IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 18, 2020, accepted August 20, 2020, date of publication August 26, 2020, date of current version September 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019495

Speech Source Separation Using Variational Autoencoder and Bandpass Filter

HAO DUC DO[1,2,4], SON THAI TRAN[1,2,3], AND DUC THANH CHAU[1,2]

[1]Faculty of Information Technology, University of Science, Ho Chi Minh City 70000, Vietnam
[2]Viet Nam National University, Ho Chi Minh City 70000, Vietnam
[3]Office of Education and Training, University of Science, Ho Chi Minh City 70000, Vietnam
[4]AI Lab, OLLI Technology JSC, Ho Chi Minh City 70000, Vietnam

Corresponding author: Son Thai Tran (ttson@fit.hcmus.edu.vn)

This work was supported by the OLLI Technology JSC.

ABSTRACT Speech source separation is essential for speech-related applications because this process enhances the input speech signal for the main processing model. Most of the current approaches for this task focus on separating the speech of commonly high-frequency noises or a particular background sound. They cannot clear the signals which intersect with the human speech in its frequency range. To deal with this problem, we propose a hybrid approach combining a variational autoencoder (VAE) and a bandpass filter (BPF). This method can extract and enhance the speech signal in the mixture of many elements such as speech signal, the high-frequency noises, and many kinds of different background sounds which interfere with the speech sound. Experimental results showed that our model can extract effectively the speech signal with 15.02 dB in Signal to Interference Ratio (SIR) and 12.99 dB in Signal to Distortion Ratio (SDR). On the other hand, we can adjust the passband to identify the range of frequency at the output signal to apply for a particular application like gender classification.

INDEX TERMS Generative model, variational autoencoder, bandpass filter, speech separation.

I. INTRODUCTION

In many speech-related applications, the quality of the input speech signal holds a significant role in the whole system because it affects directly to the workflow of the main model. To improve the speech signal quality or enhance the speech signal, it is necessary to separate the speech out of the raw input signal. This means that the raw input signal should be separated into a speech signal and the remaining signal called the interfering signal. This process is also called Speech Source Separation (SSS) [1], a specific case of Blind Source Separation (BSS) [2]. SSS is one of the most important tasks to deal with in the pre-processing phase since it controls the signal we push into the main algorithm is good or not. In reality, the interfering signal includes background sounds and noises. The background sounds are the sounds that always exist in the environment and interfere with the human speech such as music sound, traffic sound, or television sound. It cannot be known which and when the background sounds mask the speech signal. These sounds, when mixed into the speech, can cause a lot of deviating results in computation. On the other hand, many kinds of noise usually exist in the input signal including thermal noise, or the noise caused by the works of signal receivers. If these unexpected elements still exist in the speech signal when it is passed into the main model, the final results will be falsified.

The term ''blind'' in BSS implies that there is no given information about the noise or background, so this causes a lot of difficulties when canceling the background sounds. Since there is no limit for background, their frequencies arrange from very low to very high and then intersect to the frequency distribution of the main signal. Similar to the background, noise can exist at any frequency range and everywhere in the signal. The differences between the backgrounds and the noises are mainly two aspects: amplitudes and distributions. The common backgrounds, such as music sounds or traffic sounds, have big amplitudes and unique distributions so it is easily recognized by the human ear. Differently, the noises do not exist as a clear distribution and are much smaller than human speech. Both background sounds and noises mask the speech signal in different ways so it is a big challenge to reduce their impacts and enhance the speech signals.

In this research, we propose an effective approach to extract the speech signal out of a raw signal. This is the combination of a Variational Autoencoder (VAE) [3], [4] and a Bandpass Filter (BPF). First, we use a VAE to capture the

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

bottleneck features [5] in the input signal. This network will contain most of the important information about the content and prosody of the speech. Then, the signal is filtered by a BPF to capture only the frequency range of human speech, which is useful for the application. Generally, the model includes two main components including a non-deep VAE network and a BPF. This combination not only clears most of the interference from the background sounds and noises to the input sound but also holds most of the important information in the speech signal which is needed for high-level applications.

Back to the development of the SSS problem, many effective methods are proposed and then applied in the industry but none of them solves this problem completely. They are designed to deal with a particular interfering signal such as high-frequency noise or music. Generally, these methods are good approaches for SSS but there are still limited with each of them when they are applied in the real world. The next paragraphs will cluster them into three groups and then discuss more their applied-abilities.

The first group includes the works using transforms [6], [7] or digital filters [8], [9]. Basically, if the background sounds or noises are identified with high reliability, they can be cleared with a particular filter [10], [11]. This approach is usually applied to some cases when the frequencies of these interfering signals are really high or really low. It can be said that a digital filter is a good choice to clear the noises. In some cases, when the frequency range of the interfering signals intersect with the speech signal, some more modern methods like wavelet transform [12] or filter bank [13] can be applied to deal with the problem. With these approaches, they work better than one singular filter but do not clear completely the backgrounds and noises.

The second group uses components analysis techniques as the main approach. Independent component analysis (ICA) [14]–[16] and its variants [17]–[20] are the representatives for this approach. Different from the digital filter, ICA focuses on separating all the components in the signal so it is usually used when the interfering signals are the background sounds such as music sound or radio sounds. Although it can be applied for difficult cases, this method is not a perfect solution for the SSS problem. Because an ICA model is stored as a matrix, the capacity of the model or the total cases that the model can cover is limit [16], [18]. This means that one particular set of parameters for ICA only works for one particular set of components, or one ICA model cannot deal with an unknown background.

While the first approach focuses on denoising signal and the second approach focuses on separating the background sounds, the third and newest approach focuses on learning the distribution of the speech signal and then reconstructing them. To do this, in 2018 Leglaive *et al.* [21] firstly uses VAE to separate the speech signal of the mixed signal. Because the idea of this approach concentrates on how to learn the distribution of the speech signal [22], [23], the result, different from the two approaches below, does not depend on the background and noises [24], [25]. This means that this solution can be built one time and then used many times with many different interfering signals.

In this research, we inherit the strength of the third and the first approaches to form our solution via a combination. We do not choose the second approach for the combination because it only solves the SSS in particular cases. In the combination, the first component is a VAE which can learn the speech distribution and reconstruct the main content of the speech signal. The second component is a filter that clears all out-of-voiceband in the signal reconstructed by the first component. This component makes our approach different from the pure VAE approach. In a VAE model, the main content of speech is kept and then reconstruct, but with a background that has not existed before, the high frequency is difficultly removed completely. This fact motivates us to apply a filter after processing the signal with a VAE. With this method, we can clear all very high frequencies in the reconstructed signal, or our solution can extract only the speech elements from the mixed signal.

The remaining of this article is structured with 3 main sections. Section 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 via section III. We present a mathematical base, model architecture, and training method for the model in this part. In section IV, we design some experiments to validate our method. After training model, we compare our results with the other works to specify the strengths and weaknesses of our approach.

II. BLIND SOURCE SEPARATION AND SPEECH SOURCE SEPARATION

A. SOURCE SEPARATION IN SIGNAL PROCESSING

Given a mixed-signal, the main work, in this case, is how to separate 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.

