electronics

MDPI

Article

Sparse Signal Recovery through Long Short-Term Memory Networks for Compressive Sensing-Based Speech Enhancement

Vasundhara Shukla * and Preety D. Swami

Department of Electronics and Communication Engineering, University Institute of Technology RGPV, Bhopal 462033, India; preetydswami@gmail.com
* Correspondence: shuklav2015@gmail.com

Abstract: This paper presents a novel speech enhancement approach based on compressive sensing (CS) which uses long short-term memory (LSTM) networks for the simultaneous recovery and enhancement of the compressed speech signals. The advantage of this algorithm is that it does not require an iterative process to recover the compressed signals, which makes the recovery process fast and straight forward. Furthermore, the proposed approach does not require prior knowledge of signal and noise statistical properties for sensing matrix optimization because the used LSTM can directly extract and learn the required information from the training data. The proposed technique is evaluated against white, babble, and f-16 noises. To validate the effectiveness of the proposed approach, perceptual evaluation of speech quality (PESQ), short-time objective intelligibility (STOI), and signal-to-distortion ratio (SDR) were compared to other variants of OMP-based CS algorithms The experimental outcomes show that the proposed approach achieves the maximum improvements of 50.06%, 43.65%, and 374.16% for PESQ, STOI, and SDR respectively, over the different variants of OMP-based CS algorithms.

Keywords: speech enhancement; compressive sensing; sparse recovery; LSTM; deep learning

Citation: Shukla, V.; Swami, P.D. Sparse Signal Recovery through Long Short-Term Memory Networks for Compressive Sensing-Based Speech Enhancement. *Electronics* **2023**, *12*, 3097. https://doi.org/10.3390/electronics12143097

Academic Editor: Stefanos Kollias

Received: 9 May 2023
Revised: 22 June 2023
Accepted: 4 July 2023
Published: 17 July 2023

1. Introduction

In recent years, compressive sensing (CS) has garnered a significant amount of interest in the fields of signal processing, image processing, and information theory [1]. It proposes that a signal can be reconstructed with a lower number of samples (observations) when compared to the number of samples that are required by traditional Nyquist-based methods. This is in contrast to the fact that more observations are needed to reconstruct a signal using Nyquist-based methods [2].

In order for CS to function, the input signal needs to be capable of a high degree of compression, or to be more specific, it needs to be sparse. When compared to its entire length, a signal is said to have sparse components when it has a low number of active, nonzero ones. This quality can be seen in the signals either in the domain of their sample or in any other underlying transform domain, such as the Fourier or wavelet domains [3].

The CS uses an underdetermined system of linear equations to sample the signal directly into compressed form [4]. The compressed signal can be restored to its original form by using proper recovery algorithms. While recovering the signal, the denoising or enhancement of the signal can be achieved by removing certain components of the compressed signal during the recovery process.

As the compression is obtained through an underdetermined system of linear equations, the recovery or reconstruction of an uncompressed signal is a complex task. Many ideas have been proposed for efficient reconstruction of the compressed signal [5]. However, most of them, like orthogonal matching pursuit (OMP) [6], matching pursuit based on least squares (MPLS) [7], etc., are greedy, and use the iterative process for the efficient reconstruction of compressed signals. Alternatively, the greedy algorithms recover the

signal iteratively, making a local optimal choice at each iteration with the possibility of finding the global optimal solution at the end of the algorithm.

In recent years, deep learning has seen widespread use in many key and diverse application domains, including natural language processing, computer vision, and system identification, to name just a few of these areas [8]. In the field of system modeling, identification, and response estimates, it has also become one of the most active and current fields of research [9]. The process of developing mathematical models of dynamical systems that are based on observations of input and output signals is referred to as system identification. Deep learning models can be categorized into non-sequential and sequential models. In non-sequential models such as deep feedforward neural networks (DFNNs) [10], convolutional neural networks (CNNs) [11], etc., the network's output depends only on the present input and has no influence from previous inputs. Therefore, these kinds of networks are useful for the modeling of processes where the temporal order of data does not matter, such as image classification, object detection, etc. Whereas in sequential models such as recurrent neural networks (RNN) [12] and long short-term memory (LSTM) [13], etc., the output of the network depends on the previous inputs. Therefore, these kinds of networks are useful for the modeling of processes where the temporal order of data matters, such as language modeling, speech recognition, etc. While both RNN and LSTM are designed to handle sequential data, LSTM was introduced to address the vanishing gradient problem and enable RNN. Although LSTMs are powerful, they come with increased computational complexity compared to standard RNNs.

For sequential data processing, the long-short-term memory network (LSTM) is one of the recurrent neural network designs that is utilized the most frequently in deep learning. Considering its performance, many variants of the LSTM have also been developed, such as the advanced-LSTM (A-LSTM) [14] and the switchable LSTM [15]. In A-LSTM, the current state is influenced by various states from different time steps, which removes the limitations of traditional LSTM and enhances its ability to model time dependencies more effectively, whereas the switchable LSTM contains two working modes: (1) generating mode (normal LSTM) and (2) retrieving mode (used to search objects in memory). LSTM has seen extensive use in a variety of fields, particularly in time series modeling, where it has been put to use in applications such as sequence prediction, natural language processing, and speech recognition [14]. However, there are only a few papers on how to use LSTM for system identification [9] and modeling [15,16], and as far as we know, there is no article on how to use it as a sparse recovery algorithm for compressive sensing to improve the quality of speech.

Therefore, in this paper, we proposed an LSTM based approach for sparse signal recovery of compressed signals for speech enhancement. The proposed technique replaces the greedy iterative CS signal recovery algorithm with a system model made with LSTM. This speeds up the process of recovering compressed signals. The proposed technique simultaneously recovers and enhances the speech signal. In addition, the proposed technique does not require any prior knowledge of the statistical properties of the signal and noise because the used LSTM can directly pull the needed information from the training data and learn it.

The remaining sections are organized as follows: Section 2 presents a concise literature assessment of relevant work. In Section 3, the theoretical foundations of the various processes utilized in this study are examined. An explanation of the proposed technique is provided in Section 4. The datasets description evaluation metrics are explained in Sections 5 and 6. In Section 6, the evaluation findings and comprehensive analysis of the proposed technique are presented. Additionally, a comparison of several approaches is reported in this section. Finally, based on the evaluation findings, the conclusion is presented in Section 7.

2. Theoretical Background

2.1. Framing and De-Framing

Framing is the process of segmenting an input signal into discrete units, or "frames", for further processing. The frames are designed to overlap one another, as demonstrated in Figure 1. Frames a, b, and c are labeled as F_a, F_b, and F_c in this figure, whereas the overlapping frames are labeled as OL_{ab} and OL_{bc}. To prevent data loss between consecutive frames, they are intentionally overlapped.

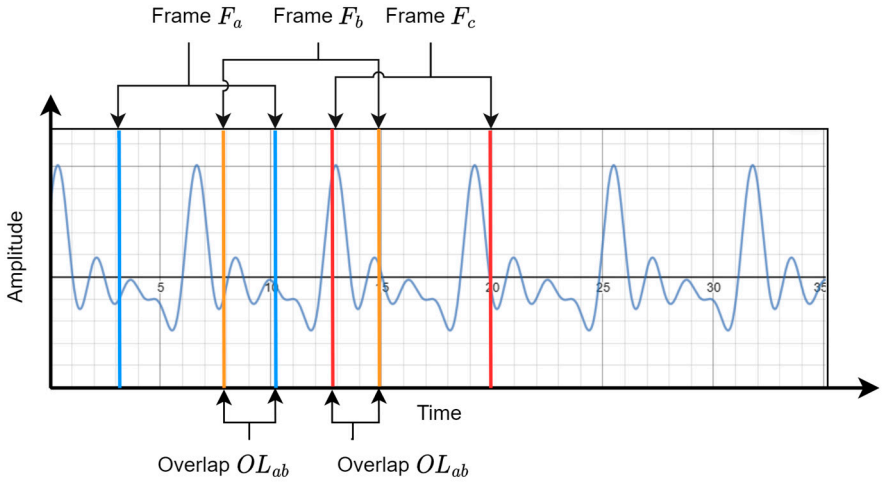

Figure 1. Process of dividing signal into frames.

Conventional spectral evaluation techniques work well for stationary-signal situations (i.e., a signal whose statistical properties remain constant over time) [17]. Being a nonstationary signal, the statistical properties of speech signals change over time. Therefore, the speech signals are processed in small chunks (frames) of samples where the properties of the signal can be assumed to be stationary. These frames of samples are then processed independently by signal processing algorithms [18]. When these processed frames are added together, it could cause spectral distortions. To avoid this, each frame is multiplied by the Hamming windows ($w(n)$) before it goes to the signal processing:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right),\ 0 \le n \le N-1 \tag{1}$$

where N denotes the samples in the speech frame. The windowed signal $y(n)$ can be calculated as follows:

$$y(n) = x(n) \times w(n) \tag{2}$$

Windowing acts as a low pass filter, enhancing the signal at the center and smoothing it at the edges.

