An on-line VAD based on Multi-Normalisation Scoring (MNS) of observation likelihoods

Igor Odriozola, Inma Hernaez, Eva Navas

Aholab Signal Processing Laboratory, University of the Basque Country (UPV/EHU), Bilbao 48016, Spain

Abstract

Voice Activity Detection (VAD) is an essential task in expert systems that rely on oral interfaces. The VAD module detects the presence of human speech and separates speech segments from silences and non-speech noises. The most popular current on-line VAD systems are based on adaptive parameters which seek to cope with varying channel and noise conditions. The main disadvantages of this approach are the need for some initialisation time to properly adjust the parameters to the incoming signal and uncertain performance in the case of poor estimation of the initial parameters. In this paper we propose a novel on-line VAD based only on previous training which does not introduce any delay. The technique is based on a strategy that we have called *Multi-Normalisation Scoring* (MNS). It consists of obtaining a vector of multiple observation likelihood scores from normalised mel-cepstral coefficients previously computed from different databases. A classifier is then used to label the incoming observation likelihood vector. Encouraging results have been obtained with a Multi-Layer Perceptron (MLP). This technique can generalise for unseen noise levels and types. A validation experiment with two current standard ITU-T VAD algorithms demonstrates the good performance of the method. Indeed, lower classification error rates are obtained for non-speech frames, while results for speech frames are similar.

Keywords: VAD, observation likelihood, MNS, on-line speech processing

1. Introduction

Voice activity detection (VAD) is a very important part of expert systems based on speech interfaces. Using VAD, audio signals are split into autonomous speech segments before being passed to the subsequent modules. Two kinds of errors must be considered: silence or noise segments being passed as speech (the *non-speech error rate*) and speech segments being misclassified as silences and then not being passed to the processing system (the *speech error rate*). Both must be kept low of course, but their importance depends on the needs and design of the expert system using the VAD.

VAD is typically the first module employed in acoustic processing systems. It is profusely used in the development of all kinds of expert systems. In Mporas et al. (2010) the authors use Automatic Speech Recognition (ASR) technology with a VAD to develop a dialogue system in a motorcycle environment. Principi et al. (2015) describe an integrated system for processing voice emergency commands using a VAD followed by ASR. VAD and ASR technologies constitute the core of the speech interface in a system using a serious game to support therapy for mental disorders in Kostoulas et al. (2012). All these systems, based on ASR, require a very low ratio of lost speech frames in order for all the meaningful audio frames to be available to the recogniser. On the other hand, if non-speech segments are passed as speech the recogniser will still be able to detect them, as they typically have a silence (or non-speech) model. The main purpose of VAD in ASR interfaces is to eliminate long silences and split the audio stream into shorter, manageable segments. Additionally computation time is reduced and consequently so is the decoding response time.

ASR is not the only technology that requires a good VAD module. Tirumala et al. (2017) identify VAD as one of the research areas for speaker recognition. For instance, VAD is included in an intelligent porch system where people are identified by their voices before entering the house (Kuan et al., 2012). VADs are also an important module in speaker segmentation and clustering systems, such as the diarisation system presented in Martínez-González et al. (2017). In

addition, VAD is an essential module in expert systems that include emotion identification (Alonso et al., 2015). For speaker and emotion recognition systems the VAD employed requires a very low number of erroneously classified silence or noise frames, since silences or noise frames do not convey emotion or the speaker's identity. A high non-speech error rate will thus lower the performance of the system. If however some speech frames are lost, the system will still be able to perform correctly.

Current VADs can be tuned to behave closer to one mode or to the other, though the ideal behaviour would of course be to reduce both non-speech and speech error rates as far as possible.

When the oral interface of an expert system picks up audio signals by means of different devices and in different environments, the VAD has to cope with different recording conditions, channel characteristics and noise levels. This is in fact the greatest challenge for the current ASR systems (Virtanen et al., 2012). VAD systems currently adapt different parameters to adjust to changing background noise conditions. However, this approach has its shortcomings: on the one hand, there is a need for an initialisation time over a segment to adjust the parameters, which introduces an undesirable delay. On the other hand, any incorrect estimation of the parameters will lead to uncertainty in the performance of the system (Graf et al., 2015). Training the VAD beforehand is one way to avoid the initial adaptation, but the trained system should be able to generalise to unseen channels or background noises. On-line VAD decision making is still a challenge.

From the point of view of acoustic features, very different parameters have been investigated: periodicity measure (Tucker, 1992; Hautamäki et al., 2007), zero-crossing rate (Benyassine, 1997), pitch (Chengalvarayan, 1999), Short Term Energy (STE) (Rabiner & Sambur, 1975) and Long Term Energy (LTE) (Ghosh et al., 2011; Ma & Nishihara, 2013), spectrum analysis (Woo et al., 2000; Marzinzik & Kollmeier, 2002), cepstral distance (Pollak & Sovka, 1995), Linear Predictive Coding (LPC) (Nemer et al., 2001) and combinations of different features

(Tanyer & Özer, 2000). More recent research has been focused on using multiple features to train a statistical model or classifier using machine learning techniques rather than on exploring more discriminative new acoustic features, which was the traditional trend.

Both Gaussian Mixture Models (GMM) and Hidden Markov Models (HMMs) have been tested in the context of VAD. In Tatarinov & Pollák (2008), speech and non-speech segments are modelled by two HMMs. A simple grammar is used to model transitions from one HMM to the other and voice detection becomes a task of finding the best path through a recognition network. It is shown that a simple HMM-based VAD functions properly when clean signals are considered. In Kingsbury et al. (2002) the same HMM strategy is followed to deal with background noise, but acoustic features and normalisation operations are used along with the results conveyed by the HMMs. In Veisi & Sameti (2012) several noisy HMMs are trained to detect different noisy non-speech segments. In this paper we also use the approach of scores generated by the HMMs.

Varela et al. (2011) addresses the problem of far-field speaker interference in human-machine oral interaction. A decision tree (DT) is trained using the scores of speech/non-speech HMMs and additional information related to far-field speech. A Support Vector Machine (SVM) is used in Enqing et al. (2002) to discriminate between speech and non-speech, and improved versions include Signal to Noise Ratio (SNR) information as in Ramirez et al. (2006a,b). Hybrid SVM/HMM architectures are also proposed for VAD in Tan et al. (2014) to retain the discriminative and non-linear properties of SVM while modelling the inter-frame correlation through a HMM. Results show a better performance for the SVM-based VAD system. However, relatively high speech error rates are still obtained. Our proposed VAD outperforms this technique and obtains a speech error rate more than three times lower.

More recently, neural networks (NN) have appeared in the literature of VAD approaches. For instance, Hughes & Mierle (2013) uses a recurrent neural network (RNN) with perceptual linear prediction (PLP) features testing clean signals. Convolutional neural networks (CNN) are also used in Thomas et al. (2014)

with mel-spectral coefficients, but adaptation with supervised data is needed for unseen channels. In Obuchi (2016) feature vectors consisting of log-mel filterbank energies are fed into a DT, an SVM and a CNN classifier. However, in this VAD approach several parameters must be adjusted to adapt to different noise conditions.