1) BLIND SOURCE SEPARATION

Traditional BSS description is formulated to solve the Cocktail party problem [26]. This means there are m sound sources, supposing human sound and background sounds, and n recording devices. In most cases, m is smaller than n, so the whole system is underdetermined and non-linear approaches should be used to reconstruct the 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 $s(t), x(t)$ denote the sets of individual sources and recorded sounds, respectively:

$$s(t) = (s_1(t), s_2(t), s_3(t), \ldots, s_m(t))^T \quad (1)$$

$$x(t) = (x_1(t), x_2(t), x_3(t), \ldots, x_n(t))^T \quad (2)$$

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:

$$x_j(t) = \sum_{i=1}^{m} a_{ji} \times s_i(t), \quad j \in [1, n] \quad (3)$$

All a_{ji} values with $j \in [1, n]$ and $i \in [1, m]$ form a matrix called mixed matrix $A = [a_{ji}]_{M \times N}$ with $A \in \mathbb{R}^{M \times N}$. In practice, each $x_j(t)$ is masked by noise $\gamma_j(t)$, BSS problem can be described by:

$$\hat{x}(t) = (\hat{x}_1(t), \hat{x}_2(t), \hat{x}_3(t), \ldots, \hat{x}_n(t))^T \quad (4)$$

$$\hat{x}_j(t) = \sum_{i=1}^{m} a_{ji} \times s_i(t) + \gamma_j(t) \quad (5)$$

$$\hat{x}(t) = A_{M \times N} \times s(t) + \gamma(t) \quad (6)$$

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.

$$x(t) \approx F(\hat{x}(t)) \quad (7)$$

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

$$s(t) = A^{-1} \times x(t) \quad (8)$$

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

$$\hat{x}_j[n] = \sum_{i=1}^{m} a_{ji} \times s_i[n] + \gamma_j[n] \quad (9)$$

$$\hat{x}[n] = A_{M \times N} \times s[n] + \gamma[n] \quad (10)$$

$$s[n] = A^{-1} \times F(\hat{x}[n]) \quad (11)$$

2) SPEECH SOURCE SEPARATION

In this work, we focus on the SSS problem. In many real-world applications such as speech recognition, speaker recognition, or voice virtual assistant, the end devices receive speech signals from users and then throw the response. 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 of the recorded sound, and that work forms the problem called speech separation.

Different from BSS, we do not consider all elements in the sources $s[n]$ in SSS. We only focus on the speech signal, so we can paraphrase the SSS as a specific case of BSS as separating the mixture into the speech signal and the remaining signal. In some situations, we do not care about the remaining element so speech separation means speech extraction. Figure 1 describes a particular illustration of the speech separation problem. Three waveforms are 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 and Gaussian noise to form the third waveform. In speech separation, the main target is extracting the first signal from the third signal.

B. EVALUATION METHOD

Following by Vincent *et al.* [27], [28], in BSS problem, the estimated signal of a source signal can be described as a mixture of four elements:

$$s_{estimated}(t) = s_{target}(t) + e_{inter}(t) + e_{noise}(t) + e_{artif}(t) \quad (12)$$

or:

$$s_{estimated}(t) = s_{target}(t) + e(t) \quad (13)$$

with:

$$e(t) = e_{inter}(t) + e_{noise}(t) + e_{artif}(t) \quad (14)$$

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

To evaluate the performance of the separation process, Vincent *et al.* [27] proposes many measures including Source to Distortion Ratio (SDR), Source to Interferences Ratio (SIR). They are too similar, the only difference is that SDR reflects total distortion introduced by both interfering signal and processing method, while SIR measures the distortion introduced by the background sound [13]. With $\hat{x}$ and s_{est} are the input and output of the whole model, the total distortion is computed by:

$$D = \frac{||s_{est}[n]||^2 - \left\langle s_{est}[n] \cdot \frac{\hat{x}[n]}{||\hat{x}[n]||} \right\rangle^2}{\left\langle s_{est}[n] \cdot \frac{\hat{x}[n]}{||\hat{x}[n]||} \right\rangle^2} \quad (15)$$

with $||.||$ and $\langle \cdot \rangle$ denote second norm and dot product, respectively. On the other hand, estimated signal s_{est} can also be rewritten as:

$$s_{est}[n] = \left\langle s_{est}[n] \cdot \frac{\hat{x}[n]}{||\hat{x}[n]||} \right\rangle \frac{\hat{x}[n]}{||\hat{x}[n]||} + e \quad (16)$$

So the total energy of noise e is computed via:

$$||e||^2 = ||s_{est}[n]||^2 - \left\langle s_{est}[n] \cdot \frac{\hat{x}[n]}{||\hat{x}[n]||} \right\rangle^2 \quad (17)$$

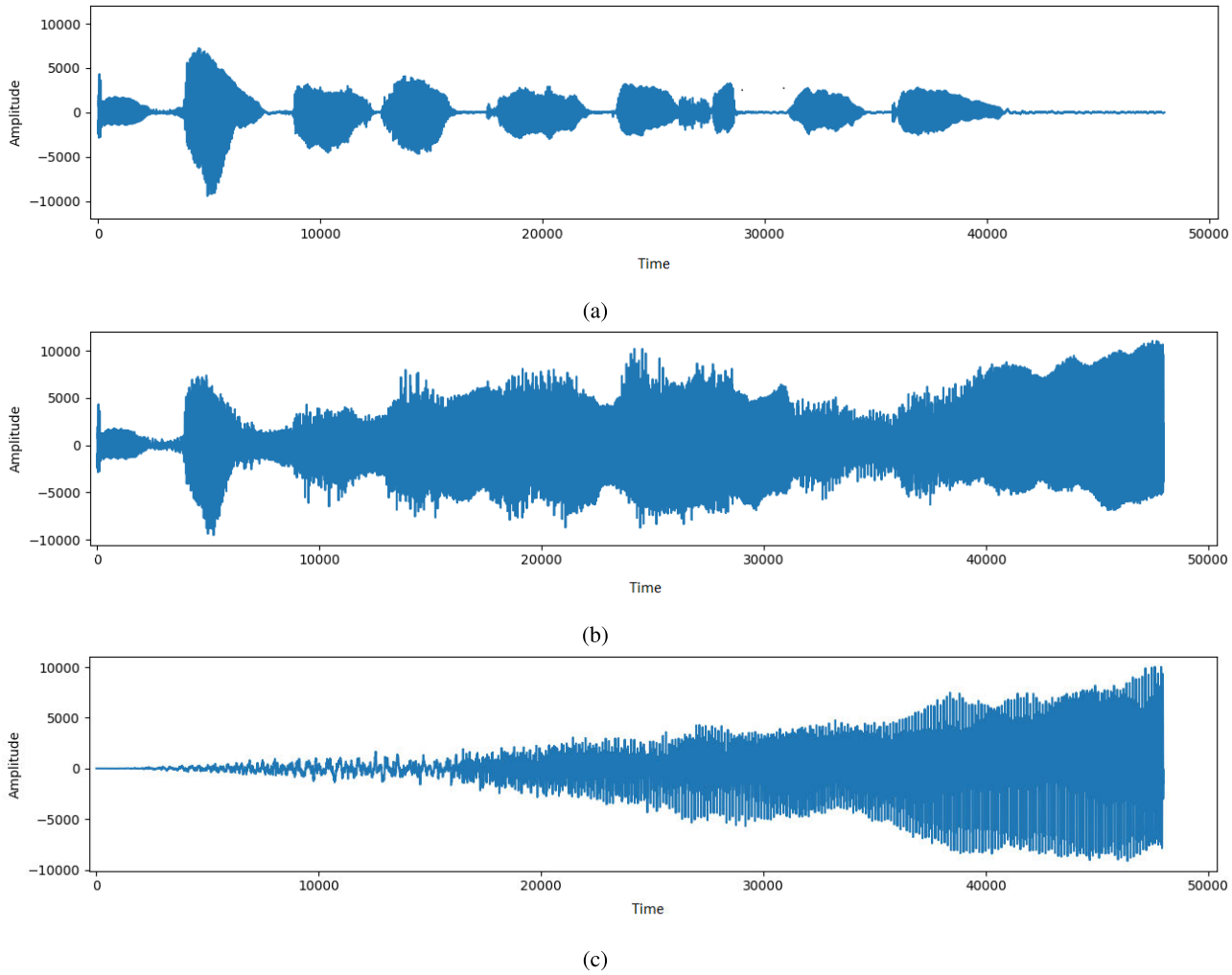

FIGURE 1. Waveform of (a) human speech, (b) trumpet sound as the background signal, and (c) the mixture of human speech, trumpet sound, and Gaussian noise.

When $\langle s_{est}[n] \cdot \hat{x}[n] \rangle \to 0, D \to +\infty$, SDR can be described approximately by:

$$SDR = 10 \times \log_{10} \frac{1}{D} \tag{18}$$

The distortion caused by the interfering signal is:

$$D_{inter} = \frac{||e_{inter}||^2}{\left\langle s_{est}[n] \cdot \frac{\hat{x}[n]}{||\hat{x}[n]||} \right\rangle^2} \tag{19}$$

Therefore, SIR is identified by the equation below:

$$SIR = 10 \times \log_{10} \frac{1}{D_{inter}} \tag{20}$$

In our experiments, the interfering sounds are particular sounds that are masked into the speech sound. So we can compute e_{inter} easily by computing the energy of these sounds and then compute the SDR and SIR.