2.2. Voice Activity Detector

Voice activity detection (VAD) refers to the challenge of identifying a speaker's voice amongst other sounds in a noisy environment. When it comes to speech processing modules and applications the detection of presence of speech in noisy audio signals is a crucial preprocessing phase [19]. In this work, the VAD detector based on the work presented in [19] is utilized. According to this model, the four different feature streams used are as follows:

1. Spectral Shape
2. Spectro-temporal modulations
3. Voicing
4. Long term variability

These feature streams are extracted from the input audio stream. The streams are divided into small frames, and then the DCT of each feature stream is taken. The DCT contains a large number of coefficients. However, the first 5 DCT components are sufficient to extract the most relevant context information from those frames.

A stream combination is then simply obtained by stacking the four context-expanded features streams into a single 20-dimensional frame vector. Finally, an MLP classifier was trained on these feature vectors. Speech segments are detected by thresholding the ratio of speech and non-speech outputs of MLP. Figure 2 shows an overview of the VAD.

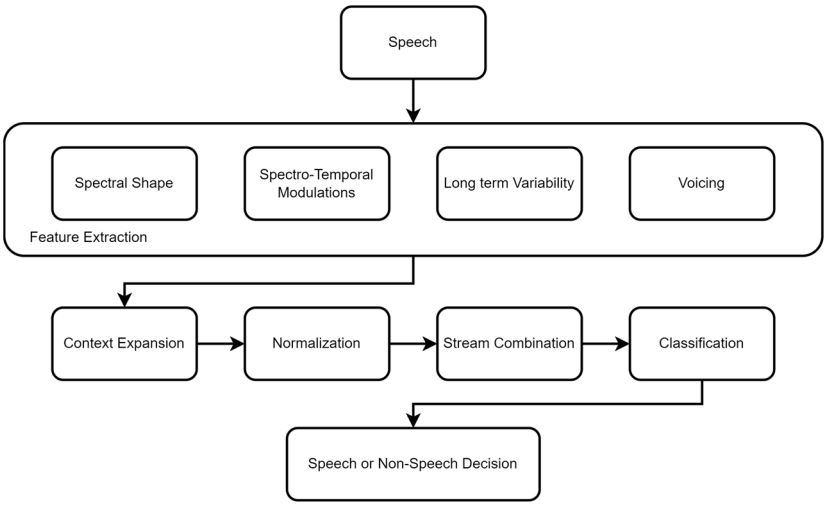

Figure 2. Overview of pre-processing steps in the used VAD.

2.3. Compressive Sensing

Compressive sensing is a signal processing technique that allows for the efficient acquisition and reconstruction of signals using fewer measurements than traditional methods. It is particularly useful in scenarios where signals are sparse or compressible, meaning that they have a lot of redundancy or can be represented by a small number of parameters. The basic idea behind compressive sensing is to acquire a small number of random linear projections of the signal, rather than measuring the signal directly. These projections are combined into a measurement vector, which is then used to reconstruct the original signal using a mathematical algorithm. Compressive sensing allows for the acquisition of signals using fewer measurements, which can be useful in scenarios where data storage or transmission is limited. Additionally, compressive sensing is often more robust to noise and measurement errors than traditional methods, as it can exploit the underlying structure of the signal to reduce the impact of these errors.

The basic equation for CS sparse signal recovery is given as follows:

$$[y]_{M\times 1} = [\phi]_{M\times N} \times [x]_{N\times 1} + [\nu]_{M\times 1} \tag{3}$$

where x and y denotes the sparse signal and observation vector respectively, while ϕ and v denotes the sensing matrix and noise respectively.

The following are the required conditions to recover the x from y:

1. $M << N$.
2. x must be k sparse, i.e., $k << N$.
3. ϕ must be a full rank matrix.

The estimation of the sparse signal x from observation y through Equation (3) needs the inverse of ϕ:

$$x \approx \phi^{-1} y, \text{ neglecting the } \nu \tag{4}$$

However, the above equation is only applicable for the square and non-singular ϕ matrix, and because in the present case ϕ is a non-square matrix, and also ϕ could be a singular matrix, the above equation cannot be used in the present case.

Therefore, to find a solution to the issue presented above the pseudo-inverse of the matrix ϕ can be employed. Pseudo-inverses are defined for matrices with real or complex entries, and are unique for these matrices. The singular value decomposition (SVD) is used to calculate the pseudo-inverse. An SVD in linear algebra is a way of factorizing a matrix of real or complex numbers. A matrix is decomposed by SVD into the following three matrices:

$$[\phi]_{M\times N} = [U]_{M\times M}[S]_{M\times N}\left[V^T\right]_{N\times N} \tag{5}$$

where the matrices $[U]$ and $[V]$ contain the singular vectors of $[\phi]$, where $[U]$ represents the left singular vectors, and $[V]$ represents the right singular vectors. On the other hand, the singular values of $[\phi]$ are represented by the diagonal elements of $[S]$. Using singular values, the pseudo-inverse of matrix $[\phi]$ is calculated as below:

$$\phi^+ = VS^+U^T \tag{6}$$

where S^+ is obtained by replacing each non-zero singular value in S with its reciprocal. By utilizing the solution described above, the reconstruction of x can be accomplished by applying the following equation:

$$x = \left(VS^+U^T\right)y \tag{7}$$

Equation (7) may have an infinite number of solutions; however, the optimal signal recovery can be obtained by finding the x, with the minimum number of non-zero entries (known as l_0 norm) that satisfies the $y = \phi x$ as given in the following equation:

$$min||x||_0, \; subject \; to \; y = \phi x \tag{8}$$

where, $\|\cdot\|_0$ denotes the l_0 norm.

For the partial reconstruction Equation (9) can be modified as follows:

$$min||x||_1, \; subject \; to \; ||y = \phi x||_2 \leq \beta \tag{9}$$

where, $||\cdot|\quad|_1$, $||\cdot|\quad|_2$ denotes the l_1, l_2 norm operations respectively, and β is the termination threshold.

2.4. RNN and LSTM Neural Networks

2.4.1. Recurrent Neural Networks (RNN)

The learning of temporal-sequential input, such as string, video, and voice, is frequently accomplished using RNNs, which are a neural network with single or several layer architectures built up of cyclic connections [20]. This network is distinguished by the application of the memory of a previous occurrence of information to the current incoming input. When handling sequential data, RNN offers an advantage.

An RNN node is made up of the current input and output denoted by x_t and y_t, respectively, and prior and current hidden state denoted by h_{t-1} and h_t, respectively, as shown in Figure 3.

Thus,

$$h_t = \delta_{hidden}\left(W_{hidden}h_{t-1} + W_{input}x_t + b_{hidden}\right) \tag{10}$$

$$y_t = \delta_{output}\left(W_{output}h_t + b_{output}\right) \tag{11}$$

where, the hidden and output layer activation functions, are δ_{hidden} and δ_{output}, respectively.

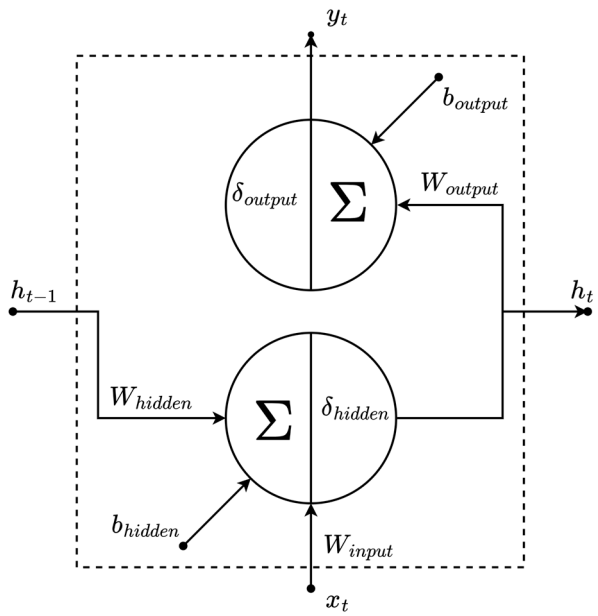

Figure 3. Schematic of an RNN node.

The input-to-hidden node connection weights are denoted by W_{input}, whereas the hidden-to-hidden and hidden-to-output node connection weights are denoted by W_{hidden}, and W_{output} respectively. The bias terms are designated as b_{output} and b_{hidden} for the output and hidden states, respectively.

The sigmoid, rectified linear unit or hyperbolic tangent are examples of existing functions with element-wise non-linearity features that can be found in the activation function in this situation.

2.4.2. Long Short-Term Memory (LSTM)

Due to disappearing or inflating gradient difficulties that impair the network's ability to back propagate gradients (long-term dependence problem), long-term sequential data can be challenging to train in standard RNN [21]. As shown in Figure 4, LSTM [22,23] replaces the standard nodes in the hidden layer of an RNN with "gates", which contain memory blocks with memory cells, to overcome the long-term dependency problem.