Regarding on-line performance, the current deep learning approaches tend to have very long inference times, mainly because neural network architectures are normally designed to be as complex as possible without considering real-time limitations (Sehgal & Kehtarnavaz, 2018). An exception is the system introduced in Zhang & Wu (2013), where a collection of different acoustic features are used to train a deep-belief neural network (DBNN). Extensive experimental results where different types of noise are tested show that it outperforms several reference VADs, even in real time. Nevertheless, this system has to compute almost 300 features in each frame, which increases system complexity. By contrast, our approach seems to get better results and is much simpler.

In this paper we present a simple but highly effective VAD based on a method that we have called *Multi-Normalisation Scoring* (MNS). This consists of classifying multiple observation likelihoods generated by an HMM trained with normalised Mel-Frequency Cepstral Coefficients (MFCC) corresponding to silence audio segments. Our proposed VAD technique makes use of a classifier which is trained beforehand, so that only a classification task needs to be performed when a new incoming speech frame arrives. This means that results are obtained on-line frame by frame and there is no need to adjust any parameter, so no initialisation period is needed. Furthermore, in comparison with two current standard ITU-T VAD algorithms, our VAD has proved to perform much better in labelling non-speech frames and to obtain similar results in labelling speech frames without increasing computing time. The VAD has been tested for different types of noise, as well as several SNR. The results show that our proposed VAD technique is able to generalise. However, noises not seen during training provoke a slight decrease in the results.

Section 2 describes the general architecture of the VAD system proposed in this paper. Section 3 describes the MNS method and its motivation. Section 4 provides a short overview of the databases used. To assess the performance of the new VAD, several databases have been chosen in an attempt to cover a variety of contexts and use a considerable amount of test speech material. The results of different experiments (under both clean and noisy conditions) are shown in Section 5. Section 6 describes a validation experiment comparing the results with two standard VADs, and some conclusions are finally drawn in Section 7.

2. General architecture of the system

The on-line VAD technique proposed in this paper consists of three core blocks, as shown in Fig. 1. The input to the system is a vector of MFCCs obtained from the current signal frame, and the output is a VAD label: *speech* or *non-speech*.

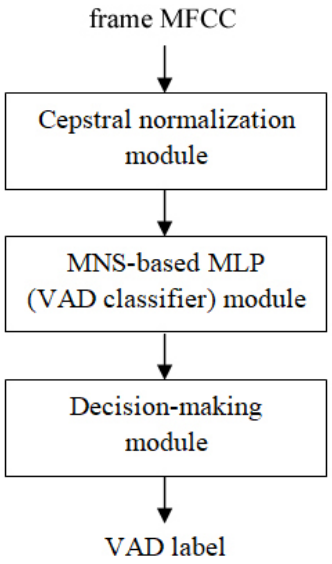

Figure 1: General architecture of the on-line VAD technique proposed here.

The three core blocks are:

1. Cepstral normalisation module: the acoustic features (MFCCs) of the incoming signal frame are normalised using different normalisation factors.
2. MNS-based MLP (VAD classifier) module: this module classifies a vector obtained by our proposed Multi-Normalisation Scoring (MNS) method, using a Multi-Layer Perceptron (MLP).
3. Decision-making module: this implements a finite-state automaton to make immediate decisions in order to cope with glitches and enhance the results.

Block 2 implements the method presented here, and is described throughout the paper. Blocks 1 and 3 are described in more detail in the following subsections.

2.1. Cepstral normalisation

Cepstral normalisation is essential to develop the VAD proposed in this paper. Indeed, as demonstrated in earlier works (Westphal, 1997), the observation likelihoods generated by the silence GMM trained with normalised MFCCs follow a fairly discriminative pattern for speech and non-speech frames. The VAD proposed in this work takes advantage of this characteristic.

Overall, parameter normalisation is indispensable to create robust acoustic models and cope with audio signals captured in different environments. The spectral subtraction approach of Boll (1979) is well established in the ASR field for compensating for the differences (channels, background noise, etc.) in the incoming signals. However, the most common practice is to perform CMVN (Cepstral Mean and Variance Normalisation) on the extracted features, as it outperforms spectral subtraction techniques (Garner, 2011).

As explained in Huang et al. (2001), the mean of an MFCC over N frames conveys the spectral characteristics of the current microphone and room acoustics. At the limit, when $N \longrightarrow \infty$ for each utterance, the means from utterances from the same recording environment can be expected to be the same. Thus, cepstral mean normalisation (CMN) permits the removal of a stationary, linear

channel transfer function; and variance normalisation (CVN) helps to compensate for the reduction of the variance of the MFCCs due to additive noise.

The classic CMVN approach (Liu et al., 1993, 1994) seeks to estimate mean and variance vectors per cepstral feature (MFCC). The feature vectors are then shifted and scaled by the estimated means and variances, so that each normalised feature has zero mean and unit variance. An effective solution for calculating reliable means and variances is to estimate them using the whole utterance (*off-line* performance). This utterance-based normalisation can result in undesirable delays, since utterance processing cannot begin until the last frame arrives. In time-synchronous (or *on-line*) systems, windows of a minimum length of 150-200 ms are typically used as a compromise between the quality of the estimated means and variances and the latency. Once an initial value is estimated, some type of recursive normalisation is usually applied in which the long-term estimates for the means and variances of the cepstral features are incrementally updated.

The initial values for means and variances can be estimated using the first M frames (and then adapting recursively). Correct estimation of these initial values depends heavily on whether these M frames contain speech or not. If there is no speech in them, computed variance values will be very small, which will strongly amplify the amplitude of the normalised signal, and vice versa. In consequence, a good estimation of the initial values for means and variances is of the utmost importance. This issue can be overcome by using the method introduced in this paper, which is based on applying multiple normalisation factors to cepstral features. This enables decisions to be made frame by frame with no need to use a window.

2.2. Decision-making module

As the decision of speech/non-speech is made frame by frame, very short segments labelled as speech can appear in the output of the VAD. These short segments usually correspond to noises and glitches and degrade the performance of the following processing system. In an off-line implementation there is usually

post-processing, but on-line implementation means making immediate decisions. In our on-line implementation, a classic state-diagram is implemented (see Fig. 2). Two parameters are considered: *minimum speech duration* (T_{min_speech}) and *minimum silence duration* (T_{min_sil}), which set the minimum number of frames that a segment must contain to be considered as speech or silence (non-speech), respectively. As can be seen in the figure, if the VAD changes its state from non-speech to speech (or vice versa) in a given frame, the next T_{min_speech} frames (or T_{min_sil}) are also analysed. If the result of checking these frames matches the state of the current frame, a state change is made; otherwise it is assumed that there has been a glitch and the VAD does not change its state. Obviously, this method adds a short delay each time a state change is found, but no delay is added when the same state is maintained. This enables the system to completely recover during non-transitional segments.

