Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 8079846, 15 pages
https://doi.org/10.1155/2017/8079846

Research Article

Evaluation of the Processing Times in Anuran Sound Classification

Amalia Luque,[1] Jesús Gómez-Bellido,[1] Alejandro Carrasco,[2] Enrique Personal,[2] and Carlos Leon[2]

[1]*Departamento de Ingeniería del Diseño, Escuela Politécnica Superior, Universidad de Sevilla, Seville, Spain*
[2]*Departamento de Tecnología Electrónica, Escuela Politécnica Superior, Universidad de Sevilla, Seville, Spain*

Correspondence should be addressed to Amalia Luque; amalialuque@us.es

Received 28 March 2017; Revised 18 May 2017; Accepted 22 June 2017; Published 27 July 2017

Academic Editor: Maximo Cobos

Nowadays, sound classification applications are becoming more common in the Wireless Acoustic Sensor Networks (WASN) scope. However, these architectures require special considerations, like looking for a balance between transmitted data and local processing. This article proposes an audio processing and classification scheme, focusing on WASN architectures. This article also analyzes in detail the time efficiency of the different stages involved (from acquisition to classification). This study provides useful information which makes it possible to choose the best tradeoff between processing time and classification result accuracy. This approach has been evaluated on a wide set of anurans songs registered in their own habitat. Among the conclusions of this work, there is an emphasis on the disparity in the classification and feature extraction and construction times for the different studied techniques, all of them notably depending on the overall feature number used.

1. Introduction

In the last few years, the number of devices focused on the monitoring and analysis of environmental parameters has grown strongly. However, sometimes the intended purpose is not related to the direct measurement of a parameter and requires the analysis of complex phenomena. An example of this is phenology, which consists of the study of periodic plant and animal life cycle and how some events are related to seasonal and climate variations [1].

Furthermore, reversing this study, it has been used for the prediction of climate evolution. A proof of this fact can be seen in some studies [2, 3] where the songs of some anuran species are proposed as an excellent indicator of climate change. However, these approaches are supported by a large number of audio recordings, which are usually collected in the field, and analyzed one by one later. Fortunately, the emergence of the Wireless Acoustic Sensor Networks (WASN) [4] has changed this approach. As an example, [5] proposes a WASN to distinguish between some anuran species (even between their different songs). For this, it extracts some MPEG-7 descriptor from audio frames and applies two simple classifiers (minimum distance and maximum likelihood) over them. Extending this study with a data mining approach, [6] increases the number of classifiers up to ten, using only frame features, without any temporal relationship between them. Furthermore, [7] proposes increasing the classification success rate, adding additional features which reflect the sequence of frames.

All of these studies are traditionally focused on comparing different techniques of audio processing, audio feature selection, or classification. However, a WASN approach requires contemplating more factors, such as execution times or the amount of transmitted information for each approach, which can seriously condition the applicability of each one.

In this sense, this paper proposes an audio processing and classification scheme, focusing on these kinds of architectures. Additionally, it is also completed with a detailed time analysis of the different processes involved in this proposed scheme (from acquisition to classification stages), providing

useful information to choose the best option with the best tradeoff between processing time and classification result accuracy.

Specifically, this paper is organized as follows: Section 2 shows an overview of the different processes that make up the proposed scheme. Section 3 briefly describes the WASN architecture for this scheme. Section 4 describes in detail the proposed audio processing scheme, explaining the different proposed approaches for each stage that comprises it. A reflection of the temporal implications of each one is raised in Section 5. Section 6 provides an extensive comparative study of the temporal requirements of each proposed approach, using a real problem (the classification of anurans species based on their song) as testbed. Finally, Section 7 sums up the conclusions.

2. Audio Process Architecture

The proposed architecture is focused in distributed solution where the audio analysis in the distributed nodes of a WASN is resolved. This network is made up by a mesh structure with dynamic routing (network topology is described later in Section 3). Thus, each network node is responsible for implementing its own audio processing, from audio acquisition to audio classification. In this sense, Figure 1 summarizes the proposed audio processing scheme, which is made by the following stages:

(1) *Sound Framing*. In this first stage, the audio signal is captured by local microphones. Each one samples the audio signal at 44.1 kHz, using a 16-bit codification. This sample rate was chosen, as will be seen later, because it is the most restrictive definition of the analyzed standards (this frequency could be set following the application requirements). This module also groups these samples in frames, which will be used as basic elements for analysis.

(2) *Feature Extraction*. It analyzes each frame separately, extracting D parameters from each one. For this extraction, two alternative approaches were proposed, based on the Multimedia Content Description Interface of MPEG-7 [8] standard or based on Mel Frequency Cepstral Coefficients (MFCCs) [9]. Both approaches will be described in detail on Section 4.1.

(3) *Feature Construction*. This stage uses the information of the previous stage. It can be considered a complementary feature extraction stage, adding information about frame evolutions (trends) or the order in which they appear (sequences). Three approaches have been proposed for this stage: no feature added; analysis of adjacent frame trends; and sequences modeling. These approaches will be described in detail on Section 4.2.

(4) *Frame Classification*. Each audio fragment (frame or sequence) is associated with one of the sound classes. This stage applies different classifiers, which have a different number of inputs, depending on previous stage choices. The proposed classification technique will be described in Section 4.3.

(5) *Sound Classification*. This final stage analyzes the partial results associated with each frame, choosing the most frequent class in the frame classification process as a global classification result.

3. Wireless Acoustic Sensor Network

The proposed WASN architecture is made up of a set of distributed nodes and a central node called base station (see Figure 2).

On the one hand, the base station is traditionally a standard PC, which has a radio adapter for the WASN connection. It acts as a gateway with other network technologies and provides centralized storage and processing capacities.

On the other hand, the distributed nodes are embedded systems, which have a wireless radio that allows them to connect with the other network elements (neighboring nodes and base station). Due to the remote location of the nodes, in a natural environment, they also require an alternative power source (i.e., solar systems), supported by batteries to guarantee their operation in adverse environmental conditions. This fact makes the consumption a critical constraint in these nodes, requiring drastic reductions in computational and radio power consumption. However, low power transceivers, such as ones based on IEEE 802.15.4 [10], have a limited coverage, precluding the communication between the base station and nodes. Due to this, mesh topologies are typical in these applications, routing the messages through neighboring nodes, and using protocols that support these structures (e.g., ZigBee [11] and 6LowPAN [12]). Additionally, another critical limit is the bandwidth restrictions. In traditional audio applications, each node often sends the raw data (441 samples per frame). However, this approach requires a lot of energy and can greatly overload the network. Against this, the proposed approach poses to send only the essential information, even reducing the payload up to a single data (the class to which the sound belongs).

Specifically, depending on user needs, different tradeoffs between the amount of transmitted information (radio consumption) and execution time (computational cost) can be established. In this sense, each network node must be able to locally characterize and classify sounds, where the lowest classification error is not the only objective. Furthermore, computational requirements of the each algorithm must also be considered for its viability over these kinds of platforms.

Due to this, in the next sections, the proposed scheme is detailed and completed later with a comprehensive analysis of their execution performance in each audio classification stages.

4. Feature Extraction and Classification

As it was introduced above, the audio features extraction is done frame by frame, obtaining several parameters for each one. Later, based on these first direct features, this information set is completed with second features construction stage, where new additional estimated information is provided.

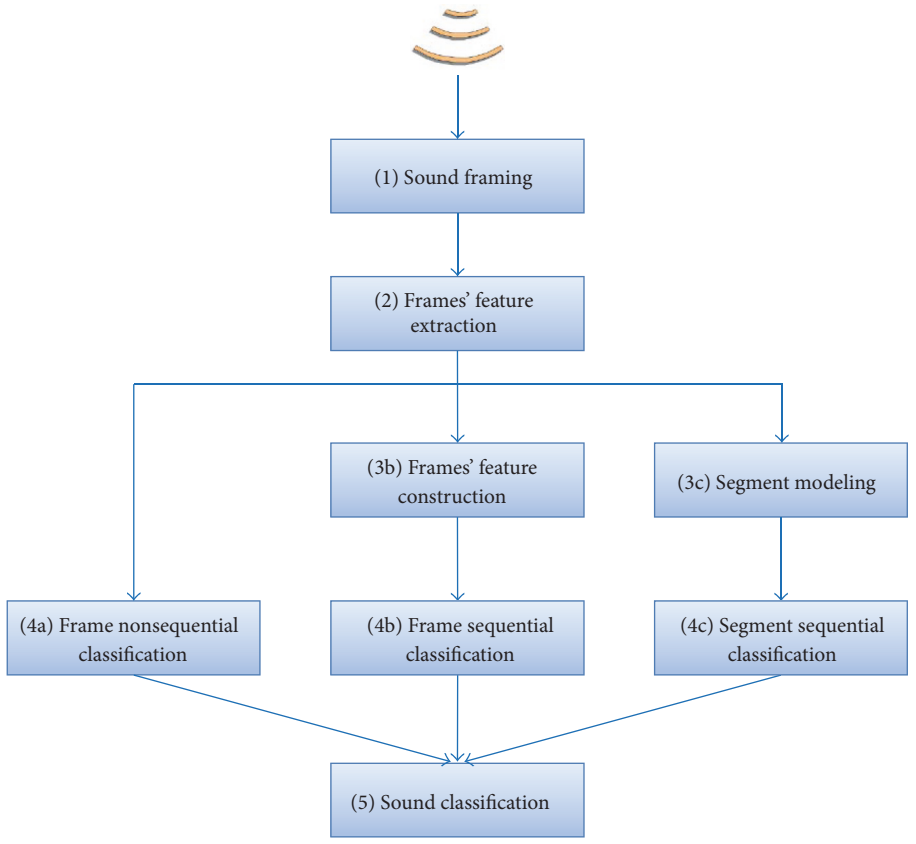

Figure 1: Audio processing scheme.

