IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version november 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Improving Speaker Recognition in Environmental Noise with Adaptive Filter

VINÍCIUS ALMEIDA DOS SANTOS [1,2], (Member, IEEE), WEMERSON DELCIO PARREIRA [2], (Member, IEEE), ANITA MARIA DA ROCHA FERNANDES [1], RAÚL GARCÍA OVEJERO[5], AND VALDERI REIS QUIETINHO LEITHARDT [3,4], (Senior Member, IEEE)

[1]Laboratory of Artificial Intelligence (LIA), University of Vale do Itajaí (UNIVALI), Florianópolis, SC 88032-005, Brazil
[2]Laboratory of Embedded and Distributed Systems (LEDS), University of Vale do Itajaí (UNIVALI), Itajaí, SC 88302-901, Brazil
[3]VALORIZA, Research Center for Endogenous Resources Valorization, Instituto Politécnico de Portalegre, 7300-555 Portalegre, Portugal
[4]COPELABS, Universidade Lusófona de Humanidades e Tecnologias, 1749-024 Lisbon, Portugal
[5]Expert Systems and Applications Laboratory, E.T.S.I.I. de Béjar, Universidad de Salamanca, 37700 Salamanca, Spain

Corresponding authors: Vinícius A. Santos (viniciusas@edu.univali.br) and Wemerson D. Parreira (parreira@univali.br).

This work was supported in part by the Spanish Agencia Estatal de Investigación, and in part by the Project Monitoring and Tracking Systems for the Improvement of Intelligent Mobility and Behavior Analysis (SiMoMIAC) under Grant PID2019-108883RB-C21/AEI/10.13039/501100011033

ABSTRACT Speaker recognition is challenging in real-world environments. Typically, studies approach noises only in an additive manner. However, real environments commonly present reverberating conditions that worsen speech processing. When not considering reverberation in the system modeling, the system may not be robust when applied to real-world conditions. In this work, we use a slight different approach to simulate reverberation, considering randomized conditions of the environment. With this approach, each VoxCeleb1 test sample is corrupted by randomly generated conditions, with diversified amplitudes of noise and speech. We generate a corrupted dataset, in which the best model EER degraded from 0.93% to 30.13%. To improve this degradation, we propose using Normalized Kernel Least-Mean-Square (NKLMS) adaptive filter. Through the use of NKLMS, we were able to improve the EER from 30.13% to 1.11%. The results indicate that NKLMS has a great potential for speech enhancement to improve speaker recognition.

INDEX TERMS Adaptive Filter, Kernel Methods, Noise Pollution, Speaker Recognition

I. INTRODUCTION

In real environments, noise and reverberation are always present, negatively affecting speech processing. It is typical for real environments to be noisy [1]. In addition to colored noises, real environments are subject to several other conditions, such as reverberation, babble noise, and speaker overlapping. Audio capture can occur anywhere, such as on the street (subject to noises from construction works, cars, wind, etc.) and in offices (subject to noises from air conditioning, electronic devices, people, etc.). Each noise has its characteristics and can impact speaker recognition differently.

Speaker recognition also is impacted by noise and reverberation. In speech-based systems, babble noise is one of the most challenging noises to filter [2]. Even speakers counting in a signal is already a considerable challenge [3], [4]. Babble noise is subject to several factors, such as emotional speaker condition [5], [6], environment acoustics (reverberation) [4], [7], and microphone position. Robustness is essential for speaker recognition, especially when used for security purposes.

In real environments, speaker recognition methods must be reliable. However, most works do not validate their methods in conditions similar to real-world – see [8] for an overview – using artificial degradation, which worsens speaker recognition reliability. Usually, works that evaluate degrading conditions focus on changes in the methods, experimenting with different feature extractors, classifiers, or scoring. To improve the impact of real-world conditions, the usage of speech enhancement might bring more robustness to systems [9].

A speech enhancement approach to attenuate signal degradation is filtering the noises and reverberation [10]. Classical filtering methods use constant parameters defined by the designer, which outline the filter behavior. However, these methods lack adaptability to unknown conditions. Several works use spectral subtraction for speech enhancement [11]–[13]. In addition, adaptive filters are efficient in environments subject to constant changes [14].

The adaptive algorithms literature is extensive, and an active research field [14], being also applied in different scenarios and applications as described in [15] and [16]. Some papers indicate that Kernel methods are improving adaptive filters [17], [18]. Therefore, we evaluated the NKLMS (Normalized Kernel Least Mean Squares) [19] in this work. Besides kernel's usage, it also combines a normalization introduced by NLMS (Normalized Least-Mean-Square).

In this work, we propose improving speaker recognition in simulated environments through speech enhancement. Thus, we: (i) generated a corrupted dataset with randomly noisy and reverberant conditions; (ii) evaluated the dataset for speaker recognition before and after the simulation; (iii) experimented with adaptive filters for speech enhancement; and (iv) evaluated the result of speaker recognition after the speech enhancement.

In Figure 1 we depict this work's steps and methods. Through this process, we were able to discover the following contributions:

1) A slight different approach to simulate reverberation, considering noisy environmental conditions;
2) Evaluating speaker recognition accuracy for the corrupted conditions;
3) Evaluating several adaptive filters for the corrupted conditions;
4) Obtaining promising results with NKLMS [19] for speech enhancement; and
5) Improving speaker recognition accuracy with NKLMS filter in noisy conditions.

This work's **main contribution** is using NKLMS [19] in pre-processing stage of speaker recognition. The speech signal presents nonlinear behavior [20], and the KLMS method is excellent for solving nonlinear problems [21]. NLMS, on the other hand, increases convergence stability, facilitating parametrization and improving robustness (compared to traditional LMS) [19]. Therefore, by joining NLMS and KLMS methods, NKLMS has excellent potential for improving speech processing, despite not being extensively used in the literature[1].

In the following sections, we will: introduce real environment conditions and simulation (Section II); introduce adaptive filters (Section III); describe our experimental setup (Section V); present and discuss our results (Section VI); and lastly, summarize the paper (Section VII).

II. REAL ENVIRONMENTS AND SIMULATION

Closed environments are subject to reverberation. Reverberation consists of the reflection of sound waves in obstacles, walls, and others [22]. As exemplified in Figure 2, sound sources are reflected, creating countless sound paths. The reverberation can make sounds more pleasant or incomprehensible, depending on the room's acoustics. One way to simulate reverberation is with RIR (Room Impulse Response).

[1]By the time of writing, there are 13 citations indexed on Google Scholar.

When simulating noises, most works evaluate noise impact in a additive manner [6], [23]–[25]. However, noises are also subject to environmental conditions. Usually, works that approach real situations use datasets that are already recorded in real environment conditions [6], [23]–[25], as SITW [26] and NIST 2010 retransmitted [27]. Works evaluating reverberation conditions usually only reverberate the speech, using the noise in an additive manner, applying it only for data augmentation, without speech enhancement [28], [29].

However, more diversified conditions can be recreated by simulating real conditions. From a more diverse dataset, more data can be generated and used to improve learning or for training methods specific for filtering noises and reverberation.

Several works already work with Room Impulse Response (RIR) problematic, which simulates room reverberation, providing libraries available for use [30]. However, most available generators do not allow diverse room shapes and are mainly for Matlab [30].

