CHAPTER 4

DNN BASED SPEECH ENHANCEMENT

4.1 INTRODUCTION TO DEEP LEARNING ALGORITHMS

In speech processing, speech enhancement is essential. Due to the highly gratifying results of the deep learning algorithms, it has been applied to various speech-processing tasks. This research focuses on speech enhancement by applying deep learning algorithms that enhance speech quality and intelligibility (Rascon, 2023).

4.2 DEEP LEARNING ALGORITHMS FOR SPEECH ENHANCEMENT

The ability of the deep learning system to learn, predict changes in data, and improvise based on the training process makes it highly proficient at capturing complex patterns in audio data. This proficiency enables the system to distinguish and suppress noise effectively, significantly improving speech quality and intelligibility.

Speakers speak differently, meaning the same phoneme or word sounds differ from speaker to speaker. Multilayer networks can support the parallel, interactive, and hierarchical processing of speech as a result of the dynamic characteristics of speech. Generally, speech can be processed in multilayer networks in two ways. Interaction between speech segments can be achieved by laying them out spatially on a layer of input units connected to a higher level of output units. It is also possible to present a single slice of the speech signal to the input units and then present subsequent slices. The system must have some memory to combine data from different time slices.

The noisy speech signal (noisy audio) is generated by adding different noises to the clean speech signal. The general flow diagram for the speech enhancement process using deep learning is given in Figure 4.1. The clean speech signal (clean audio) of 16kHz sampling frequency is converted to frequency domain using Short Time Fourier Transform (STFT). The frequency domain transformation of the signal is done using a hamming window of 75% overlap and a window length of 480. When the speech signal is

transformed into the frequency domain using the STFT, it is represented by its complex spectrogram, which includes both magnitude and phase, as shown in equation 4.1.

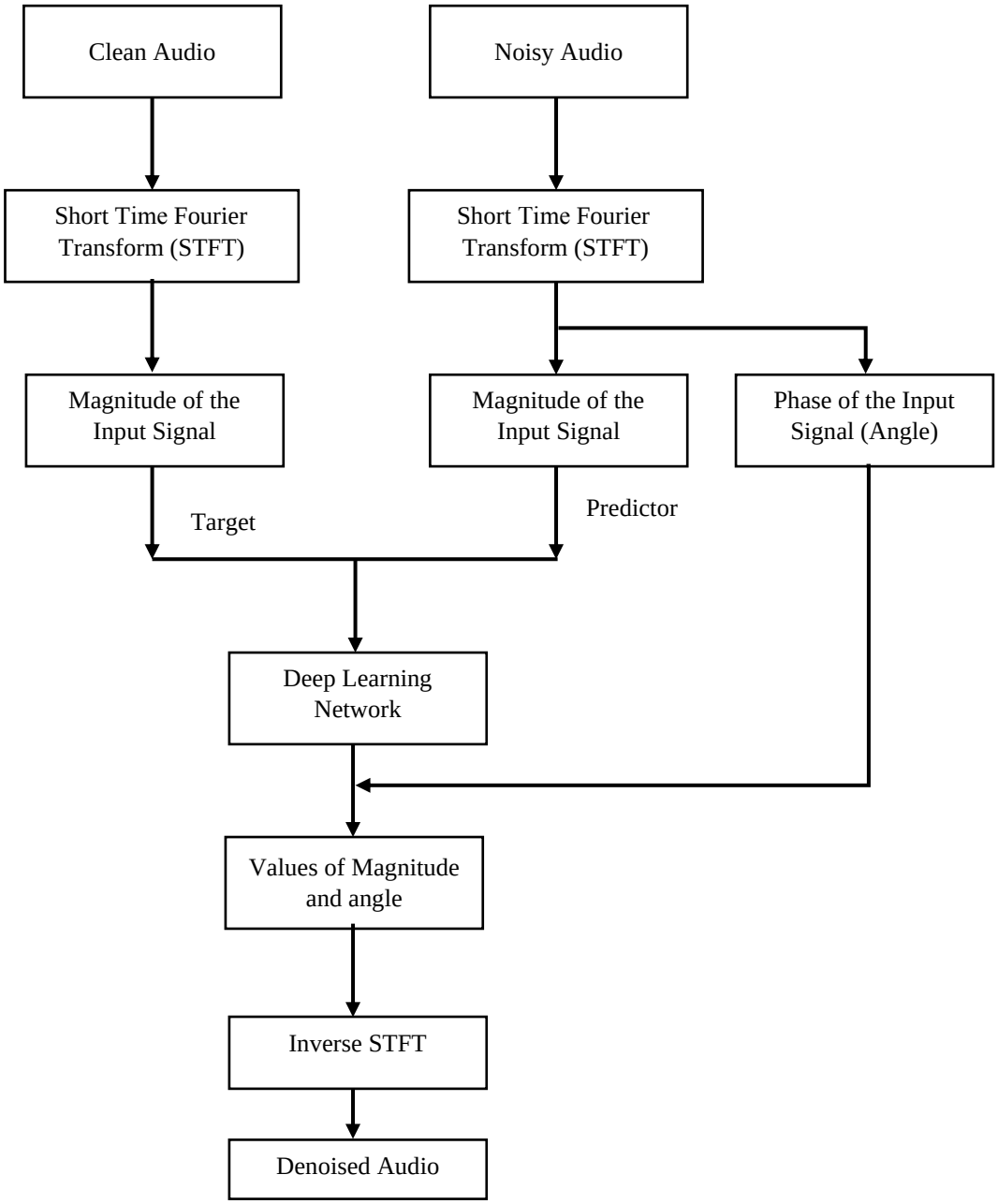


Figure 4.1 General Flow Diagram for Speech Enhancement Process using Deep Learning

$$STFT(t,f) = Magnitude(t,f) \times e^{j \times Phase(t,f)} \quad (4.1)$$

The input magnitude spectrum of the noisy speech is referred to as the predictor, and the magnitude spectrum of the clean audio is referred to as the target. The deep neural network is trained to minimize the mean square error between the predicted and target

magnitude spectrum. The output of the network is the magnitude spectrum of the denoised signal. The phase spectrum is critical for reconstructing the time-domain signal accurately. Without the correct phase information, the inverse STFT (ISTFT) could not reconstruct a signal that sounds like the original, even if the magnitude spectrum is correct. The output magnitude spectrum from the network is combined with the phase of the noisy signal to convert the denoised speech into the time domain, as given in equation 4.2.

$$Denoised\ STFT(t,f) = Denoised\ Magnitude(t,f) \times e^{\,j \times Phase_{noisy}(t,f)} \quad (4.2)$$

The inverse STFT is performed to convert the denoised speech signal to time domain.

$$Denoised\ Time\ Domain\ Signal = ISTFT(Denoised\ STFT) \quad (4.3)$$

4.2.1 Learning Parameters

The hyper-parameters involved in deep neural network training include batch size, number of epochs, optimization technique, learning rate, and loss function. The optimization function, “adam” is predominantly used for signal optimization. The process flow for implementing the deep learning algorithms is given in Figure 4.2. The predictors and the targets are reshaped to the dimension of the expected deep learning network.

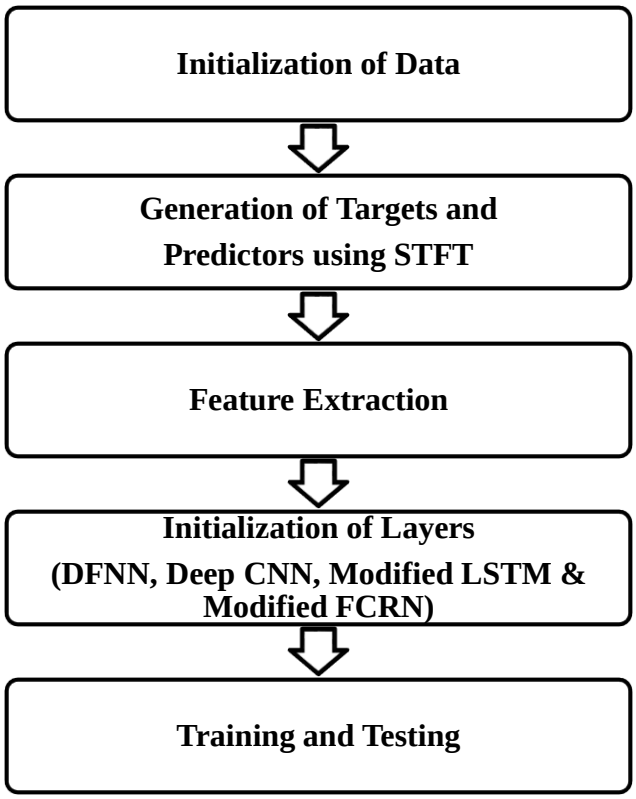


Figure 4.2 Process Flow of Implementation

The magnitude of the noisy audio signals is extracted using Short Term Fourier Transform (STFT). By reducing the mean square error between the denoised and clean speech signals, the regression network utilizes the magnitude of the noisy speech signal. Combining the output magnitude spectrum from the Deep network with the phase of the noisy signal generates an output time domain signal. DFNN, Deep CNN, modified LSTM, and modified FCRN are considered for the study to evaluate the best-performing algorithm.

4.3 DEEP FULLY CONNECTED NEURAL NETWORK (DFNN)

DFNN significantly outperforms traditional speech enhancement algorithms such as spectral subtraction, Wiener filtering, and subspace methods. DFNN provides superior performance in enhancing speech quality by learning complex patterns in noisy data and adapting to various noise types and levels through training on diverse datasets. While DFNNs are computationally intensive and require significant processing power and memory, the investment is justified by their remarkable effectiveness. Despite the need for extensive training with large datasets and the challenge of achieving real-time performance, DFNNs excel in scenarios with varying and complex noise conditions. Although less complex and more feasible for real-time processing, traditional algorithms need more adaptability and scalability. They rely on predefined rules and mathematical models, which limits their effectiveness in diverse noise environments. Despite their higher computational demands, DFNNs offer unparalleled adaptability and performance, making them a superior choice for speech enhancement.

DFNN is an artificial neural network consisting of multiple layers of fully connected neurons (Goodfellow et al., 2016). Each neuron in one layer is connected to every neuron in the next layer, and the input to the network is propagated forward through these layers to produce the output. It computes the gradient of the loss function for the network weights and is very efficient, rather than directly computing the gradient for each weight. This efficiency makes it possible to use gradient methods to train multilayer networks and update weights to minimize loss. This is done by adopting the backpropagation algorithm, which calculates the gradient of the loss function for the weights and biases of the network and uses these gradients to update the parameters through gradient descent (LeCun et al., 2012).

Backpropagation refers to the "backpropagation of errors" and is helpful in training neural networks. It compares the generated output to the desired output and generates an error report if the result does not match the generated output vector. Then, it adjusts the weights until the desired output is obtained.

The mathematical expression for calculating the output is given in equations 4.4 and 4.5 (Rumelhart et al., 1986).

Considering the input vectors, $I = [p_1, p_2, \dots\dots\dots, p_n]$, the output of the hidden layer can be calculated as,

$$n = IW + b \tag{4.4}$$

$$a = f(n) = \frac{2}{\left(1+e^{(-2*n)}\right)^{-1}} \tag{4.5}$$

Where W is the weight vector, and b is the bias input (Ramachandran et al., 2017) Sitzmann et al., (2020). Errors are calculated as the difference between the target output and the actual output of the network. The target is to minimize the average difference between them to minimize the sum of these errors. The calculation of MSE is given in equation 4.6 (Bishop, 2007).

$$\text{Mean Square Error } (MSE) = \frac{1}{M}\sum_{k=1}^{M} e(k)^2 = \frac{1}{M}\sum(t(k) - a(k))^2 \tag{4.6}$$

Where $a(k)$ is the network output, $t(k)$ represents the target output. The algorithm becomes more accurate by updating weights and bias values based on the goal average error value (Gupta & Raza, 2020).

Eight different noise signals taken for training are Washing Machine noise, Rainbow noise, Train whistle noise, Jet Airplane noise, Babble noise, Street noise, Airport noise, and Restaurant noise at different noise levels of -10dB, -5dB, 0dB, 5dB, 10dB, and 15dB. The performance of DFNN is also analyzed by testing the network with unseen noises, such as car and subway noise. The values of SNR, segSNR, PESQ, STOI, SI-SDR, and DNSMOS are evaluated to analyze the performance of the enhanced speech.

4.3.1 Structure of DFNN

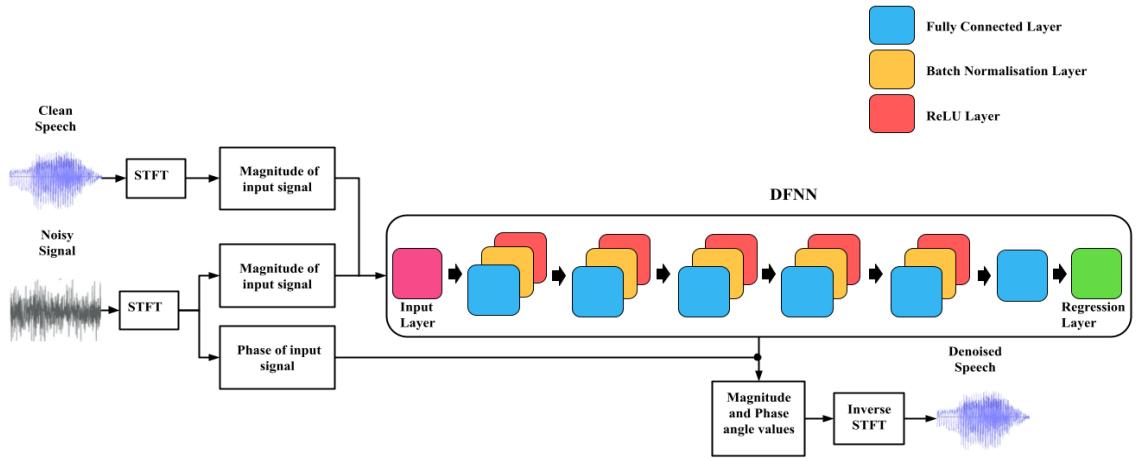


Figure 4.3 Structure of DFNN

The input layer contains one channel audio input. The six fully connected layers in the network are made to fit the dimensionality of the input and output sequences. There are five batch normalization layers and five ReLU layers; as activation and normalizing functions, they do not have any channels or hop. Lastly, a regression layer predicts the clean speech signal matching the output sequence dimension.

The input signals (clean and noisy speech) are first converted into their frequency domain representations using STFT. This transformation decomposes the signal into its magnitude and phase components Allen & Rabiner, (1977). Both clean and noisy speech signals are represented by their magnitudes after STFT. The phase information is also extracted for the noisy signal (Cheng et al., (2018).

The input to the DFNN includes the magnitude of the noisy signal and phase information, as shown in Figure 4.3 (Goodfellow et al., 2016). The network consists of multiple hidden layers, each comprising different types of layers, such as Fully Connected Layers that perform linear transformations followed by non-linear activations (Ioffe & Szegedy, 2015). Batch Normalization Layers are used to normalize the output of the previous layer and improve training stability and speed. The ReLU layer applies the ReLU activation function to introduce non-linearity (Glorot et al., 2011).

The final layer of the network is a regression layer that outputs the denoised speech signal by combining the magnitude and phase information. The denoised speech is

converted to the time domain by combining the output magnitude spectrum from the network with the phase of the noisy signal. Vihari et al., (2016)

4.3.2 Experiment

The DFNN is trained using 320 sentences from the CSTR database belonging to normal speakers. For testing, 80 sentences are selected from the test set. The network is trained using 20 hours of training data. The speech signal is processed at a sampling rate of 16000 samples per second with a window length of 480 samples. Batch normalization and dropout layers are introduced in the Deep Fully Connected Neural Network's training to normalize the input and to solve overfitting concerns. The mini-batch size of the network is 128, the learn rate drop factor is 0.9, the learn rate drop period is 1, and the optimizer used for training is "adam."

