Speech Separation in the Frequency Domain with Autoencoder

Hao D. Do[1,2,3], Son T. Tran[1,2], and Duc T. Chau[1,2]
[1] University of Science, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
[3] OLLI Technology JSC, Ho Chi Minh City, Vietnam
Email: hao@olli-ai.com.vn, {ttson, ctduc}@fit.hcmus.edu.vn

***Abstract*—**Speech separation plays an important role in a speech-related system because it can denoise, extract and enhance speech signal, and after all improve the accuracy and performance of the system. In recent years, many approaches only separate the speech out of commonly high-frequency noise or a particular background sound. We propose a more powerful approach, combining an autoencoder and a bandpass filter to separate speech signals. This combination can extract the speech in the mixture with not only high-frequency noise but also many kinds of different background sounds. Our approach can be flexibly applied for the new background sounds. Experimental results show that our model can extract fastly and effectively the speech signal with 9.01 dB in SIR and 11.26 in SDR. On the other hand, we can adjust the passband to identify the range of frequency at the output signal to apply for particular applications.

***Index Terms*—**Speech separation, autoencoder, bandpass filter, frequency domain

I. Introduction

In many speech applications, the quality of the input speech signal holds a significant role in the whole system because it affects directly the machines process. Speech separation, a kind of Blind Source Separation (BSS) [1], [2], is one of the first and the most challenging task to control whether what we push into the main algorithm is enough and good or not. In reality, background sounds always exist and mix into the main input signal like human speech. It cannot be known which and when the background sounds such as music sound, traffic sound, or television sound will affect the main signal. These sounds, when mixed into the speech, can cause a lot of deviating results in computation. If the sound source is separated into many independent frequency elements, the main signal can be extracted and reconstructed from some of them while the remaining ones are ignored because they are noises.

In this research, we propose a fast and effective approach to extract the speech signal out of the recorded sound. This is the combination of an Autoencoder - a machine learning model, and a bandpass filter - a powerful tool in signal processing. In the frequency domain, we design an Autoencoder network to capture the bottleneck features in the input signal, which contain most of the important information about the content and prosody of speech. The reconstructed signal via the Autoencoder is filtered by a bandpass filter to capture only the frequency band which is useful for the application. The model includes two main parts including a non-deep autoencoder and a bandpass filter, so it run too fast. On the other hand, the combination of Autoencoder and the bandpass filter can clear most of the background sound and noise from the recorded sound, and hold most of the important information need for high-level applications.

There are many effective proposed methods for separating the speech sound of the background sound and some of them are applied in the industrial field. The first group includes the works using digital filters. With this approach, they can separate signals and reduce the impact of out of range frequency elements. For an instant, because the main information of human speech sound concentrates from 50 Hz to 5000 Hz, they filter the elements which are lower than 50 Hz and higher than 5000 Hz. This approach only applies to some cases when the elements in the mixture are separated in frequency domain representation. The second group includes more modern methods like wavelet transform or filter bank. With these approaches, they cancel not only out of range frequency but also apart of frequencies which intersect with the human speech signal. The third group uses data dimension reduction or data compression methods. The researches extract the main elements of the signal and then use them to reconstruct the whole signal. The elements, which do not have a significant impact on the signal, including background sounds and noises, can be reduced. The shared point of these above works is reducing well out of range frequency elements and partly of intersecting elements.

We show that our approach can reduce perfectly out of range frequency elements and mostly intersect elements. We construct a neural network architecture with an autoencoder style to process the signal in the frequency domain. This approach can be extended to apply to many similar applications related to speech or signal analysis to enhance the input signal. Because using the strength of the neural network, we interfere with all frequency of a signal to keep the speech signal and reduce the background signal. Our experimental results show that we reduce mostly background elements. This is a significant improvement in comparison with the state of the art result presented by Ning Yang [1] in 2017.

Manuscript received May 1, 2020; revised September 20, 2020.
doi:10.12720/jcm.15.11.841-848

The proposed approach is a good solution for speech-related applications. Firstly, the model separates speech from the signal very fast, so it can use in real-time applications like voice bot or voice translator. Next, because the two main parts of the model are only small and not complicated, so it can be deployed into not only the server but also the edge devices such as the microphone. Lastly, and the most important, the proposed method can extract and enhance the speech signal from the input, so it improves performance a lot in many applications such as speech recognition and speaker identification.

