Machine Learning with Feature Extractions for Regression Estimation of Binaural Sound Source Localization

by

Philippe Y. Massicotte

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Electrical and Computer Engineering

Carleton University
Ottawa, Ontario

Abstract

Binaural sound source localization is the determination of the position of a sound source based on two data sensors, microphones, mimicking the human auditory system. Many audio processing systems in our daily work and life rely on sound source localization, such as speech enhancement/recognition and human-robot interaction. However, the accuracy of sound source localization under adverse acoustic scenarios is still hard to ensure. This thesis proposes machine learning with feature extractions to estimate the sound source localization by manipulating and analyzing data collected by public Head Related Transfer Function databases. The two proposed methods are wavelet scattering long short-term memory and wavelet scattering convolutional neural network. These developed methods are studied in classification and regression approaches for different scenarios. The results demonstrate that the proposed methods achieve excellent performance in multiple noisy environments compared to recent literature, especially in regression binaural sound source localization.

Acknowledgements

I would like to sincerely thank my research director, Prof. Hicham Chaoui, as well as my research co-director, Prof. Messaoud Ahmed Ouameur from Université du Québec à Trois-Rivières, for giving me the opportunity to work on this project. Your support and advice throughout my master's degree journey are greatly appreciated. I would also like to thank the members of the *Laboratoire des Signaux et Systèmes Intégrés* (LSSI) for their assistance in developing and advancing various parts of this project. Next, I thank my father, mother, and partners, who pushed me with their encouragement and moral support. Additionally, I want to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds de recherche – Nature et technologies (FRQNT) for the scholarships.

Thank you

Table of Contents

List of Tables

List of Illustrations

List of Abbreviations

AI	Artificial Intelligence
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
dB	Decibels
DOA	Direction-Of-Arrival
DNN	Deep Neural Network
FC	Fully Connected
FT	Fourier Transform
GPU	Graphics Processing Unit
GCC	Gammatone Cepstral Coefficient
HRIR	Head Related Impulse Response
HRTF	Head Related Transfer Function
ILD	Interaural Level Difference
IRCAM	Institute for Research and Coordination in Acoustics/Music
IPD	Interaural Phase Difference
ITD	Interaural Time Difference
LSSI	*Laboratoire des Signaux et Systèmes Intégrés*
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MFCC	Mel Frequency Cepstral Coefficient
NN	Neural Network
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RRMSE	Relative Root Mean Square Error
SNN	Spiking Neural Network

SNR	Signal-to-Noise Ratio
SSL	Sound Source Localisation
STFT	Short-Time Fourier Transform
TDL	Time Delay Line
TF	Time Frequency
TF-CNN	Time-Frequency Convolutional Neural Network
TF-2CNN	Time-Frequency with Two Convolutional Neural Networks
WS	Wavelet Scattering
WS-CNN	Wavelet Scattering Convolutional Neural Network
WS-LSTM	Wavelet Scattering Long Short-Term Memory

Chapter 1: Introduction

Binaural sound source localization (SSL) has recently gained popularity for various applications [1] [2]. SSL has long been an essential and indispensable technique for animals living in the wild. SSL enables offspring but equally prey or predators in a hostile environment to be rapidly located [3] [4]. Humans and many other species' auditory systems can localize sound sources to live in surroundings which depend on their sense organs, namely their ears and the brain's information-processing ability [1] [5]. Ears are exceedingly sophisticated: information about sources from different positions is encoded in the signals from the two ears via complicated directivity patterns. This function allows for 3D localization using only two channels. From a binaural recording, an advanced 3D spatial audio positioning system would not need more information to determine a sound source location [3], as can be done by a human. This may be sufficient and necessary for human perception of 3D sounds [6] [7]. SSL can help detect particular occurrences in robotics and security systems since the sources do not require a straight line of sight, and sense is not limited to functioning camera angles. A domestic robot, Fig. 1.1, may hear what is going on in the next room by sensing sound sent through the wall, while cameras don’t have this capability [8] [9]. SSL is essential and commonly used in human society, many audio processing systems rely on SSL, speech enhancement/recognition, human-robot interaction, and industry 4.0/5.0 are application examples [10] [11]. SSL is also a crucial and, sometimes, integrated stage in separating and cleaning sources [12].

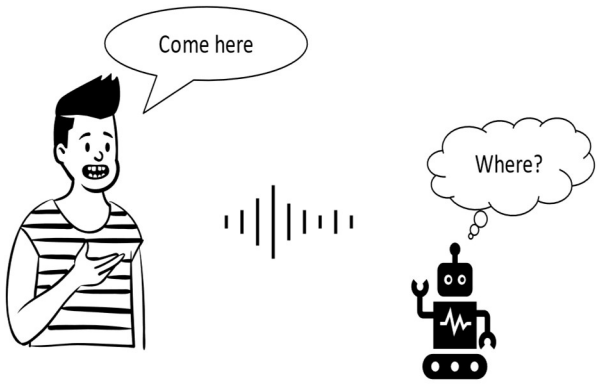


Fig. 1.1 Human-robot interaction SSL

Many efforts to precisely localize sources rely on massive multichannel arrays [13] [14]. In the case of more than two microphones, it is not necessarily considered a binaural type of SSL. However, there are several limits to these methods. The size of the arrays and the number of microphones employed determine sensitivity and accuracy. In some cases, logistical restrictions may prevent the usage of massive arrays. Calibration and channel matching are complicated processes for big arrays. The processing of multichannel signals is likewise not simple. Inspired by human and animal binaural hearing, several researches have advocated using a dummy head with two microphones with fine-tuned directivities and sophisticated signal processing techniques to accomplish source localization, with promising results [2] [8] [15] [16] [17] [18] [19] [20] [21]. However, the accuracy estimation of a sound source is a significant challenge in noisy environments. Furthermore, the introduction of Machine Learning (ML) approaches [22], Neural Networks (NN) [23], and Deep Neural Networks (DNN) [20] have been used to address sound source localization challenges.

1.1 Problematic

Estimating the direction of sound is a non-trivial problem. Most works focus on estimation based on classification models. These models forecast predetermined positions in a limited number and often at middle distances in space. Most research in SSL uses multisource, such as an array of microphones, to ensure acceptable precision [9]. A challenge for the binaural SSL is to use only two microphones as input data to mimic the human auditory system limited to two ears on each side of the head.

The interest of this thesis is in the estimation problem according to a regression model. This model is more difficult since an infinity of position estimates is possible compared to a limited number of categories in a classification model. Also, the study in a boisterous environment complicates estimation quality compared to a classification method. Environmental noise is a significant obstacle in spatial hearing and sound source identification, mainly when dealing with dynamic sound sources (e.g., cocktail parties). This type of variation might significantly impact the performance of the localization model, resulting in a decrease in accuracy compared to a noise-free environment. Various Signals to Noise Ratios (SNR) are used to covert the studied methods' performance.

1.2 Objectives

The project's main objective is to propose machine learning methods with feature extraction for regression estimation of binaural SSL by manipulating features buried in data collected in a public Head Related Transfer Function (HRTF) database.

Methods based on ML with feature extraction are explored in this thesis to address this challenge. Here is a list of the objectives that this thesis aims to fulfill:

- Perform reviewing of the related works and background study.
- Generate and prepare the appropriate public data sets to conduct the experimental study of the binaural SSL models.
- Propose regression methods based on Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM).
- Propose new feature extraction based on wavelet-scattering applied on LSTM and CNN for classification and regression.
- Compare the performance of binaural SSL in classification and regression in different noisy scenarios: i) diverse types of sound signals, ii) diverse ranges of SNR levels for training, validation and test phases.

1.3 Methodology

The following steps are realized to fulfil the main objective: 1) study the problem definition based on the literature, 2) explore the open datasets and prepare the data scenarios, 3) apply an algorithm based on machine learning without feature extraction, 4) implement feature extraction, 5) explore the performance and improve the SSL accuracy, also study the noisy environment effect and robustness. 6) Finally, compare results with the existing literature method using the same datasets and scenarios [24].

1.4 Contribution of the Study

The main contributions of this thesis are as follows:

- To the author's knowledge, this is one of the first attempts to apply an algorithm based on Long Short-Term Memory methods with feature extraction for binaural SSL.
- Feature extraction based on scattering decomposition using wavelet scattering.
- Explore the performance and improve the binaural SSL accuracy compared with literature reference methods based on the same HRTF and noise dataset and applied in the same scenarios.
- Proposed machine learning based on LSTM and CNN in a regression model with comparative results. Usually, classification models are used in SSL.

1.5 Thesis Outline

The following is the organization of the thesis:

- Chapter 1 introduces the problem of estimating the location of a sound source in various SNR conditions and how machine learning methods are viable techniques to perform this task. The problem is outlined, and the thesis objectives and methodology are discussed.
- Chapter 2 provides a literature review relevant to this thesis. It briefly highlights the literature's methods and techniques to locate the sound source using binaural recordings.

- Chapter 3 details the NARX and LSTM methods and the performance contributions when the proposed scattering decomposition-based feature extraction method is applied for SSL.
- Chapter 4 details the WS-LSTM method and results to estimate the angles in azimuth and elevation for both classification and regression SSL estimation.
- Chapter 5 details the WS-CNN method and results to estimate the angles in azimuth and elevation for both classification and regression SSL estimation.
- Chapter 6 summarizes the results.
- Chapter 7 highlights this thesis's key findings and contributions to this research area and identifies future investigation areas.

Chapter 2: State of the Art of Binaural SSL

The primary objective of the study was to examine human hearing systems and attempt to replicate the human capacity to pinpoint multiple sound sources using only two sensors, the ears [18] [25] [26]. Several systems use only two sensors to achieve great localization accuracy in the presence of background noises [25] [26]. The significant advancement in binaural SSL is associated with considerable advances in information technology and computer power, particularly machine learning systems for signal processing. This thesis studies SSL with binaural signals, applies a filter to both left and right signals for better feature extraction (e.g., wavelet filter), and then uses a machine learning method to determine the loudspeaker location around a manikin head.

2.1 Background

The location of a single speaker around a human head is described by three components, azimuth θ, elevation φ, and distance r. Fig. 2.1 shows these three components around a head model illustrating the three components.

Binaural hearing is a characteristic of the human auditory system that uses numerous cues collected from both ear signals to offer spatial information about sound sources. Binaural cues created by the signal difference between the ears play an essential part in the localization process.

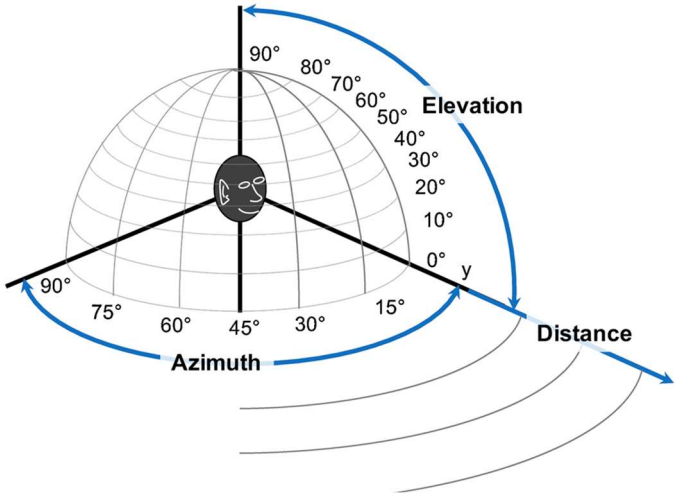


Fig. 2.1 Sound source localization polar coordinates in azimuth, elevation, and distance [3]

Two primary cues are involved in estimating the SSL coming from one specific location, such as a loudspeaker, the interaural time difference (ITD) and the interaural level difference (ILD) [21]. ITD is a time value describing the time difference for a single sound reaching the right and left ear. This interaural time difference Δt is identified in Fig. 2.2. The brain can partially determine where a sound is coming from by using this information. Equation (1) describes the ITD where ψ_L and ψ_R refers to distributed phase of the left ear and the right ear, respectively.

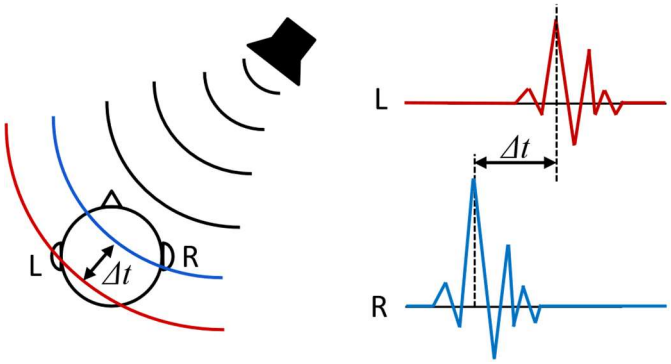


Fig. 2.2 Interaural Time Difference (ITD)

$$ITD = \frac{\Delta\psi}{2\pi f} = \frac{\psi_L - \psi_R}{2\pi f} \quad (1)$$

ILD represents the level difference of a single source signal between two ears, ΔI. The ΔI can be visualized in Fig. 2.3. This difference in power perceived at the ears, combined with the ITD, allows the brain to determine the origin of a sound. Equation (2) describes the ITD proportional aspect of the sound pressure between the two ears, where H_L and H_R represent the left and right ear HRTFs, respectively [27].

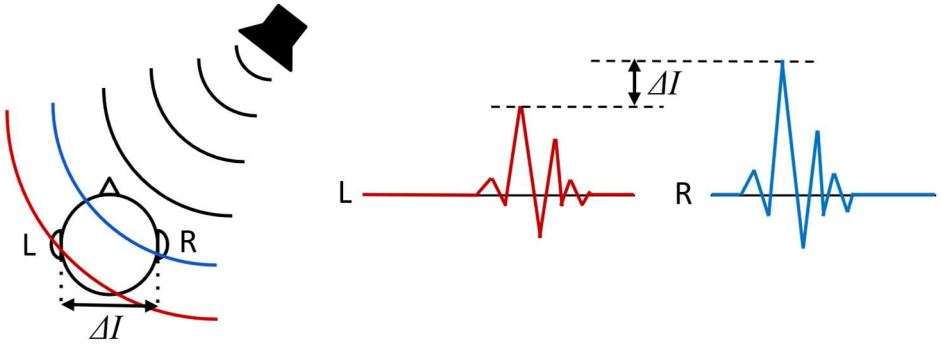


Fig. 2.3 Interaural Level Differences (ILD)

$$ILD = 20\log_{10}\frac{H_R}{H_L}\ dB \quad (2)$$

Many approaches in the literature rely solely on ITD and ILD cues retrieved from a sound recording to estimate the SSL [16] [28]. Although sound sources at different locations might produce the same ITDs, these sources are located inside what is known as the cone of confusion [29] [30]. Due to the cone for confusion phenomenon, elevation localization has received little attention since typical interaural distinctions are insufficient for elevation localization. Studies only concern the azimuth component estimation of the sound source. However, elevation estimation is crucial for several binaural SSL applications, such as

speech enhancement [31] and speaker separation [32]. In the case of human-robot interaction, Fig. 1.1, the elevation of the human speaking is usually higher than the elevation of the robot since most robots do not have a similar height as humans. In this case, the robot must be able to localize the human speaker in both azimuth and elevation components. Due to the impact of the cone of confusion phenomenon on the ability to differentiate SSL in elevation, some additional cues for elevation estimation have been proposed in previous techniques, such as spectral cues [21], head-related transfer function (HRTF) [33], and interaural matching filters [34].

When a sound is received at the eardrums, the diffraction by the torso, the head and the outer ear causes a modification of the propagation and the perception of the sound compared to the initial sound at the source. The HRTF represents the transfer function between the sound pressure at the entrance of the obstructed ear and the sound pressure at the center of the head when the listener is absent.

The following equations (3) determine both ears' left and right HRTFs (H_L and H_R).

$$H_L(r,\theta,\varphi,f,a) = \frac{P_L(r,\theta,\varphi,f,a)}{P_0(r,f)} \text{ and } H_R(r,\theta,\varphi,f,a) = \frac{P_R(r,\theta,\varphi,f,a)}{P_0(r,f)} \quad (3)$$

As stated previously, SSL can be described using a three spherical coordinate system (r, θ, φ), Fig. 2.1. P_L and P_R are the sound pressure for the left and right ears in the frequency domain. P_0 is also a sound pressure in the frequency domain, but when the head is absent. r is the distance between the head center and the sound source, θ the azimuth angle, typically between -180 and +180 degrees, and φ is the elevation angle between -90 and +90

degrees. Parameter a includes the anatomic dimensions of the subject head, such as the position and size of the ears and the head radius.

HRTFs can be captured in anechoic environments as the Fourier transform (FT) of the head-related impulse response (HRIR). HRTF is the binaural impulse response from a given source position in the time domain [28]. HRTFs have several physical qualities linked to frequency and time domain properties and measured HRTFs may be used to assess various localization cues [35]. The HRTFs are different for each subject and depend on the physiological structures. Two individuals will have different HRTFs even if the sound and the loudspeaker location are the same.

To use the binaural recording from two microphones present at the ears for a sound source at a particular location, as the raw data of an SSL method, a dataset of simulated recordings can be created from any acoustic signal and the suitable HRTF dataset specific to that location.

$$s(t) * F_L(t) \text{ and } s(t) * F_R(t) \tag{4}$$

According to equation (4), source signal $s(t)$, Fig. 2.4, is convolved (*) with the HRIR filters of the left ear (F_L) and the HRIR filter of the right ear (F_R), Fig. 2.5, for each loudspeaker position. The result of these convolutions are two soundtracks, which represent the signals in the time domain that would be recorded in the ear using a microphone, Fig. 2.6. It can be noted that if those signals are played through headphones in the form of a stereo soundtrack, the listener would perceive the sound as coming from the location used during the HRTF recording.

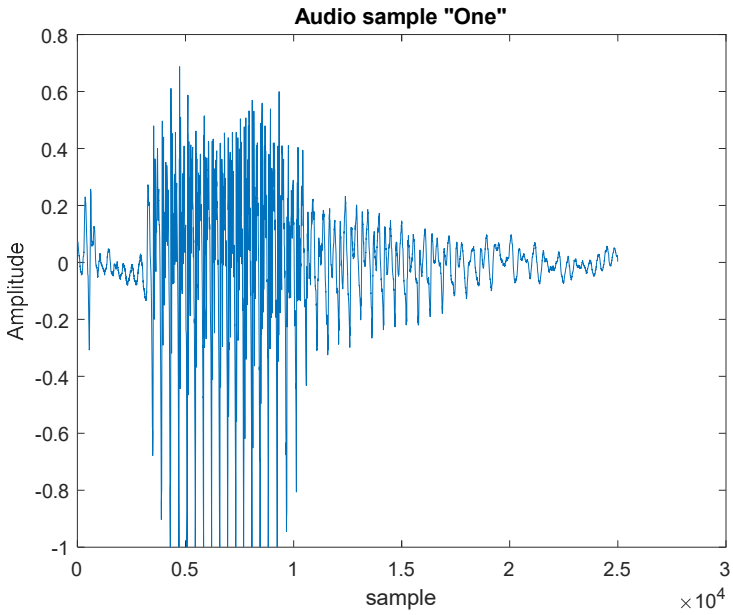


Fig. 2.4 Sample of the source signal, *s*(*t*)

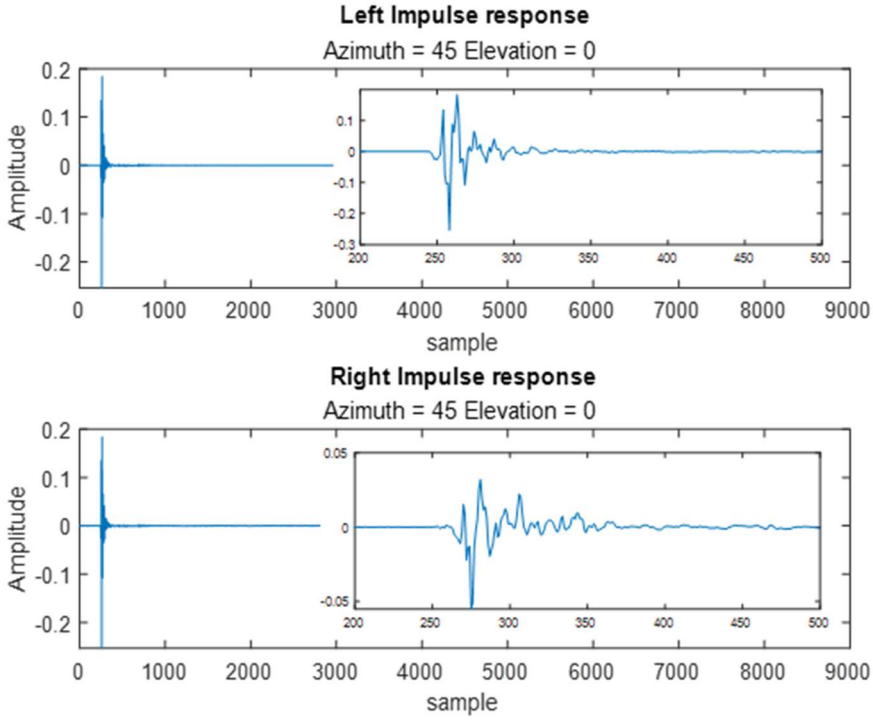


Fig. 2.5 Impulse response for a) Left ear and b) Right ear

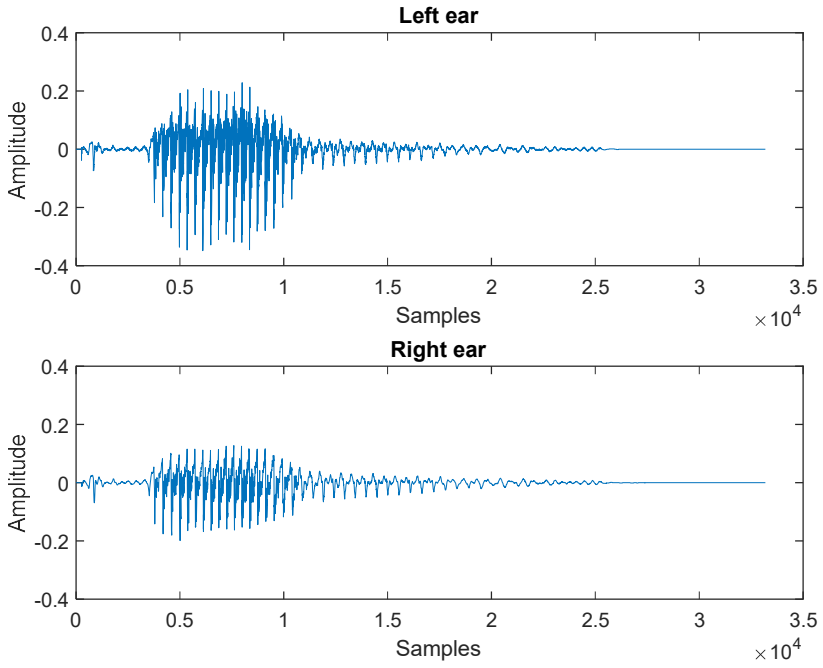


Fig. 2.6 Signal at ear entrance a) Left ear, b) Right ear

Multiple HRTF databases are available to be used for this type of research [36] [37] [38]. Each database available has its advantages and disadvantages, such as the number of speaker positions, the signal precision, the data filters applied to the signals and the human subject or manikin used for the recording. Since every subject head generates a different HRTF for the exact speaker location, measurements of the KEMAR manikin (synthetic head) are generally available in HRTF datasets [39]. These standard measurements are recorded with a single loudspeaker at different interaural-polar azimuths and different interaural-polar elevations, producing considerable data information considering the diversity and length of sound sequences.

2.2 Sound Source Localisation Methods

Many algorithms based on HRTF are found to resolve this well-known challenge of binaural localization for direction-of-arrival (DOA) estimation [1] [19] [40] [24]. Some methods have been developed based on artificial intelligence and machine learning to estimate the SSL [17] [18] [40] [24] [41] [42] [43] [44]. We can list many algorithm-based methods on the HRTF applied to binaural SSL.

From an HRTF database of multiple recordings of sound at both ears of human models with a single loudspeaker placed at different locations around the head, a dataset of recordings of the sound filtered by HRIRs can be created. The common idea of most algorithms to perform well is the nonlinear and dynamic model, like the ear canal. Methods based on the hidden Markov model (HMM) are proposed to track sources [26]. Data from HRTF-based models for SSL has been used in neural network systems [40] [45]. Artificial intelligence-based methods use sounds picked up in the ear canal of both ears as input to estimate the SSL around a human head using features buried in the HRTF.

Fig. 2.7 shows a flowchart of the methodology used to estimate the orientation in azimuth of a sound source using a method based on machine learning. First, it is necessary to separate the dataset of signals modified by the HRIR into two parts, one of which will be reserved for the learning step of the NN and the other for the validation and test, which consists of an evaluation of the performance of the model. The data is passed through an optional feature extraction algorithm which aims to highlight specific information deemed more relevant for the NN. Initially, the NN is in a learning phase to adapt the weights in

the layers to be later used with the test dataset to evaluate the model's performance. The output of the NN is a scalar value indicating the approximate azimuth orientation of the sound source in the case of Fig. 2.7.

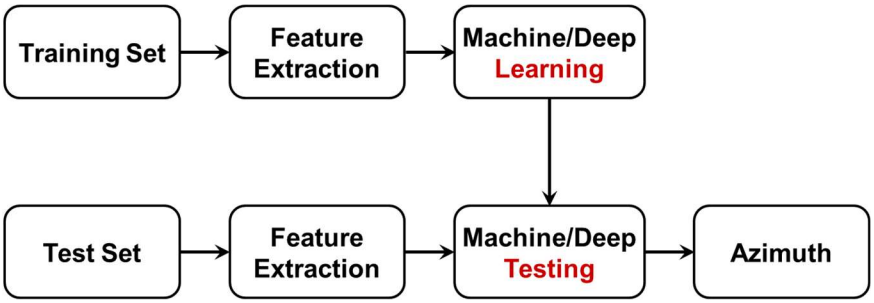


Fig. 2.7 The flowchart of the machine learning

Many NN types exist and have been applied to the SSL challenge; one helpful model is the multilayer neural network (MNN), Fig. 2.8, with frequency filters as input, which is based on the information contained in the relation between the HRTF and localization of the sound source.

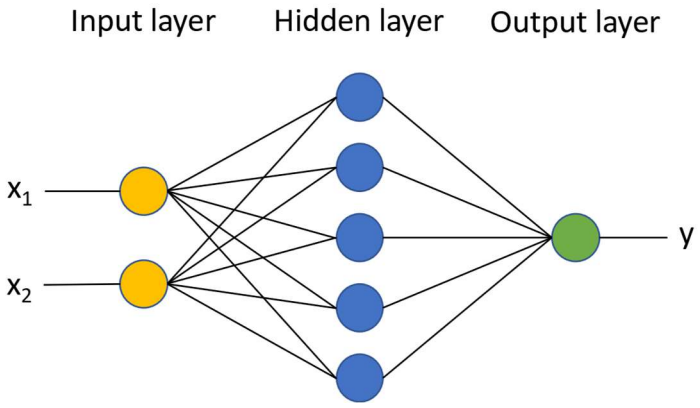


Fig. 2.8 Traditional Neural Network model

A derivative version of the traditional NN is the Deep Neural Network (DNN) [26], Fig. 2.9. The main difference between the traditional NN and DNN is that DNN has many

hidden layers of neurons. As a result, DNN generally requires massive input data to be adequately trained. It is also true that DNN is more computationally intensive due to the weight of input data to adapt during training.

Few papers on NN using LSTM have been published for SSL [46], and no literature has been found concerning binaural SSL. This proposed method is detailed in Chapter 3.

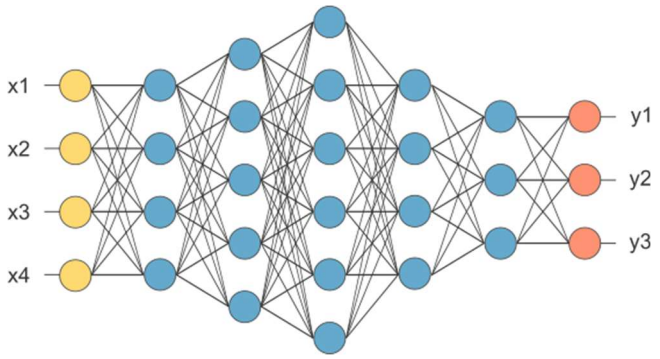


Fig. 2.9 Deep Neural Network model [26]

Another well-known machine learning method is the Convolutional Neural Network (CNN), Fig. 2.10. It generally uses a matrix data form as input data. It consists of multiple layers, such as convolutional layers, pooling layers, activation functions, and fully connected layers.

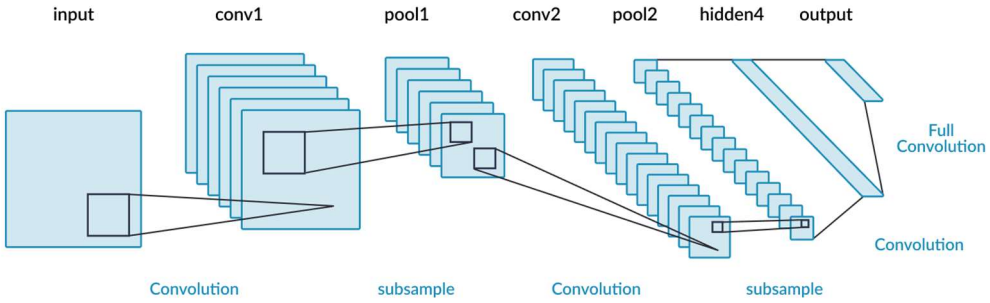


Fig. 2.10 Convolutional Neural Network [3]

Subsequently, the results obtained in this thesis will be compared with those obtained from a reference method based on a time-frequency convolutional neural network (TF-CNN) with multitask learning [24].

