A Comparative Study of Time and Frequency Domain Approaches to Deep Learning based Speech Enhancement

Soha A. Nossier
Dept. of Engineering and Computing
University of East London
London, UK
soha.abdallah.nossier@gmail.com

Julie Wall
Dept. of Engineering and Computing
University of East London
London, UK
j.wall@uel.ac.uk

Mansour Moniri
Dept. of Engineering and Computing
University of East London
London, UK
m.moniri@uel.ac.uk

Cornelius Glackin
Intelligent Voice Ltd
London, UK
neil.glackin@intelligentvoice.com

Nigel Cannings
Intelligent Voice Ltd
London, UK
nigel.cannings@intelligentvoice.com

***Abstract*—Deep learning has recently made a breakthrough in the speech enhancement process. Some architectures are based on a time domain representation, while others operate in the frequency domain; however, the study and comparison of different networks working in time and frequency is not reported in the literature. In this paper, this comparison between time and frequency domain learning for five Deep Neural Network (DNN) based speech enhancement architectures is presented. The comparison covers the evaluation of the output speech using four objective evaluation metrics: PESQ, STOI, LSD, and SSNR increase. Furthermore, the complexity of the five networks was investigated by comparing the number of parameters and processing time for each architecture. Finally some of the factors that affect learning in time and frequency were discussed. The primary results of this paper show that fully connected based architectures generate speech with low overall perception when learning in the time domain. On the other hand, convolutional based designs give acceptable performance in both frequency and time domains. However, time domain implementations show an inferior generalization ability. Frequency domain based learning was proved to be better than time domain when the complex spectrogram is used in the training process. Additionally, feature extraction is also proved to be very effective in DNN based supervised speech enhancement, whether it is performed at the beginning, or implicitly by bottleneck layer features. Finally, it was concluded that the choice of the working domain is mainly restricted by the type and design of the architecture used.**

***Index Terms*—Deep Learning, Speech Enhancement, Time Domain, Frequency Domain**

I. Introduction

Speech enhancement is one of the most challenging tasks in the signal processing field. It is the process of removing noise from speech, in order to increase quality and intelligibility. There are many applications for speech enhancement, for example, it is an essential process in hearing aids, mobile communication systems, Automatic Speech Recognition, head phones, and VoIP communication [1].

Researchers have been developing speech enhancement techniques for decades, which predict clean speech based on statistical assumptions about the relationship between speech and noise [2]. Recently, a new era of speech enhancement has emerged with the introduction of deep learning based techniques [3]. These techniques learn the mapping function that maps noisy speech to clean speech, without any statistical assumption, by training a DNN. This network is fed by a huge amount of data for pairs of clean and noisy speech, and then adjusts its parameters during the supervised learning process so as to generate the best prediction for the target clean speech [4], [5].

When training a speech enhancement neural network, the speech signal can be fed to the network in the time domain, or a transformation to the frequency domain can be applied first. Many research in the literature advocates the frequency domain approach [6]–[10] because speech signals are of a highly non-stationary nature and its components vary in both time and frequency. Consequently, a transformation to the frequency domain using a technique such as Short Time Fourier Transform (STFT) will result in a better representation of the speech signal [11], as information such as the harmonics and how the frequency amplitude varies in time can be known, and this leads to better network training. However, there are approaches that operate in the time domain [12]–[16], as it is believed that deep learning as a data driven approach has the ability to learn features during training. Accordingly, it is better to feed the DNN with the time domain signal and leave

This research is co-sponsored by Intelligent Voice Ltd.

it to learn on its own the most important features, because some useful information may be lost after transferring to the frequency domain.

In this paper, a comparison will be carried out between five different DNN based speech enhancement best performing architectures operating in time and frequency, so as to work out how the behaviour of each network changes with respect to the used domain, and then determine for each architecture, whether it is suitable to learn in time, or frequency, or both domains. The factors that affect implementations in time and frequency will be also investigated.

The rest of this paper is organized as follows. In Section II, a literature review is presented to show how our work will contribute to the literature. A brief review of different DNN based speech enhancement architectures and the design of the implemented networks will be presented in Section III. The experiments carried out and their setup are given in Section IV. The results and discussion are shown in Section V. Finally, Section VI provides the conclusion.

II. Literature Review

For DNN based speech enhancement in the frequency domain, the feature extraction process is based on generating the spectrogram of the signal, which carries information about the frequencies present in the signal and its relative amplitudes. The Discrete Fourier Transform (DFT) is one of the commonly used techniques, in which the signal is broken down into its sinusoidal harmonics, that when added up generate the original signal. In order to consider the changes of these frequency contents over time, the STFT is used [11]. In STFT the signal is first divided into frames or chunks, and then the Fourier transform is applied for each frame. These frames are generated by multiplying the signal by a function called a window function, and in order to reduce artifacts at the boundaries, these frames are usually overlapped.

There are many types of windowing functions; a rectangular window is the simplest function, which is a unity function. However, as it ends abruptly, this sharp edge will lead to the appearance of frequencies that are not in the original signal, which causes the spectrum to be smeared; a problem known as spectral leakage [17]. Instead, there are another two popular windowing functions, called Hanning and Hamming windows, with a better frequency response that can be used to overcome this problem, as they ensure that the ends of the signal are close to zero [18].