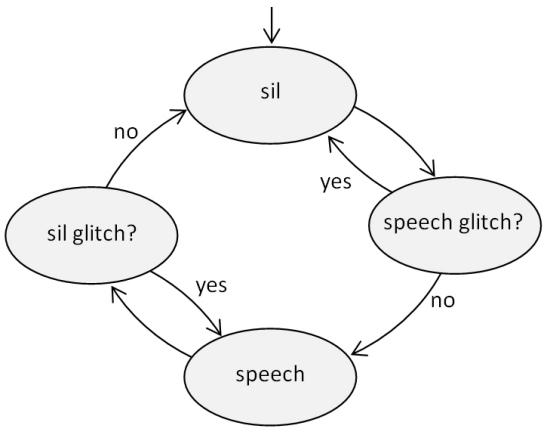

Figure 2: State-diagram for the on-line implementation of the decision-making module for glitch-removal.

For the experiments carried out in this paper, a minimum segment duration of 15 frames was empirically chosen for both T_{min_speech} and T_{min_sil}.

3. The basis of the MNS method

3.1. Observation likelihood

In speech recognition, audio segments corresponding to the same recognition unit (word, phone, triphone etc.) are gathered and processed in order to extract acoustic features (typically MFCCs) from them and train a different acoustic model for each unit. HMM is a very popular acoustic model, since it not only models the likelihood of a new observation vector but also the sequentiality of the observations.

Observation likelihoods are generated by GMMs, each of which corresponds to an HMM state. For an observation vector o_t, the observation likelihood b_j of a GMM at the j_{th} state is calculated as shown in eq. 1.

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} N\left(o_t; \mu_{jm}, \Sigma_{jm}\right) \tag{1}$$

where M is the number of mixture components, c_{jm} is the weight of the m^{th} component and $N(\cdot; \mu, \Sigma)$ is a multivariate Gaussian with mean vector μ and covariance matrix Σ.

3.2. The observation likelihood of the central state of a three-state silence HMM

The central state in a three-state HMM is a priori the most stable state of the model, since the left and right states have to cope with transitions between models. It makes sense to assume that the same goes for the silence HMM, where states at the ends have to model transitions between silence and speech.

An illustrative example is provided in Fig. 3, which shows the log-likelihoods generated by the GMM of each HMM state (s_0, s_1 and s_2) through an utterance composed of three words (notice the mouth click just before the second word). The observation likelihood curve generated by the central-state (s_1) GMM seems much more discriminative than the ones at the ends, which are more irregular.

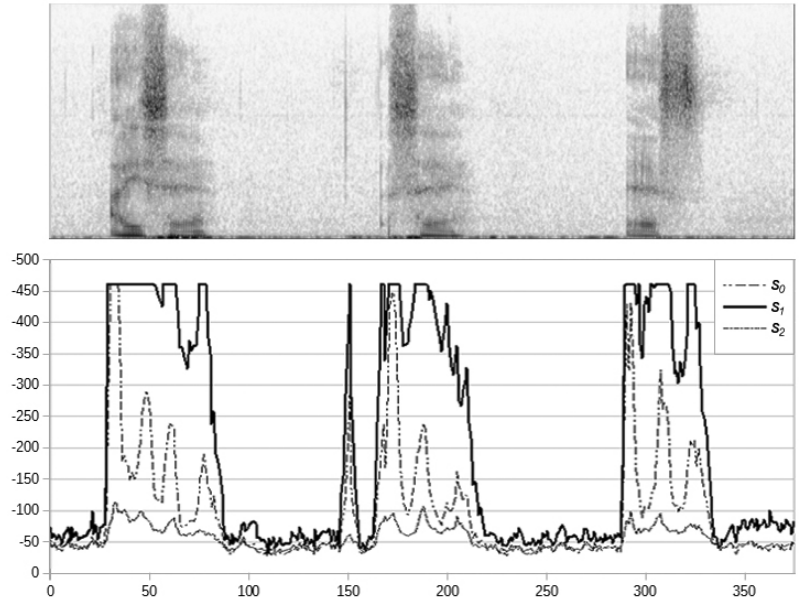

Figure 3: Spectrogram of an utterance consisting of three words (top) and observation log-likelihoods over time (frames) generated in the left state (s_0), central state (s_1) and right state (s_2) of the silence HMM trained with normalised MFCCs (bottom).

3.3. The Multi-Normalisation Scoring (MNS) method

The MNS method consists of generating multiple observation likelihood scores by normalising the MFCCs using means and variances computed from different speech datasets obtained under different recording conditions. The observation likelihood vectors thus obtained can characterise the behaviour of the speech and non-speech frames in different conditions. As an illustrative example, Fig. 4 shows the behaviour of the scores obtained by normalising a signal picked up from near (B signal, top) and another from afar (E signal, bottom) with the pre-calculated means and variances obtained from four datasets recorded simultaneously at four distances: close (B), desktop (C), medium (D) and far (E) (for more details about the code names see Section 4).

Assuming that the differently normalised scores of the non-speech segments follow a pattern (see score vector s_i in Fig. 4), it is likely that the speech scores