The remaining of this paper is structured with 3 main parts. Part II presents many works and researches related to the problem of blind source separation. We also summarize some signal transforms because they are the essential method to translate the signal from the time domain to the frequency domain and on the reverse side. Our proposed model is described in detail clearly in part III. We present a mathematical base, model architecture, and training method for the model in this part. In part IV, we design n experiments to validate our method. After training model, we compare our results with the other works to specify the strength and weaknesses of our approach.

II. Related Works

In recent years, there are a lot of techniques proposed to solve the BSS problem. They are explored deeply to reconstruct the main signal from signal mixtures, especially the speech mixtures [2], [3]. There are many effective algorithms for BSS and the most popular methods include Principal Component Analysis (PCA), Singular Value Decomposition (SVD), and Independent Component Analysis (ICA), [4]–[7]. These algorithms have been used in many applications, such as signal processing, wireless communication, and Electrocardiogram technologies [8], [9].

Time-domain frequency representation is the original form of any kind of signal. It has been widely used as a tool to separate a mixture of signals into many components and then has applications radar processing, speech processing, and audio signal processing [10]–[13]. In these applications, many convolutional methods are used to deal with continuous signals BSS. However, these solutions work with an expensive computational cost and require hardware resources for implementation [14].

Short-Time Fourier Transform (STFT) has been applied to determine the individual frequency components in each time segment corresponding with the fact that the signal changes over time. The STFT divides the mixed signal into small segments of equal lengths to calculate the Fourier spectrum of individual components to plot the changing spectrum as a function of time [15], [16]. The STFT uses Time Window (TW) of fixed sizes to obtain the local signals to analyze each time frame. After getting the local components, Fourier Transform (FT) is applied for further analysis. This analysis may produce poor results in the temporal domain [17]. To minimize fluctuations from obtained results, many adaptive algorithms are introduced in the STFT domain [18]–[22].

III. Autoencoder for BSS

A. BSS Problem

Given a mixed-signal, the main work here is separating the mixture into N independent signals. There is no information about the mixture and its elements. On the other hand, the way they mix is not known, so it can be a linear mixture or nonlinear mixture. In speech-mixed signals, one element is the pure speech which is created by a human, and the others include background sounds from the environment such as TV sound, music, fan sound, or traffic sound. In that case, the BSS problem is how to extract the speech sound and all other background sounds out of the mixture.

Traditional BSS description is formulated to solve the Cocktail party problem. This means there are m sound sources, supposing human sound and background sounds, and n recording devices. In most cases, n is smaller than m, so the whole system is underdetermined and non-linear approaches should be used to reconstruct the original sources. In other cases, the problem can be solved better because there is more provided information, but these cases are not common in the real world. At home or work office, the number of sources corresponding with the number of background is many while the number of recording devices is usually one.

Let $s(t) = (s_1(t), s_2(t), s_3(t), \dots, s_m(t))^T$ and $x(t) = (x_1(t), x_2(t), x_3(t), \dots, x_n(t))^T$ denote the sets of individual sources, and individual recording devices, respectively. Each elements of $x(t)$ is considered as a combination of all sources $s_i(t)$ in $s(t)$, so this can be rewrite as follow:

$$x_j(t) = \sum_{i=1}^{m} a_{ji} s_i(t), j \in [1, n]$$

All a_{ji} with $j \in [1, n]$ and $i \in [1, m]$ values form a matrix called mixed matrix $A = [a_{ji}] \in R^{mxn}$. In practice, each $x_j(t)$ attacked by noise $\gamma_j(t)$, BSS problem can be described by $x''_j(t) = (x''_1(t), x''_2(t), x''_3(t), \dots, x''_n(t))^T$:

$$x''_j(t) = \sum_{i=1}^{m} a_{ji} s_i(t) + \gamma_j(t)$$

or:

$$x''(t) = A * s(t) + \gamma(t)$$

Because the noise signal $\gamma_j(t)$ can be solved effectively by using digital filters, the main work in BSS problem is finding inverse matrix of A.