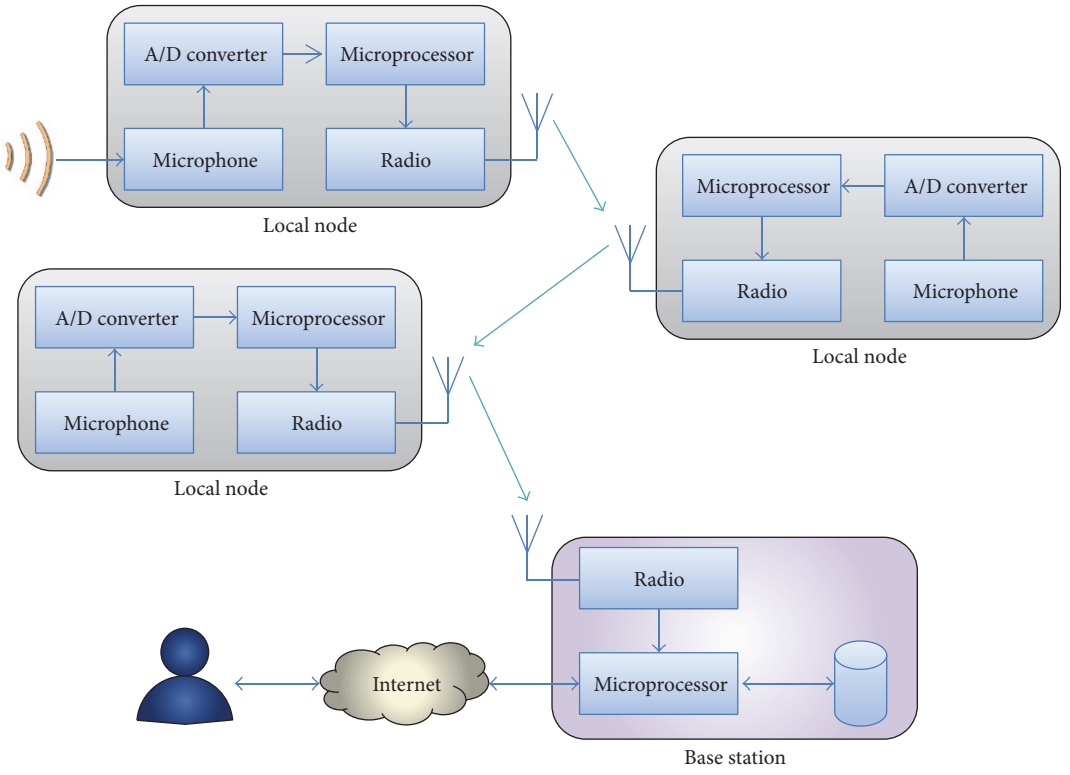

Figure 2: WASN architecture.

Table 1: MPEG-7 features and their origin analysis.

Feature	Symbol	Based on
Total power	P_t	Spectrogram analysis
Relevant power (power in a certain frequency band)	P_r	
Power centroid	C_p	
Spectral dispersion	D_s	
Spectrum flatness	Fl	
Frequency of the formants (×3) (the three first formants are considered)	FF_n	Linear prediction coding (LPC) analysis
Bandwidth of the formants (×3) (the three first formants are considered)	FB_n	
Pitch	Pi	
Harmonic centroid	C_H	
Harmonic spectral deviation	D_H	
Harmonic spectral spread	S_H	
Harmonic spectral variation	V_H	
Harmonicity ratio	R_H	Harmonicity analysis
Upper limit of harmonicity	F_H	

Both analyses are detailed in the two next subsections, while the classification stage is analyzed in the third.

4.1. Frame Feature Extraction. In this work, the feature extraction of a frame has two approaches. On the one hand, the first proposed approach consists of extracting the features defined by MPEG-7 standard. This standard defines a sample rate of 44.1 kHz and recommends a *hopSize* of 10 ms. Both constraints define the frame size for this application, involving a set of 441 samples. To characterize this information, up to 18 parameters have been defined ($D = 18$, see Table 1), which are derived from three kinds of base analysis:

(i) *Spectrogram Analysis* $S(f)$. It uses the Fast Fourier Transform (FFT) to identify the frequency components of a frame. From this analysis, up to 5 MPEG-7 features are defined (see Table 1).

(ii) *Linear Prediction Coding (LPC) Analysis*. It poses that a sound $s(n)$ can be calculated as a linear combination of past samples and an error signal. LPC analysis models a sound source using a harmonic generator, a noise generator, and a digital filter (which characterizes the vocal tract). The characteristic polynomial roots of this filter are complex, play a key role in this technique, and determine the different formants (resonant notes) in the audio samples. The formants are defined by its frequencies (f_i) and bandwidths (B_i). From this LPC analysis, up to 11 MPEG-7 parameters are defined (see Table 1).

(iii) *Harmonicity Analysis*. It represents the degree of acoustic periodicity and is based on an autocorrelation analysis $\rho(k)$ of the audio samples $s(n)$. From this analysis, up to 2 MPEG-7 features are defined (see Table 1).

For more details, MPEG-7 standard [5, 8] widely describes the definitions and extraction techniques of these features.

On the other hand, other alternatives propose an MFCC analysis for the feature extraction. MFCCs are based on the sound cepstral through its homomorphic processing [13]. Thus, this analysis is a widely extended method for audio features extraction (i.e., for speech recognition). However, MFCCs have the disadvantage that they do not have any general purpose standardized method, although, for telephony applications, the ETSI standard [14] defines an extended procedure to obtain these coefficients. However, this approach requires some tuning to make it comparable with the first feature extraction alternative described above. Specifically, this modification is related to the frame size. ETSI standard proposes a frame length of 25 ms for a sample rate of 16 kHz, obtaining 400 samples per frame. In our case, the sample rate chosen for this work (44.1 kHz) is not defined in this standard and, keeping the frame length, the number of samples per frame increases above a thousand. So, resembling the MPEG-7 approach, a frame size of 10 ms has been proposed, which leads to a number of samples per frame (441) quite similar to the ETSI standard recommendation (400). Furthermore, according to this approach, the number of MFCCs to represent a frame is 13 ($D = 13$).

4.2. Frame Feature Construction. In previous section, D direct features were extracted from each frame. However, these features do not consider the intrinsic sequential characteristic of the sound temporal evolution. So, this sequential information should be added constructing some more new features. Three constructing feature approaches have been considered: no new feature being added (for comparison purposes); trend analysis of adjacent frames; and sequences (groups of N frames) modeling.

4.2.1. No Feature Construction. This approach represents "a" or left branch in Figure 1 and consists directly in not deriving any additional information, considering the direct frame information enough for the next classification stage.

4.2.2. Feature Construction Using Frame-Trend Analysis. This approach represents "b" or the center branch in Figure 1 and consists of combining extracted information of the frame under analysis with the extracted features of its neighbors, obtaining C, new features for the next stage. Specifically, up to three alternatives are proposed as follows:

(a) Regional Dispersion (RegDis) [15]. This approach consists of using an analysis sequence of N frames (composed by the frame under analysis and its adjacent ones), each one being characterized by its D extracted features. The general idea for this feature construction technique is to use the time axis to construct new temporal axis based features. Commonly, these techniques are based on the frame feature's values without considering their order, which is usually called a bag of features. Average values or some other related statistics are usually employed. In our case some of the anuran calls to be classified show the typical croaking of a frog while others

are similar to a whistle. The croaking sound is produced by repeatedly opening and closing the vocal cords (roughly, every 10 msec., the frame length) leading to a sequence of frames featured with highly spread values. On the other hand, the whistle-like sounds is produced by a continuous air flow showing low spread in feature values. So, to incorporate this information in the classification process a new set of features is constructed considering not the average but the spread of the extracted feature values. And to avoid the influence of outliers, the interquartile range instead of the standard deviation is selected. In the implementation used in this paper, first for every frame, a "window" centered in that frame is considered, using the closest neighbor frames. And for every original parameter, a new derived parameter is constructed. For this purpose the values of the original parameter for every frame in the window are considered. The interquartile range of these values (the difference between 75th and 25th percentiles) is computed, and this value is considered the new derived parameter. In this way, the number of constructed features is $C = D$, so up to $2\times D$ parameters (a vector in $\mathbb{R}^{2\times D}$) are now identifying a frame, where C of them include some kind of sequential information. In this approach a 10-frame window size (100 msec.) has been used.