To provide an accessible RIR generator with an open and flexible code, Scheibler et al. [30] developed a Python library called `pyroomacoustics`. The library includes an RIR generator based on ISM (Image-Source Mirror), which visually maps the position of sound sources and microphones for simulation. The programming with `pyroomacoustics` library is high level, allowing a clean and intuitive code. In this work, we use this library to simulate different room conditions, and evaluate speaker recognition quality.

III. REVIEW ON ADAPTIVE FILTERS

Adaptive filters are great for environments with constant conditions variability [14]. Historically, they are helpful for several applications, such as communications, active noise control, and biomedical engineering. The adaptive filters have great potential for improving many systems.

A. LEAST-MEAN-SQUARE – LMS

The adaptive algorithms had their beginning with LMS [31]. LMS works based on a weight vector $\mathbf{w}(n)$, given:

$$\mathbf{w}(n) = [w_0(n), w_1(n), w_2(n), \cdots, w_{M-1}(n)]^\top. \quad (1)$$

Which have a vector position for each M sample previously processed. When processing new samples, each new weight is calculated as follows:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_w J(n). \quad (2)$$

In which μ is the adaptation/learning step, and the gradient vector is based on a stochastic gradient, defined by:

$$\nabla_{\mathbf{w}} J(n) = -\mathbf{u}(n)\, e^*(n), \quad (3)$$

where $\mathbf{u}(n)$ is the input signal and $e(n)$ is the deviation of the desired signal $d(n)$,

$$e(n) = d(n) - \mathbf{w}^H(n)\, \mathbf{u}(n). \quad (4)$$

Given the matrix $\mathbf{R}$ defined positive on input signal $\mathbf{u}(n)$ and $\mathbf{p}$ the cross-correlation vector between $\mathbf{u}(n)$ and $d(n)$

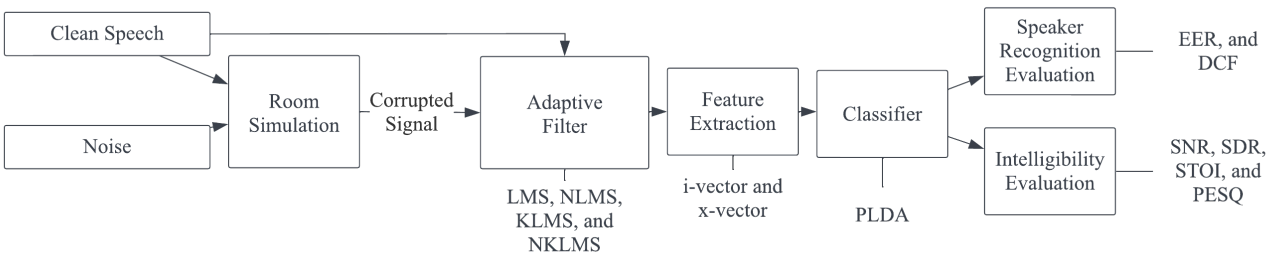

FIGURE 1. Full process used in this work.

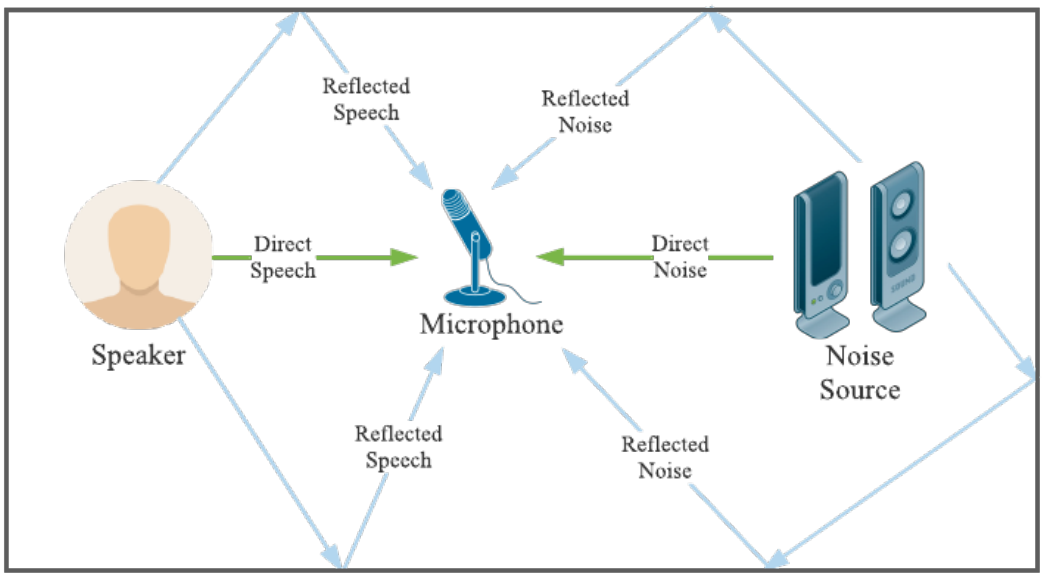

FIGURE 2. Sound propagation in a room.

the solution, $\mathbf{w}_o$, which minimizes the Mean Squared Error (MSE), usually known as the Wiener solution, given by:

$$\mathbf{w}_o = \mathbf{R}^{-1}(n)\,\mathbf{p}(n). \tag{5}$$

The input signal is processed at each iteration, and the algorithm adapts, adding a new weight for the weights vector. Later approaches introduce the dictionary concept, where the number of weights is no longer infinite but with a fixed number of weights. An issue with LMS is finding a step size adequate for the problem at hand [14]. Newer versions of LMS may be used to improve the learning stability.

B. NORMALIZED LMS – NLMS

The algorithm NLMS introduces an improvement to LMS stability [14]. By normalizing the step size, the weight update equation changes from (2) to:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{\|\mathbf{u}(n)\|^2}\,\nabla_w J(n), \tag{6}$$

Where μ is based on the amplitude of the input signal. NLMS adaptive filter is an extension of the popular LMS adaptive filter [32].

C. KERNEL LMS – KLMS

A challenge when dealing with voice processing is its nonlinear characteristic [33]. Therefore, it is necessary to explore filters with nonlinear behavior. Kernel-based methods are becoming more popular, mainly as complements of existing methods [34]. Consequently, it was also introduced as a complement to LMS with kernel.

The KLMS was recognized as a solution to deal with nonlinear adaptive filtering [21]. The main change from LMS to KLMS is a mapping of the input signal $\mathcal{U}$ to a Hilbert space $\mathcal{H}$ through a kernel function.

