiveFatalECG_Thorax1, and mHealth. We utilize LSTM to construct layers of the generator and discriminator for capturing time dependence, and inspired by SE module in CNN, we strengthen the discriminator by designing an SSE module with 1D convolution layer for capturing space dependence of the data, which decouples time and space dependence of data by LSTM and SSE. Besides, TS-GAN adopts the gradient penalty of WGAN-GP to avoid mode collapse.

3 PRELIMINARY

3.1 Wasserstein GANs

Traditional GANs derive from a two-player game, and contain two main networks of a generator and a discriminator, which are trained in an adversarial way. Given real-data distribution X_r and generated one X_g , the

adversarial loss of training can be described as Equation (1):

$$\min_G \max_D L_{adv} = \mathbb{E}_{x_r \sim X_r} [\log(D(x_r))] + \mathbb{E}_{x_g \sim X_g} [\log[1 - D(x_g)]], \quad (1)$$

where parameters of G are hidden in $x_g = G(x_z)$, and x_z is a noise sampled from a specific distribution, such as $N[0,1]$ and $U[0,1]$. The adversarial loss function minimizes the **Jensen-Shannon (JS)** divergence between X_r and X_g , and when there are no or few intersecting parts between X_r and X_g , JS divergence is a constant that causes the **back propagation (BP)** gradient to become 0 (gradient vanishing). Whereas, Earth-Mover distance is a function of parameters of the generator even though there is no intersecting part between X_r and X_g [4], so it can be used to solve the issue of gradient vanishing. Thus, in the research of WGANs [4], Earth-Mover distance is proposed to replace JS divergence as Equation (2):

$$W(X_r, X_g) = \inf_{\gamma \sim \prod(X_r, X_g)} \mathbb{E}_{(x_r, x_g) \sim \gamma} [| | | x_r - x_g | | |], \quad (2)$$

where $\prod(X_r, X_g)$ represents all possible joint probability distributions of X_r and X_g . In WGANs, to achieve Earth-Mover distance expressed by Equation (2), the discriminator D is constrained by Equation (3):

$$W(X_r, X_g) = \frac{1}{K} \sup_{||D||_L \leq K} \mathbb{E}_{x_r \sim X_r} [D(x_r)] - \mathbb{E}_{x_g \sim X_g} [D(x_g)], \quad (3)$$

where K is a constant. Further, researchers of WGAN-GP achieve the Lipschitz constraint by simply adding gradient penalty to loss function that restricts the updating value of parameters of D in $[-1, 1]$ as Equation (4):

$$\min_G \max_D L_{adv} = \mathbb{E}_{x_r \sim X_r} [D(x_r)] + \mathbb{E}_{x_g \sim X_g} [-D(x_g)] - \lambda \mathbb{E}_{\hat{x} \sim \hat{X}} [(| | | \nabla_{\hat{x}} D(\hat{x}) | | |_2 - 1)^2], \quad (4)$$

where \hat{x} represents a linear interpolation between real data x_r and generated data x_g .

3.2 LSTM

LSTM, an improved **recurrent neural network (RNN)**, is often used to regress or classify sequential data and meanwhile eliminates the risk of gradient vanishing and exploding in traditional RNN when dealing with long sequential data [22]. The reason why LSTM has a much better memory than RNN is that it contains three gates: a forgetting gate, an input gate and an output gate. As shown in Figure 1, given a sequential sample x_s , LSTM produces an output h_t once every time step x_{s_t} is dealt.

In the forgetting gate, f_t is produced to decide how to forget information in the last output h_t in Equation (5):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_{s_t}] + b_f). \quad (5)$$

In the input gate, i_t is produced to decide how many new information can be added to the cell and \tilde{C}_t is produced to participate in the update of the cell C_t as in Equation (6):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_{s_t}] + b_i), \quad (6a)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_{s_t}] + b_c), \quad (6b)$$

$$C_t = f_t \times C_{t-1} + \tilde{C}_t \times i_t. \quad (6c)$$

In the output gate, o_t is produced to filter what information cannot be output and the output h_t at time step x_{s_t} is calculated in Equation (7):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_{s_t}] + b_o), \quad (7a)$$

$$h_t = o_t \times \tanh(C_t). \quad (7b)$$

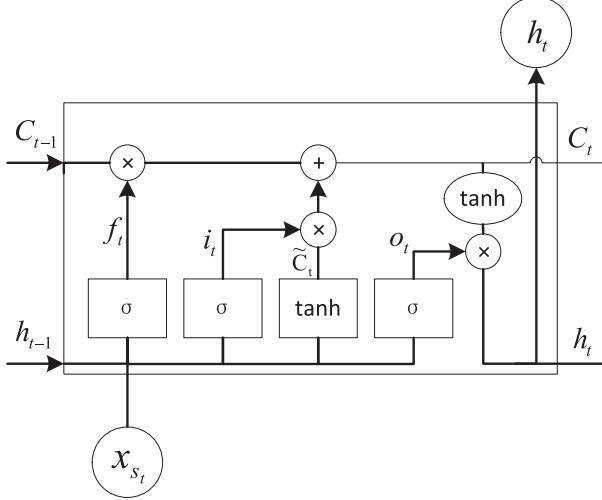


Fig. 1. The data flow of LSTM at one time step.

For Equations (5)–(7), $W_f, W_i, W_o, b_f, b_i, b_c, b_o$ are all trainable parameter matrices. Due to the great memory of LSTM, if the length of x_s is T , then h_T at the T time step is chosen to be the final classification result y_s .

4 APPROACH

In this section, we detail our approach TS-GAN for health data augmentation. First, we introduce the overview of our TS-GAN, then show how to segment the dataset, and finally give details of main components of the model and the process of achieving data augmentation.

4.1 Overview

We propose TS-GAN based on LSTM for time-series data augmentation leveraging the sensors embedded in medical or mobile devices that collect raw data of human vitals. TS-GAN aims to learn a generative model that creates time-series data with the same space and time dependence as the real data. Specifically, we design an LSTM-based generator for creating realistic data and an LSTM-based discriminator for determining how similar the generated data is to the real data, as illustrated in Figure 2.

As demonstrated in Figure 2 where we use dotted lines with arrows to represent the back propagation and use solid lines with arrows to represent the data flow, TS-GAN for data augmentation can be divided into two processes, i.e., the training process and augmentation process. In the training process, we train the generator G and the discriminator D at the same time by adopting the adversarial training policy, where the detailed architectures of G and D are shown in Figures 3 and 4, respectively. Specifically, as illustrated in Figure 3, the linear layer in G samples random **one-dimensional (1D)** noise z from a latent space with a distribution of p_z . Then, it maps and reshapes z to a sequential type intermediate vector x^s , which is turned into time-series data \hat{x} by $LSTM_G$ (an LSTM network in the generator). When real and generated data enter into D , they first undergo pre-processing where outliers of the data are filtered and the filtered data is then normalized, as demonstrated in Figure 4. After pre-processing, features of the time-series data are captured and enlarged by the 1D convolution and SSE module. The former enlarges data features to get a feature map whose length is maintained without pooling operation, which shortens time-series data, and the latter learns a vector of weights representing the significance of each feature of this feature map, which is multiplied by the weights vector later. Finally, $LSTM_D$, a binary LSTM-based classifier improved by the WGAN-GP, gives a scalar to label the treated feature map fake

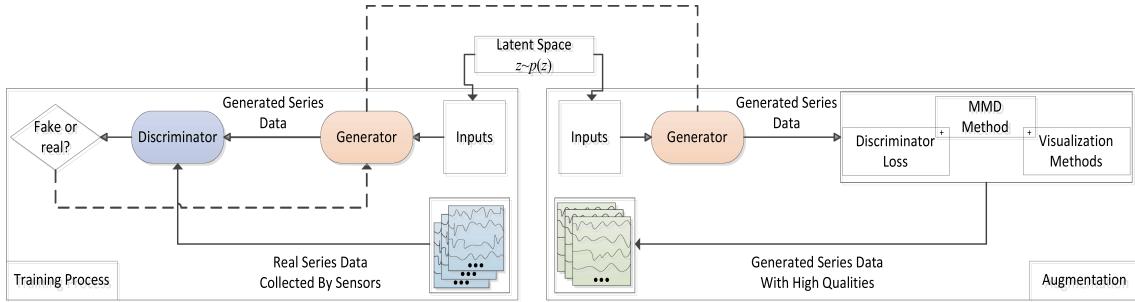


Fig. 2. TS-GAN for data augmentation. TS-GAN process consists of the training process and augmentation process. In the training process, the generator G and the discriminator D are trained at the same time by adopting the adversarial training policy. In augmentation process, the well-trained generator G is used for data augmentation, which is evaluated by three methods, i.e., discriminator loss, MMD, and visualization methods (PCA, t-SNE, and sequence diagrams) to select generated data with high quality to be added to enlarge the training set.

