

Received September 22, 2021, accepted October 29, 2021, date of publication November 8, 2021, date of current version November 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3126345

Contextual Imputation With Missing Sequence of EEG Signals Using Generative Adversarial Networks

WOONGHEE LEE, JAEYOUNG LEE, AND YOUNGHOON KIM¹

Department of Applied Artificial Intelligence, Hanyang University ERICA, Ansan 15588, South Korea

Corresponding author: Younghoon Kim (nongaussian@hanyang.ac.kr)

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korea Government [Ministry of Science and ICT (MSIT)] through the Artificial Intelligence Convergence Research Center (Hanyang University ERICA), under Grant 2020-0-01343, and in part by the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) under Grant 2020R1G1A1011471.

ABSTRACT Missing values are very prevalent in real world; they are caused by various reasons such as user mistakes or device failures. They often cause critical problems especially in medical and healthcare application since they can lead to incorrect diagnosis or even cause system failure. Many of recent imputation techniques have adopted machine learning-based generative methods such as generative adversarial networks (GANs) to deal with missing values in medical data. They are, however, incapable of reproducing realistic time-series signals preserving important latent features such as sleep stages that are important context in many medical applications using electroencephalogram (EEG). In this study, we propose a novel GAN-based technique generating realistic EEG signal sequences which are not only shown natural but also correctly classified with sleep stages by implanting the latent features in the synthetic sequence. By experiments, we confirm that our model generates not only more realistic EEG signals than a recent GAN-based model but also preserve auxiliary information such as sleep stages. Furthermore, we demonstrate that existing machine learning methods based on EEG data still work well without sacrificing performance using the imputed data by using our method.

INDEX TERMS Missing data imputation, electroencephalogram (EEG), generated adversarial network (GAN).

I. INTRODUCTION

In most of time series data analysis, missing values coming up by various reasons such as user mistakes or device failures lead to performance degradation or even cause system failure. Recent imputation techniques have adopted not only traditional statistical imputation but also machine learning based generative method to deal with missing values. These methods, however, are incapable of generating realistic time-series signals involving important latent information which is necessary for being exploited in the target application such as sleep disorder diagnosis based on electroencephalogram (EEG).

Having complete dataset in real world is unfortunately almost impossible [1]. Especially, in medicine and healthcare,

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry².

it was also reported that the majority of records contains a large number of missing values [2], [3]. The failed recording is possibly because of a malfunction of the recording device, lost records or a mistake in electrode attachment [4]. In addition, it is difficult to record complete EEG data, because of the strict requirements of recording environments or for the subjects [5]. Accordingly, most of applications utilizing such medical datasets suffer from missing values so that they may make wrong alerts or incorrect diagnoses [6], [7]. Furthermore, doctors or clinicians may also have trouble scoring sleep stages or diagnose sleep disorders due to such missing data because they need to consider the context through the precedent and following signal sequences according to [8]. To make matters worse, a missing value in such circumstances is usually occurred repeatedly until the cause is removed since recording lasts long time without frequent monitoring. Thus, existing imputing methods cannot

handle such cases effectively even if they can reconstruct for a single or short-term missing values by interpolation based on adjacent non-missing values.

A. OUR CONTRIBUTION

In this paper, we develop a novel deep neural network-based technique to complete the missing EEG signals which not only look natural, but also preserve contextual information well that is significant for the analysis of the data. In detail, we assume that our dataset includes the sequences of EEG signals and annotations of sleep stages periodically labelled in the sequences. The sleep stage is a category of 5 types indicating REM, Sleeping 1~3 and Wake, which is annotated by expertise aiming at diagnosis of epileptic, neurological and sleep disorder, measurement of mental health conditions, and psycho-physiological research [9]. Hence, preserving such characteristics in the created EEG signals is an important goal in our work, and there does not exist a technique completing missing values with the consideration of the contextual features to our best knowledge.

To generate realistic EEG signals to replace missing values, we adopt a generative adversarial networks (GAN). Due to its successful applications in image generation, GAN has been widely used in the imputation for image and time series data as well. Image inpainting which fills missing pixels of a picture also has achieved remarkable performance by using GANs [10]–[13], however, all the techniques are based on the assumption that they are available with complete dataset, that is, image without missing parts. Furthermore, a recent work [14] utilizes GAN to generate fake EEG signals for data augmentation but the model does not consider any context at all. Therefore, we adopt GAN for not only EEG signal generation but also acquiring contextual information for data augmentation. The contribution of our work can be summarized as follows:

- We suggest a novel GAN-based technique to generate synthetic EEG signals which looks realistic as well as retains important features in the medical context called sleep stage.
- In experiments, we confirm that our proposed model generates realistic EEG signals by showing the similarity between real and fake ones in both time and frequency domains as well as evaluating the accuracy of classifying sleep stages of generated signals. Furthermore, we also show that applications based on EEG data still work well without sacrificing performance using the imputed data by replacing missing signals with synthesized ones.

We have evaluated our generative model mainly focused on EEG signals with sleep stage labels in this work, but the model can be easily extended to learn and generate any time series data such as electrocardiogram (ECG) signals with types of arrhythmia for heart disease diagnosis.

II. RELATED WORK

A. IMPUTATION OF MISSING VALUES

Missing values in a dataset largely can be tackled with two strategies; the inherent consideration of missing values by developing robust models or algorithm which is not affected by missing values and the explicit modification of dataset by imputation or deletion to obtain a complete dataset without missing values. Deletion, which simply removes all records or even columns which including any missing values, and simple statistical imputation, which replaces missing values with mean, median or the most frequent value, have been used traditionally [2]. However, deletions may lead to lose too much information in a dataset and simple imputations usually fail to produce plausible data that look realistic considering context. Moreover, the first strategy to develop a robust model handling missing values inherently has a limitation that we have to devise a separate technique for every application with the dataset.

Recent development in machine learning techniques has enabled us to replace missing values with realistically generated one. The imputation methods for time series in [15] and user-rated movie scores in [16] formulated their data as a user-item matrix and utilized matrix factorization to estimate the missing entries. As early recurrent neural networks had been used for imputing missing values in time series data in [17] and recent generative models such as auto encoder and generative adversarial network also exploited to deal with missing values; for example, both models are extended to guess missing entries in tabular datasets in [1], [18] and especially, GANs were adopted extensively for the application called *image inpainting*, which completes missing part of an image, in [10]–[13], [19]. In [20], an imputation technique for time-series sensor data was developed using the sequence-to-sequence network, which is suitable for discrete time-series data. Moreover, [21] and [22] acquire auto encoder and transformer to reconstruct missing regions in input frames respectively.

These state-of-the-art techniques have shown a remarkable performance, however, they require complete data sets for training. In fact, most of dataset obtained in real world inevitably suffered from missing values such as EEG signals collected from medical devices, seismic signals from distributed sensors and many other observation in nature. Image inpainting works well using GANs since we can obtain images without missing values without difficulties. These works also are adopting GAN architectures like as we are, but based on the assumption that complete instances are available for training.

B. GENERATIVE ADVERSARIAL NETWORK (GAN)

As mentioned before, we adopted GAN for generate the missing values in this paper. GAN is a framework introduced by [23]. It trains a generator G and a discriminator D together; G generally maps a noise signal to a fake sample which is indistinguishable from real images while D classifies

between real and fake samples. GANs have been approved to be effective in various problems and domains such as image generation [24], [25], image translation [26], [27], audio synthesis [28], [29], image resolution enhancement [30], [31], image de-noising [32], [33] and feature generation [34].

When certain labels for context such as categories for any purpose are available in training data set, we can vary GAN to exploit such information to generate more realistic instances using auxiliary classifier [35]. As auxiliary classifier learns to classify the additional label correctly, generator is trained to generate samples holding characteristics which look naturally when considering the labels. We also adopt the auxiliary classifier to generate realistic EEG signal considering sleep stages available in the data set. Moreover, as several optimization techniques such as Wasserstein GAN (WGAN) [36] and gradient penalty [37] have been developed due to its inborn instability in training [38], we also exploit these techniques in our model to stabilize training process of GAN.