On the other hand, we also apply the Perceptual Evaluation of Speech Quality (PESQ) [29] to measure the quality of the output speech signal. This measure is an industry-standard and widely applied for voice device manufacturers. Because this research aims to apply the proposed approach for application, we use PESQ to evaluate the output voice.

C. VOICEBAND OF THE SPEECH SIGNAL

Although the audible range of human ears in the frequency domain is from infrasound (20 Hz) to ultrasound (20,000 Hz), the real distribution of speech elements is not uniform. Figure 2 shows a clear illustration of this distort distribution. In this case, the most dense area is from $f_{min} \sim 20\ Hz$ to $f_{max} \sim 4,000\ Hz$. Phonetic researches show the fact that most of content in the speech spreads in the range under $f \sim 3,400\ Hz$ [30], [31]. The number f may be different based on the researches, but it always concentrates on value $3,400\ Hz$. This range is usually called voice-band.On the other hand, the frequency elements under $f_{max} \sim 3,400\ Hz$ presents the speaker properties or other suprasegmental features such as dialect or emotion. 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 $f_{begin} = 20\ Hz$ to $f_{end} = 5,000\ Hz$ to commit that we

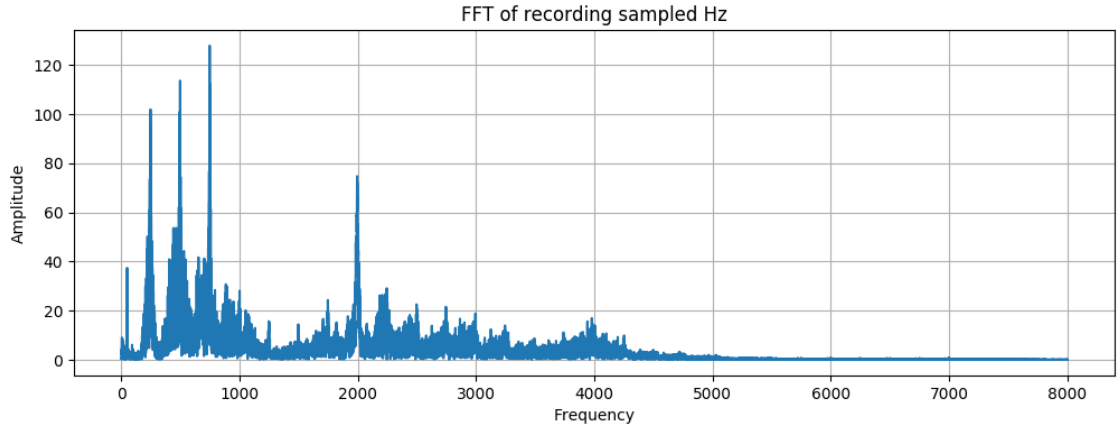

FIGURE 2. Frequency distribution of a human speech sample.

capture all information from the input signal in the frequency domain.

III. OUR APPROACH FOR SSS: COMBINING A VAE AND A BPF

Our solution for the BSS problem is the combination of a VAE network with a Chebyshev filter in the frequency domain. The mixed signal is transformed by Short Time Fourier Transform (STFT), then pushed to the processing block, and finally is computed by the Inverse Short Time Fourier Transform (ISTFT) algorithm to reconstruct into the time domain. The illustrations for this whole process is described in figure 3.

From left to right, respectively, the blocks are source signal s, mixture $\hat{x}$, processing blocks, and estimated signal s_{est} at the end. The model mainly processes input in the frequency domain, then reconstructs the signal to time-domain via the ISTFT algorithm. Finally, the performance and quality of the model are validated by comparing the estimated signal and the original speech signal.

A. SHORT TIME FOURIER TRANSFORM

Most of the properties of human speech are represented effectively in the frequency domain instead of the time domain. The content of speech, genders, emotions, and dialects are recognized by the combination of the high-power frequency elements. This leads us to decide to transform the input signal from the time domain to the frequency domain via Fourier transform and mainly solve the separation problem in this domain.

STFT is present by the formula:

$$F(\tau, \omega) = \int_{-\infty}^{\infty} f(t) w(t-\tau) e^{-i\omega t} dt \tag{21}$$

In the STFT formula, the left part presents the amplitude of each element in the time-frequency domain. Particularly, $F(\tau, \omega)$ is the amplitude at time τ of frequency ω. On the other hand, the window function $w[.]$ in the right part defines where and how the sub-range of the signal is taken to present into the frequency domain. In this research, we use Blackman - Harris function, a special case of Hamming function.

From the frequency domain, the signal is converted to time domain using ISTFT via the equation below:

$$f(t) = \frac{1}{2\pi \times w(t-\tau)} \int_{-\infty}^{\infty} F(\tau, \omega) e^{i\omega t} d\omega \tag{22}$$

B. VARIATIONAL AUTOENCODER

In the frequency domain, we aim to transform or convert the input (speech signal with noise) to the output (clean speech signal). We build a machine learning model that can identify if a harmonic element belongs to the original speech signal or noise and then try to keep all elements of the speech signal. With this approach, the model can filter and hold most of the useful information to reconstruct the original speech. In this work, an autoencoder's well-known variant VAE is used as the main processor for this process.

Autoencoder [32], [33] is a special kind of neural network which is usually used to extract features or denoise the input. Ideally, the input and output of autoencoders are the same because this network aims to compress the input data and then reconstruct it. To do this, the network contains two sub-network named encoder and decoder and these two networks are linked by a small layer called Code which is smaller than the input and output layers. 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 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.

Let $\Theta, \Phi, x, h, y, \sigma, w, b$ denote data space, code space, input, code value, output, activation function, weight, and bias. A 3-layer autoencoder can be formulated as follow:

$$Encoder : \Theta \rightarrow \Phi, h = \sigma_{in}(w_{in} \times x + b_{in}) \tag{23}$$

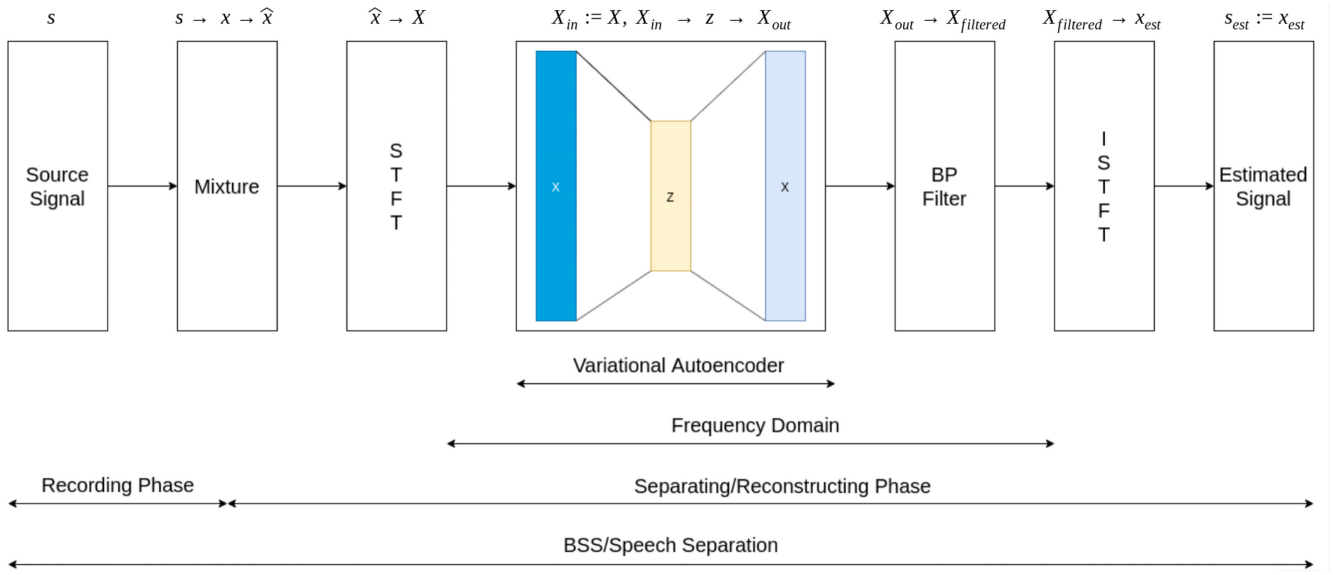

FIGURE 3. Block diagram for the hybrid approach to separate speech signal from the mixture.