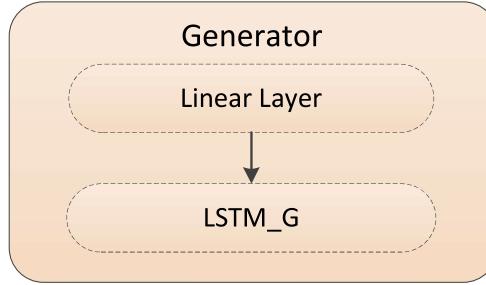


Fig. 3. Data flow of the generator G . G is composed of a linear-layer module and LSTM network ($LSTM_G$).

or real. Likewise, the adversarial training is performed between G and D as far as the adversarial loss converges. After adversarial training, the generator G of TS-GAN finally learns: (i) the distribution of the data within each time step corresponding to tiny activities of the subject, i.e., the space dependence at every time step; and (ii) the time dependence among all time steps. Specifically, the SSE module with 1D convolution in the discriminator enlarges and interconnects all the features at each time step, ensuring space dependence and allowing $LSTM_D$ to focus on capturing time dependence. The strengthened discriminator that decouples time and space dependence is capable of forcing the generator to create ideal time series.

In the augmentation process, the well-trained generator G is used for data augmentation. Particularly, we propose three methods, i.e., discriminator loss, MMD, and visualization methods (PCA, t-SNE, and sequence diagrams) to select generated data with high quality and only these data can be added to enlarge the training set.

4.2 Data Segmentation

The longest length of time-series data that can be processed by G and D is fixed; therefore, the length of the training data also should be fixed by segmenting the dataset. We use a sliding window with a size of T and a sliding step of s to divide the training data into a certain number of windows. Given the dataset $X \subseteq \mathbb{R}^{L \times M}$, where L is the total time steps and M refers to the dimension numbers, we can get n data windows by Equation (8):

$$n = \frac{L - T}{s} + 1. \quad (8)$$

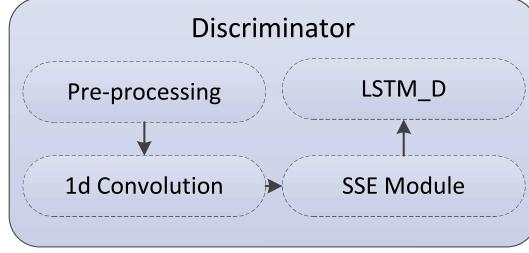


Fig. 4. Data flow of the discriminator D. D is composed of a pre-processing module, a 1D-convolution layer, an SSE module and LSTM networks (LSTM_D).

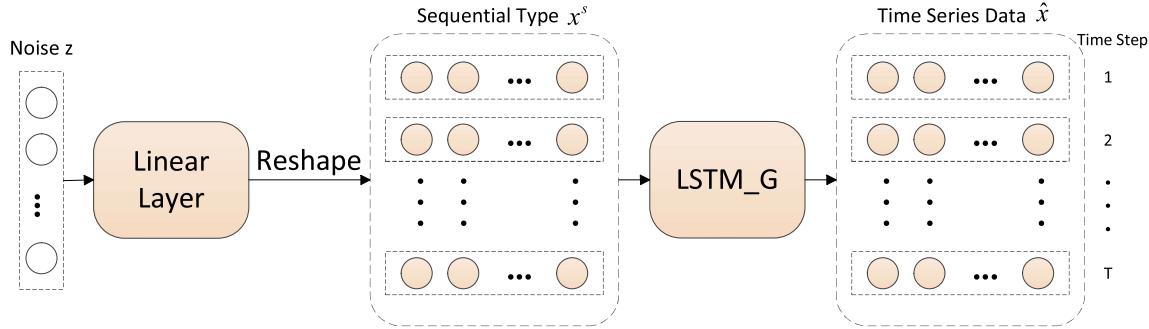


Fig. 5. Generator G in TS-GAN. The linear layer maps and reshapes z into sequential data x^s , which is then processed by $LSTM_G$ to get generated time-series data \hat{x} .

Based on these data windows, we obtain the training samples $X = \{x_1, x_2, \dots, x_n\} \subseteq \mathbb{R}^{T \times M}$. For the sake of a better understanding to the following parts, we take mHealth dataset (detailed in Section 5) as an example to describe the data preprocessing and data flow in TS-GAN in the rest of this section. mHealth dataset contains total time steps of 98,303 and channels of 23 for each subject (10 in total). For the window size of mHealth dataset, we have conducted experiments to show that the classification accuracy of a trained classifier almost reached the peak (78.24%) at the segmentation length of 200 and decreased when the length increased. Thus, we set the window size of mHealth dataset as 200, and set the sliding step as 1 for obtaining more training samples. By Equation (8), we get segmented training set $X = \{x_1, x_2, \dots, x_{98104}\} \subseteq \mathbb{R}^{200 \times 23}$ for each subject.

4.3 Generator

The time-series generation process in generator D is shown in Figure 5. Noise $z \subseteq \mathbb{R}^{1 \times 30}$ is a random vector from the latent space, and the linear layer maps z into $\mathbb{R}^{1 \times (200 \times 23)}$, which is then reshaped to sequential data $x^s \subseteq \mathbb{R}^{200 \times 23}$. Next, $LSTM_G$ with a capability of great memory executes regression, where the realistic sequential data x^s are mapped into the generated time-series data $\hat{x} \subseteq \mathbb{R}^{200 \times 23}$.

In fact, unlike generating images, features and lengths of each data window are very limited, and the time series can be generated without much input noise. Different from the study in Reference [38], the linear layer in our generator avoids sampling noise with many features at every time step. Specifically, we use a 1D noise z with 30 features embedded in higher space by the linear layer, and the final generated data \hat{x} are derived from low latent space, reducing the regression difficulty of $LSTM_G$.

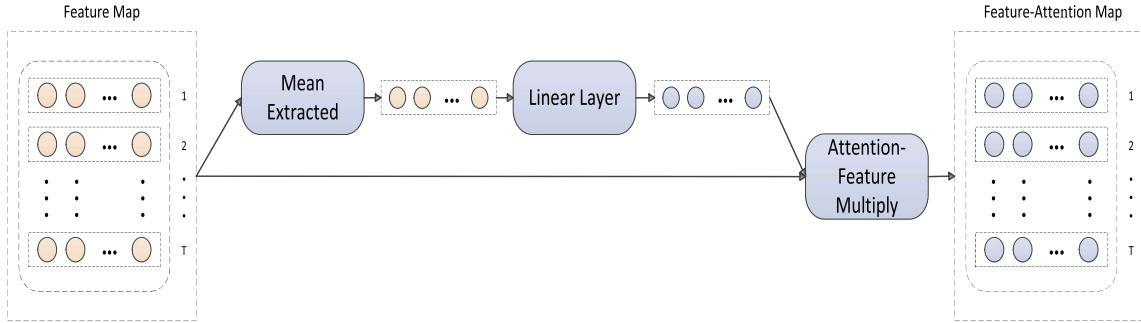


Fig. 6. SSE module in discriminator D . The pre-processed time-series data are dimension-enlarged by 1D convolution to get a feature map m , and then an SSE module learns an attention vector a , which undergoes attention-feature multiplication to m to a feature-attention map f .

4.4 Discriminator

The real time-series data or the generated data are distinguished by the discriminator D . After filtered and normalized through the pre-processing, features of the time-series data are dimension-enlarged by a 1D convolution layer with c kernels ($c = 46$) to get a feature map m , which is also a time sequence with 46 features and length of 200. Then, an SSE module learns an attention vector based on the feature map, and the data flow in the SSE module is shown in Figure 6. Given a feature map $m \subseteq \mathbb{R}^{T \times c}$, the mean value of each feature \bar{m}^j is extracted by Equation (9):

$$\bar{m}^j = \frac{1}{T} \sum_{i=1}^T m_i^j, \quad j = 1, 2, \dots, c. \quad (9)$$

Next, the linear layer maps the mean value vector $\bar{m} = \{\bar{m}^1, \bar{m}^2, \dots, \bar{m}^c\}$ into an attention vector $a = \{a^1, a^2, \dots, a^c\}$ which undergoes attention-feature multiplication to the input feature map m to get a feature-attention map f in Equation (10):

$$f_i^j = a^j \cdot m_i^j \quad (i = 1, 2, \dots, T; j = 1, 2, \dots, c). \quad (10)$$

In the 1D convolution layer, 46 kernels enlarge the dimension number of the time-series data from 23 to 46 and interconnect all the features at each time step to ensure space dependence. However, this operation may introduce some dimensions that make no difference to the final classification, which may cause the training loss hard to converge. Therefore, we design an SSE module to learn an attention vector to reflect the significance of each dimension in the feature map, thereby obtaining the feature-attention map to better describe space dependence of the original time-series data, as illustrated in Figure 6. For the feature-attention map f , each feature of f_i contains all the features of original time series at time step i , and thus $LSTM_D$ can focus on time dependence among all time steps in f and give a scalar to label it fake or not. The last layer of $LSTM_D$ is improved according to WGAN-GP, and Equation (4) is used as the loss function.