A neural network rather more inspired by the biological system has also been developed and studied in the literature, the Spiking Neural Network (SNN) [17] [18] [42] [43]. It is based on the principle of spiking found in the brain's neurons, which enables the processing of information from the sensory system shown in Fig. 2.11. SNN has been used as a learning method to mimic the ears localizing a sound [18] [17] [42]. Furthermore, filter banks can be added as feature extraction before the SNN to improve the quality of localization and the imitation of the ear canal.

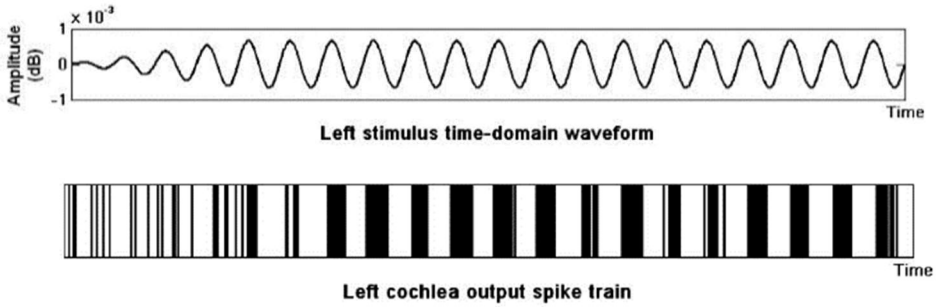


Fig. 2.11 Spiking model [45]

2.3 Feature Extractions Methods

Many SSL published works implement feature extraction techniques [2] [10] [16] [21] [30]. Applying feature extraction methods to improve sound location estimation has the purpose, among other things, of highlighting specific essential signal characteristics for a particular sound location which could help differentiate it from other locations. This step

is added before processing signals in the NN, as indicated in red in Fig. 2.12. There are many feature extraction methods, some more inspired by the biological auditory system than others [16] [17] [45]. These feature techniques have the advantage of having frequency bands closer to the response of human hearing systems than conventional equidistant frequency bands. Feature extraction methods also reduce computational complexity and, thus, computation times since the data presented to neural networks are often shorter and less complex.

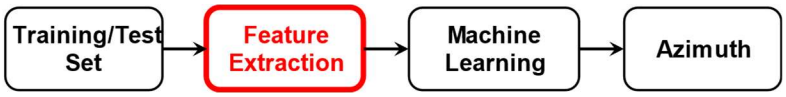


Fig. 2.12 The flowchart of the feature extraction methods

2.4 Time-frequency Convolutional Neural Network

Li's team published many exciting works on SSL classification methods, including feature extraction, followed by CNN [24] [41]. This section explains one of the most recent methods based on ML SSL classification, Time-Frequency CNN (TF-CNN). The proposed methods are compared to the TF-CNN method. Fig. 2.13 shows the proposed method's flowchart based on time-frequency feature extraction CNN.

The TF-CNN method can be split into three main components, the time-frequency feature extraction, the TF-CNN and the multitask neural network. The first component, the time-frequency feature extraction, aims to highlight the localization cues forwarded as input to the TF-CNN. Extracted localization cues from this component are the IPD/ITD and the ILD, obtained by applying Short-time Fourier Transform (STFT) to the binaural signal

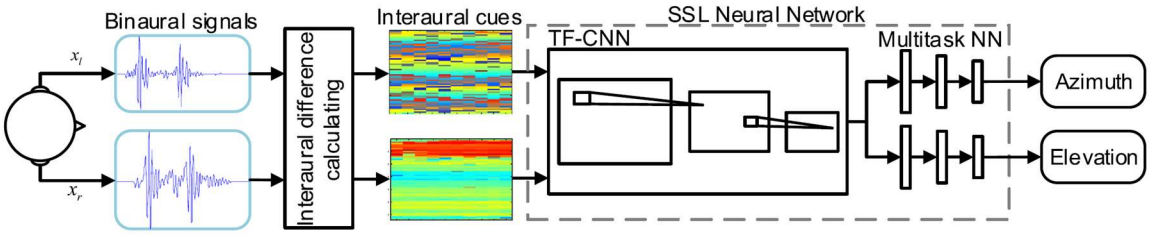


Fig. 2.13 Flowchart of the TF-CNN SSL system. Time-frequency interaural cues, i.e., IPD and ILD, are extracted as localization cues. SSL method consists of TF-CNN and multitasks neural network. [24]

generated with the HRTF. Equation (5) allows the extraction of Interaural Phase Difference (IPD) φ at the *k-th* audio frame and ω*-th* frequency bin from the STFT transformed Y_l and Y_r.

$$\phi(k,\omega) = \angle \frac{Y_r(k,\omega)}{Y_l(k,\omega)} \tag{5}$$

Equation (6) allows the extraction of ILD θ at the *k-th* audio frame and ω*-th* frequency bin from the STFT transformed Y_l and Y_r.

$$\theta(k,\omega) = 20\, log_{10} \frac{|Y_r(k,\omega)|}{|Y_l(k,\omega)|} \tag{6}$$

The retrieved IPD and ILD are stacked independently across many frames and frequencies into IPD and ILD matrices, $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$, of fixed size, (7) and (8), which are input features for the CNN.

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi(1,1) & \phi(1,2) & \cdots & \phi(1,F) \\ \phi(2,1) & \phi(2,2) & \cdots & \phi(2,F) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(K,1) & \phi(K,2) & \cdots & \phi(K,F) \end{bmatrix} \tag{7}$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta(1,1) & \theta(1,2) & \cdots & \theta(1,F) \\ \theta(2,1) & \theta(2,2) & \cdots & \theta(2,F) \\ \vdots & \vdots & \ddots & \vdots \\ \theta(K,1) & \theta(K,2) & \cdots & \theta(K,F) \end{bmatrix} \tag{8}$$

Fig. 2.14 shows a sample of those two matrices distributed as a function of azimuth and elevation. We obtained the same results as [24] to confirm the proper construction of features. Fig. 2.14a and Fig. 2.14b are the IPD and ILD distributions at a 0-degree elevation, and Fig. 2.14c, and Fig. 2.14d are the IPD and ILD distributions from a sound source at an azimuth angle of 40 degrees. The sound source type used to create Fig. 2.14 is a Dirac function without noise since the focus is only on the information buried in the HRTF. In Fig. 2.14a and Fig. 2.14b, it can be noted that the distribution of the frequency intensity varies according to the azimuth angle. The distribution is also symmetrical and centred at 0 degrees azimuth. This symmetry is due to the similar location of the ears on the human head.

In the second part, the TF-CNN aims to process and integrate obtained time-frequency interaural cues across time and frequency domains. TF-CNN learns time-frequency information by performing a 2D convolutional operation on an interaural input feature, resulting in a discriminative shared feature for subsequent multitask SSL.

The CNN architecture layers are Fig. 2.15.

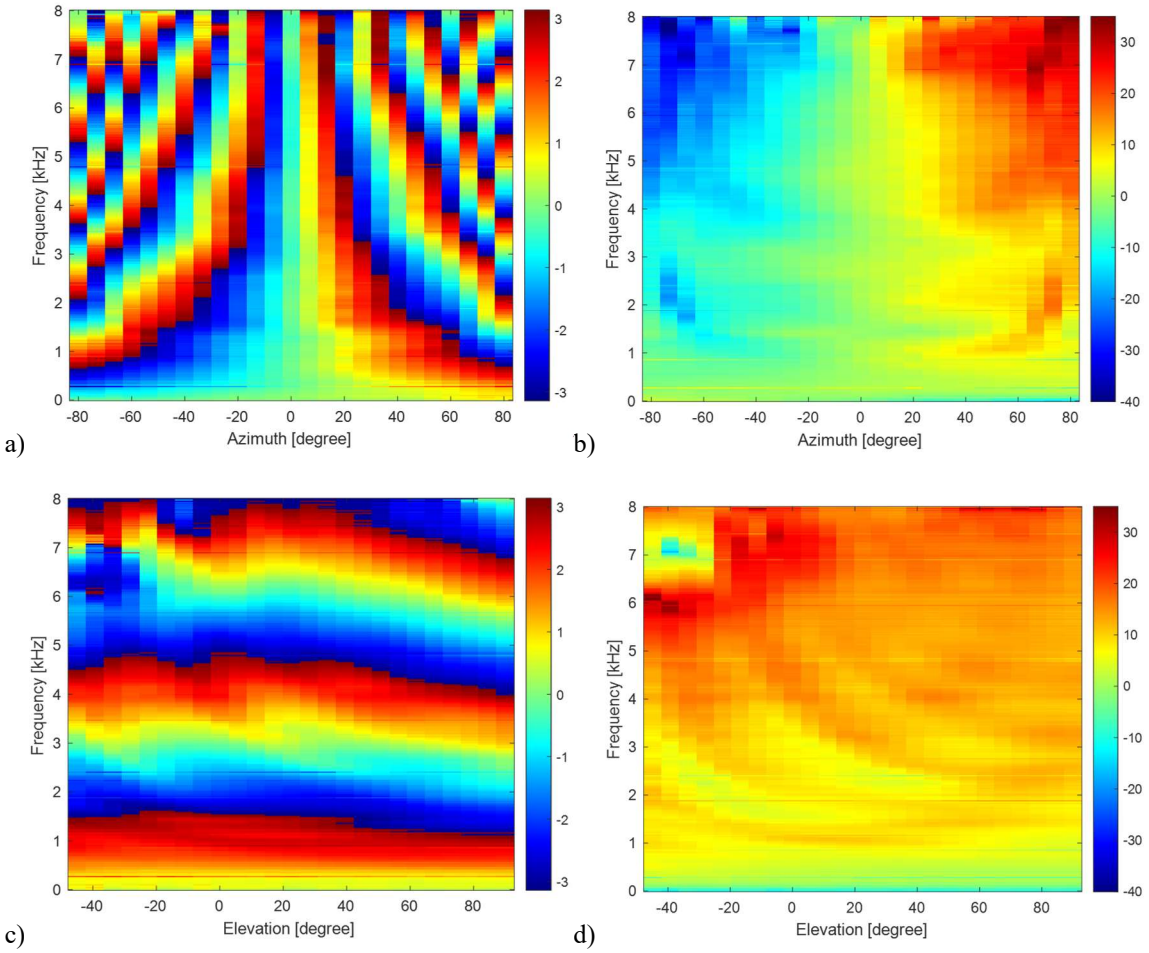


Fig. 2.14 IPD distribution (a) and ILD distribution (b) versus azimuth where elevation is 0◦, IPD distribution (c) and ILD distribution (d) versus elevation where azimuth is 40◦.

The TF-CNN consists of four convolution layers with different numbers of filters and four batch normalization layers followed by Rectified Linear Unit (ReLU) activation layers. The output of the TF-CNN is named Shared features. As the third part of the method, a multitask NN is designed as four fully connected (FC) layers after the last ReLU. The multitask NN aims to estimate the azimuth and elevation of the sound source from the shared features at the output of the TF-CNN. The final FC layer has the same number of

1	"	Image Input	18×320×1 images with 'zerocenter' normalization
2	"	Convolution	32 5×5 convolutions with stride [3 3] and padding 'same'
3	"	Batch Normalization	Batch normalization
4	"	ReLU	ReLU
5	"	Convolution	64 3×3 convolutions with stride [2 2] and padding 'same'
6	"	Batch Normalization	Batch normalization
7	"	ReLU	ReLU
8	"	Convolution	96 3×3 convolutions with stride [2 2] and padding 'same'
9	"	Batch Normalization	Batch normalization
10	"	ReLU	ReLU
11	"	Convolution	128 3×3 convolutions with stride [2 2] and padding 'same'
12	"	Batch Normalization	Batch normalization
13	"	ReLU	ReLU
14	"	Fully Connected	1024 fully connected layer
15	"	Fully Connected	512 fully connected layer
16	"	Fully Connected	256 fully connected layer
17	"	Fully Connected	25 fully connected layer
18	"	Softmax	softmax
19	"	Classification Output	crossentropyex

Fig. 2.15 The CNN architecture layers of the TF-CNN.

weights as the number of possible azimuths (25) or elevations (25) locations. Using a SoftMax layer, the final azimuth and elevation angles are estimated.

Together, the three components forming the TF-CNN with multitasking showed a reliable solution in SSL for azimuth and elevation localization for the conditions studied by Li's team.

In the TF-CNN method in [24], only one CNN is used simultaneously estimate the azimuth and elevation angles. In the same way that WS-LSTM, two CNNs are proposed to maximize the performance for classification and regression, one for azimuth and one for elevation estimation, as shown in Fig. 2.16. To distinguish from the TF-CNN [24] method using a single CNN, this proposal will be named TF-2CNN. Thus, the comparison between the proposals and the feature extraction TF from [25] will be fairer.

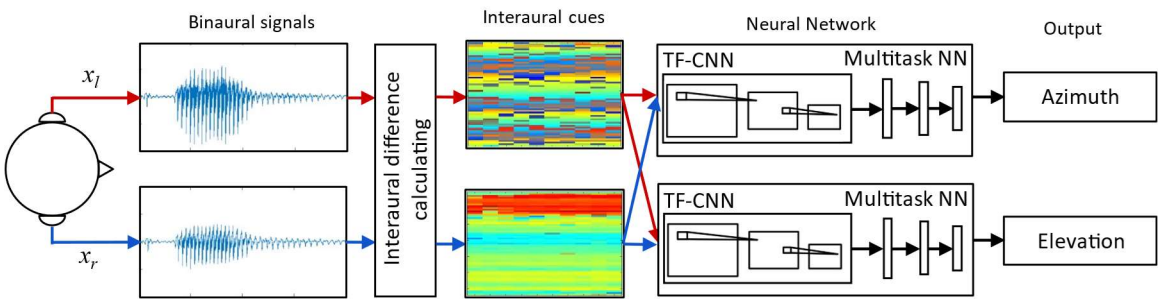


Fig. 2.16 Flowchart of the TF-2CNN SSL system. Time-frequency interaural cues, i.e., IPD and ILD, are extracted as localization cues. SSL method consists of TF-CNN and multitasks neural network

2.5 Regression source localization

In addition to classification, another investigated aspect in this project is the ability of the methods to assess regression for SSL. The regression, within the framework of this project, consists in trying to estimate the position of the loudspeaker when it is located at a position that the model did not encounter during the learning step. A regression situation represents a real-world SSL implementation more than a classification situation since a sound source's possible location is usually continuous and rarely limited to specific locations. In comparison, because the output of classification functions is a discrete number, it cannot forecast the precise value of a continuous variable. In an experimental situation where the goal is to estimate the speaker's location around a head model, all locations are possible, especially for azimuth, which regularly ranges from 0 to 360 degrees. An example of regression using experimental datasets could be estimating the location of a loudspeaker when it is located at 30 degrees in azimuth while the training has been performed only when the azimuth is at 15 and 45 degrees. Generally, it is more complex to perform regression rather than classification.

There has been little research on the regression approach for SSL. The authors in [47] present a CNN with a regression model using the same method of TF-CNN using TF feature extraction followed by one CNN with a regression output layer. The authors used the same azimuth and elevation locations for the train, validation and test phase. A very different definition of regression from the regression presented in this thesis. Their regression is only based on different SNR levels for training and testing. The authors in [48] applied a deep CNN for regression to the azimuth angle of the sound. However, the research was more focussed on a reverberant environment where they used the samples from four random speakers as the training data and the recording from a fifth speaker as the validation and testing data. In addition, no HRTF was used for the dataset; the authors used a sound source from the 2-D correlogram in [49].

2.6 Computing Implementations

The project is entirely simulated and uses experimental opensource pre-recorded datasets. The signal processing, feature extraction, and machine learning methods are implemented on MATLAB R2020b-R2022a. The heavy computation, such as the machine learning training phase, is executed on the *Laboratoire des Signaux et Systèmes Integrés*'s equipment on a computer operating Windows 10 Pro with an Intel Core i9-9900X CPU @3.50 GHz processor with 64 GB of DDR4 RAM. The computer also has two NVIDIA GeForce RTX 2080 Ti graphics cards.

Computing the data obtained from all speaker positions using MATLAB consumes a tremendous amount of memory resources. The massive computation also requires the

Central Processing Unit (CPU) to be powerful enough. Moreover, Graphical Processing Units (GPUs) are optimized to train artificial intelligence and deep learning models to process multiple calculations simultaneously. GPUs have many cores, which allows for the better computation of multiple parallel processes. GPUs are a crucial part of modern computing. Their computing and high-performance networking transform computational science and AI. GPU developments nowadays are contributing to the progress of machine and deep learning.

Furthermore, NVIDIA GPU provides the Compute Unified Device Architecture (CUDA), which is crucial for supporting various machine learning applications. NVIDIA created the CUDA programming interface model as a parallel computing platform. CUDA-enabled GPU allows software developers and engineers to enhance the general performance in data processing. These CUDA cores are highly beneficial and evolutionary in artificial intelligence. Although training times for NN models are high due to the computational complexity, using a high-performance computer can help reduce computational times. In the studied situation, it is also important to note that the computational time is proportional to the number of SSL used for training and testing. The more azimuth and elevation locations used, the longer the computation times.

This chapter presents the state of the art in the field of binaural SSL, emphasizing methods using deep/machine learning. This approach surpasses the traditional methods often based on complex equations attempting to model wave propagation behaviour. CNN-based methods allow us to obtain results based on learning from experimental data. The literature has been growing on the subject in recent years, and very few proposed methods emphasize

regression, the majority present results in classification. Moreover, we did not find any proposal based on the LSTM method for binaural SSL. Furthermore, few works on LSTM are applied for SSL (i.e., [46]) but none on binaural SSL. Li's team has carried out several works for several years on SSL. We were able to repeat the TF-CNN method [24], which will serve as a comparison of the proposals in the work of this thesis. Implementing the TF-2CNN method will make it possible to compare under the same conditions of simulations and be more reliable. In addition, the research [24] on TF-CNN focuses only on classification. We can also apply it in regression mode in the following chapters.

Chapter 3: NARX and LSTM Methods with Wavelet Scattering Feature Extraction

From a binaural recording, advanced 3D spatial audio positioning would not need more information to determine a sound source position, as can be done by a human. We find many algorithms based on HRTF to resolve this well-known challenge of binaural localization for direction-of-arrival (DOA) estimation [3] [40] [30]. In this Chapter, machine learning methods (NARX, LSTM) are implemented to determine the speaker location around the head using HRTF signals. From the IRCAM LISTEN HRTF database [38] of multiple recordings of sound at both ears of a human model with a single speaker placed at different locations around the head, a dataset of recordings of sounds that is filtered by HRIRs was created. The goal is to determine the SSL based on the extraction of location-specific synchrony patterns and to train a supervised algorithm to learn the mapping between synchrony patterns and locations from the set of example sounds. The expected output of the system is the localization in azimuth and elevation of a sound by providing the system with a binaural audio record of the sound.

3.1 Machine learning methods

In this Chapter, two artificial intelligence methods based on NNs are studied to estimate the location of a sound source. The first method is based on an adaptive nonlinear filter with an autoregressive structure called NARX (Nonlinear autoregressive network with exogenous inputs), which is part of the automatic machine family. The second is a machine

learning method based on LSTM NN [50] [51]. These two methods are explained in this Chapter.

3.1.1 Nonlinear autoregressive network with exogenous inputs (NARX) – Principle

The structure of the NARX model is shown in Fig. 3.1. NARX is a nonlinear filter structure with an input signal $x(n)$ and a time delay line (TDL), $\{x(n)\ x(n-1)\ \ldots\ x(n-Ndx)\}$ as well as a second input signal $y(n)$ corresponding to a return of the output signal delayed by a certain number of TDLs $\{y(n-1), y(n-2)\ \ldots\ y(n-Ndy)\}$. Note that for source localization, no feedback is used (*Ndy*=0). In this model, the number of delays *Ndx* is fixed for the input signal $y(n)$. The NARX model can be represented by (9).

$$y(n) = \mathrm{f}(y(n-1), y(n-2)\ \ldots\ y(n-Ndy), x(n)\ x(n-1)\ \ldots\ x(n-Ndx)) \quad (9)$$

Fig. 3.2 shows an example of a NARMAX model with *Ndx* = 2 on the input $x(n)$. It is a 3 neurons hidden layer NN, followed by a single-neuron output layer.

3.1.2 Long short-term memory (LSTM) - Principle

Applying an LSTM NN to the SSL problem is suggested for future study in literature reference:

> "An LSTM network […] is well suited for the analysis of time series data and may be more successful than applying static neural networks to time-averaged data." [17]

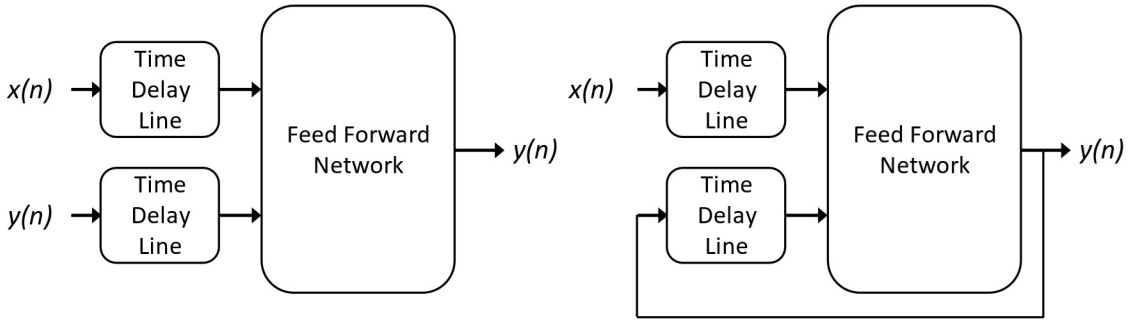


Fig. 3.1 a) NARX model and b) NARMAX

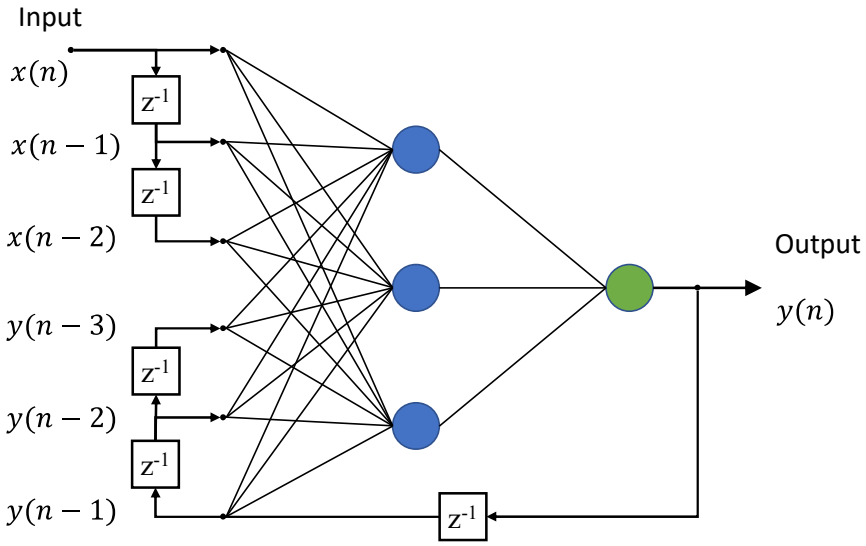


Fig. 3.2 NARMAX model for the case *Ndx* = 2 and *Ndy* = 3

The LSTM network model is a recurrent neural network that contains loops, which allow information to persist on the NN [50]. The recursive phenomenon of loops can be visualized in Fig. 3.3 with the input $x(n)$ and the output $h(n)$.

The LSTM unit, identified as *Unit* in Fig. 3.3, is illustrated in Fig. 3.4 at time step *t.* The graphic shows how the gates remember, update and output the cell (c_t) and hidden states (h_t).

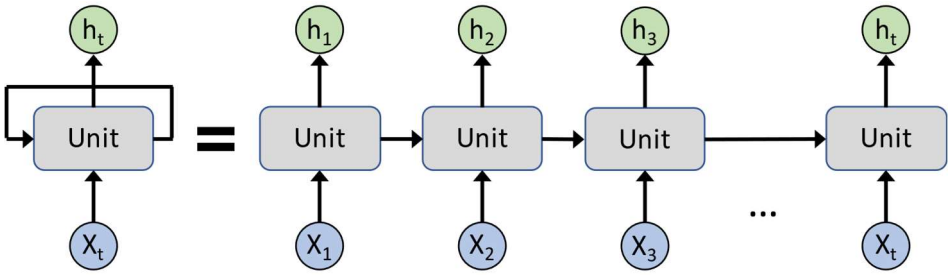


Fig. 3.3 Unrolled recurrent neural networks

The elements in Fig. 3.4 can also be expressed by their mathematical equations (10-13), where i_t is the input gate, f_t the forget gate, g_t the cell candidate and o_t the output gate. σ_c denotes the hyperbolic tangent function (tanh) to compute the state activation function, and σ_g represents the gate activation function. W, R and b are the input weights, the recurrent weights and the bias weights, respectively. [50]

$$\boldsymbol{i_t} = \boldsymbol{\sigma_g}(\boldsymbol{W_i}\mathrm{x_t} + \mathbf{R_i}\mathrm{h_t} - \mathbf{1} + \boldsymbol{b_i}) \tag{10}$$

$$\boldsymbol{f_t} = \boldsymbol{\sigma_g}(\boldsymbol{W_g}\mathrm{x_t} + \mathbf{R_f}\mathrm{h_t} - \mathbf{1} + \boldsymbol{b_f}) \tag{11}$$

$$\boldsymbol{g_t} = \boldsymbol{\sigma_c}(\boldsymbol{W_g}\mathrm{x_t} + \mathbf{R_g}\mathrm{h_t} - \mathbf{1} + \boldsymbol{b_g}) \tag{12}$$

$$\boldsymbol{o_t} = \boldsymbol{\sigma_g}(\boldsymbol{W_o}\mathrm{x_t} + \mathbf{R_o}\mathrm{h_t} - \mathbf{1} + \boldsymbol{b_o}) \tag{13}$$

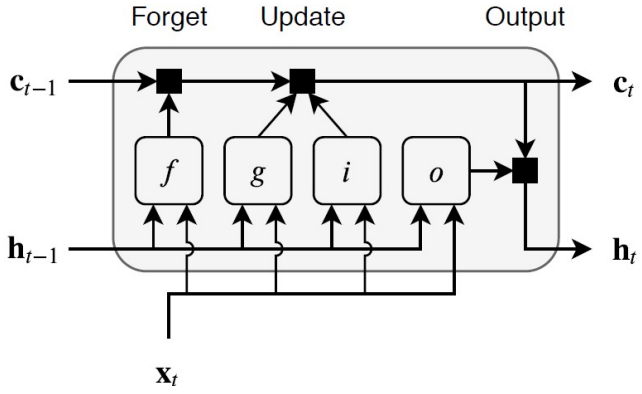


Fig. 3.4 Unit of an LSTM network [52]

The architecture of a general LSTM network for classification is depicted in Fig. 3.5. The network begins with a sequence input layer and progresses to an LSTM layer. The network concludes with a fully connected layer, a SoftMax layer, and a classification output layer to predict class labels.

Fig. 3.5 General LSTM classification flowchart

In addition to classification capabilities, an LSTM network can also be used for regression challenges, as depicted in Fig. 3.6. The network begins with a sequence input layer and progresses to an LSTM layer. A fully connected layer and a regression are output layers to complete the network.

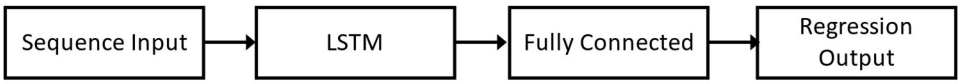


Fig. 3.6 General LSTM regression flowchart

An advantage of LSTM over standard RNN is that it reduces the problem of the gradient disappearing despite its more complex implementation [50]. Indeed, a standard RNN has the drawback of having a short memory and losing information as new information is presented, as shown in Fig. 3.7. Unlike the standard RNN, an LSTM network has a unit that controls the flow of information and controls what is forgotten, remembered, or sent out [51].

A disadvantage of an LSTM model is that the mathematical implementation is more complex than with standard RNNs like the NARX. The complexity could make applying a model based on an LSTM network difficult.