$$Decoder : \Phi \rightarrow \Theta, y = \sigma_{out}(w_{out} \times h + b_{out}) \tag{24}$$

$$Encoder - Decoder : \Theta \rightarrow \Phi y = \sigma_{out}(w_{out} \times (\sigma_{in}(w_{in} \times x + b_{in})) + b_{out}) \tag{25}$$

Then the loss function is:

$$\begin{aligned} Loss(x) &= ||x - y||^2 \\ &= (x - \sigma(w_{out} \times (\sigma(w_{in} \times x + b_{in})) + b_{out}))^2 \end{aligned} \tag{26}$$

Particularly, the loss value for ith point is computed by adding the point index to the equation below:

$$\begin{aligned} Loss(x_i) &= ||x_i - y_i||^2 \\ &= (x_i - \sigma(w_{out} \times (\sigma(w_{in} \times x_i + b_{in})) + b_{out}))^2 \end{aligned} \tag{27}$$

In real application design, an autoencoder can contain $n > 1$ layers in both encoder and decoder sub-networks. In this case, the formulas to compute the values at the hidden layers are similar to the formula (23) 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:

$$\begin{cases} h_1 = \sigma_{in}(w_{in} \times x + b_{in}) \\ h_{i+1} = \sigma_i(w_i \times h_i + b_i), i \geqslant 1 \end{cases} \tag{28}$$

with h_i, σ_i, w_i, b_i are the value, activation function, weight, and bias at ith hidden layer. Assuming there are n layers in the autoencoder including one input layer, one output layer, and $n - 2$ hidden layers, we denote $p(.)$, $q(.)$ the encoder and decoder networks.

Generally, the loss function is described by:

$$Loss(x) = ||x - y||^2 = (x - q(p(x)))^2 \tag{29}$$

or:

$$Loss(x_i) = ||x_i - y_i||^2 = (x_i - q(p(x_i)))^2 \tag{30}$$

for ith sample. With activation function, we use Leaky ReLU [34], [35] for all layers except at the output layer:

$$\begin{cases} \sigma_i(x) = x, x > 0 \\ \sigma_i(x) = k \times x, x < 0, k = 0.01 \end{cases} \tag{31}$$

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:

$$\sigma_{out}(x) = x \tag{32}$$

In an autoencoder model, the value at the code layer of a particular input is a fixed vector, but this representation does not reflect correctly the truth in the real world. Let us consider a human sound like /s/ each person pronounces this sound differently, but the signals have some similar properties to help people recognize exactly the sound. If the sound is represented by a probability distribution, it describes the signal better in comparison with a fixed number or a vector. That is the idea of a VAE, a more powerful variant of the autoencoder. Different from traditional autoencoders, the code layer in VAE is supposedly created by a prior distribution which is computed from the encoder network. In most real cases, the prior distribution is Gaussian or normal distribution, so the value at the code layer can be sampled from $\mathcal{N}(\mu, \Sigma)$ as the left illustration in figure 4 [36].

Let X, $Q(.)$, z, $P(.)$ and f denote input, encoder network, code value, decoder, and output. When training the network, if z is sampled from a distribution, it is a random variable. This leads to the fact that encoder $Q(.)$ can not be updated its parameters via the backpropagation algorithm, so the network cannot be learned. To solve this problem, z is computed as follow:

$$z = \mu + \epsilon \times \Sigma \tag{33}$$

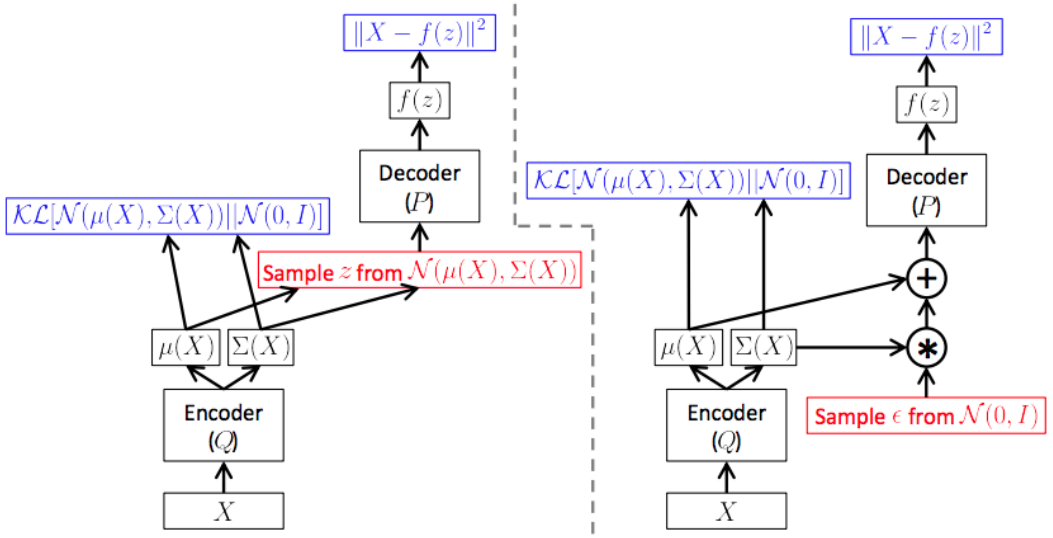

FIGURE 4. Theoretical and real form of variational autoencoder.

with ϵ is a new random variable sampled with $\mathcal{N}(0, 1)$ distribution. With this trick, the error from z can be propagated to encoder $Q(.)$ through μ, Σ, which are computed the error easily via z.

of VAE [36] are to reconstruct the input and to maintain the Gaussian distribution in the code layer, so the loss function at ith point is the summary of these two elements:

$$Loss_{\phi,\theta}(x_i) = E_{z\sim q\theta(x_i|z)} \log p_\phi(x_i|z) - \mathcal{KL}(q_\theta(z|x_i)||p_\phi(z) \quad (34)$$

with $\mathcal{KL}(q(x)||p(x))$ is the Kullback-Leibler divergence [37], [38] of two probability distribution functions. This is the measure used to compute the similarity of these two distributions:

$$\mathcal{KL}(q(x)||p(x)) = \sum_{-\infty}^{\infty}(q(x) \times \log \frac{q(x)}{p(x)}) \quad (35)$$

or:

$$\mathcal{KL}(q(x)||p(x)) = \int_{-\infty}^{\infty}(q(x) \times \log \frac{q(x)}{p(x)})dx \quad (36)$$

When $z = \mu + \epsilon \times \Sigma$ and $\epsilon \sim \mathcal{N}(0, 1)$, loss function is rewritten as follow:

$$Loss_{\phi,\theta}(x_i) = \frac{1}{L}\sum_{l=1}^{L} \log p_\phi(x_i|z_i^l) + \frac{1}{2}\sum_{j=1}^{J}(1 + \log((\Sigma_i^j)^2) - (\mu_i^j)^2 - (\Sigma_i^j)^2) \quad (37)$$

with L, J are the length of code layer and output layer, respectively.

Finally, the total loss of VAE is the sum of (26) and (37) formulas, so it can be rewritten as follow:

$$Loss_{VAE} = Loss(x_i) + Loss_{phi,\theta}(x_i) \quad (38)$$

Because VAE is a kind of multi-layer neural network, there is nothing different in training and inferring processes in comparison with a normal neural network. We use the Backpropagation algorithm [39], [40] to adjust all parameters in the network in the training phase and forward propagation for inference.

We use VAE to process the signal in the frequency domain. This means the input of VAE 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.

In setting, our network is different from the network below as our network is a little bit changed. The target output is not the same as the input. In our model, the input is the mixed signal when the output is the pure speech signal. Let us say this model receives the mixture between the main signal, in this case, it is a speech signal, with some unexpected signal, and process it to return the original speech signal. The details of our design are specified in the experimental result.