(i) 15 Epochs

During the training process of the DFNN model for 15 epochs, the RMSE showed a sharp initial decline followed by a gradual decrease, eventually stabilizing around 1.5802. Both the training and validation loss decreased rapidly at first and then leveled off, indicating a stable training process with minimal overfitting. Key details include the completion of 15 epochs over 22,125 iterations, utilizing a piece-wise learning rate schedule.

(ii) 30 Epochs

The training of the DFNN to 30 epochs exhibited a sharp initial decline in RMSE, which gradually stabilized around 1.8, culminating in a final validation RMSE of 1.2825. The training for 30 epochs, amounting to a total of 44,250 iterations, and employed a piece-wise learning rate schedule. The entire process took approximately 206 minutes. The close alignment between training and validation metrics indicates effective training with minimal overfitting.

(iii) 50 Epochs

During the training of the deep learning model, the RMSE showed a notable initial drop, followed by a gradual decrease that ultimately stabilized around 0.872. The training and validation loss exhibited a similar trend, with a rapid initial decline before leveling off

with minor fluctuations. The training was conducted over 50 epochs, involving 73,090 iterations, for a duration of 487 minutes. The alignment between training and validation metrics suggests that the model is well-fitted indicating strong training performance.

(iv) 62 Epochs

The training of DFNN is conducted to enhance noisy speech, with the learning rate set to 1.617e-06, and manual stopping employed at 62 epochs as the performance metrics demonstrated significant improvement. This configuration achieved a validation RMSE of 0.4761, reflecting a notable enhancement in the model's performance. The chosen learning rate allowed for effective fine-tuning of the model's parameters, enabling precise adjustments and better generalization. The training duration and manual stopping were sufficient for the model to refine its understanding of the data, leading to a considerable reduction in RMSE and overall improvement in performance.

(v) Inference

The data for the study is in the ratio of 80:20 for the training and testing. The trials in the training process for the DFNN model is increased till 62 epochs. As the number of epochs increases from 15 to 50, the RMSE values progressively decreases (from 1.5802 to 0.4761). This indicates that longer training allows the model to fit the data better and reduce errors. The decreasing RMSE with more epochs suggests that the model becomes more effective at noise removal in the speech signal over time. The improved accuracy with additional epochs indicates the model's enhanced ability to discern and filter out noise, leading to cleaner outputs. Across all epoch configurations, the loss metrics showed a rapid initial decrease followed by stabilization. This pattern indicates that the model quickly learns the fundamental patterns in the data early in the training process and then fine-tunes these patterns with further training. The consistent decrease and stabilization in RMSE and loss across different epochs highlight the effectiveness of the piece-wise learning rate schedule. This schedule helped the model make substantial progress and refine its learning with smaller updates. Although longer training (up to 62 epochs) results in better performance, it also significantly increases the training time (from 60 minutes to 3323 minutes). This trade-off between training duration and model performance is crucial for practical applications where computational resources and time are limited. Finally,

increasing the number of epochs improves the model's performance in terms of RMSE, indicating better noise removal in the speech signal. However, this comes at the cost of longer training times, highlighting a balance that needs to be struck based on available resources and desired performance levels. The results obtained for DFNN for quality and intelligibility metrics is given in Tables 4.2 to 4.7. The spectrograms obtained are included in the Appendix 1 (Figures 1 to 10).

4.4 DEEP CONVOLUTIONAL NEURAL NETWORK (DEEP CNN)

The current state-of-the-art speech enhancement leverages the power of Deep CNN to reduce noise and improve speech quality effectively. Deep CNNs capture intricate temporal and spectral patterns in speech signals, making them highly effective for this task (Kumar Shukla et al., 2024). Deep CNNs excel in learning hierarchical representations directly from raw data. Without handcrafted feature engineering, they can automatically extract relevant features from noisy speech signals. Deep CNNs are computationally intensive, especially during training, as they involve numerous convolutions and nonlinear operations layers. They can capture complex relationships between noisy and clean speech signals, leading to robust denoising performance across various conditions.

Deep learning uses learning methodologies to create a model based on its data. Because it is computationally efficient, convolutional neural networks can learn complicated patterns from speech signals Sarker, (2021). ReLU is used for speech enhancement to connect multiple hidden layers. Deep CNNs can denoise noisy speech signals when presented with the frames of noisy signals. Using Short-Time Fourier Transforms (STFT), clean and noisy speech signals are converted into frequency domains. The magnitude spectrum of the clean speech is taken as the target, and the noisy speech signal is taken as the predictor. With the regression network Xu et al. (2015), the magnitude of the noisy speech signal is used to reduce the mean square error between the denoised speech signal and the clean speech signal. The output denoised speech signal of the Deep CNN Model is in the frequency domain. By using the output magnitude spectrum and phase of the noisy speech signal, the denoised speech signal is converted to the time domain. Deep learning learns the spectral mapping between noisy and clean speech signals.

4.4.1 Activation Functions

There are two activation functions, namely, SoftMax and ReLU.

SoftMax

The normalized exponential function is the most common function for the last layer of a neural network as it normalizes the outputs into a probability distribution consisting of K probabilities (given K input points) as given in equation 4.7 (Goodfellow et al., 2016)

. $$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \quad (4.7)$$

ReLU

A popular activation function unit is the rectified linear unit (ReLU), and it is remarkably efficient in many models despite its simplicity.

The ReLU f(x) → output is given in equation 4.8

$$f(x) = \begin{pmatrix} x, x \geq 0 \\ 0, x < 0 \end{pmatrix} \quad (4.8)$$

where x is the input value. The ReLU ensures that the gradients will not explode as it is constant for values x ≥ 0. However, with ReLU, there is an issue where nodes can die. Since values less than 0 will not affect the network, a situation can emerge where the node is stuck in the negative space, which effectively means that the node stops contributing, i.e., it dies. Leaky ReLU To combat the issue of ReLU, Leaky ReLU is used. The activation function is given by equation 4.9.

$$f(x) = \begin{Bmatrix} x & x \geq 0 \\ \alpha x, & x < 0 \end{Bmatrix} \quad (4.9)$$

The extra parameter $0 < \alpha < 1$ ensures that the negative values contribute, to a small degree. The standard value of α is set to 0.01 (Dubey et al., 2022).

The clean speech and the noisy signal are converted from the time domain to frequency domain using short-time fourier transform (STFT). This transformation results in a complex-valued representation that captures the signal's frequency components over time. The magnitude spectra of both the clean and noisy signals are extracted from their STFT representations, representing the amplitude of each frequency component. The magnitude spectra of the clean and noisy signals are fed into the Deep CNN.

4.4.2 Structure of Deep CNN

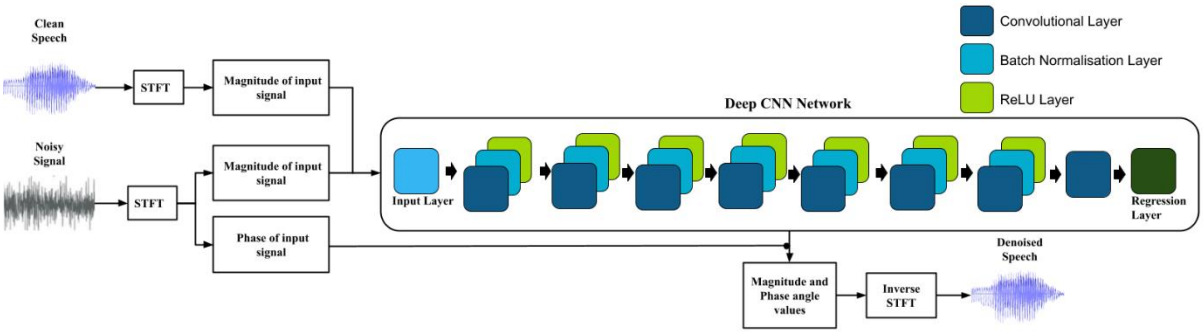


Figure 4.4 Structure of Deep CNN

The input layer has one channel for audio input. The network comprises eight convolutional layers, each with a stride and hop for each convolutional layer, starting with 16 channels and typically extending to 512 channels, to enable detailed feature extraction. The convolutional layers are followed by seven ReLU layers and seven batch normalization layers. Finally, the regression layer predicts the single value and matches the output sequence dimension.

The denoising algorithm utilizes the Convolution Layer, Batch Normalization Layer, and ReLU Layer as shown in Figure 4.4. The network consists of a Convolution Layer that applies convolutional filters to the input, capturing local patterns in the magnitude spectrum. The layer has a 3x3 filter with stride 1 and no padding. The batch Normalization Layer normalizes the output of the convolutional layer to stabilize and speed up the training process. ReLU Layer applies the Rectified Linear Unit activation function, setting all negative values to zero, introducing non-linearity, and helping to avoid the vanishing gradient problem. The regression layer maps the features extracted by the convolutional layers to the target output, the clean magnitude spectrum. The output of the Deep CNN is the estimated magnitude spectrum of the clean speech. This is combined with the phase spectrum of the noisy signal to reconstruct the denoised speech signal. The estimated clean magnitude spectrum and the original phase spectrum of the noisy signal are combined and converted back to the time domain using the ISTFT. This process reconstructs the denoised speech signal as close to the original clean speech as possible. This method leverages deep learning to learn the mapping between noisy and clean speech spectra, effectively denoising the speech signal by enhancing the magnitude spectrum while preserving the phase information from the noisy input.

4.4.3 Experiment

The Deep CNN is trained using 320 sentences from the CSTR database belonging to normal speakers. For testing, 80 sentences are selected from the test set. Over 20 hours of training data are used to train the network. The speech signal with a sampling rate of 16000 samples per second is processed using a window length of 480 samples.The training models for CNN use batch normalization to normalize the input speech signal. The network's parameters include a mini-batch size of 128, learning rate of 0.001, learn rate drop factor of 0.9, and learn rate drop period of 1. The optimizer used during training is called "adam".

(i) 15 Epochs

In the process of training the Deep CNN for 15 epochs, the RMSE indicates a significant initial drop followed by a gradual decrease, eventually stabilizing around 1.7243. The training and validation loss, decreases rapidly initially and remains relatively stable with minor fluctuations. Key training details include completion of 15 epochs, 22365 iterations, piece-wise learning rate schedule, and final learning rate of 0.0002877. The entire training process lasted approximately 74 minutes.

(ii) 30 Epochs

During the training of the Deep CNN to 30 epochs, the RMSE showed a steep initial decline, followed by a gradual stabilization around 1.102. The training and validation loss also decreased sharply at the beginning and then remained stable with slight fluctuations. The model completed 30 epochs, totaling 44,640 iterations, utilizing a piece-wise learning rate schedule, with a final learning rate of 0.0000471. The entire training process took approximately 570 minutes on a single CPU, reflecting an extensive and thorough training effort.

(iii) 50 Epochs

The RMSE exhibited a substantial initial decline at the start of Deep CNN training, followed by a gradual decrease, eventually stabilizing at approximately 0.7667. The loss for both training and validation also decreased sharply at the beginning and then remained relatively stable, with minor fluctuations. The model was trained for 50 epochs, completing

73,950 iterations, using a piece-wise learning rate schedule that concluded with a final learning rate of 0.0000571. The entire training process took approximately 2,370 minutes reflecting a comprehensive and extended training effort.

(iv) 55 Epochs

The training is conducted to enhance noisy speech, with the learning rate set to 3.3814e-06, and manual stopping employed at 55 epochs as the performance metrics showed significant improvement. This setup achieved a validation RMSE of 0.4226, reflecting a marked improvement in the model's performance. The chosen learning rate facilitated effective fine-tuning of the model's parameters, enabling precise adjustments and improved generalization. The training duration was sufficient for the model to refine its understanding of the data, resulting in a substantial reduction in RMSE and an overall improvement in performance metrics.

(v) Inference

The data for the study is in the ratio of 80:20 for the training and testing. The trials in the training process for the Deep CNN model is increased till 55 epochs. As epochs increased from 15 to 55, the RMSE values progressively decreased (from 1.7243 to 0.4226). This indicates that extended training allows the model to fit the data better and reduce errors. The decreasing RMSE with more epochs suggests that the model becomes increasingly effective at noise removal in the speech signal over time. Improved accuracy with additional epochs demonstrates the model's enhanced ability to discern and filter out noise, resulting in enhanced speech. Across all epoch configurations, the loss metrics showed a rapid initial decrease followed by stabilization. This pattern indicates that the model learns fundamental patterns early in training and then fine-tunes these patterns with further training. The consistent decrease and stabilization in RMSE and loss across different epochs highlight the effectiveness of the piece-wise learning rate schedule. This schedule likely facilitated substantial initial progress and refinement through smaller updates. Although longer training (up to 55 epochs) results in better performance, it significantly increases training time (74 to 3683 minutes). This trade-off between training duration and model performance is crucial for practical applications where computational resources and time are limited. Ultimately, increasing the number of epochs enhances the

model's performance in terms of RMSE, indicating better noise removal in the speech signal. The results obtained for Deep CNN for quality and intelligibility metrics is given in Tables 4.2 to 4.7. The spectrograms obtained are included in the Appendix 2 (Figures 1 to 10).

4.5 LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM) networks, a form of recurrent neural networks (RNNs), have reshaped the landscape of speech enhancement, offering robust performance in noisy environments Hochreiter & Schmidhuber, (1997). Unlike traditional algorithms like the iterative wiener filter, Wavelet Wiener, and Least Mean Square (LMS), LSTMs capture intricate temporal patterns and dependencies within speech signals. Traditional methods often rely on statistical models of noise and speech, making assumptions that may not fully encapsulate real-world variability. LSTMs dynamically adapt to varying noise conditions by leveraging long-term contextual information without explicit noise modeling, resulting in more natural and intelligible enhanced speech. Thus, while traditional approaches have paved the way for speech enhancement, LSTMs represent a significant leap forward by harnessing deep learning for superior performance in unpredictable environments.

LSTM consists of several key components, such as the memory cell state, the input gate, the forget gate, the output gate, and the peepholes, and represents the input and output vectors of the block at time t, as shown in Figure 4.5. The logistic sigmoid non-linear activation function is employed in every gate. As a result, the network can dynamically decide what information to update, store, discard, and output (Lu & Salem, 2017). For the separation of noise from speech, the LSTM can capture the inherent statistical properties of speech and noise, particularly under non-stationary noise conditions (Tkachenko et al., 2017).

A model should consider the long-term context when dealing with speech separation and model the temporal dependencies. The LSTM is a Long short-term memory that provides a solution for separating noise from the speech signal. A memory cell, introduced by RNN to facilitate information exchange over time, mitigates this problem. LSTM comprises three distinct components: the forget gate, the input gate, and the output gate. LSTMs acknowledge and remember information entering a network and discard

unnecessary information. By utilizing the forget state feature, unimportant information has been removed from the network state. Sequential data, like time and audio data, are analyzed using the LSTM network Tang et al. (2019).