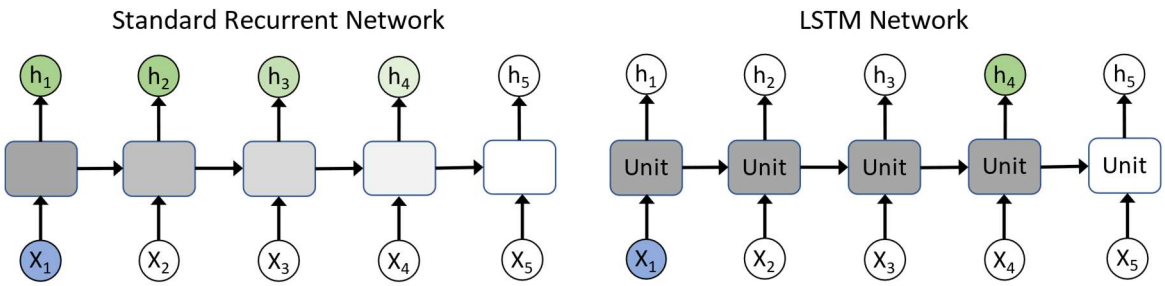


Fig. 3.7 Gradient dissipation problem

3.1.3 Head Related Transfer Function Dataset

The open database of HRIR signals used for this Chapiter comes from a recording made by the Room Acoustics Team, IRCAM [38] in an anechoic chamber, Fig. 3.8. It consists of multiple sound recordings in both ears using small microphones inside the ear canals, Fig. 3.9. Those measurements are made on several human models and the KEMAR manikin (synthetic head), Fig. 3.9a. They changed the location of a single speaker by moving it with a crane, Fig. 3.10. They thus made sound recordings filtered by impulse responses linked to the head (HRIR) of the subject. The different speaker locations are grouped in Table 3.1**,** and 187 total speaker locations were made. Recording parameters were based on a logarithmic scan of 8192 points (44100 Hz) and two-channel inputs (left and right ear).

Fig. 3.8 Anechoic chamber for recording [38]

a)  b) 

Fig. 3.9 a) KEMAR manikin, b) Binaural microphone [53]

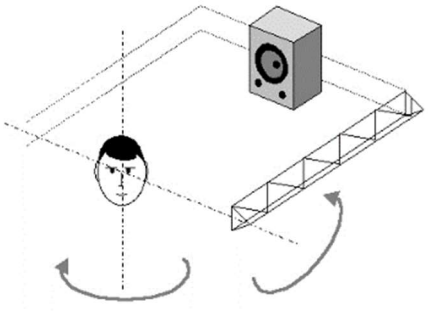


Fig. 3.10 Control of loudspeaker position using a crane [38]

Table 3.1 HRIR measurement points [38]

Elevation (degrees)	Azimuth increment (degrees)	Points per elevation
-45	15	24
-30	15	24
-15	15	24
0	15	24
15	15	24
30	15	24
45	15	24
60	30	12
75	60	6
90	360	1

In this thesis, only measurements on Subject #21, the KEMAR manikin, are used in the data generation and performance evaluation, Fig. 3.9. As mentioned in Chapter 2, this manikin is used in most studies in SSL since it keeps the subject dimensions constant throughout each proposed method. The subject dimensions significantly impact HRTF generated in datasets [39] [53]. Thus, using the same subject will help compare the results.

3.1.4 Binaural signal generation from HRTF

To implement the NN, it is first required to generate a dataset containing the binaural signals, which will then be used in the NNs. The first step in generating this dataset is to create the sound signal that would be played over a single loudspeaker at each of the azimuth locations. According to (4), a mono signal $s(t)$ with a sampling frequency of 44100 Hz is used to generate the signals dataset. The mono signal $s(t)$ can have different shapes, so several possibilities have been tested, including a sine wave, white noise,

uniform noise, recording of the word "one," and music. The $s(t)$ signals used in the following simulation results are detailed for each scenario studied.

3.2 Method Flowchart

Fig. 3.11 shows a flowchart of the methodology used to estimate an SSL in azimuth and elevation. The two binaural signals are generated using an HRTF dataset at a specific speaker location. Those signals are used as input to a neural network to estimate the azimuth coordinate of the loudspeaker specific to this HRTF.

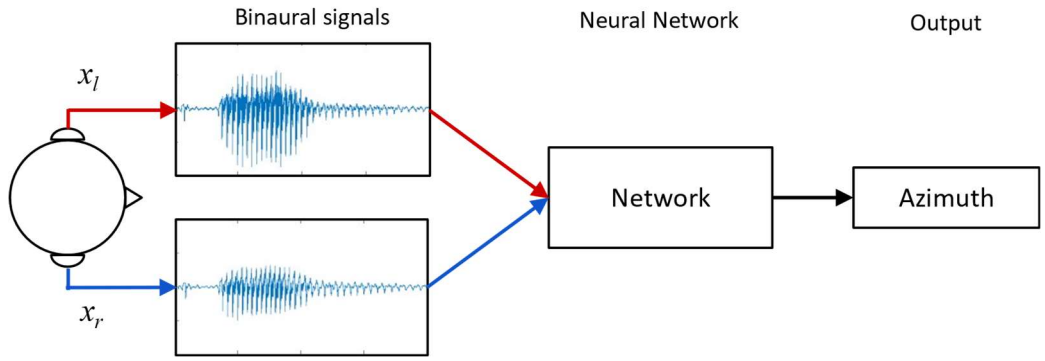


Fig. 3.11 Flowchart of the proposed SSL system based on machine learning without features extraction technique

3.3 NARX and LSTM results without features extraction

This thesis section includes the different simulation results obtained according to each simulation condition. First, a scenario was studied using the NARX model NN, followed by three scenarios with the LSTM model NN in the case without a feature extraction technique. Subsequently, a study with adding a feature extraction method is performed using only the model based on LSTM.

For each scenario, a dataset was created containing different signals. Each dataset includes the training signal *x*, the signal input to the NN during the training phase to realize the model's training. There is also the output signal *y*, which is the actual value of the location of the loudspeaker corresponding to the input signal *x*. This dataset also contains the validation signals *x* and *y*, which are in the same form as the training signals, and they are used to perform the validation during the training phase of the NN. Finally, the dataset contains the generalization signals *x*; the sound signals passed as input to the NN, and the generalization signal *y*, which are the desired outputs of the NN.

The generalization part allows the performance of each method to be evaluated using unknown data in the method. Different graphs were plotted to visualize the performance of each model, and the Relative Root Mean Square Error (RRMSE), defined by (14), was computed by comparing the estimated output $\hat{y}(n)$ to the actual output values $y(n)$.

$$RRMSE = \sqrt{\frac{\sum(y(n)-\hat{y}(n))^2}{\sum y(n)^2}} \tag{14}$$

Where *n* is the time step for all generalization sequences, including a fixed number of azimuth angles, at each angle, the time step number is *Ts*×44100 Hz, with *Ts* the sequence duration at each azimuth angle. In this thesis Chapter, *Ts* = 0.1, 0.2 and 0.4 s. RRMSE performance metrics are used to facilitate the comparison of the results for a different type of source signal; the errors are normalized with respect to the source signals. By dividing by the power of the source signal, performance results for different source signals can be

better compared. Comparing azimuth and elevation angle estimations for different methods and source signals becomes easier.

3.3.1 Nonlinear autoregressive network with exogenous inputs (NARX) – Results

The model based on the NARX NN has been evaluated using several types of signals, including sine waves, white noise, uniform noise, recording the pronunciation of the word "one," and music such as an organ. Different soundtrack duration for each location was tested for each signal played on the loudspeaker, including 0.1 s, 0.2 s and 0.4 s. The parameters of the NARX NN were also adjusted between tests to obtain the best possible estimation performance of the loudspeaker location, including the minimization of the RRMSE. Fig. 3.12a shows the result performance with the different *Mx* representing the number of sample delays used for each ear's signal. It defines the number of LSTM inputs and the number of neurons on the hidden layer. The best obtained RRMSE is 0.217 for *Mx* = 50 and 30 neurones on hidden layers, Fig. 3.12b, which is not very low, in addition to the computation times being rather long, about 7 hours. The studies based on the NARX approach were limited, considering the substantial computational time.

In addition, the results obtained while implementing a model based on the LSTM NN model are more promising, so the emphasis has been on studying this model.

3.3.2 Long short-term memory (LSTM) – Results

The LSTM-based NN has been tested in three scenarios: uniformly distributed random sequences, regression source localization and musical sound (organ).

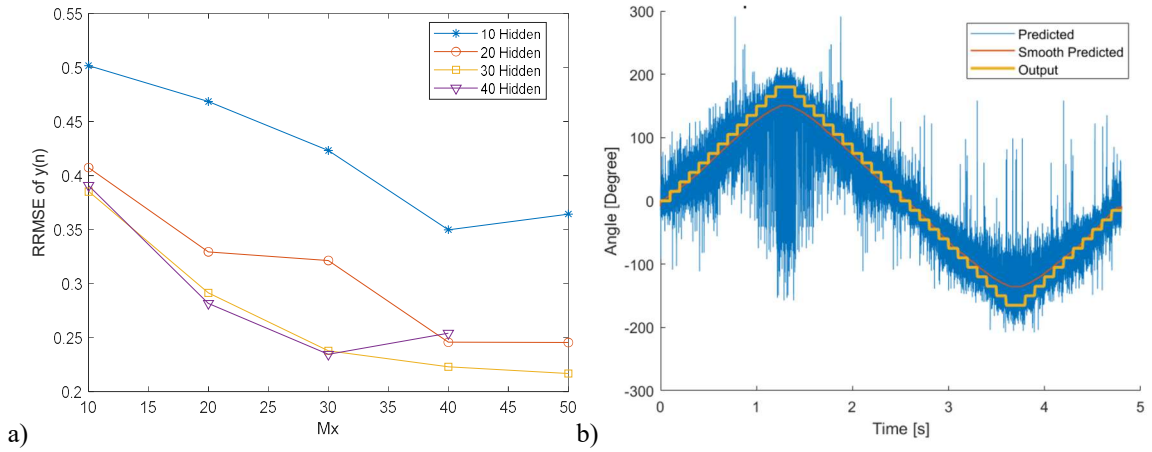


Fig. 3.12 NARX performance results: a) RRMSE results for different NARX sizes for *Mx* and neurons on hidden layers, and b) best results RRMSE=0.217 obtained for *Mx*=50 and 30 neurons on the hidden layer

Scenario #1 – Uniformly distributed random sequences

Scenario 1 uses a different uniformly distributed random sequence for training, validation, and generalization as the sound signal for the loudspeaker. The learning is done on 24 angles, with a constant elevation of 0 degrees and the sequence duration is 0.1 s for each angle. In addition, each of the 24 angles is repeated 4 times, Fig. 3.13a. The corresponding signal for the desired *y* output during training can be seen in Fig. 3.13b. As for the validation, the shape is the same as the training signals. Still, every 24 angles are repeated only two times, Fig. 3.13c and Fig. 3.13d. The generalization was performed on a signal containing all 24 angles with a soundtrack duration of 0.1 s per angle. Each angle was repeated 10 times to ensure the calculation of an RRMSE most representative of the performance, Fig. 3.13e and Fig. 3.13f.

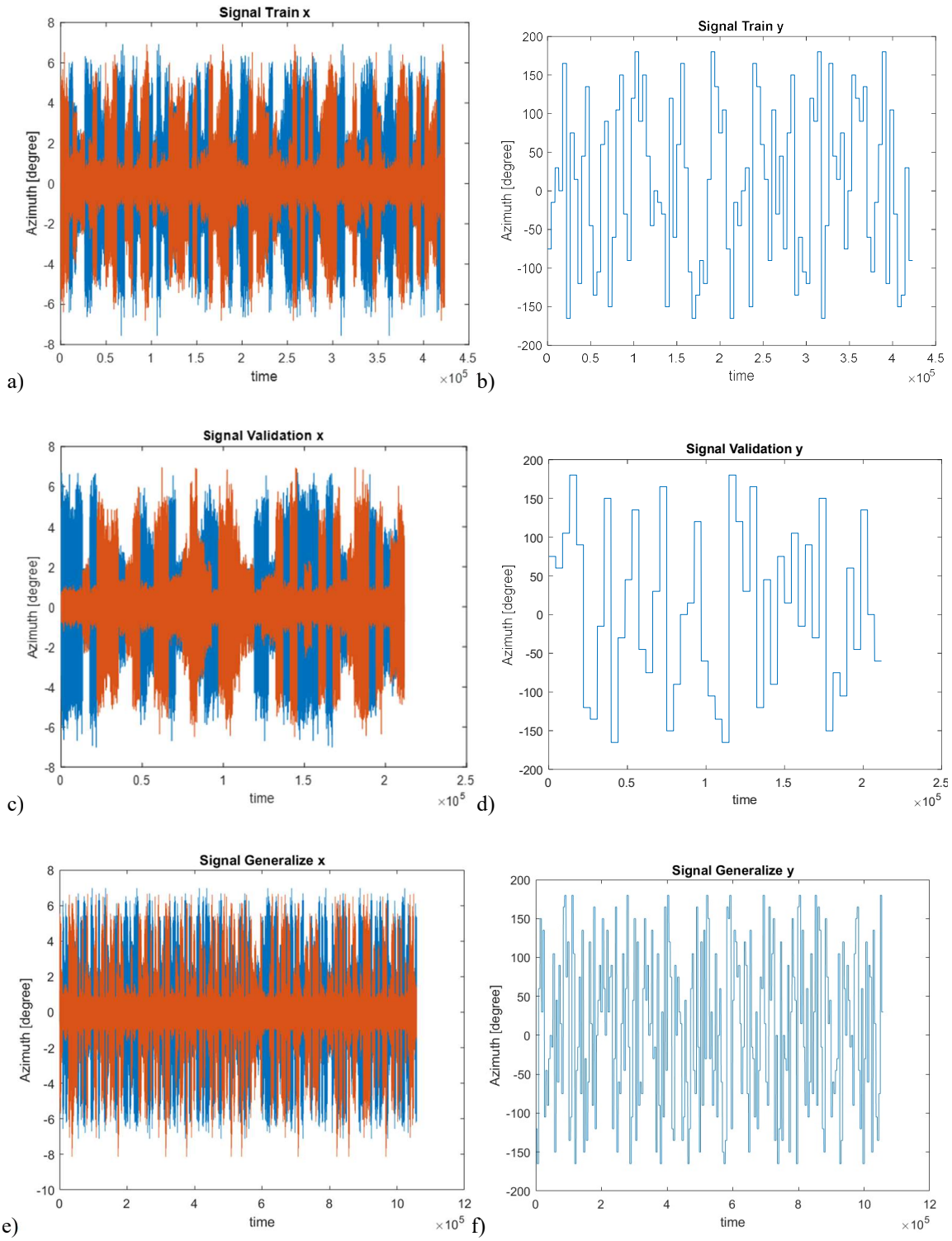


Fig. 3.13 Scenario 1 – a) Training signal *x*, b) Training signal *y*, c) Validation signal *x*, d) Validation signal *y*, e) Generalization signal *x*, f) Generalization signal *y*

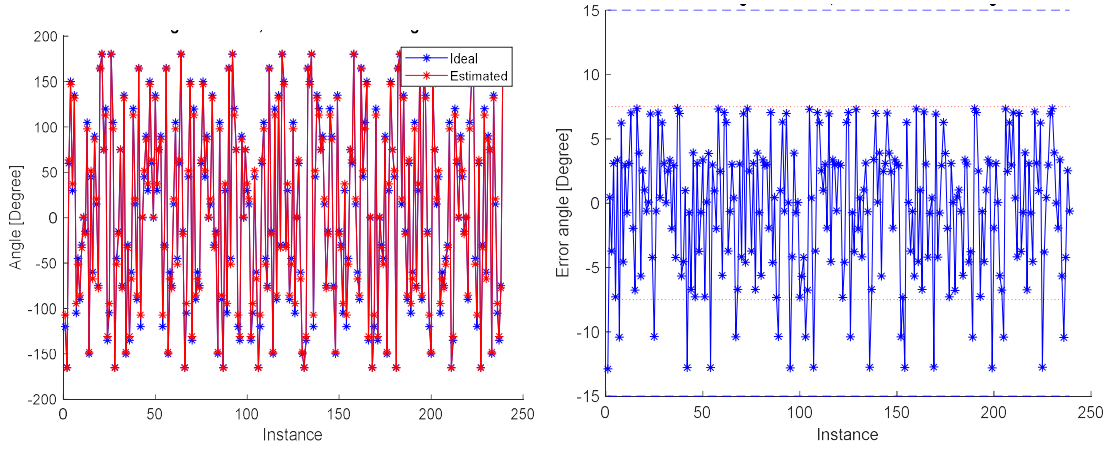


Fig. 3.14 Scenario 1 – Results to estimate 240 angle positions with *Mx*=30 and Hidden Units=40, RRMSE angle = 0.0521 resulting 0/240 (0%) errors superior to ±15° and 20/240 (8%) errors superior to ±7.5°

The LSTM network configuration that gave the lowest RRMSE, 0.052, is with a delay of *Mx* = 30 on the inputs and 40 units on the hidden layer, Fig. 3.14.

Scenario #2 – Regression source localization

Scenario 2 has similarities with Scenario #1, but this time the evaluation is more focused on performance in a regression condition. The signals used for training and validation are still uniformly distributed in random sequences. However, training and validation are done on only 12 of the 24 available angles, each 30-degree step and the tested sequence duration for each loudspeaker location is 0.1 s, 0.2 s and 0.4 s, Fig. 3.15. The generalization was performed on a signal containing all 24 angles, each 15-degree step, with a soundtrack duration of 0.1 s per angle, Fig. 3.15e and Fig. 3.15f.

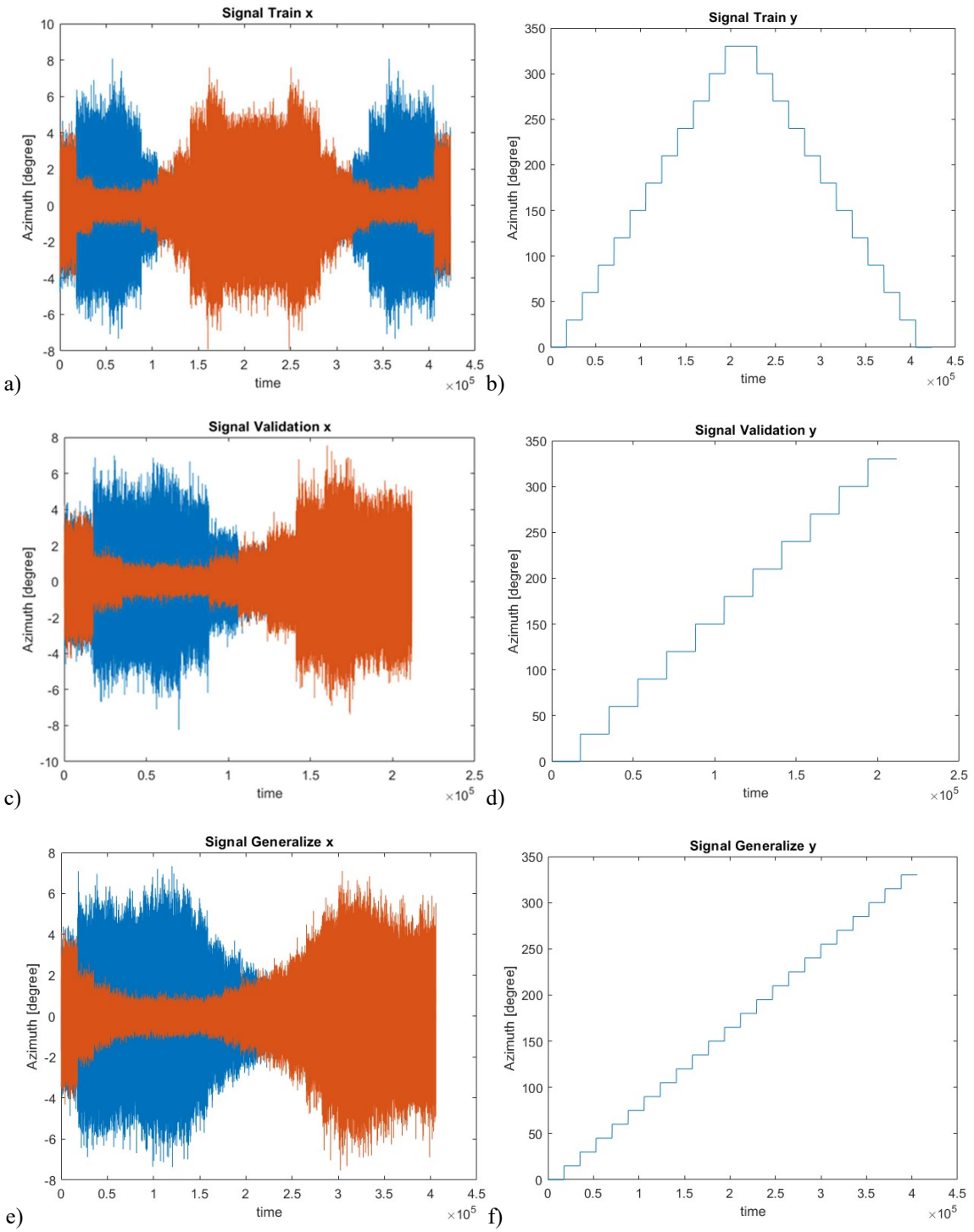


Fig. 3.15 Scenario 2 – Sequence of 0.1 s at 44.1 kHz a) Training signal *x*, b) Training signal *y*, sequence of 0.2 s at 44.1 kHz c) Validation signal *x*, d) Validation signal *y*, and sequence of 0.4s at 44.1 kHz e) Generalization signal *x*, f) Generalization signal *y*

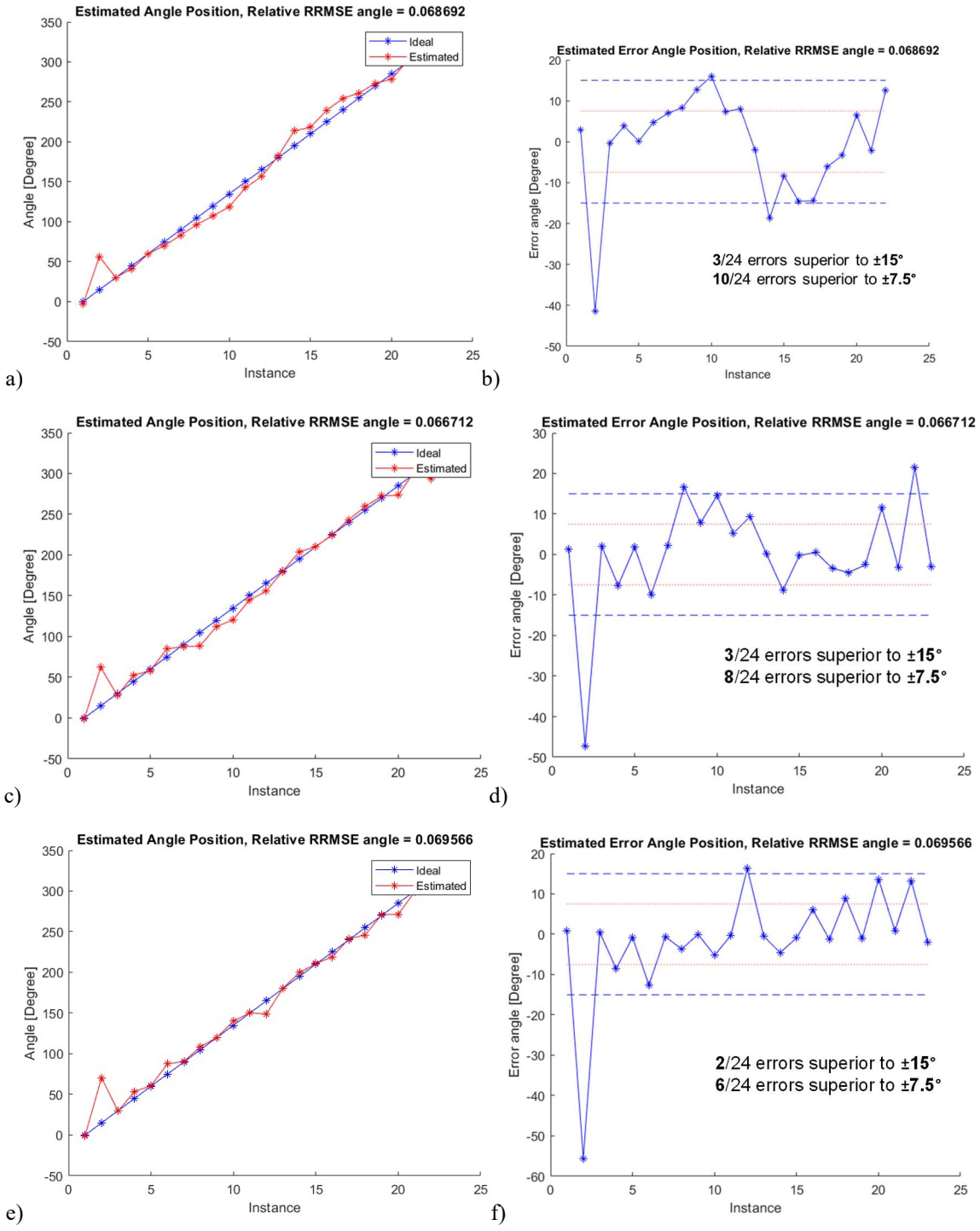


Fig. 3.16 Scenario 2 – Results to estimate 24 angle positions with *Mx*=50 and neuron hidden of 40 for different time step a) b) 0.1 s, c) d) 0.2 s and e) f) 0.4 s

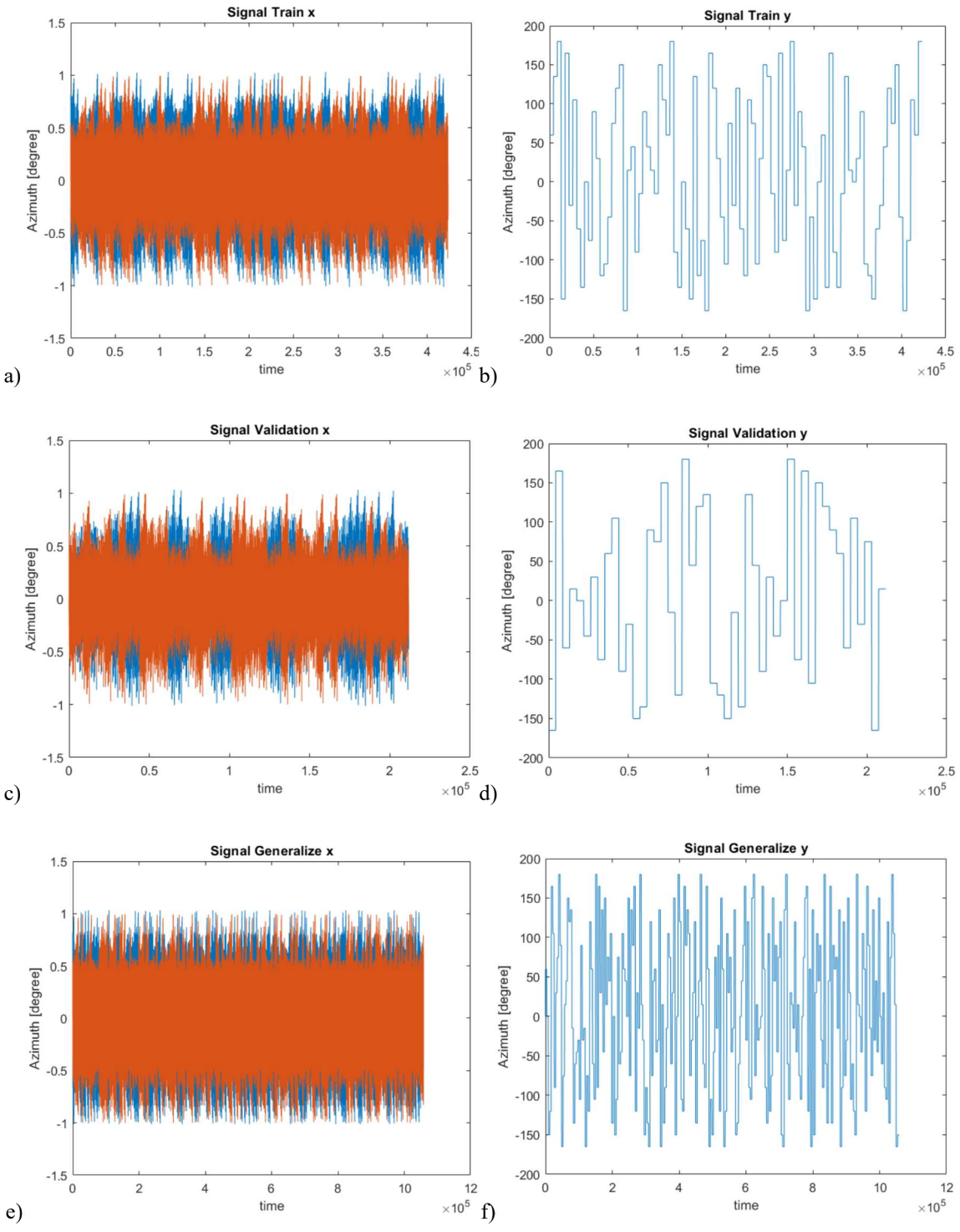


Fig. 3.17 Scenario 3 – a) Training signal *x*, b) Training signal *y*, c) Validation signal *x*, d) Validation signal *y*, e) Generalization signal *x*, f) Generalization signal *y*

The configuration of the LSTM network, which gave the lowest RRMSE for each sound frame duration tested (0.1 s, 0.2 s, 0.4 s), is with a delay *Mx* = 50 on the inputs and 40 units on the hidden layer.

