Denoising Time Series Data Using Asymmetric Generative Adversarial Networks

Sunil Gandhi[1(✉)], Tim Oates[1], Tinoosh Mohsenin[1], and David Hairston[2]

[1] University of Maryland, Baltimore County, Baltimore, USA
{sunilga1,oates,tinoosh}@umbc.edu
[2] U.S. Army Research Laboratory, Adelphi, USA
william.d.hairston4.civ@mail.mil

Abstract. Denoising data is a preprocessing step for several time series mining algorithms. This step is especially important if the noise in data originates from diverse sources. Consequently, it is commonly used in biomedical applications that use Electroencephalography (EEG) data. In EEG data noise can occur due to ocular, muscular and cardiac activities. In this paper, we explicitly learn to remove noise from time series data without assuming a prior distribution of noise. We propose an online, fully automated, end-to-end system for denoising time series data. Our model for denoising time series is trained using unpaired training corpora and does not need information about the source of the noise or how it is manifested in the time series. We propose a new architecture called *AsymmetricGAN* that uses a generative adversarial network for denoising time series data. To analyze our approach, we create a synthetic dataset that is easy to visualize and interpret. We also evaluate and show the effectiveness of our approach on an existing EEG dataset.

1 Introduction

Time series data mining is an important area of research and has applications in a variety of domains including healthcare, econometrics, and speech recognition. Consequently, a large number of methods for time series classification, clustering, anomaly detection and motif discovery have been proposed. Although these methods can handle some noise, they are not effective when noise originates from different sources and has diverse characteristics.

Consider, for example, a widely used method for time series featurization called Symbolic Aggregate approXimation (SAX) [10] that assumes time series are generated from a single normal distribution. As shown in [5] this assumption does not hold in several real life time series datasets. Other techniques assume noise comes from a Gaussian distribution and estimate the parameters of that distribution [13]. This assumption does not hold for data sources like Electroencephalography (EEG), where noise can have diverse characteristics and originate from different sources [15]. Hence, in this work, we focus on learning the characteristics of noise in EEG data and removing it as a preprocessing step.

D. Phung et al. (Eds.): PAKDD 2018, LNAI 10939, pp. 285–296, 2018.
https://doi.org/10.1007/978-3-319-93040-4_23

Electroencephalography (EEG) is a technique that records the electrical activity of the brain by placing electrodes on the scalp. As it is noninvasive and cost-effective, it is widely used in brain-computer interfaces (BCI), determining cognitive states, seizure detection, and monitoring neurological disorders [11,14]. Unfortunately, EEG data is often contaminated by different forms of noise called "artifacts". Artifacts are undesired signals in EEG data originating from sources other than brain activity. Artifacts can occur from diverse sources like ocular, muscular, and cardiac activities, or external sources like electrodes and line noise. The occurrence of external artifacts can be reduced by proper placement of electrodes, but it is impossible to avoid artifacts of biological origin. Not only do artifacts increase the chance of false alarms in seizure detection [12], they can also alter the shape of neurological events [16]. Therefore, artifact detection and removal is an important preprocessing step for EEG data.

Given the importance of artifact removal from EEG data, a large number of denoising techniques have been proposed in the neuroimaging literature. This includes techniques like artifact rejection that focus on detecting artifacts and removing segments where they are present. Although simple, artifact rejection methods can lead to excessive loss of information. Regression of denoised signals using a reference channel is another commonly used technique. Regression techniques need reference signals and cannot be trained using unpaired training data. Thus the approach is specific to EEG data and not broadly applicable to time series data. Another popular technique is to use independent component analysis (ICA) to decompose the signal into independent source signals and identify sources corresponding to noise. The identification of source signals has to be done manually or using supervised classifiers. This requires a human in the loop or additional annotation. Also, ICA has high computational complexity and large memory requirements, making it unsuitable for real-time applications.

We solve these problems by proposing an online, fully automated, end-to-end system for denoising time series trained using unpaired training corpora. An online and fully automated system makes it useful in real-time applications. Unlike [8], our system is trained end-to-end and has fewer hyperparameters to be optimized and is easy to deploy. Being able to train on unpaired training corpora allows our method to be useful in a wide variety of applications. For training of our network, we only need a set of clean signals and set of noisy signals. We do not need paired training data, i.e., we do not need clean versions of the noisy data. This is particularly useful for applications like artifact removal in EEG data as we cannot record clean versions of noisy EEG.

