Biosignal Data Augmentation Based on Generative Adversarial Networks*

Shota Harada[1], Hideaki Hayashi[2], and Seiichi Uchida[2]

Abstract— **In this paper, we propose a synthetic generation method for time-series data based on generative adversarial networks (GANs) and related application to data augmentation for biosignal classification. GANs are a recently proposed framework for learning a generative model, where two neural networks, one generates synthetic data and the other discriminates synthetic and real data, are trained while competing with each other. In the proposed method, each neural network in GANs is developed based on a recurrent neural network using long short-term memories, thereby allowing the adaptation of the GANs framework to time-series data generation. In the experiments, we confirmed the capability of the proposed method for generating synthetic biosignals using the ECG and EEG datasets. We also showed the effectiveness of the proposed method for data augmentation in the biosignal classification problem.**

I. Introduction

Biosignals such as electrocardiogram (ECG) and electroencephalogram (EEG) strongly reflect human internal states. In particular, abnormality in the human body including diseases can cause visible changes in the patterns of biosignals. For example, epilepsy induces drastic changes in the amplitude and frequency of EEG. Abnormality in the human body can therefore be detected by classifying patterns of biosignals. In fact, physicians refer to the pattern of biosignals to diagnose diseases and to decide the direction of treatment.

Many studies have investigated the classification of biosignals using machine learning techniques including deep learning [1], [2]. Thanks to the development of deep learning, some studies achieved drastic increase in classification accuracy.

One of the limitations of deep learning-based techniques is that a large amount of training data is required to obtain enough accuracy; thus the number of training data is important in the practical applications. Unfortunately, the number of training data is often limited in the real-world biosignal classification problems. There are two main reasons for this. One is that the number of participants for the measurement is limited under some conditions. The other is that a special qualification or expert knowledge is required for annotation to label each sample. Therefore, it is necessary to manage with limited training data.

To synthetically increase the number of training data, many data augmentation techniques have been proposed. Appropriate data augmentation is effective for improving classification accuracy [3]. The simplest method for data augmentation is the synthesis of given data using interpolation and extrapolation. The disadvantage of this method is that the variety of synthesized data is limited depending on the density of given data. Another method often used is a generative model. In generative models, the generation process of the observations is drawn from a probability distribution, and new data are then generated based on a random process. Generally speaking, generative models show a good performance only if the assumed model structure enough approximates the true data distribution. To satisfy this requirement, mathematical models constructed based on the domain dependent knowledge are often required.

Generative adversarial networks (GANs) [4] are a recently proposed framework for learning a generative model. In GANs, two neural networks, one generates synthetic data and the other discriminates the synthetic data from real data, are simultaneously trained while competing with each other. Using this framework, it is possible to generate data similar to given observations without domain dependent knowledge of the target. In recent years, many studies using GANs have been proposed, and the effectiveness for data augmentation has also been reported [5]. However, these studies mainly focus on the generation of images, and only few studies reported the generation of time-series data [6]. To the best of our knowledge, the generation of biosignals using GANs has not been reported.

In this paper, we propose a generation method for time-series data based on GANs and related application to data augmentation for biosignal classification. In the proposed method, each neural network in GANs is developed based on a recurrent neural network (RNN) using long short-term memories (LSTM) [7] for its hidden layers, thereby allowing the adaptation of the GAN framework to time-series data generation. In the experiments, we verified the capability of the proposed method for generating synthetic biosignals using two datasets.

The primary contributions of this work are as follows:

- The capability of GANs for generating biosignals is verified.
- The effectiveness for data augmentation was evaluated via biosignal classification experiments.

*This work was partially supported by JSPS KAKENHI Grant Numbers JP17K19402 and JP17K12752.