The following are the functionalities of different elements of the network:

- The activation of fresh information into the memory cell is controlled by the input gate (i_t).
- The output flow is controlled by the output gate (o_t).
- The forget gate (f_t) determines when to erase the internal state data.
- The main input to the memory cell is controlled by the input modulation gate (g_t).
- Cell internal recurrence is controlled by the internal state (c_t).
- The earlier data sample information is controlled by the hidden state (h_t) within the context window.

$$i_t = \delta(U_i x_t + W_i h_{t-1} + b_i) \tag{12}$$

$$o_t = \delta(U_o x_t + W_o h_{t-1} + b_o) \tag{13}$$

$$f_t = \delta\left(U_f x_t + W_f h_{t-1} + b_f\right) \tag{14}$$

$$g_t = \delta\left(U_g x_t + W_g h_{t-1} + b_g\right) \tag{15}$$

$$c_t = f_t c_{t-1} + g_t i_t \tag{16}$$

$$h_t = \tanh(c_t)o_t \tag{17}$$

where the b denotes the bias vector and the W and U denotes the weight matrices. As seen in Equations (12)–(15), the LSTM-RNN learns the optimal values of b, U, and W for the cell gates when trained through a dataset.

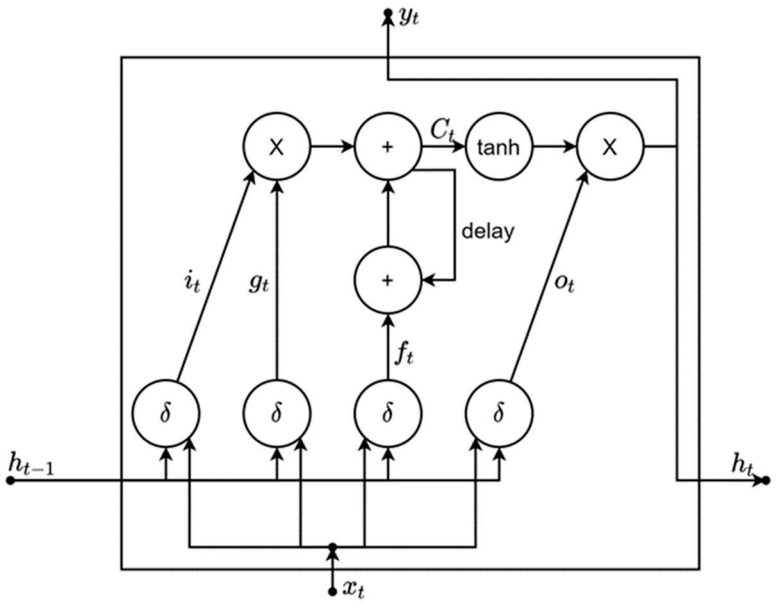

Figure 4. Architecture of an LSTM memory cell. Memory cell gates determine the output while forgetting and updating the earlier hidden states to govern the new information states.

3. Proposed Approach

3.1. Deep Learning System Modelling

The LSTM model's architecture is depicted in Figure 5, while its parametric details are presented in Table 1. It is a regression model that employs a recurrent neural network (RNN) with long short-term memory (LSTM) cells to detect long-term dependencies in the sequential input data.

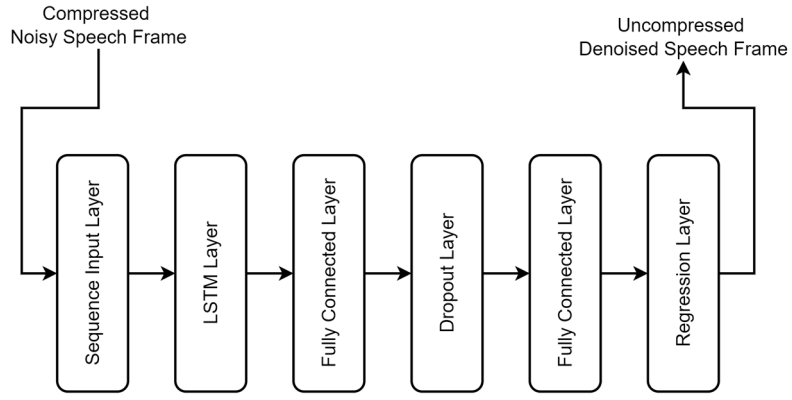

Figure 5. Structure of the LSTM model.

The first layer of the network is a "Sequence Input Layer", which is responsible for pre-processing sequential data by converting it into a format that can be fed into the next layer. In this work, we apply the compressed and noisy speech frame to this layer. The second layer is an "LSTM Layer", which comprises 50 LSTM cells and generates a sequence of hidden states. The output of the "LSTM Layer" is then fed to a "Fully Connected Layer" with 50 neurons, followed by a "Dropout Layer" with a dropout rate of 0.5, which randomly

sets some of the output values to zero during training to prevent overfitting. The fifth layer is another "Fully Connected Layer" with an output size equal to the frame size of the uncompressed speech frame. The final layer is a "Regression Layer", which produces the ultimate output. We use the Adam optimizer to train the model with a batch training data size of 20 and a gradient threshold of 1. We maintain a constant learning rate of 0.01 throughout the training.

Table 1. The configuration of the RNN-LSTM network, with training parameters.

Name of the Parameter	Value
First layer	Sequence Input Layer, with size equal to the observation vector ($[y]_{M\times 1}$) in Equation (3).
Second Layer	LSTM Layer with 50 Hidden Units.
Third Layer	Fully Connected Layer with output size 50.
Fourth Layer	Dropout Layer with dropout probability of 0.25.
Fifth Layer	Fully Connected Layer with output size equals to the sparse signal vector ($[x]_{N\times 1}$) in Equation (3).
Sixth Layer	Regression Layer.
Maximum Epochs	250.
Optimizer	Adam.
Learning Rate	0.01.
Gradient Threshold	1.0.
Batch Training Size	20.

3.2. Enhancement Algorithm

An illustrative block diagram of the proposed method is shown in Figure 6. Referring to this diagram, the proposed method can be divided into two stages: (1) training and (2) denoising.

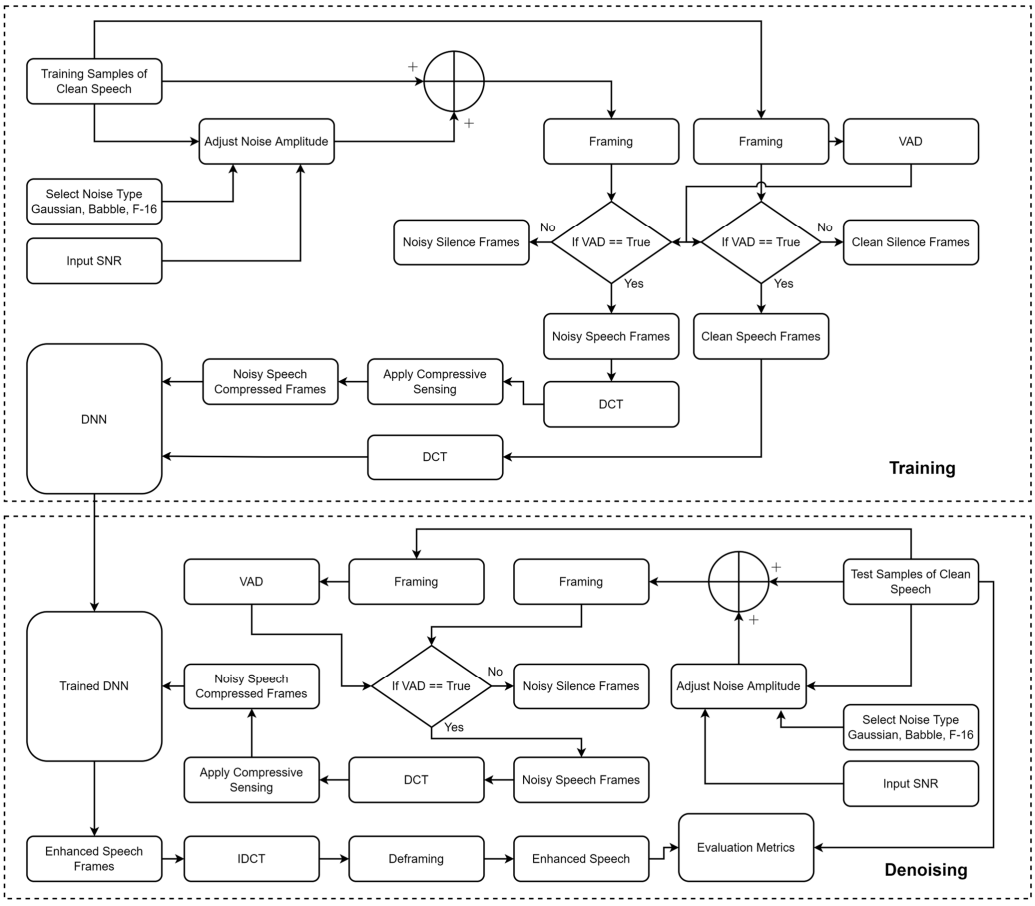

Figure 6. Architecture of the proposed speech enhancement system.