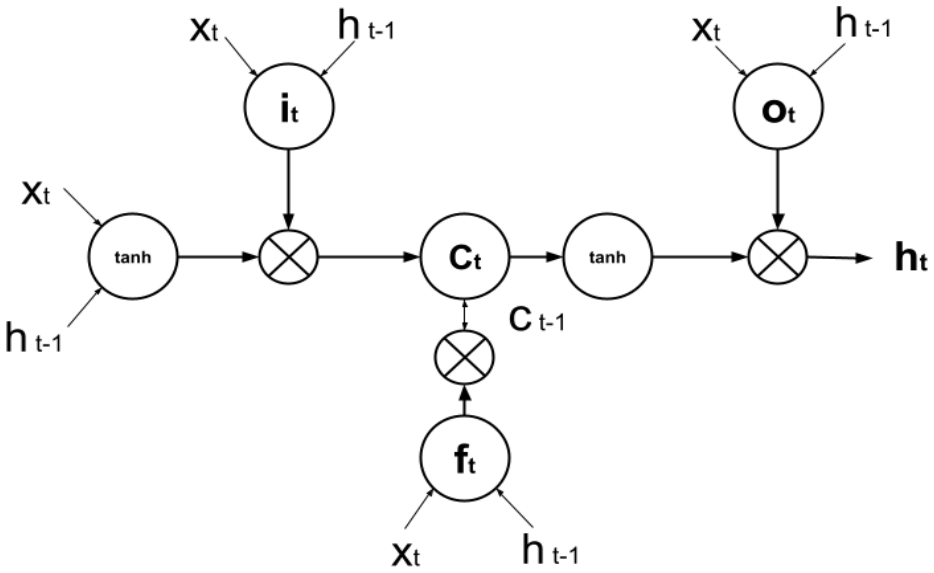


Figure 4.5 Illustration of LSTM

Information can be added to or removed from the cell state in LSTM and is regulated by gates. Alternatively, these gates allow information to flow into and out of the cell. It contains a pointwise multiplication operation and a sigmoid neural net layer that supports the mechanism.

The implementation of the LSTM network (Staudemeyer & Morris, 2019) is given in equation 4.10 to equation 4.14.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{4.10}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \tag{4.11}$$

$$c_t = f_t \otimes c_{t-1}i_t + \tanh \otimes (W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{4.12}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{4.13}$$

$$h_t = o_t \otimes \tanh\,(c_t) \tag{4.14}$$

where, t is the frame index, σ is the logistic function, W and b denotes the weights and bias from cell to gate, respectively. Three hundred speech utterances train the LSTM network with various noise types and levels.

4.5.1 Modified LSTM Technique

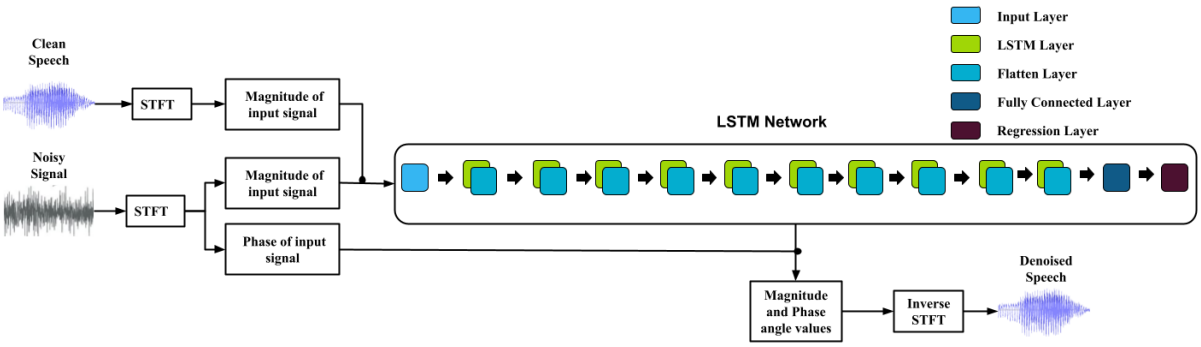


Figure 4.6 Structure of Modified LSTM Technique

The LSTM technique comprises of LSTM layers which is composed of multiple LSTM units, with each unit containing input gate, forget gate, output gate and a memory cell that stores information over time. The combination of these gates allows LSTMs to capture long-term dependencies in data while discarding irrelevant information. Multiple LSTM layers can be stacked to build a deep LSTM network, enabling it to capture even more complex temporal patterns.

The process starts with two input signals: clean speech and noisy speech. These signals are represented as time-domain waveforms. The clean and noisy speech signals are applied with a Short-Time Fourier Transform (STFT) to convert them from the time domain to the frequency domain. STFT helps analyze the signal in small time segments, providing a time-frequency representation. STFT produces two components for each signal: the magnitude and the phase of the input signal. The magnitude components of both clean and noisy signals are fed into the modified LSTM network. Sequence Input Layer takes the sequence of magnitude values as input. Multiple LSTM layers are used to learn the temporal dependencies in the input sequence. LSTMs effectively handle long-term dependencies due to their gating mechanisms (input gates, forget gates, and output gates) Tang et al. (2019). Flatten layer flattens the output from the LSTM layers, preparing it for further dense layers. A fully connected layer maps the learned features to the output space.

Regression Layer outputs the denoised magnitude values. The modified LSTM network's output, consisting of denoised magnitude values, is combined with the original phase angle values of the noisy signal. This combination is essential because the phase information is crucial for accurately reconstructing the time-domain signal. The combined magnitude and phase values undergo an inverse STFT process to transform the signal from the frequency domain to the time domain, resulting in the denoised speech signal.

In the process of enhancing the noisy speech, the LSTM layers are added and its efficacy on improving the performance metrics are evaluated. It is tested with 8,9,10 and 15 layers. Figure 4.6 shows the structure of the modified LSTM technique with 10-layers.

4.5.2 Experiment

The modified LSTM technique is trained using 320 sentences from the CSTR database belonging to normal speakers. For testing, 80 sentences are selected from the test set. The data for the study is in the ratio of 80:20 for the training and testing. The training is performed with different LSTM layers to determine the one that is best suited for enhancing noisy speech. The architecture of the LSTM network is modified to tune the network to enhance the noisy speech signal.

The training data for the network is around 20 hours. The speech signal of sampling rate 16000 samples per second with the window length of 480 samples is considered. The network's parameters include a mini-batch size of 128, an initial learning rate of 0.001, a learn rate dropout factor of 0.9, a learn rate dropout period of 1, and the optimizer used during training is "Adam". The dropout factor is used to overcome the overfitting issue.

8-layer LSTM with MSE Loss

The 8-layer LSTM with MSE loss shows moderate performance during training. The training RMSE stabilizes around 4, and the validation RMSE achieves a value of 4.9553 when the training is done for 50 epochs. The decreasing loss curves confirm that the model is learning, but the error is not minimized and final validation RMSE is 4.6799.

9-layer LSTM with MSE loss

The performance of the 9-layer LSTM technique with MSE loss is better than the 8-layer LSTM with MSE loss. The reduction in RMSE is observed during training, but the

expectation of reducing the MSE needs to be met. Training for an increased epoch does not aid in reducing RMSE and improving performance metrics.

10-layer with MSE loss

The modified LSTM technique consists of ten LSTM layers, with 512 units per layer, and focuses on temporal sequence modeling without a hop size. Following the LSTM layers, 10 flatten layers are used to convert the multi-dimensional input into a one-dimensional vector. The fully connected layer often matches the dimensions needed for the output sequence to guarantee that it can handle the denoised signal appropriately, even without a hop size. The regression layer helps in matching the output sequence dimension.

(i) 10 Epochs

Initially, a moderate learning rate of 0.00038742 over 10 epochs is used, the validation RMSE is 3.5401, indicating substantial room for improvement. The high RMSE suggests that the model is in the early learning stages and yet to capture data patterns fully.

(ii) 25 Epochs

During the training of the modified LSTM technique to 25 epochs, the validation RMSE dropped to 2.7578, reflecting significant improvement. The reduction in learning rate, coupled with the extended training, facilitated more precise parameter updates, promoting better convergence and further reducing the RMSE.

(iii) 50 Epochs

The learning rate was further reduced to 5.7264e-06, and training continued for 50 epochs. The RMSE exhibited a substantial initial decline before gradually stabilizing around 0.7667. The loss for both training and validation sharply decreased at the start, then remained relatively stable with minor fluctuations. The model completed 73,950 iterations, following a piece-wise learning rate schedule with a final learning rate of 0.0000571. The training process reached 2,370 minutes, providing ample time for the model to learn effectively, leading to a significant improvement in the validation RMSE.

(iv) 67 Epochs

In the process of training, the learning rate is set to 0.5501e-07, and training is conducted for 67 epochs before being manually stopped. This configuration achieved a validation RMSE of 0.4651, highlighting a significant enhancement in the model's performance. The relatively low RMSE indicates that the model's predictions were well-aligned with the actual data, demonstrating improved accuracy. The selected learning rate allowed for effective fine-tuning of the model parameters, enabling more precise adjustments and enhanced generalization. The training duration provided sufficient time for the model to better understand the data, leading to a considerable reduction in RMSE and overall improvement in performance metrics.

The 10-layer LSTM with MSE loss balances model complexity and the ability to generalize for unseen data. During training, the training RMSE steadily decreases as the model learns and improves its fit to the training data. Simultaneously, the validation RMSE decreases, demonstrating that the model generalizes well to unseen data. The 10-layer LSTM achieves good improvement in performance metrics at 67 epochs.

15-layer with MSE loss

The training progress of the 15-layer LSTM technique with MSE loss for 50 epochs shows that the training RMSE and validation RMSE reduce with an increase in epochs, and after a specific duration, the training RMSE decreases rapidly, but the validation RMSE remains constant and starts to increase. This shows that the model starts to overfit, and it cannot adapt to unseen data. To validate that the 15-layer LSTM with MSE loss results in overfitting, the training is continued till 67 epochs, which shows that with increased epochs, the training RMSE keeps decreasing and the validation RMSE increases and reaches 2.4815.

(v) Inference

The baseline LSTM technique by Tang et al. (2019) consists of LSTM layers for speech denoising and Joint Progressive Learning (JPL framework) for eliminating reverberation. 3-LSTM layers have been used for denoising. The conventional LSTM technique is modified by changing the LSTM layers. The 8-layer LSTM with MSE loss shows moderate performance but does not fully minimize the error even with increased epochs. The 9-layer LSTM with MSE loss shows better performance but still does not

significantly improve with more epochs. The 10-layer LSTM with MSE loss balances model complexity and the ability to generalize for unseen data. The 15-layer LSTM with MSE loss overfits the data, where the validation RMSE starts increasing while training RMSE decreases.

In this research work, the 10-layer LSTM technique shows better performance in enhancing the speech compared to the 8-layer, 9-layer and 15-layer LSTM techniques. From the results shown in Table 4.1, it is observed that the 10-layer LSTM technique performs better on all performance metrics such as SNR (33.16), segSNR (11.13), PESQ (2.74), STOI (0.75), SI-SDR (8.71) and DNSMOS (3.45). The performance metrics of 15-layer LSTM shows that performance declines as the 15-layer LSTM technique results in overfitting.

Table 4.1 Performance Metrics for modified LSTM layers

Layer	Epoch	SNR	segSNR	PESQ	STOI	SI-SDR	DNSMOS
8	50	22.45	4.84	1.12	0.54	6.11	1.98
	67	23.01	4.96	1.20	0.56	6.5	2.1
9	50	22.65	5.12	1.15	0.55	6.20	2.05
	67	23.5	5.29	1.25	0.57	6.7	2.2
10	50	32.47	10.58	2.69	0.73	8.66	3.32
	67	33.16	11.13	2.74	0.75	8.71	3.45
15	50	22.73	5.19	1.23	0.52	6.48	2.14
	67	22.64	5.07	1.21	0.49	6.39	2.13

The trials in the training process for the modified LSTM technique is increased till 67 epochs. The training of the 10-layer LSTM technique with MSE loss demonstrated significant improvements in model performance, as evidenced by the consistently decreasing validation RMSE. The progressive training strategy of lowering learning rates and increasing training durations proved highly effective. The marked reduction in validation RMSE throughout the phases highlights the robustness of this approach, leading to significant improvements in model accuracy and generalization capabilities. This

training regimen highlights the modified LSTM technique's ability to learn efficiently from data and adapt to complex patterns with continued refinement.

In this research work, the 10-layer LSTM technique shows better performance in enhancing the speech compared to the 8-layer, 9-layer and 15-layer LSTM techniques. From the results shown in Table 4.1, it is observed that the 10-layer LSTM technique performs better on all performance metrics such as SNR (33.16), segSNR (11.13), PESQ (2.74), STOI (0.75), SI-SDR (8.71) and DNSMOS (3.45). The performance metrics of 15-layer LSTM shows that performance declines as the 15-layer LSTM technique results in overfitting. The results obtained for modified LSTM technique for quality and intelligibility metrics is given in Tables 4.2 to 4.7. The spectrograms obtained are included in the Appendix 3 (Figures 1 to 10).

4.6 FULLY CONVOLUTIONAL RECURRENT NETWORK (FCRN)

The current state of the art in speech enhancement involves a variety of deep learning approaches, including Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). The most recent deep learning algorithm FCRN effectively captures both the spectral and temporal features of speech signals. The model can effectively reconstruct clean speech signals even in significant noise by incorporating both magnitude and phase information during denoising.

The speech enhancement process involves deep neural network architectures integrating convolutional and recurrent layers to improve performance in single-channel speech enhancement tasks. Convolutional Recurrent Neural Networks (CRNs) leverage Convolutional Encoder-Decoder (CED) structures to handle the temporal and spectral characteristics of speech. These models use convolutional layers for feature extraction and Long Short Term Memory (LSTM) layers for capturing temporal dependencies. While effective, CRNs face challenges with managing internal representations, as LSTMs often require discarding the structured organization of feature maps from convolutional layers in favor of fully connected processing. This can lead to inefficiencies in feature handling and an increased number of trainable parameters.

The modified FCRN technique represents a hybrid architecture that combines the strengths of both convolutional and recurrent layers. Convolutional layers effectively capture spatial features and patterns, while recurrent layers excel in modeling temporal

dependencies and sequential context. By combining these two layers, FCRNs offer a unique capability to process sequential data with variable temporal lengths suitable for speech enhancement. Strake et al., (2020)

The architecture of modified FCRN technique for speech enhancement is typically composed of three main parts:

i. A modified convolutional layer that extracts feature from the noisy speech signal.

ii. LSTM layer that captures the temporal dependency of the noisy speech signal. Modeling these temporal relationships helps the network distinguish between noise and speech more effectively.

iii. A loss function that measures the difference between the enhanced and clean speech signals.

The modified layer consists of convolutional layers denoted by Conv, which determines the number of filter kernels and the kernel size on the feature axis. Additionally, maximum pooling layers with a stride of 2 are on the feature axis, and the convolutional layers employ leaky ReLU activations.

Convolutional Layer

The convolutional layer is used to extract local features from the input sequence. It applies a set of learnable filters (kernels) to the input, producing feature maps highlighting specific patterns. The convolutional operation can be represented in equation 4.15

$$h_i = f(\sum_{j=1}^{N} W_j x_{i+j-1} + b) \quad (4.15)$$

where,

h_i is the output feature at position i in the feature map

f is the activation function (e.g., ReLU)

N is the size of the filter

W_j is the learnable weight for the j element in the filter

x_{i+j-1} is the input element at position $i + j - 1$

b is the bias term

Max Pooling

Max pooling layers are commonly used to reduce the spatial dimensions of the feature maps while preserving the most important information. The max operation selects the highest value in the subsequence as given in equation 4.16 Gholamalinezhad & Khosravi, (2020).

$$h_{pool} = \max\left(h_{i:i+k-1}\right) \tag{4.16}$$

where,

h_{pool} is the output of the max pooling layer

$h_{i:i+k-1}$ represents the subsequence of length k starting from position i in the feature map.

Bidirectional LSTM

LSTM is a type of recurrent neural network (RNN) designed to handle sequential data by maintaining hidden states and cell states that capture long-term dependencies (Shabanian et al., 2017). A bidirectional LSTM processes the input sequence forward and backward, providing a more comprehensive understanding of the temporal context. The forward and backward LSTM operations are represented in equation 4.17 and equation 4.18 (Hochreiter & Schmidhuber, 1997).