4.5 Augmentation

We train TS-GAN on the segmented dataset $X = \{x_1, x_2, \dots, x_{98104}\} \subseteq \mathbb{R}^{200 \times 23}$, and after adversarial training, the generator is capable of producing many time-series data by sampling random noise $z \subseteq \mathbb{R}^{1 \times 30}$ with the distribution of $U[0, 1]$. Finally, we can choose k generated data windows with high quality evaluated by the five metrics, i.e., discriminator loss, MMD, PCA, t-SNE, and sequence diagrams to form a generated dataset $X^g = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_k\}$. For downstream classification, we choose k to be roughly equal to the size of the original training set, which has been shown to provide optimal classification performance improvement in References [6, 37].

5 EXPERIMENTS

To evaluate the quality of health data generated by TS-GAN, we conduct comparative experiments on two commonly-used univariate datasets with small volume, i.e., ECG_200 and NonInvasiveFatalECG_Thorax1 [11], and a multivariate dataset, i.e., mHealth [5], augmented by TS-GAN with another three state-of-the-art GANs, i.e., TimeGAN [47], C-RNN-GAN [38], and CWGAN [37], respectively. In this section, we first qualitatively evaluate these synthetic data by discriminator loss, MMD and visualization (i.e., PCA, t-SNE, and sequence diagrams). Then, we quantitatively evaluate TS-GAN and other GANs by using augmented training sets from different GANs as inputs to train LSTM classifiers for comparing the classification performance, and the division of training set and testing set for each dataset is introduced in detail in Section 5.1. Finally, we conduct experiments to study the effect of the proportion of the generated data added to each original training set on the classifier accuracy.

5.1 Dataset

In our experiments, we utilize three public sensor-based health datasets to evaluate TS-GAN: ECG_200, NonInvasiveFatalECG_Thorax1 [11], and mHealth [5]. In the process of optimizing GANs, all the data of each class in the dataset are used for training GANs that generate data of the corresponding class. In the process of training and testing LSTM classifiers, we adopt fivefold cross-validation, where the original dataset is split into training set Tr (fourfold) and testing set Te (onefold), and then use the trained GANs to obtain generated dataset X^g with the same volume as Tr , and the augmented training set is $Ta = Tr \cap X^g$, where we balance the proportion of positive and negative samples. During the testing, we evaluate the performance of LSTM classifiers on Te .

ECG_200 [11]. The ECG dataset contains 200 ECG time series, where each time series represents a heartbeat and is composed of 96 cardiac electrical activity measurements. Among the 200 time series of ECG_200, 133 are marked as “normal” (positive), while the remaining 67 are marked as “abnormal” (negative).

NonInvasiveFatalECG_Thorax1 [11]. This dataset contains 88 ECG time series, where each time series indicates a fetal electrocardiogram and is composed of 750 left chest cardiac electrical signal measurements. In the 88 time series, 52 data are marked as “abnormal” (negative) and 36 data are marked as “normal” (positive).

mHealth [5]. The mobile Health dataset comprises body motion and vital signs recordings for 10 volunteers of diverse profile while performing several physical activities. Sensors positioned on the subject’s chest provide two-lead ECG measurements, which can be potentially used for basic heart monitoring, checking for various arrhythmias or looking at the effects of exercise on the ECG. mHealth contains 98,303 health-related measurements for each volunteer and each measurement is composed of 23 dimensions. For the experiments, we segment the mHealth data by a sliding window with the length of 200 and a sliding step of 1.

5.2 Quality of Augmented Data

To evaluate the quality of data augmented by TS-GAN, we conduct experiments of discriminator loss, **maximum mean discrepancy (MMD)**, and visualization in terms of PCA, t-SNE, and sequence diagrams on the three health datasets where we use “subject_0” of mHealth in this subsection, since mHealth has 10 classes.

5.2.1 Discriminator Loss. Discriminator loss indicates the Earth-Mover distance between the real data and the generated data when the network converges [37]. That is, the TS-GAN generates high-quality data if the discriminator loss is approximately 0. To evaluate the quality of data generated by TS-GAN, we conduct comparison of the discriminator loss during training among augmented data by TS-GAN, C-RNN-GAN, TimeGAN, and CWGAN, on the three datasets of ECG_200, NonInvasiveFatalECG_Thorax1, and mHealth, respectively, as illustrated in Figure 7. As shown in Figure 7, overall, the discriminator losses of all the GANs converge on the three datasets. Specifically, for C-RNN-GAN, the loss converges quickly right after 34,000 iterations on mHealth dataset, but it has obvious jittering on NonInvasiveFatalECG_Thorax1 dataset, which affects the stability of model training. For TimeGAN, the loss converges smoothly only on NonInvasiveFatalECG_Thorax1 dataset, and the loss does not converge to a small value on the other two datasets (within 45,000 iterations). For CWGAN, the loss

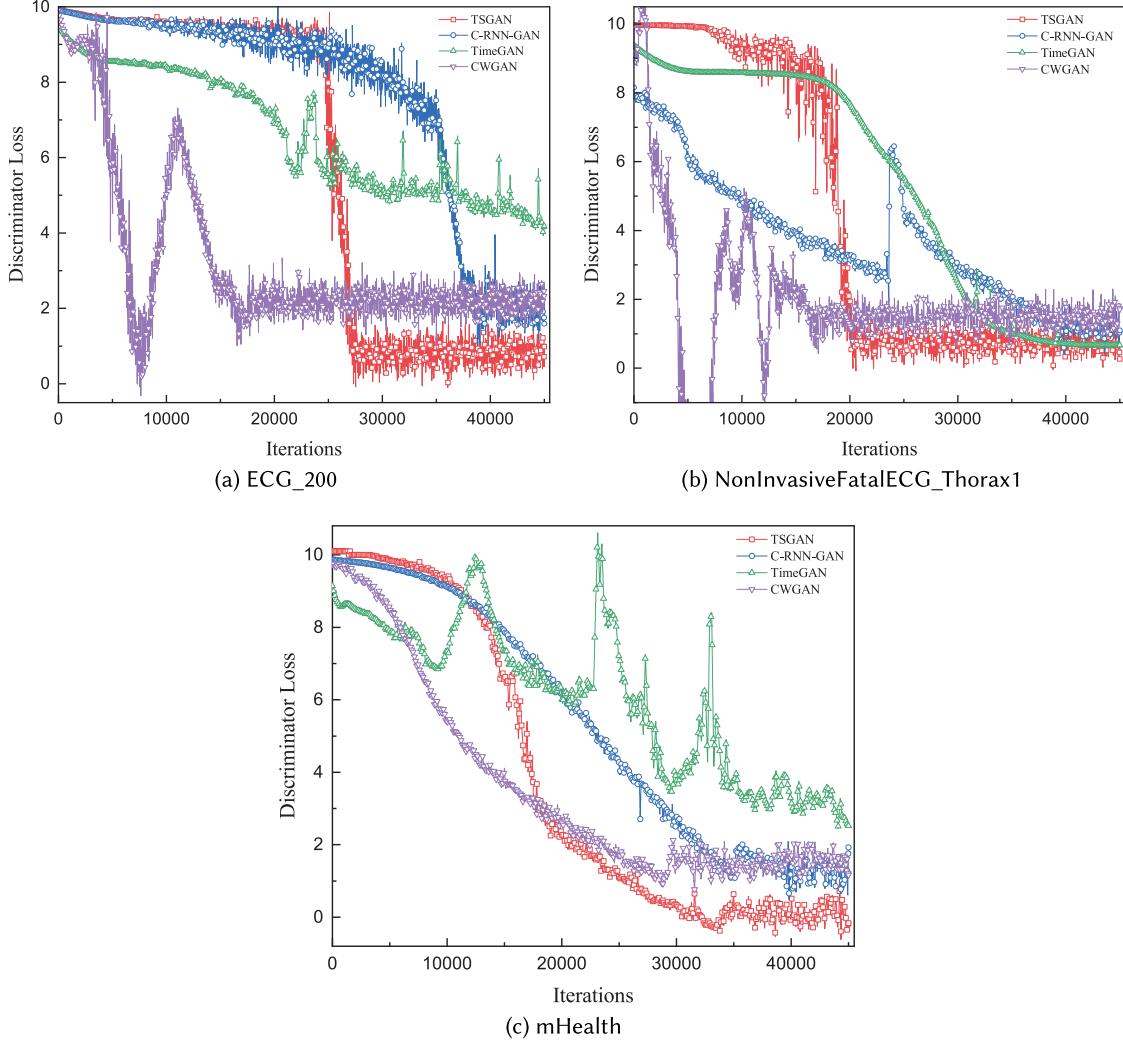


Fig. 7. Discrminator loss during training.

decreases steadily only on mHealth dataset, but with smaller amount of data, the loss fluctuates greatly and the decline is very unstable on the other two ECG datasets. For TS-GAN, on all datasets, the loss converges quickly, where the convergence value is small (around 0) and the process is stable.