In Figure 3 the weight-update equation is

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\, e(n)\, \boldsymbol{\kappa}_{\boldsymbol{\gamma}}(n) \tag{7}$$

where $\boldsymbol{\kappa}_{\boldsymbol{\gamma}}(n) = [\kappa(\mathbf{u}(n), \mathbf{u}(\boldsymbol{\gamma}_1)), \kappa(\mathbf{u}(n), \mathbf{u}(\boldsymbol{\gamma}_2)), \ldots,$

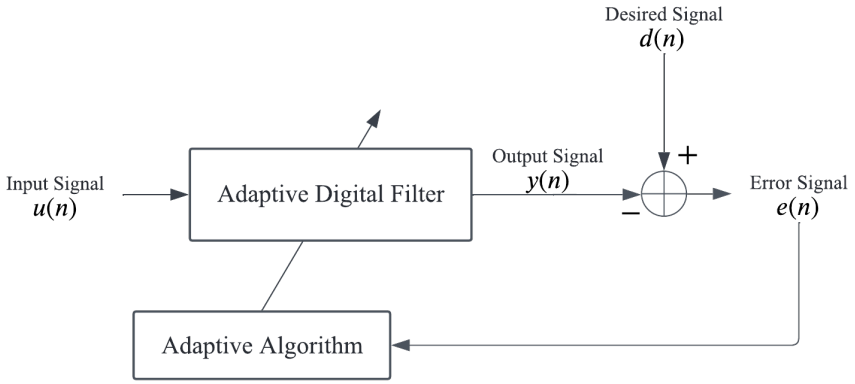

FIGURE 3. Adaptive Filter algorithm.

$\kappa(\mathbf{u}(n), \mathbf{u}(\boldsymbol{\gamma}_M))]^\top$ is a vector of kernels at time $n > M$ and $\kappa(., \mathbf{u}(\gamma_m))$ is the mth function of the dictionary.

With the KLMS algorithm, it is necessary to select a kernel method to use [34]. In this work, we used the Gaussian kernel, given by:

$$\kappa_{\text{Gauss}}(\mathbf{x}, \mathbf{x}') = \exp(-\nu \|\mathbf{x} - \mathbf{x}'\|^2), \tag{8}$$

where $\nu \in \mathbb{R}_{>0}$ and $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^p \subseteq \mathcal{U}$.

Several kernel methods are used in the literature [35], [36]. Other standard kernels are Laplacian and Polynomial, but new kernel methods can be designed depending on the application.

D. NORMALIZED KLMS – NKLMS

For further improvement on the KLMS algorithm, a joint approach of NLMS and KLMS can be used [19]. NKLMS improves the tracking and convergence speed of the adaptive algorithm. The normalization is the same as in NLMS, Eq. (6)), but happens after the Hilbert space mapping, leading to

$$\mathbf{w}(n+1) = \mathbf{w}(n) - 2\,\frac{\mu}{\|\boldsymbol{\kappa}_{\boldsymbol{\gamma}}(n)\|^2}\,e(n)\,\boldsymbol{\kappa}_{\boldsymbol{\gamma}}(n). \tag{9}$$

Comparing NLMS to NKLMS, there are some advantages [19]:i) easy implementation; ii) easier to choose working parameters; iii) faster convergence time; iv) little computational time added; and v) increased robustness because of normalization, which solves scaling issues.

The NKLMS algorithm [19] is not extensively explored in the literature, having few indexed citations.

IV. EVALUATED METHODS

Speaker recognition usually use a feature extraction and classification [13]. Sometimes, these are designated as front-end and back-end, respectively. For feature extraction, some common methods are MFCC, i-vectors, x-vectors, combinations of different features, and many more. For classification, there is also a high diversity of methods, such as Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), Dynamic Time Warping (DTW), Vector Quantization (VQ), Neural Networks (NN), Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), Probabilistic LDA (PLDA), and Euclidean Distance. For more details, we recommend reading Mohd Hanifa et. al. review [13].

In this work, we evaluated the dataset corruption with four methods: i-vectors [28], x-vectors [28], ECAPA-TDNN [37], and ResnetSE34 [38].

As evaluated by [39], the ECAPA-TDNN and ResnetSE34 methods are not the state of art for speaker verification, but achieve EERs close to 1%. In the following subsections we disclose about the methods evaluated in this paper.

A. I-VECTORS

Most speaker recognition systems used i-vectors [28]. I-vector is a method for extracting acoustic features from signal with GMM-UBM (Gaussian Mixture Model - Universal Background Model). For classification/scoring, a PLDA is used. The evaluated method in this work was implemented by [28].

B. X-VECTORS

X-vectors were proposed by Snyder et. al. [28], training to discriminate between speakers. The implementation is available as a Kaldi Toolkit [40] Recipe named `voxceleb`[2].

Uses filterbanks as input of a Deep Neural Network (DNN). The DNN is trained to classify the speakers, in which the output layer is the classification of speaker in one-hot encoding manner. The second last hidden layer is used as the speaker embeddings.

In this method, using data augmentation improves embedding learning, unlike i-vectors. For classification, PLDA is used in the same manner as i-vectors.

[2] Available at https://github.com/kaldi-asr/kaldi/tree/master/egs/voxceleb.

C. ResNetSE34

This implementation is based on the usage of Squeeze and Excitation (SE) block [38]. SE is a new architectural unit for neural networks, which explores the relationship between channels. It was proposed to improve Convolutional Neural Networks (CNN), and achieved state-of-art performance at several experiments.

We evaluated the ResNetSE34 implementation from [41][3], which was also commented at [39]. This implementation achieved an EER of 0,93% at VoxCeleb test split. Despite not presenting state-of-art results at VoxCeleb dataset, it still achieves competitive results.

D. ECAPA-TDNN

ECAPA-TDNN – Emphasized Channel Attention, Propagation, and Aggregation in Time Delay Neural Network – proposes multiple improvements to x-vector speaker embedding [37]. The work changed the neural network topology at several points, introducing the SE block [38].

In this work, we used the implementation made available by [41].

V. EXPERIMENTAL SETUP

In this work, we generated a dataset from VoxCeleb and MUSAN. After simulating diversified conditions with `pyroomacoustics` library, we evaluated the quality of speaker recognition. Later, we explored the best parameters of LMS, NLMS, KLMS, and NKLMS adaptive filters. We applied the filters to the generated dataset using the best parameters, reevaluating the speaker recognition quality.

Speech dataset: VoxCeleb [42] is a popular dataset [8] for speaker recognition. It was created from public YouTube videos, using audio and video information to identify the current speaker. Its corpus has 1251 speakers, with a diverse amount of people and conditions. In this work, we used the VoxCeleb1 test split to validate the speaker recognition quality. The test split includes 40 speakers and a total of 4,874 utterances, at a sample frequency of 16,000.

Noise dataset: MUSAN [43] is a dataset containing three types of sound: music, speech, and noises. It was created by crawling several sources, seeking variability in the sounds. In this work, we used these sounds randomly, considering all of them as noise. With the MUSAN dataset we could create diversified conditions for the generated dataset.

Generated dataset: Using both VoxCeleb1 and MUSAN datasets, we generated a new dataset through simulation. The simulations were made with `pyroomacoustics` library, with randomized settings. For each VoxCeleb1 sample, we:

i. Created a room with random dimensions:
   - width and depth: random integer from 5 to 15;
   - height: random integer from 2 to 5;

ii. Selected a random material for the room walls, ceiling, and floor (random reverberation coefficients);
iii. Positioned the speaker randomly;
iv. Selected a random noise from the MUSAN dataset;
vi. Normalized the noise to the same amplitude as the speaker; and
vii. Positioned the noise randomly.

This approach will give the generated dataset random noise, distortion, and intelligibility conditions.

Speech Enhancement: The generated dataset is then filtered by LMS, NLMS, KLMS, and NKLMS adaptive filter algorithms. There are innumerable ways to choose an adaptive filter input and learning target. In this work, we use the approach in Figure 4, where the filter learns to predict the clean signal.