Fig. 3.16 shows the result minimizing the angle errors outside of the limit of ±15 degrees and ±7.5 degrees. The number of angle errors outside the boundary is obtained for estimating the 24 possible locations of the loudspeaker around the head model using sequence duration of 0.1, 0.2, and 0.4 s at each location.

Table 3.2 Estimate angle position in generalization of 240 angles with organ soundtrack depending on training parameters of the LSTM

Computation time (s)	*Mx*	Nb Hidden	RRMSE	Err > ±15°	Err > ±7°
345	50	40	0,4993	20	31
848	60	40	0,1632	15	24
423	70	40	0,1678	21	33
312	80	40	0,1451	9	15
408	50	50	0,1013	9	19
401	**60**	**50**	**0,0828**	**3**	**12**
954	70	50	0,1243	7	14
422	80	50	0,2211	28	37

Scenario #3 – Musical sound (Organ)

Scenario 3 differs from scenarios #1 and #2 in that the signal used during training, validation and generalization is a musical track, more precisely an organ instrument, with a duration of 0.1 s per speaker azimuth angle, Fig. 3.17.

For each of the three scenarios, several parameters for the neural network were executed, intending to obtain the configuration that produces the lowest possible value of RRMSE.

Table 3.2 shows some results obtained when estimating the angle position with the organ soundtrack depending on the training parameters of the LSTM NN. The lowest RRMSE obtained is 0.0828 with $Mx = 60$, and the number of hidden units is 50.

3.4 LSTM method with feature extraction

3.4.1 Feature extraction to improve sound localization

The flowchart in Fig. 3.18 is based on the flowchart in Fig. 3.11 in the previous sections without feature extraction. In Fig. 3.18, the feature extraction is added between the binaural signals and the input of the NN to amplify the binaural cues aiming to increase the SSL performance. The feature extraction method proposed in this thesis is scattering decomposition using wavelet scattering (WS) [54] [55]. The wavelet processing for time-frequency analysis is well known. It is a powerful tool for time-frequency analysis. The wavelet scattering was derived for feature extraction for machine and deep learning applications. One interesting characteristic is the insensitivity to translations of the input signal. WS generates a scattergram according to frequency and time, with each signal recorded at the left and right ears for all possible orientations of the loudspeaker. The scattergram is generated by filtering the input signal using a filter bank, as shown in Fig. 3.19.

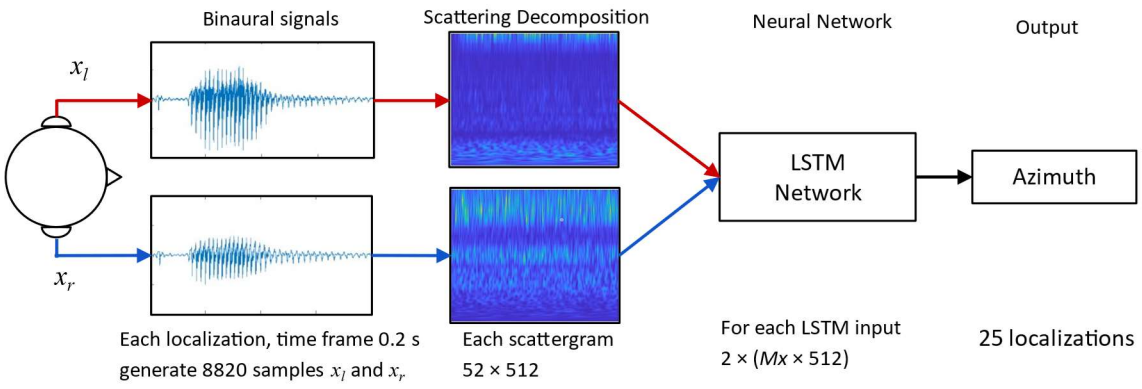


Fig. 3.18 Flowchart of the proposed LSTM SSL system with feature extraction scattering decomposition

Thus, if 24 azimuth orientations are used, 24 scattergrams will be generated, forming a database of 24 images. Each image has a dimension of $X \times Y$, corresponding to time, and frequency, respectively. These images are input to the LSTM NN as a vector for each X_i

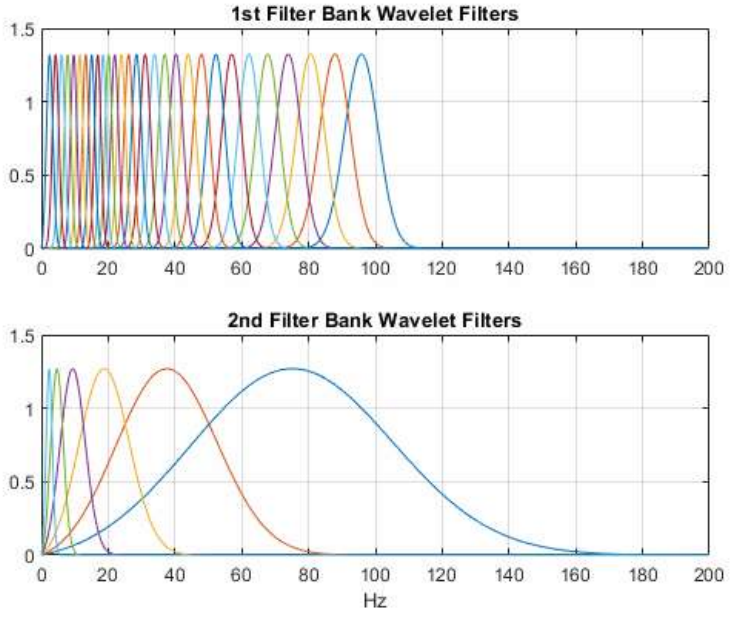


Fig. 3.19 Filter banks example [56]

frequency of a length *Y* time. During generalizations, a new scattergram image is generated and sent to the NN, and the network's output is the estimation of the orientation of the sound source. Fig. 3.20 shows an example of multiple feature extractions based on scattering decomposition for the left and right ears. Fig. 3.20a for a 0-degree azimuth and 0-degree elevation orientation, the loudspeaker is directly in front of the subject face. As expected, the left and right ear scattergram are similar at this location since the loudspeaker is at the same distance from both ears. For Fig. 3.20b, the elevation stays 0-degree, but the azimuth changes to -80 degrees. In this localization, the loudspeaker is at the same level as the subject's head but -80 degrees to the left, according to the frontal position. In this situation, it is expected to observe more signal activity for the left ear than the right ear since the loudspeaker is closer to the left ear than the right ear.

Fig. 3.21a and Fig. 3.21b represent the frequency intensity distribution of the WS of the left and right ear, respectively, when the SSL is always at 0 degrees in elevation, and the azimuth angle varies from -80 degrees (left of the head) to 80 degrees (right of the head). For instance, when the SSL is -80 degrees, meaning to the complete left of the subject head, the frequency intensity is higher in Fig. 3.21a, left ear, compared to Fig. 3.21b, right ear. Since the SSL is closer to the left ear in this situation, it is expected to have more activity in the left ear WS compared to the right ear WS. The level of activity variation represents some SSL cues the NN algorithm will take advantage of to estimate the SSL. The more the azimuth angle increases toward 80 degrees toward the right ear, the more the intensity will increase in the right ear WS and decrease in the left ear. As expected, the symmetry

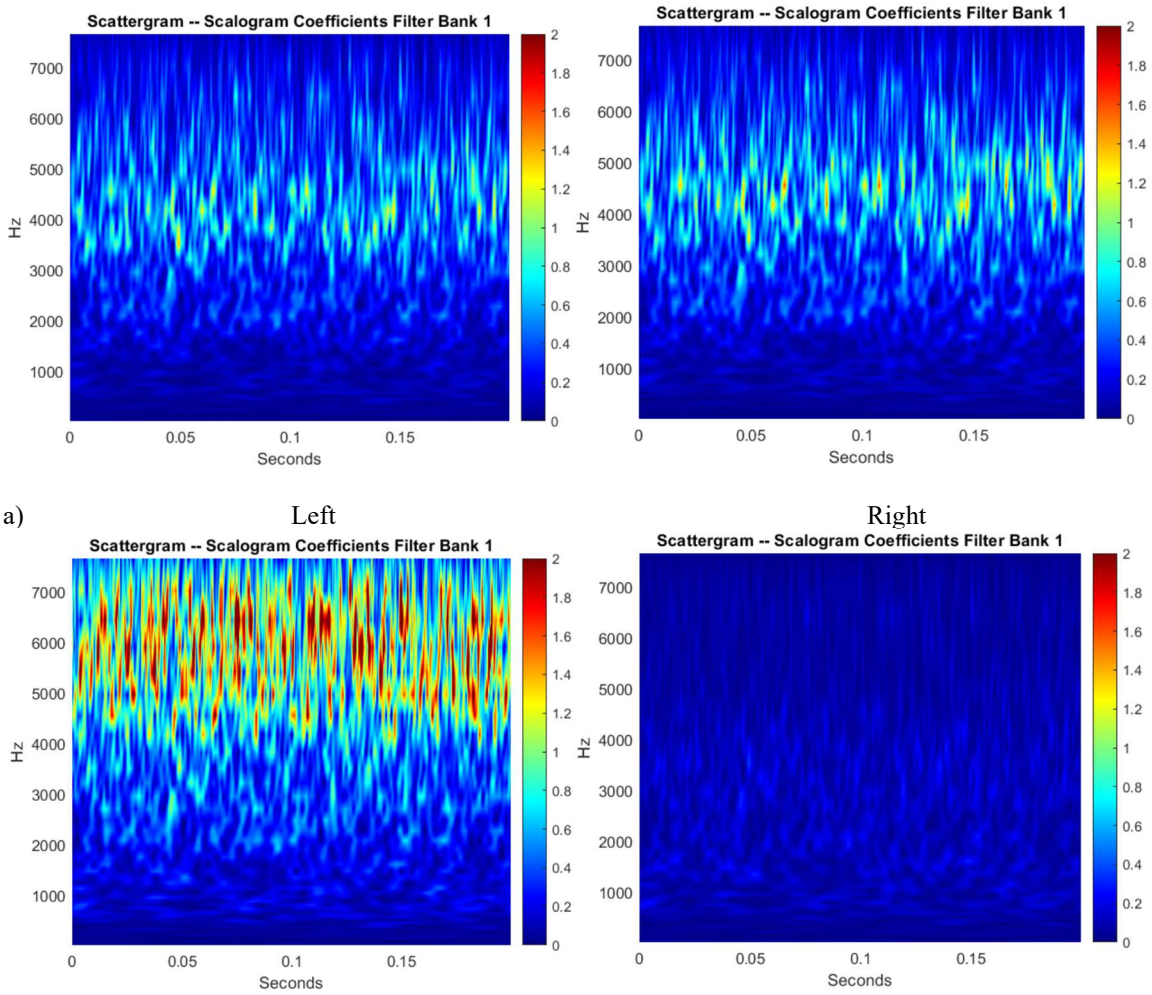


Fig. 3.20 Feature extraction based on scattering decomposition with left and right ears for a) Azimuth 0° and elevation 0°, b) Azimuth is -80° and elevation 0°. Results were calculated with the HRTFs of subject #21 KEMAR in the CIPIC HRTF database.

between the left and right ear, WS can be noted, and at 0-degree azimuth, both WS is identical.

Fig. 3.21c and Fig. 3.21d show how the elevation SSL is more challenging to estimate than the azimuth SSL. The azimuth angle is maintained at 40 degrees in the situation presented,

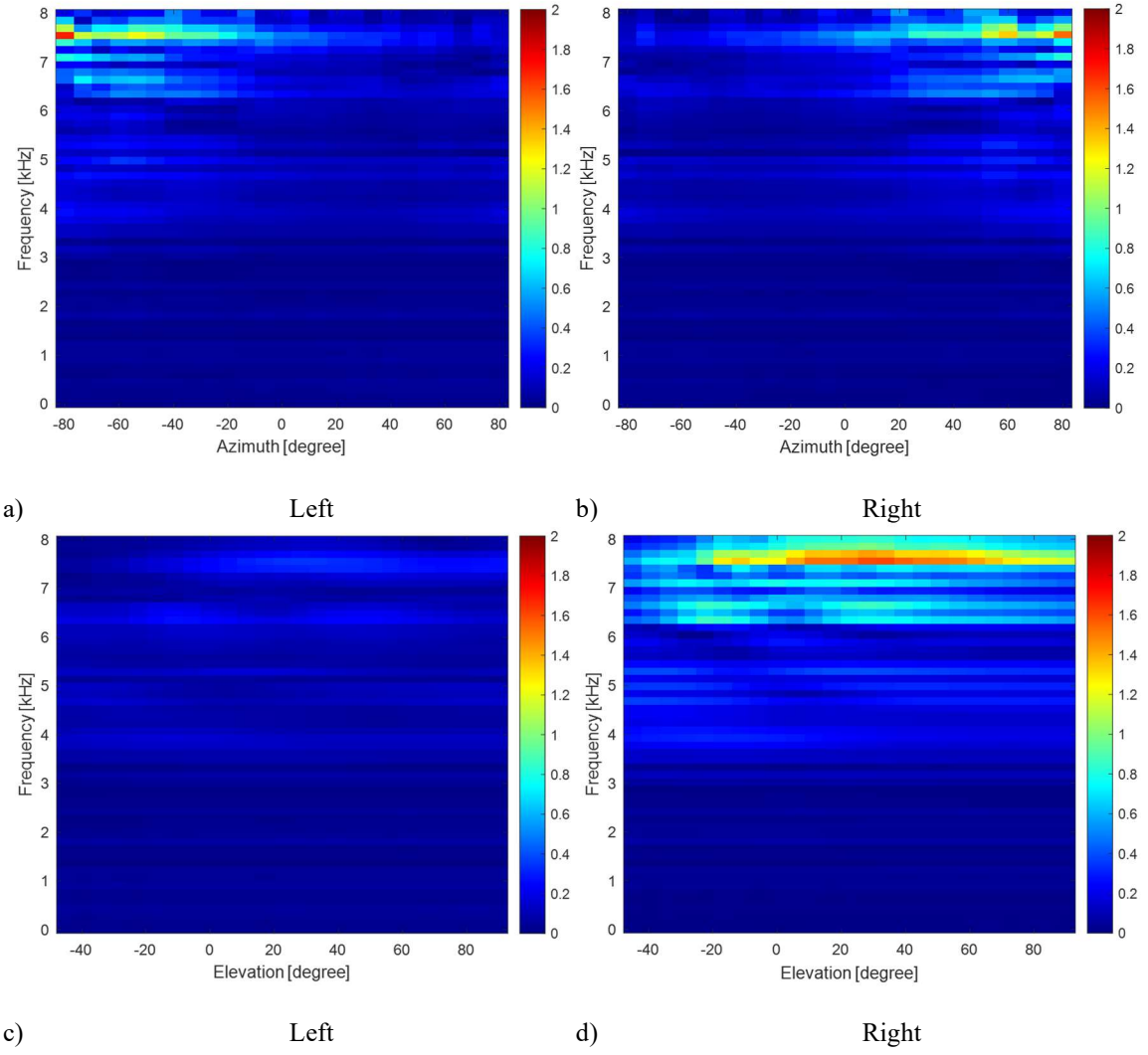


Fig. 3.21 Left ear WS distribution (a) and right ear WS distribution (b) versus azimuth when elevation is 0◦, left ear WS distribution (c) and right ear WS distribution (d) versus elevation when azimuth is 40◦.

and the elevation angles vary from -45 to 90 degrees. Since the loudspeaker is closer and always kept closer to the right ear, low activity is present in the left ear WS (Fig. 3.21c). For the right ear (Fig. 3.21d), the intensity variation occurs when the loudspeaker reaches

the limits of -45 and 90 degrees. However, due to the cone of confusion effect, only a slight intensity variation is observed when an elevation angle change.

In sum, Fig. 3.20 and Fig. 3.21 highlight how the elevation angle is more complex to estimate than the azimuth angle and how the WS feature extraction technique emphasizes the SSL cues used by the NN. The sound source type used to create Fig. 3.20 and Fig. 3.21 is a uniform data sequence without additive noise since the focus is only on the information buried in the HRTF.

This thesis proposes and applies a data sequence to the LSTM inputs. To explain the input sequences, assuming the case of a time sequence of 0.1 s with a sample rate of 44.1 kHz. In this case, from Fig. 3.22, the image matrix size will be 52 (step frequency) × 512 (step time). The input length at each time step was less than 2×52 and less than 512 for each epoch.

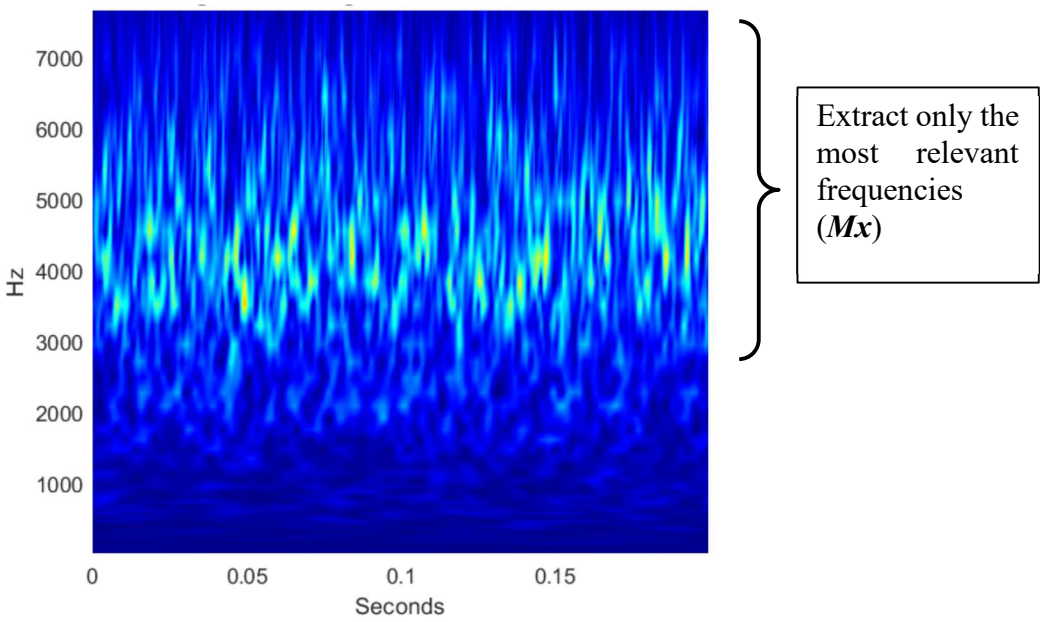


Fig. 3.22 Extraction of the most relevant frequencies (*Mx*) from 0 to 8 kHz.

The input of the LSTM consists of using at each time step (x-axis in Fig. 3.22) one frequency sequence of length Mx for each ear, with $Mx = 1, 2, \ldots, 52$. The high and medium frequency levels represent the sound source location, Fig. 3.22. The value of Mx was less than 30 (excluding low-frequency levels), and the number of time sequences was limited to 512. As expected, the training complexity per epoch is many times reduced compared to without the feature extraction step.

3.4.2 LSTM Results with features extraction

Two scenarios are studied using the LSTM method with the feature extraction technique to compare without feature extraction. They appear in Fig. 3.1 for Scenario #1, Fig. 3.15 for Scenario #2 and Fig. 3.17 for Scenario #3. To compare the results using feature extraction to the previous results without feature extraction, the signals used are the same as those used without feature extraction.

Scenario #1 – Uniformly distributed random sequences

Fig. 3.23 shows different results of the algorithm's performance depending on the NN parameters, this type of figure helps to find the best parameters combination for the NN. In this case, the best performances are when Mx=30 and hidden units=10.

When using the best parameters chosen previously, Mx=30 and hidden units=10, the performance for Scenario #1 when using uniformly distributed random sequences as the input signal is shown in Fig. 3.24. The RRMSE is low, 0.0303, with all loudspeaker locations estimated correctly.

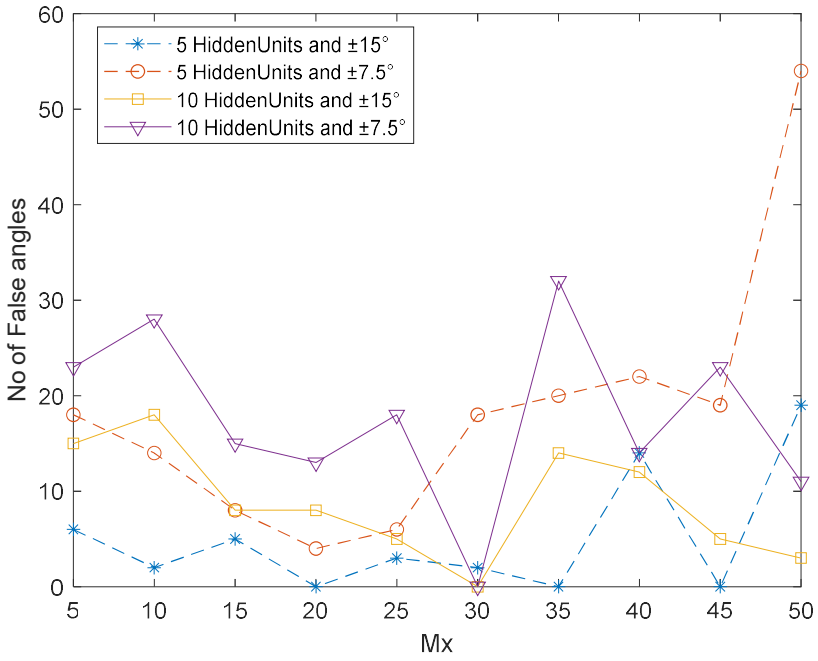


Fig. 3.23 Scenario #1 – Performance results to estimate 96 angle positions for different LSTM sizes of *Mx* and Hidden Units

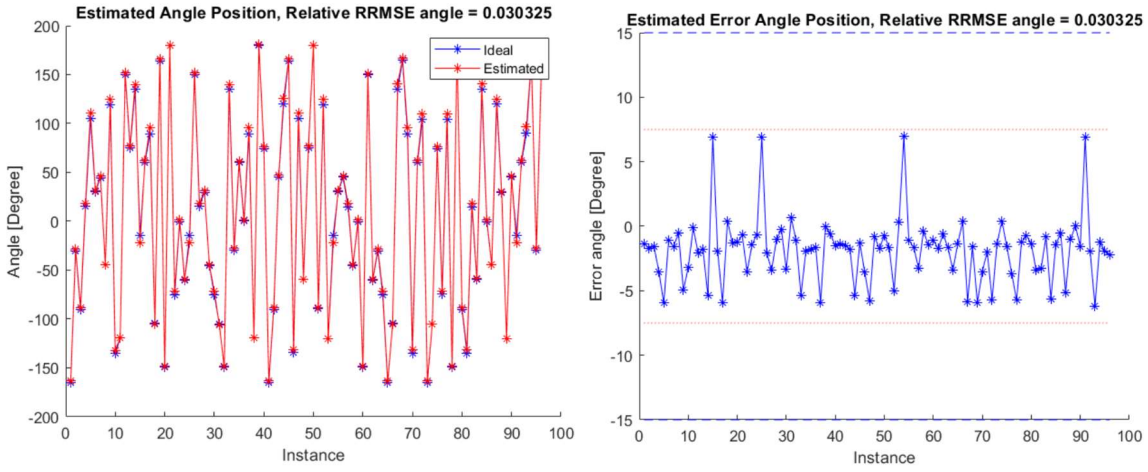


Fig. 3.24 Scenario 1 – Results to estimate 240 angle positions with *Mx*=30 and Hidden Units=10, RRMSE angle = 0.0303 resulting 0/240 (0%) errors superior to ±15° and 0/240 (0%) errors superior to ±7.5°

Scenario #2 – Regression source localization

Fig. 3.25 contains the results for the loudspeaker position estimation when performing regression for the sound source localization. The RRMSE value is 0.028, which is excellent for a regression problem.

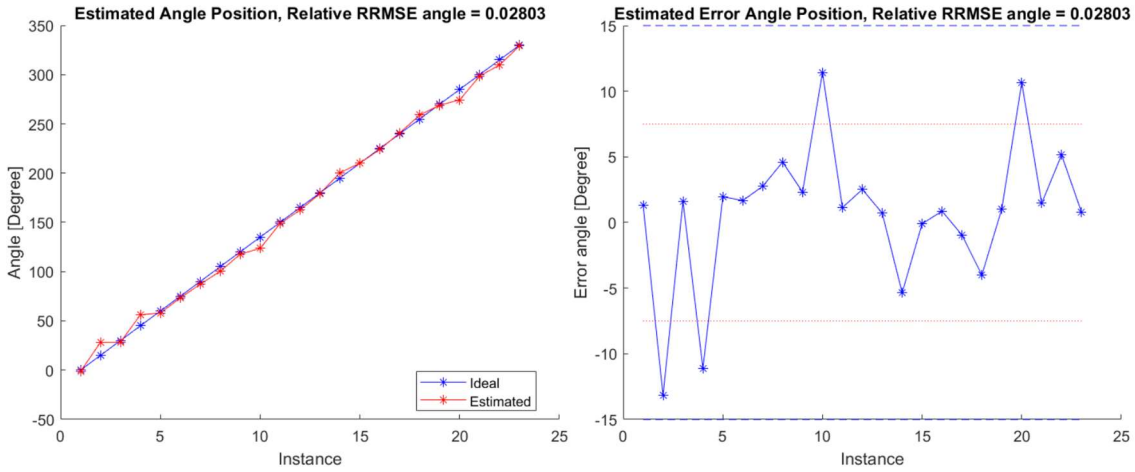


Fig. 3.25 Scenario 2 – Results to estimate 24 angle positions with *Mx*=20 and Hidden Units=20, RRMSE angle = 0.0280 resulting 0/24 (0%) errors superior to ±15° and 4/24 (16%) errors superior to ±7.5°

Scenario #3 – Musical sound

Usually, two parameters are changed to improve performance, *Mx* and the number of hidden LSTM units. When one of the two parameters remains fixed, the second parameter can be varied. The result of these modifications can be seen in Fig. 3.26, where the number of hidden units is set, and *Mx* varies. According to the results, the number of the wrong estimated angle of loudspeaker positions is minimized when *Mx* = 20. This is how the optimal parameters for the neural network are found.

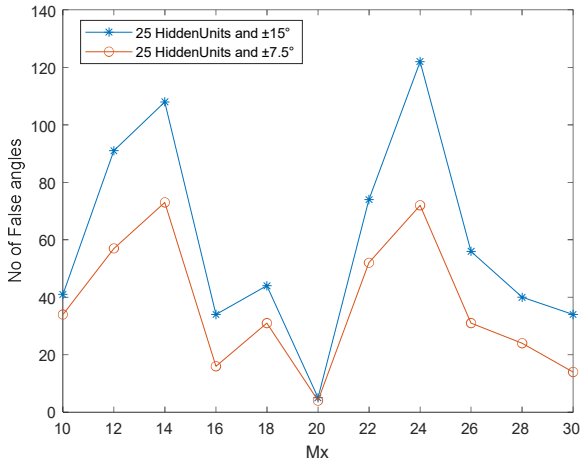


Fig. 3.26 Scenario #3 – Performance results to estimate 240 angle positions for different LSTM sizes of *Mx* and 25 Hidden Units

Fig. 3.27 shows the best result for estimating the speaker's position when the soundtrack played by the speaker is music from an organ instrument. Fig. 3.27b is a zoom of Fig. 3.27a to see how the neural network's output oscillates around the actual position of the loudspeaker. An average of this oscillation is calculated to obtain the estimated position of the loudspeaker. Fig. 3.27c is a better representation of the result where it seems that the algorithm has correctly estimated most positions; the RRMSE error is 0.1277.

3.5 NARX and LSTM discussions

The NARX method studied was limited to only one scenario. The results presented a low accuracy with high computational volume, so the study concentrated on using the LSTM network.

The LSTM has shown the three scenarios better results using feature extraction. The results obtained for each scenario can be compared using the LSTM-without feature extraction in

Fig. 3.14, Fig. 3.16 and Table 3.2, respectively, with the LSTM-with features in Fig. 3.24, Fig. 3.25 and Fig. 3.27.

In Scenario #1, the LSTM-with feature extraction presents a total number of angle errors outside the ±7.5° of 0% compared with 8% for LSTM-without feature extraction.

In Scenario #2, the regression for only the 0.1 s case was studied. The LSTM-with feature presents a total number of angle errors outside the ±7.5° of 16% compared with 33% for the LSTM-without feature.

In Scenario #3, the musical sequence, the LSTM-with feature, presents a total number of angle errors outside the ±7.5° of 1.7% compared with 5% for LSTM-without feature extraction. The LSTM-with feature shows only 3 angle errors outside of the ±15°. This result is impressive compared to the NARX and the LSTM without features.

3.5.1 Complexity analysis discussions

The idea is to exploit discrimination parameters in the time-frequency domain to improve the information characterizing the ear channels. Similarly, based on this improved information using feature extraction, the volume of data to train the LSTM network is reduced compared to the nonfeature method (Section 3.3). For example, for the time sequence of 0.1s, from Fig. 3.22, the matrix size will be 52 (step frequency) × 512 (step time).