The inferring speed depends on the complicated of the model including the VAE network and the filter. Because the filter runs with a fixed cost, the model complexity mainly depends on the VAE network. Let n, k denote the number and the maximum size of layers in this network. An input sample, correspond with a k-dimension vector, will be passed through n multiplication between a k-dimension vector with a kxk-dimension matrix. The cost for each multiplication is $O(k^2)$ so the total computation complexity is $O(nk^2)$. Most of the cases, because n is not a large number, the model is not complicated in computation.

C. BANDPASS FILTER

After reconstructing speech signal via the autoencoder network, we use a BPF to eliminate all frequency elements which are out of the common range of human speech. In particular, most human speech signal spreads in the range from 50 Hz to

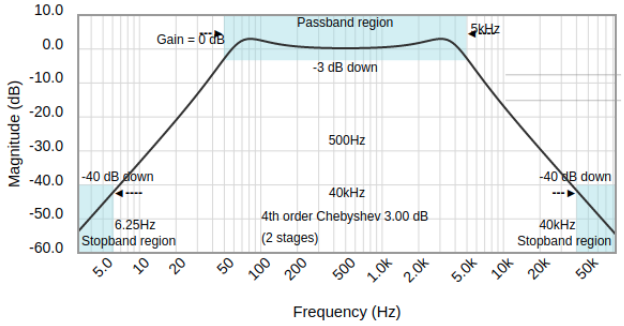

FIGURE 5. Gain response for 4th order Chebyshev Bandpass filter.

5000 Hz. With this process, the main energy of the speech signal is kept while the remaining elements are ignored. We choose filter Chebyshev type 1 [41] for this research because it is good enough and fast processing.

The BPF is designed to pass the signal through the band 50 Hz and 5000 Hz. This is the combination of a high pass filter with the cutoff frequency is 50 Hz and a low pass filter with the cutoff frequency is 5000 Hz. The remaining of this section describes in detail the low pass filter while the high pass filter is designed as a similar method.

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

$$G_n(\omega) = H_n(j\omega) = \frac{1}{\sqrt{1 + r^2 T_n^2(\frac{\omega}{\omega_0})}} \tag{39}$$

with r is the ripple factor, T_n is nth order Chebyshev polynomial, and ω_0 is the cutoff frequency. Parameter r is determined by:

$$r = \sqrt{10^{\frac{\Phi}{10}} - 1} \tag{40}$$

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 nth order Chebyshev polynomial [42], which is a recursion function:

$$\begin{cases} T_0(x) = 1 \\ T_1(x) = x \\ T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x), n > 0 \end{cases} \tag{41}$$

IV. EXPERIMENTAL RESULT

A. DATASET

In this work, we use the trumpet sounds, water sounds, and traffic sounds as the background sounds. Particularly, the backgrounds are chosen because of their properties. First, we evaluate our model with a clear and clean background so we choose an instrument sound (trumpet in this case). The spectrum of this background has a clear and stable distribution. It is different from human speech distribution, so this test is the easiest case to evaluate our method. Second, we test our approach with an unclean background: we choose many kinds of water sound including waterfall, rain, stream, and the running water at the faucet. The distributions of water in these cases are not stable and sometimes interfere with the speech signal. This test is more difficult than the test with an instrument. The final background is a more complicated sound: we choose traffic sounds. This kind includes many different sources such as engine sounds, car horns sound, blowing wind, etc. The recorded sound is a mixture of many known elements and unknown elements. This is the most difficult test case for our method.

To present the main speech signal, we use the TIMIT [43]. 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 background sound to form the mixed sound. After that, the mixed sound is mixed with random Gaussian noise to create the mixture. This sound is quite similar to the sound in the real world, where the recorded sound is always masked by random noise during the recording process. Let us assume with a speech signal $s[n]$, we get its mixture $\hat{x}[n]$. Thus, we can represent a pair of input and output of the whole model as follow:

$$input, output = \hat{x}[n], s[n] \tag{42}$$

The background sounds are cloned into many versions with many different powers by multiplying the sound signal with an array of real random 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.

On the other hand, we also use VIVOS dataset [44], a commonly used dataset for Vietnamese speech recognition to check whether our proposed approach can work independently with the language. Basically, this dataset is organized similarly to TIMIT. The differences between these two datasets include the languages, number of recorders, and the recording per recorder. In experiments, we use a random subset of these datasets instead of using all of their recordings. This way helps us to validate the model many times with many distinguished test cases.

In the testing phase, we mixed the background sounds and Gaussian noise into the speech sounds with 3 *dB* and 10 *dB* in terms of SDR. We choose these parameters to simulate the common environment in the real world.

B. HYPERPARAMETER FOR MODEL

At STFT block, we set the size for window function 20-millisecond. Two consecutive frames overlap 10 milliseconds. Each frame then multiplies with the Backman-Harris

window function. Next, the frame is transformed by STFT with 128 factors. These factors are the complex numbers, then we replace them with 256 factors including 128 real and 128 image coefficients. Then the signal is represented by an array of 256-dimension vectors.

We use many configurations for our autoencoder with the size of layers is diversity. Supposing we are processing a frame of a sample, let us denote the input and output of autoencoder network by:

$$input, output = X_i^j, S_i^j \tag{43}$$

with X and S are the representations for the input and output in 42. Besides, i and j are the numerical index of speech signal sample and the numerical index frame in the sample. This means that the autoencoder transforms STFT values from the mixture into a pure speech signal.

After reconstructing the speech signal by VAE, we apply the Chebyshev BPF with 4th order to clear all out of range frequency elements. We set passband ripple $\Phi = 3dB$ and stopband $-40dB$. Besides, the low frequency f_{min} is set at 20 Hz and the high frequency f_{max} is at 5000 Hz.

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 chose randomly ten people with one hundred utterances. Then we mixed all of these utterances with the trumpet sounds and Gaussian noise and used 90 samples in the training phase, 10 samples in the testing phase. After the model had been convergent, we reconstructed ten remaining samples to get the output signals and finally compared them with the ground truths.

As can be seen in table 1, all 8 cases show that our model can extract effectively the speech sound out of the mixture because the SIR and SDR are positive. Although the results are good, they are not stable. In dialect 4, the result is too high, while in dialect 5 and 8, the results are much lower. In speech separation, the difficulty of the problem is represented by the difference between the human sound and the background sounds. If the properties and the distributions of these sounds are similar, the model separates them in an imperfect way. In dialect 5 and 8, their sounds have the high tones and the distribution in each range in the frequency domain is overlapped partly with the trumpet sounds. It leads to the fact that the results are not good in this case. This similarity also means that with a particular dialect, the optimal configuration for VAE and hyperparameters for are not the same, and we should try with many configs to find the best solution for a particular dialect.

2) MANY DIALECTS VERSUS ONE BACKGROUND

In this experiment, we chose randomly 100 utterances from TIMIT and then mixed them with trumpet sounds with many different amplitudes and Gaussian noise. After that, we processed them with our proposed model to extract speech signal. The particular results are shown in table 2.

TABLE 1. Evaluation in TIMIT dataset with total 8 dialect sounds.

Dialect	SDR (dB)	SIR (dB)
1	8.75	11.02
2	6.11	8.20
3	6.80	7.54
4	12.34	13.02
5	4.28	4.35
6	9.61	10.11
7	9.36	8.69
8	4.14	4.92
Average	7.42	8.2

TABLE 2. Evaluation in TIMIT dataset with mixed dialect sounds.

Test case	SDR	SIR
1	8.65	9.83
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

The results in this experiment are generally better than in the previous experiment. This can be explained by the fact that the data distribution in these experiments is not the same. When training with one dialect, the model biased to that dialect and fell into a non-universal solution. If the model is trained with many different types of dialects, it can learn much unequal distribution from data. On the other hand, some dialects are more common and then include more samples than the others. When we selected randomly from the whole dataset, the samples per dialect were not balanced, followed by the real distribution. This leads to the fact that the result, in this case, is not equal to the previous experiment, and particularly, better and closer to the real applications.

3) ONE DIALECT VERSUS MANY BACKGROUNDS

The main difference between a VAE and a filter or an ICA model is what the model learns. While the filter and the ICA model learn how to clear the interfering signal, the VAE learns the distribution of speech signals. This fact leads VAE can works within many different kinds of background sounds and noises because the VAE model considers these signals as the remaining signal after the separation process. In a mixture, the VAE can identify exactly where the speech signal is and then extract it out of the mixed signal. To evaluate this ability of VAE, in this experiment, we masked the speech sounds by many kinds of backgrounds and Gaussian noise and then performed them with the proposed model.