5.2.2 MMD. MMD can describe the distance between distributions of the generated data and real data [31]. That is, the data generated by TS-GAN are high-quality if the distance is close to 0. To evaluate of data quality generated by TS-GAN, we also conduct comparison of the MMD during training among augmented data by TS-GAN, C-RNN-GAN, TimeGAN, and CWGAN, on the three datasets, respectively, as illustrated in Figure 8. As shown in Figure 8, the downward trends of MMD for all the GANs are basically the same as the convergence trends of the discriminator loss. Concretely, for the MMD of C-RNN-GAN, its decline is relatively slow and the decline process is jittery on NonInvasiveFatalECG_Thorax1 dataset. TimeGAN only performs well on

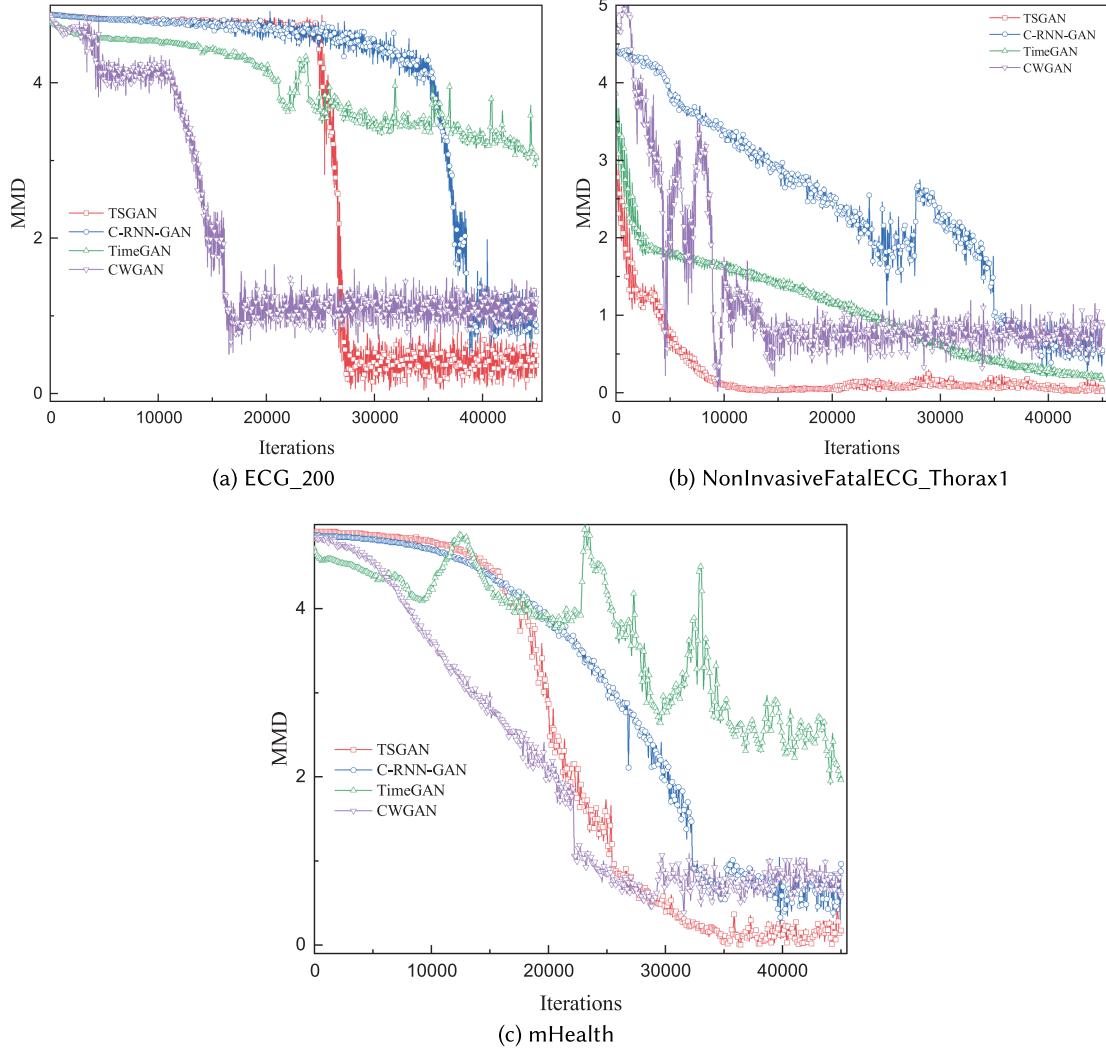


Fig. 8. MMD during training.

NonInvasiveFatalECG_Thorax1 dataset, and CWGAN fluctuates greatly on ECG datasets. Nevertheless, the MMD of TS-GAN drops to near 0 on all datasets and the decline process is fast and stable.

5.2.3 Visualization. To visualize the augmented data, we calculate PCA and t-SNE on mHealth dataset, and sequence diagrams on ECG_200 and NonInvasiveFatalECG_Thorax1 datasets, respectively, since the two ECG datasets have only one dimension.

PCA and t-SNE. We use PCA and t-SNE for the visualization of the augmented data by the state-of-the-art GANs on mHealth dataset. PCA reduces the dimension of the generated dataset X^g and real dataset X (flattened in the time dimension) to two. To better study the distribution of the real data, we make a density map based on PCA by the following two steps: (1) partitioning projection plane, and (2) calculating area density, to visualize the probability density of the data in a two-dimension plane.

- Partition projection plane. Assuming all the real data $X = \{x_1, x_2, \dots, x_{98104}\} \subseteq \mathbb{R}^{200 \times 23}$ is mapped into a two-dimension rectangle domain with the length of a and the width of b by PCA to get $V = \{v_1, v_2, \dots, v_{98104}\} \subseteq \mathbb{R}^{1 \times 2}$, we divide the domain into n squares by Equation (11):

$$n = \frac{a \cdot b}{\epsilon^2}, \quad (11)$$

where ϵ is the side length of squares. Thus, we can get a square set $S = \{s_1, s_2, \dots, s_n\}$.

- Calculate area density. For every v_i in V , we define an indicator function in Equation (12):

$$I(v_i, s_j) = \begin{cases} 1, & v_i \in s_j, \\ 0, & v_i \notin s_j, \end{cases} \quad (12)$$

We can estimate the probability density that v_i from V is contained by s_j in Equation (13):

$$f(s_j) = \frac{1}{m} \cdot \sum_{i=1}^m I(v_i, s_j), \quad (13)$$

where m is the element number of V .

By Equations (11)–(13), we can better visualize the distribution of X and observe the distribution of generated data X^g relative to X , as shown in Figure 9, where the distribution of real data is shown as density map and red points refer to generated data. According to these equations, only the red points falling into or near the yellow areas have the same distribution as the real data. In Figure 9, although some red points generated by C-RNN-GAN fall into all the yellow areas, there are many red points are far from yellow areas, which is caused by the large variance of red points. Almost all the red points generated by CWGAN fall into or near the yellow areas at the top, but there is no sample near other yellow areas, which is caused by the large deviation of red points. All the red points generated by TS-GAN and TimeGAN fall into or near the yellow areas, but the area enclosed by dotted lines of TimeGAN is much smaller than TS-GAN, and no red point from TimeGAN falls into the yellow area at the top left, which means light mode collapse occurs in TimeGAN so that the data generated by TimeGAN is not as diverse as TS-GAN.

t-SNE maps the high-dimensional generated data non-linearly into two dimensions, where the corresponding data can be visualized [45]. We visualize the generated data by state-of-the-art GANs on mHealth dataset, as demonstrated in Figure 10, where blue points refer to generated data and green ones refer to the original. As illustrated in Figure 10, overall, the data generated by four GANs roughly obey the distribution of the original data; the data from TimeGAN, however, are so concentrated that many features are missing while the other three GANs can generate data covering more features of real data.