We create this method by leveraging recent advances in unpaired image to image translation [19] using generative adversarial networks. We modify an existing generative adversarial architecture [19] for denoising time series data. We analyze our approach on a synthetic dataset and examine the effectiveness of our approach in removing artifacts from EEG data. The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 defines the problem. Section 4 describes our system for learning a model for denoising time series from unpaired training data. In Sect. 5, we evaluate our model on synthetic and EEG data. Section 6 concludes and discusses future work.

2 Related Work

Time series data mining algorithms can broadly be classified as model-based and model-free. Model-based algorithms focus on creating features and models that are robust to noise in the data by assuming priors on the distribution of the noise. Model-free algorithms do not assume a prior on the distribution of noise and explicitly learn to remove noise as a preprocessing step [6]. Model-based algorithms often make assumptions about the distribution of noise in the data. For example, [13] assumes that noise comes from a Gaussian distribution and estimates the parameters of the distribution. This makes these techniques ineffective when noise has different characteristics and originates from diverse sources [6]. For such applications, model-free denoising is often used. In this work, we focus on model-free denoising because of its applicability for a wide range of applications.

Model-free denoising techniques are often used in biomedical applications for processing EEG or functional Magnetic Resonance Imaging (fMRI) data. A simple technique for denoising EEG data is artifact rejection that focuses on detecting artifacts and removing the segments of data where artifacts are present. Although simple, artifact rejection methods can lead to excessive loss of information [15]. In this work, we propose a method for denoising the time series instead of removing entire segments of noisy data.

Regression of a denoised signal using a reference channel like Electrocardiography (ECG), Electrooculography (EOG) or Electromyography (EMG) is a common approach to removing artifacts from EEG data [15]. These techniques fail if a reference signal is not available. The need for a reference signal makes these approaches EEG-specific. Wavelet transforms are commonly used to decompose a signal into a set of coefficients at various scales, which represent the similarity of the signal to the wavelet at that scale. The artifacts found by this method are often removed by thresholding the coefficients and reconstructing the signal from the filtered representation. Artifact removal based on the wavelet transform relies on the artifacts being decomposable in a wavelet basis. Thus, the mother wavelet, the shrinkage rule, and the threshold are important to the design of the noise removal method [2]. Also, these techniques process each channel separately and could miss important clues for removing the artifact.

Blind source separation methods that use Independent Component Analysis (ICA) are the most popular techniques for artifact removal from EEG data [3,15]. ICA is used to recover independent source signals called components and then the components corresponding to artifacts are identified. These components are either manually identified or a classifier is trained to identify components corresponding to noise. This requires a human in the loop or the annotation of components corresponding to noise. Such annotation is expensive and may not always be available. Recently, a technique was proposed that used unpaired training data to identify components corresponding to noise by formulating it as multi-instance learning problem [8]. Although this method was trained using unpaired training data, it performed ICA which has high computational complexity and large memory requirements, making it unsuitable for real-time applications. Also, their architecture consists of several subsystems like ICA, SAX and multi-instance learning. Each subsystem has its own hyperparameters and tuning them jointly is a challenging task.

Recently, several architectures for time series classification using deep neural networks have been proposed. Wang and Oates proposed a method for encoding time series data as images and using them to classify the time series [17]. Other methods that use raw data as input to a convolutional network for time series classification have been proposed [18]. Convolutional neural networks have also been used for classification of EEG data [1]. These methods demonstrate usefulness of a deep neural network for processing time series data. However, none of these methods have used convnets to denoise time series data.

Generative Adversarial Networks (GANs) [7] have achieved impressive results in computer vision tasks like image generation, image editing, and representation learning. GANs have also been used for removing compression artifacts from images [4]. Recently, CycleGAN, a method for performing unpaired image to image translation using generative adversarial networks, was proposed [19]. The key idea was introduction of an objective function that translates an image from the source domain to the target domain and reconstructs the original image. We modify this network for denoising time series data. To the best of our knowledge, ours is the only work that uses GANs for denoising time series data.

3 Problem Definition

Given a noisy time series A, our goal is to generate time series B that is a denoised version of A. If N is the noise in time series A, then we assume $A = B + N$. We want to learn to characteristics of noise from training data and remove noise from the original signal.