On the other hand, the representation of the speech in the time domain carries information about the voltage change over time as the pressure of the sound waves is converted into electrical signals by microphones. As a result, the relationship between neighbouring features in time defines the frequency information of the signal [19]. A framing to the time domain signal, using any window function described earlier, is usually performed on the speech signal before being fed to the network, so as to fix the length of the input features.

Representation of the speech signal in either time or frequency domain has its advantages and disadvantages. Although working in the frequency domain gives the network directly important information about the input signal which leads to lower network parameters, it adds more computation as the audio signal is originally represented in time. The STFT operation is done at the beginning, and the inverse operation is done at the end, so it increases the computational cost and the time taken to process and output the audio [13]. Furthermore, the transformation process affects the amplitude of the signal so compensation to this loss should be done by scaling the amplitude of output signal, which may affect the quality of the output speech. Moreover, when operating in the frequency domain it is most common to deal only with the amplitude of the speech signal, assuming that the phase is insensitive to noise [20] so the phase of the noisy speech is extracted to be added to the estimated signal when reconstructing the audio. However, this assumption does not always hold, as some studies show the importance of phase in improving the performance [21], [22]. Many techniques have been proposed to retrieve the clean phase, or are based on the use of the complex spectrograms in order to solve this issue [23]–[25], but finally all these result in more computations done to the signal.

On the other hand, working in time results in fewer computations, as the framing of the input signal is the only required operation, and some researchers even work with the waveform without the framing process. Moreover, the phase information is estimated during the training that leads to better prediction of the clean signal, and also no scaling is needed for the output signal. However, working in the time domain results in a much higher number of network parameters due to the large frame size used, which is proved to be better than smaller frames [12], [26]. This larger number of parameters increases the size of the model, and restricts its applicability in some real time implementations as the model may not fit into the hardware [27]. Additionally, although deep learning is a data driven approach, feature based learning has been proved to positively affect the learning process [28]. Based on this fact, the network performance is expected to be affected when the input does not carry much information, and more effort may be needed to design a more complex network suitable to extract this information in the training process.

According to our knowledge, the comparison between different DNN based speech enhancement architectures working in the time and frequency domains has not been addressed yet. How the different networks' performance is affected by the operating domain, and what is the best performing architecture in each domain are questions still needed to be answered. This comprehensive comparison is the subject of this paper.

III. Speech Enhancement Architectures

In this section, details of the five implemented architectures are given. The design of these architectures is based on implementations found in the literature; however, some modifications in the setup were applied so as to perform a fair comparison between their operation in time and frequency. Moreover, the 1D convolutions only are used in both time and

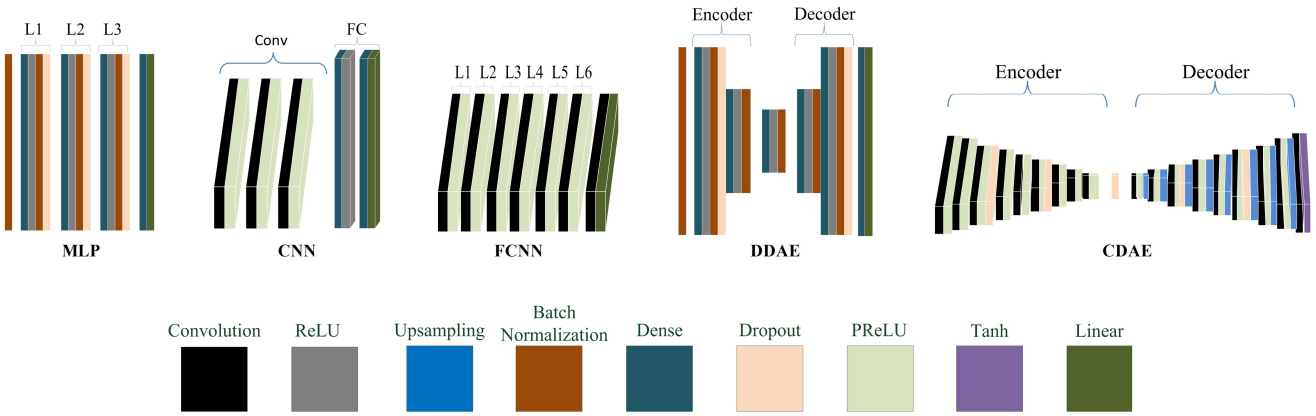

Fig. 1: The Five Implemented Deep Neural Network Architectures

frequency implementations, because they have been proven to be more efficient in audio processing, and results in a low computational cost which makes the network suitable for real time applications [29], [30].

A Multi Layer Perceptron (MLP) is the basic and simplest speech deep learning enhancement architecture. In this architecture, all the nodes of the hidden layers are fully connected to each other; however, this architecture has a huge computational cost. Many speech enhancement networks are based on the MLP architecture [6], [7], [31], and almost all of them operate in the frequency domain, as the fully connected layer cannot represent high and low frequencies at the same time resulting in the failure of the layer to reconstruct the speech signal in time [13]. However, a study managed to output the speech signal in the time domain by training an MLP in frequency and performing the conversion process back to the time domain during the training procedure [32].