(b) Δ Parameters. This second approach characterizes the trend (ascending or descending) that follows a frame feature sequence. It is in some sense the derivative of each extracted feature, following the expression of [16]. In this sense, for each frame, one trend feature per each extracted one is constructed ($C = D$). Additionally, this procedure can also be extended to second-order derivative (Δ^2 parameters) or even higher. The total number of features after applying this technique will be $D + C = 2 \times D$ (in case of using Δ parameters) or $3 \times D$ ($C = 2 \times D$, in case of using Δ and Δ^2 parameters).

(c) Sliding Windows (SW) [17]. This last trend-analysis approach proposes the use of a short window made up of a sequence of w adjacent frames, centered in the frame under analysis. In this approach, the constructed features are the set of the D extracted features for every frame under the window. Therefore, in this method, the total number of features for a frame will be $w \times D$.

4.2.3. Feature Construction Using Sequence-Based Modeling. The last alternative is represented by "c" or right branch in Figure 1. It consists of using techniques which directly analyze sets of frames (or audio segments). Specifically, for this paper, two approaches have been studied.

(a) Autoregressive Integrated Moving-Average (ARIMA) Models [18]. This method starts from a $\mathbf{X}$ matrix, which characterizes an audio sequence with N frames, transforming it into a vector $\mathbf{A}$, which is made up by coefficient matrices ($\mathbf{X} \rightarrow \mathbf{A}$). The matrix $\mathbf{X}$ has a dimension of $N\times D$, containing N vectors of parameter ($\mathbf{X_i} \in \mathbb{R}^D$) associated with each frame of the audio segment. Thus, to obtain the vector $\mathbf{A}$, it considers that the sequence of frame features ($\mathbf{X_i}$) is the result of a Vector ARIMA temporal series, VARIMA (p, d, q). It is defined by (1), in which p is the order of the autoregressive model, d is the degree of differencing, and q is the order of the moving-average model:

$$\mathbf{X_i^{(d)}} = \mathbf{C_0} + \sum_{k=1}^{p} \mathbf{A_k} \mathbf{X_{i-k}^{(d)}} + \sum_{k=1}^{q} \mathbf{B_k} \boldsymbol{\varepsilon}_{\mathbf{i-k}} + \boldsymbol{\varepsilon}_{\mathbf{i}}. \quad (1)$$

$\mathbf{A_k}$ and $\mathbf{B_k}$ are two coefficient matrices, which have a $D\times D$ dimension. $\mathbf{C_0}$ is a vector, which represents the average vector time series and has D components. Usually, this time series is normalized, so that $\mathbf{C_0}$ vector has a null mean and it is being typically omitted. Due to this, the parameter number to characterize a sound segment is $(p + q) \times D^2$. Additionally, it is also typical to assume that the time series is stationary ($d = 0$), and VARMA models can be approximated by equivalent VAR models ($q = 0$). Therefore, using the Akaike Information Criterion (AIC) [19], it is possible to find an optimal value of the model order (p) and $\mathbf{A_k}$ matrix using a maximum likelihood technique [20].

In this sense, this method provides $p \times D^2$ features to characterize each sound segment, which will be used by nonsequential classifiers on the next stage.

(b) Hidden Markov Models (HMM) [21]. Firstly, a HMM takes the D extracted features of each frame ($\mathbf{X_i} \in \mathbb{R}^D$) of the segment, quantizing them [22] and obtaining an observation O_i, which is defined by the integer code c_k in the $[0, C - 1]$ range. An HMM has several connected states (defined by $\mathbf{S}$), which produce an observation sequence. For isolated "words" (anuran calls) recognition, with a distinct HMM designed for each class, a left-right model is the most appropriate, and the number of states should roughly correspond to the number of sounds (phonemes) within the call. However, differences in error rate for values of N close to 5 are small. The structure and the value of N have been taken from [21]. The S_a state generates the c_k code with a E_{ak} probability and evolves to S_b with a T_{ab} probability. $\mathbf{E}$ and $\mathbf{T}$ matrices of each class θ are obtained by the pattern frames of each class ($\boldsymbol{\Pi_\theta}$), using a forward-backward algorithm [23]. Once the parameters of an HMM are estimated (following structure proposed in Figure 3), this algorithm takes the observation sequence of a sound segment (formed by N frames), which is characterized by its $N \times D$ features and computes the probability that the sequence had been generated by the HMM of each class. Finally, the segment is labeled as belonging to the sound class with the highest probability from the above computation.

4.3. Feature and Sound Classification. Once the different alternatives of frame featuring have been analyzed, the next step is using these features to identify the class to which they belong (step (4) of all branches in Figure 1). Except for classifiers such as HMM, which intrinsically consider the sequential character of the sound, the remaining classification procedures proposed in this paper have a nonsequential philosophy. That is, they require increasing their input set with some additional constructed features to acquire the sequential information (using the methods explained in Section 4.2 or by building ARIMA models). All of the classifiers that will be considered perform a supervised classification. That is, they compare the constructed features of a sequence to sound

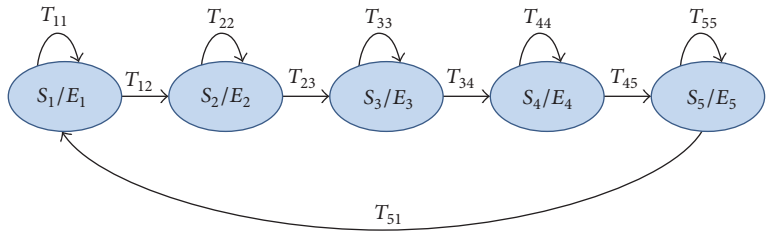

Figure 3: HMM structure.

patterns of known classes and identify the class to which it belongs. Specifically, as a representative example of these kinds of techniques, several classifiers have been studied in this paper: minimum distance (MinDis) [24], maximum likelihood (MaxLik) [25], decision trees (DecTr) [26], k-nearest neighbors (kNN) [27], support vector machine (SVM) [28], logistic regression (LogReg) [29], neural networks (Neur) [30], discriminant function (Discr) [31], and Bayesian classifiers (Bayes) [32]. This set represents general purpose classifiers which are well suited for this kind of applications [6, 33].

In the final stage, (5) of Figure 1, once all frames of a sound have been classified, the most repeated class in them is finally assigned as the global classification for the audio file.

5. Considerations about Classification Times

In previous sections, different implementations or alternatives for animal sound analysis have been proposed. However, from an implementation point of view, these algorithms are not trivial and may require a lot of execution time.

In this sense, an exhaustive time analysis of each stage is essential to guarantee the real-time application. Specifically, and according to the previous section, the analysis time can be divided into five stages: audio acquisition, frame feature extraction (direct frame analysis), frame or sequence feature construction (frame set or sequence analysis), feature classification of each frame, and finally the global sound classification. However, for some of them, their processing times are not static. Specifically, as was described in previous sections, an animal sound can be characterized by a set of P features (or by a point in the $\mathbb{R}^P$ space). Therefore, as will be seen in the next sections, this space dimension (or feature number) is a keystone in processing time studies, affecting the following ways:

(i) The features extraction time of each frame grows when the number of these parameters increases.

(ii) The features construction time of additional information for each frame (or sequence) grows when the number of direct or additional parameters increases.

(iii) The classification time of each frame (or sequence) increases with its features dependency.

(iv) As it will be addressed in Section 6.4 (see Figure 22), the classifier generation time increases with the number of features for most algorithms, some of these growths being very intense (between one or two orders of magnitude).

Considering the three first times in the former list, their sum is an important restriction in real-time audio processing applications, where this total time must always be less than the audio fragment duration. In this sense, this constraint makes an exhaustive comparative time study of all proposed alternatives essential, seeking the best tradeoff between the feature number and the time available.

Moreover, although not directly related to real-time applications, the time needed to obtain the classifiers is also related to the feature space dimension. Due to this, a comparative analysis of this time could also be useful, especially in applications with a dynamic knowledge base in which the training process is repeated periodically.

From all of the above, these times have been studied in the next section in detail. This analysis makes the comparison between the different proposed alternatives possible, identifying the least computationally demanding.

6. Results and Discussions

As a testbed of the previously described strategy, 63 sound files provided by the Zoological Sound Library [34] were used. Specifically, these files correspond to two anuran species; the *Epidalea calamita* (natterjack toad) and the *Alytes obstetricans* (common midwife toad), with a total of 605,300 frames, every one of 10 ms. length, that is, a total of 6,053 seconds. These audio files have a total duration of 1 h:40′:53″, an average duration per file of 96 seconds (1′:36″) and a median duration of 53 seconds. This is a large dataset as the total number of observations which has to be classified is 605,300 (most of the algorithms considered in this paper are frame-based classifiers). For training purposes, a small portion of these frames (13,903), properly selected and labeled by biologists, was used as sound patterns (see detailed summary in Table 2).

Furthermore, a common characteristic to all of these sounds is that they were recorded in a natural habitat with a significant presence of noise (wind, water, rain, traffic, voices, etc.), which poses an additional challenge in the classification process.

Although the whole process was designed to be finally implemented in distributed nodes, this study was implemented over a laboratory prototype, equipped with an Intel® Core™ i7-4770 processor at 3.4 GHz and 8 GB of RAM. All

Table 2: Testbed audio details.