3.2.1. Training of LSTM

The proposed approach utilizes long short-term memory (LSTM) for the dual purpose of compressed signal recovery and denoising. Thus, a dataset containing compressed noisy speech frames and their corresponding uncompressed clean speech frames was required to train the LSTM for this task.

To obtain this dataset, we employed clean speech samples from the NOIZEUS dataset [24], which were corrupted by noises (white, babble, and f-16) taken from the Noisex92 dataset [25] at varying signal-to-noise ratios (SNRs). As the compressive sensing (CS) algorithm is block-based, the speech signals were divided into frames of fixed size with dimensions equivalent to the columns in the sensing matrix. These frames were then subjected to voice activity detection (VAD), and frames containing silence (non-speech) were discarded. This process resulted in two sets of frames, one of which contained the noisy speech (F_{noisy}) and the other, the respective clean speech (F_{clean}). The noisy speech frames were then transformed into the frequency domain via discrete cosine transform (DCT), where insignificant components were zeroed to obtain sparse frames (F_{sparse}). The sparse frames were subsequently compressed using the CS (Equation (3)), generating compressed noisy frames ($F_{compressed}$). The LSTM network depicted in Figure 5 was trained (the learning curve of the network is shown in Figure 7) on the compressed noisy and respective clean speech frames ($F_{compressed}$ and F_{clean}, respectively) generated by the aforementioned process. Once trained, this LSTM network could be used for both decompression and denoising purposes.

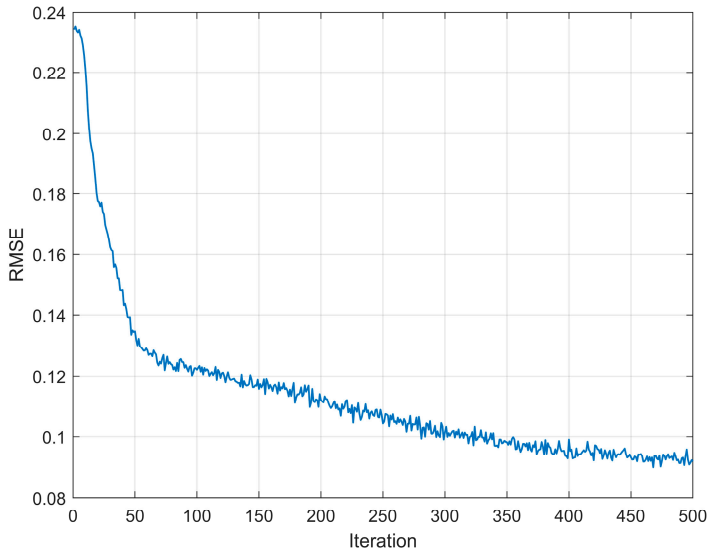

Figure 7. LSTM training RMSE at each iteration.

3.2.2. Speech Decompression and Denoising Using Trained LSTM

The trained LSTM can be used to recover the $\mathcal{F}_{clean}$ from the $\mathcal{F}_{compressed}$. To validate this a new set of $\mathcal{F}^{test}_{clean}$ $\mathcal{F}^{test}_{compressed}$ is generated through the same process as used during the training of LSTM. The $\mathcal{F}^{test}_{compressed}$ is decompressed and denoised using LSTM and compared against the $\mathcal{F}^{test}_{clean}$ for various speech quality measures.

4. Dataset

To test the performance of the proposed algorithm the audio files are collected from the following two datasets outlined below.

4.1. NOIZEUS Dataset

The NOIZEUS dataset [24] is a dataset of audio recordings and corresponding noise signals created to evaluate noise reduction algorithms. The dataset was created by researchers at the University of Texas at Dallas and is available for download. The NOIZEUS

dataset comprises 300 audio files, including 5 different noise types: white noise, pink noise, street noise, car noise, and babble noise. Each noise type has 6 different SNR levels, ranging from −5 dB to 10 dB, and includes 10 different noise samples. Apart from the noise signals, the dataset also includes 300 mixed signals obtained by mixing 10 different clean speech signals with the 5 noise types at the 6 different SNR levels. These clean speech signals can be used as reference signals for evaluating noise reduction algorithms.

4.2. Noisex92 Dataset

The Noisex92 dataset [25] is a dataset of audio recordings and corresponding noise signals created to evaluate speech enhancement algorithms. It was created by researchers at the Georgia Institute of Technology and is available for download. The Noisex92 dataset comprises 92 different noise signals, including both environmental noises such as car, train, and airplane sounds, as well as artificial noises such as white noise and babble noise. Each noise signal is 30 s long and is provided at a sampling rate of 16 kHz. In addition to the noise signals, the Noisex92 dataset also includes a set of clean speech signals recorded in a quiet environment. These clean speech signals are provided at a sampling rate of 8 kHz and can be used as reference signals for evaluating speech enhancement algorithms.

Both datasets have been widely used in research to assess speech enhancement and noise reduction algorithms because they provide a standardized set of test signals that can be used to compare different algorithms.

5. Performance Evaluation Metrics

Finding a relevant and effective evaluation metric for assessing speech quality, similarity, and intelligibility is a challenging task. It is worth noting that a metric that may be suitable for certain systems might not be convenient for others [26]. Therefore, to evaluate the performance of the proposed algorithm, the following three speech quality assessment measures were adopted:

5.1. Perceptual Evaluation of Speech Quality (PESQ)

PESQ [27] is an objective speech quality assessment algorithm that measures the quality of a degraded speech signal compared to a reference (clean) speech signal. It was developed by the International Telecommunication Union (ITU) and is described in ITU-T Recommendation P.862. PESQ uses a model of the human auditory system to simulate the effect of signal distortions on speech quality. It analyzes the degraded speech signal to extract perceptual features like loudness, spectral balance, and temporal properties. These features are then compared to those of the reference speech signal, and a quality score is generated to reflect the perceived speech quality. The output of PESQ is a score ranging from −0.5 to 4.5, with higher scores indicating better speech quality. This score is used to evaluate the performance of speech enhancement algorithms, codecs, and other speech processing systems.

5.2. Short-Time Objective Intelligibility (STOI)

STOI [28] is an objective speech quality measure which evaluates the intelligibility of a degraded speech signal by comparing it with a reference (clean) speech signal. STOI uses short-term spectral and temporal characteristics of speech signals to compute a measure of similarity between the degraded and reference speech signals. It estimates the overlap between the spectral content of the two signals using the modulation transfer function (MTF). The MTF describes the amount of spectral detail preserved in the degraded speech signal compared to the reference speech signal. STOI divides the speech signals into short time frames, usually 20–30 ms long. For each frame, it calculates the MTF and derives a similarity measure based on the degree of overlap between the MTF of the degraded and reference speech signals. It then combines the similarity measures across all time frames to generate a single STOI score that reflects the overall intelligibility of the degraded speech signal. The output of STOI is a score that ranges from 0 to 1, with higher scores indicating

better speech intelligibility. The score can be used to evaluate the performance of speech enhancement algorithms, codecs, and other speech processing systems.

5.3. Signal-to-Distortion Ratio (SDR)

Signal-to-distortion ratio (SDR) [29] is a metric used to measure the quality of an audio signal in the presence of distortion or interference. It quantifies the ratio of the power of the clean (desired) signal to the power of the distortion or interference. The SDR is usually expressed in decibels (dB), which is a logarithmic scale. A higher SDR value indicates better signal quality, as it means the desired signal power is higher compared to the distortion or interference power. Conversely, a lower SDR value indicates more pronounced distortion or interference relative to the desired signal.

The perceptual evaluation of speech quality (PESQ) metric has gained significant adoption as a standard for accurately assessing the quality of speech signals. Its ability to predict subjective quality ratings with high precision has contributed to its widespread usage. Conversely, the short-time objective intelligibility (STOI) metric has also emerged as a commonly accepted standard for evaluating speech intelligibility. It offers advantages such as rapid processing, user-friendly implementation, and consistent outcomes across diverse speech and noise characteristics.

Additionally, the signal-to-distortion ratio (SDR) metric finds frequent application in various audio processing domains, including source separation, denoising, and audio enhancement. Its primary purpose is to measure the efficacy of algorithms and techniques in preserving the inherent quality of the original signal while effectively mitigating undesirable artifacts and noise.

All these measures are commonly used in various industries, including telecommunications, broadcasting, and consumer electronics, to evaluate the performance of speech processing systems.