The main challenge in learning a mapping from noisy signal A to clean signal B is a lack of availability of paired training examples. We generally do not have a clean version of a noisy signal. This is because manually removing noise is an expensive task and needs domain expertise. But, it is much easier to collect signals with artifacts and signals without artifacts. Thus, we want to learn a mapping from a noisy signal to a clean signal using only a set of unpaired noisy signals and a set of clean signals.

Recently, a cycle generative adversarial network (cycleGAN) [19] was proposed to perform unpaired image to image translation. They presented an approach for translating an image from source domain A to target domain B, trained using unpaired data. This translation was performed by two networks, network F that translates the image from domain A to domain B and network G that translates the image from domain B to domain A. For training the network, cycleGAN introduced a cycle-consistency loss to enforce $G(F(A)) \approx A$. In this architecture network G does not use any information from the original image to reconstruct the signal. This makes the cycleGAN architecture unsuitable for denoising of time series data. For reconstruction of the original signal from a noisy signal, the network does not have access to the predicted noise component. Thus it does not know the nature or location of the noise. We solve this problem by creating an asymmetric variation of this architecture. In this architecture, we preserve the noise in the signal and use it for reconstruction of the original signal. We explain the architecture of our generative adversarial network in the next section.

4 Asymmetric Generative Adversarial Network

Our goal is to train an end-to-end system to remove artifacts from time series data using generative adversarial networks. We want to learn a mapping function from noisy signal A to denoised signal B given training signals $\{a_i\}_{i=1}^N$ and $\{b_j\}_{j=1}^M$ where $a_i \in A$ and $b_j \in B$. We denote the data distribution of these signals as $a \sim p_{data}(a)$ and $b \sim p_{data}(b)$.

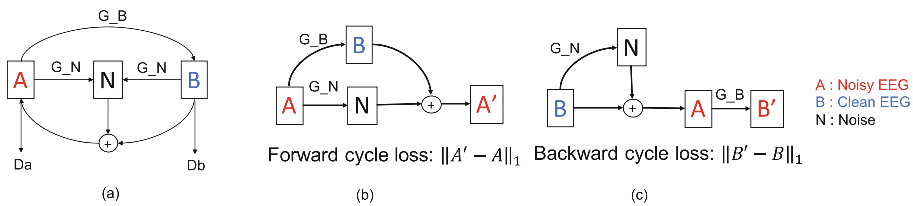

Fig. 1. Asymmetric GAN architecture

Figure 1(a) gives the architecture of our network. In this figure, boxes represent the signals and arrows represent neural networks or operations performed on input signals. As illustrated in the figure, G_B represents the function that maps noisy time series to clean time series. G_N is function that extracts noise if the input is a noisy time series A. It generates noise if the input is a clean time series B. The noise generated from G_N and clean time series B are added to get a noisy time series. The functions G_B and G_N are realized by convolutional autoencoders that are described in Sect. 4.1.

D_a and D_b are two adversarial discriminators. D_a aims to distinguish between noisy time series A and time series generated by adding noise $B + G\_N(B)$. D_b aims to distinguish between clean time series B and denoised time series $G\_B(A)$. We describe the architecture of the discriminators in Sect. 4.1.

To train this architecture, just like in cycleGAN, four losses are used, two adversarial losses and two cycle consistency loss. The adversarial losses are used for training the two mapping functions. For mapping function $G\_B : A \to B$, discriminator D_b matches the distribution of time series denoised by the generator and the distribution of clean time series. This loss function is given by Eq. 1.

$$L_{gan}(G_B, D_b, A, B) = \mathbb{E}_{b \sim p_{data(b)}}[log D_b(b)] + \mathbb{E}_{a \sim p_{data(a)}}[log(1 - D_b(G\_B(a)))] \tag{1}$$

Similarly, for mapping function $B \rightarrow A$, discriminator D_a matches the distribution of time series with noise generated by G_N and the distribution of noisy time series. This loss function is given by Eq. 2.

$$L_{gan}(G_A, D_a, B, A) = \mathbb{E}_{a \sim p_{data(a)}}[log D_a(a)] + \mathbb{E}_{b \sim p_{data(b)}}[log(1 - D_a(b + G\_N(b)))] \tag{2}$$

Adversarial losses ensure that the distribution of generated signals A and B match the target distribution. But generator networks can map input signals to any random permutation of signals in the target domain. Adversarial losses cannot guarantee that $G\_B(A)$ is a denoised version of noisy input time series A. To enforce this relation between noisy input and clean output by the generator we use a cycle consistency loss.