Sound class	Sound		Patterns		
	Files	Seconds	Files	Seconds	Frames
Epidalea calamita (mating call)	23	2,576	2	21	1,439 (10.35%)
Epidalea calamita (release call)	10	415	1	29	248 (1.78%)
Alytes obstetricans	30	3,062	2	89	375 (2.70%)
Silence/noise	—	—	—	—	11,841 (85.17%)
Total	63	6,053	5	139	13,903 (100%)

the algorithms have been coded in MATLAB® with an implementation that does not explicitly exploit code parallelism over the different cores. However, the MATLAB by default built-in multithreading computation has been exploited.

The next sections show and discuss processing time results related to the classification of these sounds.

6.1. Frame Feature Extraction Time. As it was mentioned in Section 4.1, obtaining the MPEG-7 features of a single frame requires applying three basic techniques; spectrogram, LPC, and harmonicity analysis. Later on, a specific derivation is also necessary for each feature. Table 3 summarizes all of these times where it can be seen that, for instance, obtaining the power centroid (C_p) requires computing a spectrogram (primary process) and performing an additional specific center-of-mass calculation (or secondary process). Obviously, to obtain other features based on the same primary process, only adding the time of its secondary process is required. This fact can condition the feature selection, the feature type (primary process dependency) being more important than the number of them within it.

On the other hand, the MFCC features use a single process, being calculated all at once (see Table 3).

In summary, the extraction time for the full MPEG-7 feature set is 3.2 ms (approx. 1/3 of frame duration). MFCC feature set requires 45 μs, a time significantly lower than the previous one (and lower than the duration of the frame). MFCCs are calculated simultaneously, and they use an algorithm based on a spectrogram analysis (due to this, its time is similar to the MPEG-7 spectrogram process).

In this sense, a reduction in MPEG-7 feature dimensionality (reduction in the number of features extracted) will improve this time of frame features extraction. However, as discussed above, this time is strongly conditioned by the parameter type (or their primary process needs), obtaining a significant reduction when any of them is not necessary. Conversely, a reduction in MFCC feature dimensionality does not involve any reduction in this time, since all are obtained simultaneously.

6.2. Frame Feature Construction Time. Following the techniques described in Section 4.2, the construction of additional features extends the information associated with each frame. In this sense, Table 4 shows, for every feature construction technique (first column), the time spent (4th column) and the accuracy obtained (5th column) when these constructed features are used with the best of the classifiers considered in Section 4.3 (6th column). As it can be seen, all the times, with the exception of the ARIMA method, show very small values (below 1% of the total frame duration).

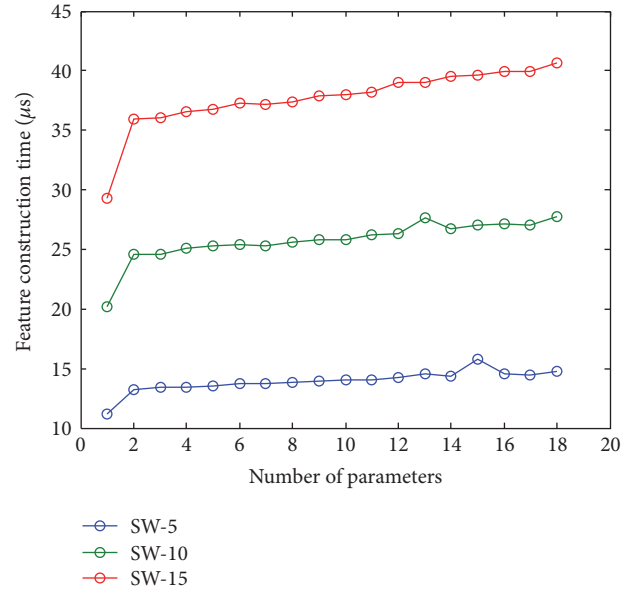

Figure 4: Sliding window behavior for different number of features.

However, the calculation times of these parameters present a significant dependence on the number of parameters. Figure 4 shows the relationship between SW construction time and the number of features (for different window sizes).

In this figure, it is easy to note that the construction time shows an approximately linear behavior. Moreover, this time also has a lineal dependence on the window size (as it can be clearly seen in Figure 5). Similar behavior was obtained for MFCCs.

In HMM technique, for each sequence, the feature construction consists of converting the original parameter vector ($\mathbf{X_i}$) into a scalar observation (O_i) through a quantization process. In this sense, the HMM processing time also significantly depends on the number of features. Figure 6 shows this dependency for the case of MPEG-7 parameters. As it can be seen in this figure, HMM construction time is defined by a piecewise function, approximately linear to steps between sections.

Moreover, ARIMA analysis consists of converting the original parameter matrix ($\mathbf{X} \in \mathbb{R}^{N\times D}$) into a vector of coefficient matrices ($\mathbf{A} \in \mathbb{R}^{p\times D^2}$), modeling its time series. This technique characterizes an audio sequence (or a frame set). Due to this, for an adequate comparison with other techniques, the ARIMA sequence feature constructing times have been normalized to its equivalent frame times (dividing by the number of frames in the sequence).

Like other techniques, this time also significantly depends on the number of features. Figure 7 depicts this time

Table 3: Time analysis of the frame feature extraction.

Parameter type	Requirement	Feature	Processing time	
			Secondary (μs)	Total (μs)
MPEG-7 (17)	Spectrogram, primary processing time 41.33 μs	P_t	2.48	43.80
		P_r	20.23	61.55
		C_p	9.42	50.75
		D_s	14.01	55.33
		Fl	52.22	93.55
	LPC, primary processing time 1,777.92 μs	FF_n	0.00	1,777.92
		FB_n	0.00	1,777.92
		Pi	0.00	1,777.92
		C_H	5.86	1,783.78
		D_H	8.75	1,786.67
		S_H	1.87	1,779.79
		V_H	2.78	1,780.70
	Harmonicity, primary processing time 1,262.02 μs	R_H	0.00	1,262.02
		F_H	0.00	1,262.02
MFCC (13)			44.29	44.29

Table 4: Time analysis of the feature construction process.

Feature constr.	Feature type	Number of features	Processing time (μs)	Accuracy	Best clas.
RegDis	MFCC	13	85.74	92.59%	Bayes
	MPEG-7	18	99.60	91.53%	DecTr
Δ	MFCC	13	0.388	94.71%	Bayes
$\Delta + \Delta^2$	MFCC	13	0.652	94.71%	Bayes
SW (5 frames)	MFCC	13	10.62	94.71%	Bayes
	MPEG-7	18	14.72	91.53%	DecTr
HMM	MPEG-7	18	84.39	84.13%	—
ARIMA (3,0,0)	MPEG-7	18	25,613.0	70.37%	Bayes

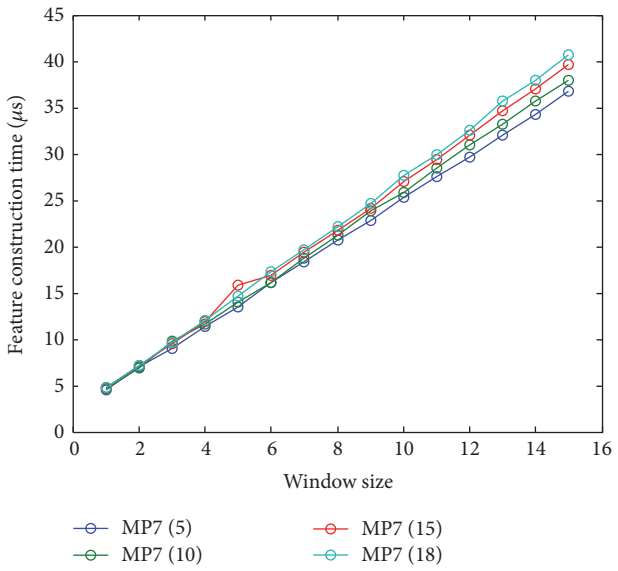

Figure 5: Sliding window behavior for different window size.

dependency when MPEG-7 features are used, showing an exponential increase when the number of features characterizing each frame (D) also increases.

6.3. Frame (or Sequence) Classification Time. Once the feature extraction and construction processes are analyzed, the next step must be to analyze the classification procedure based on these features.

In a first stage, only extracted (or nonsequential) features will be considered for classification purposes ((4a) or left branch approach in Figure 1). As an example, Figure 8 shows the time spent by the decision tree (the best classifier among the proposed ones) to classify sounds of different duration (or different number of frames), using the complete MPEG-7 feature set. This classification time follows a clear linear behavior (red line), this trend being similar to behavior obtained for the every proposed classifier. Therefore, it is possible to assume that the sound classification time is approximately proportional to the number of its frames, or, in other words, that the classification time per frame is approximately constant for the different proposed classifiers.