Sequence Diagrams. We use sequence diagrams for the visualization of ECG_200 dataset and NonInvasiveFatalECG_Thorax1 dataset. We show the real data with generated data from state-of-the-art GANs in time domain as shown in Figures 11 and 12. For ECG_200 dataset, all the GANs can generate time-series data that fit the trend of real data on the whole; however, only TS-GAN generates almost all the local features of real data and smoothly fits it. TimeGAN omits many small values and the generated values fluctuate greatly. C-RNN-GAN also omits some small values by generating many extreme outliers. CWGAN tends to generate noisy and unstable time-series data. On NonInvasiveFatalECG_Thorax1 dataset, generation performance of all the GANs is almost the same as that on ECG_200 dataset. C-RNN-GAN performs better than the rest two GANs on this dataset, and TimeGAN shows a poor result. However, data generated by TS-GAN perfectly fit original time series, which surpass the other three generation results.

Therefore, after 45,000 rounds of training, comparing with state-of-the-art GANs, we obtain the following conclusions: (1) TS-GAN shows the best convergence performance (approximately 0) in terms of the discriminator loss and MMD; (2) the distribution of data generated by TS-GAN is the most similar one to the real distribution in terms of PCA and t-SNE on mHealth dataset; (3) time-series data generated by TS-GAN are the most consistent

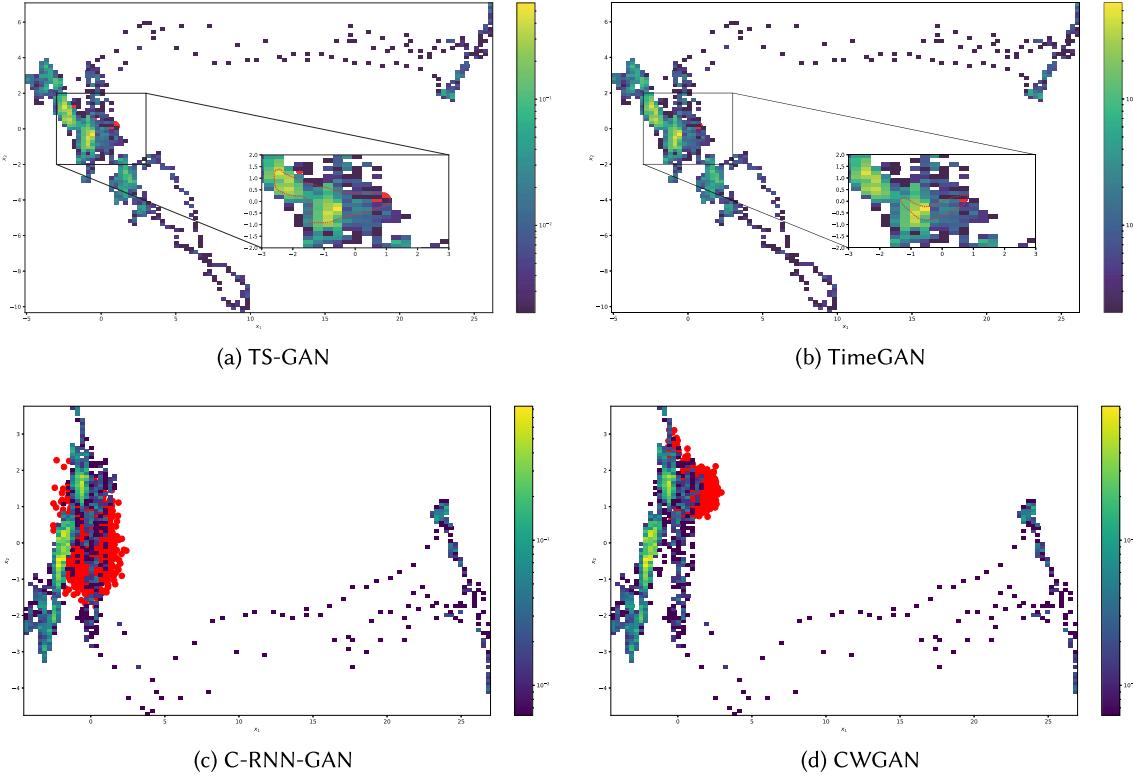


Fig. 9. Visualization of mHealth by PCA. The density map implies the distribution of real data X , and red points refer to time-series data generated by GANs.

with the real sequence diagram in terms of sequence diagrams experiments. In a real application scenario, in the augmentation process, since the noise sampled by the generator each time is sampled randomly from a fixed distribution, not all the time-series data generated each time are of high quality. To solve this problem, as shown in Figure 2, the data with high quality in all the generated data (the closer to the real distribution, the higher the quality) can be selected according to the discriminator loss, MMD and visualization methods to enlarge the original training set.

5.3 Accuracy of Classification

We train TS-GAN and three state-of-the-art GANs, i.e., TimeGAN, C-RNN-GAN, and CWGAN, with the same iterations to generate time-series data of ECG_200, NonInvasiveFatalECG_Thorax1, and mHealth, respectively. Then, we use fivefold cross-validation to train an LSTM binary classifier on each dataset, and we add the same volume of generated data from each GANs to training folds (fourfold), where we balance the proportion of positive and negative samples. Finally, we use the loss function of cross entropy to optimize the classifiers' parameters, and use the four indicators (Equation (14)) to evaluate the performance of the LSTM classifier on the test set (onefold):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (14a)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (14b)$$

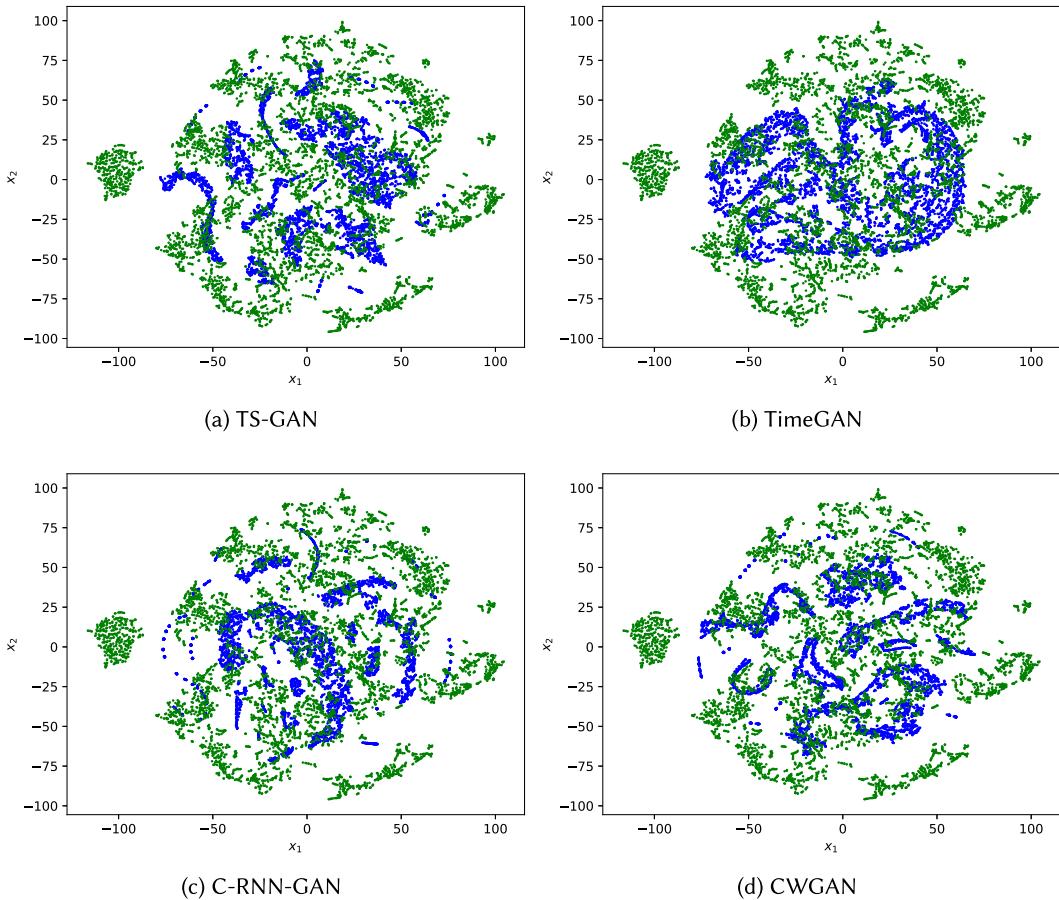


Fig. 10. Visualization of mHealth by t-SNE. The green points refer to real data, and the blue points refer to time series generated by GANs.