Forward LSTM

$$\text{forward}_{\text{h}_\text{t}} = \text{LSTM}(\text{forward}_{\text{h}_{\text{t}-\text{i}}}, \text{x}_\text{t}) \tag{4.17}$$

Backward LSTM

$$\text{Backward}_{\text{h}_\text{t}} = \text{LSTM}(\text{backward}_{\text{h}_{\text{t}+\text{i}}}, \text{x}_\text{t}) \tag{4.18}$$

where,

$\text{forward}_{\text{h}_\text{t}}$and $\text{backward}_{\text{h}_\text{t}}$ are the forward and backward hidden states at time step t, respectively

x_t is the input at time step 't'

Concatenation

To combine information from both the forward and backward LSTMs, the outputs of the two directions are concatenated as given in equation 4.19

$$h_t = \left[forward_{h_t}.\ backward_{h_t}\right] \tag{4.19}$$

where,

h_t is the concatenated hidden state at time step 't'

[·] represents the concatenation operation

4.6.1 Modified FCRN Technique

In the FCRN algorithm, convolutional layers are responsible for learning spatial features by capturing patterns in local areas of the input, while the recurrent layers are designed to handle sequential dependencies. The modified FCRN structure builds on the standard FCRN by integrating two bi-directional LSTM (Bi-LSTM) layers, along with the convolutional layers, significantly enhancing its ability to capture temporal dependencies.

The baseline FCRN by Strake et al. (2020), comprises of the LSTM that works well in the removal of noise but the encoder-decoder part is changed and modified convolutional layers are added to increase the efficiency of the network in speech enhancement. The modified FCRN consists of two bi-directional LSTM (Bi-LSTM) layers that process the sequence data in both forward and backward directions, allowing the network to leverage context from both past and future frames and the convolutional layers which significantly enhances the network's ability to learn temporal dependencies and manage different noise levels, making it very adaptable for real-world applications.

The modified FCRN consists of input layer with one channel. Seven convolutional layers with the first layer having 16 channels, a 3x3 kernel and the seventh layer having 1024 channels with a stride of 2 and single hop for each convolutional layer. The layers get increasingly more complex over time. With an average of 480 units, each of the two bi-LSTM layers focuses on temporal sequence modeling. The number of units in the fully connected layer typically equals the output sequence dimension to guarantee that it can process the denoised signal accurately. Lastly, the regression layer is utilized.

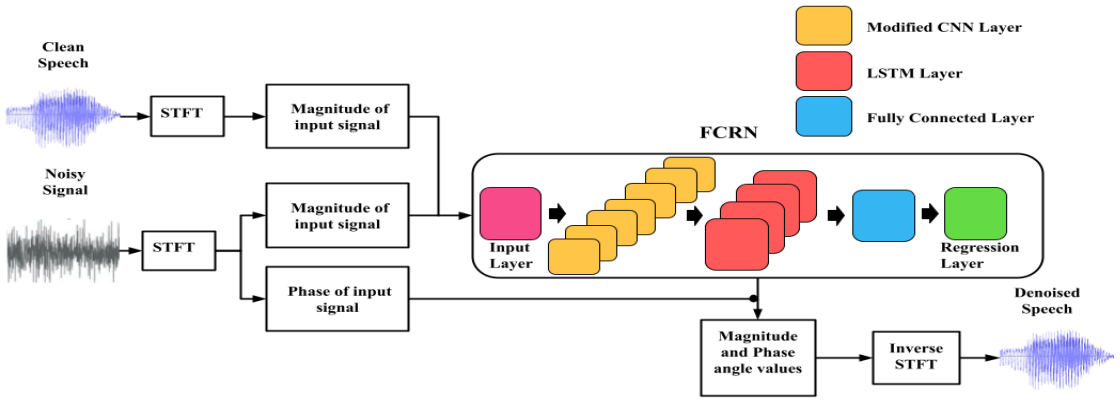


Figure 4.7 Structure of Modified FCRN Technique

Figure 4.7 illustrates the modified Fully Connected Recurrent Neural (modified FCRN) technique for speech denoising. The process begins with clean speech and a noisy signal being converted from the time domain to the frequency domain using a Short-Time Fourier Transform (STFT). The magnitudes of both the clean and noisy signals are extracted from these transformed signals, while the phase information is only extracted from the noisy signal.

The magnitude of the noisy signal is given as input to the network. The network starts with an input layer followed by several modified Convolutional Neural Network (CNN) layers designed to extract local features from the input magnitude. The modified CNN layer is often made up of 1D and 2D convolutions. 1D convolutions are applied along the time axis to capture temporal correlations, and 2D convolutions are performed on frequency domain representations to capture time-frequency patterns. These layers are used to capture local patterns in the magnitude of the noisy signal. CNNs are proficient at recognizing spatial hierarchies and local dependencies, which is crucial for identifying and mitigating noise components in speech. The modified CNN layers process the input magnitude to extract high-level features. These layers help distinguish noise from speech signals based on spatial patterns.

The network includes bi-LSTM layers, which capture temporal dependencies in the data, leveraging the sequential nature of speech. Bi-LSTM layers are designed to handle time-series data, making them ideal for speech processing. They remember information over long sequences, allowing the network to maintain context and continuity, essential for effective noise reduction. The Bi-LSTM layers are fully connected layers that perform

further processing and refinement of the features. The fully connected layer aggregates the learned features and makes predictions about the clean magnitude, effectively reducing noise (Strake et al., 2020). The regression layer outputs the estimated clean magnitude based on the learned features.

The output from the modified FCRN provides the enhanced magnitude spectrum, which is combined with the original phase information from the noisy signal. This combination yields the magnitude and phase angle values needed to reconstruct the denoised speech signal. Finally, the inverse STFT is applied to convert the processed signal back to the time domain, resulting in the denoised speech output.

4.6.2 Experiment

The modified FCRN model is trained using 320 sentences from the CSTR database belonging to normal speakers. For testing, 80 sentences are selected from the test set. The data for the study is in the ratio of 80:20 for the training and testing. The training data for the network is around 20 hours. The speech signal with a sampling rate of 16K samples per second is processed with a window length of 480 samples. The network's parameters include a mini-batch size of 128, an initial learning rate of 0.001, a learn rate dropout factor of 0.9, a learn rate dropout period of 1, and the optimizer used during training is "Adam."

(i) 10 Epochs

The training phase utilizes a relatively high learning rate of 0.00038742 and the validation RMSE of 4.2454 is achieved for 10 epochs of training. This value indicates that the model is in the early stages of learning and still has significant room for improvement. The high RMSE suggests that the model is yet to effectively capture the patterns in the data.

(ii) 30 Epochs

Based on the analysis made for 10 epochs of modified training, the training was extended to 30 epochs with a learning rate significantly reduced to 4.710e-05. The validation RMSE dropped to 2.2592, indicating substantial improvement. The reduction in learning rate allowed the model to make more refined weight updates, leading to better

convergence. The extended training duration also gave the model more opportunities to learn from the data, significantly decreasing RMSE.

(iii) 50 Epochs

The learning rate is further reduced to 5.7264e-06 to enhance the speech, and the training continued for 50 epochs. The validation RMSE improved dramatically to 0.9695, which resulted in a good performance in the metrics. The very low learning rate in this phase enabled fine-tuning of the model parameters, leading to precise adjustments and better generalization. The extended training duration ensures that the model has ample opportunity to refine its understanding of the data, substantially reducing RMSE.

(iv) 100 Epochs

In order to enhance the speech in terms of quality and intelligibility, the learning rate was reduced to 2.9513e-08, and training was extended to 100 epochs and 100,000 iterations to ensure thorough learning and generalization of the model. The reduction in validation RMSE to 0.4105 resulted in excellent performance in the metrics corresponding to quality and intelligibility. This extensive training period allows the model to fine-tune its weights, capture complex temporal and spatial dependencies, and minimize the loss function effectively, resulting in high-quality denoised speech outputs.

To determine accuracy of the modified FCRN technique, the stopping criterion of the training phase is based on the RMSE which gives maximum accuracy when it is below 0.5. Accuracy is maximum for all the performance metrics when the number of epochs is 100 and remains constant as epochs increases.

(v) Inference

The trials in the training process for the modified FCRN technique is increased till 100 epochs. The speech denoised by the modified FCRN algorithm proved more efficient than the existing deep learning algorithms. This comprehensive evaluation demonstrates the effectiveness enhancing the denoising capabilities of the modified FCRN model, thus validating its potential for real-world applications. The comparison between the modified FCRN model and the original baseline revealed several key insights. While the modified model exhibited superior denoising performance in objective metrics, it also demonstrated

enhanced robustness to various types of noise and background interference. However, further analysis revealed areas where additional refinement is needed, particularly optimizing computational efficiency without sacrificing denoising accuracy. Overall, the modified FCRN represents a significant advancement in speech enhancement technology, with potential applications in various real-world scenarios.

The results obtained for the various metrics for quality and intelligibility for all the four algorithms are given in Tables 4.2 to 4.7. Since the speech enhancement results are the best among the four algorithms, the spectrograms obtained for the modified FCRN technique are given in Figures 4.14 to 4.16 and Appendix 4 (Figures 1 to 10).

Table 4.2 Performance of Deep Learning Algorithms for Various Noise Types at -10 dB Noise Level

Noise Type	SNR				segSNR				PESQ				STOI				SI-SDR				DNSMOS			
	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN
Washing Machine	24.53	28.59	29.35	39.25	3.71	4.95	6.42	8.21	1.67	1.91	2.13	3.27	0.47	0.51	0.54	0.79	3.15	4.45	4.95	8.58	2.2	2.63	2.91	4.04
Rainbow	24.75	28.96	30.98	39.16	3.41	4.55	5.85	7.55	1.62	1.79	1.97	2.98	0.42	0.47	0.49	0.75	3.39	4.67	5.25	9.03	2.31	2.76	2.78	4.089
Babble	24.45	27.85	30.06	40.16	3.45	4.89	6.33	8.1	1.56	1.93	2.08	3.89	0.42	0.48	0.55	0.88	3.5	4.78	5.36	9.78	2.29	2.74	2.89	4.005
Airport	26.24	27.91	29.68	38.46	4.35	5.69	6.9	9.11	1.68	1.9	2.14	3.98	0.5	0.51	0.59	0.88	3.64	4.92	5.5	8.46	2.21	2.71	2.95	3.74
Jet Plane	25.63	31.26	34.98	40.9	4.42	5.98	7.18	9.46	1.45	1.67	1.91	3.57	0.46	0.53	0.58	0.77	3.48	4.76	5.34	8.87	2.27	2.48	2.72	4
Street	24.71	25.36	30.57	39.45	3.82	4.92	6.16	9.01	1.52	1.79	1.95	3.45	0.39	0.45	0.49	0.81	3.68	4.85	5.37	8.26	2.3	2.6	2.76	4.05
Train Whistle	27.74	32.21	33.79	40.13	4.68	5.91	7.14	8.46	1.74	1.97	2.09	3.6	0.5	0.58	0.64	0.84	3.48	4.81	5.54	8.55	2.51	2.78	2.9	4.01
Restaurant	25.32	28.14	30.91	39.46	3.59	4.66	5.75	8.45	1.45	1.67	1.93	4.01	0.39	0.48	0.53	0.88	3.48	4.97	5.56	8.24	2.25	2.48	2.74	3.84
Car	22.54	27.58	28.92	38.07	3.13	4.29	6.21	8.23	1.35	1.63	2.02	3.68	0.32	0.41	0.52	0.83	3.14	4.48	5.2	8.43	2.03	2.53	2.75	3.99
Subway	21.47	29.07	29.43	39.86	3.46	4.55	6.43	8.97	1.38	1.67	2.07	3.81	0.36	0.43	0.54	0.85	3.21	4.54	5.36	8.54	2.05	2.59	2.83	4.04

Table 4.3 Performance of Deep Learning Algorithms for Various Noise Types at -5 dB Noise Level

Noise Type	SNR				segSNR				PESQ				STOI				SI-SDR				DNSMOS			
	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN
Washing Machine	25.21	29.74	30.11	39.46	5.31	6.55	8.13	9.21	1.75	1.98	2.2	3.58	0.54	0.58	0.58	0.81	4.34	4.76	6.36	12.87	2.39	2.74	3.05	4.12
Rainbow	25.16	29.74	31.76	39.45	5.45	6.23	7.48	7.99	1.74	1.89	2.03	3.18	0.5	0.52	0.55	0.8	4.7	4.98	6.14	11.25	2.46	2.61	2.84	4.1
Babble	25.86	28.61	31.29	40.26	6.05	7.39	9.04	9.57	1.67	2.06	2.24	3.9	0.49	0.52	0.63	0.89	4.81	5.09	6.23	10.98	2.42	2.87	3.05	4.105
Airport	27.8	29.02	30.09	39.48	7.08	8.49	9.75	9.93	1.81	2.09	2.47	4	0.58	0.61	0.67	0.89	4.95	5.23	6.03	10.57	2.38	2.9	3.28	3.89
Jet Plane	26.67	32.65	36.08	41.59	6.92	8.64	9.87	11	1.69	1.92	2.17	3.74	0.51	0.6	0.64	0.84	4.79	5.07	6.23	10.98	2.51	2.73	2.98	4.15
Street	25.56	26.69	31.54	39.89	6.18	7.48	8.78	9.26	1.78	1.91	2.24	3.59	0.45	0.51	0.58	0.84	5.29	5.21	6.4	10.65	2.56	2.72	3.05	4.18
Train Whistle	28.64	33.67	34.58	40.29	7.21	8.46	9.56	9.95	1.93	2.14	2.21	3.78	0.59	0.63	0.71	0.85	5.1	5.42	6.43	10.89	2.7	2.95	3.02	4.15
Restaurant	26.08	29.77	31.46	40.13	5.32	6.35	7.54	8.65	1.75	1.97	2.37	4.09	0.44	0.54	0.59	0.9	5.04	5.58	6.78	10.69	2.5	2.78	3.18	4.07
Car	23.42	28.34	30.71	38.28	5.17	6.97	8.92	9.72	1.47	1.8	2.31	3.74	0.38	0.48	0.59	0.85	4.28	5.64	6.07	10.77	2.19	2.65	2.89	4.05
Subway	22.32	30.46	31.29	40.02	5.96	6.79	9.23	10.51	1.64	1.79	2.14	3.99	0.45	0.49	0.62	0.88	4.76	5.76	6.19	11.06	2.24	2.69	3.16	4.15

Table 4.4 Performance of Deep Learning Algorithms for Various Noise Types at 0 dB Noise Level

Noise Type	SNR				segSNR				PESQ				STOI				SI-SDR				DNSMOS			
	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN
Washing Machine	26.97	30.58	31.9	39.79	6.34	7.58	9.16	10.51	1.85	2.2	2.5	3.79	0.57	0.6	0.6	0.92	6.12	7.26	8.23	13.87	2.62	2.96	3.28	4.29
Rainbow	26.99	30.57	32.89	39.59	6.14	7.14	8.62	8.26	1.8	1.99	2.21	3.55	0.57	0.6	0.58	0.9	6.43	7.48	8.43	12.59	2.59	2.87	3.02	4.108
Babble	26.91	29.26	32.48	40.35	7.4	8.91	10.42	11.46	1.78	2.14	2.49	4.01	0.51	0.55	0.69	0.9	6.59	7.64	8.59	11.57	2.6	2.95	3.3	4.22
Airport	28.08	30.25	32.2	40.01	8.43	9.66	10.82	11.9	1.99	2.22	2.74	4	0.63	0.65	0.73	0.92	6.75	7.8	8.75	11.99	2.68	3.03	3.55	3.99
Jet Plane	27.01	33.29	37.61	42	8.29	10.12	11.54	11.79	1.65	2.17	2.49	3.89	0.55	0.67	0.71	0.86	6.52	7.57	8.52	11.57	2.46	2.98	3.3	4.23
Street	26.93	27.93	32.33	40.55	7.53	8.74	9.95	10.36	1.94	2.12	2.5	3.77	0.49	0.56	0.61	0.89	7.06	7.68	8.76	11.98	2.72	2.93	3.31	4.2
Train Whistle	29.13	34.77	35.23	40.35	8.38	9.71	11.06	11.15	2.04	2.3	2.44	3.85	0.69	0.7	0.79	0.85	6.87	7.62	8.83	12	2.81	3.11	3.25	4.19
Restaurant	26.96	30.61	32.97	40.57	7.05	8.24	9.58	9.72	1.78	2.24	2.59	4.13	0.53	0.6	0.63	0.94	6.61	7.78	9.15	12.49	2.64	3.05	3.4	4.11
Car	24.17	28.97	31.36	38.61	5.86	8.58	10.3	10.92	1.53	1.96	2.57	3.97	0.42	0.54	0.67	0.88	5.23	7.51	8.43	11.88	2.35	2.86	3.12	4.12
Subway	23.28	31.1	32.14	40.12	7.33	8.95	10.82	11.3	1.83	2	2.48	4.11	0.55	0.56	0.67	0.89	5.68	7.63	8.55	12.4	2.38	2.93	3.43	4.29