The MLP architecture used, based on [6], consists of 3 hidden fully connected layers of 2,048 units and Rectified Linear Unit (ReLU) activations, followed by batch normalization to improve the performance and training stability, and a dropout layer of 20% rate to avoid overfitting to the training data. The final output layer is a linear activation function to predict the target.

The Convolutional Neural Network (CNN) is another architecture used to solve the high computational cost and complexity problems of the MLP by using the convolution operation in both forward and backward propagation steps, so as to reduce the network parameters based on two ideas: parameter sharing and sparsity of connections. Parameter sharing means that a feature takes advantage of other features in a certain part of the input, and uses it in another part, while sparsity of connections means that the output value in each layer does not depend on all the inputs of the previous layer [4]. The final prediction layer of a CNN is usually a fully connected layer. The CNN is also proven to be better at dealing with the speech enhancement problem than an MLP [10], [33], [34].

The CNN architecture used, based on [10], [33], consists of 3 convolutional layers and two fully connected layers. We used a 1D convolution, instead of 2D, as it is more suitable for audio representation, and each convolution layer is followed by PReLU activation functions, instead of ReLU. The number of convolution layer filters are set to 64, and we used kernels of size 20. The first fully connected layer consists of 512 units with ReLU activation, and the final prediction fully connected layer is of the linear activation type.

In an attempt to further decrease the computations of CNNs, another version of CNN was proposed, known as the Fully Convolutional Neural Network (FCNN), in which the fully connected layers are replaced with convolutional layers so as to have a network with all convolutional layers. Some FCNN based speech enhancement networks are found to operate in the frequency domain [9], [35], while others are time domain based implementations [13], [14]. The FCNN architecture used here is based on [13]; however, 6 1D convolutional layers with Parametric Rectifiel Linear Unit (PReLU) activations are used, instead of 2D convolution with ReLU activation, and a final convolution layer with linear activation for predicting the output. The filter size used is 64, and the kernel size is 20, and they are constant across all layers.

The Autoencoder is another architecture that aims to output a similar representation as the input using two separate networks: the encoder and the decoder. The encoder compresses the input by removing any unimportant information so as to finally give a compact form of the input data, and then the decoder reconstructs the input [4]. Denoising autoencoders are used in speech enhancement based on the idea that the noise is considered as unimportant information when trying to represent clean speech, so it is reduced significantly during the compression process [36]. There are two types of denoising autoencoders: the deep denoising autoencoder (DDAE) and the convolutional denoising autoencoder (CDAE). Some speech enhancement architectures are based on DDAE [8], [37], and most known architectures are operating in the frequency domain, while recently CDAE is used in both time and frequency [12], [38], and it is proven to be a very promising speech enhancement architecture.

The DDAE architecture used, based on [8], consists of

encoder and decoder networks; each has 2 fully connected layers of 2,048, and 500 hidden units. Each of these fully connected layers is followed by a ReLU activation, and batch normalization layer. Another 2 dropout layers of 10% rate were added after the first layer of the encoder and the last layer of the decoder. The bottleneck middle layer has 180 units. The final output layer is of linear activation.

The CDAE architecture used, based on [12], [15], is another FCNN; however, it is based on autoencoder. It has 9 convolution layers in each of the encoder and decoder networks, followed by PReLU activation. Strided convolution was used with stride size of 2 in the encoder layers, while up sampling of size 2 was used in the decoder. Each three successive layers are of the same filter and kernel size. The filter size increases across hidden layers; 64, 128, and 256 filter sizes were used, while the kernel size decreases; 7, 5, and 3 kernel sizes were used. Dropout of rate 20% was used after every three layers. The activation function of the final convolution layer is Tanh. Skip connections are applied between the encoder and the decoder, retaining important information as processing proceeds deeper into the network.

IV. Experimental Setup

Speech and noise datasets were collected for the training and testing procedures. The clean speech corpus used in the experiments is the Voice Bank corpus [39]. A random selection of 4,730 audio files was carried out to make $\sim$ 5 hours of clean speech for training purposes. Another 450 clean speech audio files of about 30 minutes duration were randomly selected for testing purposes.

A diversity of noise environments were randomly mixed to the clean speech files. A total of 105 noise environments were used in the training process, 90 from the 100 Noise Environment dataset [40] and 15 from NOISEX-92 corpus [41]. For testing purposes, 9 seen different crowd noise and 1 Additive White Gaussian Noise (AWGN) are used, and another 10 unseen noise environments taken from the rest of 100 Noise Environment dataset that were not used in the training process. The selected noise environments for testing purposes are a mix of human generated noise, such as the crying sound, yawning sound, and human crowd sounds, and other non-human generated noise, such as AWGN, phone dialing, shower noise, tooth brushing, and wood creaks.

All the audio files are down-sampled to 8kHz as this range has the most important speech features, and the noise audios were truncated or repeated to be of the same length as the clean speech audio. The training was done at 0 dB SNR, so amplitude scaling was carried out for clean speech and noise audios to be of the same intensity level, while testing was done using 6 different SNRs, -5 to 20 with a step of 5, and also the average and standard deviation of these values were calculated. Due to the fact that these experiments were done in both time and frequency domain, two different preprocessing techniques were performed to the audio files before being fed to the DNN, which are framing and STFT, respectively. For time domain training, the audio files were divided into frames of size 2,048 and 50% overlap using a Hamming window, while for the frequency domain training, a frame size of 256 with 50% overlap was used, and an extra step was then performed by applying the FFT to these frames using an FFT of size 256 to get a good resolution in both time and frequency.