$$Precision = \frac{TP}{TP + FP}, \quad (14c)$$

$$F_1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}. \quad (14d)$$

Both ECG_200 and NonInvasiveFatalECG_Thorax1 datasets contain two classes of data, so the four indicators in Equation (14) can be directly calculated. For mHealth dataset that contains 10 classes, we randomly choose ten groups of data ($G = \{g_1, g_2, \dots, g_{10}\}$) for data augmentation, where each group g_i consists of two classes, i.e., $|\mathcal{Y}(g_i)| = 2$. We calculate the four indicators for every group, and take the mean value of all groups as the final score of the dataset. Taking the accuracy on mHealth dataset augmented by TS-GAN as an example, we can calculate the final accuracy $ACC_{mHealth_TS-GAN}$ for TS-GAN by Equation (15):

$$ACC_mHealth_TS-GAN = \frac{ACC_TS-GAN_g_1 + ACC_TS-GAN_g_2 + \dots + ACC_TS-GAN_g_{10}}{10}. \quad (15)$$

In per group of mHealth, positive samples refer to one volunteer's health data in this group and negative samples refer to the other volunteer, and the nature of the three datasets for training the LSTM classifiers is

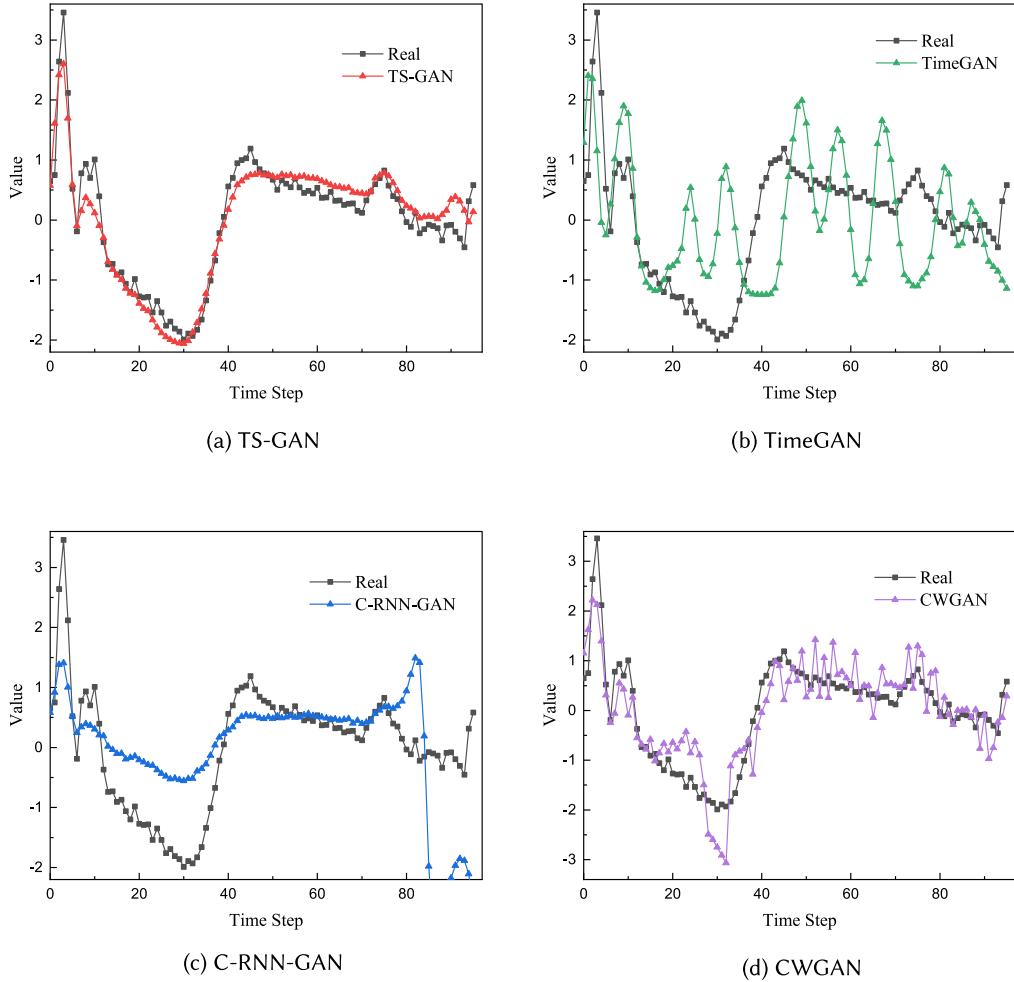


Fig. 11. Visualization of ECG_200 in time domain.

summarized in Table 1. To intuitively compare the effect of each GANs on the classifier performance, we take the classifier trained from the datasets without any data augmentation as the baseline. Thus, we obtain the performance results of the LSTM classifier on the augmented training sets from each GANs and the baseline in Table 2. As illustrated in Table 2, for ECG_200 and NonInvasiveFatalECG_Thorax1, each value represents the average across the five testing folds, and for mHealth, every value is the average of ten split groups calculated by the method of Equation (15), where each term of molecule is the mean of the five testing folds. The four indicators of the classifier trained on all the datasets augmented by TS-GAN exceed other GANs and baseline, which demonstrates the effectiveness of TS-GAN, where the generated data can improve the performance of the classification model. Specifically, for NonInvasiveFatalECG_Thorax1 dataset, there are few data for GAN training, and thus the performance of the four metrics is relatively worse, especially the lower recalls. Moreover, on the two ECG datasets, C-RNN-GAN and CWGAN greatly reduce the accuracy of the classifier, which means the patterns of data generated by these two GANs are not consistent with the original time series.

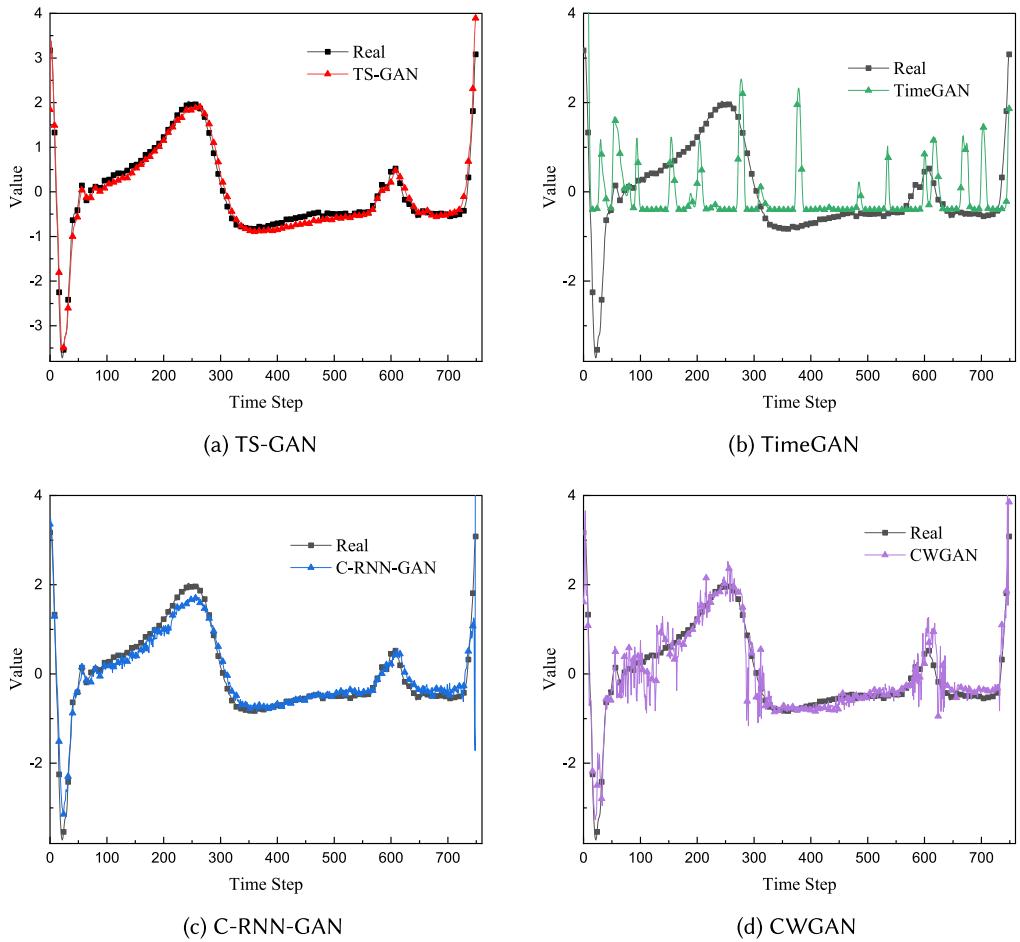


Fig. 12. Visualization of NoInvasiveFatalECG_Thorax1 in time domain.