C. MACHINE LEARNING FOR EEG SEQUENCE GENERATION

In medicine and healthcare, machine learning has also been widely adopted to complete the missing values [2], [39]. While these works had been based on traditional approaches such as PCA/ICA, multilayer perceptron, random forest and SVM, GAN is also being hired recently in many applications of the area. For example, it is used for generating synthetic EEG signals in [14], [40], and [41] in the purpose of data augmentation. Other traditional techniques, SMOTE [42] and ADASYN [43] which had been proposed to alleviate class imbalance, have also been exploited to enhance an automated classifier of EEG signals [44]. However, those techniques are not adequate to generate realistic EEG signals. Another variation of GAN was introduced in [45] to up-sample EEG signal with a higher resolution. In addition, [5] discovered that DCGAN outperforms the traditional methods to augment EEG signal such as geometric transform, auto encoder and variational auto encoder.

These works have shown to successfully generate EEG signal and be exploited for improving classification. However, they neither consider missing values of EEG sequence for training nor create realistic and practically useful samples exhibiting contextual features such as sleep stages.

In spite of recurrent neural network (RNN) based model is capable for handling sequences, running long short-term memory (LSTM) networks on a long sequence fails such as 480 points of a sequence [46]. Because a sequence of EEG signal is usually equal to and more than 3000 points, we do not consider RNN as well as LSTM based model.

III. SIG-GAN: GENERATIVE ADVERSARIAL NETWORKS FOR SIGNAL SEQUENCES

In this section, we define notations for describing data and models and then, introduce our proposed SIG-GAN, a GAN-based network for imputing missing signal sequence in EEG data.

A. DATA DESCRIPTION

Let $\mathbb{S} = \{S_1, \dots, S_n\}$ be a collection of n sequences of signal segments where S_i denotes a sequence $\{s_{i1}, \dots, s_{it_i}\}$ of t_i signal segments s_{ij} ($i \in [1, n]$ and $j \in [1, t_i]$). Each sequence of signal segments S_i contains EEG signals, (e.g., 3,000 times recorded signal during 30 seconds with 100 Hz sampling frequency), collected through a medical test called *polysomnography (PSG) study* using electronic devices, and has a label c_i which is one of 5 types of sleep stages annotated by medical doctors or technologist. The notation c_i is used interchangeably to denote the sleep stage name or indicate a one-hot encoded vector in this paper. Note that a signal segment in the dataset may be a segment with missing values as our assumption.

To tell the process of data collection in PSG test briefly, it is performed overnight with a patient while the body functions of the subject such as brain activity (EEG), eye movement (EOG) and heart rhythm (ECG) are continuously recorded, i.e., those signals represent the electrical activity for each organ. Furthermore, as the technicians monitor the subject, they periodically score a – usually, 30 seconds long – signal segment as one of 5 sleep stages, which represent the stages of sleep cycles including W, N1, N2, N3 and R, following scoring manuals such as the American Academy of Sleep Medicine (AASM) [8]. Accordingly, each sequence S_i corresponds to a sequence of 30-second long signal segments collected through a night from a patient.

B. MOTIVATION AND PROBLEM DEFINITION

In a PSG test, occasionally recording failure can be caused by various reason such as the malfunction of the electrodes. Since the recoding error can last long time until its cause is corrected by technicians, the failure may result in a long sequence of missing signals over several segments. According to the sleep stage scoring manual in [8], a practitioner needs to consider the context through the precedent and following signal segments to determine sleep stages of given segments. For example, let a precedent sequence S_1 of 30 seconds signal segments is annotated as stage N1 and it contains K complex which is a strong evidence of stage N2 at the last 10 seconds. Then, the following sequence S_2 is scored as stage N2 unless there is evidence of shifting to another sleep stage [8]. As stated by the manual, we assume the contextual information can be preserved during a sequence just after the precedent one at least. Therefore, we consider the case that a sequence (30 seconds long) can be lost given a precedent one. Furthermore, computer aided diagnosis based on EEG signals also depends on the context of sequences for decision making. Thus, such missing segments may cause critical failure in diagnosis.

Hence, we suggest a generative method based on GAN for imputing missing signal segments which creates fake EEG signal segmentations that look natural as well as preserve contextual information like sleep stages. Fig. 1 illustrates the architecture of our proposed network. To trace the context

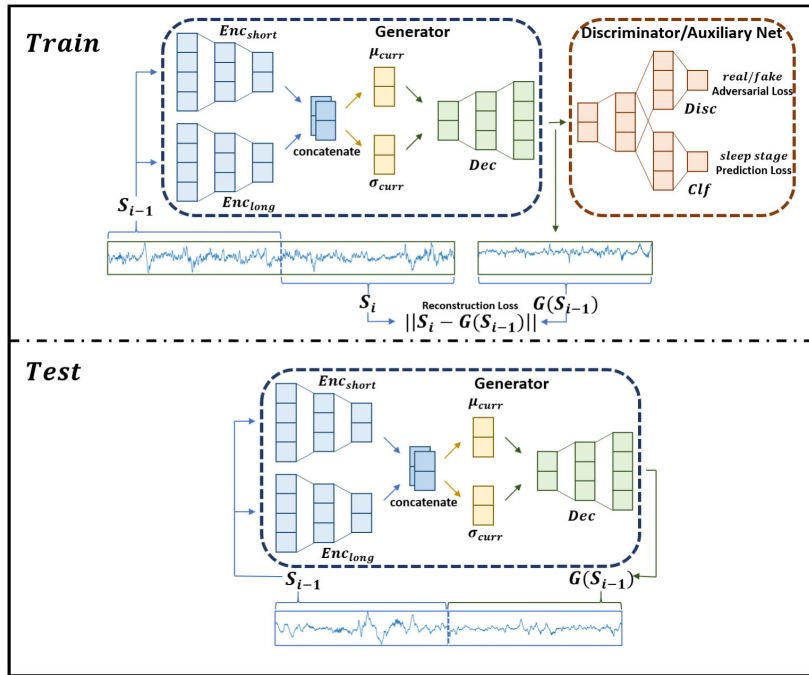


FIGURE 1. Overview of proposed model which contains training and test phase.

changing along the EEG signal segmentation, we adopt a generator G in the manner of auto-encoder. It takes a signal segment as input and generate a segment which can be expected to follow the next. The discriminator determines if the input segmentation is fake or real. The auxiliary classifier C infers the sleep stage of a given signal segmentation as AC-GAN does in [35], which is proven that it stabilizes training well so that the output of G follow the real input distribution. Naturally, C and D share the convolutional layers since they should utilize common local features for their own decisions in each downstream network.

C. FORMULATION FOR TRAINING PHASE

For training, we define three types of losses and select training samples for each loss as follows:

- **Adversarial loss:** It leads G to output a realistic fake signal segment given a preceding segment as input while D to distinguish between real and fake segments. Computing this loss requires a single signal segment and the training set S_i of segments sampled from \mathbb{S} is referred to as S_{adv} .
- **Reconstruction loss:** This is for fitting G to imitate the following signal segments as much as possible. To calculate the loss, we sample a set S_{rec} of pairs $\langle S_{i-1}, S_i \rangle$ of adjacent two non-missing signal segments from \mathbb{S} .
- **Prediction loss:** It enables G to be aware of the contextual information of sleep stage c_i as well as shepherds C and D to catch the real data distribution stably. The training datasets for computing the loss are split into two cases: $S_{pred,0} = \{\langle S_{i-1}, c_i \rangle\}$ when computing the

loss and learn the parameters for G with a fake segment $G(S_{i-1})$ which estimates S_i , and $S_{pred,1} = \{\langle S_i, c_i \rangle\}$ for training D with a real non-missing segment S_i .