There are two cycle consistency losses, forward cycle loss and backward loss as shown in Fig. 1(b) and (c), respectively. For the forward cycle, we separate noise and the clean signal from the noisy signal using networks G_B and G_N, respectively. We calculate the l1-norm of the original signal and addition of the clean signal and noise as the forward cycle loss. The forward cycle loss is given by Eq. 3.

$$L_{forward\_cyc} = \mathbb{E}_{a \sim p_{data(a)}}||(G\_B(a) + G\_N(a)) - a||_1 \tag{3}$$

For the backward cycle, we add noise generated by generator G_N and use network G_B to clean the signal. The l1-norm of the original clean signal and the signal denoised by G_B is the backward cycle loss. This loss is given by Eq. 4.

$$L_{backward\_cyc} = \mathbb{E}_{b \sim p_{data(b)}}||G\_B(b + G\_N(b)) - b||_1 \tag{4}$$

Notice that the forward and backward cycles are not symmetric. Because of this, we call our GAN architecture asymmetricGAN. Asymmetry is introduced because in the forward cycle we use the noise extracted from the noisy signal to reconstruct the original signal. The usage of noise is necessary to ensure that the network is not penalized for adding noise at the wrong location. It also reduces the burden of generating the exact noise signal for correctly reconstructing the original signal. Forward and backward cycle losses are multiplied with hyperparameters λ_A and λ_B before adding to the final loss of the network.

4.1 Generator and Discriminator Network Architecture

Figure 2(a) shows the architecture of the generator. It is a convolutional autoencoder used for implementing functions G_B and G_N in asymmetricGAN. In

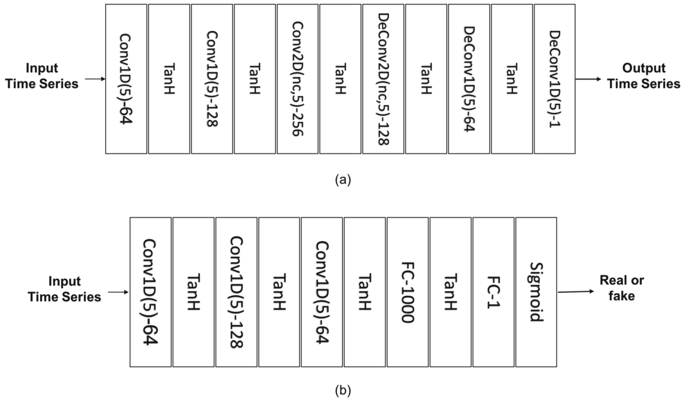

Fig. 2. (a) Generator architecture (b) Discriminator architecture

the figure, layer conv1D(5)-64 represents a layer of 64 features of $1D$ convolution with filter size 5. Figure 2(b) shows the architecture of the discriminator used for predicting if the input signal is real or generated. This architecture was used to implement networks D_a and D_b. In our architecture, we use $1D$ convolution with filter size K to capture temporal patterns in the time series and $2D$ convolution with filter size $N \times k$ to capture patterns across different channels, where N is the number of channels in time series. To create a network that is independent of the relative ordering of time series channels, we do not use $2D$ convolution with a spatial dimension of filter size less than N. This is important for the EEG data as the channel indices do not correspond to the spatial locations of the electrodes.

To train the asymmetricGAN and reduce model oscillation, we use the strategy of saving the history of generated time series [19]. We update the discriminators using a history of generated time series rather than the ones produced by the latest generative networks. We keep a buffer that stores the 50 previously generated time series for training. In the next section, we show the effectiveness of our method on synthetic and EEG dataset.

5 Experimental Setup and Results

5.1 Synthetic Dataset

Artifact removal from EEG data is a problem where ground truth does not exist because we do not have a clean version of the noisy signal. This makes the evaluation of artifact removal methods difficult. Also, visualizing and understanding EEG data is a time consuming task. We solve this problem by creating a simpler synthetic dataset.