Table 4.5 Performance of Deep Learning Algorithms for Various Noise Types at 5 dB Noise Level

Noise Type	SNR				segSNR				PESQ				STOI				SI-SDR				DNSMOS			
	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN
Washing Machine	28.46	33.43	36.15	40.01	7.93	9.17	10.95	12.13	2.06	2.49	2.68	3.88	0.6	0.68	0.72	0.91	9.32	10.66	11.21	14.97	2.93	3.18	3.42	4.35
Rainbow	28.32	32.56	35.28	40.09	8.09	9.58	10.37	10.12	2.1	2.39	2.51	3.9	0.65	0.71	0.76	0.91	9.58	10.87	11.5	13.46	2.85	3.41	3.32	4.205
Babble	27.95	32.45	34.37	41.01	8.96	10.45	12.05	12.88	1.9	2.34	2.88	4.09	0.58	0.61	0.65	0.92	9.74	10.78	11.66	12.79	2.89	3.15	3.69	4.238
Airport	30.72	31.58	34.24	40.59	10.14	11.79	13.07	13.59	2.2	2.54	2.94	4.16	0.67	0.75	0.78	0.94	9.92	10.96	11.84	12.59	2.82	3.35	3.75	4.08
Jet Plane	29.99	34.24	38.96	42.74	9.62	11.52	12.86	12.97	2.17	2.39	2.82	4.21	0.61	0.71	0.77	0.89	9.67	10.96	11.59	12.87	3.02	3.2	3.63	4.54
Street	28.26	30.6	34.81	40.99	9.06	10.37	11.68	12.84	2.15	2.49	2.8	4.1	0.59	0.64	0.74	0.9	9.87	11.09	12.33	12.56	2.93	3.3	3.61	4.4
Train Whistle	30.77	36.86	38.27	40.49	10.1	11.29	12.69	13.89	2.35	2.53	2.75	4.01	0.75	0.82	0.76	0.86	9.73	11.01	12.04	14.19	3.12	3.34	3.56	4.25
Restaurant	29.94	32.62	35.46	41	8.96	10.03	11.26	11.9	1.96	2.63	2.91	4.25	0.59	0.65	0.7	0.95	9.68	11.17	12.56	14.99	2.78	3.44	3.72	4.24
Car	25.79	32.18	33.4	38.83	7.81	10.32	11.93	12.66	1.83	2.19	2.74	4.09	0.52	0.59	0.64	0.9	8.38	10.92	11.5	13.27	2.43	3.23	3.26	4.24
Subway	25.02	32.05	34.37	40.27	8.66	10.83	11.53	12.88	2.12	2.37	2.62	4.23	0.61	0.6	0.73	0.92	.8.49	11.29	12.12	14.07	2.62	3.29	3.63	4.38

Table 4.6 Performance of Deep Learning Algorithms for Various Noise Types at 10 dB Noise Level

Noise Type	SNR				segSNR				PESQ				STOI				SI-SDR				DNSMOS			
	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN
Washing Machine	29.63	34.85	37.3	40.15	9.02	10.41	12.3	13.6	2.24	2.7	3.15	3.98	0.63	0.75	0.71	0.94	11.52	13.26	15.41	16	3.1	3.36	3.85	4.55
Rainbow	31.51	34.66	37.11	40.6	9.17	9.7	11.7	12.25	2.29	2.56	2.85	4.1	0.64	0.74	0.77	0.95	11.56	13.46	15.68	15.65	3.22	3.37	3.66	4.289
Babble	29.57	33	35.81	41.11	10.51	12.29	13.08	13.89	2.28	2.46	3.25	4.11	0.61	0.63	0.73	0.95	12.67	12.68	15.81	15.96	3.26	3.27	4.06	4.341
Airport	32.55	33.81	37.53	41.57	11.08	12.75	14.55	15.78	2.49	2.88	3.47	4.18	0.69	0.79	0.86	0.96	11.87	12.87	15.98	16.01	3.23	3.69	4.22	4.15
Jet Plane	30.05	35.13	39.58	43.12	10.53	12.51	13.92	13.99	2.32	2.68	3.36	4.23	0.61	0.77	0.86	0.94	11.65	13.55	15.77	16.56	3.14	3.49	4.17	4.78
Street	30.45	31.96	35.64	41.03	9.69	11.14	12.59	14.12	2.44	2.79	3.34	4.22	0.6	0.7	0.77	0.95	11.64	13.69	15.88	16.65	3.22	3.6	4.15	4.42
Train Whistle	32.61	37.01	40.56	40.85	11.08	12.49	14.05	15.08	2.54	2.79	3.08	4.25	0.75	0.78	0.83	0.92	11.44	13.55	15.97	16.25	3.31	3.6	3.89	4.34
Restaurant	30.14	34.11	37.34	41.1	10.51	11.51	12.85	14.16	2.57	3.05	3.45	4.28	0.59	0.72	0.78	0.97	11.39	13.71	15.68	16.55	3.39	3.86	4.15	4.3
Car	26.41	32.73	35.69	38.97	8.89	12.44	13.26	13.85	2.02	2.45	3.38	4.17	0.53	0.61	0.71	0.92	10.31	12.52	14.65	15.26	2.76	3.51	3.69	4.33
Subway	27.21	32.94	35.5	40.38	9.57	12.59	12.49	13.5	2.31	2.67	3.09	4.34	0.61	0.67	0.76	0.93	10.93	13.4	15.77	16.13	3.08	3.56	4.07	4.44

Table 4.7 Performance of Deep Learning Algorithms for Various Noise Types at 15 dB Noise Level

Noise Type	SNR				segSNR				PESQ				STOI				SI-SDR				DNSMOS			
	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN	DFNN	Deep CNN	Modified LSTM	Modified FCRN
Washing Machine	32.83	35.74	38.39	40.65	12.15	13.45	16.24	16.31	2.55	2.92	3.86	4.01	0.67	0.81	0.79	0.96	15.97	16.66	19.61	20.02	3.32	3.6	4.1	4.78
Rainbow	33.84	35.52	38.17	40.89	12.98	13.08	15.14	15.88	2.56	2.88	3.7	4.12	0.64	0.8	0.86	0.96	15.98	16.96	19.89	19.88	3.37	3.69	4.35	4.489
Babble	34.28	33.79	39.3	41.88	13.98	14.97	15.96	16.55	2.52	2.97	3.56	4.18	0.68	0.66	0.78	0.97	16.21	17.11	20.04	20.9	3.39	3.78	4.37	4.41
Airport	33.83	34.57	41.59	41.71	14.7	16.2	17.8	17.91	2.57	3.23	3.96	4.2	0.7	0.83	0.91	0.95	16.41	17.31	20.24	20.9	3.35	4.04	4.56	4.62
Jet Plane	31.01	36.11	40.76	43.25	14.09	16.06	17.57	17.99	2.54	2.72	3.94	4.31	0.62	0.77	0.89	0.96	16.07	17.05	19.98	20.01	3.38	3.53	4.53	4.8
Street	31.78	33.62	38.23	41.25	13.56	15.06	16.56	16.9	2.6	2.98	4.19	4.33	0.62	0.7	0.81	0.96	16.57	17.19	20.21	20.89	3.41	3.79	4.56	4.69
Train Whistle	33.89	37.92	42.28	42.49	14.84	16.25	17.85	18.13	2.63	2.98	3.77	4.36	0.72	0.89	0.9	0.92	16.37	17.17	20.18	20.88	3.45	3.79	4.58	4.61
Restaurant	30.95	35.67	38.38	41.26	13.82	14.71	16.06	16.79	2.79	3.34	4.03	4.3	0.6	0.76	0.81	0.97	16.32	17.33	19.96	20.56	3.61	4.15	4.55	4.59
Car	27.69	33.52	36.41	39.47	12.7	13.52	15.84	16.59	2.29	2.64	3.62	4.28	0.55	0.67	0.76	0.93	12.85	16.02	19.58	20.29	3.12	3.65	3.98	4.39
Subway	28.54	34.29	37.09	40.63	13.13	13.65	14.94	17.3	2.46	2.86	3.83	4.42	0.58	0.69	0.78	0.94	13.19	16.54	19.95	20.43	3.26	3.78	4.34	4.52

4.7 RESULTS AND DISCUSSION

The experimental setup used for the study includes the same hardware and software as the traditional algorithms. Tables 4.2 to 4.7 represent the comparative analysis of DNN algorithms at various noise types and levels. It is observed that the modified FCRN technique effectively improved speech quality and intelligibility based on the metrics SNR, segSNR, PESQ, STOI, SI-SDR, and DNSMOS.

Signal-to-Noise Ratio (SNR) is a critical performance metric used to evaluate the effectiveness of speech enhancement algorithms. It measures the robustness of these algorithms and their ability to handle various noise levels, with higher SNR values indicating better performance. Based on the analysis in Figure 4.8, the modified FCRN technique demonstrates superior performance across multiple noise types, achieving higher SNR values, particularly at higher noise levels. Notably, the modified FCRN technique effectively enhances speech quality for challenging environments such as babble, airport, train whistle, and jet plane noise. Additionally, it shows commendable noise removal capabilities for other types of noise, including washing machine noise, rainbow noise, restaurant noise, and street noise. In comparison, the modified LSTM technique outperforms the deep CNN and DFNN algorithms. This indicates that LSTM's architecture is more adept at enhancing speech by effectively managing diverse and complex noise conditions. The Deep CNN algorithm by L. Wang et al. (2021) achieves an SNR of 13.2 at -5 dB, 14.3 at 0dB, and 16.7 at 5dB, while the proposed algorithm greatly improves SNR to 33.67 at -5dB, 34.77 at 0dB and 36.86 at 5dB. These results illustrate that the proposed Deep CNN algorithm enhances signal quality across varying noise levels. The results suggest that the modified FCRN and modified LSTM are more robust and reliable for speech enhancement tasks, particularly in environments with significant background noise.

Segmental Signal-to-Noise Ratio (segSNR) is a crucial metric for evaluating the effectiveness of noise reduction algorithms by quantifying the similarity between clean and enhanced speech at a frame level. The analysis depicted in Figure 4.9 demonstrates that the modified FCRN algorithm exhibits significant noise reduction for various noise types at a 15dB noise level. Notably, the modified FCRN algorithm performs remarkably

well in reducing noise for challenging types, such as train whistles and airport and jet plane noises. This higher segSNR value signifies a higher similarity between the clean and enhanced speech signals, indicating superior noise removal capabilities. Moderate noise removal is also observed with the modified FCRN technique for babble, street, washing machine, restaurant, and rainbow noise at the same noise level, showcasing its versatility across different noise environments. Comparatively, the modified LSTM algorithm demonstrates exceptional noise reduction, particularly for train whistle noise, highlighting its effectiveness in specific noise conditions. The Deep CNN algorithm shows efficient noise removal for train whistle, airport, and jet plane noise while providing moderate noise reduction for other noise types. Meanwhile, the DFNN algorithm, although effective, ranks third in noise reduction performance compared to the other deep learning algorithms. Overall, the analysis of segSNR provides a clear indication of the noise removal capabilities of each algorithm, with modified FCRN and modified LSTM leading in performance, followed by Deep CNN and DFNN. This underscores the robustness of modified FCRN and modified LSTM in achieving higher similarity between clean and enhanced speech signals, making them preferable choices for effective noise reduction.

In Figure 4.10, the PESQ (Perceptual Evaluation of Speech Quality) values for the speech enhanced using the modified FCRN architecture stand out among the four DNN algorithms, indicating the highest quality. PESQ, an index ranging from 1 to 5, measures speech quality, with higher scores signifying superior quality. The baseline FCRN algorithm by Strake et al., (2020) achieves a PESQ of 2.79 at -5 dB, whereas the results of the modified FCRN technique proposed achieves the PESQ of 4.09 at -5dB which is significantly higher than the original baseline FCRN by Strake et al . The modified FCRN technique consistently achieves the highest PESQ values, particularly for train whistle, street, jet plane, and restaurant noise, rated as 'Very Good.' Additionally, the modified FCRN technique effectively handles other challenging noise types, such as babble and airport noise, maintaining high PESQ scores indicative of good speech quality. Even for noises like rainbow and washing machine noise, the modified FCRN technique achieves PESQ values above 4, rated as 'Good,' demonstrating its robustness in various noisy environments. The analysis of PESQ values highlights the superior ability of the modified FCRN technique to enhance speech quality across different noise conditions. The modified

LSTM technique is rated as the second-best for speech enhancement, showcasing its effectiveness next to modified FCRN technique. The baseline LSTM algorithm by Tang et al. (2019) achieves PESQ of 1.34 at -5 dB, 1.70 at 0dB, 1.99 at 5dB, 2.17 at 10dB and 2.27 at 15dB whereas the results of the modified LSTM technique proposed significantly improves PESQ to 2.47 at -5dB, 2.74 at 0dB, 2.94 at 5dB, 3.47 at 10dB, 4.19 at 15dB. These results illustrate the superior ability of the proposed LSTM algorithm to enhance signal quality across varying noise levels. Meanwhile, the Deep CNN and DFNN algorithms follow, with their performance also contributing to notable improvements in speech quality, although to a lesser extent than modified FCRN and modified LSTM. This evaluation emphasizes the efficacy of modified FCRN and modified LSTM in providing high-quality speech enhancement in the presence of additive noise.

The Short-Time Objective Intelligibility (STOI) metric is a widely used and validated measure for assessing the intelligibility of degraded speech. It ranges from 0 to 1, with higher values indicating better intelligibility. Figure 4.11 illustrates the STOI scores obtained for various noise types and algorithms, highlighting the effectiveness of the different speech enhancement systems. The baseline FCRN algorithm by Strake et al., (2020) achieves STOI score of 0.89, whereas the results of the modified FCRN technique proposed achieves a score of 0.90 at a -5dB noise level. The modified FCRN technique demonstrates marginal yet consistent improvement in speech intelligibility across various noise types. It achieves the highest STOI scores for challenging environments such as babble and restaurant noise, showcasing its robust performance. For other noise types like street noise, jet plane noise, rainbow noise, and washing machine noise, the modified FCRN technique maintains high intelligibility scores, further affirming its reliability. The baseline LSTM algorithm by Tang et al. (2019) achieves STOI scores of 0.54 at -5 dB, 0.66 at 0dB, 0.74 at 5dB, 0.78 at 10dB and 0.81 at 15dB whereas the results of modified LSTM technique greatly improves STOI to 0.71 at -5dB, 0.79 at 0dB, 0.78 at 5dB, 0.86 at 10dB and 0.91 at 15dB. These results illustrate the superior ability of the modified LSTM technique to enhance signal quality across varying noise levels. The modified LSTM technique performs exceptionally well, particularly for airport and train whistle noise at higher SNR levels. It effectively enhances the intelligibility of speech in the presence of jet plane and rainbow noise, demonstrating its capability to manage diverse noisy

conditions. The modified LSTM's performance remains strong at slightly lower SNR levels, maintaining good intelligibility scores across various noise types. While not leading, Deep CNN shows commendable performance in enhancing speech intelligibility, particularly in the presence of washing machine noise, rainbow noise, airport noise, and train whistle noise. It provides marginally good intelligibility scores, indicating its potential in various noisy environments. The DFNN algorithm, although rated lower than modified FCRN, modified LSTM, and Deep CNN, still contributes to improved speech intelligibility, with scores indicating moderate performance. Overall, the analysis of STOI scores highlights the modified FCRN and modified LSTM techniques as leading solutions for enhancing speech intelligibility in noisy environments, with Deep CNN also providing notable improvements. This evaluation underscores the effectiveness of these deep learning algorithms in developing robust speech enhancement systems.