6. Results Analysis

The enhancement method that is presented herein is compared with three baseline methods, OMP [6], CoSaMP [30], StOMP [4], K-SVD based CS technique (K-SVDCS) [31] (compared only for babble noise, as the paper provided the results for this noise only.), and some other human perception related objective functions-based methods like DNN-PMSEQ [32], and DNN (wMSE-SVS) [32]. A comparison is made for speech signals affected by white, babble, and F-16 noises. The noise samples are collected from the Noisex92 dataset, whereas the clean speech samples are collected from the NOIZEUS dataset. For evaluation of the proposed algorithm, the clean speech samples are mixed with one of these noises at a time. The amplitude of the noise is modified before mixing according to the required SNR. Finally, to compare the performance of each algorithm, two objective measures, perceptual evaluation of speech quality (PESQ) and short-time objective intelligibility (STOI), are utilized. The performance comparison of the proposed technique with different variants of the OMP algorithm for white, babble, and f-16 noises is presented in Tables 2–4, respectively.

Table 2. PESQ, STOI, and SDR results comparison for white noise, using NOIZEUS dataset sample sp05.wav.

	Techniques											
	OMP			**CoSaMP**			**StOMP**			**Proposed**		
SNR (dB)	**PESQ**	**STOI**	**SDR**	**PESQ**	**STOI**	**SDR**	**PESQ**	**STOI**	**SDR**	**PESQ**	**STOI**	**SDR**
0	1.801	0.688	9.740	1.759	0.634	7.226	1.796	0.608	8.576	2.383	0.820	6.647
5	2.319	0.764	9.786	2.069	0.689	7.229	2.173	0.699	8.453	2.593	0.808	10.982
10	2.57	0.793	13.230	2.403	0.734	12.002	2.509	0.768	12.642	2.680	0.841	13.593
15	2.759	0.837	16.099	2.611	0.776	15.445	2.708	0.817	15.748	2.778	0.848	15.162
20	2.957	0.877	17.650	2.760	0.824	17.471	2.874	0.869	17.574	2.985	0.889	16.226

Table 3. PESQ, STOI, and SDR results comparison for babble noise, using NOIZEUS dataset sample sp05.wav.

	Techniques														
	OMP			CoSaMP			StOMP			K-SVDCS			Proposed		
SNR (dB)	PESQ	STOI	SDR	PESQ	STOI	SDR	PESQ	STOI	SDR	PESQ	STOI	SDR	PESQ	STOI	SDR
0	1.78	0.652	1.138	1.917	0.683	1.098	1.969	0.644	1.002	1.96	0.66	--	2.547	0.807	4.237
5	2.216	0.736	6.565	2.251	0.726	4.693	2.308	0.715	4.942	2.28	0.72	--	2.583	0.813	7.664
10	2.434	0.811	10.708	2.513	0.759	9.592	2.513	0.771	9.876	2.52	0.79	--	2.720	0.826	11.471
15	2.606	0.827	14.032	2.646	0.774	13.743	2.709	0.805	13.770	2.69	0.81	--	2.752	0.842	14.035
20	2.667	0.859	16.552	2.772	0.816	16.496	2.853	0.847	16.505	2.85	0.83	--	2.861	0.866	15.182

Table 4. PESQ, STOI, and SDR results comparison for f-16 noise, using NOIZEUS dataset sample sp05.wav.

	Techniques											
	OMP			CoSaMP			StOMP			Proposed		
SNR (dB)	PESQ	STOI	SDR	PESQ	STOI	SDR	PESQ	STOI	SDR	PESQ	STOI	SDR
0	1.684	0.659	2.138	1.887	0.629	1.037	1.942	0.591	1.461	2.527	0.849	4.917
5	2.101	0.751	7.399	2.183	0.682	5.005	2.268	0.69	6.002	2.648	0.835	9.625
10	2.509	0.787	11.496	2.496	0.745	10.128	2.577	0.756	10.669	2.926	0.837	12.788
15	2.654	0.836	14.828	2.632	0.756	14.233	2.747	0.808	14.422	2.873	0.849	14.370
20	2.838	0.862	16.959	2.747	0.796	16.758	2.878	0.839	16.808	2.951	0.903	13.453

For the white noise scenario (Table 2), the proposed algorithm achieves a PESQ value of 2.383 at 0 dB SNR, which is significantly higher than the values achieved by OMP (1.801), CoSaMP (1.759), and StOMP (1.796). At higher SNR levels (5 dB to 15 dB), the proposed algorithm continues to outperform the other algorithms, but the relative improvement drops. Lastly, at 20 dB SNR, the proposed algorithm achieves a PESQ value of 2.985, which is slightly higher than the value achieved by OMP (2.957). For the STOI value, the proposed algorithm achieves a STOI value of 0.753 at 0 dB SNR, which is significantly higher than the values achieved by OMP (0.688), CoSaMP (0.634), and StOMP (0.608). At higher SNR levels, the proposed algorithm continues to outperform the other algorithms. The proposed algorithm shows a trend of decreasing percentage improvement in the PESQ and SOTI as SNR increases. In terms of relative improvements over other algorithms, the proposed algorithm achieves the highest PESQ percentage improvement of 32.35%, 35.47%, and 32.68% over OMP, CoSaMP, and StOMP, respectively, at 0 dB SNR (Figure 8). Whereas the highest STOI percentage improvement was 19.19% over OMP at 0 dB, 29.34% over CoSaMP at 0 dB, and 34.87% over StOMP at 0 dB (Figure 9). The evaluation results for SDR show that OMP outperforms other algorithms at 0 dB, 15 dB, and 20 dB SNRs, whereas the proposed algorithm outperforms others at 5 dB and 10 dB SNRs. In terms of percentage improvements, the proposed algorithm achieves improvements of 12.22%, 51.92%, and 29.92% over the OMP, CoSaMP, and StOMP algorithms, respectively, at 10 dB SNR (Figure 10).

For the babble noise scenario (Table 3), the proposed algorithm achieves a PESQ value of 2.547 at 0 dB SNR, which is significantly higher than the values achieved by OMP (1.78), CoSaMP (1.917), StOMP (1.969), and K-SVDCS (1.98). The proposed algorithm continues to outperform the other algorithms, but the relative improvement drops at higher SNRs. For STOI values, the proposed algorithm achieves a STOI result of 0.807 at 0 dB SNR, which is much higher than the results achieved by OMP (0.652), CoSaMP (0.683), StOMP (0.644), and K-SVDCS (0.66). At 5 dB SNR, the proposed algorithm achieves a STOI result of 0.813, which is somewhat higher than the results achieved by OMP (0.736), CoSaMP (0.726), StOMP (0.715), and K-SVDCS (0.72). At 10 dB SNR, the proposed algorithm achieves a STOI result of 0.826, which is higher than the results achieved by OMP (0.811), CoSaMP (0.759), StOMP (0.771), and K-SVDCS (0.79). In terms of relative improvements over other algorithms, the proposed algorithm achieves the highest PESQ percentage

improvement of 43.09%, 32.86%, 29.36%, and 29.95% over OMP, CoSaMP, StOMP, and K-SVDCS, respectively, at 0 dB SNR (Figure 11), whereas the highest SOTI percentage improvement was 23.77%, 18.16%, 25.31%, and 22.27% over OMP, CoSaMP, StOMP, and K-SVDCS, respectively, at 0 dB SNR (Figure 12). The evaluation results for SDR show that the proposed algorithm outperforms other algorithms at 0 dB SNR by a huge 272.32%, 285.88%, and 322.85%, whereas the proposed algorithm outperforms others at 5 dB and 10 dB SNRs. While the proposed algorithm continuously outperforms other algorithms until 15 dB, the margin drops significantly, and at 20 dB, other algorithms take over the proposed algorithm (Figure 13).

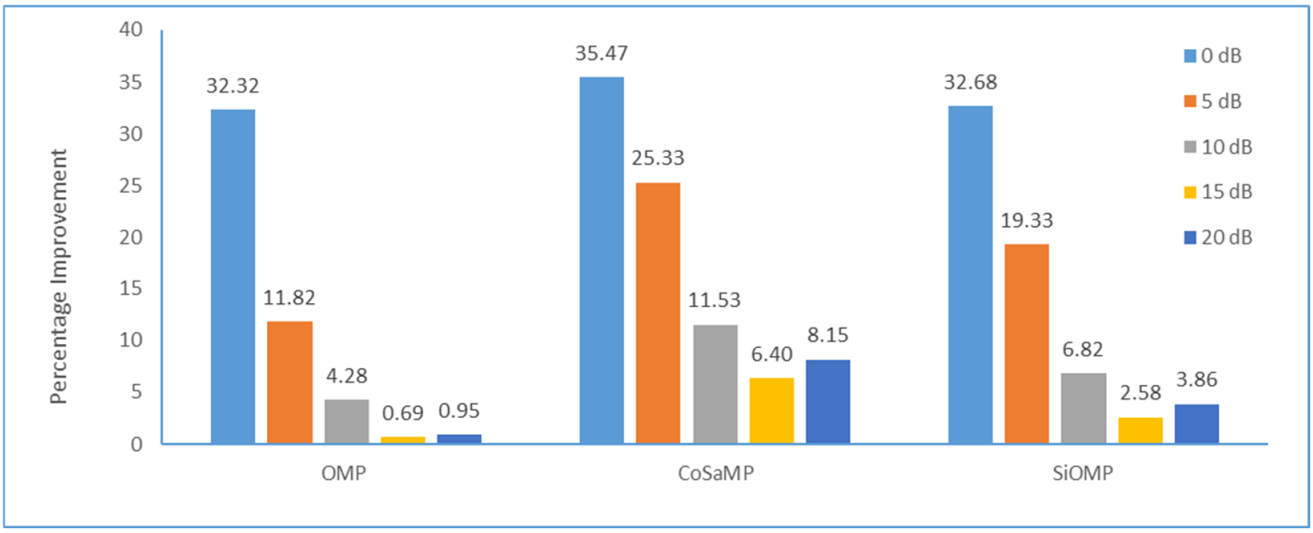

Figure 8. PESQ percentage improvement achieved by proposed algorithm over different algorithms for different noise levels of white noise.