In our synthetic dataset, we create a clean signal as a linear combination of a sine and a square wave. We create a noisy signal as a linear combination of sine, square and sawtooth waves. Thus, the sawtooth wave is playing the role of the noise we'd like to remove. The period of the sine and square waves is randomly selected between 2 and 5. The sawtooth wave has a period of 6. These signals are mixed to get 3 linear combinations. The mixing matrix is fixed and contains random numbers between 0.1 and 2. Both clean and noisy time series have a sample size of 1000 each. Our training set contains 4000 signals, the validation set has 1000 signals and test set has 100 signals. Figure 3(a) shows a noisy signal and Fig. 3(b) shows a clean signal.

To perform this task the network has to implicitly learn the nature of the noise and mixing matrix without direct supervisory information on either of these variables. Unlike the EEG dataset, this dataset is easy to create, visualize and interpret. This simplifies the validation of denoising time series data.

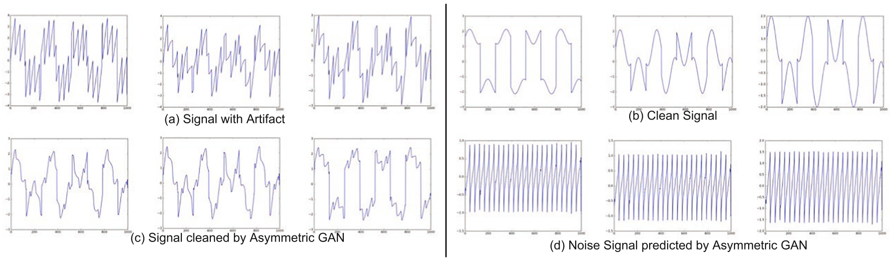

Fig. 3. Artificial data result

We train our network on the synthetic dataset using the adam optimizer and a learning rate of 0.0002. Both weights on cycle loss, λ_A and λ_B are 0.5. All weights were initialized from a Gaussian distribution with mean 0 and standard deviation 0.02. We test our network by measuring mean squared error (MSE) between the generated signal and ground truth clean signal. Figure 3 visualizes an example result on the test data. Figure 3(a) shows the noisy signal, Fig. 3(b) shows the ground truth clean signal, Fig. 3(c) shows the signal denoised by our network, and Fig. 3(d) is the noise detected by the network. We can observe from Fig. 3 that the ground truth clean signal and denoised signal are similar, and noise predicted by the network is similar to the sawtooth wave. The MSE between the ground truth clean signal and denoised signal for the example in Fig. 3 is 0.4065. The average MSE error for entire test set is 0.4180 and standard deviation is 0.011. This demonstrates the effectiveness of our network on the synthetic dataset.

5.2 EEG Dataset

We test our network for artifact removal on a dataset generated by the US Army Research Laboratory that has been previously discussed in [9]. We briefly describe the dataset here. Readers can refer to [9] for more details. The dataset was recorded using a 64-channel Biosemi ActiveTwo System. Participants in the study performed a block of artifact-inducing facial and head movements. The exact details of each movement were not controlled by the experimenter but rather left up to the participant to perform in their most natural manner. The seven movements performed included clenching the jaw, moving the jaw vertically, blinking both eyes, moving the eyes leftward then back to center, moving the eyes upwards then back to center, raising and lowering eyebrows, and rotating the head side-to-side. Each type of movement was performed in a separate run 20 times. At the beginning of each run, participants were told which movement to perform. For each run, a male voice initially counted down from 3 at a rate of every 2 s, followed by a tone every 2 s, and participants performed the movement in time with the tone. The participants were told to make the movement for the first second of the 2 s period, and then to return to a relaxed state for the remaining 1 s. A baseline dataset was recorded for each participant. Participants were told to not move and look straight at the computer screen for the baseline. We use this part of the dataset as "clean" data. Analyses here focuses on eyebrow movements in a single channel frontal electrode, with the intention of extrapolating results to other more complicated movements in the future.

Despite the fact that during collection of clean data patients were instructed to not move and look straight at the computer screen, we noticed that there were artifacts even in "clean" data. Manually annotating all artifacts from all channels is a time consuming task. So in this work, we focus on ocular artifacts in the Fp1 electrode of the frontal region. We manually remove all patients that have more than two ocular artifacts in the clean data and do not have artifacts in the region of eyebrow raising. In the resulting dataset, we have 4 patients with clean data and 10 patients with noisy data. Each patient's clean data has 4836 samples and noisy data has 420354 samples. We use this manually annotated data in all experiments below.