The Scale-Invariant Signal-to-Distortion Ratio (SI-SDR) provides valuable insights into the performance of noise reduction algorithms, with more positive values indicating better quality of enhanced speech. In Figure 4.12, the modified FCRN technique outperforms the other algorithms, particularly excelling in challenging noise environments such as babble, airport, street, and train whistle. This superior performance demonstrates the modified FCRN's effectiveness in significantly improving speech quality across various noise types. Following the modified FCRN, the modified LSTM technique also exhibits strong performance in noise reduction, especially for airport noise, street noise, train whistle noise, and babble noise at a 15dB noise level. The modified LSTM technique maintains moderate performance for other noises, including jet plane, restaurant, rainbow, and washing machine noise, indicating its versatility and reliability in diverse noise conditions. The Deep CNN algorithm ranks third in SI-SDR performance, showing effectiveness in noise removal and enhancing speech quality, although to a slightly lesser extent than modified FCRN and modified LSTM techniques. The DFNN algorithm, while still contributing to noise reduction, is rated fourth in comparison, demonstrating moderate improvements in speech quality. Overall, the analysis of SI-SDR values underscores the modified FCRN technique as the leading solution for noise reduction, followed closely by the modified LSTM technique. Both architectures exhibit significant enhancements in speech quality across various noise environments, with Deep CNN and DFNN also

providing notable contributions to noise reduction. This evaluation highlights the robustness and effectiveness of these deep learning algorithms in developing advanced speech enhancement systems.

The Deep Noise Suppression Mean Opinion Score (DNSMOS) by Abdulatif et al. (2024) is an objective measure used to evaluate the perceived improvement in speech quality achieved by deep noise suppression algorithms. DNSMOS scores range from 1 to 5, with higher values indicating better speech intelligibility and quality. A score of 5 denotes perfect quality, while a score of 1 indicates poor quality Li et al., (2022). In the analysis shown in Figure 4.13, the modified FCRN architecture consistently achieves the highest DNSMOS values, particularly excelling in challenging conditions such as jet plane noise and washing machine noise. This indicates that the modified FCRN is highly effective in enhancing perceived speech quality across various noise environments. The modified LSTM technique also demonstrates strong performance, particularly for train whistle noise, airport noise, and street noise. The algorithm effectively enhances speech intelligibility across other noises, with DNSMOS values indicating good to excellent speech quality. The Deep CNN algorithm performs well, especially in the presence of restaurant noise and airport noise. However, its DNSMOS values suggest moderate speech quality improvement compared to the modified FCRN and modified LSTM techniques. The Deep CNN algorithm performs moderately for other noise types, indicating its capability to handle noise suppression, albeit with less consistency. The DFNN algorithm, while effective, is rated fourth in terms of DNSMOS values, reflecting a moderate improvement in speech quality. This positions DFNN as a less effective option than the other algorithms, though it still contributes to the overall noise suppression efforts. Overall, the DNSMOS analysis highlights the modified FCRN and modified LSTM technique as the leading algorithms in enhancing perceived speech quality, with Deep CNN and DFNN providing fair to moderate improvements. This evaluation underscores the effectiveness of modified FCRN and modified LSTM technique in delivering high-quality speech intelligibility across various noise levels.

The deep learning algorithms are tested using unseen noises, such as cars and subways. For the unseen noises, values of performance metrics in DFNN are not equivalent to those of the other seen noises. The performance metrics show moderate improvement in Deep CNN and modified LSTM compared to the noises seen. From the results of modified FCRN, it is evident that for the unseen noises, modified FCRN shows the performance equivalent to the seen noises.

The modified FCRN architecture performed well in reducing noise and improving quality and intelligibility. Based on the values of DNSMOS, it is evident that the model improved the subjective quality of enhanced speech signals. Based on the results, it is inferred that the modified FCRN network and modified LSTM technique can effectively solve speech enhancement tasks like a mobile device, a hearing aid, or a speech prosthesis unit. Both modified FCRN and modified LSTM can be used effectively for speech enhancement. The performance of Deep CNN and DFNN is also effective and acceptable.

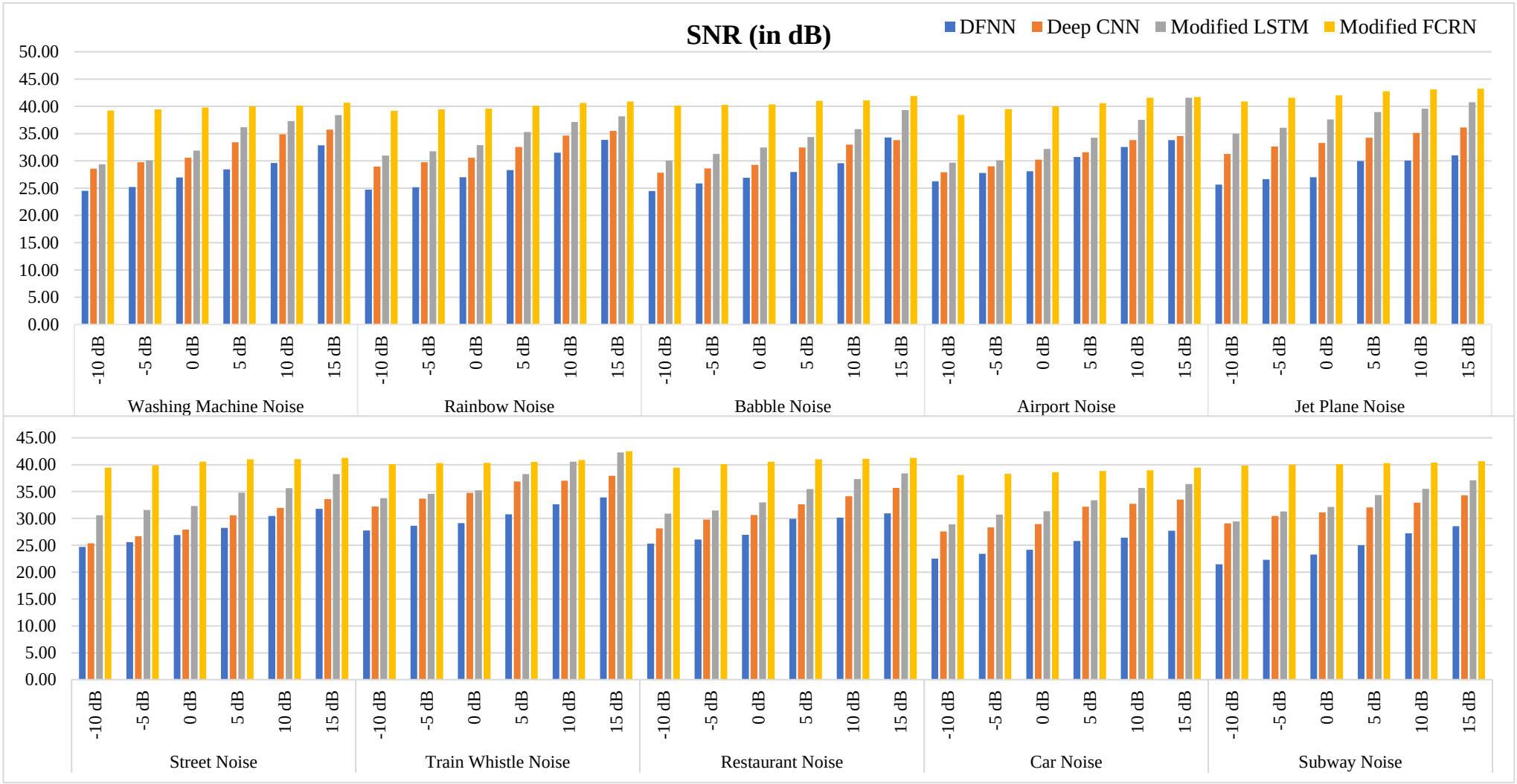


Figure 4.8 Comparative Analysis of SNR of DNN Algorithms at Various Noise Types and Levels

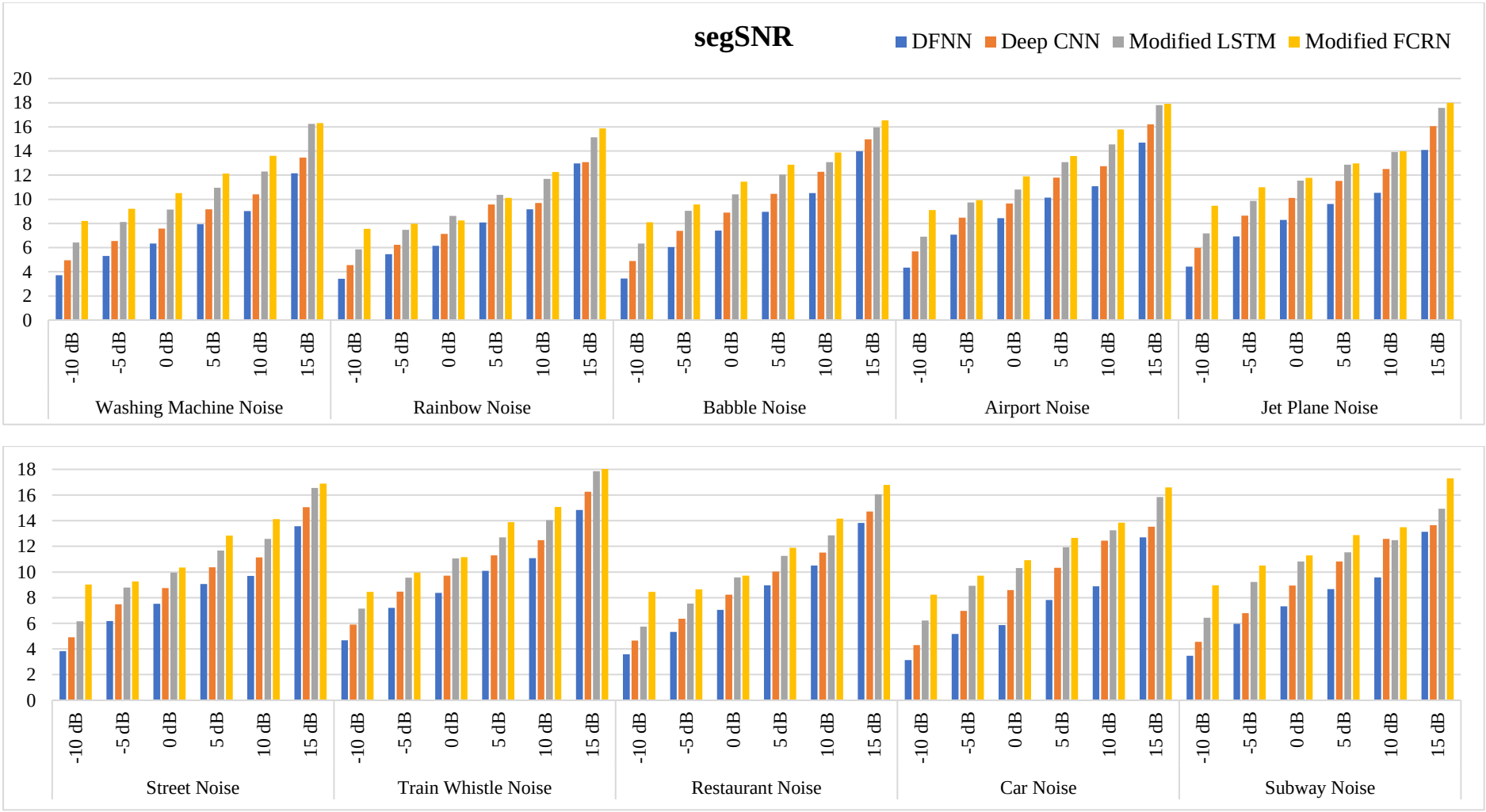


Figure 4.9 Comparative Analysis of segSNR of DNN Algorithms at Various Noise Types and Levels

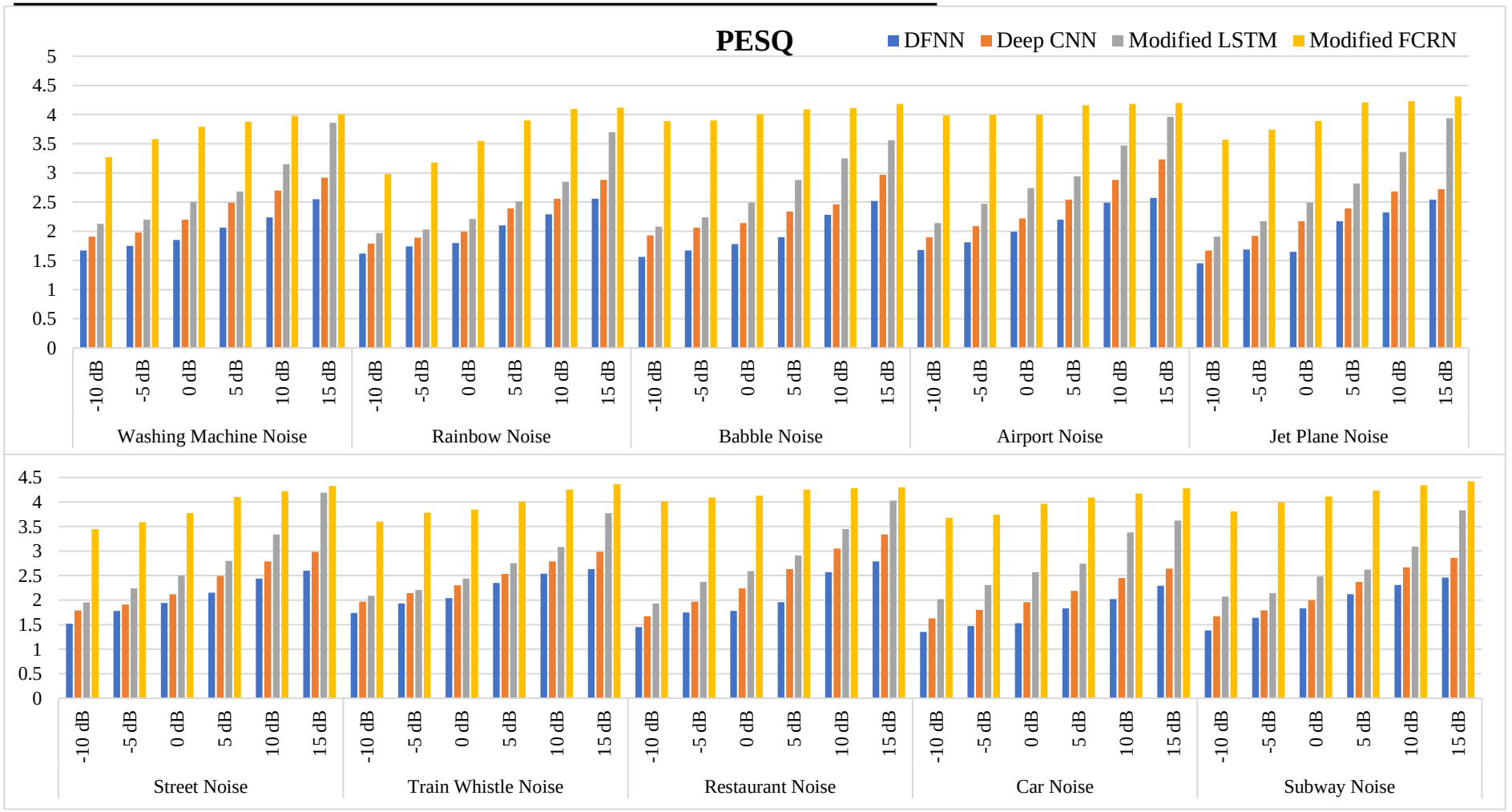


Figure 4.10 Comparative Analysis of PESQ of DNN Algorithms at Various Noise Types and Levels