Without the feature extraction, the input sequences applied to LSTM were $1\times 2Mx$, where Mx represents the number of frequencies used from the scattergram decomposition for each

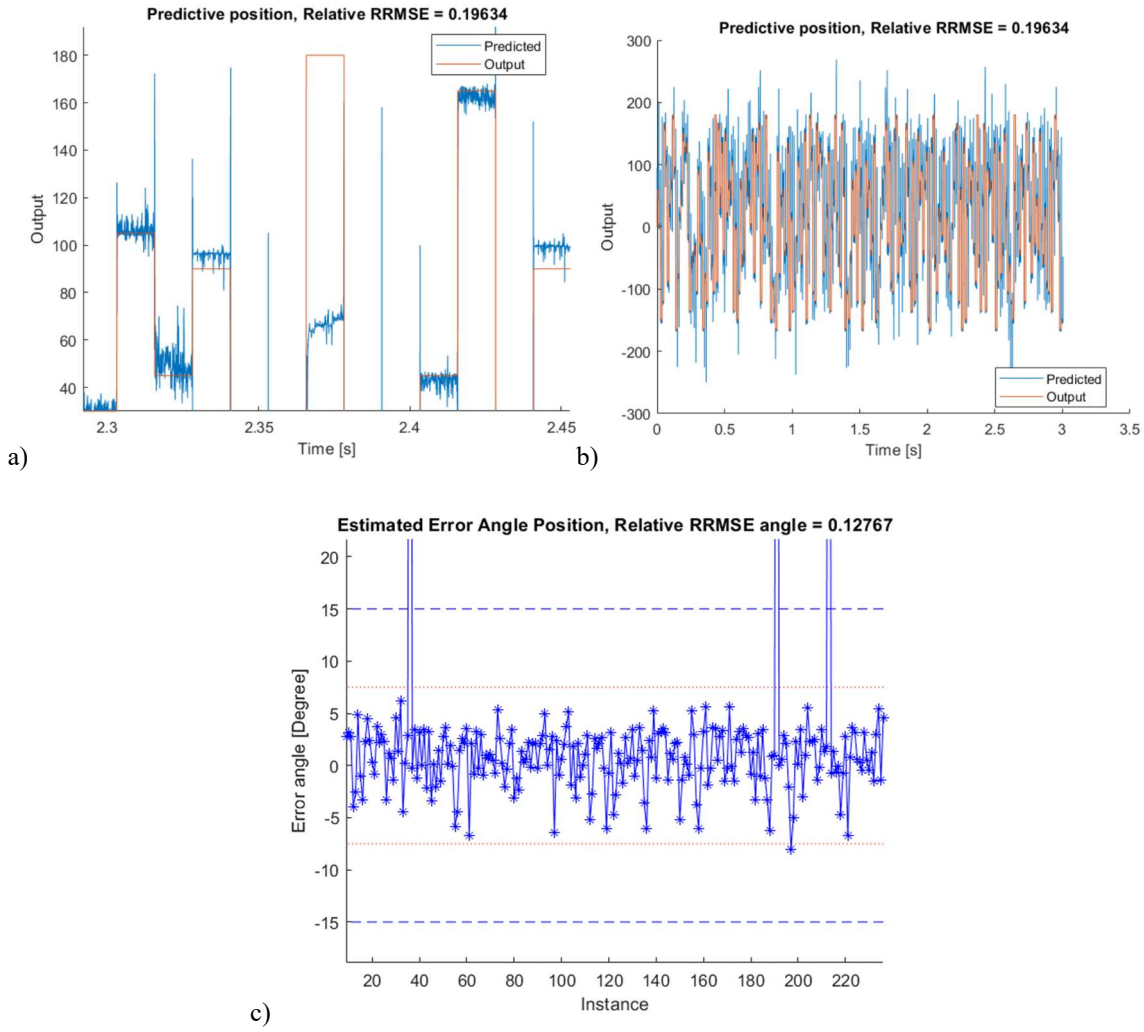


Fig. 3.27 Scenario 3 – Results to estimate 240 angle positions with *Mx*=20 and Hidden Units=25, a) RRMSE on *y*(*n*), b) zoomed plot of a), and c) RRMSE angle of 0.128 resulting in 3/240 (1.25%) errors superior to ±15° and 4/240 (1.7%) errors superior to ±7.5°

ear and defines the number of LSTM inputs. At 44.1 kHz and a time sequence of 0.1s, 4410 data sequences of length 2 × *Mx* for each epoch need to be applied, where *Mx* needs to be fixed depending on the training problem. In Section 3.3, Scenario #3, the optimal value was *Mx*=60 to minimize the classification error inside ±7.5 degrees.

Without feature extraction, each epoch has 120 × 4410 data sequences and only 60 × 512 after the feature extraction step, a reduction factor of nearly 20 times fewer data. This factor corresponds to the speedup at each epoch for the same number of hidden units used for LSTM. Based on the results without and with features, for the same scenario, a reduction of hidden units with features and a reduction in the number of training epochs to minimize the angle detection error is observed. Consequently, a significant decrease in computational time was observed.

Sound source localization is used in many applications where performance quality and low complexity are constant challenges. Machine learning represents recent approaches to estimate sound source localization or detection based on unknown source signals to increase performance. This Chapter has applied the NARX and LSTM methods directly to the received signal from the left and right ears. To improve the performance, feature extraction was applied. The scattering decomposition was used to extract the space-time information parameters sensitive to SSL. The classification and regression for uniform noise and musical sources were performed. Comparison results show better performance using a feature extraction technique than without feature extraction. HRIR dataset is used at different speaker locations to generate data to locate the single speaker coordinates precisely. The results show a good quality of source localization using a short sequence of 0.1 s, 0.2 s and 0.4 s with uniform random and music sequences, the localization error diminishing with the sequence length. Furthermore, localization prediction is shown based on the LSTM regression method with an error of less than 2%.

Chapter 4: LSTM with Scattering Decomposition-Based Feature Extraction (WS-LSTM)

4.1 Proposed features and WS- LSTM method

SSL determines the position of a sound source based on two types of cues that contain head-related transfer functions. Many audio processing systems rely on SSL, for example, speech enhancement/recognition and human-robot interaction. However, the accuracy of SSL under adverse acoustic scenarios is still hard to be ensured. To improve the estimation accuracy of the loudspeaker position in noisy conditions, an LSTM with a feature extraction technique based on scattering decomposition (WS-LSTM) is proposed. For azimuth conditions, the obtained results demonstrate that the proposed method achieved excellent performance in multiple noisy environments compared to the most recent literature methods.

4.2 Proposed WS-LSTM Architecture

This thesis proposes a neural network method using LSTM [50] to localize sound sources around a head using data acquired from an HRTF database. Also, this method implements a feature extraction technique based on wavelet scattering decomposition. The proposed method determines sound localization based on the extraction of location-specific synchrony patterns and trains a supervised algorithm to learn the mapping between synchrony patterns and locations from the set of example sounds. The system's output is

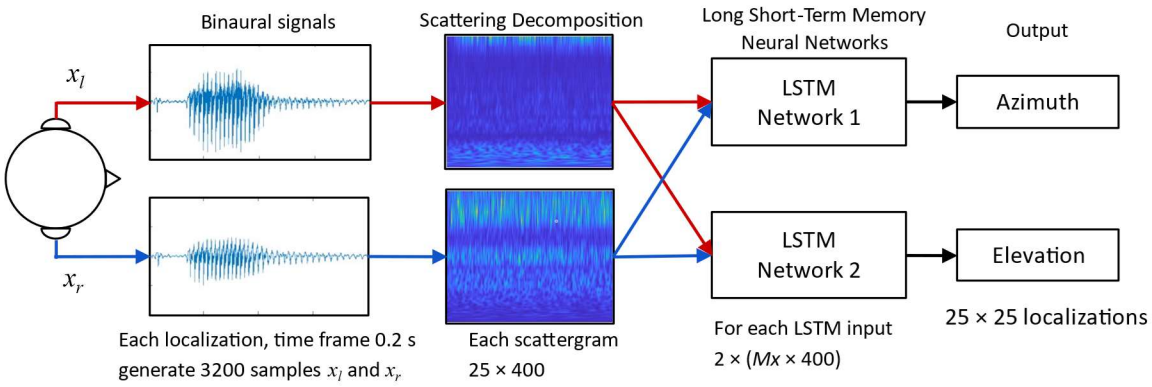


Fig. 4.1 Flowchart of the proposed WS-LSTM SSL system

the spatial source location of a sound by providing the system with a binaural audio record of the sound.

Fig. 4.1 shows a flowchart of the methodology used to estimate an SSL in azimuth and elevation, the same as the previous Chapter but with the addition of a second LSTM network for the elevation estimation. First, it is necessary to generate the dataset of signals modified by the HRIR into three parts. One of them will be reserved for the learning step to adapt the NN's weight parameters. The others will be used for the validation step to choose the training iteration number minimizing the validation error. It will also be used for the generalization steps to evaluate the model's performance in multiple scenarios. In this Chapter, different additive noise levels are tested. In all stages, the datasets are passed through a feature extraction algorithm highlighting certain information deemed more relevant for the LSTMs. Initially, the LSTMs are in a learning phase; the objective is to adapt the weights in the layers later used with the generalization dataset to evaluate the

model's performance. The LSTM outputs are scalar values indicating the approximate azimuth and elevation of the SSL.

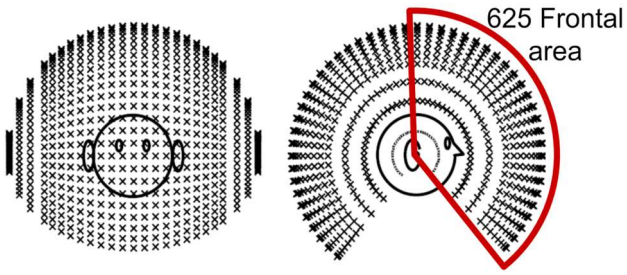
4.3 Feature Method

In this Chapter, data sequences are proposed and applied to the LSTM inputs. A time sequence of 0.2 s at a 16 kHz sample rate is used to create the binaural signal dataset. 16 kHz is selected because it matches the soundtrack sampling rate. Some tests have been carried out and 0.2 s at 16 kHz shows a good compromise between performance and complexity in terms of time and memory usage. Also, the literature often uses a time sequence of 0.2 s at 16 kHz, which facilitates comparisons. In this case, the image matrix size will be 54 (step frequency) × 400 (step time). The number of step frequency sizes defined the number of LSTM inputs used for each left and right ear data sequence, with $Mx = 1, 2, \ldots, 54$. So, the number of LSTM inputs is $2 \times Mx$. The input length used all step time (400) at each epoch.

Therefore, the input of the LSTM consists of choosing the best step frequency and maximizing the SSL performance. Only the most relevant frequencies that are the most prone to containing the information about the SSL are selected. The high and medium frequency levels are more representative of the SSL. The value of Mx studied was between 5 and 35 (excluding low-frequency levels), and the number of time sequences is limited to 400-time steps. As expected, the training complexity per epoch is reduced compared to without the feature extraction step. To clarify, in our case, without feature extraction, 16000 Hz × 0.2 s = 3200 time steps are needed.

4.4 Head Related Transfer Function Dataset

The HRTF signals in this Chapter come from an open CIPIC HRTF Database [37]. The open CIPIC HRTF database comprises massive measurements of HRIR on 45 human subjects and the KEMAR manikin by playing records of multiple sounds at both ears of human models with a single loudspeaker placed at different interaural-polar azimuths and interaural-polar elevations around the head. A loudspeaker one meter from the head moves to 1250 locations, Fig. 4.2, producing a large amount of azimuth and elevation data that takes into account the diversity and length of sound sequences. The recordings were made in an anechoic chamber to reduce echoes and undesired noises. Several sound recordings are registered from both ears using small microphones inside the ear canals.



a) b)

Fig. 4.2 1250 single speaker locations (a) and frontal area (b) [37]

In this thesis, only measurements on Subject #21, the KEMAR manikin, are used in the data generation and performance evaluation. Also, only 625 of the 1250 single speaker locations in the frontal area of the subject will be considered Fig. 4.2b. 25 azimuth localizations range from -80° to +80°, and 25 elevation localizations range from -45° to +90°, thus composing 625 localizations in total. Recording parameters are at a sampling rate of 44.1 kHz, and two-channel inputs represent the left and right ears. Fig. 4.2 shows

the 625 different localizations in the red area. The sampling rate of 44.1 kHz is later downsampled to 16 kHz to match the soundtrack sampling rate.

It is required to generate a database containing the binaural signals, which will be used in the proposed method to implement the NN. This mono signal has a sampling frequency of 16 kHz; it can have different shapes, so several possibilities have been tested. The first step is to create the sound signal played over a single loudspeaker at each location. Following (4), the binaural signals received by each ear are described as (15).

$$x_i(m) = s(m) * h_i(m) + v_i(m), \;\; i = l, r \tag{15}$$

According to (15), source signal *s(m)*, a uniform or Gaussian signal, is convolved (*) with the HRIR filters of the left ear $h_l(m)$ and the HRIR filter of the right ear $h_r(m)$ for each 625 loudspeaker positions. These convolutions are two soundtracks at a sampling frequency of 16 kHz representing the signals in the time domain that would be recorded in the ear using a microphone. After this convolution process, a Gaussian, speech babble of F16 noise *v(m)* is added to simulate a noisy environment [57] with various SNRs (16) ranging from 35dB to -5 dB.

$$SNR = 10 \log_{10} \left(\frac{Signal}{Noise} \right) dB \tag{16}$$

The speech babble noise file was recorded by the Speech Research Unit (SRU) at Institute for Perception-TNO, Netherlands, United Kingdom [57]. It was recorded in a canteen with 100 people speaking, so individual voices are barely audible. The F16 noise recording

comes from the same open sound database. The F16 noise was recorded in a two-seat F-16, travelling at a speed of 500 knots at an altitude of 300-600 feet.

4.5 Results and Discussion

This section contains the obtained performance results of the proposed method showing that the proposed technique has good accuracy on SSL on a frontal head area for azimuth and elevation. The results of the WS-LSTM method will be compared to the reference method based on a time-frequency with two convolutional neural networks (TF-2CNN) with multitask learning [24] introduced in Chapter 2.

The training was realized using a uniform noise input soundtrack $s(m)$, with added Gaussian or Babble noise $v(m)$ from the SRU database [57] at SNRs values of 30dB, 20dB, 10dB, and 0dB, Fig. 4.3a, all plots in Fig. 4.3 are only for 35dB. Similar figures at the four other SNR levels were concatenated with these plots. The corresponding signal for the desired y output during training can be seen in Fig. 4.3b. As for the validation, the shape is similar to the training signals for the 625 SSL, Fig. 4.3c and Fig. 4.3d.

The testing was also accomplished using a uniform noise type signal as input soundtrack $s(m)$, with a newly added Gaussian or Babble noise $v(m)$ at SNRs values of 35 dB, 25 dB, 15 dB, 5 dB, and -5 dB, Fig. 4.3e and Fig. 4.3f. Results for 35 dB and -5 dB become extrapolation range for the testing sequence, the expected accuracy for those levels may be lower. The binaural soundtrack duration of the sequence is 0.2 s for each of the 625 locations.

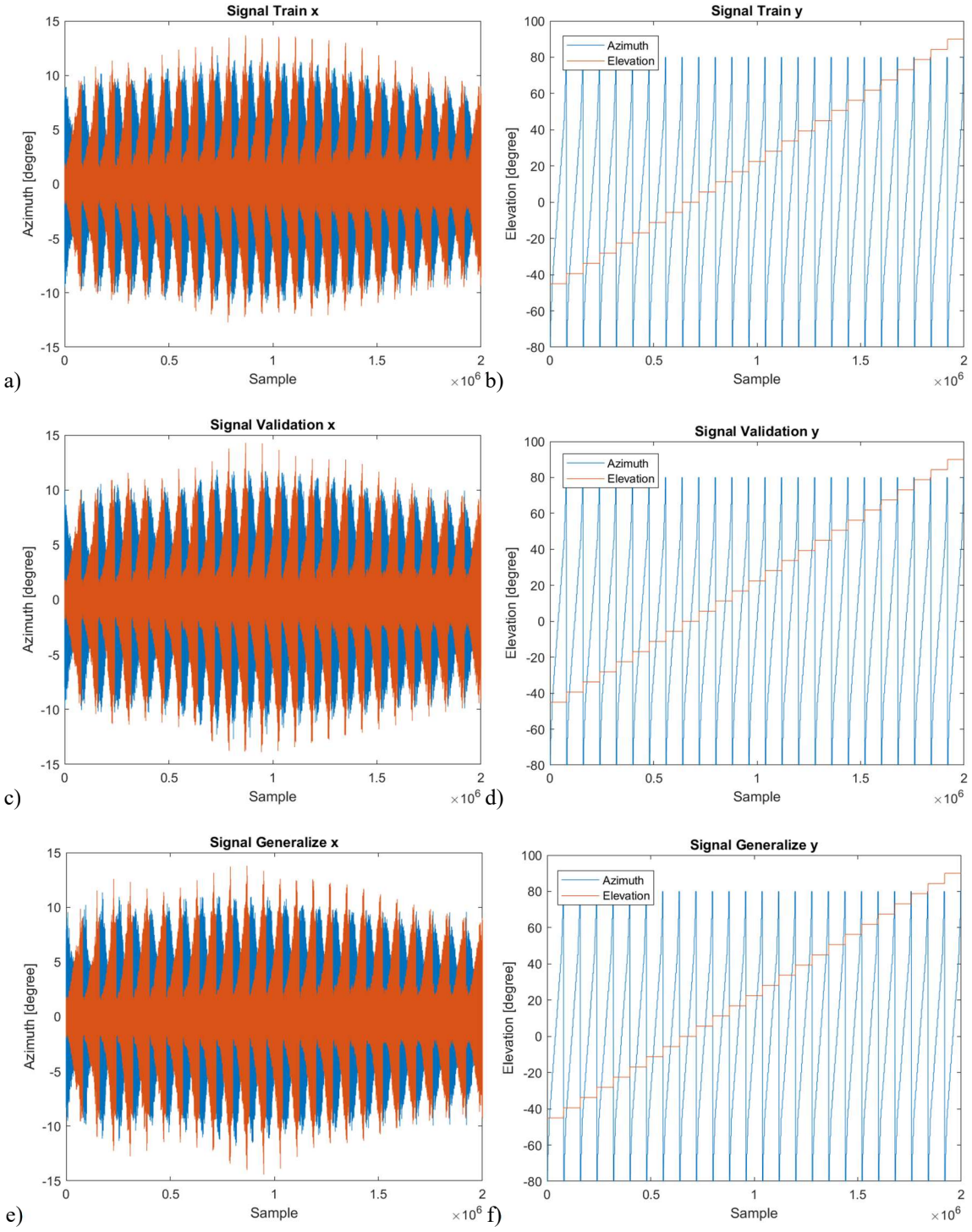


Fig. 4.3 a) Training signal x, b) Training signal y, c) Validation signal x, d) Validation signal y, e) Generalization signal x, f) Generalization signal y

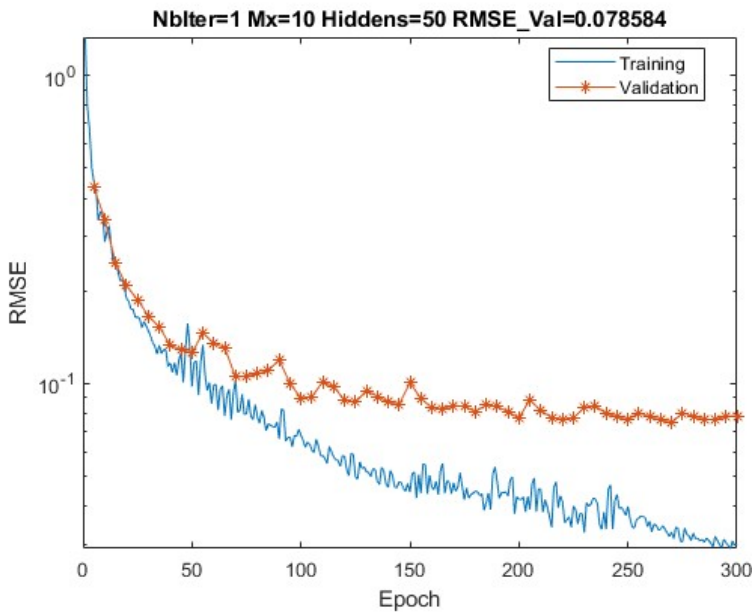


Fig. 4.4 Training curves sample

The Adam adaptation method is used for the two vanilla LSTMs with a learning rate of 0.01 and a learning rate drop factor of 0.95 every 20 epochs. Two critical parameters were adjusted and tested multiple times to optimize the results. They are the hidden units and the number of inputs $2\times Mx$ of the LSTM. The hidden units refer to the number of recurrent cells on the LSTM layer. The *Mx* value refers to the number of frequencies used from the scattergram decomposition for each ear (left/right) and defines the number of LSTM inputs. Fig. 4.4 shows a typical training curve with the training and the validation root mean square error (RMSE) results. RMSE is defined by (17).

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}\left(y(n)-\hat{y}(n)\right)^2}{N}} \tag{17}$$

where *N* defines the number of estimated positions used in azimuth and elevation.

A number of hidden units of 50 was used to obtain the training curves in Fig. 4.4. The *Mx* value was set to 25, which means that from the 25 frequency steps in the scattergram decomposition, 25-step frequencies from the left and 25-step frequency from the right ear were used generating scattergrams of size 50 × 400 used as LSTM input.

A confusion matrix is utilized to evaluate the overall generalization performance of the proposed method after proper training. Fig. 4.5 is a sample of the confusion matrix resulting from the training in Fig. 4.4. The goal is to obtain all the predicted values on the diagonal of the matrix, which means that the 25 angles of the predicted values (x-axis) correspond to the 25 angles of the actual values (y-axis).

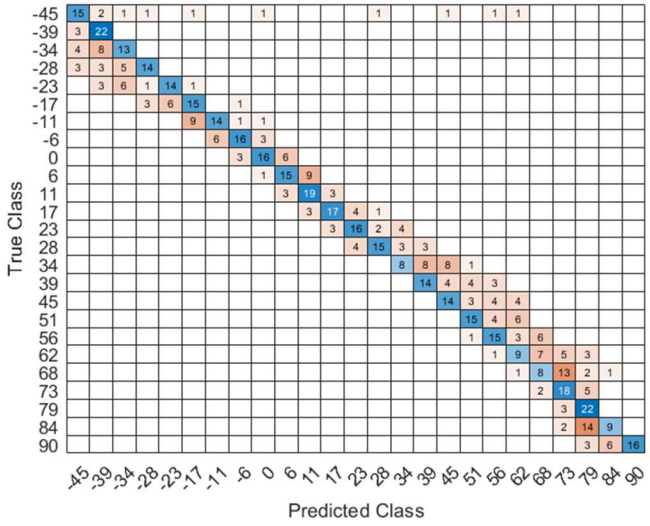


Fig. 4.5 Confusion matrix of typical results with WS-LSTM for *Mx* = 10, SNR = 15 dB with a global accuracy of 59%

The parameters of the two LSTMs, such as the hidden unit's quantity and the *Mx* values, were set empirically by testing multiple values for both parameters and converged to a better result. Fig. 4.6 shows a sample of the global accuracy versus the number of

frequencies (*Mx*) and different numbers of hidden units. The result shows that the accuracy increases with *Mx*, and the global accuracy is maximized and steadier for *Mx* = 25 and 70 hidden units. Accuracy means the average of all 625 localization correctly localized when the error between the perfect angle and output angle estimated is less than a threshold (±5° or ±2.5°). From Fig. 4.6, the highest accuracy is obtained with 80 hidden units and 30 for *Mx*. However, using 80 hidden units creates a fluctuation in the accuracy level. Thus, 70 hidden units are selected with an *Mx* value of 25 since it is steadier with a minimal impact on the accuracy. Also, a small number of hidden units makes the training faster, and a smaller value for *Mx* reduces the size of the scattergram inputted to the neural network. All results are obtained with *Mx* = 25 and 70 hidden units.

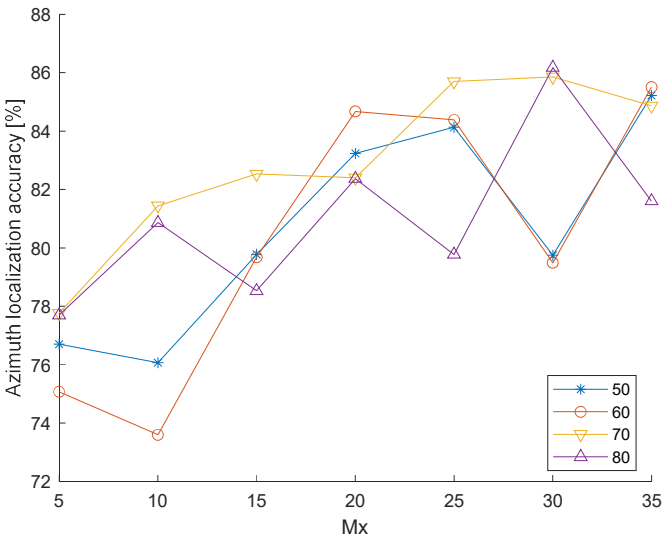


Fig. 4.6 Average azimuth localization accuracy with WS-LSTM for uniform sound for various *Mx* and hidden unit values

The signal processing, the feature filters, and the machine learning methods were implemented on MATLAB R2021b [58]. The heavy computation, such as the machine

learning training phase, was executed on a computer operating Windows 10 Pro with an Intel Core i9-9900X CPU @ 3.50 GHz and an NVIDIA RTX 2080 Ti. Training average duration is about 3 hours (40 s per epoch for 300 epochs) due to the higher number of loudspeaker localization (625), hidden units, *Mx*, and data sequences length.

4.5.1 Classification Model

The proposed method is compared with TF-2CNN to evaluate the performance. The proposed method used a signal source sequence of only 0.2 s, using a uniform sequence for each sequence for training, validation, and testing. Our motivation is to reduce the huge WS-LSTM computational times compared to CNN methods.

Table 4.1 and Table 4.2 present azimuth angle estimation and elevation angle estimation, respectively. Note that since the azimuth estimation and the elevation estimation result from two distinct LSTMs, the parameters and the training duration varied between the two LSTMs. The comparison is made regarding accuracy performance in multiple SNRs conditions with both Gaussian and speech babble noise. Contrary to TF-2CNN based on classification results, the WS-LSTM proposed method is based on regression estimating elevation and azimuth angles. Thereby, the accuracy of the proposed method is based on RRMSE and error margin values for angle estimations. To classify the estimated angle, two error margin values for the estimation of the SSL are considered, ±5° and ±2.5°. Margin errors were selected to be close enough to the exact SSL angle values. The lower accuracy for the elevation estimation is standard in almost all methods of SSL, mainly due to the symmetrical position of the two ears. For instance, when a sound in front of the head

Table 4.1 Azimuth localization accuracy for Gaussian and babble noise using a time sequence of 0.2 s of 25 × 25 (625) angles for TF-2CNN and WS-LSTM with several SNRs and using a uniform sound source

Noise	SNR	TF-2CNN		WS-LSTM		
		RRMSE	Accuracy (%)	RRMSE	Accuracy (%)	
					±5°	±2.5°
Gaussian	35	0.0685	98.97	0.0678	97.59	89.79
	25	0.0160	99.74	0.0624	97.78	89.65
	15	0.0084	99.80	0.0610	98.10	89.25
	5	0.0122	99.54	0.1341	82.32	67.73
	-5	0.1630	80.67	0.4600	36.95	18.50
Speech Babble	35	0.0000	100.00	0.0591	95.39	82.21
	25	0.0000	100.00	0.0530	97.57	88.32
	15	0.0000	100.00	0.0526	97.06	90.59
	5	0.0013	100.00	0.0602	98.08	88.30
	-5	0.0301	99.69	0.0960	96.50	87.87
	Average	**0.0300**	**97.84**	**0.1106**	**89.73**	**79.22**

Table 4.2 Elevation localization accuracy for Gaussian and babble noise using a time sequence of 0.2 s of 25 × 25 (625) angles for TF-2CNN and WS-LSTM with several SNRs and using a uniform sound source

Noise	SNR	TF-2CNN		WS-LSTM		
		RRMSE	Accuracy (%)	RRMSE	Accuracy (%)	
					±5°	±2.5°
Gaussian	35	0.5295	53.47	0.1228	73.07	53.83
	25	0.5064	55.62	0.0937	80.87	60.32
	15	0.5014	50.79	0.1126	73.58	51.54
	5	0.7249	29.53	0.2241	53.42	33.67
	-5	1.1069	8.60	0.5537	16.66	9.30
Speech Babble	35	0.6359	38.95	0.1026	80.67	59.94
	25	0.6363	38.85	0.0985	81.38	60.93
	15	0.6431	38.87	0.0882	82.58	64.30
	5	0.6671	38.20	0.0915	84.74	64.14
	-5	0.7075	37.56	0.1406	78.69	60.58
	Average	**0.6659**	**39.04**	**0.1628**	**70.57**	**51.86**

is higher than the horizontal line, both ears are at the same distance from the sound source. Compared to azimuth estimation, when a sound is not directly in front of the head, it is closer to one ear than the other. This eventually impacts the ITD and ILD, the HRTF, and the recorded signal at both ears. Table 4.1 and Table 4.2 show lower RRMSE for azimuth than elevation. The WS-LSTM presents good accuracy and is superior to TF-2CNN for elevation for all SNR levels and noises.