We train an asymmetricGAN on EEG data where the noisy signal contains artifacts corresponding to eyebrow raising and the clean signal does not contain any ocular artifacts. We use a sliding window of size 1000 over clean and noisy data as input to the network. Our network does not need to know the exact location of the artifact. Any window that contains an entire or partial artifact is considered noisy. The sliding window approach makes our model invariant to artifact location. Also, as the network can remove artifacts from a window and does not need to process the entire time series, it can be used in online real-time applications.

The DC component in EEG data is different for each recording. We normalize every window of clean and noisy data to remove the DC offset from the data. We remove the DC offset by subtracting the median of the data in the window. Normalizing by subtracting the median is more robust to outliers compared to subtraction of the mean. This preprocessing step ensures that the amplitude of clean segments of data in clean and noisy signals is centered around zero.

We use preprocessed windows of clean and noisy EEG data to train an asymmetricGAN. The parameters used for the network, the optimizer, and the initialization are similar to the one used for the synthetic dataset. Except in this case the number of input and output channels is one and λ_A and λ_B are 0.05.

Evaluation of EEG data is challenging as the ground truth noiseless signals are not known. Multiple approaches to evaluation have been proposed in recent years, however, authors do not agree on a single mechanism for evaluating artifact removal [15]. In this work, we give qualitative results and use an artifact detector to evaluate our method.

Qualitative Results. We have clean EEG for 4 patients and noisy EEG for 10 patients. To generate qualitative results, we train an asymmetricGAN using clean EEG from all 4 patients and noisy EEG from 7 randomly selected patients. We use noisy EEG from the remaining 3 patients to test our network. We generate qualitative results by performing a forward pass through network G_B over non-overlapping windows of 1000 samples on noisy data. We concatenate the output of the network over non-overlapping windows to get denoised EEG data. Figure 4 shows the result of denoising the noisy EEG of a patient from the test set. Visualization of the other two patients is shown in the appendix due to space constraints.

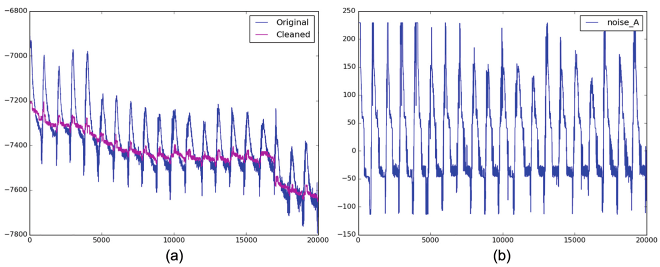

Fig. 4. EEG data result

Figure 4(a) shows the original noisy EEG signal and the signal after artifact removal. Figure 4(b) shows the artifact predicted by the network in EEG data. For collection of the original noisy EEG, the patient was instructed to raise their eyebrow every 2 s. For every eyebrow raise, there was spike in amplitude at the Fp1 electrode. As shown in the figure, the network learns that the spike in amplitude is because of the artifact and removes it. Artifacts extracted by our network as shown in Fig. 4(b) are similar to the artifacts that occur when an eyebrow is raised according to the existing literature [15].

Evaluation by Detection. In this section, we use artifact detection as a way of measuring the performance of artifact removal. This is less subjective, automatic and provides a quantitative measure of performance of artifact removal. We first train artifact detection and artifact removal on different datasets. We use the artifact detector to classify every window in EEG data denoised by the artifact removal algorithm. The error is given by the percent of windows where an artifact was detected in denoised EEG data.

For this experiment, we split the datasets into two parts. The first split contains clean EEG for 2 patients and noisy EEG for 6 patients. The first split is further divided into clean EEG of 2 patients and noisy EEG of 4 patients as a training set for the asymmetricGAN. The noisy EEG of the remaining 2 patients from the first split is used as test data for the asymmetricGAN.

The second split contains clean EEG of 2 patients and noisy EEG of 4 patients. It is used to train artifact detection network. The second split is further divided into training and test sets for artifact detection, each containing clean EEG for one patient and noisy EEG for two patients.

We use the network similar to the discriminator network explained in Sect. 4.1 to train an artifact detection algorithm. The preprocessing method of sliding a window over clean and noisy EEG and median normalization is also the same for the artifact detection network. The network is trained using the Adam optimizer and a learning rate of 0.0002. On training, we get an accuracy of 97.39% on test data for artifact detection.