1) ADVERSARIAL LOSS

To compute the adversarial loss of G and D , the training set S_{adv} of segment samples $S_i \in \mathbb{S}$ is utilized. By adopting the adversarial loss of WGAN [36], we can formulate its corresponding optimization problem as below

$$\min_{\theta_G} \max_{\theta_D} \sum_{S_i \in S_{adv}} D_{\theta_D}(S_i) - \sum_{S_{i-1} \in S_{adv}} D_{\theta_D}(G_{\theta_G}(S_{i-1})), \quad (1)$$

where θ_G and θ_D are trainable parameters of generator G and discriminator D respectively. While D is trained to tell the observed segment S_i as real and forged segment $G(S_{i-1})$ as fake, G becomes to output $G(S_{i-1})$ which deceives D to answer it is real by minimizing the second term in Eqn. (1). Furthermore, since adversarial loss suffers from unstable training [37], we add the gradient penalty loss $-\mathbb{E}_{\hat{S} \sim \mathbb{P}(\hat{S})} [(\|\nabla D_{\theta_D}(\hat{S})\|_2 - 1)^2]$ into the objective in Eqn. (1), where \hat{S} is a sampled segment from linear interpolated distribution $\mathbb{P}(\hat{S})$ between the real and generated data [36].

2) RECONSTRUCTION LOSS

To obtain G which imitates the next signal segments given a sample segment, we impose the reconstruction loss which is defined as L_1 distance between a sample segment S_i in S_{rec} and $G(S_{i-1})$. Its related optimization can be shown

as following

$$\min_{\theta_G} \sum_{\langle S_{i-1}, S_i \rangle \in S_{rec}} |S_i - G_{\theta_G}(S_{i-1})|_1. \quad (2)$$

Moreover, we inject reparameterization trick as suggested in [47]. Therefore, our model maps the distribution of signal generation into Gaussian distribution with mean μ_{curr} and variance σ_{curr} as shown in Fig. 1. We omit the part of reparameterization trick in Eq.2 for readability.

3) PREDICTION LOSS

To achieve our goal that G outputs a fake signal whose sleep stage is correctly recognized, we exploit the auxiliary classifier C as [35]. With each sample S_{i-1} from a sample set $S_{pred,0} = \{\langle S_{i-1}, c_i \rangle\}$, we define the prediction loss so that G learns to generate $G(S_{i-1})$ whose desired sleep stage is c_i as follows

$$\min_{\theta_G} \sum_{\langle S_{i-1}, c_i \rangle \in S_{pred,0}} \mathcal{L}(c_i, C(G_{\theta_G}(S_{i-1}))) \quad (3)$$

where $C(G(S_{i-1}))$ is the sleep stage predicted by C with the input segment $G(S_{i-1})$, and $\mathcal{L}(\cdot, \cdot)$ indicates the cross entropy between two distributions. Furthermore, the prediction loss is also utilized for training C with samples $S_{pred,1} = \{\langle S_i, c_i \rangle\}$ to predict the correct sleep stage with a real signals S_i as the following:

$$\min_{\theta_C} \sum_{\langle S_i, c_i \rangle \in S_{pred,1}} \mathcal{L}(c_i, C_{\theta_C}(S_i)) \quad (4)$$

where θ_C is the learnable parameters for C .

4) OVERALL OBJECTIVE

Our full optimization problem is

$$\min_{\theta_G, \theta_C} \max_{\theta_D} \sum_{S_i \in S_{adv}} D_{\theta_D}(S_i) - \sum_{S_{i-1} \in S_{adv}} D_{\theta_D}(G_{\theta_G}(S_{i-1})) \quad (5)$$

$$- \lambda_1 \mathbb{E}_{\hat{S} \sim \mathbb{P}(\hat{S})} \left[\left(\left| \nabla D_{\theta_D}(\hat{S}) \right|_2 - 1 \right)^2 \right] \quad (6)$$

$$+ \lambda_2 \sum_{\langle S_{i-1}, S_i \rangle \in S_{rec}} |S_i - G_{\theta_G}(S_{i-1})|_1 \quad (7)$$

$$+ \lambda_3 \left(\sum_{\langle S_{i-1}, c_i \rangle \in S_{pred,0}} \mathcal{L}(c_i, C_{\theta_C}(G_{\theta_G}(S_{i-1}))) \quad (8)$$

$$+ \sum_{\langle S_i, c_i \rangle \in S_{pred,1}} \mathcal{L}(c_i, C_{\theta_C}(S_i)) \right) \quad (9)$$

where λ_1 , λ_2 and λ_3 are the hyper-parameters to control the relative importance of gradient penalty, the reconstruction and prediction losses respectively.

D. FORMULATION FOR TESTING PHASE

With missing segment S_i whose preceding one S_{i-1} is present in \mathbb{S} , $G(S_{i-1})$ simply is used to impute the missing segment. In a real application to use our method, however, detecting missing segments from a sequence of signals is another issue. Fortunately, we can simply utilize the discriminator D to find

the missing segments. In our evaluation, we find that D often fails to detect missing intervals if the signals definitely do not look like EEG at all, for instance, simple uniform values or random values. Thus, we additionally trained D to detect such cases as missing segments with synthetically generated non-EEG signals.

IV. IMPLEMENTATION DETAILS

We implemented SIG-GAN using the machine learning framework TensorFlow [48].

A. NETWORK ARCHITECTURE

For the encoder of generator G , we borrow the architecture of DeepSleepNet [49] which employs two sequences of convolutional layers in parallel. As shown in Fig. 1, the encoder of G takes the input signal through two different 1-dimensional convolutional neural networks Enc_{short} and Enc_{long} whose filter sizes are 11 and 101 respectively. This architecture aims to capture the features that appear with high and low frequencies adaptively. Each segment of signal involves 3,000 EEG signals since the sequence is split by every 30 seconds where the data was sampled at 100Hz.

The detailed structure of SIG-GAN is shown on Table 1 and Table 2. A signal segment with 3,000 dimensional vector is fed into the encoder and we set same padding for the dimension of time-axis to keep 3,000 dimensionality while the channel-axis is decreased from 32, 16 and 8 in both Enc_{short} and Enc_{long} , i.e., the encoder has 64, 32 and 16 channels at each layer. The outputs of each encoder Enc_{short} and Enc_{long} are then concatenated along channel-axis. After that, we exploit reparameterization trick which is proposed by [47]. Then, we put two layers of transpose CNNs for the decoder.

TABLE 1. Network architecture of generator.

Signal Input, shape=(Batch size, 3000, 1)	
Conv1D (32, 11, 1), BN, ReLU	Conv1D (32, 101, 1), BN, ReLU
Conv1D (16, 11, 1), BN, ReLU	Conv1D (16, 101, 1), BN, ReLU
Conv1D (8, 11, 1), BN, ReLU	Conv1D (8, 101, 1), BN, ReLU
Concatenation	
Conv1D (8, 31, 1)	Conv1D (8, 31, 1)
Reparameterization	
Conv1DTrans (8, 31, 1), BN, Leaky ReLU	
Conv1DTrans (16, 31, 1), BN, Leaky ReLU	
Conv1DTrans (1, 31, 1)	

TABLE 2. Network architecture of discriminator/auxiliary net.

Signal Input, shape=(Batch size, 3000, 1)	
Conv1D (8, 31, 2), BN, Leaky ReLU	
Conv1D (16, 31, 2), BN, Leaky ReLU	
Conv1D (32, 31, 2), BN, Leaky ReLU	
Classifier	Discriminator
Flatten	Flatten
Dense (5), Softmax	Dense (1), Sigmoid
Output, shape=(Batch size, 5)	Output, shape=(Batch size, 1)

The discriminator D and auxiliary network C are simply stacks of convolutional layers. They share the first three layers

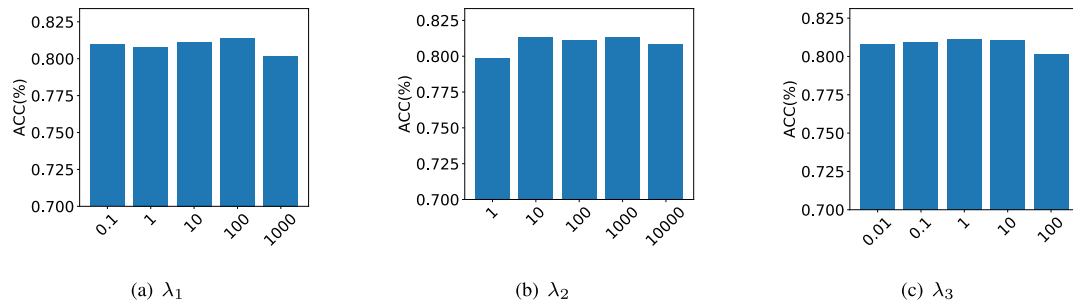


FIGURE 2. Classification accuracy with generated EEG signals with varying hyperparameters λ_1 , λ_2 and λ_3 with isolated other variables.

whose channel-axis are varied from 8 to 32. Then, D and C consist of fully-connected layer with a single and five output nodes respectively. Remind that D is for judging a segment real or fake and C classifies a segment into the five sleep stages. We leverage ReLU activation function [50] in the encoder to allow the model to learn fast, and we adopt LeakyReLU activation function [51] to force the decoder to generate high quality of the signal [24]. Moreover, we adopt batch normalization [52] with every layer to relieve the problem of poor initialization [24].