4.5.2 Regression Model

The most interesting comparison of the proposed method is the capability of regression output to estimate azimuth and elevation angles. Thus, not limiting the system output to 625 possible speaker localizations for classification methods. In this scenario, the training and validation were realized using the same conditions as the previous results. Only 169 speaker locations are used for training and validation to evaluate the regression performance. Those 169 locations are distributed as 13 locations in azimuth and 13 in elevation. The testing was performed on all 625 locations. The results are shown in Table 4.3 and Table 4.4. RRSME higher than in Table 4.3 and Table 4.4 conditions are observed but giving a good accuracy considering that only 27% of localizations are used for training and validation.

In this Chapter, we presented classification and regression results for binaural SSL using the WS feature extraction, followed by an LSTM network for each azimuth and elevation angle. To evaluate the performances, we used the metrics RRMSE, MAE and accuracy with an error range of ±2.5° and ±5°. Furthermore, we used two types of source signals,

Table 4.3 Azimuth and elevation localization accuracy for uniform sound source using 13 × 13 (169) angles of 0.2 s time sequence for the training-validation phase and 25 × 25 (625) angles for the test with WS-LSTM with several SNRs and noises

Noise	SNR	Azimuth				Elevation			
		MAE (°)	RRMSE	Accuracy (%)		MAE (°)	RRMSE	Accuracy (%)	
				±5°	±2.5°			±5°	±2.5°
Gaussian	35	1.95	0.0870	90.13	78.10	5.23	0.1538	58.49	38.34
	25	2.00	0.0886	90.22	76.96	4.48	0.1291	62.69	40.75
	15	2.03	0.0902	89.97	77.10	4.48	0.1294	62.84	40.75
	5	2.68	0.1116	83.69	68.84	5.35	0.1559	58.46	34.91
	-5	5.49	0.1908	59.82	31.04	25.87	0.6877	11.51	5.97
Speech Babble	35	2.02	0.0895	89.62	77.22	3.60	0.1070	73.11	49.55
	25	1.94	0.0853	89.87	78.98	3.19	0.0968	77.97	53.39
	15	2.12	0.0923	88.35	75.56	3.51	0.1057	74.23	50.82
	5	2.07	0.0890	88.99	76.29	3.67	0.1108	73.39	49.86
	-5	2.14	0.0924	88.06	75.98	3.69	0.1115	72.32	49.86
F16	35	1.82	0.0820	90.79	80.36	3.96	0.1132	70.10	44.86
	25	1.74	0.0781	91.77	80.76	3.75	0.1071	71.05	46.34
	15	1.76	0.0787	91.73	80.48	3.90	0.1141	70.40	46.07
	5	1.83	0.0813	90.60	79.79	4.01	0.1147	69.10	44.58
	-5	2.62	0.1130	86.31	70.60	7.25	0.2171	49.79	30.30
	Average	**2.28**	**0.0966**	**87.33**	**73.87**	**5.73**	**0.1636**	**63.70**	**41.76**

Table 4.4 Azimuth and elevation localization accuracy for Gaussian sound source using 13 × 13 (169) angles of 0.2 s time sequence for the training-validation phase and 25 × 25 (625) angles for the test with WS-LSTM with several SNRs and noises

Noise	SNR	Azimuth				Elevation			
		MAE (°)	RRMSE	Accuracy (%)		MAE (°)	RRMSE	Accuracy (%)	
				±5°	±2.5°			±5°	±2.5°
Gaussian	35	6.56	0.2591	67.42	43.07	29.00	0.7854	11.88	6.18
	25	7.24	0.2836	64.64	41.17	29.87	0.8116	11.09	5.69
	15	7.07	0.2766	65.12	40.58	29.14	0.7863	11.07	5.65
	5	6.14	0.2331	63.66	38.77	29.34	0.7846	12.44	6.52
	-5	6.12	0.2220	58.26	39.25	33.58	0.9159	11.65	5.38
Speech Babble	35	1.91	0.0906	90.06	81.60	3.84	0.1134	68.05	47.36
	25	1.82	0.0846	91.20	81.53	4.17	0.1235	65.59	45.05
	15	1.85	0.0890	89.73	82.61	4.36	0.1300	63.42	44.33
	5	1.88	0.0897	89.91	81.82	4.63	0.1417	63.96	44.21
	-5	1.87	0.0872	90.29	80.91	3.98	0.1193	67.61	46.67
F16	35	1.85	0.0878	90.29	82.06	4.01	0.1175	66.82	44.96
	25	1.99	0.0920	89.28	79.87	3.80	0.1112	69.39	46.88
	15	2.03	0.0934	88.94	78.58	4.34	0.1268	63.91	42.85
	5	1.97	0.0903	89.60	79.89	4.21	0.1228	65.22	42.88
	-5	2.66	0.1162	86.25	71.04	7.56	0.2357	48.30	29.56
	Average	**3.53**	**0.1464**	**80.98**	**66.85**	**13.05**	**0.3617**	**46.69**	**30.94**

uniform and Gaussian noises, with three types of additive noises, Gaussian noise, speech babble, and F16. To measure the robustness of the methods, we considered different noise levels for learning and testing.

The detailed analysis and discussion of the performances will be carried out in Chapter 6, following obtaining the results for the TF-2CNN method in regression mode, which will be presented in Chapter 5.

Chapter 5: CNN with Scattering Decomposition-Based Feature Extraction (WS-CNN)

This Chapter introduces another approach for the machine learning section of the proposed method, using the same feature extraction method as introduced in Chapter 3, wavelet scattering, but using a convolutional neural network (CNN) instead of an LSTM network.

As explained in Chapter 2, Pang et al. [24] introduced a binaural SSL method based on time-frequency CNN (TF-2CNN), with multitask learning to estimate the localization of 625 loudspeaker localisations in the frontal area of a head model. They used the IPD and the ILD cues as input to their TC-CNN method to classify a sound source in 625 possible locations. The proposed method studied in this Chapter is based on wavelet scattering CNN (WS-CNN) from a classification and regression point of view.

5.1 Proposed features and WS-CNN method

The flowchart of the proposed WS-CNN SSL method Fig. 5.1 is similar to the WS-LSTM method Fig. 4.1, except for the neural network element now a CNN and for the processing applied to the scattergram generated for the binaural signals before being used as input to the NN. The scattergram generation is the same as for the LSTM method. However, for the WS-CNN method, each scattergram is divided with an overlapping into 15 slides for the azimuth CNN and 7 for the elevation CNN.

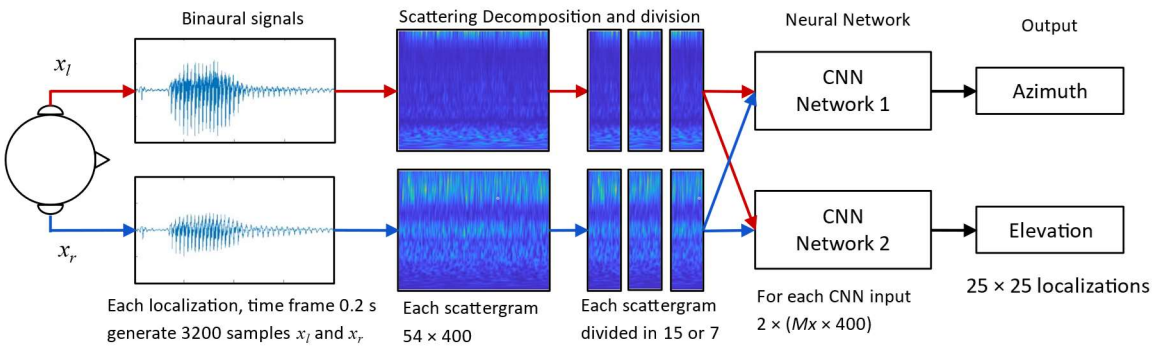


Fig. 5.1 Flowchart of the proposed WS-CNN SSL method

Compared to a simple division, the overlap allows having almost the same number of images while having twice as much information per image. For example, in azimuth, with a normal division in 16 equal parts for the width (time component), the images are obtained by taking the first 50 pixels (1 to 50), then the next 50 pixels (51 to 100) until the last pixel (351 to 400 for the 0.2s signals). The total ends up with 16 images of 54 × 50. Making an overlap here consists of taking the first 100 pixels (1 to 100), then the next 100 pixels by shifting by 50 pixels (51 to 150, then 101 to 200 etc...). The total ends up with 15 images of 54 × 100 with the overlap. We, therefore, have twice as much information in the images created with the overlay, with one image less. As shown in Fig. 5.2, the accuracy obtained during testing showed that dividing the scattergram into multiple slides with overlapping (15 azimuths and 7 elevations) increased accuracy compared to dividing without overlapping.

5.2 Proposed WS-CNN Classification Architecture

With the interaural features of wavelet scattering constructed in Chapter 3 as input, a CNN configuration is proposed to estimate the azimuth and elevation based on the cues

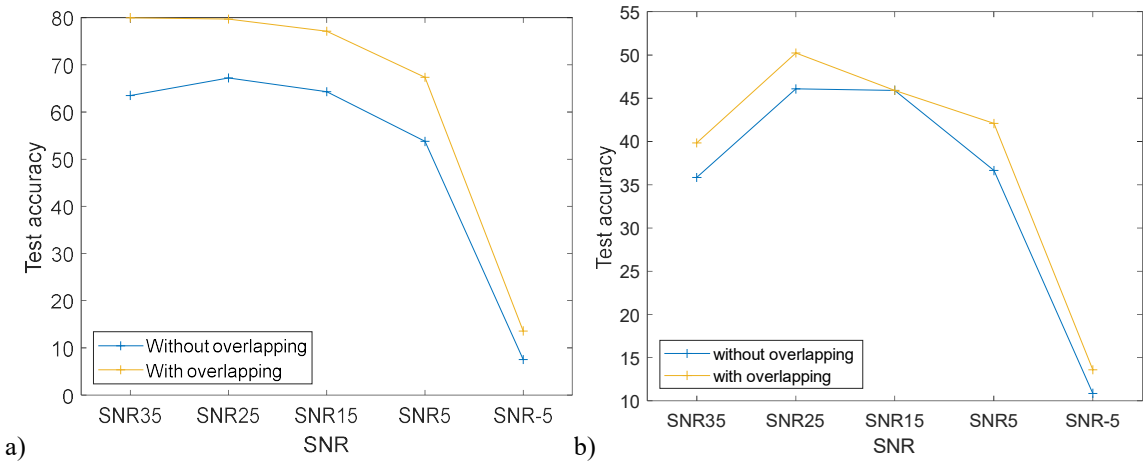


Fig. 5.2 a) Azimuth accuracy without and with overlapping, b) Elevation accuracy without and with overlapping

emphasized in the wavelet scattering matrices. The configuration of the CNN was carried out by empirically testing different designs of the number and type of layers by evaluating their performance according to the accuracy obtained on a data segment reserved for the test. A validation dataset is used to monitor the model's performance during the training and to adjust the model hyperparameters to find the optimal design. When the accuracy of the validation dataset does not increase during the training phase, the training process is terminated after a given number of epochs. The model's structure is then changed to test if the modifications increase the performance. This technique is continued until the new CNN yields acceptable results. To counter overfitting, the suggested CNN design is kept as compact as feasible [59]. Regarding computing and memory resources, the necessary time for training and prediction is proportional to the number of layers. The WS-CNN convolution architecture layers are shown in Fig. 5.3.

1	"	Image Input	54×50×2 images with 'zerocenter' normalization
2	"	Convolution	32 3×3 convolutions with stride [1 1] and padding 'same'
3	"	Batch Normalization	Batch normalization
4	"	ReLU	ReLU
5	"	Max Pooling	2×2 max pooling with stride [1 1] and padding [0 0 0 0]
6	"	Convolution	64 3×3 convolutions with stride [1 1] and padding 'same'
7	"	Batch Normalization	Batch normalization
8	"	ReLU	ReLU
9	"	Max Pooling	2×2 max pooling with stride [1 1] and padding [0 0 0 0]
10	"	Convolution	96 3×3 convolutions with stride [1 1] and padding 'same'
11	"	Batch Normalization	Batch normalization
12	"	ReLU	ReLU
13	"	Max Pooling	2×2 max pooling with stride [1 1] and padding [0 0 0 0]
14	"	Convolution	128 3×3 convolutions with stride [1 1] and padding 'same'
15	"	Batch Normalization	Batch normalization
16	"	ReLU	ReLU
17	"	Max Pooling	2×2 max pooling with stride [1 1] and padding [0 0 0 0]
18	"	Fully Connected	25 fully connected layer
19	"	Softmax	softmax
20	"	Classification Output	crossentropyex

Fig. 5.3 The CNN architecture layers of WS-CNN in classification model

The scattergrams are first rearranged in a 3D configuration of 54 × 50 × 2, the 54 frequencies, the 50 time-steps and the two channels for the left and right ears. The CNN network comprises four repetitions of a layer block, each with different filters depending on their sequential order. Each block consists of a convolutional layer, a batch normalizations layer, an activation function, and a max pooling layer. The convolution layer in the first block has 32 channels, the size of the kernel is 3 x 3, the padding is same, and the stride is 1. The configuration of all the convolutional layers is the same as for the first layer except for the channel sizes, which are 64, 96, and 128 for the second, third and fourth convolutional layers, respectively. A batch normalization layer [60] is used to improve the stability of the CNN network, and Rectified Linear Unit (ReLU) activation function [59] is used after each convolutional layer. After the four blocks, a flatten layer is used to flatten the output of the last max pooling to a feature vector. For the classification,

the size of the fully connected layer is the same number as the amount of azimuth or elevation. A classification output layer of type *crossentropyex* in MATLAB is added to obtain a classification output type. Table 5.1 shows the hyperparameter configurations.

Table 5.1 The settings of the training option used in the WS-CNN Classification

Hyperparameter	Configuration
Optimizer	Adam
Max Epochs	30
Learning Rate	1e-2
Learn Rate Drop Period	5
Output Network	Best validation lost
Execution Environment	GPU
Validation Frequency	250

5.3 Results and Discussion WS-CNN Classification

The WS-CNN evaluation was performed with two types of source signal $s(n)$, uniform and Gaussian, and with three types of additive noise, Gaussian, speech babble and F16. A new realization of the source signal and noise is made at each position. A sequence of 0.2 s and a sampling frequency of 16 kHz is considered, similar conditions as in Chapter 4. To ensure a fair comparison in this chapter, the results are compared with the reference method TF-2CNN with the same generated signals. A classification evaluation is done in this section.

The classification uses a learning phase of 5 sequences of 0.2s for a total of 1s at each 25 × 25 positions, 25 in azimuth and 25 in elevation totalling 625 loudspeaker locations. New sequences of 5 × 0.2 s (1 s) are performed for validation. SNR levels of 30, 20, 10, and 0 dB are used for each type of noise. For the test (or generalization), 15 new sequences

of 0.2 s per position are carried out, totalling a sequence of 3 s for the test for SNR levels of 35, 25, 15, 5 and -5 dB. Thus, we perform both classification tests with new realizations for SNR levels not used during the learning phase and validation.

The results are shown in Table 5.2 and Table 5.3 and also in Fig. 5.4 and Fig. 5.5. We observe the following points:

- The two methods, TF-2CNN and WS-CNN, offer similar performance results in azimuth and nearly 100% accuracy. In other words, this means that both ways estimate all the sound source locations correctly for all the locations tested. The speech babble and F16 noises present the maximum performance, whether with a uniform or Gaussian source signal. We observe a difference between TF-2CNN and WS-CNN for the Gaussian noise at -5dB. TF-2CNN presents better performance.
- In elevation, the TF-2CNN method presents a better and more marked classification performance for the case with Gaussian noise, whether with a uniform or a Gaussian source signal.

In summary, TF-2CNN presents better performance than WS-CNN in elevation. Also, the WS-CNN method archives maximum performance in azimuth estimation.

Table 5.2 WS-CNN and TF-2CNN classification elevation test accuracy for uniform source and noise combination with 1 s signal training and validation and 3 s signal for the test with 625 SSL localization

Noise	SNR	AZIMUTH				ELEVATION			
		TF-2CNN		WS-CNN		TF-2CNN		WS-CNN	
		RRMSE	Accuracy (%)	RRMSE	Accuracy (%)	RRMSE	Accuracy (%)	RRMSE	Accuracy (%)
Gaussian	35	0.0000	100.00	0.0000	100.00	0.0393	98.62	0.0884	74.85
	25	0.0000	100.00	0.0000	100.00	0.0247	98.19	0.0855	75.95
	15	0.0000	100.00	0.0022	99.97	0.0786	91.94	0.0940	74.12
	5	0.0022	99.97	0.0193	97.96	0.3489	64.34	0.1780	60.28
	-5	0.1411	86.62	0.2290	55.99	0.9667	14.57	0.7646	6.77
Babble	35	0.0000	100.00	0.0000	100.00	0.0106	99.48	0.0251	97.05
	25	0.0000	100.00	0.0000	100.00	0.0095	99.45	0.0255	97.14
	15	0.0000	100.00	0.0000	100.00	0.0080	99.56	0.0241	97.39
	5	0.0000	100.00	0.0000	100.00	0.0089	99.52	0.0260	96.87
	-5	0.0013	99.99	0.0028	99.95	0.1114	97.40	0.0331	96.01
F16	35	0.0000	100.00	0.0000	100.00	0.0074	99.69	0.0370	93.21
	25	0.0000	100.00	0.0000	100.00	0.0077	99.66	0.0378	93.03
	15	0.0000	100.00	0.0000	100.00	0.0104	99.39	0.0362	93.06
	5	0.0000	100.00	0.0000	100.00	0.0438	96.36	0.0528	88.85
	-5	0.0253	99.48	0.0240	97.28	0.3936	71.37	0.1467	63.06
	AVERAGE	**0.0113**	**99.07**	**0.0185**	**96.74**	**0.14**	**88.64**	**0.1103**	**80.51**

Table 5.3 WS-CNN and TF-2CNN classification azimuth test accuracy for Gaussian source and noise combination with 1 s signal training and validation and 3 s signal for the test with 625 SSL localization

Noise	SNR	AZIMUTH				ELEVATION			
		TF-2CNN		WS-CNN		TF-2CNN		WS-CNN	
		RRMSE	Accuracy (%)	RRMSE	Accuracy (%)	RRMSE	Accuracy (%)	RRMSE	Accuracy (%)
Gaussian	35	0.0000	100.00	0.0000	100.00	0.0234	97.47	0.0754	80.80
	25	0.0000	100.00	0.0000	100.00	0.0242	97.01	0.0922	79.71
	15	0.0000	100.00	0.0013	99.99	0.0691	89.95	0.1082	75.15
	5	0.0096	99.75	0.0211	97.98	0.3613	62.26	0.1924	58.09
	-5	0.1854	82.69	0.2147	63.35	0.9590	14.52	0.6954	13.12
Babble	35	0.0000	100.00	0.0000	100.00	0.0094	99.52	0.0254	96.52
	25	0.0000	100.00	0.0000	100.00	0.0095	99.51	0.0256	96.31
	15	0.0000	100.00	0.0000	100.00	0.0065	99.74	0.0272	96.22
	5	0.0000	100.00	0.0000	100.00	0.0142	99.55	0.0279	96.36
	-5	0.0127	99.99	0.0000	100.00	0.1482	97.51	0.0358	94.22
F16	35	0.0000	100.00	0.0000	100.00	0.0022	99.97	0.0279	95.80
	25	0.0000	100.00	0.0000	100.00	0.0043	99.87	0.0296	95.69
	15	0.0000	100.00	0.0000	100.00	0.0074	99.66	0.0314	94.57
	5	0.0000	100.00	0.0000	100.00	0.0393	97.27	0.0467	90.33
	-5	0.0250	99.71	0.0246	97.75	0.3818	74.40	0.1959	65.64
	AVERAGE	**0.0155**	**98.81**	**0.0174**	**97.27**	**0.14**	**88.55**	**0.1091**	**81.90**

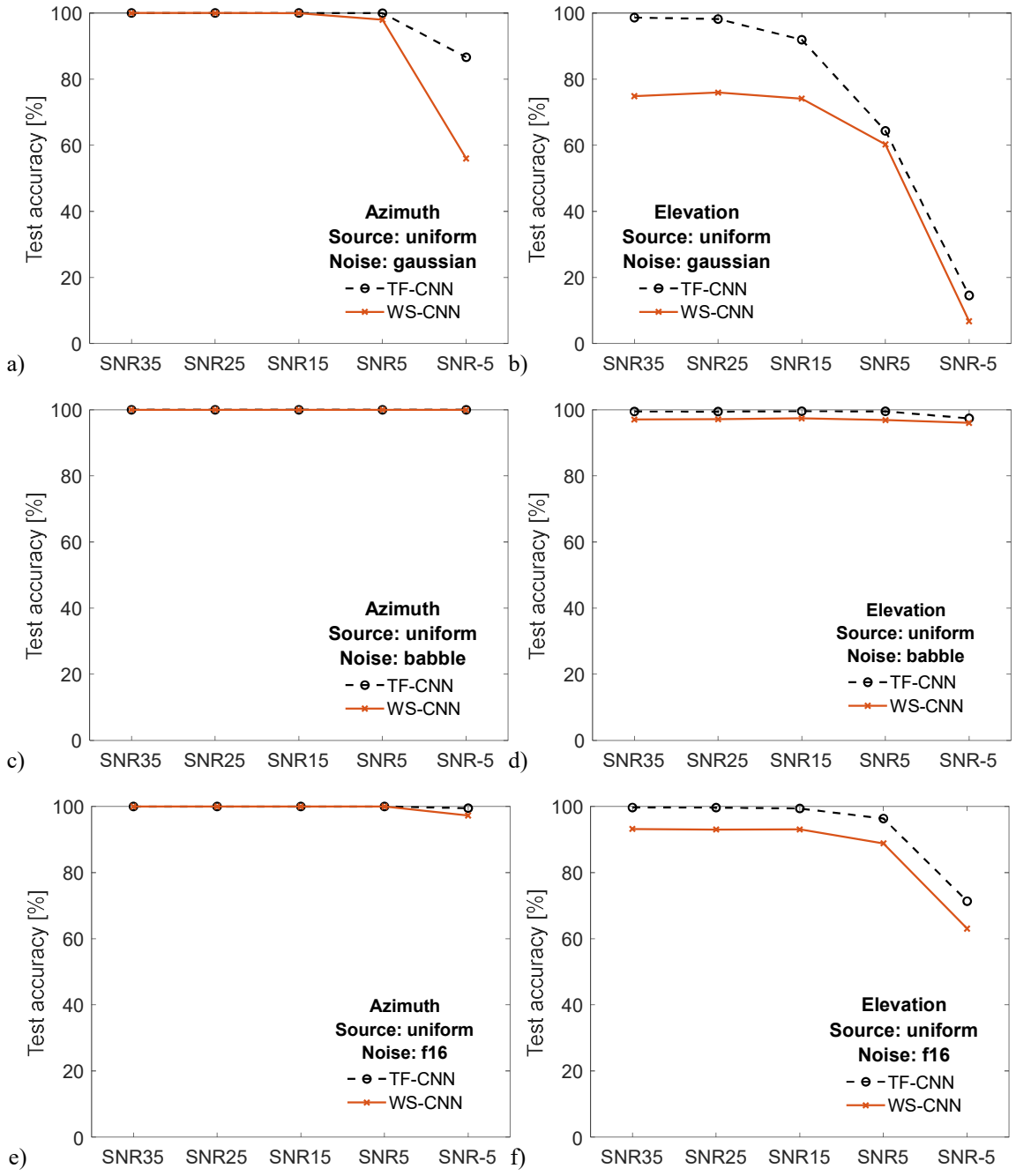


Fig. 5.4 WS-CNN and TF-2CNN classification test accuracy in azimuth and elevation with a uniform source for a gaussian, babble and F16 noise with 1s signal training and validation and 3s signal for the test with 625 SSL localization

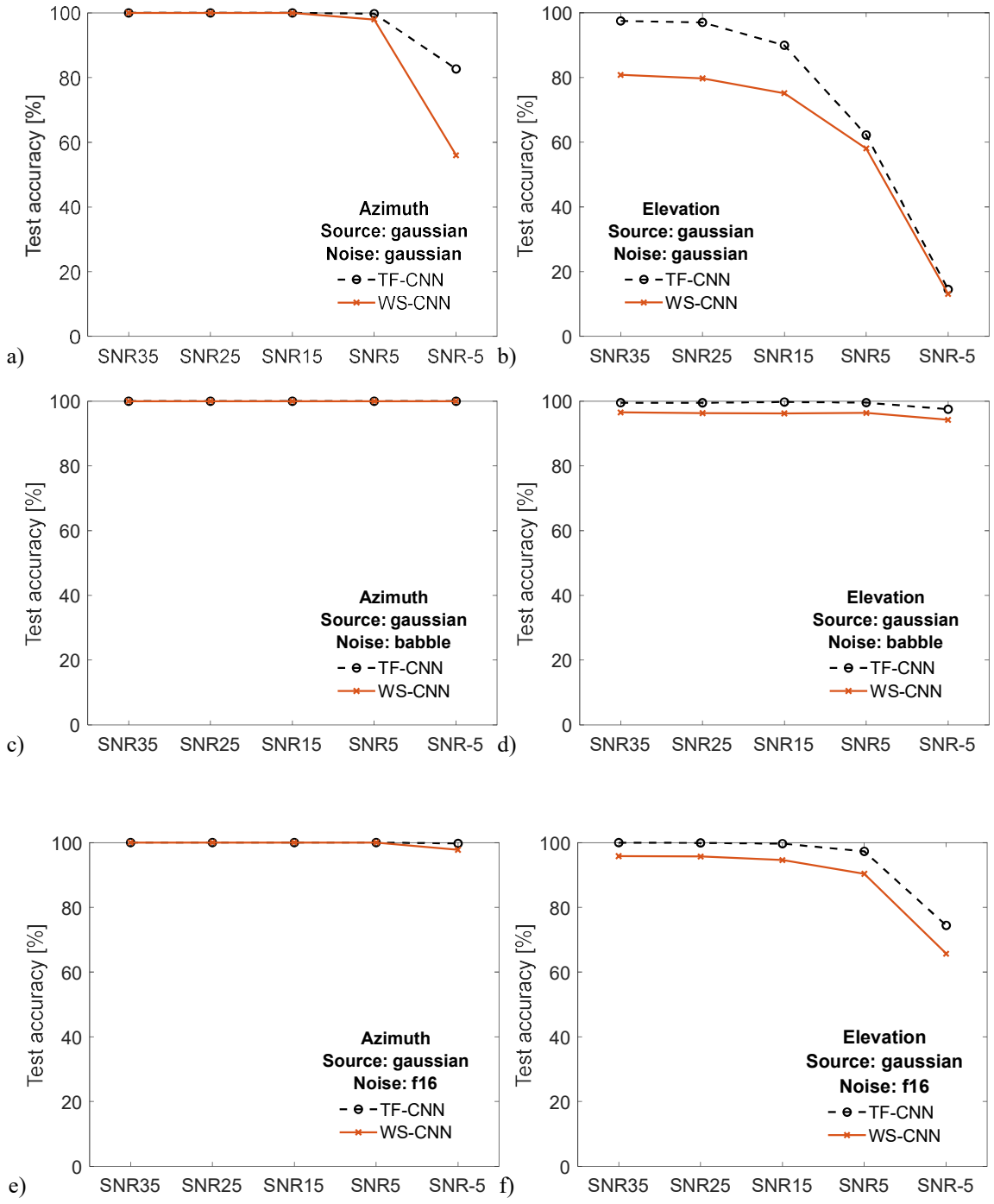


Fig. 5.5 WS-CNN and TF-2CNN classification test accuracy in azimuth and elevation with Gaussian source for a gaussian, babble and F16 noise with 1 s signal training and validation and 3 s signal for the test with 625 SSL localization

5.4 Proposed WS-CNN Regression Architecture

In contrast to earlier research, a CNN is employed as a regression function rather than a classification function since it is more suited to continuous variable output like SSL. The WS-CNN regression network architecture is almost identical to the WS-CNN classification presented in the previous section. The difference is that the SoftMax layer is removed in the last layers, and the fully connected layer is changed from 25 to only one weight to obtain a regression-type output. The final layer is now a regression output of type mean square error in MATLAB. After optimization, the WS-CNN regression architecture layers are shown in Fig. 5.6.

1	"	Image Input	54×50×2 images with 'zerocenter' normalization
2	"	Convolution	32 5×5 convolutions with stride [1 1] and padding 'same'
3	"	Batch Normalization	Batch normalization
4	"	ReLU	ReLU
5	"	Max Pooling	2×2 max pooling with stride [1 1] and padding [0 0 0 0]
6	"	Convolution	64 3×3 convolutions with stride [1 1] and padding 'same'
7	"	Batch Normalization	Batch normalization
8	"	ReLU	ReLU
9	"	Max Pooling	2×2 max pooling with stride [1 1] and padding [0 0 0 0]
10	"	Convolution	96 3×3 convolutions with stride [1 1] and padding 'same'
11	"	Batch Normalization	Batch normalization
12	"	ReLU	ReLU
13	"	Max Pooling	2×2 max pooling with stride [1 1] and padding [0 0 0 0]
14	"	Convolution	128 3×3 convolutions with stride [1 1] and padding 'same'
15	"	Batch Normalization	Batch normalization
16	"	ReLU	ReLU
17	"	Max Pooling	2×2 max pooling with stride [1 1] and padding [0 0 0 0]
18	"	Fully Connected	1 fully connected layer
19	"	Regression Output	mean-squared-error

Fig. 5.6 The CNN architecture layers of WS-CNN in regression model

Table 5.4 shows the hyperparameter configurations used with the WS-CNN regression.