In this sense, Table 5 shows a summary of this time analysis. Additionally, it also shows the classification time relative to the standard frame length (10 ms), the relative classification speed (the number of frames classified in a frame length),

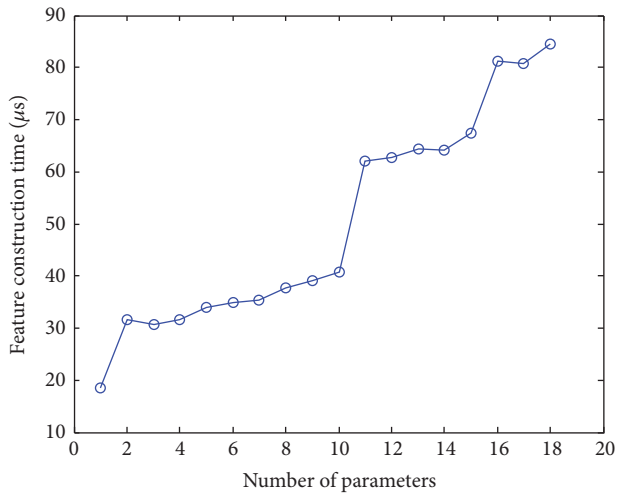

Figure 6: HMM behavior for different number of features.

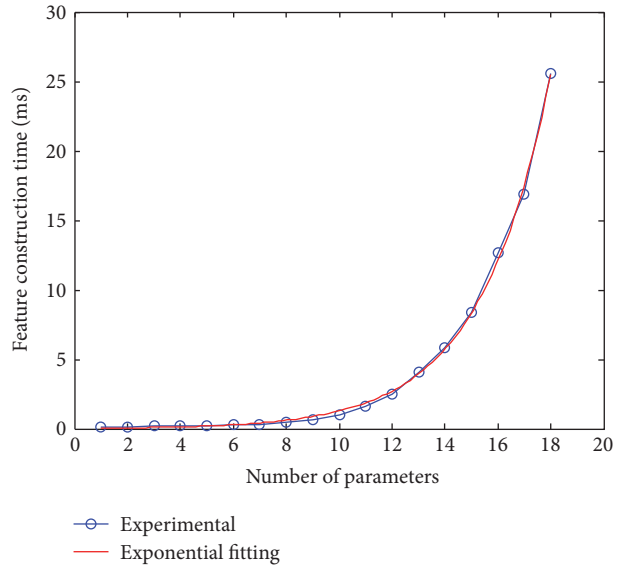

Figure 7: ARIMA behavior for different number of features.

and the accuracy of the classification process when 18 MPEG-7 features are used. A more detailed explanation of the classification performance can be found in [6].

Obviously, for real-time audio processing, this relative time must be less than 100% (or in the same words, the relative speed must be greater than 1). As it has been previously shown, all the algorithms fulfill these conditions, however two of them being significantly slower: maximum likelihood and k-nearest neighbors. This information is also depicted in Figure 9 (using a logarithmic scale), which also indicates the upper time limit (using a dashed red line) if a real-time analysis is required.

However, the classification time per frame directly depends on the number of features used (or input parameters). In this sense, Figure 10 shows this dependence, where its vertical axis is normalized by the maximum classification time of each technique (obtained for 18 features).

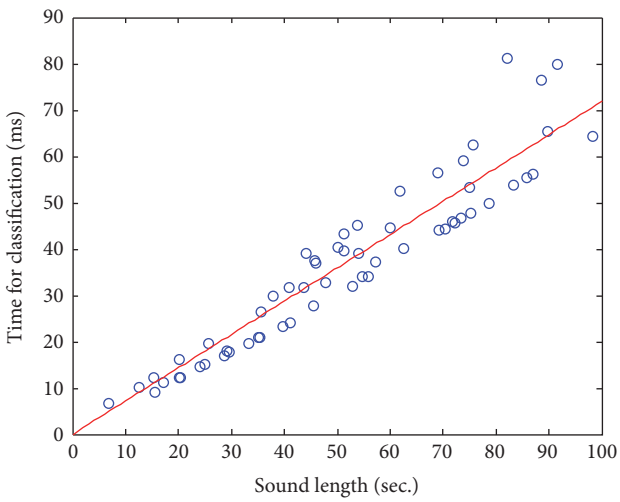

Figure 8: Decision tree classification time versus sound duration.

Table 5: Time analysis of the classification stage.

Classifier	Classification time (μs)	Normalized time (%)	Speed (classif. per frame)	Accuracy (MPEG-7)
MinDis	15	0.15%	690	58.73%
MaxLik	1175	11.75%	9	86.24%
DecTr	7	0.07%	1389	91.53%
kNN	207	2.07%	48	82.01%
SVM	27	0.27%	372	82.01%
LogReg	7	0.07%	1515	76.72%
Neur	8	0.08%	1333	75.66%
Discr	8	0.08%	1299	77.78%
Bayes	7	0.07%	1449	80.95%

In general, it is possible to identify an upward trend in the classification time with the number of features for most algorithms. Figure 11 shows the linear regression line for this dependence. It reflects a moderate 20% increase when the number of features increases from its minimum to its maximum value.

Another issue which has to be addressed is the effect of the number of classes on the processing times. It has no influence on feature extraction and feature construction times as these processes precede (and independent of) the definition of classes. However, the number of classes does potentially have influence on classification times. To explore this topic the original dataset has been modified introducing additional classes (anuran species or sounds) and labeling every frame with a uniformly distributed random class (silence/noise is considered as a class). It has to be underlined that they are not real classes and their only purpose is for testing the impact of the number and distribution of classes on processing times. Figure 12 shows the results obtained for every algorithm and its linear regression (dashed red line in the figure). For most of the classifiers, there is a moderate increase when the number of classes increases.

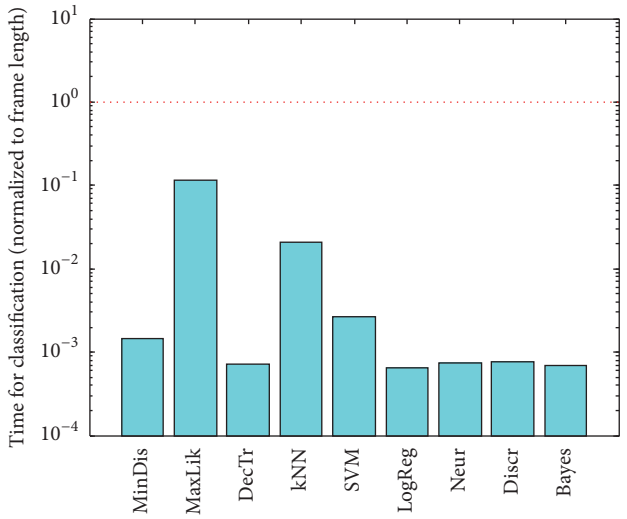

FIGURE 9: Relative classification time per frame using $D = 18$ extracted features.

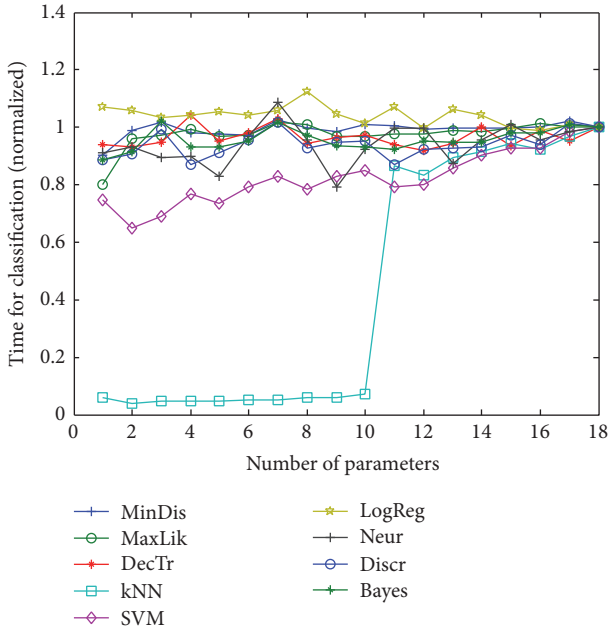

FIGURE 10: Normalized classification time for different number of features.

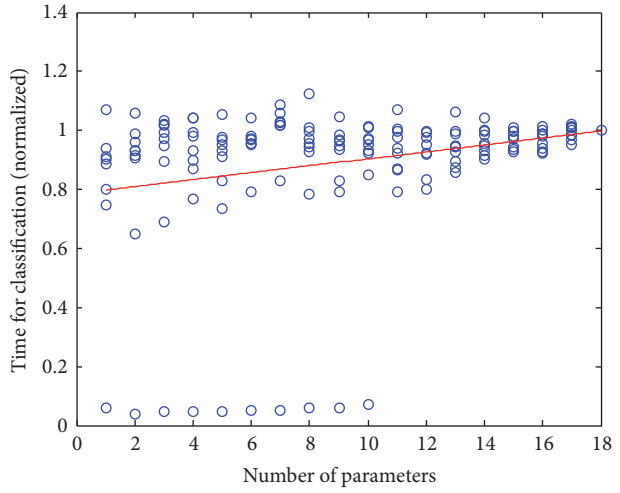

FIGURE 11: Linear regression of classification time (results of all classifiers).