The numbers in parentheses in Table 1 and Table 2 are the number of filters, filter size and strides, e.g., (32, 11, 1) in the second row means that the layer is structured with 32 filters, 11 sized filter and 1 stride.

V. EXPERIMENTS

In this section, we empirically evaluate the performance of our proposed networks. We implement all deep neural networks using TensorFlow 2 on python 3.7. All experiments reported in this section are performed on the machines with Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz and 128GB of main memory running Ubuntu 16 OS. We also utilize a single GPU card NVIDIA GeForce GTX 1080 Ti equipped with 11GB of memory.

A. TRAINING DETAILS

We utilize Adam optimizer [53] and set the batch size and learning rate to 16 and 0.0001 respectively. We empirically select the weights for gradient penalty, reconstruction and prediction losses in Eqn. 5 as: $\lambda_1 = 10$, $\lambda_2 = 100$ and $\lambda_3 = 1$. Fig. 2 shows the performance evaluation with varying the loss weights; we tested the sleep stage classification with DeepSleepNet [49] using a dataset with 12% missing values. The graph shows the accuracy of classification with varying λ_1 from 0.1 to 1, 000, λ_2 from 1 to 10, 000 and λ_3 from 0.01 to 100 respectively. The performance does not differ much with varying the weights and we determined the default setting accordingly.

B. IMPLEMENTED ALGORITHMS

For comparative performance evaluation, we implemented three models that impute missing data by generating synthetic EEG signals as follows:

- **RANDOM:** This method imputes missing signals with randomly sampled signals whose values are between -1 and 1 . There are three strategies for sampling signals: i) sampling a constant value repeatedly, ii) sampling with a linear function, and iii) independently sampling random value following a uniform distribution. To sample a constant value, it selects a random number in range $[-1, 1]$, and replace all missing signals with the selected value. Linear signal sampling is to substitute the missing part with a line whose slope and intercept are randomly determined. Finally, the last method samples random values in range of $[-1, 1]$ following a uniform distribution independently and identically as many times as the number of missing signals. In our experiments, we tested all three strategies for evaluation but they show similar performance, and thus provide the result by using the last method for RANDOM.
- **EEGGAN:** It is a GAN-based model presented in [14] to synthetically generate EEG signals. The model of EEGGAN adopts the progressive GAN [54] for its generator based on the vanilla GAN model in [23] which does not exploit auxiliary information such as sleep stages. Thus, it does not utilize the sleep stage labels that are available in our training datasets. EEGGAN consists of 391,506 parameters in the generator and 303,461 in the discriminator while it takes 0.14 G FLOPs for the forward computation.
- **SIGGAN:** This represents our proposed model described in Section III. SIGGAN is built with 90,345 and 58,462 weight parameters in the generator and discriminator respectively, which requires much smaller memory space than EEGGAN. For the time complexity, it costs 0.54 G FLOPs.

1) EVALUATION TOOLS FOR GENERATED EEG SIGNALS

Note that the purpose of our technique is to replace missing EEG signals with synthetically generated ones so that medical software or devices dealing with EEG signals operate normally without sacrificing performance much. Thus, in our experiments, we assume automatic sleep stage scoring for such application and utilize two deep learning-based classifiers, which are DeepSleepNet [49] and SleepEEGNet [55]. These classifiers and the GAN-based EEG signal generators,

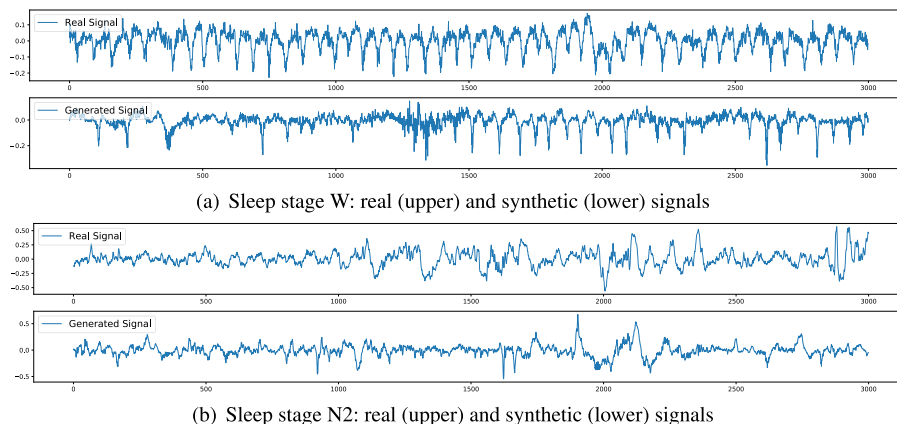


FIGURE 3. Time-frequency representation visualization of real and generated signals.

SIGGAN and EEGGAN, are trained separately and their synthetic signals are input to the classifiers to test if they still work well. We implement both classifiers in TensorFlow with hyperparameters presented in each paper.

- **DeepSleepNet [49]:** It is a classifier that determines the sleep stage with 30-second long EEG signal. It includes two sequences of convolutional layers with different-size filters, similar to our SIG-GAN model, for feature recognition. To classify by considering preceding and following signals, it also adopts the bi-directional long short-term memory (LSTM) network.
- **SleepEEGNet [55]:** This is another classifier that scores sleep stages, whose architecture is similar to that of DeepSleepNet, but it adopts sequential encoder-decoder structures using bi-directional LSTM and employs attention mechanism as well.

C. DATASET

For the training and test dataset, we downloaded a publicly available Sleep-EDF database [56], which includes 153 recordings obtained from 44 healthy people and 22 patients who had mild difficulty falling asleep. Each recording has as long as about 8 hours and sampled with 100 Hz frequency. Furthermore, the sleep stage of every 30-second long segment is scored manually by well-trained practitioners according to R&K rule [57] and AASM [8] which categorizes a segment into 5 classes (W, N1, N2, N3, REM). For preprocessing, we normalize the data to be ranged in [-1, 1].

In training phase, since our SIG-GAN takes two segments each of which is sampled for 30 seconds as inputs, we sampled 6,000 pairs of two segments from the downloaded Sleep-EDF database. For validation in the training phase, we utilize Monte Carlo cross-validation [58]; the data is split into training, validation and test sets, which are 50%, 25% and 25% respectively. Moreover, in our training set, the sleep stage ratio of segments in W, N1, N2, N3 and REM are 21%, 6%, 41%, 14% and 17% respectively. To mitigate the

problem of suffering from class imbalance, we oversampled minor classes to balance their ratio as other works handling EEG data did [49], [59], [60]. For all performance evaluation, we repeated Monte Carlo cross-validation 10 times and average the quality measures to show the performance.

D. REPRODUCIBILITY OF EEG SIGNAL FEATURES

We first test if the implemented EEG signal generators reproduce the realistic ones similarly to the real EEG signals. For case study, we plot some selected real and synthetic time series signals as well as spectrograms showing them in the time-frequency domain. To figure out that the generated signals show similar distributions in their frequencies, we examine the energy density over frequency by using band pass filters.