$$x(t) \approx F(x''(t))$$

where *F(.)* is a noise filter; and source elements $s(t)$ can be found by $x(t)$ the inverse of matrix A;

$$s(t) = A^{-1} x(t)$$

When the signals are represented in discrete domain, the three equations below can be rewritten by:

$$x''_j[n] = \sum_{i=1}^{m} a_{ji} s_i[n] + \gamma_j[n]$$

$$x''[n] = A * s[n] + \gamma[n]$$

$$s[n] = A^{-1}x[n] \approx A^{-1}F(x''[n])$$

In this work, we focus on a variant of BSS which is called speech separation. In many real-world applications such as speech recognition, speaker recognition, or voice virtual assistant, the end devices receive speech signal from users and then throw the responds. It is hard to record human voices in a clean environment because background sounds exist everywhere in the house, so it is needed to separate the speech signal out of the recorded sound, and that work forms the problem called speech separation.

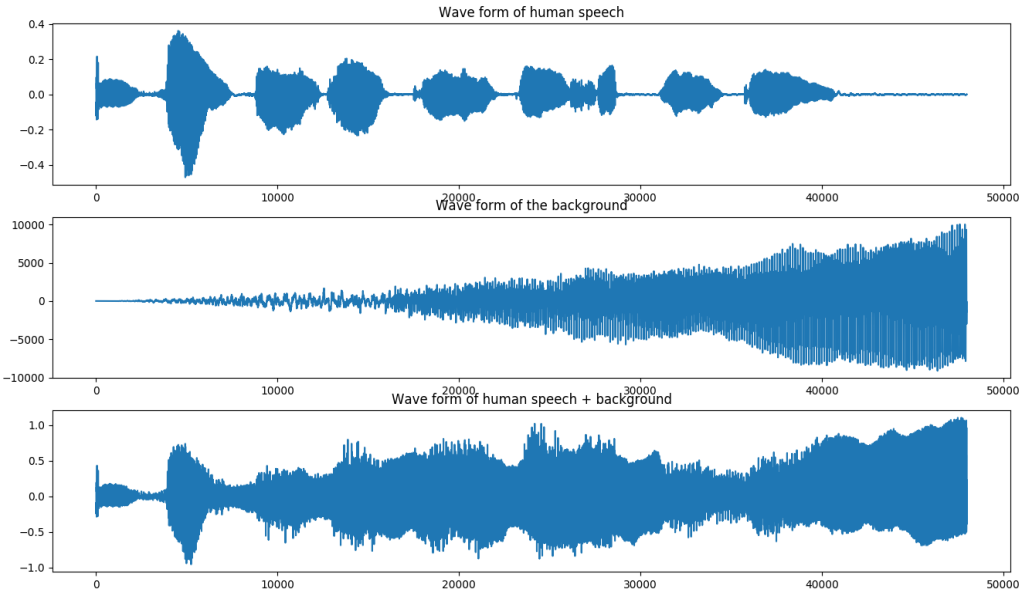

Fig. 1. Waveform of human speech, trumpet, and the mixture.

Different from BSS, we do not consider all elements in the sources s[n] in speech recognition. We only focus on the speech signal, so we can paraphrase the BSS as the smaller BSS as separating the mixture into the speech signal and the remaining. In some cases, we do not care about the remaining element, and so speech separation means speech extraction. Fig. 1 describes a particular illustration of the speech separation problem. There are three waveforms corresponding with three signals. The first is the description of pure human speech. This is the signal which is recorded in a professionally recorded room, so it contains no noise and background sounds. The second signal is a clipped trumpet (an instrument) sound. This sound is clear and clean. We then mix these two signals to form the third waveform. It contains two elements including speech sound and trumpet sound. In speech separation, the main mission is extracting the first signal from the third signal.