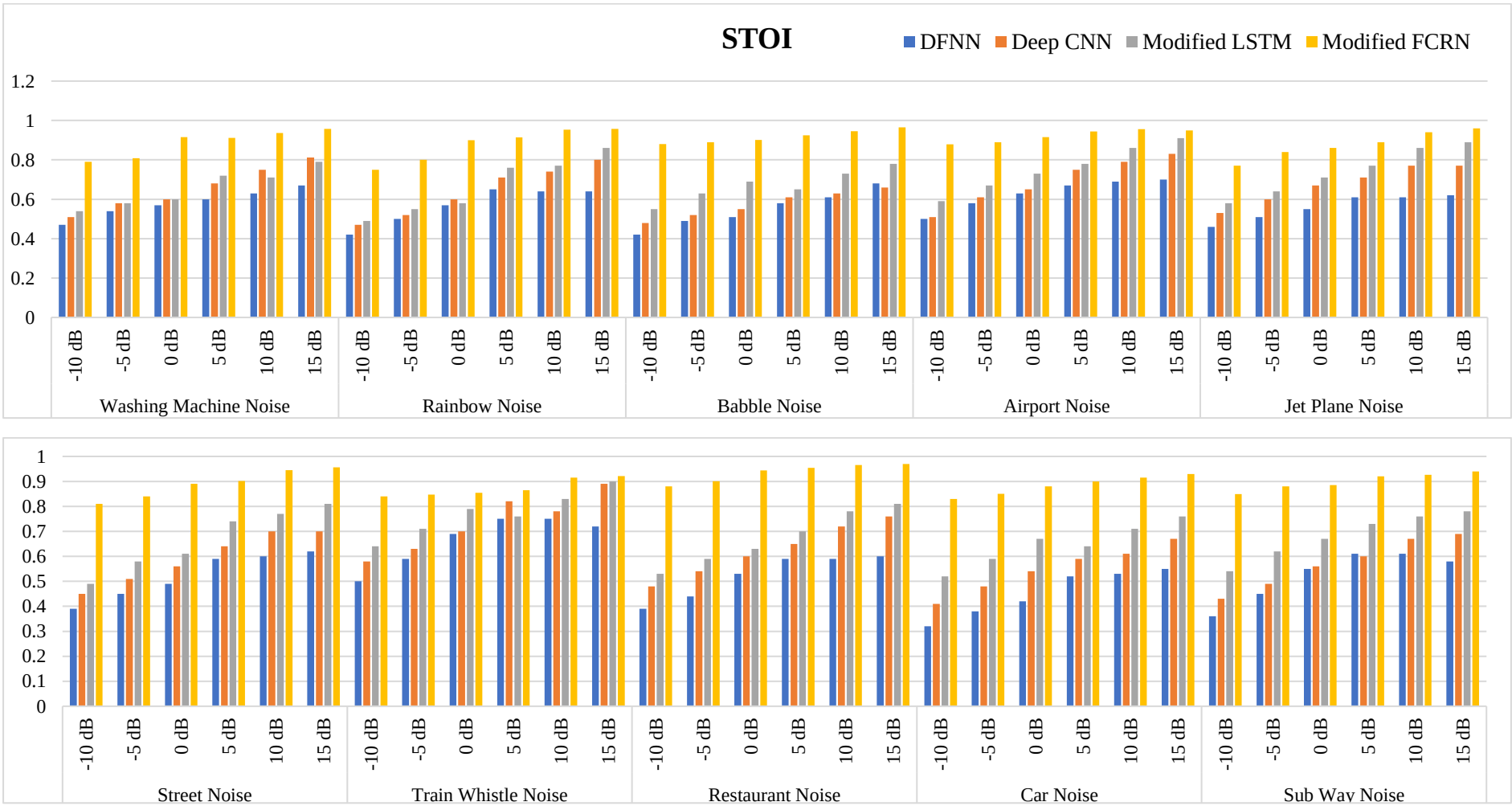


Figure 4.11 Comparative Analysis of STOI of DNN Algorithms at Various Noise Types and Levels

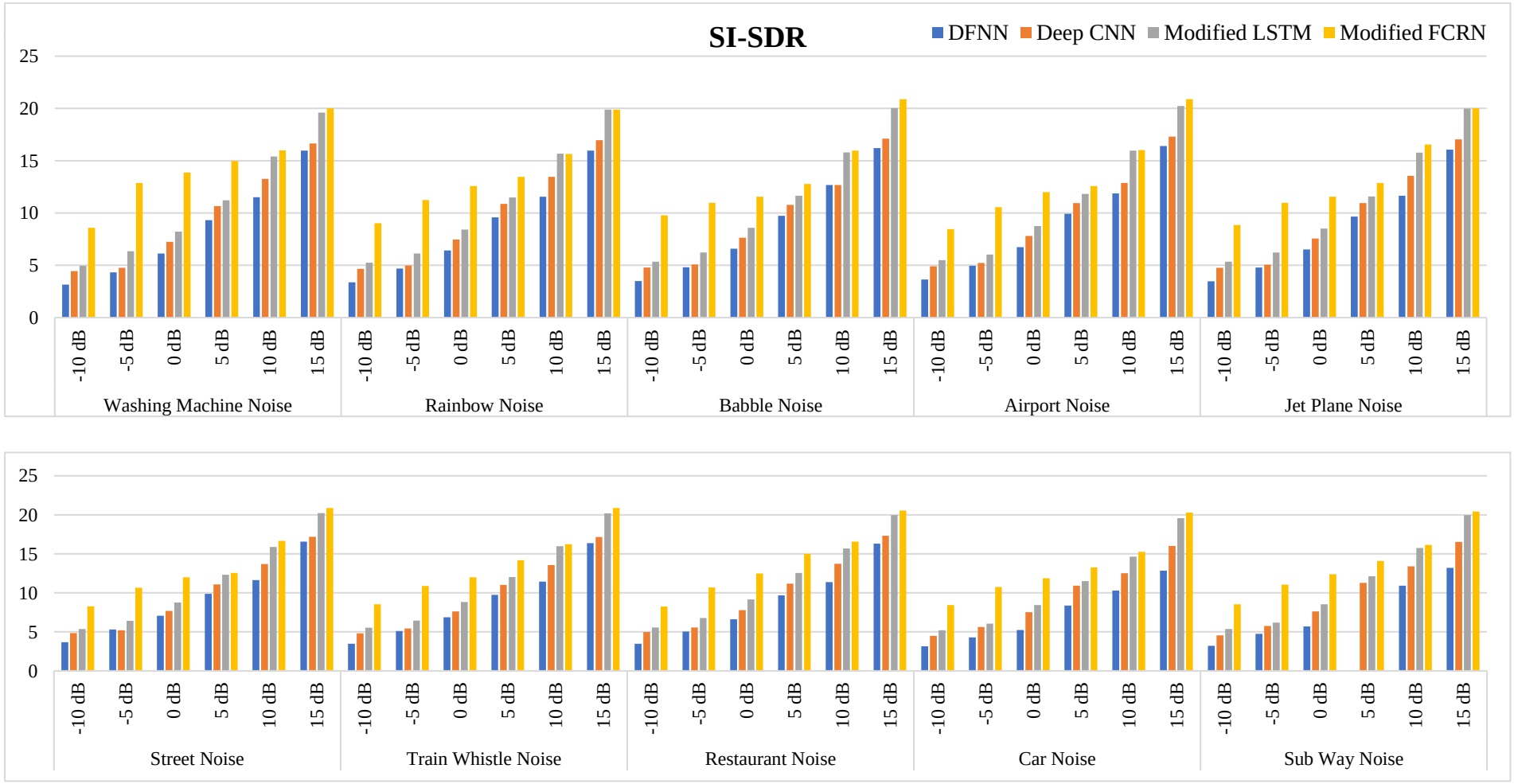


Figure 4.12 Comparative Analysis of SI-SDR of DNN Algorithms at Various Noise Types and Levels

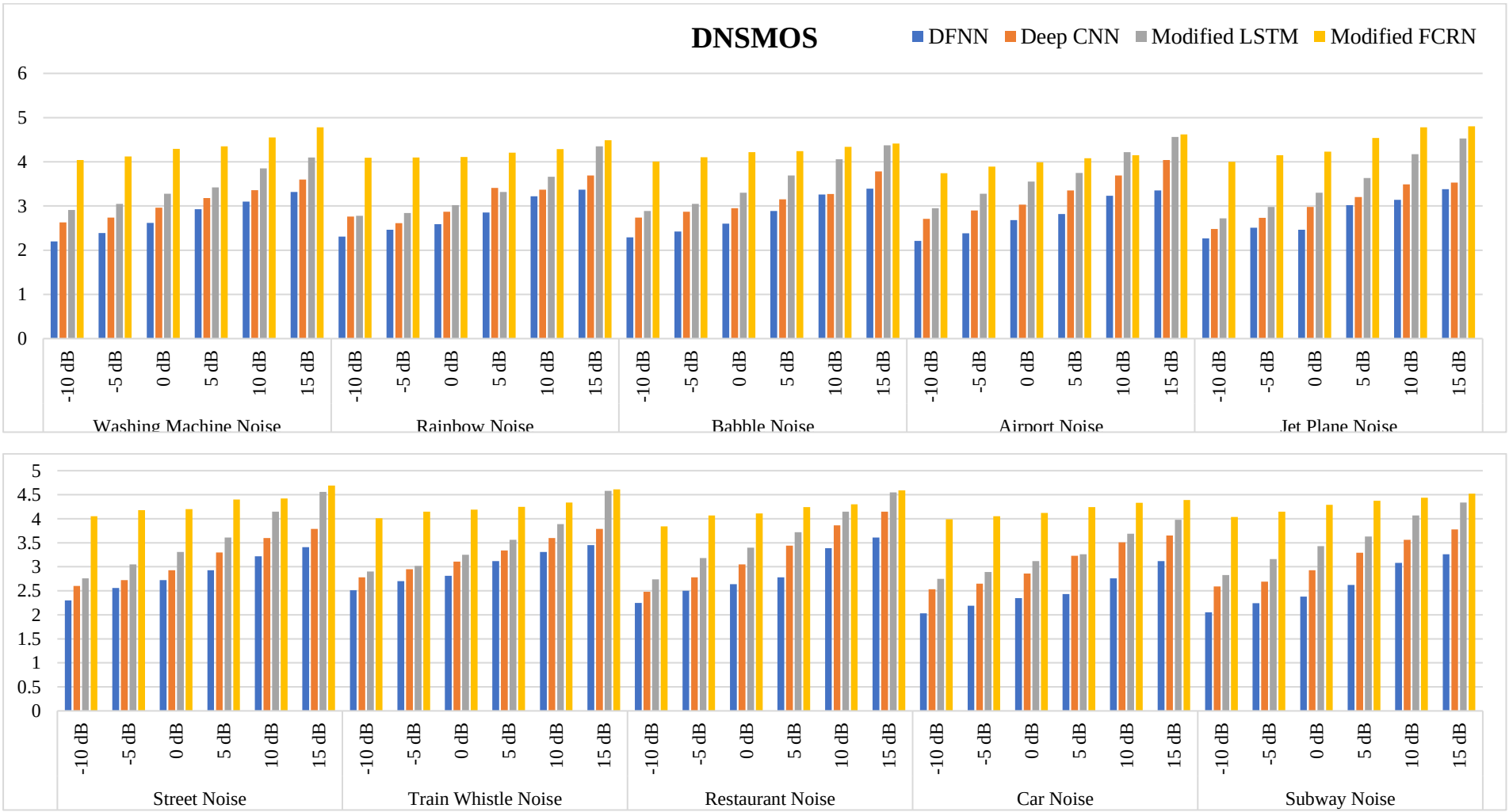
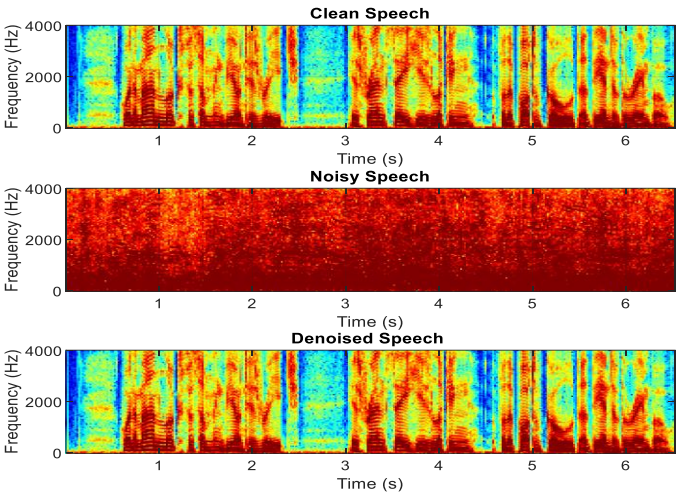


Figure 4.13 Comparative Analysis of DNSMOS of DNN Algorithms at Various Noise Types and Levels

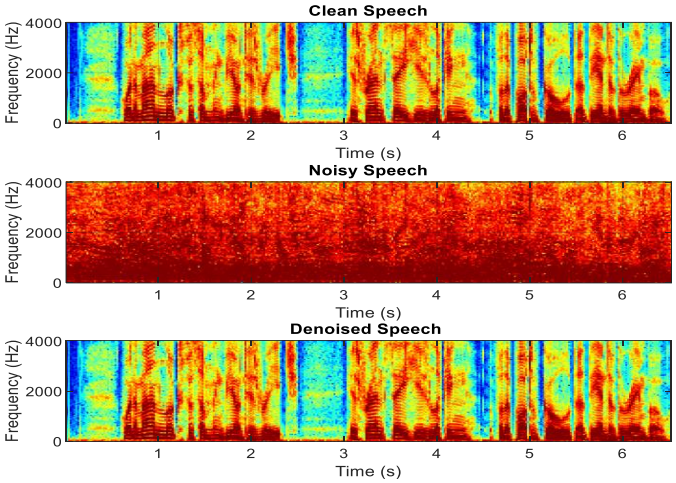
A spectrogram is a visual representation of the spectrum of frequencies of the speech signal as it varies with time. The y-axis indicates frequencies of 4000 Hz, and the x-axis shows the time of 6 seconds. The frequency range between 1 kHz and 4 kHz is vital for intelligibility. The representation of the spectrograms is plotted with time in the x-axis and frequency in the y-axis from Figures 4.14 to 4.16. The spectrograms of the clean speech signal, noisy speech signal, and denoised signals for different kinds of background noises added to clean speech with different SNRs help illustrate the implemented algorithms' behavior and performance. The effectiveness of the speech enhancement algorithm in enhancing speech is evident when comparing the clean speech spectrogram with the denoised spectrogram. The spectrogram consists of colours such as blue, cyan, yellow and red. The quieter speech and pauses have lesser intensity and are represented in blue; when the intensity of speech increases, the colour transitions from blue to cyan. As the intensity of speech increases, the color shifts to yellow. Yellow represents higher intensity and corresponds to more intense frequencies in speech. Red signifies the strongest and most intense speech frequencies in the spectrogram.