1) SIGNALS IN TIME SERIES

In Fig. 3, we plotted some randomly selected samples for both targeted and generated EEG signals. To see if we can obtain the signals using SIGGAN showing the characteristics of sleep stages appropriately, we show real and synthetic signals labelled as W and N2. To select the 30-second segments of SIGGAN for W and N2, we sampled pairs of segments classified as the corresponding sleep stage by DeepSleepNet. Due to the limit of space, we show only two pairs of samples but we can see in the figures that the generated signal is quite similar to the real one and they also mimic distinguishable features in EEG signals according to the sleep stages.

2) QUALITY AND DIVERSITY

Evaluation of the performance of generative models is open problem. To evaluate quality and diversity of generated signal, we measure inception score (IS) [61] and Frchet inception distance (FID) [62]. IS measures the quality and diversity based on computing KL-divergence between the class distribution given generated sample and marginal class distribution using the pre-trained inception networks [63]. IS is formulated as $IS = \exp \left[\mathbb{E}_x D_{KL} (p(y|x} || p(y)) \right]$ where

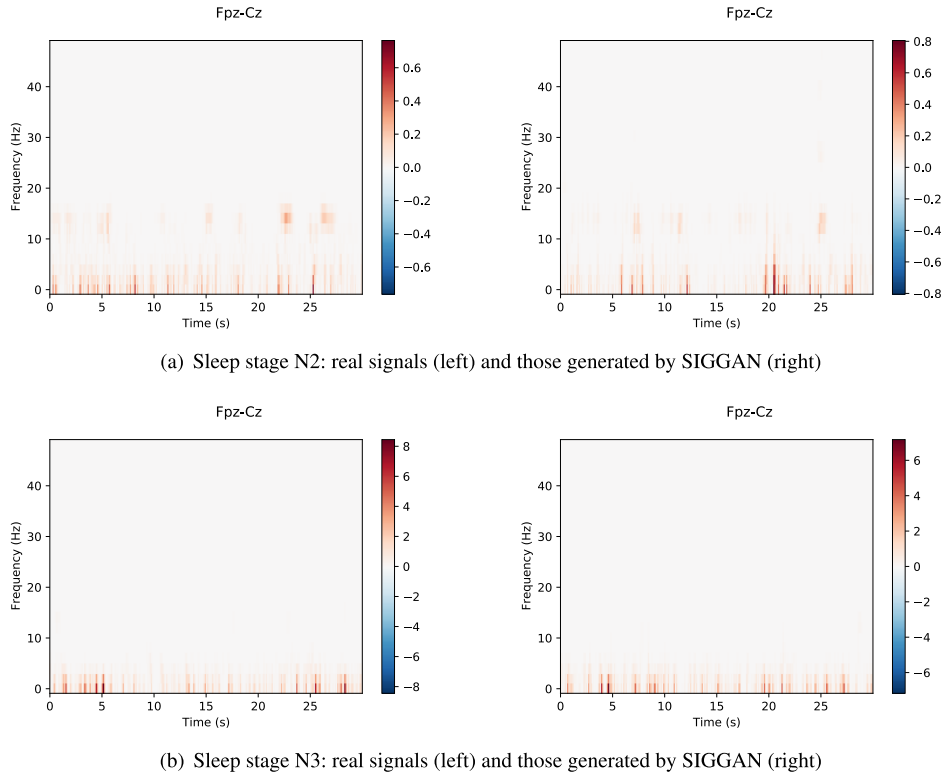


FIGURE 4. Time-frequency representation of real and synthetic signals.

TABLE 3. Quality and diversity.

Type of Signal	IS	FID
Real Signal	2.53	0.00
RANDOM	1.85	5763.02
EEGGAN	1.49	73.08
Generated by SIGGAN	2.02	39.53

x is generated sample by a GAN model and y is predicted class by the pre-trained inception model. Since we cannot directly exploit pre-trained inception networks using ImageNet dataset, we train only sub-part of DeepSleepNet which consists of convolutional neural networks instead. Similarly, various GAN based works apply well known classifiers rather than inception networks to evaluate IS and FID for each domain such as audio synthesis [64], [65] and EEG signal generation [14]. We calculate and compare our IS with real signal, RANDOM and EEGGAN at second column in Table 3. At the first row, the score 2.53 indicates upper bound of IS. The results prove that our method is comparably higher than baselines. It indicates that generated signal by SIGGAN has better quality and diversity than baselines.

Although IS has been the first popular metric to evaluate GANs, it does not utilize any ground truth samples. Therefore, FID [62] has been proposed to capture the similarity of generated samples to real ones. FID embeds generated samples and real ones into feature space respectively using the pre-trained inception network. Then the embedded

features are estimated as continuous multivariate Gaussian. FID measures distance between those two Gaussian distributions using formula such as $FID(r, x) = \|\mu_r - \mu_x\|_2^2 + Tr(\Sigma_r + \Sigma_x - 2(\Sigma_r \Sigma_x)^{\frac{1}{2}})$ where (μ_r, Σ_r) and (μ_x, Σ_x) are the mean and covariance of the samples of real ones r and generated ones x respectively. In Table 3, we produce FID between real signal and others. Since real signal has zero distance between itself, FID of real signal is 0 and it is the lower bound of FID. The results show that generated signal using ours is the closest distance whereas FID of RANDOM and EEGGAN are way more higher. It proves that the suggested model generates the most similar signal to real signal.

3) TIME-FREQUENCY REPRESENTATION

By simply plotting the EEG signals in time domain, it is hard to find characteristics of them in terms of frequency. To represent signals over time and frequency simultaneously, we show time-frequency representation (TFR) of the selected ones in Fig. 4. In Fig. 4(a) and Fig. 4(b), we plotted the real (left) and generated (right) signals of sleep stage N2 and N3 respectively. The TFRs also show that the generated signals closely resemble the real EEG signals.

In addition, we also select two segments by EEGGAN randomly and show their TFRs in Fig. 5. Note that because EEGGAN does not take any auxiliary inputs such as sleep stages, we cannot choose the segments with a specific sleep stage. We discover that almost TFRs of the signals by EEGGAN show the similar patterns shown in Fig. 5 and they

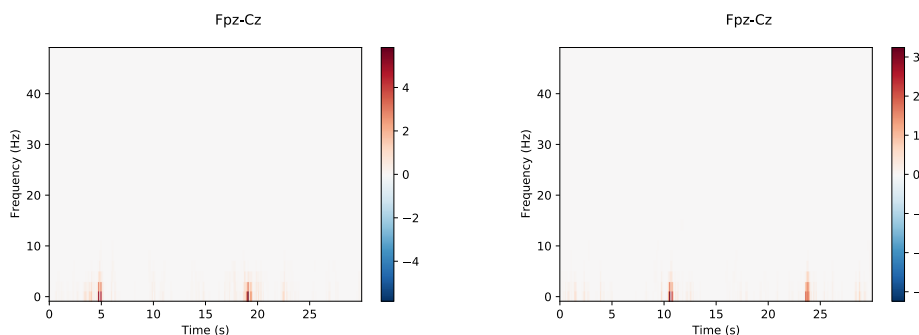


FIGURE 5. Examples of two segments generated by EEGAN.