To get an error metric, we use trained artifact removal to denoise noisy EEG in the test data. Then we classify every window in the denoised signal using artifact detector. The total number of windows in the test data were 38002. All these windows originally had either entire or partial artifact. The total number of windows having artifacts based on classification of the artifact detector after denoising are 10499. This shows that our artifact removal algorithm was able to change the classification of artifact detector from noisy to clean 72.37% of the time.

6 Conclusion

This paper presents an online, fully automated, end-to-end system for denoising time series data. Our system for denoising is trained using unpaired training corpus. It does not need any information about the source of the noise or how it is manifested in the time series data. We created a synthetic dataset and used it to evaluate our network. We also used model to remove artifacts from existing EEG dataset. In future, we intend to use our architecture for removing artifacts originating from other sources like muscular or cardiac activities from EEG data.

References

1. Bashivan, P., Rish, I., Yeasin, M., Codella, N.: Learning representations from EEG with deep recurrent-convolutional neural networks. arXiv preprint arXiv:1511.06448 (2015)
2. Daly, I., Nicolaou, N., Nasuto, S.J., Warwick, K.: Automated artifact removal from the electroencephalogram: a comparative study. Clin. EEG Neurosci. **44**(4), 291–306 (2013)

3. Fitzgibbon, S.P., Powers, D.M., Pope, K.J., Clark, C.R.: Removal of EEG noise and artifact using blind source separation. J. Clin. Neurophysiol. **24**(3), 232–243 (2007)
4. Galteri, L., Seidenari, L., Bertini, M., Del Bimbo, A.: Deep generative adversarial compression artifact removal. arXiv preprint arXiv:1704.02518 (2017)
5. Gandhi, S., Oates, T., Boedihardjo, A., Chen, C., Lin, J., Senin, P., Frankenstein, S., Wang, X.: A generative model for time series discretization based on multiple normal distributions. In: Proceedings of the 8th Workshop on Ph.D. Workshop in Information and Knowledge Management, pp. 19–25. ACM (2015)
6. Gao, J., Sultan, H., Hu, J., Tung, W.W.: Denoising nonlinear time series by adaptive filtering and wavelet shrinkage: a comparison. IEEE Sig. Process. Lett. **17**(3), 237–240 (2010)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
8. Jafari, A., Gandhi, S., Konuru, S.H., Hairston, W.D., Oates, T., Mohsenin, T.: An EEG artifact identification embedded system using ICA and multi-instance learning. In: 2017 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–4. IEEE (2017)
9. Lawhern, V., Hairston, W.D., McDowell, K., Westerfield, M., Robbins, K.: Detection and classification of subject-generated artifacts in EEG signals using autoregressive models. J. Neurosci. Methods **208**(2), 181–189 (2012)
10. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Min. Knowl. Disc. **15**(2), 107–144 (2007)
11. Ocbagabir, H.T., Aboalayon, K.A., Faezipour, M.: Efficient EEG analysis for seizure monitoring in epileptic patients. In: 2013 IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1–6. IEEE (2013)
12. Seneviratne, U., Mohamed, A., Cook, M., D'Souza, W.: The utility of ambulatory electroencephalography in routine clinical practice: a critical review. Epilepsy Res. **105**(1), 1–12 (2013)
13. Tracey, B.H., Miller, E.L.: Nonlocal means denoising of ECG signals. IEEE Trans. Biomed. Eng. **59**(9), 2383–2386 (2012)
14. Turner, J., Page, A., Mohsenin, T., Oates, T.: Deep belief networks used on high resolution multichannel electroencephalography data for seizure detection. In: 2014 AAAI Spring Symposium Series (2014)
15. Urigüen, J.A., Garcia-Zapirain, B.: EEG artifact removal state-of-the-art and guidelines. J. Neural Eng. **12**(3), 031001 (2015)
16. Vaughan, T.M., Heetderks, W., Trejo, L., Rymer, W., Weinrich, M., Moore, M., Kübler, A., Dobkin, B., Birbaumer, N., Donchin, E., et al.: Brain-computer interface technology: a review of the second international meeting (2003)
17. Wang, Z., Oates, T.: Imaging time-series to improve classification and imputation. arXiv preprint arXiv:1506.00327 (2015)
18. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1578–1585. IEEE (2017)
19. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593 (2017)