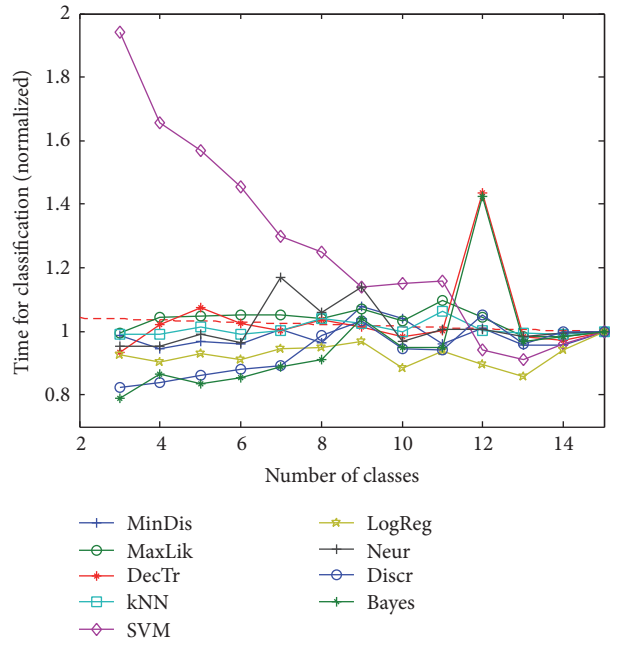

FIGURE 12: Normalized classification time for different number of classes and its linear regression (dashed red line).

Following the scheme proposed in this paper, the next step considers the sound sequential information using frame-trend features (branch (4b) of Figure 1). This classification extracted and constructed features are combined, using them as input of the nonsequential classifiers being described above. Specifically, frame trends are extracted using regional dispersion, Δ parameters, or sliding window. However, as it was seen above, these construction techniques can significantly increase the total number of features. In this sense, sliding window is the most restrictive (worst) case which, using a window with w frames, determines the use of $w \times D$ features in the classifier.

Figure 13 shows the normalized classification time (for a 10 ms frame length) corresponding to each analyzed algorithm, when the full set of MPEG-7 features (18) and the SW with a window size of 10 is used.

As in the previous analysis, all studied classifiers fulfill the time constraints to operate in real-time mode, nevertheless the maximum being the likelihood and k-NN results close to the feasibility limit.

Obviously, this time also depends on the number of features, which directly depends on the configuration of the construction method (i.e., window size for SW). Figure 14 shows the classification time as a function of the number of parameters when a feature construction method is used. In

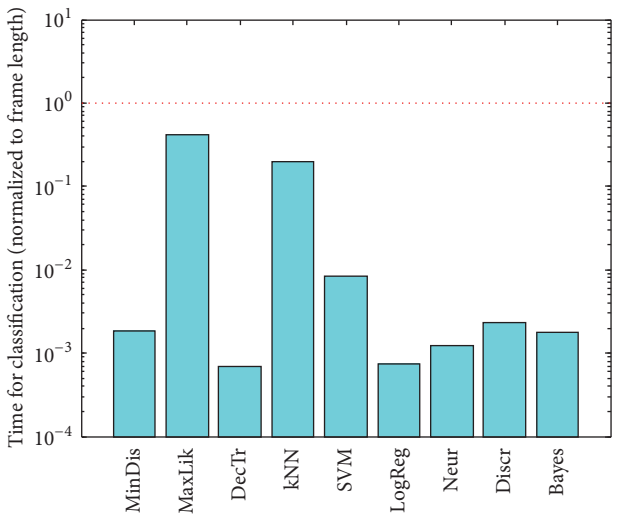

Figure 13: Relative classification time per frame using sliding windows.

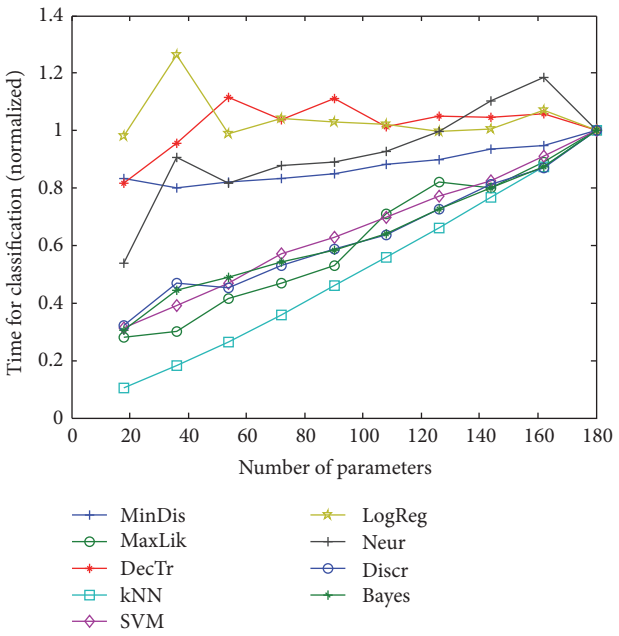

Figure 14: Normalized classification time for different number of features (extended to constructed features).

this figure, a SW with a window size of 10 frames has been used and the time values (vertical axis) have been normalized by the maximum feature dimension (10×18).

In this sense, it is easy to note that there exists a global increasing behavior in the classification time depending on the number of feature. This trend is clearly shown in Figure 15 where the linear regression of global data (all classifiers) has been represented. Thus, it is possible to observe that the global difference between 18 features (window size of 1, not adding any trend-frame information) and 180 features (window size of 10, maximum studied dimensionality) shows a remarkable increase of approximately 50%.

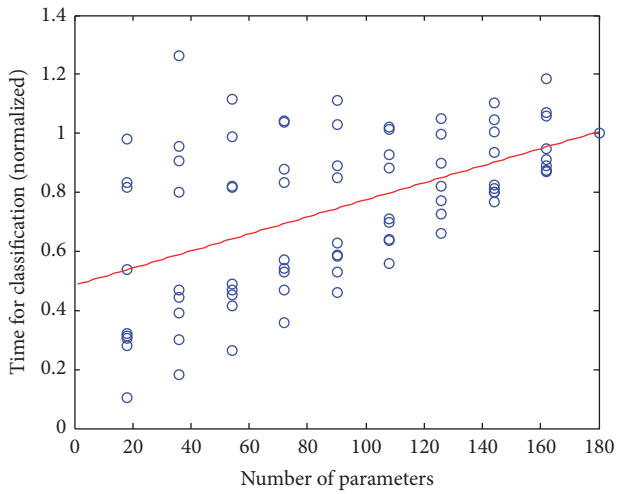

Figure 15: Linear regression of classification time (results of all classifiers extended to constructed features).

Table 6: HMM classification time for an audio segment.

Classifier	Classification time	Classification speed	Accuracy (MPEG-7)
HMM	12.56 ms/s (1.26%)	80	84.13%

To finish the study of the classification time, the last topic to be addressed is the sound segment (frame or sequence) classification ((4c) or right branch of Figure 1). Specifically, in this approach, two techniques have been evaluated: HMMs and ARIMA models.

Figure 16 shows the HMM classification time as a function of the sound duration, when the minimum or the maximum number of MPEG-7 features are used. In this figure, it is easy to identify a clear linear trend where the classification time increases with the sound duration. So, it can be concluded that the unitary classification time is approximately constant, and it increases with the number of features. Specifically, Table 6 shows the HMM classification time and classification speed for the 18 MPEG-7 features set. As the unitary classification time is less than 100%, it is possible to claim that this algorithm is suitable for real-time processing. Figure 17 reflects the dependency of this classification time with respect to the number of features.

Conversely, ARIMA approach (as it was described above) uses the same classifiers as those applied for frame classification, now usually increasing the size of the feature set. Therefore, the classification time is the same as that has already been analyzed above and is reflected in Figure 14. The best result corresponds to the Bayesian classifier with an accuracy of 70.37%.

Finally, the last step is the classification of the full sound file (process (5) of Figure 1). But this classification is just a simple count of frame or segment classes, where the sound file is labeled as belonging to the most frequent frame class. Thus, its processing time (approx. 10 ns) is negligible in comparison

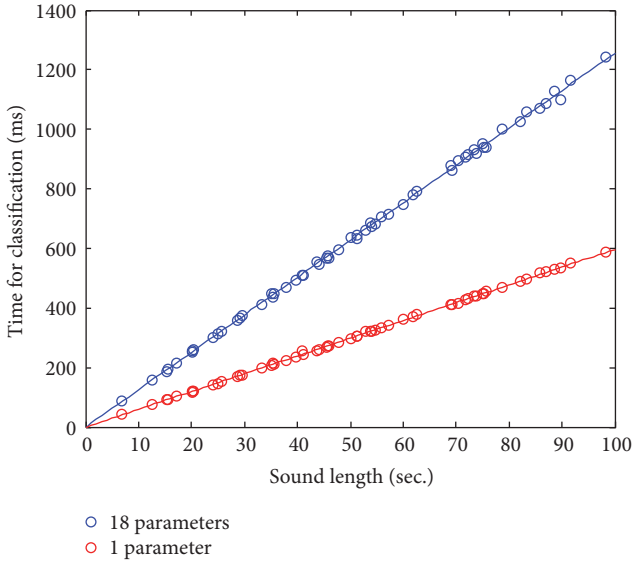

Figure 16: HMM classification time for different durations.