B. Sort Time Fourier Transform

Fourier Transform (FT) presents the signal in the frequency domain so the signal is represented by a summary of many sinusoidal elements. Here is the formula for FT of $y[n]$, a discrete signal with N elements:

$$Y[k] = \sum_{n=0}^{N-1} y[n] e^{-j2\pi kn/N}$$

These factors show the power of sinusoidal waves and hence the properties of the signal. FT only shows the global attributes of the signal because each sub-wave is computed from the whole signal. That causes the development of Short-Time Fourier Transform, a transform that can show the values of sinusoidal waves and their changes by the time. The difference between FT and STFT is shown in Fig. 2. as follow:

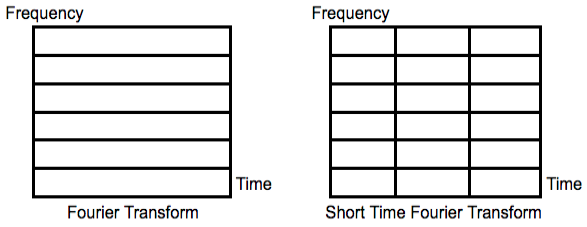

Fig. 2. Resolution of fourier transform and short time fourier transform.

STFT is present by the formula:

$$Y(m,k) = \sum_{n=0}^{N-1} y[n] w[n-m] e^{-j2\pi kn/N}$$

In STFT formula, the left part presents the amplitude of each element in the time-frequency domain. Particularly, $Y(m,k)$ is the amplitude at time m of frequency k. On the other hand, the window function w[t]

in the right part defines where and how the sub-range of the signal is taken to present into the frequency domain. There are many different window functions including rectangle, Gaussian, Black man - Harris window, etc. In this research, we use Black man - Harris function [23]:

$$w[n] = a_0 - a_1 cos\left(\frac{2\pi n}{N}\right) + a_2 cos\left(\frac{4\pi n}{N}\right) - a_3 cos\left(\frac{6\pi n}{N}\right)$$

This function is used because it minimizes the side-slope levels of the window by adding three more sinusoidal elements.

From frequency domain, the signal is converted to time domain using Inverse Fourier Transform (IFT) as follow:

$$y[n] = (1/N)\sum_{k=0}^{N-1} \left(Y[k]e^{-j2\pi kn/N}\right)$$

With Inverse Short Time Fourier Transform, we use Filter Bank Summation (FBS) [24] to reconstruct the signal in time domain:

$$y[n] = (1/w[0])(1/N)\sum_{k=0}^{N-1} \left(Y[m,k]e^{-j2\pi kn/N}\right)$$

Although the audible range of human ears in the frequency domain is from 20 Hz to 20,000 Hz, the real distribution of speech elements is not uniform. Fig. 3 shows a clear illustration of this distort distribution. In this case, the densest area is from 0 Hz to 5,000 Hz. Phonetic researches show the fact that most of the content in the speech spread in the range under 1,100 Hz. On the other hand, the frequency elements under 5,000 Hz presents the speaker's properties. The likelihood of whether a person can recognize a human speech is his acquaintance or not depends on the distribution of the signal in this range. Not to miss anything of speech, we analyze the signal in the range from 0 to 8,000 Hz to commit that we capture all information from the input signal in the frequency domain.

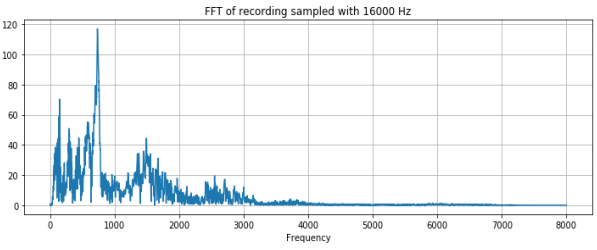

Fig. 3. Frequency distribution of speech signal

C. Autoencoder

This is a special kind of neural network that requires the input layer and the output layer are the same sizes. This is a representative of the unsupervised learning algorithm because nothing is used except the input data itself. Fig. 4 illustrates the architecture of a common autoencoder.

An Autoencoder contains two sub-network including Encoder and Decoder. These two networks are linked by a small layer called Code. This layer must be smaller than the input and output layers. When using Autoencoder to extract features, the output is expected the same with the input in not only layer size but also the value. Encoder network compresses all information in the input layer to the Code layer, and then, the Decoder network reconstructs the information from the Code layer to the Output layer. If the value at the output layer is nearly equal with at the input, that means the main information can be reconstructed with the Code layer, or the Code layer contains most of the important information of the input layer.