Kernel Function: In this work, we use the Gaussian Kernel function, which is commonly used. The kernel function can be chosen based on prior knowledge about the signal behavior [44, chap. 13]; however, there is no definite way to select the best kernel. Other kernels will also show positive results for speech enhancement, and, possibly, better performances.

Parameters selection of the adaptive filters: Choosing the best kernel function and parameters is tricky [8]. In this work, we evaluated several configurations by filtering the same 50 samples from the generated dataset. In a brute-force manner, we evaluated each configuration with the metrics SNR, SDR, STOI, and PESQ. The best configurations were used in our speaker recognition experiments. In [8], this approach for choosing parameters is also used.

The evaluated parameters were:

i. For LMS and NLMS:
   (a) Step Sizes– μ: {0.1; 0.2; 0.3; 0.4}.
   (b) Filter Order: {16; 32; 64; 128; 256; 512; 1024}.

ii. For kernel methods:
   (a) Step sizes – μ: {0.001; 0.05; 0.01; 0.1; 0.2; 0.4; 0.6}.
   (b) Dictionary sizes (or filter order): {16; 32; 64; 128}.
   (c) Gaussian kernel parameter – ν: {1e-1; 1e-2; 1e-3; 1e-4; 0.2; 0.4; 0.6; 0.8; 1.4; 1.6; 1.8; 2.2; 3.0; 1; 5; 10; 20; 50; 100}

The kernel methods took a longer execution time. Therefore, we evaluated fewer dictionary sizes.

Recognition model: In all models, we used pretrained models made available online. For i-vectors and x-vectors, two versions of the Kaldi recipes were provided, named v1 and v2, respectively. The pretrained recipe models are available at https://kaldi-asr.org/models/m7 [28]. For ResNetSE34 and ECAPA-TDNN, the code and pretrained models are available at https://github.com/ranchlai/speaker-verification [41].

Parameters selection of the recognition models: The parameters of the evaluated models were previously configured and trained. The following number of features were extracted:

- i-vectors: 400;
- x-vectors: 512;

[3] Available at https://github.com/ranchlai/speaker-verification.

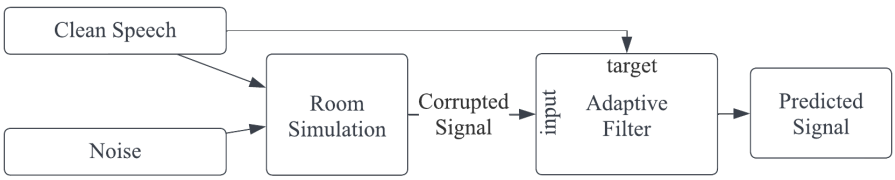

FIGURE 4. Filtering Approach.

- ResNetSE34: 256;
- ECAPA-TDNN: 192.

We recommend going to [28] for more details in i-vectors and x-vectors configuration. For ECAPA-TDNN, the specifications were detailed at [37]. The ResNetSE34 model implementation is available at [41].

A. EVALUATION METRICS

In this work, we evaluate the results with intelligibility and classification metrics. Intelligibility metrics compare the estimated signal to the expected result, giving an idea of the filter quality. We also evaluate the expected (original) signal with itself, which helps to understand the metric boundaries. The classification metrics evaluate the speaker recognition quality.

In the following, we summarize the used metrics.

1) Intelligibility

SNR – Signal to Noise Ratio: There are several metrics in the literature that evaluate speech quality and intelligibility. The default signal quality metric is the SNR [10]. The SNR is a metric adequate for quality when the system presents additive noise unrelated to the main signal. The SNR is defined by the rate of the input signal average amplitude $M(A_{in})$ by the absolute average difference of the original signal $M(A_o)$ and the reconstructed noise $M(A_r)$:

$$\text{SNR} = \frac{M(A_{in})}{|M(A_o) - M(A_r)|}. \tag{10}$$

Signal to Distortion Ratio – SDR: The SDR was created to evaluate source separation methods [45]. SDR seeks to give an overall quality measure of sound sources.

Short-Time Objective Intelligibility – STOI: The STOI metric computes the average correlations across all 1/3-octave bands and 384ms blocks [1]. The average correlation creates an intelligibility score. When subject to nonlinear processes, this metric may fail to evaluate intelligibility correctly.

Perceptual Evaluation of Speech Quality – PESQ: The PESQ metric is recommended by the ITU-T study group[4] as an objective measure of speech quality [1]. PESQ metric was created to evaluate codec and network conditions reliably.

2) Classification

The usual metric to evaluate speaker recognition quality is the **Equal Error Rate (EER)**, also called Correlation Error rate. EER is dependent on FRR and FAR metrics, given by [46]:

$$\text{FRR} = \frac{\text{TN}}{\text{TP} + \text{TN}} = \text{False Refection Rate} \tag{11}$$

and

$$\text{FAR} = \frac{\text{FP}}{\text{FP} + \text{FN}} = \text{False Alarm Rate} \tag{12}$$

where FP is false positives; FN is false negatives; TP is true positives and TN is true negatives.

The EER is defined by the decision threshold where FAR = FRR, seeking a balance in the system. The FAR and FRR metrics are inversely proportional, therefore: if FAR is high and FRR low, the system will be user "friendly", however, insecure; if FRR is high and FAR low, the system

[4]Recommendation P.862

TABLE 1. Intelligibility evaluation of the VoxCeleb1 test split.

Condition	SNR	SDR	STOI	PESQ
Original	92.11 ± 5.59	inf	1.00 ± 0.00	4.64 ± 0.00
Corrupted	-5.62 ± 1.94	-7.95 ± 6.69	0.33 ± 0.16	1.10 ± 0.13
LMS	-14.93 ± 15.05	-14.54 ± 15.46	0.47 ± 0.20	1.29 ± 0.41
NLMS	3.69 ± 1.61	3.58 ± 2.61	0.80 ± 0.08	1.60 ± 0.28
KLMS	8.49 ± 3.91	27.45 ± 9.54	0.99 ± 0.03	4.10 ± 0.64
NKLMS	$\mathbf{9.01} \pm 3.31$	$\mathbf{46.26} \pm 12.03$	$\mathbf{1.00} \pm 0.00$	$\mathbf{4.64} \pm 0.02$

Legend: SNR – Signal to Noise Ratio; SDR – Signal to Distortion Ratio; STOI – Short-Time Objective Intelligibility; PESQ – Perceptual Evaluation of Speech Quality.

will not be user "friendly", but secure. The system designer must decide the system balance.

Other metric common for speaker recognition is the **Detection Cost Function – DCF**, which is a weighted sum of EER and FAR [46]:

$$C_{Det}(\theta) = C_{Miss} \times P_{\text{Miss|Target}}(\theta) \times P_{\text{Target}} + C_{\text{FalseAlarm}} \times P_{\text{FalseAlarm|Nontarget}}(\theta) \times (1 - P_{\text{Target}}), \quad (13)$$

where θ is the decision threshold, C_{Miss} the false rejection cost, $C_{\text{FalseAlarm}}$ the false acceptance cost, P_{Target} the target speakers probability. There are several variations of the equation, mostly with NIST's different editions.