Table 3 shows that all SDR and SIR values are positive. Although there is a difference between the results of the different backgrounds, this difference is not significant. The result of this experiment demonstrates that the proposed approach does not depend on the interfering sounds.

4) MANY DIALECTS VERSUS MANY BACKGROUNDS

As the same purpose with experiment 3, in this test, we would like to check if the proposed model can learn the human

TABLE 3. Evaluation in TIMIT dataset with a mixed of 3 background sounds.

Dialect	Trumplet SDR/SIR	Water SDR/SIR	Traffic SDR/SIR	All 3 sounds SDR/SIR
1	8.75/11.02	6.57/8.82	6.75/10.27	6.09/7.01
2	6.11/8.20	7.28/8.12	7.11/9.05	4.29/6.87
3	6.80/7.54	6.78/7.14	9.80/9.98	4.01/6.44
4	12.34/13.02	9.78/11.25	9.34/10.04	7.22/9.59
5	4.28/4.35	5.22/6.15	6.28/7.36	4.82/5.06
6	9.61/10.11	8.11/11.92	8.61/10.54	6.17/6.70
7	9.36/8.69	9.36/10.49	7.36/8.98	5.51/9.03
8	4.14/4.92	7.15/9.24	6.18/8.21	4.58/4.99
Average	7.42/8.2	7.53/9.14	7.68/9.30	5.34/6.96

TABLE 4. Evaluation in TIMIT dataset with mixed dialect and background sounds.

Test case	Trumpet SDR/SIR	Water SDR/SIR	Traffic SDR/SIR	All 3 sounds SDR/SIR
1	7.02/8.54	6.11/7.51	6.43/6.97	5.16/8.44
2	6.97/8.02	5.22/7.47	6.61/8.18	4.89/7.90
3	6.31/7.80	6.12/6.99	7.01/7.93	5.22/6.90
4	6.50/8.32	6.90/8.21	6.10/7.87	5.26/7.07
5	5.89/7.71	6.54/8.82	6.27/8.35	5.11/7.38
Average	6.54/8.08	6.18/7.80	6.48/7.86	5.13/7.54

speech distribution or not. In the previous experiment, we only used one dialect for each test case. That test was much easier in comparison with this test because the distribution of all 8 dialects was more complicated than each dialect. If the model worked well in this test, it could be inferred that this approach could be extended and applied for many real applications.

The result in table 4 shows that the combination between a VAE and a BPF can be applied for many dialects and many backgrounds. Despite the fact that the values of SDR and SIR meaures are not so good, the result is stable in many test cases. This is a evidence to infer the proposed model can be used in the real applications.

5) SPEED OF SEPARATING PROCESS

In this test case, we implemented some different configurations for our VAE to show the relationship between the complexity, speed, and performance of our models. In these configurations, the middle layers were the latent layers or code layers z. The number shown in the table 5 are the number of latent dimensions, not the real number in use. Particularly, the size of the code layer is twice the size in table 5 because each latent dimension requires two parameters including a number for the mean and a number for the variance. For example, in case 1, the configuration [256, 80, 35, 80, 256] means that the size of the code layer is 35 dimensions with 70 nodes including 35 nodes for means and 35 nodes for variances.

We reused the data used in the first case in the experiment IV-C2. Particularly, that set contains 100 utterances, which were randomly chosen from the TIMIT dataset. In the training phase, we used 90 samples of them and then inferred 10 remaining ones. All testing samples are extracted randomly 1-second per sample before passed into the processing block. In this experiment, we used a CPU Intel Core i5 3.0 GHz with 8GB RAM to evaluate the processing speed of proposed method. The results with many different configurations are shown in table 5.

In table 5, from the results in these five test cases, we can conclude that the performance of our approach mainly depends on two factors including the depth of VAE and the size of the code layer z. The deeper VAE, the better result, and we should choose carefully the size for code space. If the size of the code is too big, the model is bigger and then run slower. It also presents the data into a sparse space so the reconstructing phase works ineffectively. If the size is too small, it cannot memorize all the needed information to reconstruct the original signal. To choose the optimal size, we need to analyze the data and then try many times with many different configs.

6) A TEST CASE WITH WHOLE TIMIT DATASET

In this test, we chose the 3rd configuration for VAE in experiment IV-C5 for training and testing. We used the whole TIMIT for this test case. Particularly, we used the default separation of TIMIT for the training and testing process.

From table 6, in comparison with the other methods including wavelet [12], time-frequency filter bank [13], ICA [17], and VAE [25], our model gets a high result. We achieve 12.99 *dB* in SDR measure, which is the highest score, and 15.02 *dB* in SIR. These results are good evidence to demonstrate the efficiency of the proposed hybrid approach.

In our model, the VAE component plays the main role in the separating process. When we only apply VAE for SSS, the total distortion SDR is nearly approximate the best result by [13]. The result by only BPF is much lower than only VAE in terms of both SDR, SIR, and PESQ. If these two components are combined to form the full model, it achieves a higher PESQ than the result at [25].

7) AN IMPLEMENTATION FOR MULTILINGUAL SPEECH

This experiment is aimed to check whether the VAE approach is dependent or independent with the language. In this test, we also used the VIVOS dataset instead of only TIMIT. This dataset is commonly used for Vietnamese speech recognition research. It contains over 28.000 utterances which are recorded from nearly 50 people. We implemented this test in the same way with the test IV-C2. We chose randomly five subsets from VIVOS with 100 samples per set, which is equal to each test case with TIMIT. Then we divided them into the training set and testing set with the same size as the first experiment. We also mixed 50 random samples from the VIVOS subset with 50 random samples from the TIMIT subset and then used them as the third data subset. The particular results are shown in figure 7.

From table 7, the results show us the fact that the proposed model can extract speech signals from the mixture efficiently. Both SDR and SIR measures are positive and high in all

TABLE 5. Evaluation in TIMIT dataset with many VAE configurations.

Test case	VAE config	Code size	Speed (ms/sample)	SDR (dB)	SIR (dB)
1	[256, 80, 35, 80, 256]	35	0.37	6.12	8.24
2	[256, 100, 50, 35, 50, 100, 256]	35	0.66	7.81	9.98
3	[256, 100, 50, 20, 50, 100, 256]	20	0.59	8.65	9.83
4	[256, 150, 80, 50, 20, 50, 80, 150, 256]	20	1.17	9.01	9.95
5	[256, 150, 80, 50, 10, 50, 80, 150, 256]	10	1.15	7.40	7.72