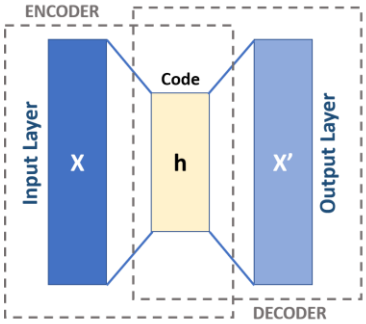

Fig. 4. An autoencoder with encoder and decoder sub-networks.

Denoting input space Θ, code space Φ, input x, output x'activation function σ, weight w, and bias b, a 3-layer Autoencoder can be formulated as follow:

$$Encoder: \Theta \to \Phi, h = \sigma_{in}(w_{in} * x + b_{in})$$

$$Decoder: \Phi \to \Theta, x' = \sigma_{out}(w_{out} * h + b_{out})$$

$$Encoder - Decoder: \Theta \to \Theta\ x' = \sigma_{out}(w_{out} * \sigma_{in}(w_{in} * x + b_{in}) + b_{out})$$

Then the lost function is:

$$L(x, x') = ||x - x'||^2$$

$$= (x - \sigma_{out}(w_{out} * \sigma_{in}(w_{in} * x + b_{in}) + b_{out}))^2$$

In real design, an AutoEncoder can contain many layers in both Encoder and Decoder sub networks. In this case, the formulas are similar with the formulas below with the only difference is the output of a layer is the input for the next layer, or the formula can be described as a nested function:

$$h_{i+1} = \sigma_i(w_i * h_i + b_i)$$

With $h_i$is the value at i^{th}hidden layer. Assuming there are $n$layers in the AutoEncoder including one input layer, one output layer and $n-2$ hidden layers, we use Leaky ReLU activation function for all layers except at the output layer:

$$\sigma_i(x) = x, x > 0$$

$$\sigma_i(x) = kx, x < 0\ and\ k = 0.01$$

At the output layer, we do not use any specific activation function because the main purpose of this layer is to reconstruct the value at the input, so we apply Identity function for this layer:

$$\sigma_{output}(x) = x$$

Because Autoencoder is just a multilayer neural network, there is nothing different in training and inferring processes in comparison with a normal neural network. We use the Backpropagation algorithm to learn all parameters in the network in the training phase and forward propagation for inference.

We use Autoencoder to process the signal in the frequency domain. This means the input of Autoencoder is the STFT of mixed-signal, and the expected output is the STFT of the speech signal. We then use this frequency representation to compute the value of speech signals in the time domain.

D. Bandpass Filter

The bandpass filter is designed to pass the signal through the band 50 Hz and 5000 Hz. This is the combination between a high pass filter with the cutoff frequency is 50 Hz and a low pass filter with the cutoff frequency is 5000 Hz. We filter this range because in contains most of main information of speech signal. The remaining of this section describes in detail the low pass filter while the high pass filter is designed as the similar method.

The gain or amplitude response of the Chebyshev low pass filter is:

$$G_n(\omega) = H_n(j\omega) = \frac{1}{\sqrt{1 + \varepsilon^2 T_n^2\left(\frac{\omega}{\omega_0}\right)}}$$

With ε is the ripple factor, T_n is n^{th} order Chebyshev polynomial, and ω_0 is the cutoff frequency. These element, except ω_0 are determined by:

$$\varepsilon = \sqrt{(10^{\varphi/10} - 1)}$$

With φ is the passband ripple, a constant which is usually set by a small number to show the difference between maximum and minimum values of gain in passband region (Fig. 5). T_n is n^{th} order Chebyshev polynomial, which is a recursion function:

$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x), \text{ n>0}$$

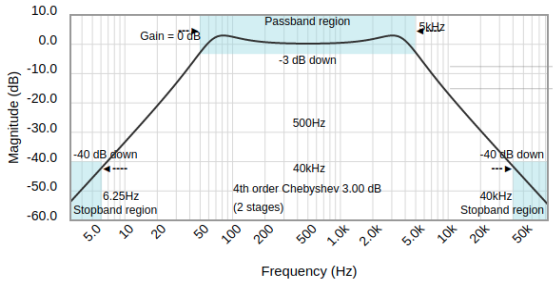

Fig. 5. Gain response for 4^{th} order Chebyshev bandpass filter.