Figure 4.14 represents the spectrogram image of babble noise at various noise levels for modified FCRN. The babble noise masks the speech, making it less distinct. The chaotic and cluttered pattern is represented by a mixture of blue, cyan, yellow, and red colors, indicating various levels of noise interference. The overall energy distribution in the enhanced speech spectrogram shows a higher SNR, with speech components being more prominent relative to the noise. This indicates that the speech has been made more intelligible. Figure 4.15 represents the spectrogram image of train whistle noise for modified FCRN. Train whistle typically produce sound at relatively high and narrow frequency ranges, often concentrated in a few distinct frequency bands. Train whistle noises are often characterized by a steady, prolonged tone or a sequence of tones that can change over time with rising and falling pitch. In a spectrogram where train whistle noise is combined with speech, the whistle's distinct tones appear as prominent horizontal bands across the time axis, potentially overlapping and masking the speech components. Speech formants and harmonics are less visible, especially when the loud train whistle noise occupies similar frequency ranges. In the enhanced speech spectrogram, these frequency bands are diminished, revealing the speech features and leading to a more precise and

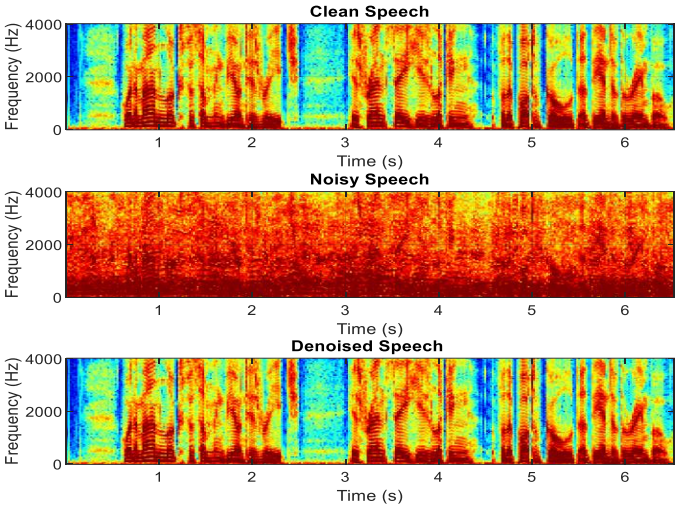
distinct representation of the speech in the spectrogram. Figure 4.16 represents the spectrogram image of subway noise at various noise levels for modified FCRN. Subway noise typically includes a mix of low-frequency rumbles, mid-frequency mechanical sounds, and occasional high-frequency screeches or hisses. The low-frequency components, often due to the train's movement and machinery, dominate the spectrogram, appearing as broad, continuous bands at the lower end of the frequency spectrum. Mid to high-frequency components may appear as sporadic lines or bursts, corresponding to specific mechanical noises or high-pitched sounds. The spectrogram of subway noise often shows continuous low-frequency energy, which can be sustained over long periods. In the enhanced speech, the continuous low-frequency band characteristic of subway noise is significantly reduced or removed. Sudden, sporadic bursts of noise are attenuated, resulting in a cleaner spectrogram. This helps preserve speech components' integrity, which appear more stable and less disrupted.



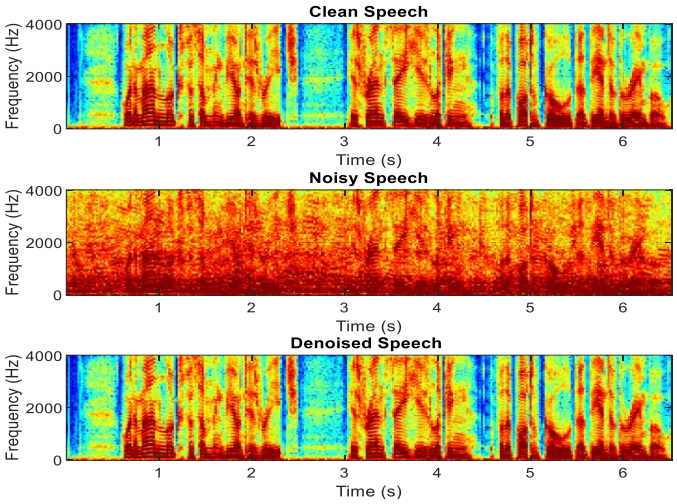
a) -10 dB



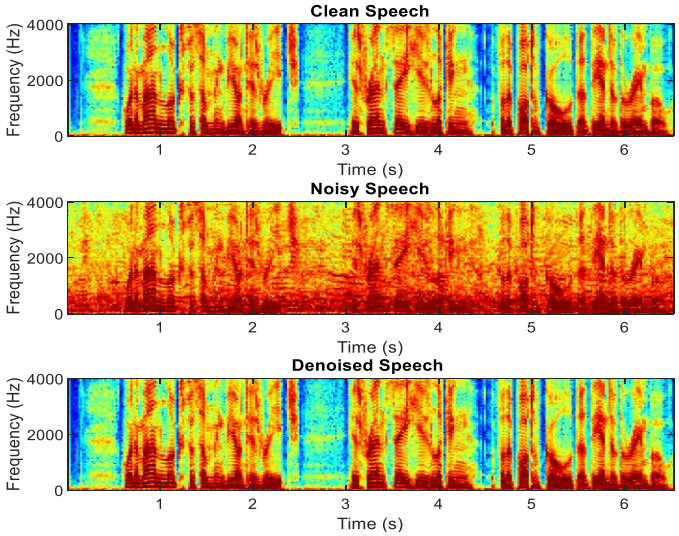
b) -5 dB



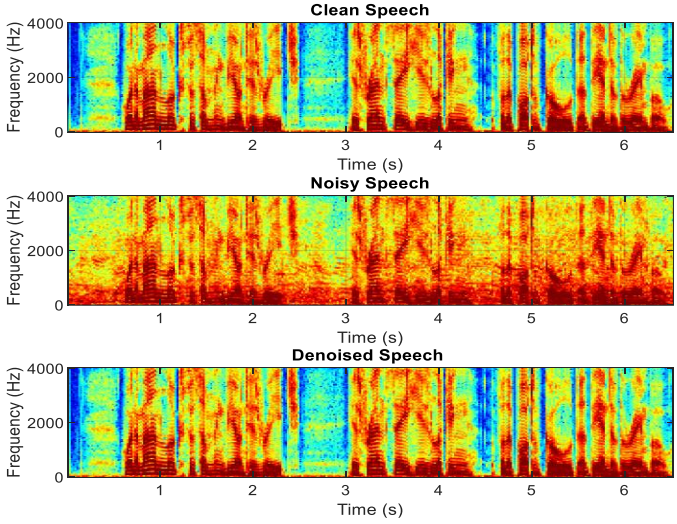
c) 0 dB



d) 5 dB



e) 10 dB



f) 15 dB

Figure 4.14 Modified FCRN - Spectrogram Images of Babble Noise for various Noise Levels

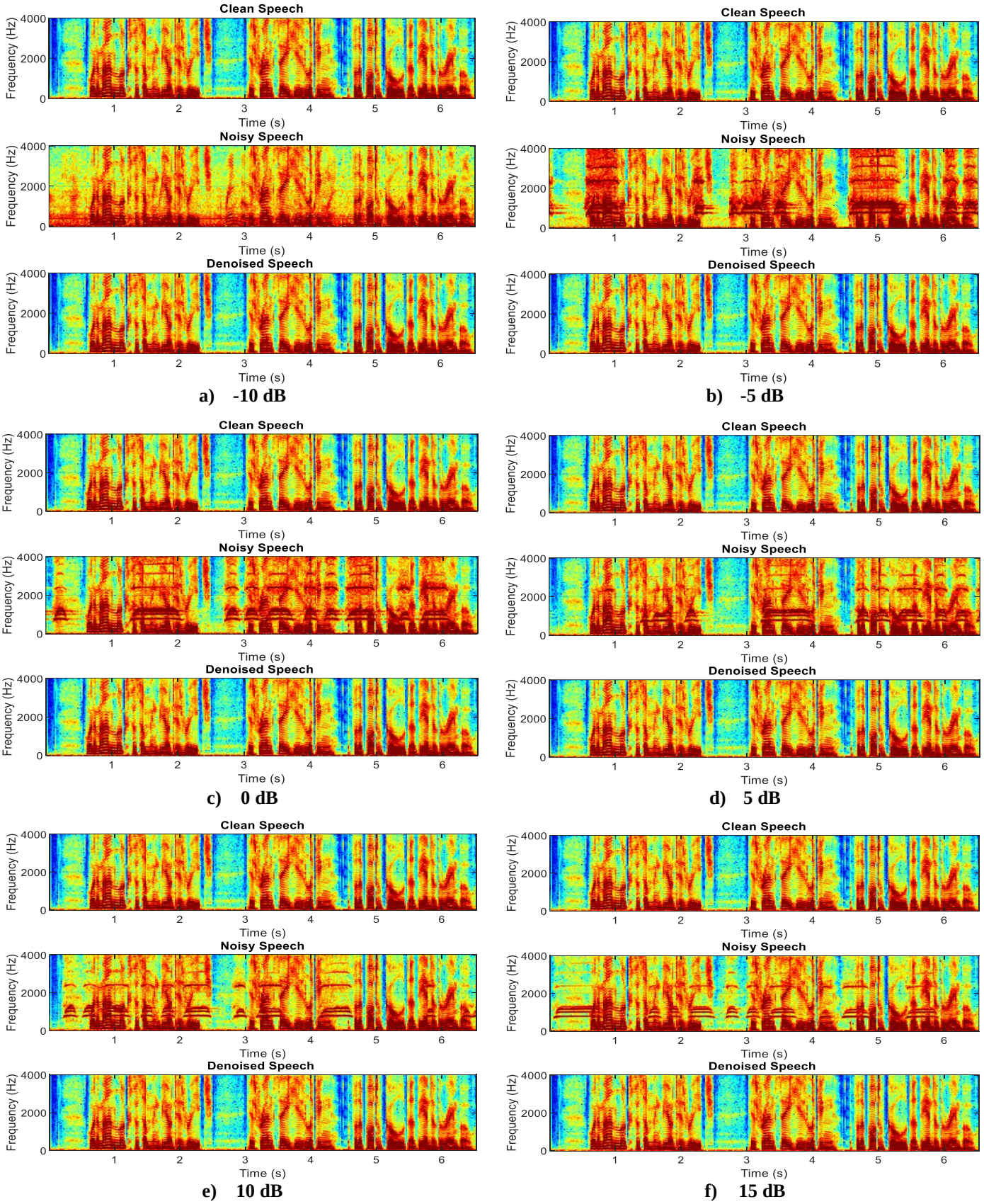
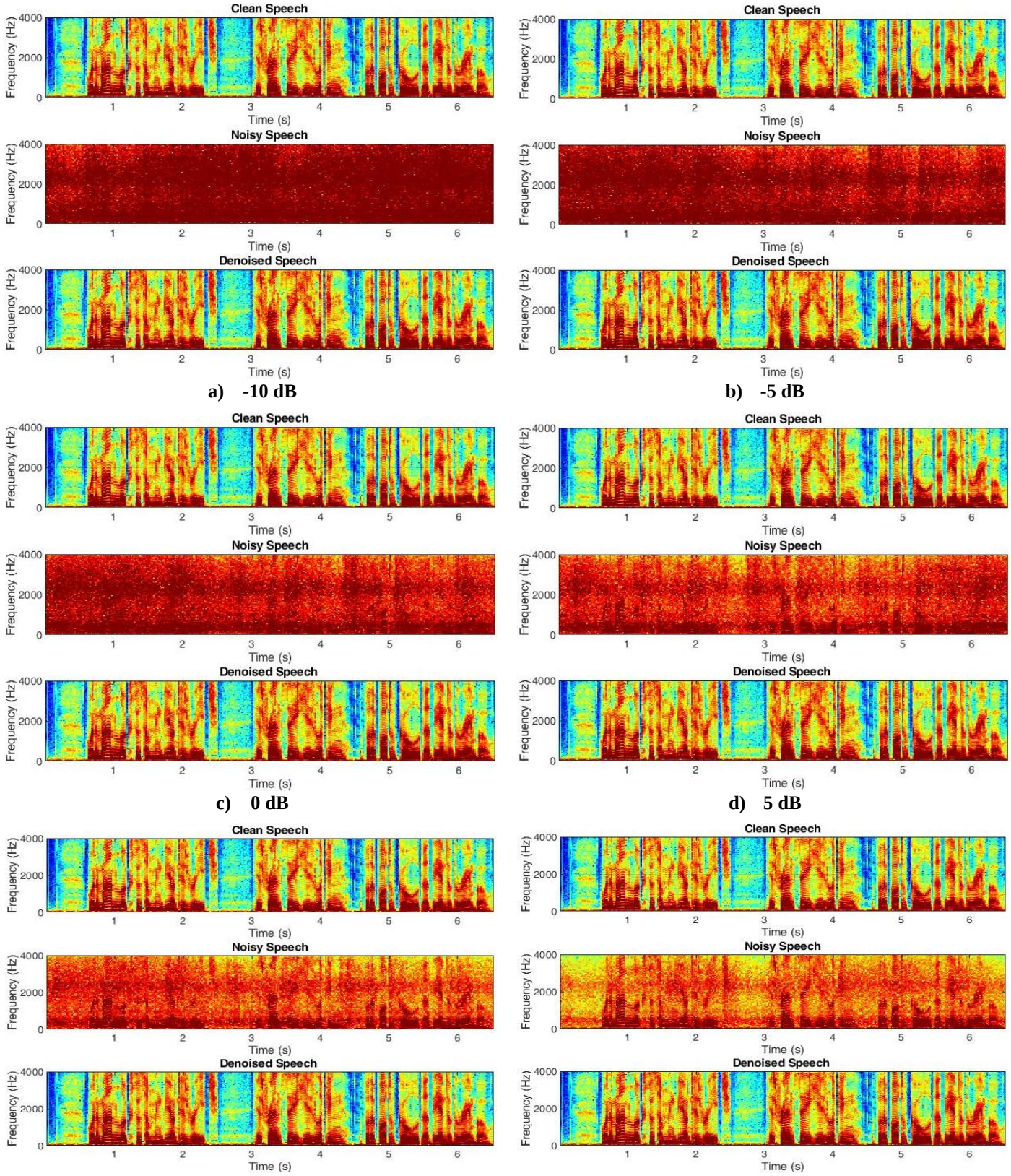


Figure 4.15 Modified FCRN - Spectrogram Images of Train Whistle Noise for various Noise Levels



a) -10 dB **b) -5 dB**

c) 0 dB **d) 5 dB**

e) 10 dB **f) 15 dB**

Figure 4.16 Modified FCRN - Spectrogram Images of Subway Noise for various Noise Levels

4.8 TRAINING METRICS OF DEEP NEURAL NETWORKS

Training metrics used in the network design are epochs, RMSE, and learning rate, as shown in Table 4.8. These metrics are pivotal in evaluating and optimizing the performance of a neural network. Epochs represent the number of times the model has seen the entire dataset, impacting its learning process. Root Mean Square Error (RMSE) measures the accuracy of predictions by quantifying the average error between predicted and actual values. The learning rate controls the step size during weight updates, influencing the speed and stability of the training process. These metrics provide crucial insights into model performance, convergence, and training efficiency.

Table 4.8 Training Metrics of the Networks

Training Metrics	DFNN	Deep CNN	Modified LSTM	Modified FCRN
Epochs (Early Stopping)	62	55	67	100
Learning rate	1.62 e-06	3.38 e-06	0.55e-07	2.95 e-08
RMSE	0.4761	0.4226	0.4651	0.4105

4.9 SUMMARY

The conventional algorithms are performing moderately in terms of quality and intelligibility of the denoised speech signal. Implementing deep learning algorithms has resulted in excellent performance in terms of quality and intelligibility compared to the conventional speech enhancement algorithms discussed in Chapter 3.

Chapter 5 deals with the second part of the research work which discusses the changes in voice production process for patients affected by laryngeal cancer and speech production after laryngectomy.