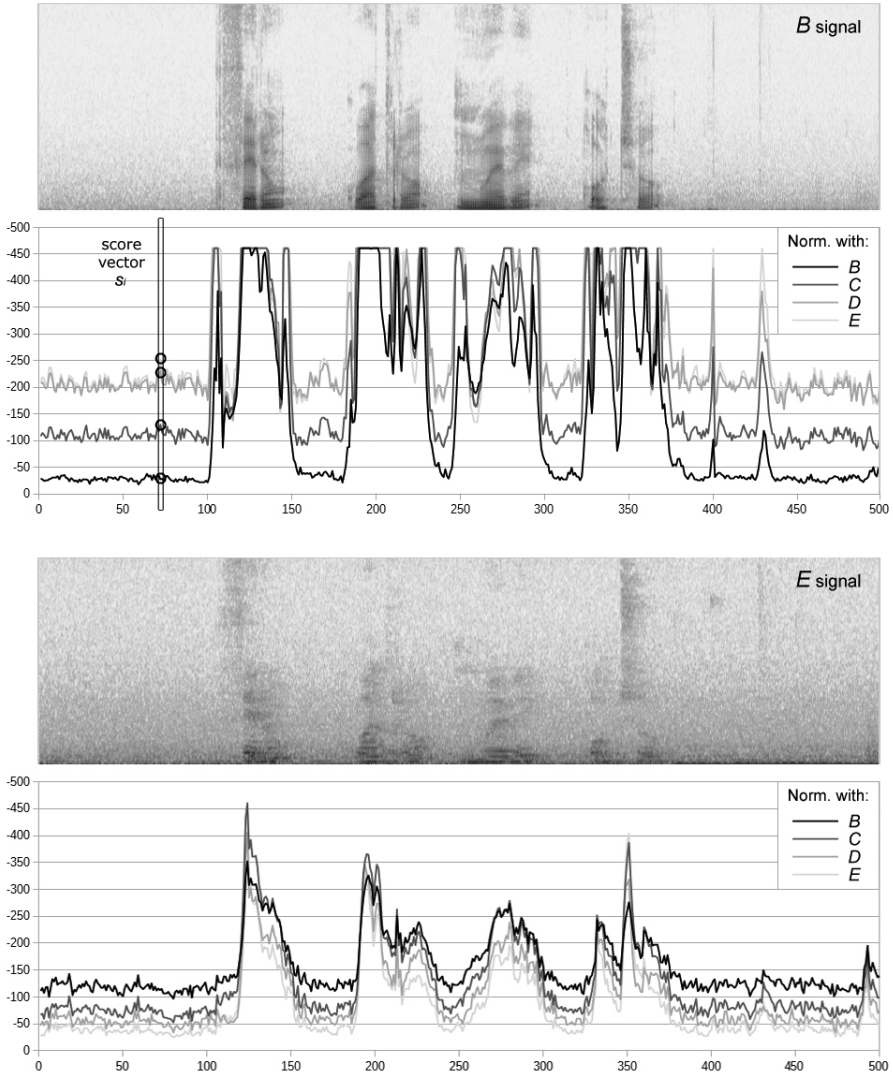

Figure 4: Spectrogram and central-state silence HMM observation log-likelihoods of a B signal (top) and an E signal (bottom) over time (frames) for different normalisation modes using pre-calculated means and variances from datasets B, C, D and E. The vertical narrow box marks the vector of scores in frame i.

do likewise. If so, only a good classifier would be needed to detect those patterns and classify the vector as belonging to a speech or non-speech frame.

4. Speech databases

Four speech databases have been used in this study. Firstly, we used the *Basque Speecon-like* database (Odriozola et al., 2014), specifically the *close-talk* channel, to train the HMM for silence frames. Using this HMM, an MLP was trained by applying the MNS method to the files of the *Basque Speecon-like* database and a subset of the *Spanish Speecon* database used in an ECESS evaluation campaign of voice activity and voicing detection (Kotnik et al., 2008). The latter contains four channels or datasets corresponding to different recording distances: C_0, C_1, C_2 and C_3.

The initial VAD experiment was performed by testing the files from a third database: the *TIMIT Acoustic-Phonetic Continuous Speech Corpus* (Garofolo et al., 1993). The second VAD experiment was carried out by testing the system with noisy signals. For that purpose, the *Noisy TIMIT* speech database (Abdulaziz & Kepuska, 2017) was considered, in particular the *babble* noise dataset and *white* noise dataset *Test* blocks. Each dataset comprises 10 subsets each of which corresponds to a different SNR (from 50 to 5 dB, in 5 dB steps). For the third VAD experiment, 4 of these 10 subsets (35, 25, 15 and 5 dB) were also included in the training material to train a new MLP, with the purpose of making the system more robust against noise. The files tested were the same as in the second experiment.

Finally, the results were compared using two standard VAD algorithms, and the same files as in experiments 2 and 3 were tested: the *Test* blocks of the *babble* noise and the *white* noise datasets of the *Noisy TIMIT* speech database.

Table 1 shows the main characteristics of the databases and the channels of each database used for this research. Each channel's code name as indicated in the table is used hereinafter.

5. MNS-based VAD experiments

The VAD accuracy experiments carried out in this study consist of assessing the ability of the system to discriminate between speech and non-speech

Table 1: Main characteristics of the databases (and channels) used in this paper.

Database	*Basque Speecon-like*		*Spanish Speecon - ECESS*				*TIMIT*	*Noisy TIMIT*	
Code	*R*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*
Channels used	Close-talk	Desktop mic	very close (C_0)	close (C_1)	medium (C_2)	far (C_3)	Headset-mounted and far-field mic	*babble* noise (50-5 dB)	*white* noise (50-5 dB)
Language	Basque		Spanish				English (USA)	English (USA)	
Environment	Office		Office, public place, entertainment, car				Studio	Studio + additive noise	
Speakers	230		60				630	630	
Files / speaker	316		17				10	600	
Total content (h)	109.95		1.41				5.37	322.2	
Speech content (%)	47.90		51.77				86.57	86.57	
Labelling	Phonetic Forced Alignment		Manually				Manually	From *TIMIT*	
Sample rate	16 kHz		16 kHz				20 kHz (down-sampled 16 kHz)	16 kHz	

segments in terms of the *non-speech error rate* (ER_0) and *speech error rate* (ER_1). These two rates are computed as the fractions of the non-speech frames and speech frames that are incorrectly classified ($N_{0,1}$ and $N_{1,0}$, respectively) as a proportion of the number of real non-speech frames and speech frames in the whole database (N_0^{ref} and N_1^{ref}, respectively), as shown in equation 2. In addition, the TER (total error rate) is also computed as the quotient between the total number of incorrectly classified frames and the total number of frames (equation 3).

$$ER_0 = \frac{N_{0,1}}{N_0^{ref}} \times 100; ER_1 = \frac{N_{1,0}}{N_1^{ref}} \times 100 \quad (2)$$

$$TER = \frac{N_{0,1} + N_{1,0}}{N_0^{ref} + N_1^{ref}} \times 100 \quad (3)$$

The silence HMM was trained using the R database. Acoustic parameters include 13 MFCCs and 13 first and 13 second order derivatives, and they were modelled with 32 mixture GMMs. The audio signals were windowed into 25 ms length frames picking up a frame each 10 ms. For the training of the silence HMM, these parameters were normalised using the means and variances computed from the files belonging to the same session (all the utterances corresponding to the same speaker).

Different classifiers were tested to see whether the scores obtained using the MNS method were valid, and the best results were obtained using a Multi-Layer Perceptron (MLP) (Widrow et al., 1988; Delashmit & Manry, 2005), a classifier that can distinguish data that are not linearly separable (Collobert & Bengio). For these experiments, MLPs were trained using WEKA (Waikato Environment for Knowledge Analysis), a popular free, open-source software written in the Java language for data-mining tasks (Holmes et al., 1994; Hall et al., 2009).

5.1. MNS-based VAD experiment using an MLP

To prepare the data to train the MLP, 1 020 files of each of the datasets R, A, B, C, D and E were considered. All the files were processed to obtain observation likelihoods (generated by the central-state GMM of the three-state silence HMM trained with dataset R), after normalising the MFCCs using the means and variances precomputed from each dataset. Thus, vectors of 6 scores were generated from the frames of all the files belonging to each dataset. Altogether, 3 096 632 score vectors were obtained, 49.08 % of which correspond to speech and 50.92 % to non-speech, i.e. they are well balanced (for further details, see Table 3).

The MLP used for this task contains 6 nodes in the input layer (one for each score) and 2 nodes in the output layer (one for each category: *speech* and

non-speech). Half the sum of both node amounts (4 nodes) were chosen for the hidden layer.

To test the MNS-based MLP, a separate database was chosen: dataset F (*TIMIT*). All the files (6 300) from this dataset were processed in on-line mode; i.e. MFCCs were normalised on-line with the means and variances computed from subsets R, A, B, C, D and E, giving a vector of 6 scores frame by frame. Then each score vector was classified by the MLP. The results of this experiment are shown in Table 2.