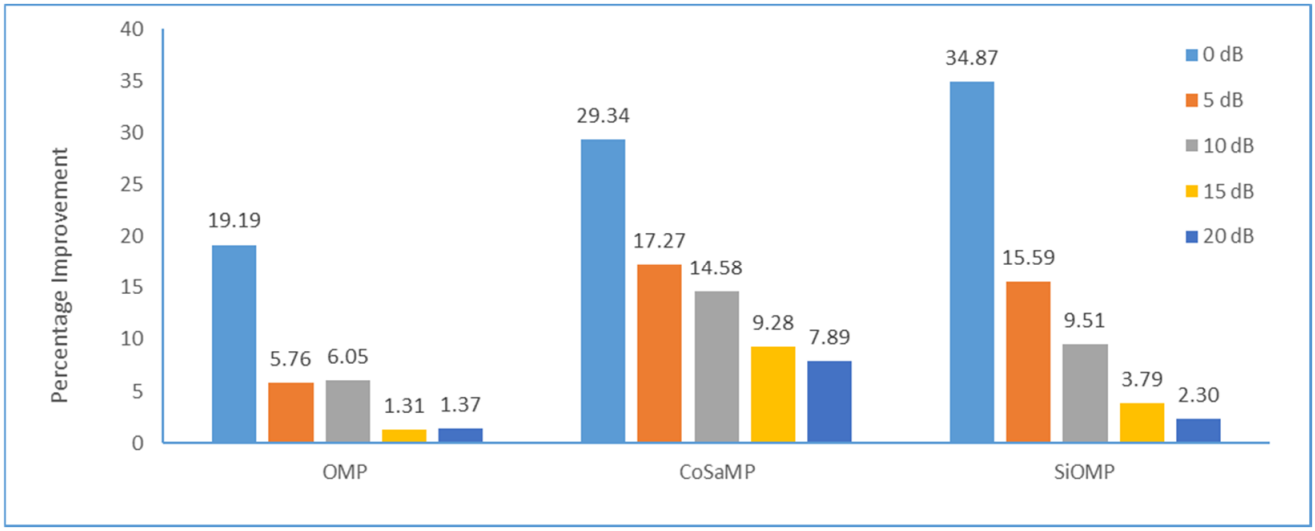

Figure 9. STOI percentage improvement achieved by proposed algorithm over different algorithms for different noise levels of white noise.

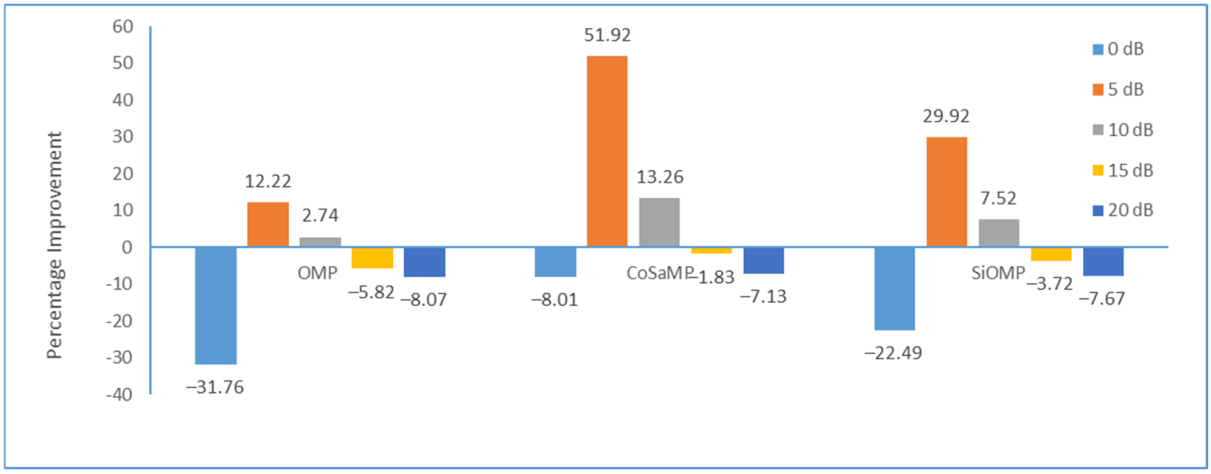

Figure 10. SDR percentage improvement achieved by proposed algorithm over different algorithms for different noise levels of white noise.

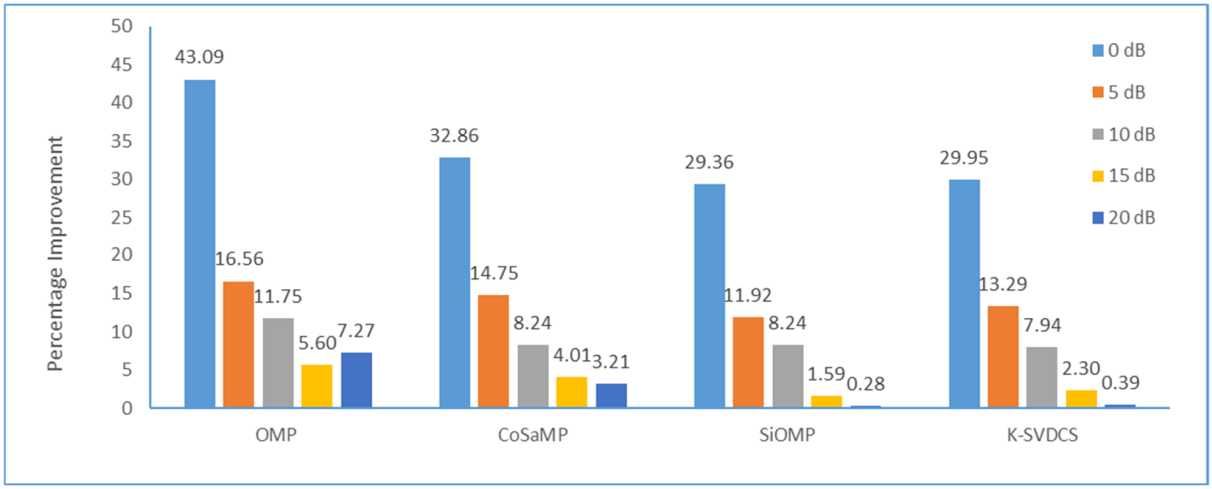

Figure 11. PESQ percentage improvement achieved by proposed algorithm over different algorithms for different noise levels of babble noise.

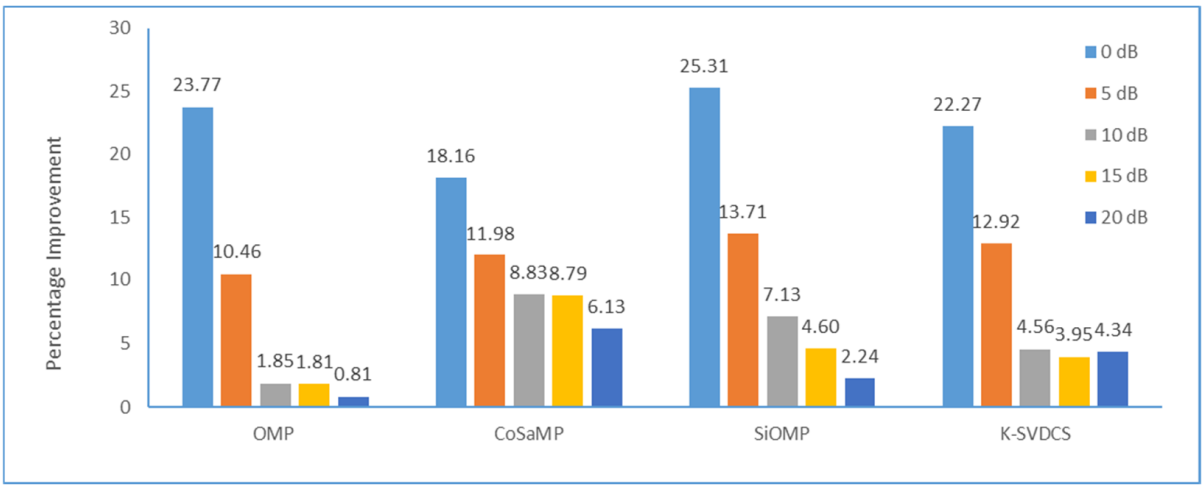

Figure 12. STOI percentage improvement achieved by proposed algorithm over different algorithms for different noise levels of babble noise.