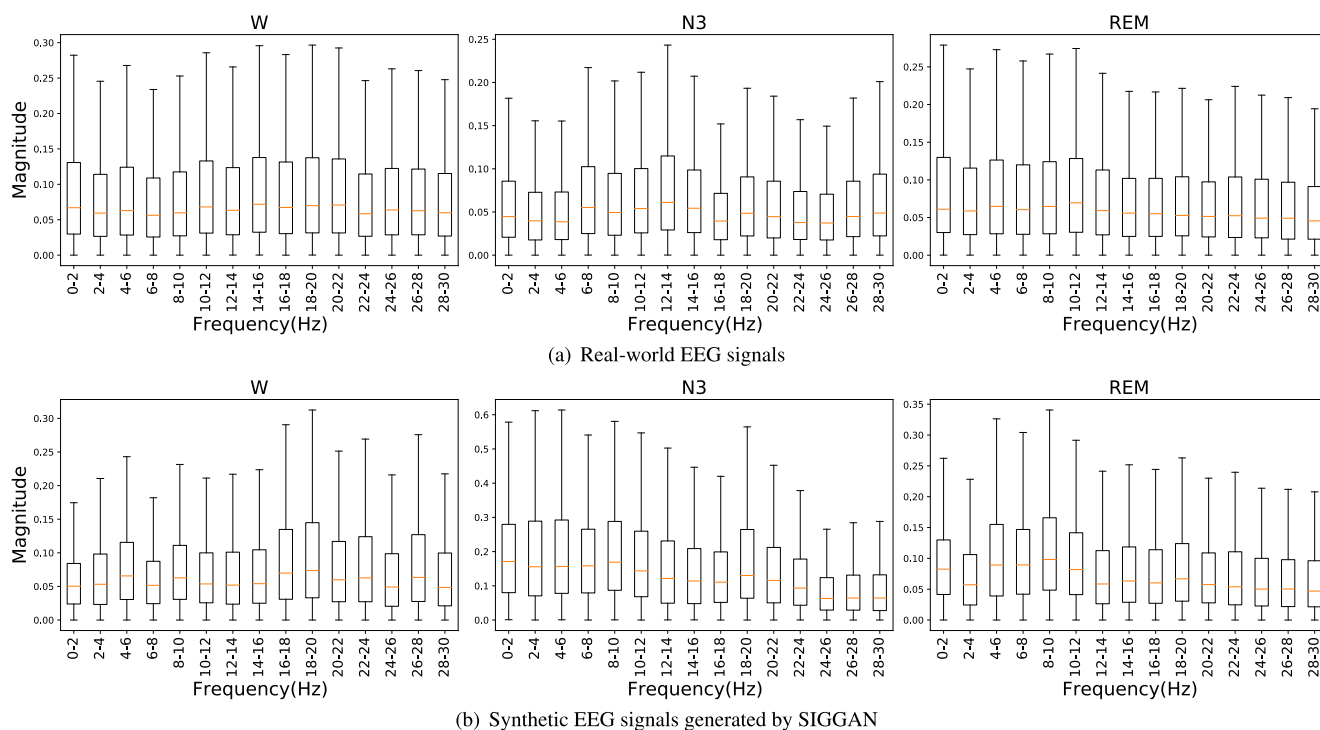


FIGURE 6. Magnitude distribution over frequency with real signals and those generated by SIGGAN.

do not preserve the features of real EEG signals as shown in Fig. 4.

4) BAND PASS FILTER

In Fig. 6, we plotted the magnitude distribution over frequency domain with the real and synthetic signals generated by SIGGAN. To see if the generated ones reproduce the contextual features of EEG signals shown in frequency domain, we calculated the distribution with three sleep stages W, N3 and REM separately. For the analysis, we utilized 10 band-pass filters whose bands of frequencies are ranged from 0Hz to 30Hz by interval of 2Hz. According to AASM sleep stage scoring manual in [8] and the research on brain waves about sleep stages [66], it is known that EEG signals labelled W show relatively high magnitude in high frequencies including alpha (8–12Hz) and beta (12–30Hz) waves while those are recorded in N3 and REM. It also states that when people fall

in deep sleep, the signals in low frequencies become stronger than other frequencies. The graphs in the first column of Fig. 6 show that the average and maximum magnitude in each band has a peak in high frequencies at about 16–22Hz in both the real and synthetic EEG signals similarly. Furthermore, as known as that REM shows typically large amounts of theta (4–8Hz), the graphs in the last column of Fig. 6 also confirm that the signals generated by our SIGGAN look very realistic similarly to the real world EEG signals. Most of distribution with fake signals generated by EEGAN, however, shows similar one to the graph in Fig. 7 without varying in sleep stages.

E. EVALUATION BY SLEEP STAGE SCORING

Our SIGGAN model aims to impute missing EEG signals with the realistic synthetic data generated based on the preceding signals so that any devices and software utilizing

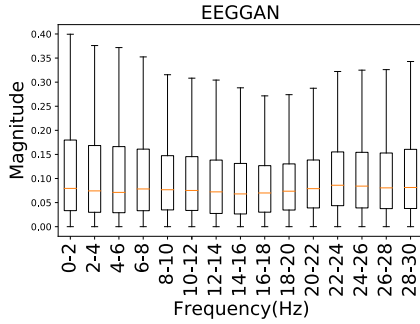


FIGURE 7. Magnitude distribution over frequency with signals generated by EEGGAN.

TABLE 4. Classification accuracy with individual segments.

Type of Signal	Sleep stage accuracy (%) (std)
Real Signal	82.85 (5.09)
Generated by SIGGAN	65.67 (6.66)
Generated without adversarial loss	45.05 (2.84)
Generated without auxiliary classifier	65.12 (7.96)
Generated without gp-loss	62.61 (19.00)
Generated without prediction loss	24.29 (12.47)
Generated without reconstruction loss	71.52 (2.97)

EEG signals measured by polysomnography (PSG) study. Thus, it is desirable that the segment generated to impute the missing part preserves the correct sleep stage score if it was correctly measured without failure. In our experiment, we test if the generated signals are correctly classified to preserve the quality of data in EEG applications.

1) CLASSIFICATION WITH INDIVIDUAL SEGMENTS

To evaluate the performance in EEG applications, we sampled a test dataset by sampling 5,762 pairs of adjacent 30-second segmentations as ground truth, and tested if the applications such as *DeepSleepNet* and *SleepEEGNet* can identify the sleep stages of the generated EEG segments, which are output by SIGGAN based on the first 30-second signals, correctly as the same labels as the ground truth segments have. Note that we selected the test dataset to be evenly distributed over sleep stages.

Overall, *DeepSleepNet* classifies the signals generated by SIGGAN into its correct sleep stage scores with 65.67% of accuracy in average while it is 82.85% for real signals as shown in Table 4. In Fig. 8, we depict the confusion matrices that *DeepSleepNet* produces with the test dataset. In each matrix, a row represents the ratio of each sleep stage which are classified into each sleep stages by *DeepSleepNet*. The results show that *DeepSleepNet* achieves about 80% of accuracy with the ground truth dataset as shown in Fig. 8(a).

Fig. 8(a) indicates us that since the real EEG signals of N1 and R stages look similar (e.g., they typically have a large amount of low frequency such as alpha waves), *DeepSleepNet* often confuses them as demonstrated in the second and last rows. Similarly, for the synthetic EEG signals by SIGGAN, the classification tends to be wrong with the stage R as shown

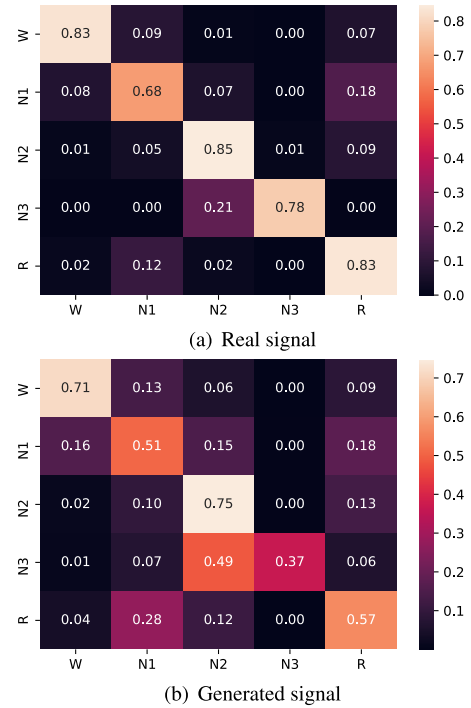


FIGURE 8. A comparison of confusion matrices between real signal and generated signal using a deep learning classifier trained by real signal.

in Fig. 8(b), but it still obtains reasonably high accuracy for the signals of N1 and R. For sleep stages N2 and N3 which are characterized by slow frequency and high amplitude such as delta waves, we can find that the classifier shows quite low accuracy with real EEG signals, and hence becomes to confuse the segments of N2 and N3 with the generated signals as shown in the third row of the confusion matrix in Fig. 8(b). The reason why the missing part of N3 stage is largely misclassified into N2 is that in the original training data used for fitting our GAN model, the signals of N3, which represents the deepest stage of sleep, was not long enough. Thus, SIGGAN tends to generate signals of N2 instead of N3.