Table 2: TER, ER_0 and ER_1 of the on-line VAD experiment on the *TIMIT* corpus.

TER	ER_0	ER_1
4.98	19.68	2.70

The results of the on-line MNS method proposed in this work can be considered as quite good when compared with other VADs, as shown in Section 6. ER_1 is low, which means that speech frames are quite correctly classified, so very few of them would be left out. However, ER_0 is quite high, with most of the errors being made at the ends of the speech segments. This means that almost one in five non-speech frames would pass on to the speech processing system.

5.2. MNS-based VAD experiment in noisy conditions

The MNS technique introduced in the previous section must be evaluated by testing noisy speech files in order to assess the robustness. For that purpose, two noisy datasets were considered: G (*babble* noise) and H (*white* noise), the most natural noises for a system hosted on a remote server. Both datasets contain 10 subsets with different SNRs, ranging from 50 to 5 dB in 5 dB steps.

The experiment consisted of testing the same MLP used in the previous section (trained with clean signals) with the files corresponding to the *Test* blocks of datasets G and H (noisy signals): in total, 25 200 files (1 260 files in

each of the corresponding 20 different SNR subsets). Fig. 5 shows the error rates obtained for both *babble* noise signals (orange dotted lines) and *white* noise signals (grey dotted lines) at different SNRs. As a benchmark, the results presented in Table 2 when testing dataset F (clean signals) are also shown in the figure, as horizontal black dotted lines.

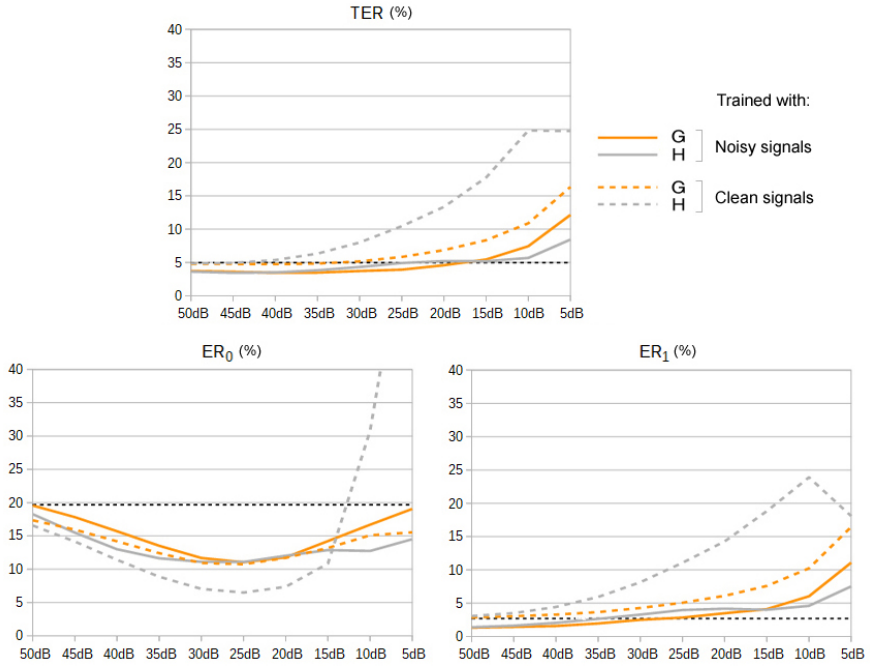

Figure 5: Error rates obtained by testing all the SNR subsets of datasets G and H (noisy signals) when the MLP is trained using clean signals (dotted lines) and clean and noisy signals (solid lines); error rates obtained by testing clean signals are also shown as horizontal lines for reference (black dotted line).

The results show that testing noisy signals affects VAD performance: the ER_1 curve deteriorates in general, and the ER_0 results show a more irregular behaviour. Note the high ER_0 values obtained for white noise at the lowest SNRs, probably due to the fact that white noise introduces energy at all frequencies and the MLP tends to classify all the frames as speech frames. Generally speaking, some deterioration was to be expected, since the scores used to train the MLP come from clean signals. That is why a new MLP was trained

using scores obtained from noisy signals, as described in the next section. It should also be noted that the TER curve looks like the ER_1 curve, due to the fact that the G and H signals from the *TIMIT* database contain more speech (86.57%) than non-speech (see Table 1).

5.3. Training the MLP with noisy signals

In this new experiment, noisy signals were included in the MLP training process. For this purpose, 4 subsets from each dataset G and H were chosen: the 35, 25, 15 and 5 dB (specifically, their *Train* blocks). Thus, the MLP training data now include the signals from the *Train* blocks of each of these 8 subsets, together with the files used in the MLP training process in the previous experiment. The results indicate whether the MLP is able to generalise when classifying signals with different SNRs.

Since the *TIMIT* database is unbalanced in terms of the amount of speech and non-speech frames (see Table 1), a large number of speech frames were randomly discarded from the *Train* block files. Table 3 shows the total numbers of frames used per dataset.

Table 3: Datasets and numbers of frames considered to train the MLP with noisy signals.

	R, A	*B, C, D, E*	*G, H*	TOTAL
non-speech fr	299 972×2	244 211×4	190 052×8	3 097 204
speech fr	234 628×2	262 647×4	244 003×8	3 471 868
Total fr	534 600×2	506 858×4	434 055×8	6 569 072

The score vectors now contain 14 elements: 6 scores obtained from the clean signals, and 4 from each of the subsets with noisy signals. Thus, the MLP configuration selected for this experiment is this: 14 nodes in the input layer, 2 nodes in the output layer, and 8 nodes for the hidden layer.

The files tested are the same ones as in the previous experiment (see Subsection 5.2). Fig. 5 shows the error rates obtained at different SNRs for both *babble* noise signals (orange solid lines) and *white* noise signals (grey solid lines).

In the light of the results, there is an improvement in ER_1 at all levels of noise, for both dataset G and dataset H, even when testing clean signals. Regarding ER_0, the improvement is remarkable for *white* noise at high noise levels. The big improvement in ER_1 together with the imbalance of the database results in an overall improvement in TERs.

To show the impact of using noisy signals on the results obtained by testing clean signals, Table 4 presents the results of the experiment performed under clean conditions (see Table 2) along with the results obtained in this last experiment for the cleanest signals (50 dB subset). The results are actually even slightly better now.

Table 4: TER, ER_0 and ER_1 of the VAD experiment including noisy signals in the training process.

		TER	ER_0	ER_1
Exp. with clean signals		4.98	19.68	2.70
Exp. with noisy signals	*babble* noise 50 dB	3.73	19.57	1.32
	white noise 50 dB	3.63	18.26	1.40

5.4. Generalisation to other types of noise

We have shown that the MLP trained with 4 subsets (the 35, 25, 15 and 5 dB) of each dataset G and H is able to generalise results for the rest of SNR values. However, seeking to learn whether the MLP trained with noisy signals can also generalise for other types of noises, we tested the MLP trained with noisy signals (see Subsection 5.3) with signals containing other types of noise. So now the test set comprises the files from the *Test* blocks of datasets G (*babble* noise) and H (*white* noise) along with the files belonging to the same *Test* blocks of the datasets *blue*, *pink*, *red* and *violet* (1260 files in each SNR subset; 12 600 in each dataset).