[1]S. Harada is with Graduate School of Systems Life Sciences Kyushu University, 744 Motooka Nishi-ku Fukuoka, 819-0395, Japan (email: shota.harada@human.ait.kyushu-u.ac.jp)

[2]H. Hayashi and S. Uchida are with Faculty of Information Science and Electrical Engineering, Kyushu University, 744 Motooka Nishi-ku Fukuoka, 819-0395, Japan (email: hayashi@ait.kyushu-u.ac.jp, uchida@ait.kyushu-u.ac.jp)

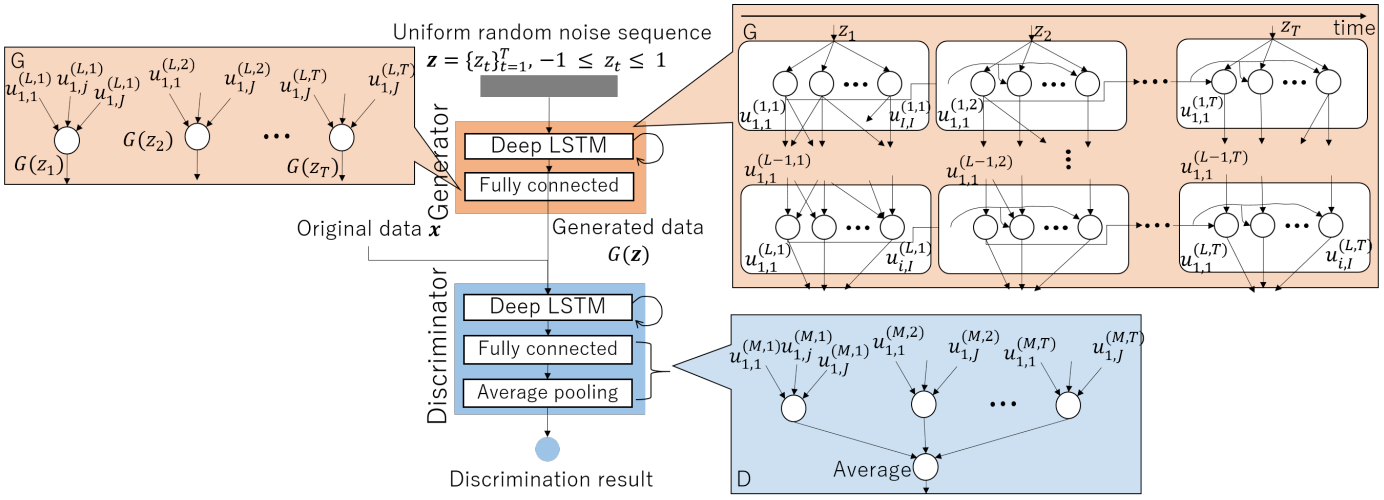


Fig. 1: Proposed time-series data generation method

II. Synthetic Time-Series Data Generation Method

A. Generative Adversarial Networks

GANs are a framework for estimating generative models proposed by Ian Goodfellow in 2014 [4]. GANs have received much attention in recent years, particularly in the computer vision community, and various derivatives have been proposed by changing their learning methods and structures.

GANs consist of two different networks. One is a generative model called a generator. A vector of random numbers is fed into this network, and data with the same dimensions as the training data is produced. The other is a discriminative model called a discriminator. In this network, training data and the data generated by the generator are input. The network then discriminates whether the input is training data or generated data.

In the GANs framework, the generator and the discriminator are trained adversarially. The relationship between them is often compared to that of banknote counterfeiters and police. The generator learns to generate data that the discriminator identifies as training data. On the other hand, the discriminator learns to discriminate training data and generated data correctly. As a result, the generator gradually becomes able to generate data similar but not exactly same as the training data. In other words, the generator learns the mapping from a distribution of random numbers onto the distribution of training data.

B. The Proposed Time-Series Data Generation Method

Fig. 1 shows the structure of the proposed method. Based on the GANs framework, the proposed method consists of a generator G and a discriminator D. In contrast to most of the existing GANs for image generation, the proposed method is constructed with a RNN based on LSTMs to adapt to time-series data.

The generator G consists of a deep LSTM layer and a fully connected layer. The deep LSTM layer has L hidden layers and I LSTM units in each layer. A sigmoid activation function is used in the fully connected layer. The generator receives a sequence $\boldsymbol{z} = \{z_t\}_{t=1}^{T}$ (T is the length of sequence of training data) as an input. At each time point t, z_t is independently sampled from uniform random distribution in $[-1, 1]$ and then fed into the generator. In the deep LSTM layer, the output from the i-th unit in the l-th layer at the time point t, $u_{i,j}^{(l,t)}$, is passed to the j-th unit in the $l+1$-th layer. The sequence of the output from the fully connect layer is treated as the generated time-series $G(\boldsymbol{z}) \in \mathbb{R}^T$.