E. Proposed Solution for BSS

Our solution for the BSS problem is the combination of an Autoencoder network with a Chebyshev bandpass filter in the frequency domain. The mixed signal is transformed by STFT, then push to the processing block, and finally is computed by the ISTFT algorithm to reconstruct into the time domain. The illustrations for this whole process is described in Fig. 6.

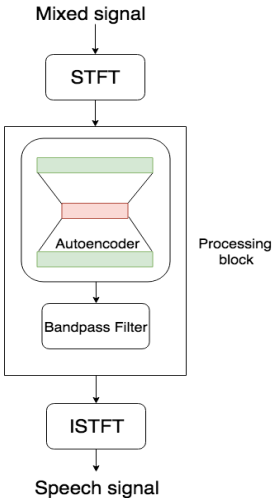

Fig. 6. The combination of Autoencoder and bandpass filter for separating speech signal from.

IV. Experimental Result

A. Dataset

In this work, we use the TIMIT dataset as the data for the main speech signal and trumpet sounds for background sound. This dataset contains a lot of recording speeches from 630 speakers. Each of them is recorded 10 times with 10 different long sentences. There are eight main dialects of English in TIMIT, so it helps us to evaluate our approach in many cases with different kinds of speech sounds. With each dialect, we choose randomly a lot of samples from the dataset (depending on the particular experiments) and then mix them with the trumpet sound to form the mixed sound. With a signal x[n], we get its mixture s[n]. Thus, we can represent a pair of input and output of the whole model as follow:

$$(input, output) = (s[n], x[n])$$

The trumpet sounds are cloned into many versions with many different powers by multiplying the sound signal with an array of real numbers. Each version corresponds with each level of the magnitude of background sound. This means that when we multiply the signal with a big number, the background in the mixture is too big, maybe bigger than the speech sound. In this way, we can evaluate the performance of our design whether it can extract the speech signal from a noisy environment or not.

B. Evaluation Method

Following by Vincent in [25], in BSS problem, the estimated signal of a source signal can be described as a mixture of four elements:

$$s_{estimated}(t) = s_{target}(t) + e_{inter}(t) + e_{noise}(t) + e_{artif}(t)$$

With $s_{target}(t)$ is the expected signal, $e_{inter}(t)$is the interference of more than one sources in the mixture, $e_{noise}(t)$ is the noise, and $e_{artif}(t)$ is the environment background like music or electric fan sounds. In speech separation, we only consider speech signal in the mixture, so we do not need to estimate and decompose for the others sources.

To evaluate the performance of separation process, Vincent propose four measures including Source to Distortion Ratio (SDR), Source to Interferences Ratio (SIR), Sources to Noise Ratio (SNR), and Sources to Artifacts Ratio (SAR) as below:

$$SDR = 10\,log_{10}\left(\frac{||s_{target}||^2}{||e_{inter}(t) + e_{noise}(t) + e_{artif}(t)||^2}\right)$$

$$SIR = 10\,log_{10}\left(\frac{||s_{target}||^2}{||e_{inter}(t)||^2}\right)$$

$$SNR = 10\,log_{10}\left(\frac{||s_{target} + e_{inter}(t)||^2}{||e_{noise}(t)||^2}\right)$$

$$SAR = 10\,log_{10}\left(\frac{||s_{target} + e_{inter}(t) + e_{noise}(t)||^2}{||e_{artif}(t)||^2}\right)$$

We use SIR and SDR to evaluate our approach in speech separation because these measures focus on target signal and the environment background signal. These measures are available in the Matlab BSS Eval toolbox. We compute SIR and SDR between pure speech signal and the mixtures to show what we extract from the input mixture.

C. Experiments and Results

1) One dialect versus one background

There are 8 dialect sounds in TIMIT and we use all of them in this experiment. With each dialect, we choose randomly ten people with all one hundred utterances. Then we mix all of these utterances with the trumpet sounds and use 90 samples in the training phase. After the model convergences, we reconstruct ten remaining samples to get the output signals and finally compare them with the ground truths. Here are the results in Table I:

TABLE I. BSS EVAL FOR EXPERIMENT 1

Dialect	SIR (db)	SDR (db)
1	8.75	11.02
2	6.11	8.20
3	6.80	7.54
4	12.34	13.02
5	2.28	2.15
6	9.61	10.11
7	9.36	8.69
8	4.14	4.92

2) Many dialects versus one background

In this experiments, we choose randomly 1000 utterances from TIMIT and then mix them with trumpet sound, then process them with our proposed model. As can be seen in the Table II, the experiments show that our approach gain a good result in comparison with the current state of the art model.