2) CLASSIFICATION WITH EEG SIGNAL SEQUENCES

With varying the ratio of missing data in the signal sequences (missing from 0% to 50%), we tested the accuracy of sleep stage scoring by *DeepSleepNet* and *SleepEEGNet* and plotted the results in Fig. 9. We evaluated the performance with three EEG signal generators RANDOM, EEGGAN and SIGGAN. For RANDOM and EEGGAN, we cannot force them to consider the preceding signals for imputing the missing data with considering the context, the missing segments were replaced with the one generated by them independently and individually. For SIGGAN, we generated segments for the missing ranges by providing the previous measured EEG signals. Since the segments are randomly dropped each the selected probability when we create the test dataset, we simply generate the next segment using the previous one

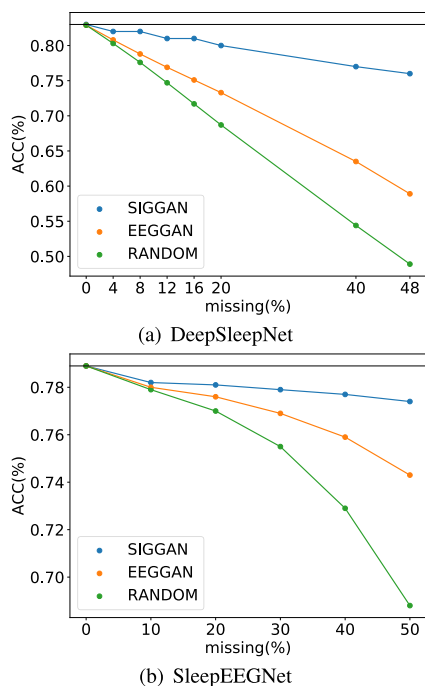


FIGURE 9. Classification with EEG signals imputed by SIGGAN, EEGAN and RANDOM; The black line indicates the best accuracy we can achieve with the ground truth.

which is also an output of SIGGAN if missing segments are located continuously. Both graphs confirm that SIGGAN outperforms significantly the other methods with all ranges of missing data ratios; *DeepSleepNet* does not suffer from performance degradation much using ours, even we remove the segments by 48% where it shows 75.75% of accuracy. Note that the classifier achieves the accuracy of 82.94% with the ground truth dataset. For the other application *SleepEEGNet*, the accuracy of the algorithm with imputed signals using SIGGAN is only decreased by 1.47% for 50% of missing data ratio.

3) DISCOVERING EFFECTS OF EACH COMPONENT

We have designed experiments to explore effect of each component in our model. Hence, we train SIGGAN without each component and evaluate the performance as we have done in the previous section **Classification with individual segments**.

We first train SIGGAN without adversarial loss which are GAN loss (Eqn. 5) and gradient penalty loss (Eqn. 6). In Table 4, the result shows that the model achieves only 45.05% accuracy without adversarial loss. Although the model still acquires reconstruction loss and prediction loss, it does not properly learn and generate signals. It means that GANs are the essential part of the model. Moreover, we investigate how the auxiliary classifier affect the model performance in Table 4. The results shows that our model with auxiliary classifier is better than without it. In addition, we reveal the contribution of each loss function in Table 4

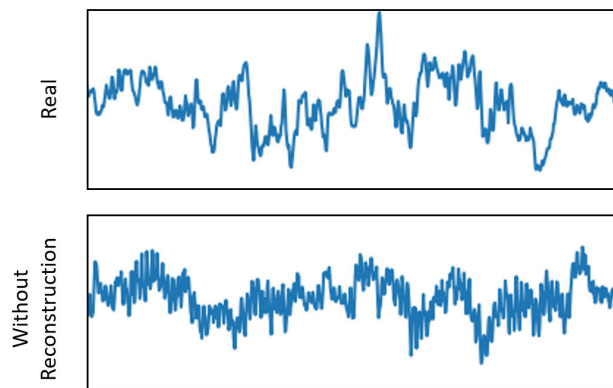


FIGURE 10. Comparison between target signal and without reconstruction loss. The first row shows the target signal. The second row represents generated signal without reconstruction loss. Without reconstruction loss, the generated signal includes high frequency noise.

such as gp-loss and prediction loss. Without each loss function, the generated signal does not work well with a classifier as proposed one. Without gp-loss, the accuracy is 62.61%, and without prediction loss, the accuracy is only 24.29%. It shows that for each component in the model and loss function are required to perform the best result. Moreover, we train the model without reconstruction loss and evaluate the performance. The accuracy is 71.52% as we can see in Table 4. It is the best classification performance. However, our goal is to generate realistic signal whereas the generated one without reconstruction loss is not as shown in Fig 10. The figure shows that generated signals contain high frequency noise. We compare distance between real (targeted) signal and generated signal using suggestion as well as generated one without reconstruction loss. The average L2 distance from ours is 6.53 and generated without reconstruction loss is 7.00. We obtain that the p-value between two distances is 0.0034 (<0.5) using t-test. Therefore, generated signal using ours is significantly similar to real signal than generated one without reconstruction loss. We analyze that the reconstruction loss boosts the model to generate signals to be realistic as target signals, however without the reconstruction loss, the model focuses on prediction loss. Consequently, the model produces high accuracy but unrealistic signals without reconstruction loss.

VI. CONCLUSION

In this paper, we developed SIG-GAN, a GAN-based deep neural network to impute missing data in the sequences of EEG signals. The proposed model was devised to acquire the context from precedent signals and create realistic signals using auxiliary labels such as sleep stages. In the experiments, we validate that the proposed model not only generates realistic EEG signal compared with real signal but also the produced signals reproduce sleep stages, which is an important characteristic of EEG signals, better than the recent GAN-based model. Using existing automatic sleep stage scoring

models, we demonstrate that the models still work correctly with the imputed dataset, and our model generates missing EEG signals realistically in terms of sleep stages.