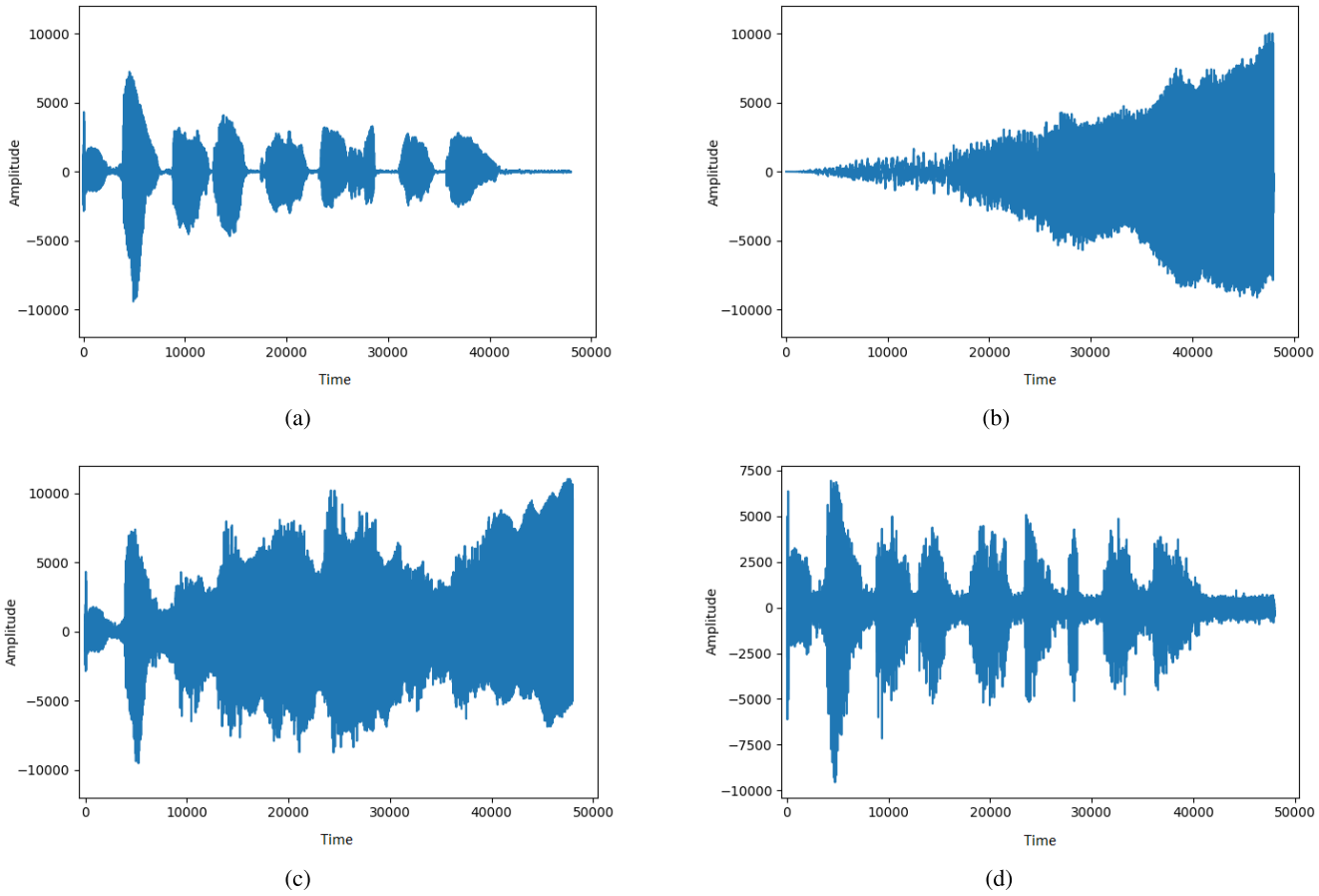

FIGURE 6. Waveform of (a) human speech, (b) trumpet sound, (c) the mixture, and (d) the reconstructed speech signal.

TABLE 6. Evaluation on whole TIMIT dataset.

Group	Model	SDR (dB)	SIR (dB)	PESQ
1	Wavelet	7.56	16.22	-
	Time-Frequency filter bank	9.47	1.09	-
2	ICA	5.98	11.92	-
3	VAE	9.47	-	2.35
Our Approach	VAE (Our implementation)	9.08	14.76	2.02
	BPF (our implementation)	6.87	10.01	0.97
	VAE + BPF	12.99	15.02	2.41

TABLE 7. Evaluation in multilingual dataset.

Test case	VIVOS SDR/SIR	TIMIT SDR/SIR	VIVOS + TIMIT SDR/SIR
1	6.05/10.93	8.65/9.83	6.33/9.85
2	5.49/12.11	9.43/12.01	6.01/10.81
3	8.84/10.51	9.14/10.35	8.21/10.09
4	4.81/10.06	8.98/11.56	5.23/9.44
5	7.17/11.72	8.87/12.54	7.18/10.54
Average	6.47/11.07	9.01/11.26	6.59/10.17

cases. This means that our approach does not depend on the languages and only depends on the distribution of the data in the dataset.

Although the experimental results when testing the model with TIMIT and VIVOS are positive, the particular result in VIVOS is lower a little bit than TIMIT. This can be explained as follow: Vietnamese (the language in VIVOS dataset) is complicated in phonetics and phonology aspect. It contains 6 tones, so in spoken language, people use more high-frequency signal to express the tone. This leads to the fact that the total energy in Vietnamese speech spread wider than in English (the language in TIMIT). To get a better result in Vietnamese, we need to change the configuration with a bigger size for the code layer and higher f_{max} in the filter. Generally, our proposed model can be applied to many

TABLE 8. Accuracy of gender classification via different recognizers.

Recognizer	Clean (%)	Mixture (%)	Separated speech (%)
Logistic Regression	95.84	90.01	94.74
Linear SVM	95.81	89.65	95.14
RBF SVM	96.44	91.54	95.70
KNN	96.23	92.22	94.10
Random Forest	96.45	89.73	96.14
Decision Tree	93.54	89.98	92.09
Gradient Boosting	96.25	93.41	94.81
Gaussian NB	94.64	90.81	94.02
Neural Net	95.23	90.67	93.77
Average	95.60	90.89	94.50

languages with a little change in the model configuration. This fact can be paraphrase as our method is mostly independent of language.

On the other hand, the lowest performance belongs to the mixed dataset. Generally, the distribution of English (in TIMIT) and Vietnamese (in VIVOS) are not the same in the frequency domain. This leads to the fact that the model learns the general distribution more hardly. Due to this not excellent result, both SDR and SIR is positive, this means our model can be applied for some case when there is more than one language in the speech.

In figure 6, we describe 4 waveforms including waveform of the original speech, background sound, mixture, and reconstructed sound. As can be seen, the reconstructed, or estimated signal has the form too similar to the original form and very different from the mixture. This demonstrates that our model can reconstruct the original speech from the mixture so well.

8) APPLYING SSS PROPOSED MODEL TO GENDER RECOGNITION

We integrated our model as a preprocessing component into a gender recognition system. Many current applications such as voicebot or recommendation systems use the information of the user such as gender to suggest suitable content. To evaluate the performance of our model, we compared three tests: clean, mixture, and reconstructed data. In this experiment, we used a training set from TIMIT for training. Then we mixed the TIMIT test set with trumpet sounds and Gaussian noise to form the mixture. We finally separated the speech out of the signal by our proposed method. All of these three kinds of test samples were passed to the recognizer to verify whether our model works or not. The particular results are shown in table 8.

In table 8, the results of 9 recognizers are too similar. This means that our model is stable and usable for many different algorithms. When we recognize the mixture, due to the impact of the background and noise, the accuracy of recognizers is much lower than the test with clean data. With the data which are processed by our model, the results are improved significantly and reach nearly the test with clean data. This experiment is evidence that our approach can be used for a real application.

V. CONCLUSION

In this work, we propose a new design for the speech separation problems using a combination between a variational autoencoder (VAE) and a bandpass filter (BPF). With this combination, our model can clear the interfering signal and noise in not only out of voiceband but also intersect with the speech signal. Particularly, we use VAE, a generative model, to reduce the impact of intersection elements on the main signal and BPF to clear all out of voiceband elements. The experimental results show that our approach is more effective than many works before. In many tests, our model shows its good results in both signal to distortion ratio (SDR), signal to interference ratio (SIR), and perceptual evaluation of speech quality (PESQ) measures with high positive values. It works on many kinds of dialects, many kinds of backgrounds, and noises. On the other hand, because this approach does not depend on language, it can be re-configurated and tuned to deal with many real applications. Finally, our last experiment shows that our approach can be used in a preprocessing component in a real application like gender classification and it works stably with many different algorithms in the main model.

REFERENCES

[1] B. Rivet, W. Wang, S. M. Naqvi, and J. A. Chambers, "Audiovisual speech source separation: An overview of key methodologies," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 125–134, May 2014.

[2] P. Comon and C. Jutten, eds, *Handbook of Blind Source Separation Independent Component Analysis and Applications*. New York, NY, USA: Academic, 2010.

[3] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[4] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.

[5] Y. Li, K. A. Lee, Y. Yuan, H. Li, and Z. Yang, "Many-to-Many voice conversion based on bottleneck features with variational autoencoder for non-parallel training data," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2018, pp. 829–833, doi: 10.23919/APSIPA.2018.8659628.

[6] B. M. Mahmmod, A. R. Ramli, S. H. Abdulhussian, S. A. R. Al-Haddad, and W. A. Jassim, "Low-distortion MMSE speech enhancement estimator based on Laplacian prior," *IEEE Access*, vol. 5, pp. 9866–9881, 2017, doi: 10.1109/ACCESS.2017.2699782.