The training target is the speech time frame in the case of the time domain training, and the spectral magnitude for the frequency domain training. As a result, the noisy phase was used in reconstructing the audio files from the frequency domain, on the assumption that the phase is not highly affected by noise. Minimum Mean Square Error (MMSE) is the loss function used during the training process with the Adam optimizer; learning rate= 0.001, β_1=0.1, β_2=0.999. A 10% validation set was used in the training process. A batch size of 128 was used, and the training is based on 50 epochs with 10% of the training data used in validation. These implementations were done using the Keras library with Tensorflow backend.

V. Results and Discussion

A. Objective Evaluation

The output speech was evaluated using the well known objective evaluation metrics: Perceptual Evaluation of Speech Quality (PESQ) [42], Short Time Objective Intelligibility (STOI) [43], Log Spectral Distortion (LSD), and Segmental Signal to Noise Ratio (SSNR) increase. The results of the different architectures, according to the four metrics, are given in Table I to IV, and summarized in Table V.

The results show that the MLP and DDAE based architectures give very bad performance in the time domain, and the network seems to be unable to learn the mapping function. The CNN works better in the frequency domain as well; however, it also gives acceptable performance in the time domain. The FCNN gives better performance in the frequency domain with respect to all metrics, except for speech intelligibility which is much better in the time domain. This means that some metrics may be better in certain domains, hence choosing between time and frequency is also based on which metric is of the highest importance in the application where speech enhancement is applied. CDAE is the only architecture that works better in the time domain with respect to all the metrics, except speech distortion. The reason for that is the different nature of this architecture, as it is trying to give a similar representation of the input, regardless of its domain, after removing the noise in the bottleneck layer, and it seems to be able to better represent the clean speech in the time domain rather than the frequency domain. It is also clear that working in the frequency domain, for all architectures, results in lower speech distortion.

The results also prove that although deep learning is a data driven approach, feature extraction is a very important stage and results in significant improvement in the performance, and that is why the frequency domain representation outperforms the time domain approach in most architectures. Regarding the CDAE architecture, in which the time domain implementation is better, the reason for that is the bottleneck representation, which is a feature extraction step on its own. Although the

TABLE I: Speech Quality Evaluation (PESQ score)

SNR	Noisy	MLP		CNN		FCNN		DDAE		CDAE	
		Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time
20	2.92	2.41	2.12	3.09	2.53	3.01	2.81	2.82	1.84	2.93	3.12
15	2.62	2.34	2.12	2.90	2.44	2.84	2.67	2.72	1.82	2.81	2.97
10	2.32	2.25	2.11	2.68	2.30	2.63	2.51	2.58	1.75	2.68	2.82
5	2.04	2.16	2.08	2.46	2.13	2.44	2.34	2.41	1.63	2.52	2.67
0	1.81	2.02	1.72	2.21	1.89	2.22	2.13	2.19	1.47	2.32	2.49
-5	1.60	1.70	1.55	1.87	1.59	1.88	1.78	1.83	1.32	2.01	2.24
AVG	2.219	**2.147**	1.949	**2.537**	2.146	**2.503**	2.374	**2.424**	1.639	2.543	**2.716**
SD	0.498	0.258	0.250	0.449	0.355	0.416	0.378	0.368	0.207	0.339	0.322

TABLE II: Speech Intelligibility Evaluation (STOI Score)

SNR	Noisy	MLP		CNN		FCNN		DDAE		CDAE	
		Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time
20	0.91	0.82	0.52	0.88	0.86	0.88	0.94	0.85	0.67	0.89	0.93
15	0.88	0.81	0.52	0.86	0.84	0.86	0.92	0.83	0.67	0.87	0.92
10	0.83	0.79	0.52	0.83	0.79	0.84	0.90	0.81	0.67	0.85	0.90
5	0.78	0.77	0.52	0.79	0.75	0.80	0.86	0.79	0.62	0.82	0.87
0	0.71	0.73	0.52	0.74	0.71	0.76	0.81	0.75	0.52	0.78	0.84
-5	0.64	0.65	0.48	0.67	0.64	0.69	0.73	0.68	0.47	0.72	0.77
AVG	0.790	**0.760**	0.512	**0.795**	0.765	0.805	**0.861**	**0.785**	0.604	0.820	**0.872**
SD	0.101	0.063	0.017	0.078	0.084	0.072	0.079	0.062	0.088	0.064	0.059

TABLE III: Log Spectral Distortion Results (LSD)