REFERENCES

- [1] J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, J. Dy and A. Krause, Eds. Stockholm, Sweden: Stockholmssmässan, Jul. 2018, pp. 5689–5698.
- [2] J. F. Rodrigues, J.-L. Pepin, L. Goeuriot, and S. Amer-Yahia, "An extensive investigation of machine learning techniques for sleep apnea screening," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 2709–2716.
- [3] N. McCombe, S. Liu, X. Ding, G. Prasad, M. Bucholc, D. P. Finn, S. Todd, P. L. McClean, and K. Wong-Lin, "Practical strategies for extreme missing data imputation in dementia diagnosis," *MedRxiv*, vol. 7, no. 13, 2020, Art. no. 20146118.
- [4] T. Miettinen, K. Myllymaa, S. Westeren-Punnonen, J. Ahlberg, T. Hukkanen, J. Töyräs, R. Lappalainen, E. Mervaala, K. Sipilä, and S. Myllymaa, "Success rate and technical quality of home polysomnography with self-applicable electrode set in subjects with possible sleep bruxism," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 4, pp. 1124–1132, Jul. 2018.
- [5] K. Zhang, G. Xu, Z. Han, K. Ma, X. Zheng, L. Chen, N. Duan, and S. Zhang, "Data augmentation for motor imagery signal classification based on a hybrid neural network," *Sensors*, vol. 20, no. 16, p. 4485, Aug. 2020.
- [6] A. Bernier-Jean, W. Beaubien-Souigny, R. Goupil, F. Madore, F. Paquette, S. Troyanov, and J. Bouchard, "Diagnosis and outcomes of acute kidney injury using surrogate and imputation methods for missing preadmission creatinine values," *BMC Nephrol.*, vol. 18, no. 1, pp. 1–9, Dec. 2017.
- [7] A. Nekouie and M. H. Moattar, "Missing value imputation for breast cancer diagnosis data using tensor factorization improved by enhanced reduced adaptive particle swarm optimization," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 31, no. 3, pp. 287–294, Jul. 2019.
- [8] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, C. Marcus, and B. V. Vaughn, "The AASM manual for the scoring of sleep and associated events," *Amer. Acad. Sleep Med.*, vol. 176, p. 2012, Oct. 2012.
- [9] M. Saini and U. Satija, "An effective and robust framework for ocular artifact removal from single-channel EEG signal based on variational mode decomposition," *IEEE Sensors J.*, vol. 20, no. 1, pp. 369–376, Jan. 2020.
- [10] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4470–4479.
- [11] D. Lee, J. Kim, W.-J. Moon, and J. C. Ye, "CollaGAN: Collaborative GAN for missing image data imputation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2487–2496.
- [12] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6882–6890.
- [13] H. Liu, B. Jiang, Y. Xiao, and C. Yang, "Coherent semantic attention for image inpainting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4169–4178.
- [14] K. G. Hartmann, R. T. Schirmmeister, and T. Ball, "EEG-GAN: Generative adversarial networks for electroencephalographic (EEG) brain signals," 2018, *arXiv:1806.01875*.
- [15] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. 2016, pp. 847–855.
- [16] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *J. Mach. Learn. Res.*, vol. 11, pp. 2287–2322, Jan. 2010.
- [17] Y. Bengio and F. Gingras, "Recurrent neural networks for missing or asynchronous data," in *Proc. Adv. Neural Inf. Process. Syst.*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds., 1996, pp. 395–401.
- [18] L. Gondara and K. Wang, "Multiple imputation using deep denoising autoencoders," 2017, *arXiv:1705.02737*.
- [19] A. Dapogny, M. Cord, and P. Pérez, "The missing data encoder: Cross-channel image completion with hide-and-seek adversarial network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, 2020, pp. 10688–10695.
- [20] J. J. Dabrowski and A. Rahman, "Sequence-to-sequence imputation of missing sensor data," in *Proc. Australas. Joint Conf. Artif. Intell.*, 2019, pp. 265–276.
- [21] A. Comas, C. Zhang, Z. Feric, O. Camps, and R. Yu, "Learning disentangled representations of videos with missing data," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3625–3635.
- [22] Y. Zeng, J. Fu, and H. Chao, "Learning joint spatial-temporal transformations for video inpainting," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 528–543.
- [23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [24] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.
- [25] S. Liu, T. Wang, D. Bau, J.-Y. Zhu, and A. Torralba, "Diverse image generation via self-conditioned GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14286–14295.
- [26] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.
- [27] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 319–345.
- [28] A. Marafioti, N. Perraudin, N. Holighaus, and P. Majdak, "Adversarial generation of time-frequency features with application in audio synthesis," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4352–4362.
- [29] K. Su, X. Liu, and E. Shlizerman, "Audeo: Audio generation for a silent performance video," 2020, *arXiv:2006.14348*.
- [30] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 63–79.
- [31] R. M. Umer, G. L. Foresti, and C. Micheloni, "Deep generative adversarial residual convolutional networks for real-world super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 438–439.
- [32] Q. Yang, P. Yan, Y. Zhang, H. Yu, Y. Shi, X. Mou, M. K. Kalra, Y. Zhang, L. Sun, and G. Wang, "Low-dose CT image denoising using a generative adversarial network with wasserstein distance and perceptual loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1348–1357, Jun. 2018.
- [33] Z. Hong, X. Fan, T. Jiang, and J. Feng, "End-to-end unpaired image denoising with conditional adversarial networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 4140–4149.
- [34] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5542–5551.
- [35] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.
- [36] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*.
- [37] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.
- [38] N. Kodali, J. D. Abernethy, J. Hays, and Z. Kira, "On convergence and stability of GANs," 2017, *arXiv:1705.07215*.
- [39] R. E. Rolón, I. E. Gareis, L. E. Di Persia, R. D. Spies, and H. L. Rufiner, "Complexity-based discrepancy measures applied to detection of apnea-hypopnea events," *Complexity*, vol. 2018, pp. 1–18, Aug. 2018.
- [40] D. Pascual, A. Aminifar, D. Atienza, P. Ryvlin, and R. Wattenhofer, "Synthetic epileptic brain activities using GANs," in *Proc. Mach. Learn. Health at Conf. Neural Inf. Process. Syst.*, 2019, pp. 1–8.
- [41] Y. Luo and B.-L. Lu, "EEG data augmentation for emotion recognition using a conditional Wasserstein GAN," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 2535–2538.
- [42] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.
- [43] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. World Congr. Comput. Intell.*, Jun. 2008, pp. 1322–1328.

- [44] H. T. Tor, C. P. Ooi, N. S. Lim-Ashworth, J. K. E. Wei, V. Jahmunah, S. L. Oh, U. R. Acharya, and D. S. S. Fung, "Automated detection of conduct disorder and attention deficit hyperactivity disorder using decomposition and nonlinear techniques with EEG signals," *Comput. Methods Programs Biomed.*, vol. 200, Mar. 2021, Art. no. 105941.
- [45] T.-J. Luo, Y. Fan, L. Chen, G. Guo, and C. Zhou, "EEG signal reconstruction using a generative adversarial network with wasserstein distance and temporal-spatial-frequency loss," *Frontiers Neuroinform.*, vol. 14, p. 15, Apr. 2020.
- [46] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," 2021, *arXiv:2012.07436*.
- [47] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
- [48] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and S. Ghemawat. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: <https://www.tensorflow.org/>
- [49] A. Supratak, H. Dong, C. Wu, and Y. Guo, "DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 11, pp. 1998–2008, Nov. 2017.
- [50] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 807–814.
- [51] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013, p. 3.
- [52] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.
- [53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [54] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.
- [55] S. Mousavi, F. Afghah, and U. R. Acharya, "SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach," 2019, *arXiv:1903.02108*.
- [56] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Obery, "Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 9, pp. 1185–1194, Sep. 2000.
- [57] T. Hori, Y. Sugita, and, "Proposed supplements and amendments to 'A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects' (the rechtschaffen & Kales (1968) standard)," *Psychiatry Clin. Neurosci.*, vol. 55, no. 3, pp. 305–310, 2001.
- [58] W. Dubitzky, M. Granzow, and D. P. Berrar, *Fundamentals of Data Mining in Genomics and Proteomics*. Boston, MA, USA: Springer, 2007.
- [59] O. Tsinalis, P. M. Matthews, and Y. Guo, "Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders," *Ann. Biomed. Eng.*, vol. 44, no. 5, pp. 1587–1597, 2016.
- [60] A. Sors, S. Bonnet, S. Mirek, L. Vercueil, and J.-F. Payen, "A convolutional neural network for sleep stage scoring from raw single-channel EEG," *Biomed. Signal Process. Control*, vol. 42, pp. 107–114, Apr. 2018.
- [61] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
- [62] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6629–6640.
- [63] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [64] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: <https://openreview.net/forum?id=ByMVTsR5KQ>
- [65] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," in *Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: <https://openreview.net/forum?id=H1xQVn09FX>
- [66] R. Armitage, "The distribution of EEG frequencies in REM and NREM sleep stages in healthy young adults," *Sleep*, vol. 18, no. 5, pp. 334–341, Jun. 1995.



WOONGHEE LEE received the B.S. degree from the Department of Industrial and Management Engineering, Hanyang University ERICA, in 2015, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering. His research interests include data mining and machine learning.



JAEOYOUNG LEE received the B.S. degree in computer science and engineering from Hanyang University ERICA, in 2020, where he is currently pursuing the M.S. degree with the Department of Computer Science and Engineering. His research interests include data mining and computer vision.



YOUNGHOON KIM received the B.S. degree in computer science and engineering and the Ph.D. degree from Seoul National University, in 2006 and 2013, respectively. He is currently an Associate Professor with Hanyang University ERICA, South Korea.