Table 5.4 The settings of the training option used in the WS-CNN Regression

Hyperparameter	Configuration
Optimizer	Adam
Max Epochs	30
Learning Rate	1e-2
Learn Rate Drop Period	5
Output Network	Best validation lost
Execution Environment	GPU
Validation Frequency	250

5.5 Results and Discussion WS-CNN Regression

The regression position estimation represents a challenge more difficult than classification. Contrarily to classification, the regression represents an analogue estimation. In this section, for the regression evaluation, we apply the same procedure as the classification, except that we consider only 13 positions in azimuth and elevation, a total of 169 positions, for the learning and validation phases. The test is carried out on 25 × 25 (625) locations. The generation of the sequences is the same as for the classification at the positions used and for the two source signals, uniform and Gaussian, and the three types of noise, Gaussian, speech babble and F16. The results relate to both TF-2CNN and WS-CNN methods.

Compared to classification, for regression, we have to define analog performance metrics: mean absolute error (MAE), ±5° and ±2.5°. MAE computes the difference between the expected y and actual $\hat{y}$ values as described by the equation (18).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (18)$$

Where N is the total number of positions evaluated, and i corresponds to the loudspeaker locations. Also, to classify, ±5° and ±2.5° were selected to be close enough to the exact SSL angle values as two error margin values for the estimation of the SSL.

Fig. 5.5, Fig. 5.6 and Fig. 5.7 present azimuth and elevation performance results in terms of ±5° and ±2.5° accuracy as well as MAE, respectively. For Fig. 5.7 on the MAE, take note of the logarithmic scale of the results for better observation in both elevation and azimuth on the same graph. Table 5.5, Table 5.6, Table 5.7 and Table 5.8 show the performance results for TF-2CNN and WS-CNN for uniform and gaussian source signals with three types of additive noise (Gaussian, speech babble and F16).

From these results, we draw the following main observations:

- In azimuth, the proposed method WS-CNN outperform the TF-2CNN in all type of noise and source signal. The gain of WS-CNN is, on average, about 50% more than TF-2CNN for azimuth positions. The WS-CNN presents for all source signal and noise types an accuracy of over 82% inside ±5° compared with about 60% for TF-2CNN.
- As mentioned before, the elevation case is a great challenge, and the accuracy is low for all methods. In elevation, like for azimuth, the proposed method WS-CNN outperforms the TF-2CNN. However, the WS-CNN presents an average accuracy of 34% compared to 15% for TF-2CNN.

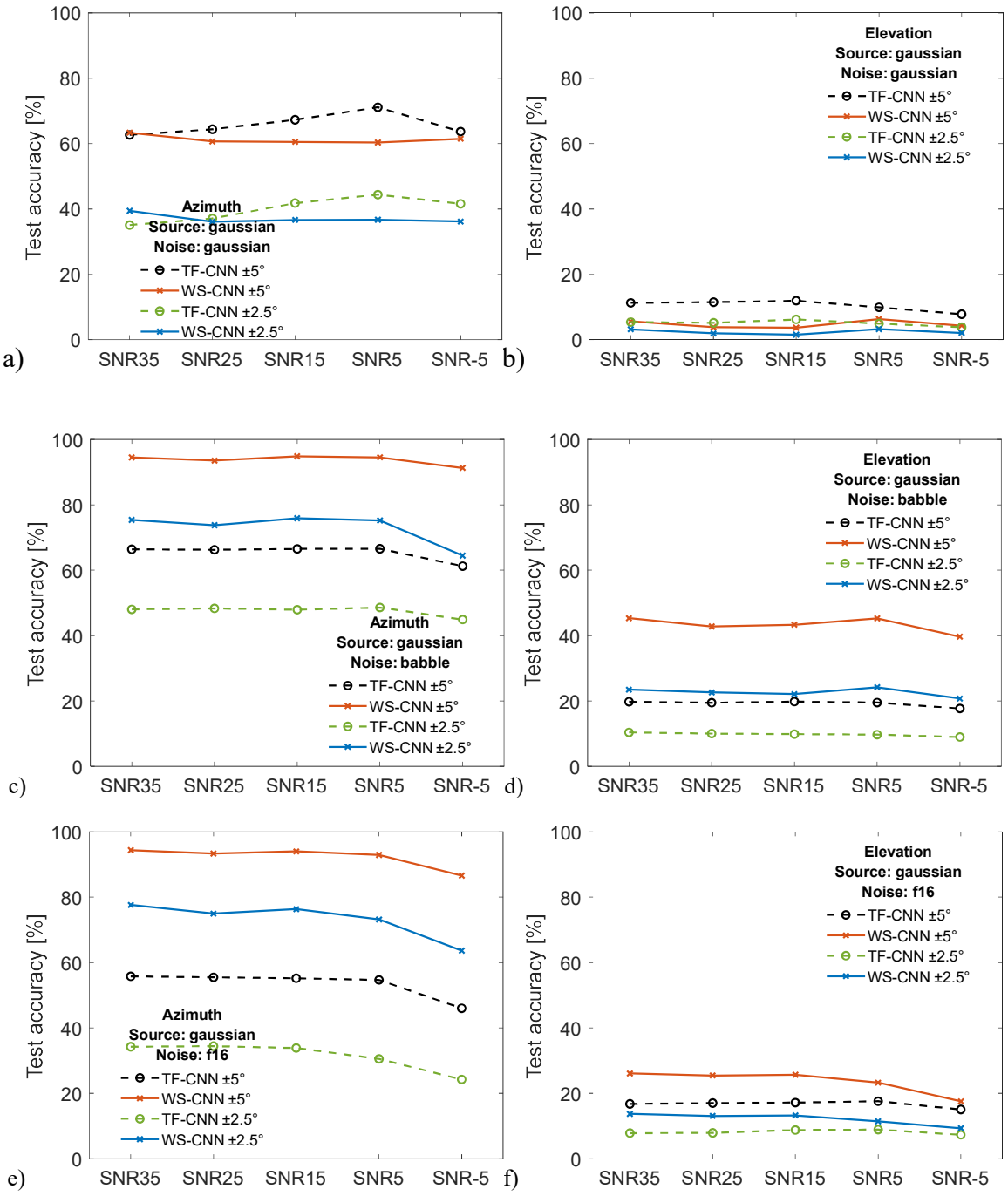


Fig. 5.5 WS-CNN and TF-2CNN regression test accuracy in azimuth and elevation for uniform source and noise combination using 1s 169 angles for training and validation and 3s 625 angles for the test

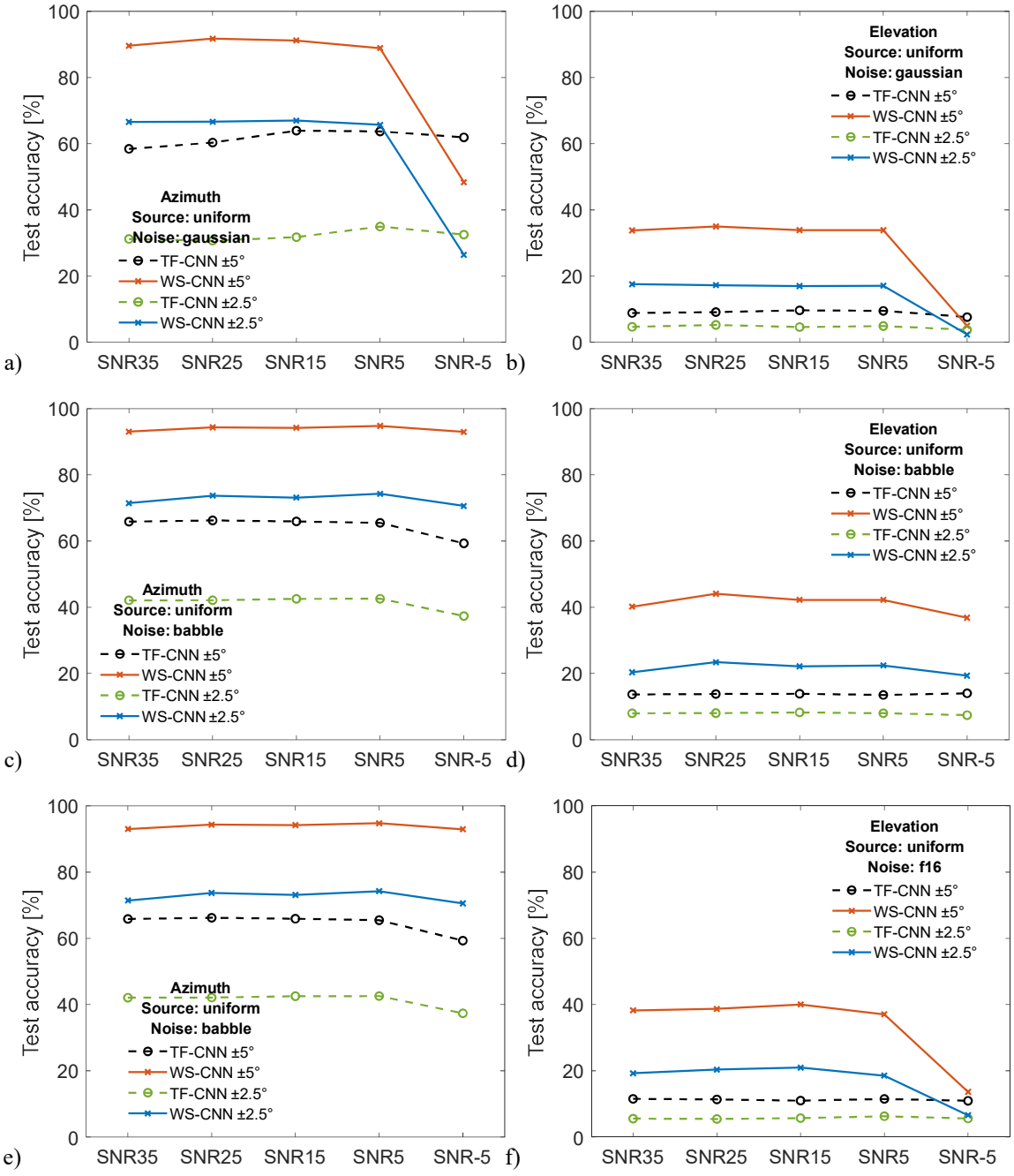


Fig. 5.6 WS-CNN and TF-2CNN regression test accuracy in azimuth and elevation for Gaussian source and noise combination using 1s 169 angles for training and validation and 3s 625 angles for the test

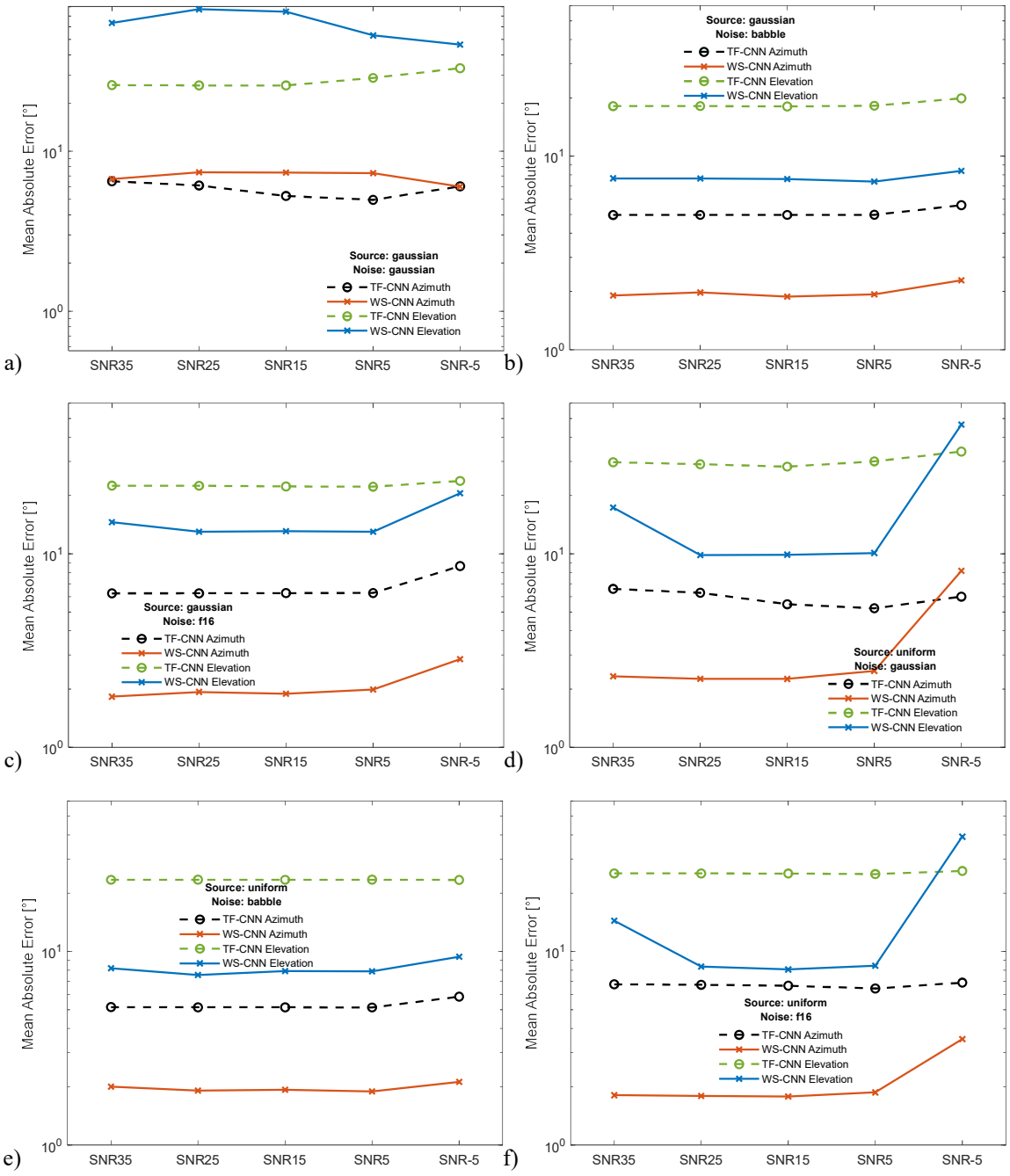


Fig. 5.7 WS-CNN and TF-2CNN regression mean absolute error in azimuth and elevation for multiple sources and noise combination using 1s 169 angles for training and validation and 3s 625 angles for the test

Table 5.5 TF-2CNN regression accuracy in azimuth and elevation with uniform signal for a gaussian, babble and F16 noise using 1 s 169 angles for training and validation and 3 s 625 angles for the test

Noise	SNR	Azimuth				Elevation			
		MAE (°)	RRMSE	Accuracy (%)		MAE (°)	RRMSE	Accuracy (%)	
				±5°	±2.5°			±5°	±2.5°
Gaussian	35	6.59	0.2367	58.44	31.22	29.71	0.7650	8.87	4.69
	25	6.29	0.2233	60.34	30.71	29.02	0.7478	9.14	5.28
	15	5.49	0.1930	63.95	31.79	28.21	0.7365	9.69	4.63
	5	5.23	0.1856	63.72	34.99	30.03	0.7732	9.53	4.94
	-5	6.01	0.2290	61.92	32.55	33.75	0.8508	7.64	3.81
Babble	35	5.16	0.1839	65.86	42.10	23.45	0.6411	13.67	7.98
	25	5.15	0.1837	66.23	42.12	23.48	0.6416	13.80	8.03
	15	5.15	0.1838	65.93	42.54	23.46	0.6415	13.88	8.25
	5	5.14	0.1834	65.49	42.58	23.48	0.6409	13.51	8.00
	-5	5.84	0.2023	59.32	37.37	23.41	0.6397	14.02	7.42
F16	35	6.76	0.2415	52.58	32.74	25.31	0.6622	11.53	5.57
	25	6.73	0.2405	53.29	32.55	25.30	0.6620	11.36	5.45
	15	6.65	0.2397	55.71	32.06	25.27	0.6607	10.99	5.72
	5	6.43	0.2302	57.17	31.52	25.10	0.6575	11.50	6.28
	-5	6.90	0.2360	51.98	28.81	26.03	0.6808	10.95	5.61
	AVERAGE	**5.97**	**0.2128**	**60.13**	**35.04**	**26.33**	**0.6934**	**11.34**	**6.11**

Table 5.6 WS-CNN regression accuracy in azimuth and elevation with uniform signal for a gaussian, babble and F16 noise using 1 s 169 angles for training and validation and 3 s 625 angles for the test

Noise	SNR	Azimuth				Elevation			
		MAE (°)	RRMSE	Accuracy (%)		MAE (°)	RRMSE	Accuracy (%)	
				±5°	±2.5°			±5°	±2.5°
Gaussian	35	2.33	0.0833	89.59	66.57	17.34	0.3012	33.81	17.57
	25	2.26	0.0803	91.72	66.65	9.84	0.2855	35.02	17.29
	15	2.26	0.0807	91.19	66.99	9.89	0.2825	33.89	17.01
	5	2.48	0.0921	88.87	65.70	10.10	0.2911	33.89	17.12
	-5	8.16	0.2817	48.38	26.44	46.48	1.1658	4.99	2.39
Babble	35	2.00	0.0733	93.03	71.45	8.18	0.2321	40.19	20.37
	25	1.91	0.0708	94.36	73.73	7.56	0.2183	44.09	23.40
	15	1.93	0.0710	94.20	73.13	7.92	0.2282	42.24	22.15
	5	1.89	0.0703	94.76	74.27	7.89	0.2288	42.25	22.43
	-5	2.12	0.0894	92.95	70.58	9.39	0.3104	36.82	19.33
F16	35	1.81	0.0701	94.71	76.38	14.41	0.2222	38.25	19.26
	25	1.79	0.0695	95.10	76.22	8.36	0.2347	38.72	20.35
	15	1.78	0.0688	95.23	76.55	8.07	0.2270	40.05	20.97
	5	1.87	0.0724	94.34	74.49	8.44	0.2346	37.06	18.55
	-5	3.53	0.1493	79.98	57.92	39.14	1.4236	13.65	6.60
	AVERAGE	**2.54**	**0.0949**	**89.23**	**67.80**	**14.20**	**0.3924**	**34.33**	**17.65**

Table 5.7 TF-2CNN regression accuracy in azimuth and elevation with Gaussian signal for a gaussian, babble and F16 noise using 1 s 169 angles for training and validation and 3 s 625 angles for the test

Noise	SNR	Azimuth				Elevation			
		MAE (°)	RRMSE	Accuracy (%)		MAE (°)	RRMSE	Accuracy (%)	
				±5°	±2.5°			±5°	±2.5°
Gaussian	35	6.49	0.2391	62.67	35.09	25.91	0.6793	11.30	5.42
	25	6.11	0.2245	64.41	37.11	25.78	0.6770	11.51	5.19
	15	5.25	0.1944	67.31	41.80	25.77	0.6822	11.96	6.23
	5	4.97	0.1959	71.14	44.41	28.73	0.7416	9.95	4.95
	-5	6.03	0.2402	63.70	41.61	33.03	0.8312	7.83	3.83
Babble	35	4.97	0.1901	66.39	48.02	18.13	0.5157	19.78	10.39
	25	4.97	0.1901	66.24	48.31	18.14	0.5159	19.46	10.02
	15	4.97	0.1899	66.52	47.91	18.06	0.5142	19.80	9.87
	5	4.98	0.1905	66.55	48.58	18.21	0.5161	19.52	9.71
	-5	5.58	0.2073	61.25	44.90	19.90	0.5488	17.72	8.97
F16	35	6.25	0.2194	55.77	34.24	22.49	0.6581	16.75	7.80
	25	6.25	0.2189	55.48	34.41	22.46	0.6567	17.00	7.87
	15	6.27	0.2172	55.17	33.85	22.29	0.6505	17.15	8.75
	5	6.27	0.2123	54.67	30.52	22.22	0.6431	17.57	8.90
	-5	8.64	0.2989	46.01	24.22	23.79	0.6664	15.04	7.29
	AVERAGE	**5.87**	**0.2152**	**61.55**	**39.67**	**22.99**	**0.6331**	**15.49**	**7.68**

Table 5.8 WS-CNN regression accuracy in azimuth and elevation with Gaussian signal for a gaussian, babble and F16 noise using 1 s 169 angles for training and validation and 3 s 625 angles for the test

Noise	SNR	Azimuth				Elevation			
		MAE (°)	RRMSE	Accuracy (%)		MAE (°)	RRMSE	Accuracy (%)	
				±5°	±2.5°			±5°	±2.5°
Gaussian	35	6.70	0.2572	63.37	39.46	63.36	1.8355	5.66	3.21
	25	7.38	0.2803	60.67	36.13	77.13	2.0500	3.86	1.96
	15	7.36	0.2797	60.55	36.66	74.39	1.9890	3.69	1.56
	5	7.29	0.2772	60.34	36.71	52.92	1.5281	6.34	3.27
	-5	6.02	0.2252	61.48	36.18	46.33	1.1715	4.32	2.11
Babble	35	1.90	0.0712	94.49	75.37	7.68	0.2125	45.31	23.47
	25	1.97	0.0726	93.53	73.77	7.67	0.2198	42.79	22.62
	15	1.88	0.0711	94.85	75.88	7.61	0.2176	43.34	22.11
	5	1.93	0.0717	94.54	75.21	7.39	0.2145	45.25	24.22
	-5	2.28	0.0794	91.29	64.46	8.39	0.2865	39.65	20.75
F16	35	1.83	0.0744	94.36	77.59	14.57	0.3656	26.07	13.70
	25	1.93	0.0774	93.32	74.94	12.98	0.3602	25.43	13.07
	15	1.89	0.0764	93.99	76.31	13.10	0.3636	25.65	13.25
	5	1.99	0.0789	92.91	73.16	13.01	0.3598	23.25	11.41
	-5	2.85	0.1303	86.59	63.63	20.58	0.6016	17.53	9.31
	AVERAGE	**3.68**	**0.1415**	**82.42**	**61.03**	**28.48**	**0.7851**	**23.88**	**12.40**

Regarding the MAE criteria, the proposed WS-CNN outperform the TF-2CNN in azimuth estimation. The azimuth MAE is, on average, about 3° for WS-CNN and 6° for TF-2CNN. The gain is more than double for the proposed method. However, in elevation, the TF-2CNN has a lower MAE.

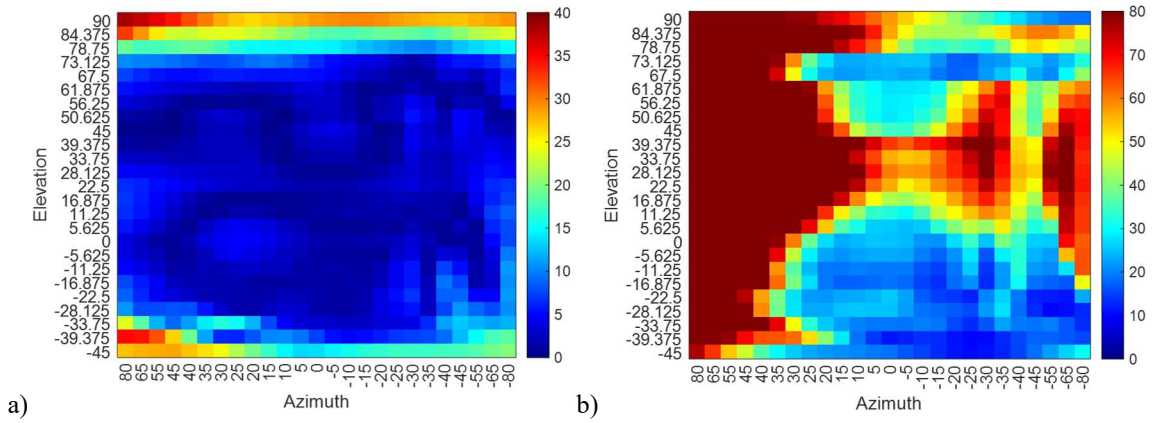


Fig. 5.7 Mean absolute error (MAE) for azimuth (a) and elevation (b) SSL estimation with a uniform sound source using 13 × 13 (169) angles of 0.2 s time sequence for the training-validation phase and 25 × 25 (625) angles at 35 dB SNR for the test

Fig. 5.7 shows the mean absolute average distribution of the SSL estimation for the test phase with the WS-CNN method for a uniform sound source in regression with a 35 dB SNR level. In Fig. 5.7a, it can be observed that the MAE values are lower when the elevation angle is closer to 0 degrees and increase when the elevation reaches 80 or -45 degrees. The MAE does not vary much according to the azimuth angle from -80 to 80 degrees. Fig. 5.7b is the MAE for elevation estimation. Due to the cone of confusion effect, the elevation estimation is much more challenging than the azimuth estimation, so it is

expected to obtain higher MAE values for elevation estimation. Also, Fig. 5.7a and Fig. 5.7b show an overall symmetry centred at 0 degrees in azimuth.

More details on result synthesis and discussions follow in Chapter 6.

This chapter presented the proposed wavelet scattering method as a feature extraction followed by a CNN for each angle output in azimuth and elevation. We could see the advantage of applying segmented input images with a 50% temporal overlap. A cost in complexity but a significant gain in precision. We used the same simulation conditions as Chapter 4, devoted to WS-LSTM, to ensure a fair comparison.

In comparison with TF-2CNN, we demonstrated gains in the accuracy of the WS proposal, followed by CNNs to estimate azimuth and elevation. Detailed discussions and a synthesis of the results of the TF-2CNN, WS-LSTM and WS-CNN methods are carried out in Chapter 6.

Chapter 6: Result Synthesis and Discussions

This Chapter is dedicated to discussing the results obtained from the proposed methods in comparison with a recent method used as a reference which was also carried out for this thesis to compare the results adequately. As presented in Chapter 3, feature extraction is important to increase the performance of loudspeaker position estimation. The proposed extraction feature is based on the scattering decomposition wavelet. This method makes it possible to extract the information in the time and frequency domain, thus forming an image for each sound frame coming from both the left and right ear. Two machine learning methods for SSL estimation from images were presented and studied in more depth, the first WS-LSTM and the WS-CNN as the second method.

6.1 Methods summarized

This section is a review of the main observations of the results obtained using the two proposed methods (WS-LSTM and WS-CNN) in comparison with the TF-2CNN method:

- The proposals respond to the project's objective: to produce an analog estimate of the SSL position from machine learning methods in regression mode.
- Azimuth estimation is of greater interest for general applications. Generally, classification or regression is difficult in elevation.
- To compare the performance, we applied the TF-2CNN method using feature extraction based on the time-frequency (TF) method [24]. The TF feature extraction consists in applying the FFT by temporally segmenting the left-right SSL listening

frame. The image construction (feature extraction) is carried out in the Fourier domain by the ratio in magnitude and phase between the left and right ear signals. Contrarily to [24], in our TF-2CNN, we proposed to use one CNN for each angle estimation, azimuth and elevation. In [24], the authors proposed only one CNN for two output estimators.

- Since the WS-LSTM method is a recursive method, it has the advantage of an online estimate. In other words, in the estimation (or test) phase, as soon as the first sample of the WS image is obtained, an estimate of the location can be obtained as output. The image presents a time base equal to the time base of the listening signal, and at each time point, a frequency sample *Mx* is shown. Thus, the estimation quality improves at each time point of *Mx* frequency samples presented at the input of the two LSTMs.
- Two types of sound sources are used to compare the performance methods: uniform and Gaussian. We consider three types of additive noises, Gaussian, speech babble and F16, with four levels of SNR for the training and validation phases: 30, 20, 10 and 0 dB. To test, we consider 5 SNR levels: 35, 25, 15, 5 and -5 dB. In classification scenarios, we used 25 × 25 (625) angles for training and validation and tested on 25 × 25 (625) angles. In regression scenarios, we used 13 × 13 (169) angles for the training and validation phases and tested on 25 × 25 (625) angles. In all cases, a new (different) realization of the noise sequence is done at each angle, each sound source signal, each additive noise and each SNR level.

- In general, all estimation methods used at an SNR level of -5 dB exhibit reduced estimation results compared to higher SNRs. This reduction is expected behaviour since the noise power is higher than the information signal. It is also an extrapolation estimate, which means at an SNR level outside of those used during the learning phase. The lower SNR in the learning phase was 5dB, and the test was done with lower SNR at -5dB.

6.2 Classification Results

Table 6.1 presents a summary from Table 4.1, Table 4.2, Table **5.2** and Table 5.3 for classification results of the performance accuracy for each method based on the average for all additive noise (Gaussian, speech babble and F16) in azimuth and elevation with uniform and Gaussian sound sources. The best performance results are in bold for each sound source and type of additive noise in azimuth and elevation. For WS-LSTM, the gaussian and speech babble noise is considered, and no result is obtained for the gaussian sound source in this work. Table 6.2 summarizes Table 6.1 to synthesize the global performance results to show the average for all types and levels of noise for each method.

From Table 6.2, we can summarize the following principal observations:

- The WS-CNN presents similar performances to TF-2CNN in classification mode. The results of TF-2CNN show good performance in azimuth but weak in elevation. In addition, it showed a poor performance in regression mode in both azimuth and elevation.