SNR	Noisy	MLP		CNN		FCNN		DDAE		CDAE	
		Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time
20	1.36	1.05	1.74	1.09	2.01	1.13	1.49	1.23	2.39	1.37	1.87
15	1.62	1.12	1.74	1.18	2.03	1.23	1.53	1.28	2.43	1.41	1.89
10	1.92	1.18	1.76	1.30	2.06	1.35	1.58	1.32	2.51	1.44	1.91
5	2.21	1.22	1.80	1.44	2.12	1.47	1.63	1.40	2.64	1.51	1.94
0	2.46	1.32	1.88	1.64	2.25	1.63	1.73	1.54	2.83	1.62	1.96
-5	2.62	1.68	2.01	1.98	2.45	1.93	1.94	1.85	2.99	1.82	1.99
AVG	2.032	**1.261**	1.823	**1.438**	2.155	**1.456**	1.650	**1.437**	2.631	**1.529**	1.926
SD	0.489	0.225	0.105	0.330	0.170	0.292	0.166	0.230	0.237	0.168	0.046

TABLE IV: Segmental Signal to Noise Ratio Increase

SNR	MLP		CNN		FCNN		DDAE		CDAE	
	Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time
20	6.44	0.59	6.98	3.67	7.13	6.41	6.73	1.98	7.07	6.82
15	7.12	0.79	7.60	3.04	7.72	6.93	7.44	2.10	7.77	8.53
10	7.56	0.73	7.92	2.53	7.98	7.97	7.85	2.36	8.23	8.91
5	7.77	1.62	7.94	2.32	7.97	7.62	7.86	3.33	8.33	8.82
0	7.65	1.41	7.46	2.50	7.63	6.81	7.63	3.75	7.98	8.25
-5	7.03	1.52	6.43	2.51	6.83	6.24	7.02	3.07	7.51	7.94
AVG	**7.262**	1.110	**7.388**	2.762	**7.542**	6.994	**7.422**	2.764	7.814	**8.212**
SD	0.502	0.457	0.586	0.507	0.468	0.677	0.459	0.721	0.473	0.771

TABLE V: Average of The Results

Metric	MLP		CNN		FCNN		DDAE		CDAE	
	Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time
PESQ	**2.147**	1.949	**2.537**	2.146	**2.503**	2.374	**2.424**	1.639	2.543	**2.716**
STOI	**0.760**	0.512	**0.795**	0.765	0.805	**0.861**	**0.785**	0.604	0.820	**0.872**
LSD	**1.261**	1.823	**1.438**	2.155	**1.456**	1.650	**1.437**	2.631	**1.529**	1.926
ΔSSNR	**7.262**	1.110	**7.388**	2.762	**7.542**	6.994	**7.422**	2.764	7.814	**8.212**

TABLE VI: Comparing Different Networks' Parameters

Metric	MLP		CNN		FCNN		DDAE		CDAE	
	Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time	Freq.	Time
P(10^6)	**8**	16	**0.2**	0.4	**0.4**	0.6	**2**	3	3	3
T(s)	21.5	**11.1**	14.1	**12.8**	24	**23.4**	15.5	**14.6**	34.5	**24**
Layers	15		10		12		21		49	

CDAE network is fed by the time domain speech signal, nonlinear transformation to another compact form of the input is done in the bottleneck layer, which means feature extraction is also done, but implicitly inside the network. As a result, this gives the network the ability to represent the clean speech in the time domain.

Regarding the overall performance of the five networks, convolutional based implementations (CNN, FCNN, and CDAE) outperform the basic fully connected layers architectures (MLP and DDAE). The CDAE architecture is a very promising architecture for speech enhancement in both time and frequency domains.

B. Networks' Complexity Comparison

Table VI shows the comparison between the used parameters in each implementation and the processing time. These results are based on running the algorithm on an NVIDIA Quadro M3000M GPU with clock 1050 MHz and 160 GB/s memory bandwidth. It is clear that the number of parameters in all the time domain implementations is much higher which leads to increased model size, as discussed in Section 2. Except for CDAE, as zero padding is performed to the input frequency feature so as to keep the input size of 2,048, so the network is able to decrease the input through the 8 layers of the encoder. Convolutional based architectures also have a lower number of parameters than fully connected architectures. This is because of the sparse connections of CNNs, and more specifically due to the use of 1D convolution in both time and frequency implementations, which leads to a decreased number of parameters. The processing time is calculated based on processing 224 speech audio files of about 15 minutes duration. The operation was done 6 times, then the average time was taken so as to consider any error caused by processing freezing. All frequency domain implementations take a longer time to process because of the transformation operation. The number of layers is also shown in the table. The CDAE architecture is the deepest architecture, 49 layers, so this is another possibility why this architecture outperforms in the time domain. Very deep neural networks are proved to be better at extracting more advanced features through the layers [44], especially in the case of convolutional based architectures [45]. It is also clear that the depth of the architecture increases the processing time.

C. Factors Affecting Time domain Learning

More experiments were done to show the effect of three factors on the performance of the fully connected architectures in the time domain, in an attempt to enhance the performance. The outcome of these experiments is represented in Table VII. These results are based on testing the network on seen and unseen data at the same 6 SNRs used before, then the average was calculated.

1) Frame Size: The effect of using smaller frame size was investigated by training the MLP and DDAE architectures using frames of size 256 instead of 2,048. Using a small frame size leads to better performance for the MLP. However, speech

intelligibility was negatively affected for the DDAE network due to the compression process, which may result in inaccurate speech reconstruction with small input frame size, especially for an architecture without skip connections which helps in retaining the information as the processing proceeds deeper from the encoder to the decoder network.