TABLE II. BSS EVAL FOR EXPERIMENT 2

Test case	SIR	SDR	SIR[1]	SDR[1]
1	8.65	9.83	9.12	11.23
2	9.43	12.01		
3	9.14	10.35		
4	8.98	11.56		
5	8.87	12.54		
Average	9.01	11.26		

V. CONCLUSION

In this work, we propose a new design in the frequency domain for the BSS problem to separate speech signals out of the mixed signal. We combine an autoencoder and a bandpass filter to reduce the impact of intersection elements on the main signal. The autoencoder is the main processor and bandpass filter to clear all out of range elements. The experimental results show that our approach is more effective than many works before so it is a potential for many real applications.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

In this research, all three authors discussed and agreed with the main approach. Hao D. Do was in charge of collecting data and implementing the algorithms. He then mainly wrote the draft of the paper. Finally, Son T. Tran and Duc T. Chau reviewed and revised the paper before submitting.

ACKNOWLEDGMENT

We would like to express our gratitude to OLLI Technology JSC for their financial support and acceptance for us to use the computing infrastructure during this work.

REFERENCES

[1] N. Yang, M. Usman, X. J. He, M. A. Jan, and L. M. Zhang, "Time-frequency filter bank: A simple approach for audio and music separation," *IEEE Access*, vol. 5, pp. 27114 – 27125, 2017.

[2] Y. Xie, K. Xie, Z. Wu, and S. Xie, "Underdetermined blind source separation of speech mixtures based on K-means clustering," in *Proc. Chinese Control Conference (CCC)*, Guangzhou, China, 2019, pp. 42-46.

[3] B. Peng, W. Liu, and D. P. Mandic, "Design of oversampled generalised discrete Fourier transform filter banks for application to subband based blind source separation," *IET Signal Process*., vol. 7, no. 9, pp. 843–853, 2013.

[4] C. Osterwise and S. L. Grant, "On over-determined frequency domain BSS," *IEEE/ACM Trans. Audio, Speech, Language Process*, vol. 22, no. 5, pp. 956–966, May 2014.