VI. RESULTS AND DISCUSSION

Selected parameters: Table 2 lists the selected parameters for the adaptive filters. These parameters are based on the parameters selection procedure described in Section V.

TABLE 2. Adaptive filter parameters.

Filter	Step Size	Filter Order	Kernel Parameter
LMS	0.20	512	-
NLMS	0.20	256	-
KLMS	0.10	16	$\nu = 1.4$
NKLMS	0.60	16	$\nu = 20$

The ν parameter only exists for Gaussian kernel algorithms. Both KLMS and NKLMS use Gaussian kernel.

High ν parameter: When evaluating the NKLMS algorithm, we observed that larger values for the ν parameter improved the intelligibility metrics. However, with values larger than 20, all metrics were stabilized.

Waveforms Figure: Figure 5 show the waveforms of a signal in several conditions. "Original" refers to the clean noise from the dataset; "Corrupted Signal" is signal corrupted by noise – as described in Section V, adding reverberation and noise – and the LMS, NLMS, KLMS, and NKLMS adaptive algorithms waveform after denoising.

Figure 6 show how the energy in different frequency bands changes over time for the waveforms of the same sample of Figure 5. Comparing (b) to (a), we can see the changes after the corrupted dataset. The next images show the spectrogram after filtering with LMS (c), NLMS (d), KLMS (e) and NKLMS (f).

Intelligibility evaluation: Table 1 presents the average intelligibility metrics from the experiments. We also show the intelligibility metrics for each condition listed. The first line shows the original dataset metrics – the baseline values. The second line shows the metrics after simulation – the corrupted dataset generated. The average SNR in the corrupted dataset is -5.54, with a standard deviation of 1.93 (most works stop evaluating at an SNR of 0). With NKLMS (bold), we achieved the best values in all intelligibility metrics.

Speaker Recognition evaluation: Tables 3 and 4 show the speaker recognition results for the experiments at the evaluated conditions. In both Tables, 3 and 4, the NKLMS method improved the evaluation quality significantly, getting close to the original EER and MinDCF metrics.

As observed in Table 3, from top to bottom models, the EER in the simulated dataset improves. This indicates that newer models are becoming more robust. However, after the application of filters, the EER almost reaches it's original performance.

ECAPA-TDNN vs ResNetSE34: The ECAPA-TDNN [37] introduced a new network topology, while ResNetSE34 [41] is based on ResNet, a common neural network architecture. Although ECAPA-TDNN have more complexity, ResNetSE34 still outperforms it.

LMS Results: As observed in Table 1, the LMS is, in average, worsening the corrupted signal SNR and SDR. After filtering, the EER is also worse than the corrupted dataset. This can explain why classification after filtered with LMS presents a worse EER in ECAPA-TDNN compared to x-vector.

NKLMS intelligibility: The NKLMS reached values close to the original set, in STOI and PESQ metrics. However, looking at NLMS STOI and PESQ, the metrics also performed reasonably well. Despite not being as close to the original signal, such as kernel-based filters still presented an improvement on EER. Therefore, a larger intelligibility evaluation does not mean that speaker recognition accuracy will be robust.

Objectively evaluating intelligibility: According to [28], there are several challenges in training models for speaker recognition. Ideally, some models might learn to the point of intrinsically ignoring any noise. However, objective speech enhancement algorithms may be beneficial when focusing on systems robustness. The dataset generation approach used in this work can also be applied for better data augmentation, which is benefical for training several models. Therefore, this is another contribution of this work, despite being untested.

NKLMS performance: In our experiments, NKLMS adaptive filter achieved the best results. All filters were learning from the same data, which includes filtering both noise and room reverberation. This indicates that the NKLMS filter, when applied to the real world, has the potential to improve speaker recognition robustness. The conditions created by the generated dataset were harsh, resulting in an SNR of -5.54 ± 1.93, when most works stop at an SNR of 0 [11], [12], [47], [48].

NKLMS cost-benefit: The computational time to execute NKLMS algorithm is highly dependent on the chosen kernel. In our experiments, with Gaussian kernel, the NKLMS computational time with the chosen filter order was not much larger than that of LMS, per example. With little complexity added, the algorithm was able to improve the EER up to 30 times when compared to LMS, which most of the experiments yielded worst results than the simulated.

Finite filter order: As commented in Section V, in this work, we approach the adaptive filters using a finite filter order (dictionary size). This parametrization allows to regulate the adaptive filters complexity, which, otherwise, would be

TABLE 3. EER evaluation of VoxCeleb1 test split for each model.

Model	Original	Simulated	LMS	NLMS	KLMS	NKLMS
i-vectors	5.34%	38.20%	39.60%	12.65%	6.38%	**5.67%**
x-vectors	3.11%	38.06%	35.45%	8.04%	3.67%	**3.30%**
ECAPA-TDNN	1.10%	36.33%	42.74%	4.70%	1.69%	**1.39%**
ResNetSE34	0.93%	30.13%	33.37%	3.72%	1.40%	**1.11%**

TABLE 4. MinDCF evaluation of VoxCeleb1 test split for each model.

Model	Original	Simulated	LMS	NLMS	KLMS	NKLMS
i-vectors	0.4970	0.9994	0.9999	0.8622	0.5562	**0.5306**
x-vectors	0.3278	0.9999	1.0000	0.6593	0.3448	**0.3433**
ECAPA-TDNN	0.0847	0.9998	0.9999	0.2887	0.1071	**0.0930**
ResNetSE34	0.0741	0.9999	0.9999	0.2537	0.0990	**0.0854**

increasing linearly at each iteration. We kept a low filter order for the kernel-based algorithms, otherwise, the computational time would greatly increase. However, even with lower filter orders, the kernel-based algorithms yielded better results. To obtain acceptable results with the linear filters, a large filter order is required.