Figure 6 shows the TERs obtained at different SNRs for the various noise types. For SNRs equal to or greater than 35 dB there is no degradation when

signals that have unseen noises are tested. For smaller SNRs, the deterioration is not very large: the maximum is for *violet* noise, which degrades by about 7 points at 15 dB with respect to both references. For *red* noise, the system actually behaves better than when the reference noises are tested.

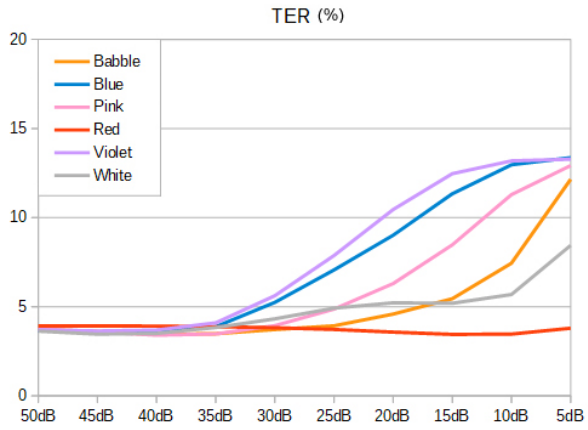

Figure 6: TERs obtained by testing all the SNR subsets of all types of noise of the Noisy TIMIT, using the MLP trained with signals containing *babble* noise and *white* noise.

6. Final experiments

Two on-line VAD algorithms standardised by ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) were tested to check the validity of the VAD technique proposed here. The algorithms belong to series G (*Transmission systems and media, digital systems and networks*), where *G.710* - *G.729* are devoted to *Coding of voice and audio signals*. The first algorithm is *G.720.1* (, ITU), which is actually a Generic Sound Activity Detector (GSAD) that can operate on 8 or 16 kHz audio input, with a VAD module. The second algorithm is *G.729* (, ITU), an 8 kbit/s speech coder that manages 8 kHz input signals, which relies on a VAD module described in its Annex B (also known as *G.729b*). Both systems use a 10-ms frame length and frame shift, and no look-ahead is needed (no delay, just the frame duration).

Further details are provided in Table 5 for both ITU systems[1] and our proposed VAD technique.

Note that the computation time is the average time per file needed by each system in a test where 10 080 files are processed, using the same computer and under the same conditions. It can vary from one computer to another, but it gives some idea of the ratios between them. Additionally, regarding the hangover scheme, the G.729b and our proposed VAD technique follow a similar state machine, and introduce a delay while it is decided whether there is a change or not. In the case of G.720.1 a conservative scheme is followed, where active indicators are emitted until a silence segment is detected.

Table 5: Comparison of some important parameters of the VAD in *G.720.1* (ITU-T), the *G.729b* algorithm (ITU-T) and our proposed VAD technique.

	G.720.1 VAD	*G.729b*	Prop. method
Bandwidth (kHz)	8, 16	8	16
Frame duration / shift (ms)	10 / 10	10 / 10	25 / 10
Computation time (ms per file)	26.8	34.87	30.7
Smoothing	No	Yes	Yes
Initialization (No. frames)	200 inactive	32	0

To test the VADs, the same data were used as in Sections 5.2 and 5.3. To test the G.729 coder VAD, the files had to be down-sampled to 8 kHz. Fig. 7 shows the error rates obtained by the two ITU algorithms (dotted and dashed lines) and our proposed VAD technique (solid lines): TER (top), ER_0 (bottom left) and ER_1 (bottom right) testing the *Test* blocks of both datasets G and H.

Regarding the ER_0, our proposed VAD technique lets at most 20 % of non-speech frames pass as speech. The minima of both ITU systems are over 30 %,

[1]The software for both systems can be downloaded from the ITU website: `http://www.itu.int/rec/T-REC-G.720.1-201001-I` and `http://www.itu.int/rec/T-REC-G.729-201206-I`, respectively.

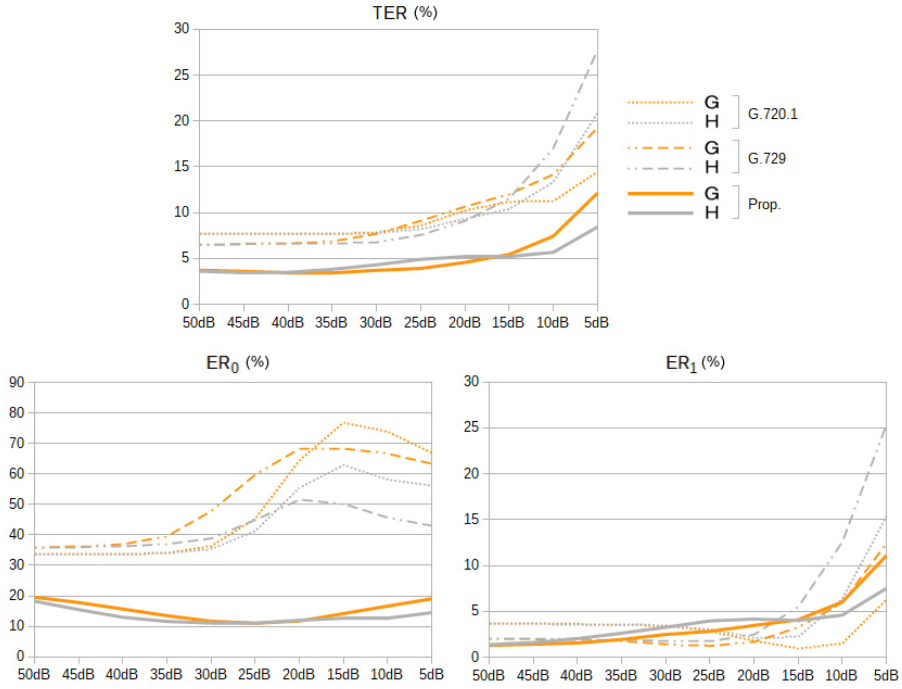

Figure 7: Error rates obtained using the ITU-T G.720.1 and G.729b standard VADs testing all the SNR subsets of datasets G and H, along with the results of the proposed VAD technique (solid lines).

and they show a significant increase as SNR gets lower, especially for *babble* noise signals. This means that many non-speech frames are classified as speech when signals are noisy.

With regard to the ER_1, the results for G.720.1 and our proposed VAD technique are quite similar for *babble* noise. For *white* noise the results are similar for low-noise signals, but our proposed technique performs better on the noisiest data. In the case of G.729b, our proposed VAD obtains better results for SNR higher than 25 dB. At 10 and 5 dB, G.729b gets better results for *babble* noise and the MNS-based VAD for *white* noise.