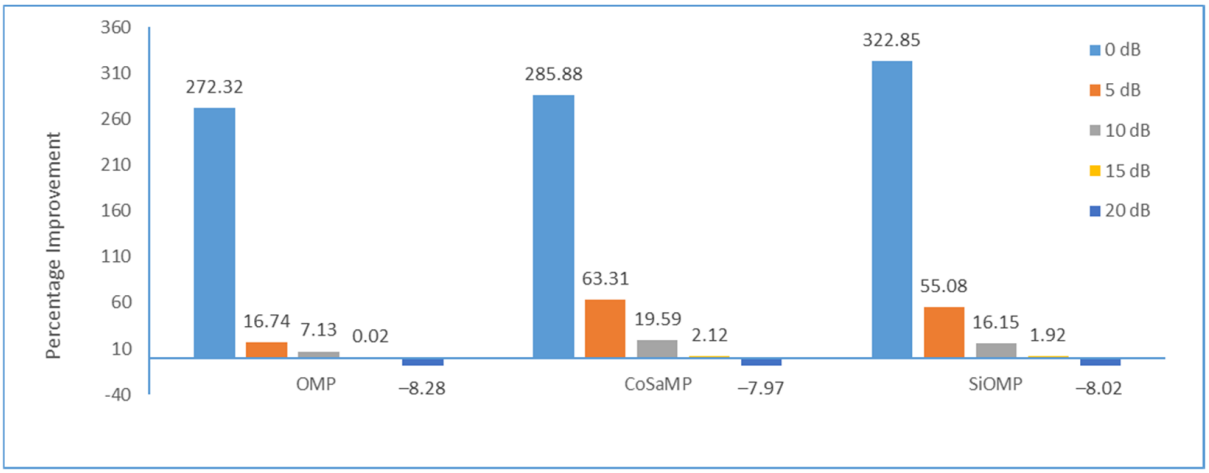

Figure 13. SDR percentage improvement achieved by proposed algorithm over different algorithms for different noise levels of babble noise.

For the f-16 noise scenario (Table 4), the proposed algorithm achieves a PESQ value of 2.527 at 0 dB SNR, which is significantly higher than the values achieved by OMP (1.684), CoSaMP (1.887), and StOMP (1.942). The proposed algorithm continues to outperform the other algorithms, but the relative improvement drops at higher SNRs. For STOI values, the proposed algorithm achieves a STOI result of 0.849 at 0 dB SNR, which is significantly higher than the results achieved by OMP (0.659), CoSaMP (0.629), and StOMP (0.591). Further improvement is seen at 5 dB SNR, where the proposed algorithm achieves a STOI result of 0.835, which is significantly higher than the results achieved by OMP (0.751), CoSaMP (0.682), and StOMP (0.69). In terms of relative improvements over other algorithms, the proposed algorithm achieves the highest PESQ percentage improvement of 50.06%, 33.92%, and 30.12% over OMP, CoSaMP, and StOMP, respectively, at 0 dB SNR (Figure 14), whereas the highest SOTI percentage improvement was 28.83%, 34.98%, and 43.65% over OMP, CoSaMP, and StOMP, respectively, at 0 dB SNR (Figure 15). The evaluation results for SDR show that the proposed algorithm outperforms other algorithms at 0 dB SNR by a huge 129.98%, 374.16%, and 236.55%, whereas the proposed algorithm outperforms others at 5 dB and 10 dB SNRs. While the proposed algorithm continuously outperforms other algorithms until 15 dB, the margin drops significantly, and at 20 dB, other algorithms take over the proposed algorithm (Figure 16).

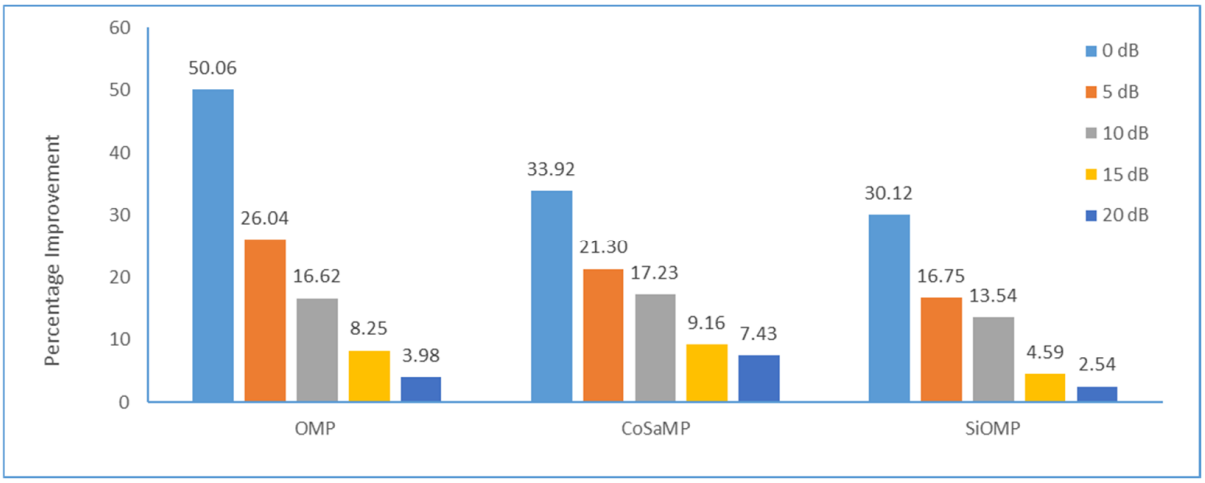

Figure 14. PESQ percentage improvement achieved by proposed algorithm over different algorithms for different noise levels of f-16 noise.

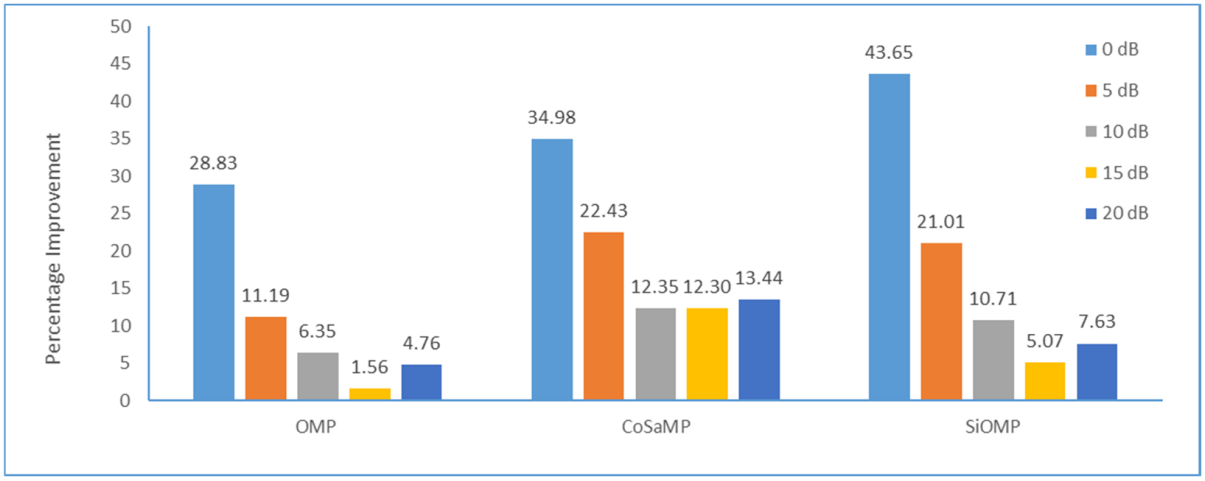

Figure 15. STOI percentage improvement achieved by proposed algorithm over different algorithms for different noise levels of f-16 noise.

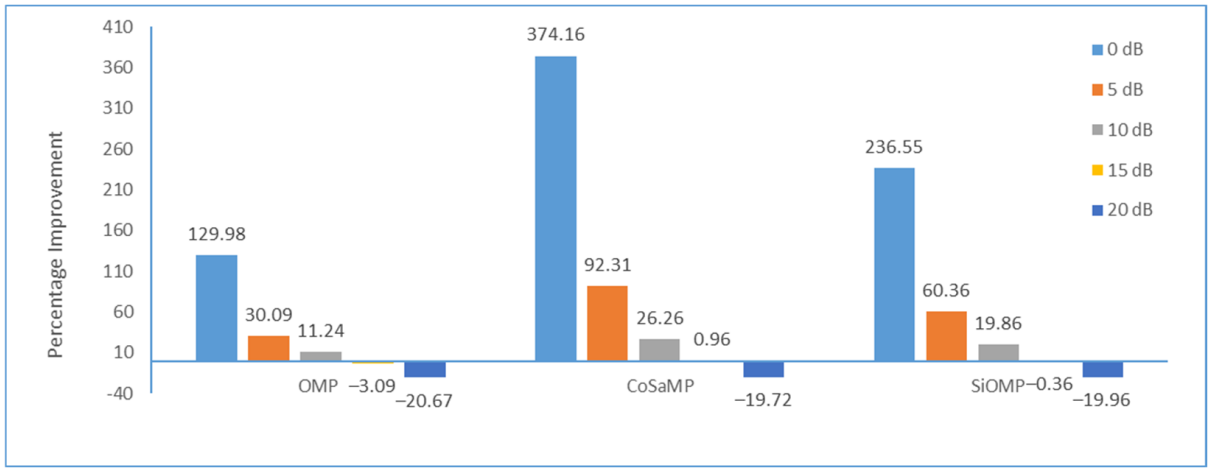

Figure 16. SDR percentage improvement achieved by proposed algorithm over different algorithms for different noise levels of f-16 noise.