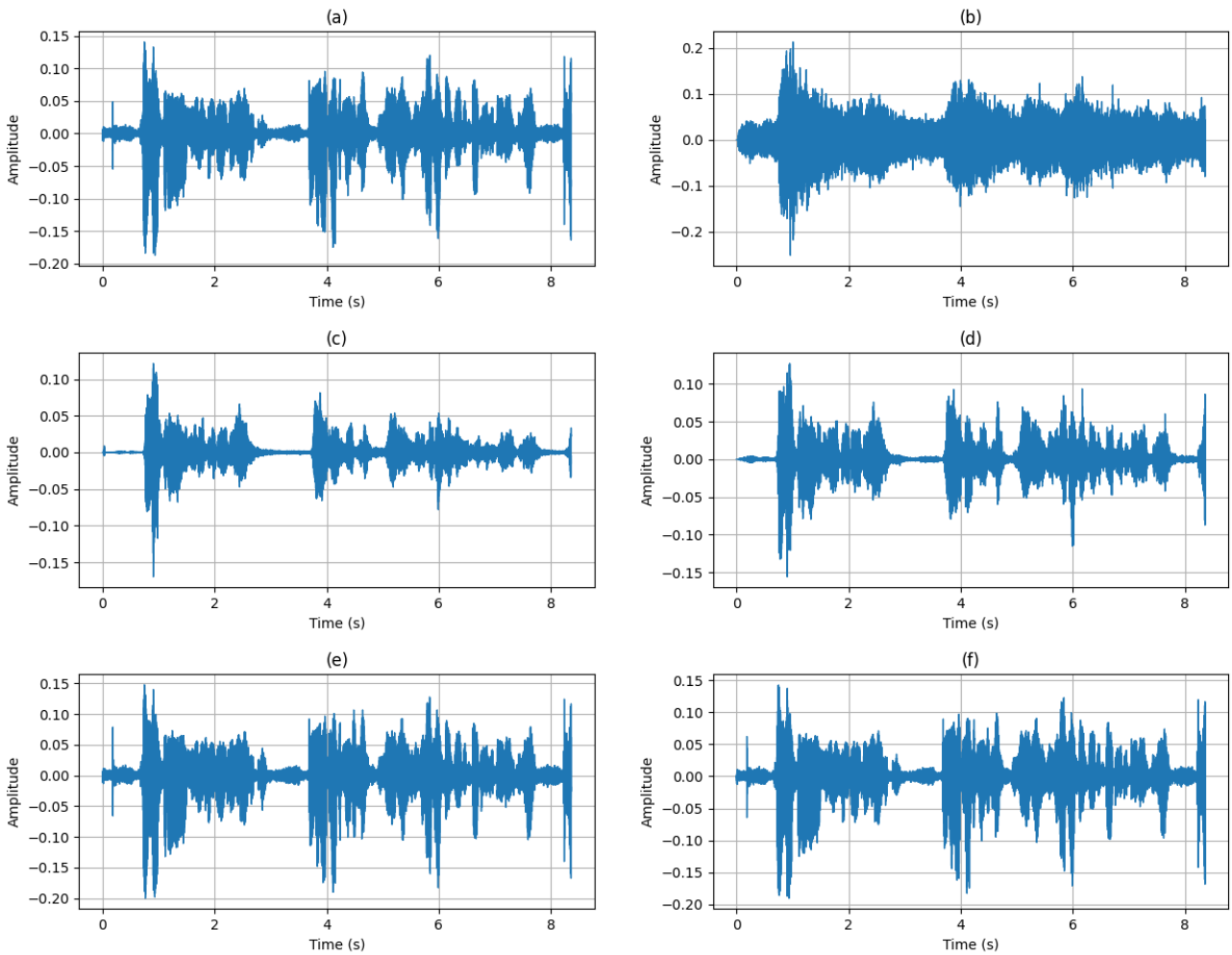

FIGURE 5. Signal Waveforms in the evaluated conditions: (a) Original signal; (b) Corrupted signal; (c) Filtered signal by LMS;(d) Filtered signal by NLMS (e) Filtered signal by KLMS; and (f) Filtered signal by NKLMS.

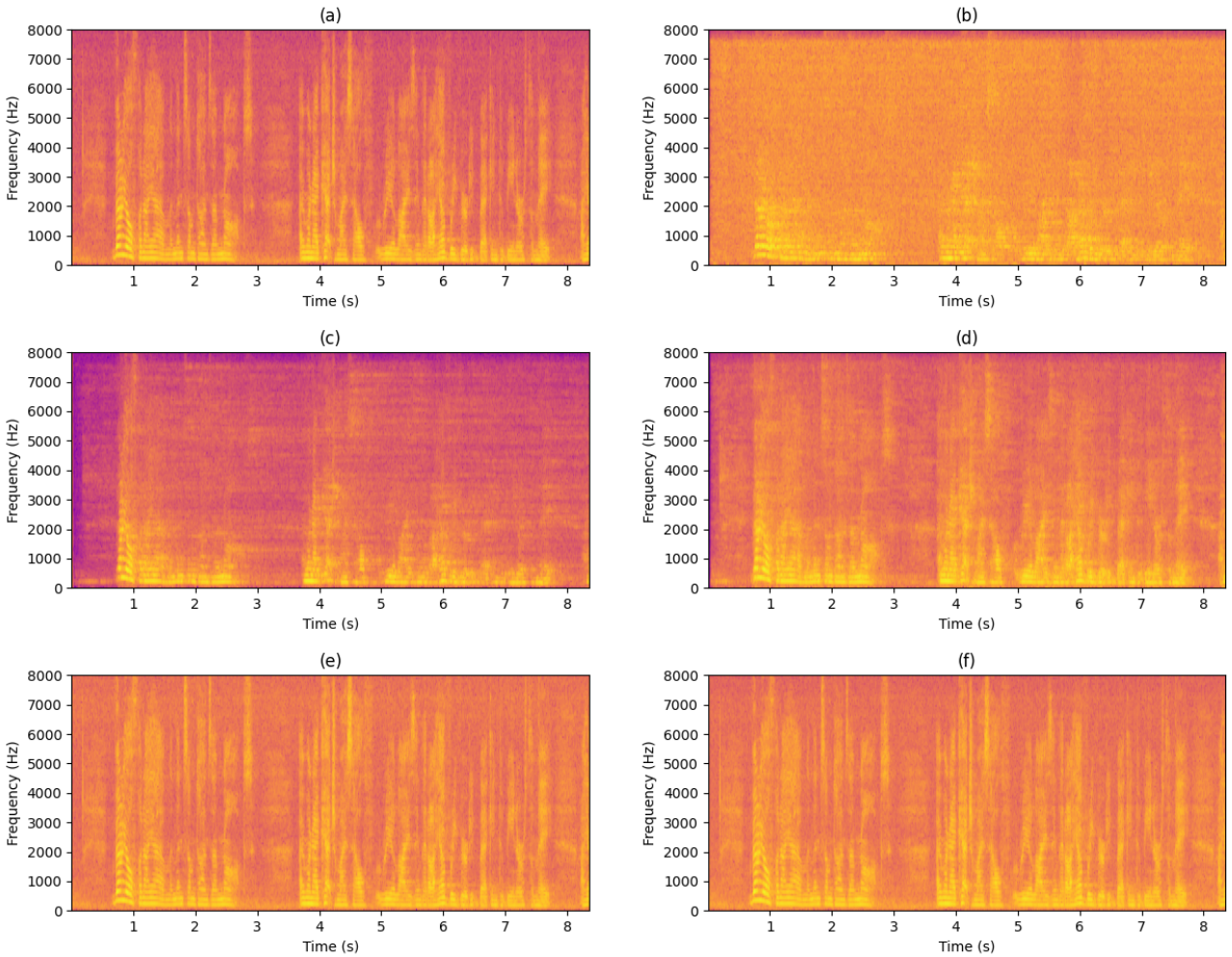

FIGURE 6. Signal Spectrograms in the evaluated conditions: (a) Original signal; (b) Corrupted signal; (c) Filtered signal by LMS;(d) Filtered signal by NLMS (e) Filtered signal by KLMS; and (f) Filtered signal by NKLMS.

Different tested orders in filters with and without kernel: We experimented with some filter orders kernel-based and non-kernel-based adaptive filters quite differently. In kernel-based algorithms (KLMS and NKLMS), when increasing the dictionary size, the algorithm presents a long runtime – note that the experiments were done in python, which negatively impacts the performance. For linear adaptive algorithms, LMS and NLMS, a larger filter order was preponderant for the adaptive learning.