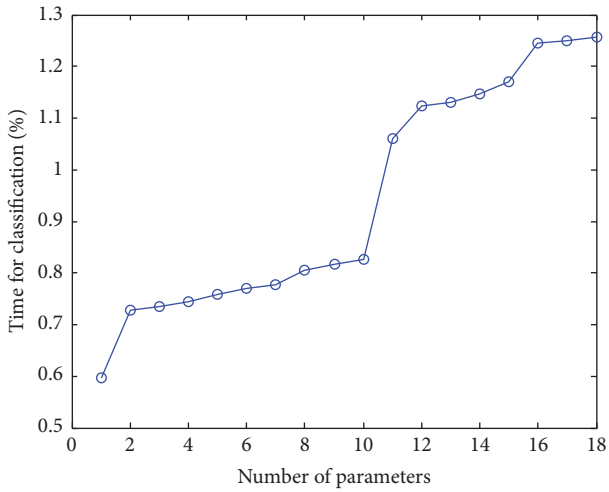

Figure 17: HMM classification time for different number of features.

with previously analyzed classification processes, so it will be ignored in this analysis.

6.4. Classifier Generation Time. After studying temporal requirements of the three proposed stages for the audio fragment classification, the next study will be the time required to obtain each classifier. Obviously, this time is less critical than those previously studied, since this stage is not properly concerned in the real-time classification process. However, this study would be of interest in cases where the knowledge base is dynamic, or it has a periodic or iterative training approach. In addition, it is true that the techniques proposed (based on supervised classification approach) may have significant deviations in the training period (depending on the training data; the number of patterns; or their content). Nevertheless, its results can be taken as a starting point

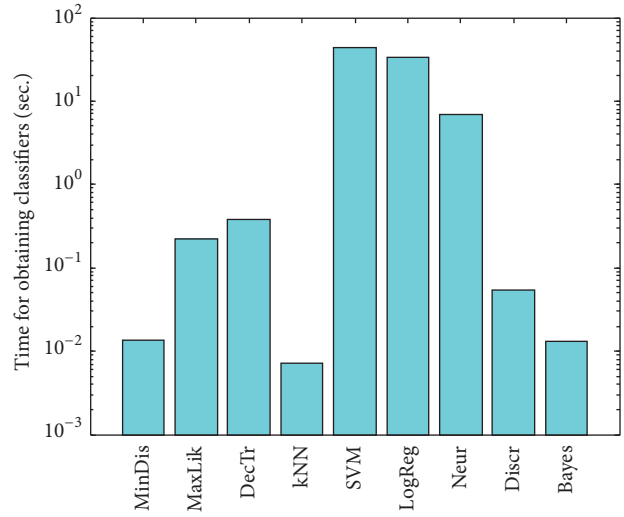

Figure 18: Classifier generation time (for the full MPEG-7 feature set).

to get some knowledge in comparing classifier generation times.

Following the structure of this paper, the first analysis is focused on the classifiers for nonsequential analysis (only a single frame is considered). Figure 18 shows (in logarithmic scale) the times required to obtain every classifier, using as input patterns the set of 13,903 frames with the full set of MPEG-7 features (18). Although some of these times are highly valued for certain algorithms (several tens of seconds), this fact does not make it a great challenge because, as it was already mentioned, the classifiers are off-line generated and they only have to be obtained once.

At a first glance, these generation times just depend on the number of features. But a deeper insight into the classification process shows that they also depend on the number of patterns and even their values. Therefore, in order to compare how the reduction of the number of features affects over these times, several trainings with different feature set (mixing all of them) have been performed as patterns. Figure 19 collects this information, averaging the data obtained for the different training data sets. It is important to note that vertical axis is logarithmically scaled, and its values are normalized by the classifier generation time when the full MPEG-7 feature set is considered (see also Figure 18).

Additionally, the number distribution and proportion of classes could have a certain impact on the time required to train a classifier. To explore this issue, as it was previously mentioned, the original dataset has been modified introducing additional classes (anuran species or sounds) with a random distribution and proportion. Figure 20 shows the results obtained that reveals, for most of the algorithm, a very limited influence (with the logistic regression classifier as the only remarkable exception).

Now, let us focus the analysis in the cases where frame sequence information is added, that is, when some features are constructed using regional dispersion, Δ parameters, or sliding window techniques. As it was seen above, these

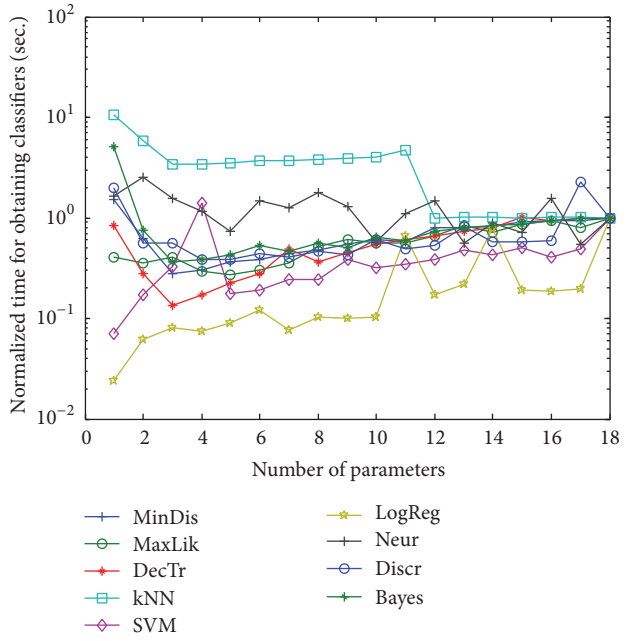

Figure 19: Normalized classifier generation time for different number of features.

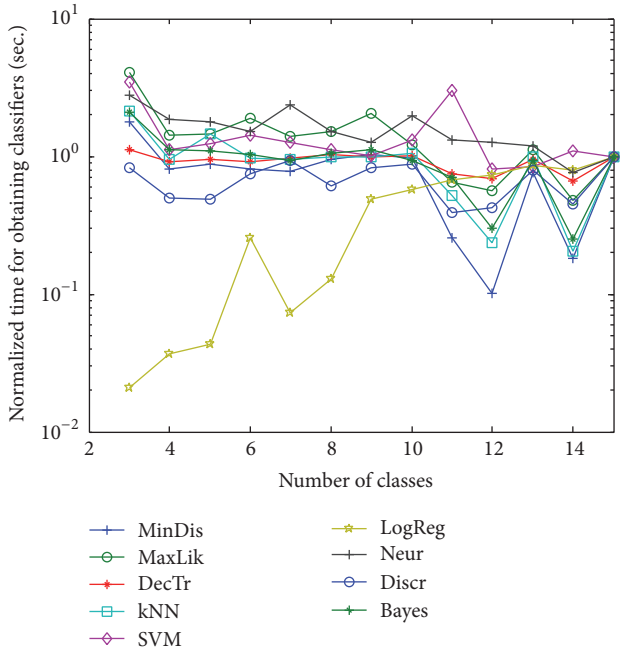

Figure 20: Normalized classifier generation time for different number of classes.

construction techniques can significantly increase the total number of features. In this sense, sliding window is the most restrictive (worst) case which, using a window with w frames, determines the use of $w \times D$ features in the classifier. Figure 21 summarizes the classifier generation times using the full MPEG-7 feature set and a window size of 10 (reaching a total of 180 features).

As for nonsequential classifiers, Figure 22 shows the generation times as a function of the number of features used.

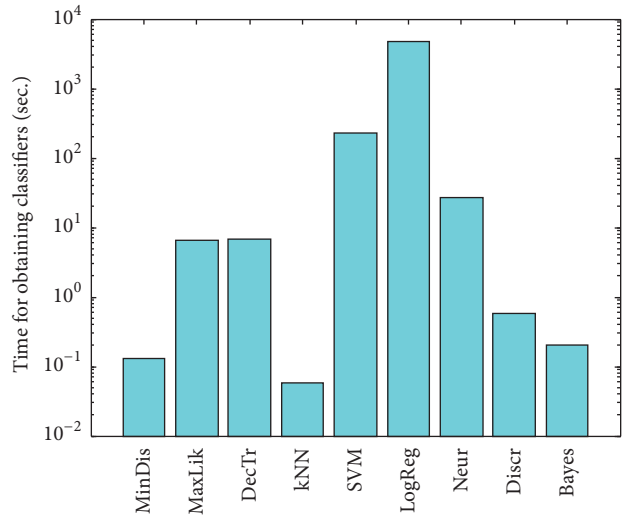

Figure 21: Classifier generation time (using the full MPEG-7 feature set and SW with a windows size of 10).

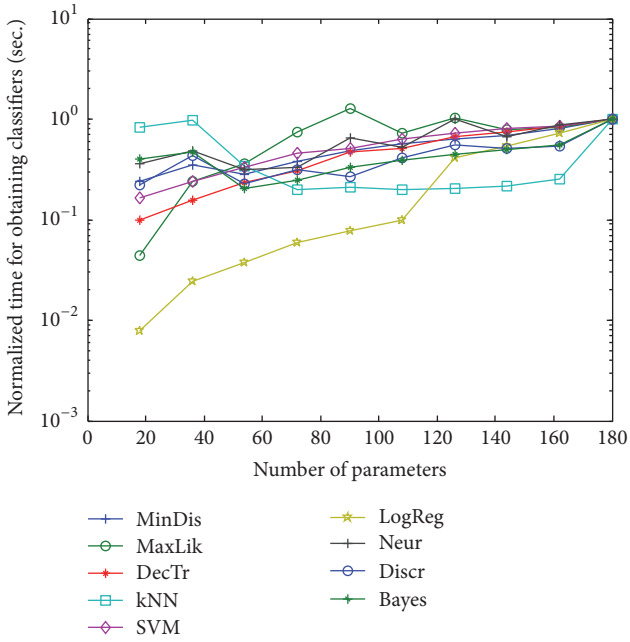

Figure 22: Normalized classifier generation time for different number of features (extended to constructed features).