In general, ER_1 results obtained by the ITU algorithms and the MNS-based VAD are comparable for high SNR signals. By contrast, for low SNR signals

the results show different behaviour for *babble* noise and *white* noise. For *babble* noise, G.720.1 gets similar results, and G.729b gets better results. For *white* noise, better results are obtained by the MNS-based system. Nevertheless, it is worth noting that ER_0 values are very high for the two ITU algorithms, which means that both systems tend to classify non-speech frames as speech when testing noisy signals.

In conclusion, our proposed VAD technique gets better TER at all noise levels. Due to the imbalance between the amount of speech and non-speech frames, the TER curves are similar in shape to those obtained for ER_1 but are shifted proportionally by ER_0. One of the advantages of the ITU systems is that they can adapt to different noise conditions on-line; however, they need an initialisation time to adjust the main parameters. In comparison, our MNS-based system is able to generalise for noise types that are not included in the training process and it requires no initialisation time, since the results do not depend on any previous frame.

7. Conclusions and future work

In this paper, we introduce a novel VAD that can be trained beforehand, so that it does not need any adaptive parameters or therefore any initialisation time to adjust those parameters. The VAD technique is based on the multi-normalisation scoring (MNS) method. MNS is based on generating an observation likelihood vector for each frame using the central-state GMM of a three-state silence HMM and normalising the cepstral features with different sets of means and variances. Thus, a classifier (a MLP) is trained using the vectors obtained from both speech frames and non-speech frames.

The performance of our proposed VAD technique when it is trained with noisy signals (*babble* noise and *white* noise) from different SNRs is better overall than the performance of the ITU-T standard systems G.720.1 and G.729b, since the classification error is considerably lower for non-speech and is comparable for speech segments. This makes our technique useful for both systems that require

low speech error rates and systems that require low non-speech error rates. Furthermore, our VAD seems to generalise the results properly for intermediate SNRs and the unseen noise types tested, which makes the system robust to different noise levels and types.

One of the greatest advantages of the MNS-based technique is that it performs on-line, making decisions frame by frame, with no need to analyse the neighbouring frames or the frames of a segment (or file) to which it belongs. In addition, the use of observation likelihoods as the basis of a VAD is also interesting due to its great simplicity. In a system where HMMs are used (as in an ASR system), the proposed VAD requires very little extra processing. The main disadvantage could be how the VAD behaves with unseen noises: it seems to be able to generalise results, but the error rate increases somewhat at some SNRs. Further research is needed to determine how the system could perform a proper generalisation.

A future research direction could be the analysis of the observation likelihoods obtained from a *speech* GMM (or several GMMs). It would be interesting to see whether their incorporation deteriorates or improves the results. Acoustic models of speech are more diverse than those for silence, so the research should include the analysis of the various patterns obtained for different speech phones or phone groups. Additionally, several (noisy) silence GMMs trained with different noisy signals could also be considered. Indeed, all the work introduced here was carried out based on a single GMM trained with clean signals. Obtaining more score vectors from different silence GMMs might provide more stable results.

Further research is also needed to analyse the generalisation of results when processing audio signals containing unseen noises. Indeed, the ability to generalise is one of the keys of the proposed VAD technique, since it does not contain any adaptive parameters which can help to adjust the system to different noise types and levels. The impact of including different types of noise and different combinations of them in the training data must be examined in depth to obtain a use that is as universal as possible.

Another possible research direction could be to test different classifiers in addition to MLPs. Recurrent Neural Networks (RNN) seem to be a good candidate since they can model sequential data with time dependences between feature vectors. This might add robustness to the proposed MNS-based VAD technique.

A challenging research direction would be to use our VAD technique in the field of acoustic event detection. It would be interesting to see how our proposed VAD behaves in scenarios where not only speech but noises of other kinds are presented and must be detected. An in-depth analysis would be required to identify what adaptations the VAD system would need.

Finally, in regard to the most practical aspect, the system needs to be tested in a real expert system. It needs to be implemented in a real-world application where an assessment must be carried out. This would give clues as to the real performance of our proposed VAD technique.

8. Acknowledgements

This work was partially supported by the EU (ERDF) under grant TEC2015-67163-C2-1-R (RESTORE) (MINECO/ERDF, EU) and by the Basque Government under grant KK-2017/00043 (BerbaOla). The authors would like to thank all the other members of the Aholab Signal Processing Laboratory for ongoing discussions and contributions to these topics.

References

Abdulaziz, A., & Kepuska, V. (2017). Noisy TIMIT speech (ldc2017s04). URL: `http://hdl.handle.net/11272/UFA9N`.

Alonso, J. B., Cabrera, J., Medina, M., & Travieso, C. M. (2015). New approach in quantification of emotional intensity from the speech signal: emotional temperature. *Expert Systems with Applications*, *42*, 9554–9564.

Benyassine, A. (1997). ITU-T Recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications. *IEEE Communications Magazine*, *35*, 64–73.

Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *27*, 113–120.

Chengalvarayan, R. (1999). Robust energy normalization using speech/nonspeech discriminator for german connected digit recognition. In *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)* (pp. 61–64).

Collobert, R., & Bengio, S. (). Links between perceptrons, MLPs and SVMs. In *International Conference on Machine Learning (ICML)* (pp. 23–30).

Delashmit, W., & Manry, M. (2005). Recent developments in multilayer perceptron neural networks. In *Proceedings of the 7th Annual Memphis Area Engineering and Science Conference, MAESC*. Memphis, USA.

Enqing, D., Guizhong, L., Yatong, Z., & Yu, C. (2002). Voice activity detection based on short-time energy and noise spectrum adaptation. In *Proc. of IEEE International Conference on Signal Processing (ICSP)* (p. 464467). IEEE.

Garner, P. N. (2011). Cepstral normalisation and the signal to noise ratio spectrum in automatic speech recognition. *Speech Communication*, *53*, 991–1001.

Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., & Pallett, D. (1993). DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon Technical Report N*, *93*.

Ghosh, P., Tsiartas, A., & Narayanan, S. (2011). Robust Voice Activity Detection using long-term signal variability. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *19*, 600–613.

Graf, S., Herbig, T., Buck, M., & Schmidt, G. (2015). Features for voice activity detection: a comparative analysis. *Journal on Audio, Speech and Music Processing (EURASIP)*, *2015*, 91.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*, 10–18.

Hautamäki, V., Tuononen, M., Niemi-Laitinen, T., & Fränti, P. (2007). Improving speaker verification by periodicity based Voice Activity Detection. In *Proc. of the International Conference on Speech and Computer (SPECOM)* (pp. 645–650).

Holmes, G., Donkin, A., & Witten, I. (1994). Weka: a machine learning workbench. In *Intelligent Information Systems,1994. Proceedings of the 1994 Second Australian and New Zealand Conference on* (pp. 357–361).

Huang, X., Acero, A., & Hon, H.-W. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. (1st ed.). Upper Saddle River, USA: Prentice Hall PTR.

Hughes, T., & Mierle, K. (2013). Recurrent neural networks for voice activity detection. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. 7378–7382).