2) Architecture Depth: In order to show the effect of the depth of the network on the performance in the time domain, two more layers were added for the MLP architecture for the network to have 5 layers instead of 3. Two more layers were also added to each of the encoder and decoder networks for the DDAE architecture in order to have 4 layers in each of them. The number of hidden units were decreased through the encoder layers, 2,049, 1,024, 500, 250, and 150 units were used; and increased in reverse order through the decoder layers. The results show that increasing the depth of the architecture has a positive impact on the overall network performance in the time domain, especially for DDAE. In should be also mentioned here that adding skip connections to the DDAE may lead to further enhancement [46], because of their ability to prevent information loss in deep architectures.

3) Dataset Size: In order to show if increasing the dataset size could enhance the network learning in the time domain or not, the dataset size was doubled by training the MLP and DDAE architectures using 10 hours of speech instead of 5 hours. For the MLP network, the output speech intelligibility score increased. However, the speech quality was negatively affected, and this may be because the network starts to overfit to the training data, so the ability to remove noise from unseen data decreased, leading to worse speech quality. Regarding the DDAE network, increasing the dataset size results in a better performance as this gives the network a better chance to learn speech features, and decreasing the number of hidden units through the encoder network prevents this architecture from overfitting. However, when continuing to increase the dataset size without changing the network design leads to the same overfitting problem experienced by the MLP.

Fig. 2 shows the PESQ and STOI scores when considering the above three factors. The original output scores based on the first experiment is also shown in the same figure for comparison. Although some of these factors result in an improvement in the performance, the output speech is still of relatively low overall quality. Consequently, there is a need for a remarkable change in the design, or the addition of techniques that will help in audio reconstruction, for these architectures to be able to learn in the time domain.

D. Factors Affecting Frequency domain Learning

1) Training Targets: The previously conducted experiments were based on spectrogram mapping as a training target for frequency domain based implementations. In this part, the use of a masking target was investigated to show how this will affect the network performance. The Ideal Ratio Mask (IRM) [47] was used in this evaluation. It is clear from the results in Table VIII that the use of IRM results in an improved performance for all architectures. As a result, using masking targets will result in further performance enhancement in the case of frequency domain learning.

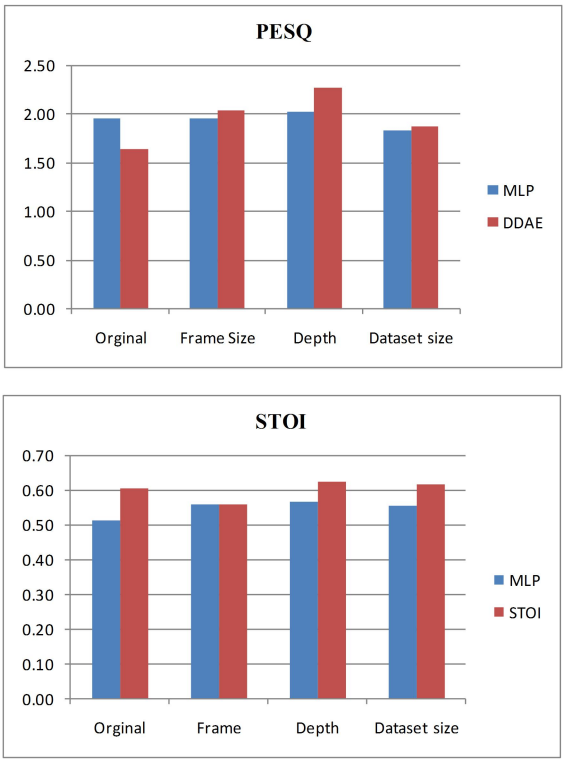

Fig. 2: The Factors Affecting Time Domain based Learning

TABLE VII: Factors Affecting Time Domain Learning

Metric	MLP			DDAE		
	Frame Size	Depth	Dataset size	Frame Size	Depth	Dataset size
PESQ	1.956	2.026	1.831	2.040	2.262	1.871
STOI	0.557	0.565	0.556	0.560	0.622	0.614

TABLE VIII: Factors Affecting Frequency Domain Learning

Metric	MLP		CNN		FCNN		DDAE		CDAE	
	IRM	cSpec	IRM	cSpec	IRM	cSpec	IRM	cSpec	IRM	cSpec
PESQ	2.388	2.149	2.564	2.536	2.637	2.510	2.430	2.105	2.667	2.393
STOI	0.801	0.680	0.808	0.792	0.819	0.794	0.814	0.675	0.834	0.776

TABLE IX: Generalization Ability Evaluation

Metric	FCNN		CDAE	
	Freq.	Time	Freq.	Time
PESQ	2.334	1.957	2.352	2.067
STOI	0.847	0.590	0.859	0.708