Table 1. Summarization of Datasets for Training LSTM Classifiers

Dataset	Class	Sliding Window Length	Dimension	Training Sample	Test Sample
ECG_200	2	96	1	160/160	40
NonInvasiveFatalECG_Thorax1	2	750	1	71/71	17
mHealth	2 (per group)	200	23	156,933/156,966	39,242

Training samples (original samples/generated samples) and testing samples of ECG_200, NonInvasiveFatalECG_Thorax1, and per group of mHealth are given.

5.4 Effect of Generated Data

In previous experiments for training LSTM binary classifiers, we add the same amount of synthetic data to each training set (fourfold) similar to References [6, 37]. In this section, we study the effect of the proportion of the generated data added to each original training set on the classifier accuracy. For ECG_200, NonInvasiveFetalECG_Thorax1 and per group of mHealth, we add 0.5 time, 1 time, 1.5 times, and 2 times of the generated data

Table 2. Performance (%) of the Classifiers

Indicator	Method	ECG_200	NonInvasiveFatalECG_Thorax1	mHealth
Accuracy	TimeGAN	85.00	88.24	94.28
	C-RNN-GAN	75.00	58.82	88.98
	CWGAN	65.00	70.59	82.94
	Baseline	72.50	76.47	78.24
	TS-GAN	97.50	94.12	98.12
Recall	TimeGAN	88.46	90.00	92.96
	C-RNN-GAN	80.77	60.00	88.24
	CWGAN	73.08	70.00	80.56
	Baseline	73.08	70.00	83.12
	TS-GAN	100.00	90.00	96.34
Precision	TimeGAN	88.46	90.00	95.48
	C-RNN-GAN	77.78	66.67	89.57
	CWGAN	73.08	77.78	84.60
	Baseline	82.61	87.50	75.56
	TS-GAN	96.30	100.00	99.90
F_1	TimeGAN	88.50	90.09	93.90
	C-RNN-GAN	79.37	63.09	88.90
	CWGAN	72.99	73.80	82.53
	Baseline	77.52	77.82	79.37
	TS-GAN	98.95	94.79	98.09

Table 3. Accuracy Performance (%) of LSTM Classifiers Trained on Datasets with Different Proportion of Generated Data from TS-GAN

Model	ECG_200					NonInvasiveFatalECG_Thorax1					mHealth					
	0.5	1	1.5	2	0.5	1	1.5	2	0.5	1	1.5	2	0.5	1	1.5	2
TS-GAN	90.00	97.50	92.50	85.00	82.35	94.12	82.35	76.47	90.62	98.12	97.46	95.22				
TimeGAN	80.00	85.00	77.50	67.50	82.35	88.24	76.47	70.59	89.90	94.28	94.73	92.31				
CWGAN	70.00	65.00	60.00	62.50	64.71	70.59	58.82	52.94	81.38	82.94	81.66	81.11				
C-RNN-GAN	70.00	75.00	62.50	65.00	52.94	58.82	58.82	52.94	85.50	88.98	89.17	85.99				

from GANs to the original training set, and we use the augmented training set to train series of LSTM binary classifiers. The accuracy performance of each classifier is listed in Table 3, and each value represents the average across the five testing folds as the same as Table 2. As depicted in Table 3, for different datasets, the classifiers trained with a ratio near 1 between the generated data and the original training set show the highest accuracy. In addition, the classification accuracy is sensitive to the proportion of generated data in the small training set, i.e., adding a few generated samples (as low as 35) to the small training set has a great impact on the accuracy.

6 CONCLUSION AND DISCUSSION

In this article, we propose a TS-GAN, a Time-series GAN architecture based on LSTM networks to augment health data collected by sensors for improving the accuracy of deep learning-based classification models. Specifically, TS-GAN includes an LSTM-based generator for creating realistic data and an LSTM-based discriminator for determining the similarity between the generated data and real data. Besides, TS-GAN utilizes Wasserstein distance, 1D convolution and SSE module to generate time-series health data with the same time and space

dependence as real data, and ensures the stability of the training process simultaneously. To prove the superiority of TS-GAN, we conduct experiments to compare the performance of TS-GAN with three state-of-the-art GAN models, i.e., TimeGAN, C-RNN-GAN, and CWGAN, qualitatively and quantitatively. In our qualitative study, we compare the discriminator loss and MMD of data from GANs, and visualize the generated data with the real data by PCA, t-SNE and sequence diagrams on the health datasets of ECG_200, NonInvasiveFatalECG_Thorax1, mHealth, and we preliminarily prove the distribution of data generated by TS-GAN is the most consistent with real data among all the GANs. In the accuracy of classification experiment, we quantitatively prove that the effectiveness of the data generated by TS-GAN exceeds that of other state-of-the-art GANs on all datasets in terms of the accuracy, recall, precision and F_1 score.

Through visualization results, i.e., PCA, t-SNE and sequence diagrams, we fundamentally find that TimeGAN tends to omit local features of original data; C-RNN-GAN generates data with many outliers and large variance; and CWGAN generates data with a great deviation. However, for univariate data, the generated data of our TS-GAN smoothly fit the trend and maintain local features of the real data, and for multivariate data, TS-GAN can generate data with almost the same distribution of the real data. The results of quantitative experiments, i.e., classifier accuracy, further prove that: (1) data generated by GANs are more effective on mHealth dataset than that on ECG_200 with small volume of data, i.e., 94.28% vs. 85.00% of TimeGAN, 88.98% vs. 75.00% of C-RNN-GAN, 82.94% vs. 65.00% of CWGAN, and 98.12% vs. 97.50% of TS-GAN, which imply the reduction of training data greatly reduces the effectiveness of GANs, and (2) whether on univariate datasets or multivariable datasets, the data generated by TS-GAN are more effective than that generated by other state-of-the-art GANs. In the end, we prove that for ECG_200, NonInvasiveFatalECG_Thorax1 and mHealth, the classifier can get the highest accuracy when the ratio of generated samples from GANs to the original training samples nears 1.

Our TS-GAN is a model that can be utilized to generate realistic time series with the same space and time dependence as real data only by an LSTM-based generator and an LSTM-based discriminator with an SSE module and 1D convolution. However, before applying TS-GAN to train downstream classifiers, series of early experiments are essential to determine the optimal hyperparameters, such as the length of the segmentation window, the proportion of generated samples to original samples, and hyperparameters of the generator and the discriminator. What is more, the performance of TS-GAN is also degraded like other GANs when trained on few-shot datasets, therefore, our future work will concentrate more on achieving augmentation of few-shot, or even zero-shot datasets.

REFERENCES

- [1] Md Momin Al Aziz, Tanbir Ahmed, Tasnia Faequa, Xiaoqian Jiang, Yiyu Yao, and Noman Mohammed. 2021. Differentially private medical texts generation using generative neural networks. *ACM Trans. Comput. Healthcare* 3, 1, Article 5 (Oct. 2021), 27 pages. <https://doi.org/10.1145/3469035>
- [2] Talal Alshammari, Nasser Alshammari, Mohamed Sedky, and Chris Howard. 2018. SIMADL: Simulated activities of daily living dataset. *Data* 3, 2 (2018), 11.
- [3] Damla Arifoglu and Abdelhamid Bouchachia. 2019. Abnormal behaviour detection for dementia sufferers via transfer learning and recursive auto-encoders. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops'19)*. IEEE, 529–534.
- [4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*. PMLR, 214–223.
- [5] Oresti Banos, Claudia Villalonga, Rafael Garcia, Alejandro Saez, Miguel Damas, Juan A. Holgado-Terriza, Sungyong Lee, Hector Pomares, and Ignacio Rojas. 2015. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *Biomed. Eng. Online* 14, 2 (2015), 1–20. Retrieved from <http://archive.ics.uci.edu/ml/datasets/mhealth+dataset>.
- [6] Christopher Bowles, Liang Chen, Ricardo Guerrero, Paul Bentley, Roger Gunn, Alexander Hammers, David Alexander Dickie, Maria Valdés Hernández, Joanna Wardlaw, and Daniel Rueckert. 2018. Gan augmentation: Augmenting training data using generative adversarial networks. Retrieved from <https://arXiv:1810.10863>.
- [7] Qili Chen, Bofan Liang, and Jiuhe Wang. 2019. A comparative study of LSTM and phased LSTM for gait prediction. *Int. J. Artificial. Intelli. App.* 10, 4 (2019), 57–66.