(ITU), I. T. U. (2010). *Recommendation ITU-T G.720.1: Generic Sound Activity Detector (Series G: Transmission Systems and Media, Digital Systems and Networks: Digital Terminal Equipments - Coding of Voice and Audio Signals)*. Technical Report Telecommunication standardization sector of ITU (ITU-T). URL: `https://www.itu.int/rec/T-REC-G.720.1`.

(ITU), I. T. U. (2012). *Recommendation ITU-T G.729: Coding of speech at 8 kbit/s using conjugate structure algebraic-code-excited linear prediction (CS-ACELP) (Series G: Transmission Systems and Media, Digital Systems and Networks: Digital Terminal Equipments - Coding of Voice and Audio Signals)*.

Technical Report Telecommunication standardization sector of ITU (ITU-T). URL: `https://www.itu.int/rec/T-REC-G.729`.

Kingsbury, B., Saon, G., Mangu, L., Padmanabhan, M., & Sarikaya, R. (2002). Robust speech recognition in noisy environments: The 2001 ibm spine evaluation system. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. I–53–I–56).

Kostoulas, T., Mporas, I., Kocsis, O., Ganchev, T., Katsaounos, N., Santamaria, J. J., Jimenez-Murcia, S., Fernandez-Aranda, F., & Fakotakis, N. (2012). Affective speech interface in serious games for supporting therapy of mental disorders. *Expert Systems with Applications*, *39*, 11072–11079.

Kotnik, B., Sendorek, P., Astrov, S., Koc, T., Ciloglu, T., Fernández, L. D., Banga, E. R., Höge, H., & Kačič, Z. (2008). Evaluation of voice activity and voicing detection. In *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)* (pp. 1642–1645).

Kuan, T.-W., Tsai, H.-C., Wang, J.-F., Wang, J.-C., Chen, B.-W., & Lin, Z.-Y. (2012). A new hybrid and dynamic fusion of multiple experts for intelligent porch system. *Expert Systems with Applications*, *39*, 9288–9296.

Liu, F.-H., Stern, R., Acero, A., & Moreno, P. (1994). Environment normalization for robust speech recognition using direct cepstral comparison. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. II–61).

Liu, F.-H., Stern, R., Huang, X., & Acero, A. (1993). Efficient cepstral normalization for robust speech recognition. In *Proc. of ARPA workshop on Human Language Technology (HLT)* (pp. 69–74).

Ma, Y., & Nishihara, A. (2013). Efficient voice activity detection algorithm using long-term spectral flatness measure. *Journal on Audio, Speech and Music Processing (EURASIP)*, *2013*, 87.

Martínez-González, B., Pardo, J. M., Echeverry-Correa, J. D., & San-Segundo, R. (2017). Spatial features selection for unsupervised speaker segmentation and clustering. *Expert Systems with Applications*, *73*, 27–42.

Marzinzik, M., & Kollmeier, B. (2002). Speech pause detection for noise spectrum estimation by tracking power envelope dynamics. *IEEE Transactions on Speech and Audio Processing*, *10*, 109–118.

Mporas, I., Kocsis, O., Ganchev, T., & Fakotakis, N. (2010). Robust speech interaction in motorcycle environment. *Expert Systems with Applications*, *37*, 1827–1835.

Nemer, E., Goubran, R., & Mahmoud, S. (2001). Robust voice activity detection using higher-order statistics in the LPC residual domain. *IEEE Transactions on Speech and Audio Processing*, *9*, 217–231.

Obuchi, Y. (2016). Framewise speech-nonspeech classification by neural networks for voice activity detection with statistical noise suppression. In *ICASSP* (pp. 5715–5719).

Odriozola, I., Hernaez, I., Torres, M., Rodriguez-Fuentes, L., Penagarikano, M., & Navas, E. (2014). Basque speecon-like and basque speechdat mdb-600: speech databases for the development of asr technology for basque. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)* (pp. 2658–2665).

Pollak, P., & Sovka, P. (1995). Cepstral speech/pause detectors. In *IEEE Workshop on Nonlinear Signal and Image Processing* (pp. 388–391).

Principi, E., Squartini, S., Bonfigli, R., Ferroni, G., & Piazza, F. (2015). An integrated system for voice command recognition and emergency detection based on audio signals. *Expert Systems with Applications*, *42*, 5668–5683.

Rabiner, L., & Sambur, M. (1975). An algorithm for determining the endpoints of isolated utterances. *Bell Systems Technical Journal*, *54*, 297–315.

Ramirez, J., Yelamos, P., Gorriz, J., Segura, J., & Garcia, L. (2006a). Speech/non-speech discrimination combining advanced feature extraction and svm learning. In *Proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH)* (pp. 1662–1665).

Ramirez, J., Yelamos, P., Gorriz, J. M., & Segura, J. C. (2006b). SVM-based speech endpoint detection using contextual speech features. *Electronic Letters*, *42*, 426–428.

Sehgal, A., & Kehtarnavaz, N. (2018). A convolutional neural network smartphone app for real-time Voice Activity Detection. *IEEE Access*, *6*, 9017–9026.

Tan, Y. W., Liu, W. J., Jiang, W., & Zheng, H. (2014). Hybrid svm/hmm architectures for statistical model-based voice activity detection. In *International Joint Conference on Neural Networks* (pp. 2875–2878).

Tanyer, S., & Özer, H. (2000). Voice activity detection in nonstationary noise. *IEEE Transactions on Speech and Audio Processing*, *8*, 478–482.

Tatarinov, J., & Pollák, P. (2008). HMM and EHMM based voice activity detectors and design of testing platform for VAD classification. *Digital Technologies*, *1*, 1–4.

Thomas, S., Ganapathy, S., Saon, G., & Soltau, H. (2014). Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. 2519–2523).

Tirumala, S. S., Shahamiri, S. R., Garhwal, A. S., & Wang, R. (2017). Speaker identification features extraction methods: A systematic review. *Expert Systems with Applications*, *90*, 250–271.

Tucker, R. (1992). Voice activity detection using a periodicity measure. *IEE Proceedings, Part I: Communications, Speech and Vision*, *4*, 377–380.

Varela, Ó., Segundo, R. S., & Hernández, L. A. (2011). Combining pulse-based features for rejecting far-field speech in a HMM-based Voice Activity Detector. *Computers and Electrical Engineering*, *37*, 589–600.

Veisi, H., & Sameti, H. (2012). Hidden-Markov-model-based voice activity detector with high speech detection rate for speech enhancement. *IET Signal Processing*, *6*, 54–63.

Virtanen, T., Singh, R., & Raj, B. (2012). *Techniques for Noise Robustness in Automatic Speech Recognition*. (1st ed.). Wiley Publishing.

Westphal, M. (1997). The use of cepstral means in conversational speech recognition. In *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH)* (pp. 1143–1146).

Widrow, B., Winter, R., & Baxter, R. (1988). Layered neural nets for pattern recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *36*, 1109–1118.

Woo, K., Yang, T., Park, K., & Lee, C. (2000). Robust voice activity detection algorithm for estimating noise spectrum. *Electronic Letters*, *36*.

Zhang, X., & Wu, J. (2013). Deep belief networks based Voice Activity Detection. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *21*, 697–710.