2) Phase Consideration: A drawback when operating in the frequency domain is to use the noisy phase when reconstructing the clean speech audio. Recently, complex spectrograms are used in order to solve this issue, where the network is learning both the magnitude and phase during the learning process. In order to show how the performance will be affected when the learning is based on complex spectrogram, the architectures were re-implemented using complex spectrogram as the training target. The real and imaginary parts of the

spectrogram were stacked together and fed to the DNNs. The obtained results are shown with the results of the IRM in Table VIII. Using complex spectrogram leads to much better speech intelligibility and quality than MLP, CNN, and DDAE implementations in the time domain. However, the overall performance in the frequency domain is negatively affected as it is more challenging for the network to enhance both the magnitude and the angle in the training process. The performance of FCNN and CDAE is shown to be better in the case of time domain than when using complex spectrogram. As a conclusion, the use of complex spectrograms can act as a compromise solution for architectures that fail to operate in the time domain, and when the use of the noisy phase in clean speech reconstruction causes a significant negative effect on the performance of frequency domain systems. It should be also mentioned that the use of Complex Ideal Ratio Mask [48] was reported to outperform IRM with respect to speech intelligibility, so cIRM can be also used to solve the noisy phase issue. Furthermore, precise choice of the input features or combining many features may also result in further improvement in the performance. Although this is not addressed in this work, but it is proved to have a positive impact [49], [50].

E. Generalization Ability

In order to show the effect of the choice of the working domain on the network generalization ability, the two networks (FCNN and CDAE) that generated good performance in both the time and frequency domain were re-evaluated using a different English speech dataset from the one used in the training process. The LibriSpeech corpus [51] was used in this evaluation. In order to make a fair comparison between these results and the results obtained in Subsection A. As before, 30 minutes duration of speech were randomly selected from the LibriSpeech corpus, and the same noise environments at the same SNR levels were used in the testing process. These results are presented in Table IX.

The results show a significant degradation in the performance for the time domain based implementation, especially in the speech intelligibility score; as the FCNN architecture outputs unintelligible speech. On the other hand, the frequency domain implementations show good generalization ability. Despite using validation set in the training regime, and dropout layers in the CDAE architecture, the time domain networks still do not generalise well to Librispeech dataset. It can be concluded from these results that time domain based learning fails to generalize, and that although a regularization dropout technique is added, such as in the case of the CDAE architecture, generalization is still an issue that must be considered for time domain based implementations. On the other hand, these results give an advantage to frequency domain based learning, as even if regularization techniques are not applied to the FCNN network, it managed to maintain a good performance for different, unseen data.

VI. Conclusion

In this paper, an investigation has been carried out on the frequency and time domain approaches for deep learning based speech enhancement, by comparing five different architectures. The results show that fully connected based architectures, MLP and DDAE, experience a significant degradation in the performance when the learning process is performed in the time domain. On the other hand, convolution based architectures, CNN, FCNN, and CDAE, give an acceptable performance in both frequency and time domains. Additionally, the CDAE architecture outperforms the other networks, regardless of the working domain.

Working in the time domain results in more intelligible speech for FCNN and CDAE designs, while working in the frequency domain gives better speech quality for all the architectures, except the CDAE, which outperforms in the time domain. However, considering the network generalization ability, the time domain implementations failed to generalize even when a regularization technique is applied. Conversely, the frequency domain implementations show a good generalization ability even for implemented architectures with no regularization technique.

Although changing the depth, frame size, and dataset size was shown to improve the overall performance of fully connected architectures learning in the time domain, a careful design and extra techniques are needed for this type of DNN when operating in the time domain, in order to generate speech with acceptable quality and intelligibility.

Most of the architectures perform better in the frequency domain. The implementations in the frequency domain also show great improvement when masking training targets are used. Furthermore, the use of a complex spectrogram is a good approach for enhancing both the magnitude and the phase, when the use of noisy speech in the reconstruction process negatively affects the performance. It can be concluded here also that although deep learning is a data driven approach, feature extraction has a great positive impact on the network performance.

Finally, it can be noticed that the choice of the working domain depends on the design of the architecture and the complex relationship between speech quality, intelligibility, and distortion, because for certain designs not all the evaluation metrics are better in a specific domain. However, when choosing to train DNNs in the time domain for the speech enhancement task, the generalization issue should be taken into consideration.

References

[1] P. C. Loizou, *Speech enhancement: theory and practice*. CRC, 2013.

[2] Y. Hu and P. C. Loizou, "Subjective comparison and evaluation of speech enhancement algorithms," *Speech Comm.*, vol. 49, no. 7-8, pp. 588–601, 2007.

[3] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 26, no. 10, pp. 1702–1726, 2018.

[4] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[6] Y. Xu, J. Du, L. Dai, and C. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 23, no. 1, pp. 7–19, 2015.

[7] D. Wang, "Deep learning reinvents the hearing aid," *IEEE Spectrum*, vol. 54, no. 3, pp. 32–37, 2017.

[8] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder." in *INTERSPEECH*, 2013, pp. 436–440.

[9] S. Park and J. Lee, "A fully convolutional neural network for speech enhancement," *arXiv:1609.07132*, 2016.

[10] S. Chakrabarty, D. Wang, and E. Habets, "Time-frequency masking based online speech enhancement with multi-channel data using convolutional neural networks," in *IWAENC*. IEEE, 2018, pp. 476–480.

[11] M. Portnoff, "Time-frequency representation of digital signals and systems based on short-time fourier analysis," *IEEE Trans. Acoust. Speech Sig. Proc.*, vol. 28, no. 1, pp. 55–69, 1980.

[12] A. Pandey and D. Wang, "A new framework for cnn-based speech enhancement in the time domain," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 27, no. 7, pp. 1179–1188, 2019.