The discriminator consists of a deep LSTM layer, a fully connected layer, and an average pooling layer. The average pooling layer outputs a scalar by averaging the input over its dimensions. Given a input sequence $\boldsymbol{x} \in \mathbb{R}^T$, the output of the discriminator $D(\boldsymbol{x})$ is a scalar value representing the probability that $\boldsymbol{x}$ came from the training data. In the discriminator, the LSTM layer consists of M hidden layers and J units for each layer, and the fully connected layer has K units with sigmoid activation functions.

In the training, D and G play the minimax game with the evaluation function defined as

$$\begin{aligned}\min_G \max_D V(D,G) =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[\log D(\boldsymbol{x})\right] \\ &+\mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}\left[\log\left(1-D\left(G\left(\boldsymbol{z}\right)\right)\right)\right],\end{aligned} \tag{1}$$

where $p_{\text{data}}(\boldsymbol{x})$ and $p_z(\boldsymbol{z})$ are the distributions of the training data and $\boldsymbol{z}$, respectively. The training procedures are as follows: First, m samples of random sequences $\{\boldsymbol{z}^{(1)}, ..., \boldsymbol{z}^{(m)}\}$ are sampled from noise distribution p_z. Similarly, m examples of data $\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}\}$ are sampled from the training dataset. The weights of D are updated by ascending its stochastic gradient:

$$\nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(\boldsymbol{x}^{(i)}\right)+\log\left(1-D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]. \tag{2}$$

Furthermore, m samples of random sequences $\{\boldsymbol{z}^{(1)}, ..., \boldsymbol{z}^{(m)}\}$ are sampled from p_z again. The weights of G are then updated by descending its stochastic gradient:

$$\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right). \tag{3}$$

These steps were repeated for the number of training iterations.

The gradients with respect to weights in the networks are calculated using the back propagation through time method. The weights are then updated based on the updating rule proposed in [8]. Using this updating rule, we can avoid the mode collapse and eliminate the bias in the data generated from the generator.

III. Evaluation of the Proposed Method for Data Augmentation

A. Experimental Setup

To verify the validity of data augmentation using the proposed method, we conducted biosignal classification experiments. The effectiveness of data augmentation using the proposed method was evaluated by adding data generated from the proposed method to training data for a classifier. In addition, we assessed the changes in accuracy in accordance with the number of augmented data and compared to those with other existing data augmentation methods.

In this experiment, we used two biosignal datasets. One is ECG dataset called "ECG 200" created by Robert Thomas Olszewski [9]. This dataset consists of 200 samples of ECG series. Each series traces the electric activity during one heartbeat, and the length of each series is 96. Out of 200 samples, 133 are labeled as normal and the remaining 67 are myocardial infarction (referred to as abnormal hereinafter). We randomly extracted 60 samples for each class to make the number uniform, and split them into 20, 4, and 36 for training, validation, and testing, respectively. The other is EEG dataset called "Epileptic Seizure Recognition Data Set" donated by Qiuyi Wura [10]. This dataset contains 9200 normal series and 2300 abnormal series recorded during epileptic seizure. We randomly extracted 2300 samples for each class to make the number uniform, and split them into 460, 184, and 1656 for training, validation, and testing, respectively. The length of each series is 178.

In the training of the proposed method, we used whole training data for each dataset. In the ECG dataset generation experiment, the numbers of LSTM units for the generator and the discriminator was $I = J = 200$, and the numbers of layers of LSTM layer were $L = M = 3$. In the EEG dataset generation experiment, the parameters were set as $I = J = 300$, and $L = M = 4$.

For classification, we used a neural network constructed based on LSTMs as a classifier. This network consists of two hidden layers with 20 LSTM units for each layer and a fully connected layer as the output layer with a softmax activation function. Full batch learning was used regardless of the number of training data. For the training data of the classifier, we used data generated from the proposed method as data augmentation in addition to the original training data. The classification accuracy was then calculated for the testing data by changing the number of augmented data from 0 to 400 at intervals of 40 samples for the ECG dataset, and from 0 to 9,200 at intervals of 920 samples for the EEG dataset.