To further validate the performance of the proposed algorithm, a comparison with deep learning based algorithms for SDR measures is presented in Table 5. These results show that the proposed technique performs better at all SNRs and achieves 137.39% and 60.56% improvements over DNN-PMSQE and DNN (wMSE-SVS), respectively, at 0 dB SNR.

Table 5. SDR comparison for white noise, using NOIZEUS dataset sample sp05.wav.

	Techniques		
SNR (dB)	**DNN PMSQE [32]**	**DNN (wMSE-SVS) [32]**	**Proposed**
0	2.80	4.14	6.647
5	7.46	7.85	10.982
10	11.08	9.94	13.593
15	13.62	10.89	15.162
20	15.02	11.19	16.226

These results indicate that the proposed algorithm is more effective than the other algorithms in improving speech quality in the presence of any kind of noise, especially at low SNR levels.

7. Conclusions

This paper presents a compressive sensing (CS) based speech enhancement approach that uses LSTM networks for simultaneous speech recovery and enhancement. This algorithm recovers compressed signals without an iterative process, making the process fast and straightforward. The proposed approach does not require prior knowledge of signal and noise statistical properties for sensing matrix optimization because the used LSTM can directly extract and learn the required information from training data. White, babble, and f-16 noises were tested. PESQ and STOI were compared to other OMP-based CS algorithms to validate the proposed approach. Experimental results show that the proposed approach outperforms OMP, CoSaMP, StOMP, K-SVDCS, and DNN based algorithms in terms of PESQ, STOI, and SDR metrics for various types of noise, including white noise, babble noise, and f-16 noise. The degree of improvement varies at different SNR levels, with the greatest improvements observed at low SNR levels. The proposed algorithm consistently achieves higher PESQ scores, indicating improved speech quality, compared to the other algorithms.

Author Contributions: Methodology, V.S.; Software, V.S.; Validation, V.S.; Formal analysis, P.D.S.; Supervision, P.D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used are publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Das, N.; Chakraborty, S.; Chaki, J.; Padhy, N.; Dey, N. Fundamentals, Present and Future Perspectives of Speech Enhancement. *Int. J. Speech Technol.* **2020**, *24*, 883–901. [CrossRef]
2. Donoho, D.L. For Most Large Underdetermined Systems of Linear Equations the Minimal ↕1-Norm Solution Is Also the Sparsest Solution. *Commun. Pure Appl. Math.* **2006**, *59*, 797–829. [CrossRef]
3. Ahani, S.; Ghaemmaghami, S.; Wang, Z.J. A Sparse Representation-Based Wavelet Domain Speech Steganography Method. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 80–91. [CrossRef]
4. Donoho, D.L.; Tsaig, Y.; Drori, I.; Starck, J.-L. Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit. *IEEE Trans. Inf. Theory* **2012**, *58*, 1094–1121. [CrossRef]
5. Crespo Marques, E.; Maciel, N.; Naviner, L.; Cai, H.; Yang, J. A Review of Sparse Recovery Algorithms. *IEEE Access* **2019**, *7*, 1300–1322. [CrossRef]
6. Yang, H.; Hao, D.; Sun, H.; Liu, Y. Speech Enhancement Using Orthogonal Matching Pursuit Algorithm. In Proceedings of the 2014 International Conference on Orange Technologies, Xi'an, China, 20–23 September 2014; pp. 101–104.
7. de Paiva, N.M.; Marques, E.C.; de Barros Naviner, L.A. Sparsity Analysis Using a Mixed Approach with Greedy and LS Algorithms on Channel Estimation. In Proceedings of the 2017 3rd International Conference on Frontiers of Signal Processing (ICFSP), Paris, France, 6–8 September 2017; pp. 91–95.
8. Shinde, P.P.; Shah, S. A Review of Machine Learning and Deep Learning Applications. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018; pp. 1–6.
9. Ljung, L.; Andersson, C.; Tiels, K.; Schön, T.B. Deep Learning and System Identification. *IFAC-PapersOnLine* **2020**, *53*, 1175–1181. [CrossRef]
10. Glorot, X.; Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
11. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent Advances in Convolutional Neural Networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
12. Salehinejad, H.; Sankar, S.; Barfett, J.; Colak, E.; Valaee, S. Recent Advances in Recurrent Neural Networks. *arXiv* **2017**, arXiv:1801.01078.
13. Staudemeyer, R.C.; Morris, E.R. Understanding LSTM—A Tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv* **2019**, arXiv:1909.09586.
14. Graves, A.; Mohamed, A.; Hinton, G. Speech Recognition with Deep Recurrent Neural Networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649.
15. Gonzalez, J.; Yu, W. Non-Linear System Modeling Using LSTM Neural Networks. *IFAC-PapersOnLine* **2018**, *51*, 485–489. [CrossRef]
16. Wang, Y. A New Concept Using LSTM Neural Networks for Dynamic System Identification. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 5324–5329.
17. Hamid, O.K. Frame Blocking and Windowing Speech Signal. *J. Inf.* **2018**, *4*, 8.

18. Prabhu, K.M.M. *Window Functions and Their Applications in Signal Processing*; Taylor & Francis: Abingdon, UK, 2014; ISBN 978-1-4665-1584-0.
19. Segbroeck, M.V.; Tsiartas, A.; Narayanan, S. A Robust Frontend for VAD: Exploiting Contextual, Discriminative and Spectral Cues of Human Voice. *Interspeech* **2013**, *5*, 704–708.
20. Kim, B.-H.; Pyun, J.-Y. ECG Identification for Personal Authentication Using LSTM-Based Deep Recurrent Neural Networks. *Sensors* **2020**, *20*, 3069. [CrossRef] [PubMed]
21. Kolen, J.F.; Kremer, S.C. Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies. In *A Field Guide to Dynamical Recurrent Networks*; IEEE: Piscataway, NJ, USA, 2001; pp. 237–243. ISBN 978-0-470-54403-7.
22. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
23. Hu, C.; Wu, Q.; Li, H.; Jian, S.; Li, N.; Lou, Z. Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation. *Water* **2018**, *10*, 1543. [CrossRef]
24. Hu, Y.; Loizou, P.C. Subjective Comparison and Evaluation of Speech Enhancement Algorithms. *Speech Commun.* **2007**, *49*, 588–601. [CrossRef]
25. Varga, A.; Steeneken, H.J.M. 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 Commun.* **1993**, *12*, 247–251. [CrossRef]
26. Al-Radhi, M.S.; Csapó, T.G.; Németh, G. Continuous Noise Masking Based Vocoder for Statistical Parametric Speech Synthesis. *IEICE Trans. Inf. Syst.* **2020**, *E103-D*, 1099–1107. [CrossRef]
27. Rix, A.W.; Beerends, J.G.; Hollier, M.P.; Hekstra, A.P. Perceptual Evaluation of Speech Quality (PESQ)—A New Method for Speech Quality Assessment of Telephone Networks and Codecs. In Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221), Salt Lake City, UT, USA, 7–11 May 2001; Volume 2, pp. 749–752.
28. Taal, C.H.; Hendriks, R.C.; Heusdens, R.; Jensen, J. A Short-Time Objective Intelligibility Measure for Time-Frequency Weighted Noisy Speech. In Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, USA, 14–19 March 2010; pp. 4214–4217.
29. Vincent, E.; Gribonval, R.; Fevotte, C. Performance Measurement in Blind Audio Source Separation. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 1462–1469. [CrossRef]
30. Cevher, V.; Waters, A. The CoSamp Algorithm. In *ELEC 639: Graphical Models Lecture Notes*; Rice University: Houston, TX, USA, 2008.
31. Haneche, H.; Boudraa, B.; Ouahabi, A. A New Way to Enhance Speech Signal Based on Compressed Sensing. *Measurement* **2020**, *151*, 107117. [CrossRef]
32. Martin-Doñas, J.M.; Gomez, A.M.; Gonzalez, J.A.; Peinado, A.M. A Deep Learning Loss Function Based on the Perceptual Evaluation of the Speech Quality. *IEEE Signal Process. Lett.* **2018**, *25*, 1680–1684. [CrossRef]