[7] B. M. Mahmmod, A. R. Ramli, T. Baker, F. Al-Obeidat, S. H. Abdulhussain, and W. A. Jassim, "Speech enhancement algorithm based on super-Gaussian modeling and orthogonal polynomials," *IEEE Access*, vol. 7, pp. 103485–103504, 2019, doi: 10.1109/ACCESS.2019.2929864.

[8] S. S. Bhogeshwar, M. K. Soni, and D. Bansal, "Design of simulink model to denoise ECG signal using various IIR FIR filters," in *Proc. Int. Conf. Rel. Optim. Inf. Technol. (ICROIT)*, Feb. 2014, pp. 477–483, doi: 10.1109/ICROIT.2014.6798370.

[9] B. B. Ahamed, D. Yuvaraj, and S. S. Priya, "Image denoising with linear and non-linear filters," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE)*, Dec. 2019, pp. 806–810, doi: 10.1109/ICCIKE47802.2019.9004429.

[10] J. Le Roux and E. Vincent, "Consistent Wiener filtering for audio source separation," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 217–220, Mar. 2013, doi: 10.1109/LSP.2012.2225617.

[11] H.-Y. Li, Q.-H. Zhao, J.-Q. Zhao, and B.-J. Xiao, "Blind separation of noisy mixed images based on Wiener filtering and independent component analysis," in *Proc. 2nd Int. Congr. Image Signal Process.*, Oct. 2009, pp. 1–5, doi: 10.1109/CISP.2009.5301437.

[12] G. Wolf, S. Mallat, and S. Shamma, "Rigid motion model for audio source separation," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1822–1831, Apr. 2016.

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

[14] T. Takatani, T. Nishikawa, and H. Saruwatari, "Blind source separation based on binaural ICA," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Sep. 2003, pp. 421–429, doi: 10.1109/ICASSP.2003.1199940.

[15] H. A. Inan and A. T. Erdogan, "Convolutive bounded component analysis algorithms for independent and dependent source separation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 697–708, Apr. 2015, doi: 10.1109/TNNLS.2014.2320817.

[16] T. Nishikawa, H. Saruwatari, and K. Shikano, "Fast-convergence blind separation of more than two sources combining ICA and beamforming," in *Proc. Abstracts. IEEE-Eurasip Nonlinear Signal Image Process. NSIP*, May 2005, p. 17, doi: 10.1109/NSIP.2005.1502238.

[17] Z. Koldovsky and P. Tichavsky, "Time-domain blind separation of audio sources on the basis of a complete ICA decomposition of an observation space," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 2, pp. 406–416, Feb. 2011, doi: 10.1109/TASL.2010.2049411.

[18] J.-C. Chao and S. C. Douglas, "Using piecewise linear nonlinearities in the natural gradient and FastICA algorithms for blind source separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2008, pp. 1813–1816, doi: 10.1109/ICASSP.2008.4517984.

[19] L. Yang, Z. Ming, and J. Longbin, "Blind source separation based on FastICA," in *Proc. 9th Int. Conf. Hybrid Intell. Syst.*, Aug. 2009, pp. 475–479, doi: 10.1109/HIS.2009.212.

[20] S.-G. Li, Z.-H. Li, Y.-L. Wang, Y. Liu, T.-T. Chen, S.-L. Tan, Z. Su, M. Gao, F. Jiang, and H.-L. Li, "Single channel blind source separation for gas Regulators' acoustic signal using eemd-fastica," in *Proc. 14th Symp. Piezoelectrcity, Acoustic Waves Device Appl. (SPAWDA)*, Nov. 2019, pp. 1–4, doi: 10.1109/SPAWDA48812.2019.9019253.

[21] S. Leglaive, L. Girin, and R. Horaud, "A variance modeling framework based on variational autoencoders for speech enhancement," in *Proc. IEEE 28th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2018, pp. 1–6, doi: 10.1109/MLSP.2018.8516711.

[22] S. Inoue, H. Kameoka, L. Li, S. Seki, and S. Makino, "Joint separation and dereverberation of reverberant mixtures with multichannel variational autoencoder," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 96–100, doi: 10.1109/ICASSP.2019.8683497.

[23] S. Seki, H. Kameoka, L. Li, T. Toda, and K. Takeda, "Underdetermined source separation based on generalized multichannel variational autoencoder," *IEEE Access*, vol. 7, pp. 168104–168115, 2019, doi: 10.1109/ACCESS.2019.2954120.

[24] E. Karamatli, A. T. Cemgil, and S. Kirbiz, "Audio source separation using variational autoencoders and weak class supervision," *IEEE Signal Process. Lett.*, vol. 26, no. 9, pp. 1349–1353, Sep. 2019, doi: 10.1109/LSP.2019.2929440.

[25] S. Leglaive, X. Alameda-Pineda, L. Girin, and R. Horaud, "A recurrent variational autoencoder for speech enhancement," *Proc. IEEE Int. Conf. Acoustic Speech Signal Process. ICASSP*, Barcelona, Spain, May 2020, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/9053164, doi: 10.1109/ICASSP40776.2020.9053164.

[26] P. G. Parande and T. G. Thomas, "A study of the cocktail party problem," in *Proc. Int. Conf. Electr. Comput. Technol. Appl. (ICECTA)*, Nov. 2017, pp. 1–5, doi: 10.1109/ICECTA.2017.8251979.

[27] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

[28] E. Vincent, M. Jafari, and M. Plumbley, "Preliminary guidelines for subjective evalutation of audio source separation algorithms," in *Proc. UK ICA Res. Netw. Workshop*, Southampton, U.K., 2006, pp. 1–5.

[29] H. Zhang, X. Zhang, and G. Gao, "Training supervised speech separation system to improve STOI and PESQ directly," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5374–5378, doi: 10.1109/ICASSP.2018.8461965.

[30] T. R. Titze, *Principles of Voice Production*. Upper Saddle River, NJ, USA: Prentice-Hall, 1994.

[31] R. J. Baken, *Clinical Measurement of Speech and Voice*. London, U.K.: Taylor & Francis, 1987.

[32] M. A. Kramer, "Nonlinear principal component analysis using auto associative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, 1991.

[33] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and Helmholtz free energy," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 3–10.

[34] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines (PDF)," in *Proc. ICML*, 2010, pp. 1–6.

[35] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. AISTATS*, Jun. 2011, pp. 315–323.

[36] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*. [Online]. Available: http://arxiv.org/abs/1606.05908

[37] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[38] S. Kullback, *Information Theory and Statistics*. Hoboken, NJ, USA: Wiley, 1959.

[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, MA, USA: MIT Press, 1986.

[41] A. B. Williams and F. J. Taylors, *Electronic Filter Design Handbook*. New York, NY, USA: McGraw-Hill, 1988.

[42] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*. 2nd ed. New York, NY, USA: Dover, 2001.

[43] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall, "The DARPA speech recognition research database: Specifications and status," in *Proc. DARPA Workshop Speech Recognit.*, 1986, pp. 93–99.

[44] H.-T. Luong and H.-Q. Vu, "A non-expert kaldi recipe for vietnamese speech recognition system," in *Proc. WLSI*, 2016, pp. 51–55.

HAO DUC DO received the B.Sc. and M.Sc. degrees in computer science from the VNUHCM-University of Science, Ho Chi Minh City, Vietnam, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree in computer science.

He is also an AI Researcher with the OLLI Technology JSC, where he leads a team of ten researchers to build a system of intelligent agents for smart home. His research interests include speech signal processing and deep learning.

SON THAI TRAN received the bachelor's degree in science from the VNUHCM-University of Science, 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.

He is currently a Senior Lecturer with the Faculty of Information Technology and the Head of the Office of Education and Training, VNUHCM-University of Science. His research interests include filtering, image processing, and pattern recognition.

DUC THANH CHAU received the Ph.D. degree in information science from JAIST, Japan, in 2014.

He is currently a Lecturer with the Faculty of Information Technology, VNUHCM-University of Science, Ho Chi Minh City. He is also the AI Technology Leader with the Cinnamon AI Laboratory. His research interest includes signal processing, particularly in spoken language (localization, enhancement, and recognition) and image processing (analysis, OCR, and information extraction).