B. Comparative Data Augmentation Methods

We compared the results using the proposed method with those using three existing data augmentation methods. The first method is noise addition. The training data are augmented by adding noise generated from a Gaussian distribution.

$$\boldsymbol{x}^{(i)'} = \boldsymbol{x}^{(i)} + \gamma \boldsymbol{X},\ \boldsymbol{X} \sim N(0, \sigma^2), \tag{4}$$

where $\boldsymbol{x}^{(i)}$ is the i-th sample of the training data, $N(0, \sigma^2)$ is a Gaussian distribution with a mean of zero and a variance of σ^2, and γ is a constant value. Variance σ^2 was calculated from the training data and γ was set as 0.5. The second method is the interpolation. New samples are synthesized based on the linear interpolation between the original samples as

$$\boldsymbol{x}^{(i)'} = \lambda \boldsymbol{x}^{(i)} + (1 - \lambda)\boldsymbol{x}^{(k)}, \tag{5}$$

where $\boldsymbol{x}^{(k)}$ is the training sample most similar to $\boldsymbol{x}^{(i)}$, and λ is the coefficient related to interpolation. The similarity between samples is calculated by dynamic time warping (DTW). The third method is the Hidden Markov Model (HMM). Each state of the HMM was constructed with a Gaussian distribution. The number of states of the HMM was determined based on Akaike's Information Criterion (AIC).

C. Results

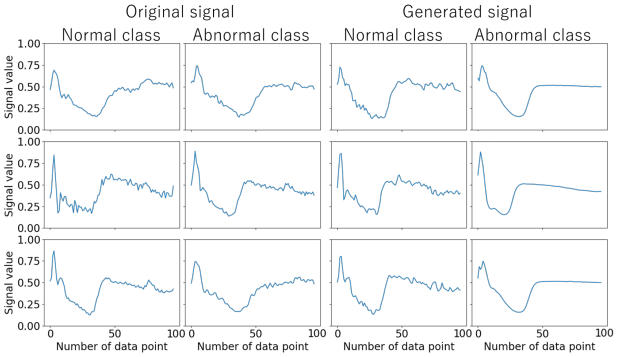
Fig. 2 shows examples of the original and generated data using the proposed method. Fig. 2(a) and (b) are the examples for the ECG dataset and the EEG dataset, respectively. In each figure, three medoids obtained by k-medoids clustering ($k = 3$) are shown as original signal examples. For generated signal examples, a sequence most similar to each original signal example is selected based on DTW. Fig. 3 shows the comparison of the generated data for the EEG dataset with the other methods.

Fig. 4 shows the comparison of classification accuracy for (a) the ECG dataset and (b) the EEG dataset. In each figure, the changes in accuracy according to the number of augmented training data are shown for each data augmentation method.

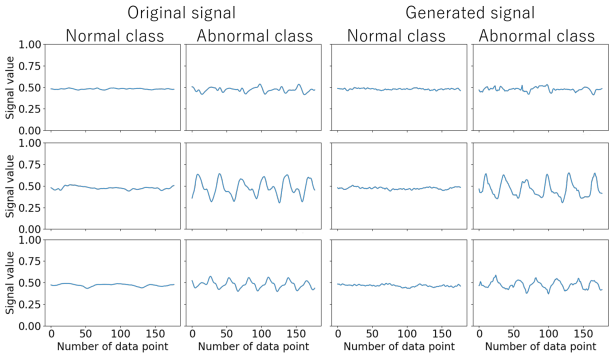
D. Discussion

In Fig. 2, it can be confirmed that the proposed method generates time-series data that have similar characteristics to the original data. For the ECG dataset, the peak near the initial time point and the rapid decrease and increase around the 30th time point are traced. Variations in frequency and amplitude for the EEG dataset are also mostly reproduced. Furthermore, the generated data have different characteristics for each class. In Fig. 3, the data generated by the noise addition and the HMM seem to have higher frequency than the original data because these methods directly use Gaussian noise during the data generation. On the other hand, the interpolation tends to generate smooth waveforms whereas the detailed features are lost.

The effectiveness of data augmentation by the proposed method was confirmed from the changes in classification



(a) ECG dataset



(b) EEG dataset

Fig. 2: Example of the original and generated signals. Three medoids obtained by k-medoids clustering ($k = 3$) from the original dataset are shown as the original signal examples. A signal most similar to each original signal exmaple is selected from the generated data using the proposed method.