- From Table 6.1, we observed better performance for TF-2CNN in azimuth compared to both others, but the WS-CNN is close to TF-2CNN with good performance in general in azimuth. However, WS-LSTM and WS-CNN lost performance in elevation for gaussian noise.
- According to Table 6.2, the classification of WS-CNN is slightly lower than TF-2CNN. We note a reduction of 1.55% in azimuth and 7.5% in elevation. Those differences remain low, given that accuracy of 97.27% is reached on average for all levels and types of noise and an accuracy of 81.90% in elevation.
- In Table 5.2 and Table 5.3, all performances are close to 100% for TF-2CNN and WS-CNN for uniform and Gaussian sound source signals.

Table 6.1 Results synthesis for classification of Gaussian, babble and F16 noises

Noise	Method	Azimuth		Elevation	
		RRMSE	Accuracy (%)	RRMSE	Accuracy (%)
Uniform Sound Source					
Gaussian	TF-2CNN	**0.0287**	**97.32**	0.2874	**72.24**
	WS-LSTM	0.1571	82.55	**0.2214**	59.52
	WS-CNN	0.0501	90.78	0.2421	58.39
Speech babble	TF-2CNN	**0.0003**	**100.00**	0.0297	99.08
	WS-LSTM	0.0642	96.92	0.1043	81.61
	WS-CNN	0.0006	99.99	**0.0006**	**99.99**
F16	TF-2CNN	0.0051	**99.90**	0.0926	**93.29**
	WS-LSTM	-	-	-	-
	WS-CNN	**0.0048**	99.46	**0.0621**	86.24
Gaussian Sound Source					
Gaussian	TF-2CNN	**0.0390**	**96.49**	0.2874	**72.24**
	WS-CNN	0.0474	92.26	**0.2327**	61.37
Speech babble	TF-2CNN	0.0025	100.00	0.0376	**99.17**
	WS-CNN	**0.0000**	**100.00**	**0.0284**	95.93
F16	TF-2CNN	0.0050	**99.94**	0.0870	**94.23**
	WS-CNN	**0.0049**	99.55	**0.0663**	88.41

Table 6.2 Classification results synthesis for TF-2CNN, WS-LSTM and WS-CNN with uniform and Gaussian sound source

Method	Azimuth		Elevation	
	RRMSE	Accuracy (%)	RRMSE	Accuracy (%)
Uniform Sound Source				
TF-2CNN	**0.0113**	**99.07**	0.1380	**88.64**
WS-LSTM*	0.1106	89.73	0.1628	70.57
WS-CNN	0.0185	96.74	**0.1103**	80.51
Gaussian Sound Source				
TF-2CNN	**0.0155**	**98.81**	0.1373	**88.55**
WS-CNN	0.0174	97.27	**0.1091**	81.90

*Results for gaussian and babble noise

6.3 Regression Results

Table 6.3 summarizes Table 4.3, Table 4.4, Table 5.5, Table 5.6, Table 5.7 and Table 5.8 for regression results on the performance accuracy for each method in azimuth and elevation with uniform and gaussian sound sources. The values shown in Table 6.3 correspond to the average accuracy for all SNR levels of each additive noise (Gaussian, speech babble and F16). The best performance results are in bold for each type of additive noise in azimuth and elevation.

Table 6.4 summarizes Table 6.3 to synthesize the global performance results to show the average for all types and levels of noise for each method.

Table 6.3 Results synthesis for regression for Gaussian, babble and F16 noises

Noise	Method	Azimuth				Elevation			
		MAE (°)	RRMSE	Accuracy (%)		MAE (°)	RRMSE	Accuracy (%)	
				±5°	±2.5°			±5°	±2.5°
Uniform Sound Source									
Gaussian	TF-2CNN	5.92	0.2135	61.67	32.25	30.14	0.7747	8.97	4.67
	WS-LSTM	**2.83**	**0.1136**	**82.77**	**66.41**	**9.08**	**0.2512**	**50.80**	**32.14**
	WS-CNN	3.50	0.1236	81.95	58.47	18.73	0.4652	28.32	14.28
Speech babble	TF-2CNN	5.28	0.1874	64.57	41.34	23.45	0.6410	13.78	7.94
	WS-LSTM	2.06	0.0897	88.98	**76.80**	**3.53**	**0.1063**	**74.20**	**50.69**
	WS-CNN	**1.97**	**0.0750**	**93.86**	72.63	8.19	0.2436	41.12	21.54
F16	TF-2CNN	6.69	0.2376	54.15	31.54	25.40	0.6646	11.27	5.73
	WS-LSTM	**1.95**	0.0866	90.24	**78.40**	**4.58**	**0.1332**	**66.09**	**42.43**
	WS-CNN	2.16	**0.0860**	**91.87**	72.31	15.68	0.4684	33.55	17.15
Gaussian Sound Source									
Gaussian	TF-2CNN	**5.77**	**0.2188**	**65.85**	40.00	**27.85**	**0.7223**	10.51	5.12
	WS-LSTM	6.63	0.2549	63.82	**40.57**	30.18	0.8168	**11.63**	**5.88**
	WS-CNN	6.95	0.2639	61.28	37.03	62.83	1.7148	4.77	2.42
Speech babble	TF-2CNN	5.09	0.1936	65.39	47.54	18.49	0.5221	19.26	9.79
	WS-LSTM	**1.87**	0.0882	90.24	**81.69**	**4.19**	**0.1256**	**65.73**	**45.52**
	WS-CNN	1.99	**0.0732**	**93.74**	72.94	7.75	0.2302	43.27	22.63
F16	TF-2CNN	6.74	0.2333	53.42	31.45	22.65	0.6550	16.70	8.12
	WS-LSTM	2.10	0.0959	88.87	**78.29**	**4.78**	**0.1428**	**62.73**	**41.43**
	WS-CNN	**2.10**	**0.0875**	**92.23**	73.13	14.85	0.4102	23.59	12.15

Table 6.4 Regression results synthesis for TF-2CNN, WS-LSTM and WS-CNN with uniform and Gaussian sound source

Method	Azimuth				Elevation			
	MAE (°)	RRMSE	Accuracy (%)		MAE (°)	RRMSE	Accuracy (%)	
			±5°	±2.5°			±5°	±2.5°
Uniform Sound Source								
TF-2CNN	5.97	0.2128	60.13	35.04	26.33	0.6934	11.34	6.11
WS-LSTM	**2.28**	0.0966	87.33	**73.87**	**5.73**	**0.1636**	**63.70**	**41.76**
WS-CNN	2.54	**0.0949**	**89.23**	67.80	14.20	0.3924	34.33	17.65
Gaussian Sound Source								
TF-2CNN	5.87	0.2152	61.55	39.67	22.99	0.6331	15.49	7.68
WS-LSTM	**3.53**	0.1464	80.98	**66.85**	**13.05**	**0.3617**	**46.69**	**30.94**
WS-CNN	3.68	**0.1415**	**82.42**	61.03	28.48	0.7851	23.88	12.40

The main observations of the results obtained for regression using the two proposed methods (WS-LSTM and WS-CNN) in comparison with the TF-2CNN method:

- In regression mode, the proposed feature extraction method based on WS-CNN has superior performance compared to the TF-2CNN method (see Table 5.7, Table 5.8, Table 5.2 and Table 5.3, as well as Fig. 5.4 and Fig. 5.5).
- To compare the WS-LSTM with WS-CNN, in classification mode, Table 4.1 and Table 4.2 for WS-LSTM show a performance average of 89.7% and 70.6% in azimuth and elevation, respectively. Table 5.3 for WS-CNN shows a performance average of 95.4% and 77.6% for gaussian and babble noises, respectively. A 6% and 10% performance improvement for WS-CNN is observed for azimuth and elevation, respectively. Note that for WS-LSTM (Table 4.1 and Table 4.2), the results are obtained with 0.2 s for the training phase compared to 1 s for WS-CNN.
- In the regression case, Table 6.4 for WS-LSTM shows a performance average of 87.3% and 63.7% in azimuth and elevation, respectively, at ±5° for a uniform sound. On the other side, Table 5.3 for WS-CNN show for all noise a performance average of 89.2% and 34.3% in azimuth and elevation, respectively. From Table 6.4, the TF-2CNN performance averages are 60.1% and 11.3% for azimuth and elevation with a uniform sound source and gaussian noise. The WS-LSTM outperforms the TF-2CNN and WS-CNN in elevation with performance gains of 5.6 and 3.0 times more, respectively. On the other hand, the WS-CNN outperforms other methods in azimuth with an increase of 45% and only 2% compared to TF-

2CNN and WS-LSTM, respectively. Note that the training sequence time is 0.2 s for WS-LSTM compared to 1 s for TF-2CNN and WS-CNN.

- Concerning the source noise effect, from Table 6.4, we observed a low impact on the performance of TF-2CNN and WS-CNN between uniform and Gaussian sound source signals. From a uniform to a Gaussian sound source, we observed performance of 2.4% (-36%), -7.3% (-26.7%) and 7.6% (30%) for TF-2CNN, WS-LSTM and WS-CNN in azimuth (elevation), respectively. The TF-2CNN loss in performance with the uniform sound source. From Table 6.3, the TF-2CNN loss in performance with additive gaussian.

6.4 Time Complexity Analysis

In terms of computational complexity, the WS method is more complex than the TF reference method. Table 6.5 shows a typical computation time for an azimuth estimation with a uniform sound source using 13 × 13 (169) angles of 0.2 s time sequence for the training-validation phase and 25 × 25 (625) angles at 35 dB SNR for the test. The same test conditions are done for all methods and implemented in MATLAB R2022a on a computer with an Intel Core i9-9900X CPU @3.50 GHz processor with 64 GB of DDR4 RAM and two NVIDIA GeForce RTX 2080 Ti graphics cards.

For training data generation to generate the image feature extraction for 0.2 s time input sequence for training and validation data, the TF and WS methods have average computation times of 0.56 s and 9 s, respectively. For a 0.2 s soundtrack, the WS method generates an image of 54 × 400 while TF generates an image of 320 × 18. The computation

time to generate the test date for 25 × 25 angles is 1 s and 19 s for TF and WS, respectively. The computation time to generate the data for training and testing was about 18 times longer than WS. Note that the TF feature extraction is based on FFT compiled function called in MATLAB, which uses FFTW – Fast Fourier Transform on the West [61] [62]. The WS is based on *wavelet Scattering*, *scatteringTransform* and *scattergram* MATLAB functions, and all three are interpreted functions in MATLAB, not compiled such as FFT. Using a compiled version of the code could help improve the performance of the three methods since the MATLAB implementation in this research is running in the MATLAB development environment as interpreted code. The TF-2CNN test data generation was done using compiled and optimized FFT function on a CPU called FFTW (Fast Fourier in the West) [62] [63]. And on the order side, the proposed method based on the wavelet scattering function called from MATLAB is not optimized or compiled. The computational times for test data generation are greatly influenced by these functions explaining the time difference between TF and WS computational time. For testing time, all functions are optimized for GPU executions, and all use the same computer with the same CPU and GPU.

Concerning the CNN network, the two methods are based on similar structures. With the exception that the TF-2CNN method has four fully connected layers (multitask) output while there is only one for the WS-CNN since it uses pooling layers. For LSTM, learning has a computation time similar to CNN but is less stable because of the recurrent aspect of the method. Table 6.5 shows a computation time for WS-LSTM and WS-CNN of 50 and 26 times longer than TF-2CNN, respectively. The TF-2CNN is faster than other methods

for two significant reasons: the image for each ear is smaller (18 × 320), and the whole image is presented to the CNN input at each 0.2 s. Otherwise, the WS-CNN used an image of size 54 × 400 splits into 16 images of size 54 × 50 with 50% overlapping. WS-LSTM uses an image size of 25 × 400 at each ear, but the recursively of the LSTM consists of presenting a vector of 25 × 1 for each ear at each time sample of 0.5 ms for 0.2 s time sequence.

Table 6.5 Computation time for an azimuth estimation with a uniform sound source using 13 × 13 (169) angles of 0.2 s time sequence for the training-validation phase and 25 × 25 (625) angles at 35 dB SNR for the test

Method	**Training data generation** 13×13 angles **[s]**	**Test data generation** 25×25 angles **[s]**	**Training** 13×13 angles (50 epochs) **[s]**	**Testing** 25×25 angles **[s]**	**Regression estimation** one angle **[ms]**
TF-2CNN	0.56	1.04	17	0.80	2.9
WS-LSTM	9.24	17.30	842	0.39	28.3
WS-CNN	9.34	20.07	446	18.90	62.4

One of the essential computation times to observe is the time of testing to compute the output estimation for all 625 angles. In Table 6.5, the faster method is WS-LSTM, 2 and 23 times faster than TF-2CNN and WS-CNN, respectively. Since the computation time to estimate only one angle from 0.2 s time sequence for each ear consists of adding the computation time to generate the feature extraction and the time for testing. Table 6.5 shows that TF-2CNN, WS-LSTM and WS-CNN used, on average, 3 ms, 28 ms and 62 ms, respectively, for regression estimation on one angle. Those three average processing times can also be considered as an estimation of the time needed to estimate the position of a sound source in an actual word implementation. The TF-2CNN is faster than the other, but

the speedup comes from the FFT compiled function compared to the three MATLAB functions used to execute the WS feature extraction. The CNN and LSTM functions from MATLAB are executed based on code optimized to run on GPU.

Chapter 7: Conclusion

Many audio processing systems at work and in our daily lives rely on sound source localization, such as speech enhancement or recognition and human-robot interaction, to name a few. Binaural sound source localization determines the position of a sound source according to head-related transfer functions in both azimuth and elevation coordinates. This master's thesis proposes a machine learning base method with feature extraction for binaural sound source localization around a manikin head, i.e., by using two microphones located in the auditory canal of each ear location around a manikin head. NARX, long short-term memory and convolutional neural network were selected for classification and regression NN model with additional feature extraction technique based on wavelet scattering decomposition to extract SSL cues from HRTF generated binaural signals. The results show the LSTM neural networks and a feature extraction algorithm based on scattergram decomposition in three scenarios. In all cases, the LSTM offers better performance than NARX and using feature extraction further improves performance accuracy.

This thesis proposed a novel binaural SSL method with wavelet scattering feature extraction applied to the right and left ears to locate the single loudspeaker azimuth and elevation with good accuracy using two LSTM (WS-LSTM) and two CNN (WS-CNN) machine learning approaches. A scattering decomposition was used to extract the space-time information parameters sensitive to SSL. Experimental results based on the IRCAM and CIPIC HRTF database proposed wavelet scattering features demonstrated as

regression to estimate azimuth and elevation angles. HRIR dataset is used at different loudspeaker locations to generate data to locate the single loudspeaker coordinates precisely. The results show a good quality of source localization using a uniform sound source and added Gaussian noise and babble noise at multiple SNRs. Most works in the literature propose classification methods and no regression SSL estimation, which is more suitable for a continuous problem like in real-world applications. In addition, the proposed method instead aims at an estimation by regression, presenting a more complex resolution. The results demonstrate that the proposed regression method performs well in several noisy environments compared to current TF-2CNN methods in the literature, considering reduced sequence data. Finally, the results of the proposal in regression mode show its ability to estimate the azimuth and elevation positions of the source using only 27% of the locations during the training phase compared to classification methods. Compared with TF-2CNN, the WS-CNN outperforms regression scenarios for many kinds of noises and SNR. The project aims, among other things, to allow an improvement of cochlear implant devices and the development of robotics close to human physiognomy. The project's benefits will improve speech recognition, human-robot interaction, and immersion in virtual reality environments.

7.1 Contribution

As part of this research work, a paper was accepted in April 2022 at the international conference NEWCAS, a flagship conference of IEEE:

[64] P. Massicotte, H. Chaoui and M. Ahmed Ouameur, "LSTM with Scattering Decomposition-Based Feature Extraction for Binaural Sound Source Localization," 20th IEEE International NEWCAS Conference (NEWCAS), Quebec, 2022, pp. 436-440.

This article presented the LSTM method with wavelet scattering as feature extraction. Preliminary results are shown in classification and regression. Furthermore, a comparison is made with the paper [24] (TF-2CNN). Note that the TF-CNN method in [24] only presents results with a single CNN for azimuth and elevation output and in classification mode. Chapter 4 and Chapter 5 of the thesis present improved results in classification and regression, respectively, where the TF-2CNN method has been restimulated under the same conditions to ensure a more reliable comparison in classification and regression.

7.2 Future Works

The proposed methods based on wavelet scattering as feature extraction applied to LSTM and CNN for SSL regression and classification open the door for future works:

- Test performances in unmatched conditions where multiple additive noises at various SNR levels are used to train the NN. The performance evaluation is done on a different type of additive noise.
- In this thesis, only the KEMAR manikin HRTFs were used to evaluate the performance of the methods. Since every HRTF are different for each subject, it would be interesting to evaluate the performance using human subject HRTFs recordings.

- This project considers only the direct path from one loudspeaker and two microphones. More practical things would be to evaluate the methods with reverberation inside real-world local, considering the walls, ceiling, and floor effect.
- Evaluate and improve the accuracy using many microphones. However, if an array of microphones is used, it will be a general SSL challenge, not binaural SSL.
- In this study, the distance of the sound source is assumed to be constant. A more challenging context can be studied considering the variable distance depth of the sound source.
- Among other potential feature extraction methods to extract localization cues for binaural SSL is the diffusion decomposition from the Gammatone Cepstral Coefficient (GTCC) [65] and the Mel Frequency Cepstral Coefficient (MFCC) [66].

Bibliography

[1] D. S. Talagala, W. Zhang, T. D. Abhayapala and A. Kamineni, "Binaural sound source localization using the frequency diversity of the head-related transfer function," *The Journal of the Acoustical Society of America,* vol. 135, pp. 1207-1217, 2014.

[2] G. Andéol, E. A. Macpherson and A. T. Sabin, "Sound localization in noise and sensitivity to spectral shape," *Hearing Research,* vol. 304, pp. 20-27, 2013.

[3] M. Risaud, J. N. Hanson, F. Gauvrit, C. Renard, P. E. Lemesre, N. X. Bonne and C. Vincent, "Sound source localization," *European Annals of Otorhinolaryngology, Head and Neck Diseases,* Vols. 135, no. 4, pp. 259-264, August 2018.

[4] H. Wallach, "The role of head movements and vestibular and visual cues in sound localization," *Journal of Experimental Psychology,* vol. 27, p. 339–368, 1970.

[5] L. Jindong, H. Erwin and S. Wermter, "Mobile robot broadband sound localisation using a biologically inspired spiking neural network," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2191-2196, 2008.

[6] H. Ziegelwanger, M. Piotr and K. Wolfgang, "Numerical calculation of listener-specific head-related transfer functions and sound localization: Microphone model and mesh discretization," *The Journal of the Acoustical Society of America,* vol. 138, pp. 208-222, 2015.

[7] R. So, N. Leung, J. Braasch and K. Leung, "A low cost, non-individualized surround sound system based upon head related transfer functions: An ergonomics study and prototype development," *Applied ergonomics,* pp. 695-707, 2006.

[8] J. Murray, H. Erwin and S. Wermter, "Robotics sound-source localization and tracking using interaural time difference and cross-correlation," *Proceedings of NeuroBotics Workshop,* pp. 89-97, 2004.

[9] J. Valin, F. Michaud, F. Rouat and D. Létourneau, "Robust sound source localization using a microphone array on a mobile robot," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 1228-1233, 2003.

[10] I. Örnolfsson, T. Dau, N. Ma and T. May, "Exploiting non-negative matrix factorization for binaural sound localization in the presence of directional interference," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 221-225, 2021.

[11] J. H. Kim, J. Choi, J. Son, G. S. Kim, J. Park and J. H. Chang, "MIMO noise suppression preserving spatial cues for sound source localization in mobile robot," *IEEE International Symposium on Circuits and Systems (ISCAS),* pp. 1-5, 2021.

[12] B. T. Taddese, Sound source localization and separation, Macalester College, 2006.

[13] H. Wang and M. Kaveh, "Coherent signal-subspace processing for the detection and estimation of angles of arrival of multiple wide-band sources," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 33, pp. 823-831, 1985.

[14] D. Pavlidi, M. Puigt, A. Griffin and A. Mouchtaris, "Real-time multiple sound source localization using a circular microphone array based on single-source confidence measures," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 2625-2628, 2012.

[15] J. Woodruff and D. Wang, "Binaural localization of multiple sources in reverberant and noisy environments," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, pp. 1503-1512, 2012.

[16] J. Ahveninen, N. Kopčo and . I. P. Jääskeläinen, "Psychophysics and neuronal bases of sound localization in humans," *Hearing research,* pp. 86-97, 2014.

[17] H. M. Abboodi, Binaural sound source localization using machine learning with spiking neural networks features extraction, Salford: University of Salford, 2019.

[18] D. Goodman and R. Brette, "Learning to localise sounds with spiking neural networks," *23rd International Conference on Neural Information Processing Systems,* p. 784–792, 2010.

[19] S. M. Kim and H. K. Kim, "Direction-of-arrival based SNR estimation for dual-microphone speech enhancement," *IEEE/ACM Trans. Audio, Speech, Language Process.,* Vols. 22, no. 12, pp. 2207-2217, December 2014.

[20] T. Ogata, K. Nakadai and N. Yalta, "Sound source localization using deep learning models," *Journal of Robotics and Mechatronics,* vol. 29, no. 1, pp. 37-48, 2017.

[21] T. Rodemann, G. Ince, F. Joublin and C. Goerick, "Using binaural and spectral cues for azimuth and elevation localization," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2185-2190, 2008.

[22] H. Chen and W. Ser, "Acoustic source localization using LS-SVMs without calibration of microphone arrays," *IEEE International Symposium on Circuits and Systems (ISCAS),* pp. 1863-1866, 2009.

[23] Y. Sun, "Indoor sound source localization with probabilistic neural network," *IEEE Transactions on Industrial Electronics,* pp. 6403-6413, 2018.

[24] C. Pang, X. Li and H. Liu, "Multitask learning of time-frequency CNN for sound source localization," *IEEE Access,* vol. 7, pp. 40725-40737, 2019.

[25] T. May, S. Van De Par and A. Kohlrausch, "A probabilistic model for robust localization based on a binaural auditory front-end," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, pp. 1-13, 2011.

[26] N. Roman and D. Wang, "Binaural tracking of multiple moving sources," *IEEE Transactions on Audio, Speech and Language Processing,* Vols. 16, no. 4, p. 728–739, 2008.

[27] Y. Haneda, S. Makino, Y. Kaneda and N. Kitawaki, "Common-acoustical-pole and zero modeling of head-related transfer functions," *IEEE Transactions on Speech and Audio Processing,* Vols. 7, no. 2, p. 188–195, 1999.

[28] X. Zhong and B. Xie, "Head-Related transfer functions and virtual auditory display," *Soundscape Semiotics - Localization and Categorization,* vol. 29 no.1, pp. 37-48, 2014.

[29] B. Kapralos, M. Jenkin and E. Milios, "Virtual audio systems," *Presence Teleoperators & Virtual Environments,* pp. 527-549, December 2008.

[30] A. D. Miller, Modeling HRTF for sound localization in normal listeners and bilateral cochlear implant users, Denver: University of Denver, 2013.

[31] P. Pertilä and J. Nikunen, "Microphone array post-filtering using supervised machine learning for speech enhancement," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH,* pp. 2675-2679, 2014.

[32] S. Araki, H. Sawada and S. Makino, "Blind speech separation in a meeting situation with maximum SNR beamformers," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. I-41-I-44, 2007.

[33] X. Wu, D. Talagala, W. Zhang and T. Abhayapala, "Binaural localization of speech sources in 3-D using a composite feature vector of the HRTF," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 2654-2658, 2015.

[34] J. Zhang and H. Liu, "Robust acoustic localization via time-delay compensation and interaural matching filter," *IEEE Transactions on Signal Processing,* vol. 63, pp. 4771-4783, 19 June 2015.

[35] D. Yao, J. Zhao, L. Cheng, J. Li, X. Li, X. Guo and Y. Yan, "An individualization approach for head-related transfer function in arbitrary directions based on deep learning," *The Journal of the Acoustical Society of America (JASA) Express Letters,* vol. 2, p. 064401, 2022.

[36] K. Iida, "Comparison of HRTF databases," in *Head-Related Transfer Function and Acoustic Virtual Reality*, Singapore, Springer Singapore, 2019, pp. 171-177.

[37] V. R. Algazi, R. O. Duda, D. M. Thompson and C. Avendano, "The CIPIC HRTF database," *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics,* pp. 99-102, 2001.

[38] "Listen HRTF database," Room Acoustics Team, IRCAM, [Online]. Available: http://recherche.ircam.fr/equipes/salles/listen/.

[39] W. G. Gardner and K. D. Martin, "HRTF measurements of a KEMAR," *The Journal of the Acoustical Society of America,* vol. 97, pp. 3907-3908, 1995.

[40] J. Wang, J. Wang, K. Qian, X. Xie and J. Kuang, "Binaural sound localization based on deep neural network and affinity propagation clustering in mismatched HRTF condition," *Journal on Audio, Speech, and Music Processing,* vol. 2020 no. 4, pp. 1-16, 2020.

[41] B. Yang, H. Liu and X. Li, "Learning deep direct-path relative transfer function for binaural sound source localization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 29, pp. 3491-3503, 2021.

[42] Z. Pan, M. Zhang, J. Wu, J. Wang and H. Li, "Multi-Tone phase coding of interaural time difference for sound source localization with spiking neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 29, pp. 2656-2670, 2021.

[43] T. Oess, M. Löhr, C. Jarvers, D. Schmid and H. Neumann, "A bio-inspired model of sound source localization on neuromorphic hardware," *2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS),* pp. 103-107, 2020.

[44] T. Song, T. Qu, X. Wu and J. Chen, "An artificial neural network model for predicting sound direction in different acoustic environments," *22nd International Congress on Acoustics,* pp. 1-10, 2016.

[45] J. Wall, "Post-Cochlear auditory modelling for sound localisation using bio-inspired techniques," Ph.D. Thesis University of Ulster Faculty of Computing and Engineering, 2010.

[46] D. Qin, J. Tang and Z. Yan, "Underwater acoustic source localization using LSTM neural network," *39th Chinese Control Conference (CCC),* 2020.

[47] T.-H. Tan, Y.-T. Lin, Y.-L. Chang and M. Alkhaleefah, "Sound source localization using a convolutional neural network and regression model," *Sensors,* vol. 21, pp. 1-17, 2021.

[48] Y. Xu, S. Afshar, R. K. Singh, R. Wang, A. v. Schaik and T. J. Hamilton, "A binaural sound localization system using deep convolutional neural networks," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1-5.

[49] Y. Xu, S. Afshar, R. K. Singh, T. J. Hamilton, R. Wang and A. v. Schaik, "A machine hearing system for binaural sound localization based on instantaneous correlation," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1-5.

[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* Vols. 9, no. 8, p. 1735–1780, 1997.

[51] A. Graves, Supervised sequence labelling with recurrent neural networks, Munich: Technical University of Munich, 2008.

[52] MathWorks, "Long short-term memory networks," MathWorks, [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html.

[53] GRAS Sound & Vibration, "KEMAR," GRAS Sound & Vibration, [Online]. Available: http://kemar.us/.

[54] J. Andén, V. Lostanlen and S. Mallat, "Joint time-frequency scattering for audio classification," *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP),* pp. 1-6, 2015.

[55] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing,* vol. 62, pp. 4114-4128, 2014.

[56] Mathworks, "Wavelet Scattering," Mathworks, [Online]. Available: https://www.mathworks.com/help/wavelet/ug/wavelet-scattering.html.

[57] Speech Research Unit (SRU) at Institute for Perception-TNO, "Signal Processing Information Base (SPIB)," Speech Research Unit (SRU) at Institute for Perception-TNO, February 1990. [Online]. Available: http://spib.linse.ufsc.br/noise.html. [Accessed August 2022].

[58] MathWorks, "MATLAB and audio toolbox," Natick, Massachusetts, United States.

[59] F. Chollet, Deep learning with python, Shelter Island: Manning Publications Co., 2017, p. 361.

[60] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," *32nd International Conference on Machine Learning,* pp. 1-9, 2015.

[61] FFTW, "FFTW," FFTW, [Online]. Available: https://www.fftw.org/. [Accessed 15 08 2022].

[62] M. Frigo and S. Johnson, "FFTW: an adaptive software architecture for the FFT," *IEEE International Conference on Acoustics, Speech and Signal Processing,* pp. 1381-1384, 1998.

[63] M. Frigo and S. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE,* vol. 93, pp. 216-231, 2005.

[64] P. Massicotte, H. Chaoui and M. Ahmed Ouameur, "LSTM with scattering decomposition-based feature extraction for binaural sound source localization," *20th IEEE International NEWCAS Conference (NEWCAS),* pp. 436-440, 2022.

[65] M. Slaney, "An efficient implementation of the Patterson-Holdsworth auditory filter bank," Apple Computer Technical Report 35, 1993.

[66] A. Das, L. Gopalakrishnan Pillai and M. C, "Human voice localization in noisy environment by SRP-PHAT and MFCC," *International Research Journal of Advanced Engineering and Science,* vol. 1, no. 3, pp. 33-37, 2016.