VII. CONCLUSION

With the goal of improving speaker verification accuracy in noisy environments, in this work we generate a dataset by simulating room reverberation conditions with noise inside the room. We used VoxCeleb1 for speech samples and MUSAN noises. In our best results, the simulated dataset degraded the EER from 0.93% to 30.13%.

However, we applied adaptive filters, which attenuated this degradation. The best filter in our experiments is NLMS, a joint of NLMS and KLMS (based on [19]). The NKLMS improved the EER degradation from 30.13% to 1.11%, which is close to the baseline EER (0.93%).

Our results indicate improvements of speaker recognition in noisy environments through NKLMS adaptive filter. Future works will involve:

- training with a filtered dataset (after simulation);
- testing more diversified noisy conditions; and
- evaluating of other speech enhancement algorithms, such as multi-run Independent Component Analysis (ICA) [49] and ICA through Entropy Bound Minimization (ICA-EBM) [50].

ACKNOWLEDGMENTS

This research was funded in part by the State Foundation for Research Support of Santa Catarina FAPESC Nº 15/2021 – Science, Technology and Innovation Programa Support to Research Groups of Associação Catarinense das Fundações Educacionais - ACAFE, registered by the Term of Grant Nº 2021TR001236.

This work was supported by national funds through the Foundation for Science and Technology, I.P. (Portuguese Foundation for Science and Technology) by the project UIDB/05064/2020 (VALORIZA—Research Center for Endogenous Resource Valorization), and Project UIDB/04111/2020, ILIND—Lusophone Institute of Investigation and Development, under project COFAC/ILIND/COPELABS/3/2020. Our thanks for the support to Almeida Technologies Ltda, São Miguel do Oeste, SC 89900-000, Brazil.

REFERENCES

[1] Philipos C Loizou. Speech enhancement: theory and practice. CRC Press, second edition, first issued in paperback edition, 2017.

[2] Nitish Krishnamurthy and John H. L. Hansen. Babble noise: Modeling, analysis, and applications. IEEE Transactions on Audio, Speech, and Language Processing, 17(7):1394–1407, Sep. 2009.

[3] V Andrei, H Cucu, and C Burileanu. Overlapped Speech Detection and Competing Speaker Counting—Humans Versus Deep Learning. IEEE Journal of Selected Topics in Signal Processing, 13(4):850–862, 2019.

[4] K Kinoshita, M Delcroix, S Araki, and T Nakatani. Tackling Real Noisy Reverberant Meetings with All-Neural Source Separation, Counting, and Diarization System. In ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 381–385, 2020.

[5] Ranya Aloufi, Hamed Haddadi, and David Boyle. Emotion Filtering at the Edge. In Proceedings of the 1st Workshop on Machine Learning on Edge in Sensor Systems, SenSys-ML 2019, pages 1–6, New York, NY, USA, 2019. Association for Computing Machinery.

[6] Ali Bou Nassif, Ismail Shahin, Shibani Hamsa, Nawel Nemmour, and Keikichi Hirose. CASA-based speaker identification using cascaded GMM-CNN classifier in noisy and emotional talking conditions. Applied Soft Computing, 103:107141, 2021.

[7] Yan Zhao, Zhong-Qiu Wang, and DeLiang Wang. Two-Stage Deep Learning for Noisy-Reverberant Speech Enhancement. IEEE/ACM Trans. Audio, Speech and Lang. Proc., 27(1):53–62, 2019.

[8] Muhammad Mohsin Kabir, M. F. Mridha, Jungpil Shin, Israt Jahan, and Abu Quwsar Ohi. A survey of speaker recognition: Fundamental theories, recognition methods and opportunities. IEEE Access, 9:79236–79263, 2021.

[9] Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Leibny Paola García-Perera, Fred Richardson, Réda Dehak, Pedro A Torres-Carrasquillo, and Najim Dehak. State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations. Computer Speech & Language, 60:101026, 2020.

[10] Lawrence R. Rabiner and Ronald W Schafer. Theory and applications of digital speech processing. Pearson/Prentice Hall, 2011.

[11] Ali I. Siam, Heba A. El-khobby, Mustafa M. Abd Elnaby, Hatem S. Abdelkader, and Fathi E. Abd El-Samie. A Novel Speech Enhancement Method Using Fourier Series Decomposition and Spectral Subtraction for Robust Speaker Identification. Wireless Personal Communications, 108(2):1055–1068, 2019.

[12] S Wang, S Li, and C Fan. A Complex Plane Spectral Subtraction Method for Vehicle Interior Speaker Recognition Systems. In 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE), pages 1–7, 2018.

[13] Rafizah Mohd Hanifa, Khalid Isa, and Shamsul Mohamad. A review on speaker recognition: Technology and challenges. Computers and Electrical Engineering, 90, mar 2021.

[14] Vítor H Nascimento and Magno T M Silva. Chapter 12 - Adaptive Filters. In Paulo S R Diniz, Johan A K Suykens, Rama Chellappa, and Sergios Theodoridis, editors, Academic Press Library in Signal Processing: Volume 1, volume 1 of Academic Press Library in Signal Processing, pages 619–761. Elsevier, 2014.

[15] Valderi Leithardt, Douglas Santos, Luis Silva, Felipe Viel, Cesar Zeferino, and Jorge Silva. A solution for dynamic management of user profiles in iot environments. IEEE Latin America Transactions, 18(07):1193–1199, 2020.

[16] Francisco García Encinas, Luís Augusto Silva, André Sales Mendes, Gabriel Villarrubia González, Valderi Reis Quietinho Leithardt, and Juan Francisco De Paz Santana. Singular spectrum analysis for source separation in drone-based audio recording. IEEE Access, 9:43444–43457, 2021.

[17] Yuqi Liu, Chao Sun, and Shouda Jiang. Kernel filtered-x lms algorithm for active noise control system with nonlinear primary path. Circuits, Systems, and Signal Processing, 37(12):5576–5594, 2018.

[18] Wemerson D Parreira, Márcio H Costa, and José CM Bermudez. Stochastic behavior analysis of the gaussian klms algorithm for a correlated input signal. Signal Processing, 152:286–291, 2018.

[19] Hamed Modaghegh, Hossein Khosravi R, Saeed Ahoon Manesh, and Hadi Sadoghi Yazdi. A new modeling algorithm - normalized kernel least mean square. In 2009 International Conference on Innovations in Information Technology (IIT), pages 120–124, 2009.

[20] Raoul Huys and Viktor K. Jirsa, editors. Nonlinear Dynamics in Human Behavior. Springer Berlin Heidelberg, 2011.

[21] Wemerson D Parreira, José Carlos M Bermudez, Cédric Richard, and Jean-Yves Tourneret. Stochastic behavior analysis of the gaussian kernel least-mean-square algorithm. IEEE Transactions on Signal Processing, 60(5):2208–2222, 2012.

[22] Matti Karjalainen Ville Pulkki. Communication Acoustics: An Introduction to Speech, Audio and Psychoacoustics. Wiley, 1 edition, 2015.

[23] Sefik Emre Eskimez, Peter Soufleris, Zhiyao Duan, and Wendi Heinzelman. Front-end speech enhancement for commercial speaker verification systems. Speech Communication, 99:101–113, 2018.