[5] S. H. Sardouie, M. B. Shamsollahi, L. Albera, and I. Merlet, "Denoising of ictal EEG data using semi-blind source separation methods based on time-frequency priors," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 3, pp. 839–847, May 2015.
[6] B. Rivet, "Source separation of multimodal data: A second-order approach based on a constrained joint block decomposition of covariance matrices," *IEEE Signal Process. Lett.*, vol. 22, no. 6, pp. 681–685, June 2015.
[7] S. Lee and H. S. Pang, "Multichannel non-negative matrix factorisation based on alternating least squares for audio source separation system," *Electron. Lett.*, vol. 51, no. 3, pp. 197–198, 2015.
[8] G. S. Fu, R. Phlypo, M. Anderson, X. L. Li, and T. Adalı, "Blind source separation by entropy rate minimization," *IEEE Trans. Signal Process*, vol. 62, no. 16, pp. 4245–4255, Aug. 2014.
[9] J. Hofmanis, O. Caspary, V. Louis-Dorr, R. Ranta, and L. Maillard, "Denoising depth EEG signals during DBS using filtering and subspace decomposition," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 10, pp. 2686–2695, Oct. 2013.
[10] O. Tichý and V. Šmídl, "Bayesian blind separation and deconvolution of dynamic image sequences using sparsity priors," *IEEE Trans. Med. Imag*., vol. 34, no. 1, pp. 258–266, Jan. 2015.
[11] J. Nikunen and T. Virtanen, "Direction of arrival based spatial covariance model for blind sound source separation," *IEEE/ACM Trans. Audio, Speech, Language Process*, vol. 22, no. 3, pp. 727–739, Mar. 2014.
[12] B. Liu, V. G. Reju, A. W. H. Khong, and V. V. Reddy, "A GMM post-filter for residual crosstalk suppression in blind source separation," *IEEE Signal Process. Lett*., vol. 21, no. 8, pp. 942–946, Aug. 2014.
[13] B. Liu, V. G. Reju, and A. W. H. Khong, "A linear source recovery method for underdetermined mixtures of uncorrelated AR-model signals without sparseness," *IEEE Trans. Signal Process*, vol. 62, no. 19, pp. 4947–4958, Oct. 2014.
[14] S. Hosseini and Y. Deville, "Blind separation of parametric nonlinear mixtures of possibly auto correlated and non-stationary sources," *IEEE Trans. Signal Process*, vol. 62, no. 24, pp. 6521–6533, Dec. 2014.
[15] Y. Zhang, P. Candra, G. Wang, and T. Xia, "2-D entropy and short-time Fourier transform to leverage GPR data analysis efficiency," *IEEE Trans. Instrum. Meas*., vol. 64, no. 1, pp. 103–111, Jan. 2015.
[16] G. Okopal, S. Wisdom, and L. Atlas, "Speech analysis with the strong uncorrelating transform," *IEEE/ACM Trans. Audio, Speech, Language Process*, vol. 23, no. 11, pp. 1858–1868, Nov. 2015.
[17] J. L. Roux and E. Vincent, "Consistent wiener filtering for audio source separation," *IEEE Signal Process. Lett*., vol. 20, no. 3, pp. 217–220, Mar. 2013.
[18] Y. G. Jin, J. W. Shin, and N. S. Kim, "Spectro-temporal filtering for multichannel speech enhancement in short-time Fourier transform domain," *IEEE Signal Process. Lett*., vol. 21, no. 3, pp. 352–355, Mar. 2014.
[19] R. E. Turner and M. Sahani, "Time-frequency analysis as probabilistic inference," *IEEE Trans. Signal Process*, vol. 62, no. 23, pp. 6171–6183, Dec. 2014.
[20] L. Stankovic, S. Stankovic, and M. Dakovic, "From the STFT to the Wigner distribution [lecture notes]," *IEEE Signal Process. Mag*., vol. 31, no. 3, pp. 163–174, May 2014.
[21] V. K. Mai, D. Pastor, A. Aïssa-El-Bey, and R. Le-Bidan, "Robust estimation of non-stationary noise power spectrum for speech enhancement," *IEEE/ACM Trans. Audio, Speech, Language Process*, vol. 23, no. 4, pp. 670–682, Apr. 2015.
[22] P. Flandrin, "Time–frequency filtering based on spectrogram zeros," *IEEE Signal Process. Lett*., vol. 22, no. 11, pp. 2137–2141, Nov. 2015.
[23] R. B. Blackman and J. W. Tukey, *The Measurement of Power Spectra from the Point of View of Communications Engineering*, Dover Publications Publishing House, 1959.
[24] T. F. Quatieri, *Discrete-time Speech Signal Processing: Principles and Practice*, Prentice Hall Publishing House, 2001.
[25] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, Institute of Electrical and Electronics Engineers, vol. 14, no. 4, pp. 1462–1469, 2006

Hao D. Duc received the B.Sc and M.Sc degrees in computer science from the University of Science, Vietnam National University at Ho Chi Minh City, Vietnam, in 2015 and 2017. He is currently a Ph.D. student in computer science at the same university. His research interests include speech signal processing and deep learning.
He is also an AI researcher at Olli Tech. JSC where he leads a team of seven researchers to build a system of intelligent agents for the smart home.

Son T. Tran received a Bachelor's degree in science from the University of Science, Vietnam National University at Ho Chi Minh City, Vietnam, in 1997, and the Ph.D. degree in engineering from the Department of Electrical and Computer Engineering, Toyota Technological Institute, Japan, in 2005.
Currently, he is a senior lecture at the Faculty of Information Technology and the head of the Office of education and training, the University of Science, Vietnam National University at Ho Chi Minh city. His research interests include filtering, image processing, and pattern recognition.

Duc T. Chau received a Ph.D. degree in Information Science from JAIST, Japan, in 2005. His research interests include signal processing, particularly in Spoken Language (Localization, Enhancement, Recognition) and Image Processing (Analysis, OCR, Information Extraction).

He is currently a lecturer at the Faculty of Information Technology, University of Science, Vietnam National University at Ho Chi Minh city. He is also an AI Tech. Leader at Cinnamon AI Lab.