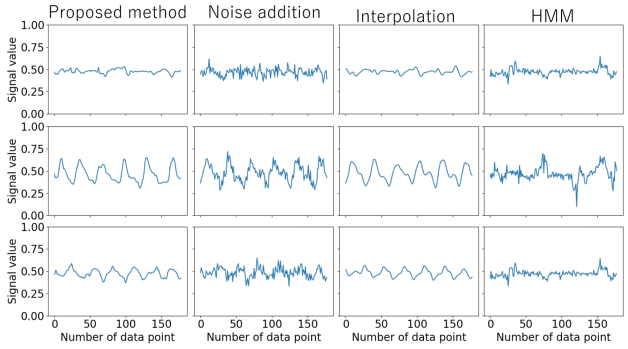
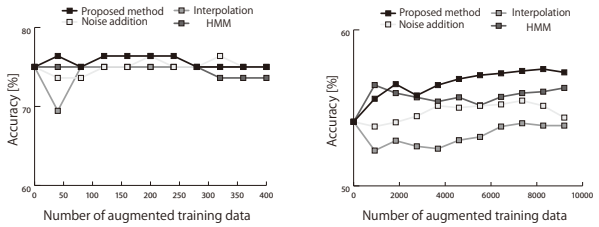


Fig. 3: Comparison of the generated data. Three examples from data generated with each method for the abnormal EEG are shown. For each method signals most similar to the original signal example shown in Fig. 2(b) are selected.



(a) ECG dataset

(b) EEG dataset

Fig. 4: Comparison of classification accuracy. The changes in classification accuracy according to the number of augmented data are shown for each data augmentation method.

accuracy in Fig. 4. In Fig. 4(a), the accuracy increased if the number of augmented data was appropriate although the increase was small and the differences from the other methods were slight. In Fig. 4(b), however, the classification accuracy monotonically increased according to the number of augmented data, and the proposed method showed the highest performance when the number of augmented data was over 2,000. These results showed the effectiveness using the data augmentation using the proposed method.

IV. Conclusion

In this paper, we proposed a generation method for synthetic time-series data based on GANs. We also applied the proposed method to data augmentation for biosignal classification. In the proposed method, each neural network in GANs is developed based on a RNN using LSTM units for its hidden layers, thereby allowing the generation of time-series data.

In the experiments conducted during the study, we confirmed the capability of the proposed method for generating synthetic biosignals using the ECG and EEG datasets. We also evaluated the effectiveness of the proposed method for data augmentation in the biosignal classification problem and compared with the other existing data augmentation methods. The results showed that the data augmentation by the proposed method is effective.

In future work, we will conduct further analysis of the relationship between the generated data and the latent space.

References

[1] N. Dewangan, "A Survey on ECG Signal Feature Extraction and Analysis Techniques," *Int. J. of Innov Res. in Electr., Electron., Instrum. and Control Eng.*, vol. 3, pp. 12–19, 06 2017.

[2] B. Pourbabaee, M. J. Roshtkhari, and K. Khorasani, "Deep convolutional neural networks and learning ecg features for screening paroxysmal atrial fibrillation patients," *IEEE Trans. on Syst., Man, and Cybern.: Syst.*, vol. PP, no. 99, pp. 1–10, 2017.

[3] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data Augmentation for Time Series Classification using Convolutional Neural Networks," in *ECML/PKDD Workshop on Adv. Anal. and Learn. on Temporal Data*, Riva Del Garda, Italy, Sept. 2016.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Adv. in Neural Inf. Process. Syst. 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[5] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "Cvae-gan: Fine-grained image generation through asymmetric training," *2017 IEEE I. Conf. Comp. Vis. (ICCV)*, pp. 2764–2773, 2017.

[6] O. Mogren, "C-rnn-gan: A continuous recurrent neural network with adversarial training," in *Constr. Learn. Workshop at NIPS 2016*, 2016, p. 1.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[8] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," in *Pro of the International Conference on Learning Representations (ICLR)*, 2017.

[9] R. T. Olszewski, "Generalized feature extraction for structural pattern recognition in time-series data," Ph.D. dissertation, Pittsburgh, PA, USA, 2001, aAI3040489.

[10] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state," *Phys. Rev. E Stat. Nonlin. Soft Matter. Phys.*, vol. 64, no. 6 Pt 1, p. 061907, Dec 2001.