[13] S. Fu, Y. Tsao, X. Lu, and H. Kawai, "Raw waveform-based speech enhancement by fully convolutional networks," in *APSIPA ASC*. IEEE, 2017, pp. 6–12.

[14] S. Fu, T. Wang, Y. Tsao, X. Lu, and H. Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 26, no. 9, pp. 1570–1584, 2018.

[15] S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech enhancement generative adversarial network," *arXiv:1703.09452*, 2017.

[16] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *ICASSP*. IEEE, 2017, pp. 421–425.

[17] F. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[18] P. Podder, T. Khan, M. Khan, and M. Rahman, "Comparative performance analysis of hamming, hanning and blackman window," *Int. J. Comput. Appl.*, vol. 96, no. 18, 2014.

[19] J. Gamboa, "Deep learning for time-series analysis," *arXiv:1701.01887*, 2017.

[20] D. Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE Trans. Acoust. Speech Sig. Proc.*, vol. 30, no. 4, pp. 679–681, 1982.

[21] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Comm.*, vol. 53, no. 4, pp. 465–494, 2011.

[22] G. Shi, M. Shanechi, and P. Aarabi, "On the importance of phase in human speech recognition," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 14, no. 5, pp. 1867–1874, 2006.

[23] Z. Ouyang, H. Yu, W. Zhu, and B. Champagne, "A fully convolutional neural network for complex spectrogram processing in speech enhancement," in *ICASSP*. IEEE, 2019, pp. 5756–5760.

[24] S. Fu, T. Hu, Y. Tsao, and X. Lu, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," in *MLSP*. IEEE, 2017, pp. 1–6.

[25] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux, "Phase processing for single-channel speech enhancement: History and recent advances," *IEEE Sig. Proc. Mag.*, vol. 32, no. 2, pp. 55–66, 2015.

[26] D. Eringis and G. Tamulevičius, "Improving speech recognition rate through analysis parameters," *J. Elect. Control and Comm. Eng.*, vol. 5, no. 1, pp. 61–66, 2014.

[27] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang, and T. Sainath, "Deep learning for audio signal processing," *J. Selected Topics Sig. Proc.*, vol. 13, no. 2, pp. 206–219, 2019.

[28] L. Hertel, H. Phan, and A. Mertins, "Comparing time and frequency domain for audio event recognition using deep learning," in *IJCNN*. IEEE, 2016, pp. 3407–3411.

[29] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, and M. Gabbouj, "1-d convolutional neural networks for signal processing applications," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8360–8364.

[30] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *arXiv preprint arXiv:1905.03554*, 2019.

[31] Q. Liu, W. Wang, P. Jackson, and Y. Tang, "A perceptually-weighted deep neural network for monaural speech enhancement in various background noise conditions," in *EUSIPCO*. IEEE, 2017, pp. 1270–1274.

[32] Y. Wang and D. Wang, "A deep neural network for time-domain signal reconstruction," in *ICASSP*. IEEE, 2015, pp. 4390–4394.

[33] T. Kounovsky and J. Malek, "Single channel speech enhancement using convolutional neural network," in *ECMSM*. IEEE, 2017, pp. 1–5.

[34] S. Fu, Y. Tsao, and X. Lu, "SNR-aware convolutional neural network modeling for speech enhancement." in *INTERSPEECH*, 2016, pp. 3768–3772.

[35] K. Tan and D. Wang, "A convolutional recurrent neural network for real-time speech enhancement." in *INTERSPEECH*, 2018, pp. 3229–3233.

[36] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[37] P. Shivakumar and P. Georgiou, "Perception optimized deep denoising autoencoders for speech enhancement," in *INTERSPEECH*, 2016, pp. 3743–3747.

[38] E. Grais and M. Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," in *Global SIP*. IEEE, 2017, pp. 1265–1269.

[39] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," in *O-COCOSDA/CASLRE*. IEEE, 2013, pp. 1–4.

[40] G. Hu, "100 nonspeech environmental sounds." [Online]. Available: http://www.cse.ohio-state.edu/pnl/corpus/HuCor- pus.html, 2014.

[41] A. Varga and H. Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Comm.*, vol. 12, no. 3, pp. 247–251, 1993.

[42] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *ITU-T Recommendation, p. 862.*, 2001.

[43] C. Taal, R. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 19, no. 7, pp. 2125–2136, 2011.

[44] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks studies on speech recognition tasks," *arXiv preprint arXiv:1301.3605*, 2013.

[45] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *ICASSP*. IEEE, 2017, pp. 4845–4849.

[46] M. Tu and X. Zhang, "Speech enhancement based on deep neural networks with skip connections," in *ICASSP*. IEEE, 2017, pp. 5565–5569.

[47] S. Srinivasan, N. Roman, and D. Wang, "Binary and ratio time-frequency masks for robust speech recognition," *Speech Comm.*, vol. 48, no. 11, pp. 1486–1501, 2006.

[48] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 24, no. 3, pp. 483–492, 2015.

[49] S. Pirhosseinloo and J. S. Brumberg, "A new feature set for masking-based monaural speech separation," in *52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 828–832.

[50] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE tran. on audio, speech, and lang. proc.*, vol. 22, no. 12, pp. 1849–1858, 2014.

[51] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.