[24] Waad Ben Kheder, Driss Matrouf, Moez Ajili, and Jean-Francois Bonastre. A Unified Joint Model to Deal With Nuisance Variabilities in the I-Vector Space. IEEE/ACM Trans. Audio, Speech and Lang. Proc., 26(3):633–645, 2018.

[25] Hassan Taherian, Zhong-Qiu Wang, Jorge Chang, and DeLiang Wang. Robust Speaker Recognition Based on Single-Channel and Multi-Channel Speech Enhancement. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28:1293–1302, 2020.

[26] Mitchell McLaren, Luciana Ferrer, Diego Castán, and Aaron D Lawson. The Speakers in the Wild (SITW) Speaker Recognition Database. In INTERSPEECH, 2016.

[27] NIST. The nist year 2010 speaker recognition evaluation plan, 2010.

[28] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5329–5333, April 2018.

[29] Miao Zhao, Yufeng Ma, Min Liu, and Minqiang Xu. The speakin system for voxceleb speaker recognition challange 2021. arXiv preprint arXiv:2109.01989, 2021.

[30] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 351–355, 2018.

[31] Bernard Widrow, John R Glover, John M McCool, John Kaunitz, Charles S Williams, Robert H Hearn, James R Zeidler, JR Eugene Dong, and Robert C Goodlin. Adaptive noise cancelling: Principles and applications. Proceedings of the IEEE, 63(12):1692–1716, 1975.

[32] Renata C. Borges, Wemerson D. Parreira, and Márcio H. Costa. Design guidelines for feedforward cancellation of the occlusion-effect in hearing aids. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pages 607–610, 2019.

[33] Márcio Holsbach Costa. Theoretical transient analysis of a hearing aid feedback canceller with a saturation type nonlinearity in the direct path. Computers in biology and medicine, 91:243–254, 2017.

[34] Weifeng Liu, Puskal P. Pokharel, and Jose C. Principe. The kernel least-mean-square algorithm. IEEE Transactions on Signal Processing, 56(2):543–554, 2008.

[35] Bernhard Schoelkopf and Alexander J Smola. Learning with kernels. Adaptive Computation and Machine Learning series. MIT Press, London, England, June 2018.

[36] Aminadabe dos Santos Pires Soares, Wemerson Delcio Parreira, Everton Granemann Souza, Chiara das Dores do Nascimento, and Sérgio Jose Melo de Almeida. Voice activity detection using generalized exponential kernels for time and frequency domains. IEEE Transactions on Circuits and Systems I: Regular Papers, 66(6):2116–2123, 2019.

[37] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In Interspeech 2020. ISCA, oct 2020.

[38] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.
[39] Nikita Kuzmin, Igor Fedorov, and Alexey Sholokhov. Magnitude-aware probabilistic speaker embeddings. In The Speaker and Language Recognition Workshop (Odyssey 2022). ISCA, jun 2022.
[40] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.
[41] ranchlai. Speaker verification using resnetse and ecapa-tdnn. https://github.com/ranchlai/speaker-verification, 2021. Online; acessed at 11-October-2022.
[42] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. Computer Science and Language, 2019.
[43] David Snyder, Guoguo Chen, and Daniel Povey. MUSAN: A Music, Speech, and Noise Corpus, 2015. arXiv:1510.08484v1.
[44] Bernhard Schölkopf and Alexander J. Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, 06 2018.
[45] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. Sdr – half-baked or well done? In ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 626–630, 2019.
[46] Man-Wai Mak and Jen-Tzung Chien. Machine Learning for Speaker Recognition. Cambridge University Press, 2020.
[47] S A El-Moneim, M A Nassar, M I Dessouky, N A Ismail, A S El-Fishawy, and F E Abd El-Samie. Text-independent speaker recognition using LSTM-RNN and speech enhancement. Multimedia Tools and Applications, 79(33-34):24013–24028, 2020.
[48] Yanpei Shi, Qiang Huang, and Thomas Hain. Speaker re-identification with speaker dependent speech enhancement. In Interspeech 2020. ISCA, October 2020.
[49] Ganesh R. Naik, Dinesh K. Kumar, and Marimuthu Palaniswami. Multi run ica and surface emg based signal processing system for recognising hand gestures. In 2008 8th IEEE International Conference on Computer and Information Technology, pages 700–705, 2008.
[50] Xi-Lin Li and Tülay Adali. Independent component analysis by entropy bound minimization. IEEE Transactions on Signal Processing, 58(10):5151–5164, 2010.

VINÍCIUS A. SANTOS received bachelor's degree in computer science from University of Vale do Itajaí (2019). Currently, attending the University of Vale do Itajaí master program in Applied Computing. CEO from Almeida Technologies company. Current fields of interest are related to speech and image processing, distributed computing, and problems optimization.

WEMERSON D. PARREIRA received the bachelor's degree in mathematics from the Federal University of Uberlândia (UFU), Uberlândia, Brazil, the M.Sc. degree in electrical engineering from UFU, and the Ph.D. degree in Electrical Engineering from the Federal University of Santa Catarina, Florianópolis, Brazil, in 2002, 2005, and 2012, respectively. He is currently an Associate Professor of the School of Sea, Science and Technology at the University of Vale do Itajaí. He is a researcher at the Laboratory of Embedded and Distributed Systems. His experience is in mathematical modeling, with emphasis on Electrical, Computer, and Biomedical Engineering. His current research interests include reproducing kernels, linear and nonlinear adaptive filtering, image processing, speech processing, and statistical signal processing.

ANITA M. R. FERNANDES graduated in Science from University of Vale do Rio Doce (1989), graduated in Data Processing Technology from University of Vale do Rio Doce (1992), Master in Computer Science from Federal University of Santa Catarina (1996) and PhD in Production Engineering from the Federal University of Santa Catarina (2000). She is currently a professor at the University of Vale do Itajaí, research professor - UNIVALI and leader of the Applied Intelligence Group at UNIVALI. Has experience in Computer Science, with emphasis on Artificial Intelligence, working mainly on the following topics: artificial intelligence applied to health, education and the environment, data science, and big data. She is also a professor of the Master in Applied Computing at UNIVALI.

RAÚL GARCÍA OVEJERO received the Industrial Technical Engineering degree in electricity, specializing in industrial electronics, in 1997, the master's degree in occupational risks, in 2006, and the Ph.D. degree in industrial engineering from the Department of Mechanics, University of Salamanca (USAL), Spain, in 2014. He became an Industrial Engineer, in 2001. Currently, he is a contracted Professor Doctor with USAL and a Researcher with the Expert Systems and Applications Laboratory (ESALab). Throughout his training, he has followed a line of research linked to the engineering field, mainly focused on smart fabrics. He has coauthored articles published in several JCR indexed journals.

VALDERI R. Q. LEITHARDT received the Ph.D. degree in computer science from INF-UFRGS, Brazil, in 2015. He is currently a Professor with the Polytechnic Institute of Portalegre and a Researcher integrated with the VALORIZA Research Centre for Endogenous Resource Valorization. He is also a Collaborating Researcher at the Expert Systems and Applications Laboratory (ESALab), University of Salamanca, Spain. His mainline of research interests include distributed systems with a focus on data privacy, communication, and programming protocols, involving scenarios and applications for the Internet of Things, smart cities, big data, cloud computing, and blockchain.