- [8] Tsz-Him Cheung and Dit-Yan Yeung. 2020. Modals: Modality-agnostic automated data augmentation in the latent space. In *Proceedings of the International Conference on Learning Representations*.
- [9] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. 2018. Autoaugment: Learning augmentation policies from data. Retrieved from <https://arXiv:1805.09501>.
- [10] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 702–703.
- [11] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharhabibi, Chotirat Ann Ratana-mahatana, and Eamonn Keogh. 2019. The UCR time-series archive. *IEEE/CAA J. Autom. Sinica* 6, 6 (2019), 1293–1305. Retrieved from https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.
- [12] Shaojiang Deng, Jiaxing Luo, and Yantao Li. 2021. CNN-based continuous authentication on smartphones with auto augmentation search. In *Information and Communications Security*, Debin Gao, Qi Li, Xiaohong Guan, and Xiaofeng Liao (Eds.). Springer International Publishing, Cham, 169–186.
- [13] Sander Dieleman, Kyle W. Willett, and Joni Dambre. 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices Roy. Astron. Soc.* 450, 2 (2015), 1441–1459.
- [14] Zahra Ebrahimi, Mohammad Loni, Masoud Daneshbalab, and Arash Gharehbaghi. 2020. A review on deep learning methods for ECG arrhythmia classification. *Expert Syst. Appl.: X* 7 (2020), 100033.
- [15] Imanne El Maachi, Guillaume-Alexandre Bilodeau, and Wassim Bouachir. 2020. Deep 1D-convnet for accurate Parkinson disease detection and severity prediction from gait. *Expert Syst. Appl.* 143 (2020), 113075.
- [16] Farzan Farnia and Asuman Ozdaglar. 2020. Do GANs always have Nash equilibria? In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Hal Daume III and Aarti Singh (Eds.), Vol. 119. PMLR, 3029–3039.
- [17] Tonya Fields, George Hsieh, and Jules Chenou. 2019. Mitigating drift in time-series data with noise augmentation. In *Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI'19)*. IEEE, 227–230.
- [18] Alexander L. Fogel and Joseph C. Kvedar. 2018. Artificial intelligence powers digital medicine. *NPJ Dig. Med.* 1, 1 (2018), 1–4.
- [19] Elizabeth Fons, Paula Dawson, Xiao-jun Zeng, John Keane, and Alexandros Iosifidis. 2021. Adaptive weighting scheme for automatic time-series data augmentation. Retrieved from <https://arXiv:2102.08310>.
- [20] Ian Goodfellow. 2016. Nips 2016 tutorial: Generative adversarial networks. Retrieved from <https://arXiv:1701.00160>.
- [21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.
- [22] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* 18, 5-6 (2005), 602–610.
- [23] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of wasserstein gans. Retrieved from <https://arXiv:1704.00028>.
- [24] Yang Guo, Dongsheng An, Xin Qi, Zhongxuan Luo, Shing-Tung Yau, Xianfeng Gu, et al. 2019. Mode collapse and regularity of optimal transportation maps. Retrieved from <https://arXiv:1902.02934>.
- [25] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7132–7141.
- [26] Amir Hussein, Marc Djandji, Reem A. Mahmoud, Mohamad Dhaybi, and Hazem Hajj. 2020. Augmenting DL with adversarial training for robust prediction of epilepsy seizures. *ACM Trans. Comput. Healthcare* 1, 3, Article 18 (June 2020), 18 pages. <https://doi.org/10.1145/3386580>
- [27] Navdeep Jaitly and Geoffrey E. Hinton. 2013. Vocal tract length perturbation (VTLP) improves speech recognition. In *Proceedings of the International Conference on Machine Learning Workshop on Deep Learning for Audio, Speech and Language (ICML'13)*, Vol. 117. 21.
- [28] Madhura Joshi, Ankit Pal, and Malaikannan Sankarasubbu. 2022. Federated learning for healthcare domain - pipeline, applications and challenges. *ACM Trans. Comput. Healthcare* 3, 4 (2022), 36 pages. <https://doi.org/10.1145/3533708>
- [29] Konrad Kluwak and Teodor Niżyński. 2020. Gait classification using LSTM networks for tagging system. In *Proceedings of the IEEE 15th International Conference of System of Systems Engineering (SoSE'20)*. IEEE, 295–300.
- [30] Sayeri Lala, Maha Shady, Anastasiya Belyaeva, and Molei Liu. 2018. Evaluation of mode collapse in generative adversarial networks. In *Proceedings of the IEEE High Performance Extreme Computing Conference (HPEC'18)*. IEEE, 1–9.
- [31] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. 2017. Mmd gan: Towards deeper understanding of moment matching network. Retrieved from <https://arXiv:1705.08584>.
- [32] Yantao Li, Hailong Hu, and Gang Zhou. 2018. Using data augmentation in continuous authentication on smartphones. *IEEE Internet Things J.* 6, 1 (2018), 628–640.
- [33] Yantao Li, Li Liu, Huafeng Qin, Shaojiang Deng, Mounim A. El-Yacoubi, and Gang Zhou. 2022. Adaptive deep feature fusion for continuous authentication with data augmentation. *IEEE Trans. Mobile Comput.* (2022), 1–16. <https://doi.org/10.1109/TMC.2022.3186614>
- [34] Yuan-Pin Lin, Yi-Hsuan Yang, and Tzyy-Ping Jung. 2014. Fusion of electroencephalographic dynamics and musical contents for estimating emotional responses in music listening. *Front Neurosci.* 8 (2014), 14 pages. <https://doi.org/10.3389/fnins.2014.00094>

- [35] Xinwen Liu, Huan Wang, Zongjin Li, and Lang Qin. 2021. Deep learning in ECG diagnosis: A review. *Knowl.-Based Syst.* 227 (2021), 107187.
- [36] Yifei Lu, Wei-Long Zheng, Binbin Li, and Bao-Liang Lu. 2015. Combining eye movements and EEG to enhance emotion recognition. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*.
- [37] Yun Luo and Bao-Liang Lu. 2018. EEG data augmentation for emotion recognition using a conditional Wasserstein GAN. In *Proceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'18)*. IEEE, 2535–2538.
- [38] Olof Mogren. 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. Retrieved from <https://arXiv:1611.09904>.
- [39] Daisuke Oba, Shinnosuke Matsuo, and Brian Kenji Iwana. 2021. Dynamic data augmentation with gating networks. Retrieved from <https://arXiv:2111.03253>.
- [40] Esteban Piacentino, Alvaro Guarner, and Cecilio Angulo. 2021. Generating synthetic ECGs using GANs for anonymizing healthcare data. *Electronics* 10, 4 (2021), 389.
- [41] Justin Salamon and Juan Pablo Bello. 2017. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Process. Lett.* 24, 3 (2017), 279–283. <https://doi.org/10.1109/LSP.2017.2657381>
- [42] Arash Shaban-Nejad, Martin Michalowski, and David L. Buckeridge. 2018. Health intelligence: How artificial intelligence transforms population and personalized health. *npj Digital Med* 1 (2018), 2 pages. <https://doi.org/10.1038/s41746-018-0058-9>
- [43] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. 2019. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4570–4580.
- [44] Terry T. Um, Franz M. J. Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. 216–220.
- [45] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 86 (2008), 2579–2605.
- [46] Yong Xia, Naren Wulan, Kuanquan Wang, and Henggui Zhang. 2017. Atrial fibrillation detection using stationary wavelet transform and deep learning. In *Proceedings of the Computing in Cardiology (CinC'17)*. IEEE, 1–4.
- [47] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. 2019. Time-series generative adversarial networks. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS'19)*, Vol. 32, 1–11.
- [48] Chenshuang Zhang, Guijin Wang, Jingwei Zhao, Pengfei Gao, Jianping Lin, and Huazhong Yang. 2017. Patient-specific ECG classification based on recurrent neural networks and clustering technique. In *Proceedings of the 13th IASTED International Conference on Biomedical Engineering (BioMed'17)*. IEEE, 63–67.
- [49] Zhen Zhao, Ze Li, Fuxin Li, and Yang Liu. 2021. CNN-LSTM based traffic prediction using spatial-temporal features. In *Journal of Physics: Conference Series*, Vol. 2037. IOP Publishing, 012065.
- [50] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2223–2232.

Received 7 July 2022; revised 19 December 2022; accepted 26 January 2023