This relationship has been analyzed with a large number of pattern combinations (mixing all of them) and then averaging all performance data from different training processes. It is important to note that vertical axis also is logarithmically scaled, and its values are normalized by the classifier generation time when the full MPEG-7 feature set is considered (180). Thus, it is easy to note that the classifier generation time increases with the number of features for most algorithms, some of these growths being very intense (between one or two orders of magnitude).

Finally, the last concern of this analysis will be the generation times for sequential classifiers, that is, HMM and

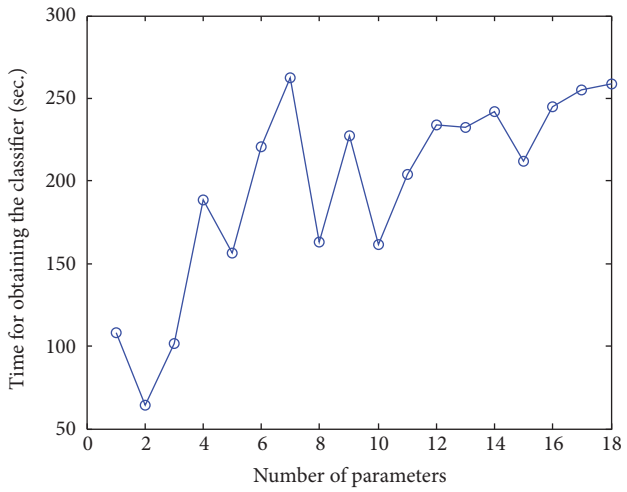

Figure 23: HMM classifier generation time for different number of features.

ARIMA models. In the first of these approaches (HMM), Figure 23 shows the time needed to obtain the classifier based on the training patterns (averaging its results among the different combinations or training sets). In this sense, it is easy to note that, for more than three features, the generation time exceeds the audio fragment duration (139 sec., see Table 2).

On the other hand, audio classification using ARIMA models uses the same classifiers previously considered, although increasing the feature set dimension. In this sense, the generation times of these classifiers will show the same results to those analyzed above (Figure 22).

7. Conclusions

Throughout this paper, an animal voice classification scheme for WASN has been proposed. This scheme proposes different alternatives to achieve this goal, always taking into account the power composition limitations of these kinds of platforms. In this sense, this paper is completed with a detailed comparative time study of each proposed algorithm within the scheme. It has been possible to find a tradeoff between the classification result accuracy and the required processing time.

From this analysis, several conclusions can be highlighted. For example, MPEG-7 feature extraction requires an important relative computational load (around of 30% of the audio fragment time). Conversely, this load falls to 0.5% for MFCC extraction time, considerably reducing the computational load. Additionally, it is easy to note that most feature construction techniques (either adding frame trend or sequential information) require a low processing cost, ranging approximately between the 1% of the frame time for regional dispersion or HMM and the 0.1% for sliding windows. Conversely, ARIMA models significantly exceed this limit where classification times exponentially grow with the number of features. For the first classification stage, it is also easy to note that the classification time depends remarkably on the type of classifier and the number of parameters (as it can be seen in the different comparisons). However, these requirements are also typically low (between 0.1% and 1% of the frame duration). Only in two of them (maximum likelihood and k-neighbors), this time reaches up to the 40%, which could jeopardize its application to real-time classification. Finally, although classifier generation times do not affect its real-time capabilities, it could be useful in systems with dynamic knowledge base, increasing some of them (i.e., logistic regression, SVM, or HMM) several orders of magnitude respect to others with lower computational cost (minimal distance and k-NN).

From an implementation approach, a first result indicates that the proposed prototype for anuran song classification is able to operate in real-time, taking all alternatives less than the audio duration. Thus, some concerns have to be taken into account when this algorithm is deployed in a WASN node (typically with fewer resources). In this sense, these potential node limitations could be easily compensated with the Digital Signal Processing (DSP) resources, commonly available in modern platforms for this purpose (i.e., ARM® Cortex®-M4 processes), which would greatly reduce feature extraction times (one of the most costly phases in the MPEG-7 approach). Additionally, a reduction in the sample rate could also be occasionally possible if it was necessary.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been supported by the *Consejería de Innovación, Ciencia y Empresa, Junta de Andalucía*, Spain, through the Excellence Project eSAPIENS (Ref. TIC-5705) and by *Telefónica*, through the "*Cátedra de Telefónica Inteligencia en la Red.*" The authors would like to thank Rafael Ignacio Marquez Martinez de Orense *("Museo Nacional de Ciencias Naturales")* and Juan Francisco Beltrán Gala (Faculty of Biology, University of Seville) for their collaboration and support.

References

[1] A. Menzel, T. H. Sparks, N. Estrella et al., "European phenological response to climate change matches the warming pattern," *Global Change Biology*, vol. 12, no. 10, pp. 1969–1976, 2006.

[2] R. Márquez and J. Bosch, "Advertisement calls of the midwife toads Alytes (Amphibia, Anura, Discoglossidae) in continental Spain," *Journal of Zoological Systematics and Evolutionary Research*, vol. 33, no. 3-4, pp. 185–192, 1995.

[3] D. Llusia, R. Márquez, J. F. Beltrán, M. Benítez, and J. P. do Amaral, "Calling behaviour under climate change: geographical and seasonal variation of calling temperatures in ectotherms," *Global Change Biology*, vol. 19, no. 9, pp. 2655–2674, 2013.

[4] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32–39, 2007.

[5] J. Luque, D. F. Larios, E. Personal, J. Barbancho, and C. León, "Evaluation of MPEG-7-based audio descriptors for animal voice recognition over wireless acoustic sensor networks," *Sensors (Switzerland)*, vol. 16, no. 5, article 717, 2016.

[6] J. Romero, A. Luque, and A. Carrasco, "Anuran sound classification using MPEG-7 frame descriptors," in *Proceedings of the 17th Conferencia de la Asociación Española Para la Inteligencia Artificial (CAEPIA)*, Salamanca, Spain, 2016.

[7] J. Romero, A. Luque, and A. Carrasco, "Animal sound classification using sequential classifiers," in *Proceedings of the 10th International Conference on Bio-Inspired Systems and Signal Processing*, pp. 242–247, Porto, Portugal, 2017.

[8] ISO15938-4:2001, "MPEG-7: Multimedia Content Description Interface, Part 4: Audio," 2001.

[9] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of MFCC," *Journal of Computer Science and Technology*, vol. 16, no. 6, pp. 582–589, 2001.

[10] IEEE Standard for Low-Rate Wireless Networks, "IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)," 2016, pp. 1–709.

[11] ZigBee Specification, ZigBee Alliance, ZigBee Document 053474r06, 2006.

[12] J. W. Hui and D. E. Culler, "Extending IP to low-power, wireless personal area networks," *IEEE Internet Computing*, vol. 12, no. 4, pp. 37–45, 2008.

[13] S. Young, G. Evermann, M. Gales et al., *The HTK Book*, vol. 3, Cambridge University Engineering Department, Cambridge, UK, 2002.

[14] ETSI Std 202 050, "1.3 Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithms," 2002.

[15] M. Schaidnagel, T. Connolly, and F. Laux, "Automated feature construction for classification of time ordered data sequences," *International Journal on Advances in Software*, vol. 7, no. 3, pp. 632–641, 2014.

[16] S. Sharma, A. Shukla, and P. Mishra, "Speech and language recognition using MFCC and DELTA-MFCC," *International Journal of Engineering Trends and Technology*, vol. 12, no. 9, pp. 449–452, 2014.

[17] C. C. Aggarwal, *Data Streams: Models and Algorithms*, vol. 31, Springer Science & Business Media, New York, NY, USA, 2007.

[18] G. E. Box, G. M. Jenkins, G. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, Wiley Series in Probability and Statistics, John Wiley & Sons, Hoboken, NJ, USA, 5th edition, 2015.

[19] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[20] C. Hevia, "Maximum likelihood estimation of an ARMA (p, q) model," The World Bank, DECRG, 2008.

[21] L. R. Rabiner, "Tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[22] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications Systems*, vol. 28, no. 1, pp. 84–95, 1980.

[23] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," *Bulletin of the American Mathematical Society*, vol. 73, pp. 360–363, 1967.

[24] A. G. Wacker and D. A. Landgrebe, "The minimum distance approach to classification," The Laboratory for Applications of Remote Sensin (Purdue University), 1971.

[25] L. L. Cam, "Maximum likelihood: an introduction," *International Statistical Review/Revue Internationale de Statistique*, vol. 58, no. 2, pp. 153–171, 1990.

[26] L. Rokach and O. Maimon, "Data mining with decision trees: theory and applications," World Scientific, 2014.

[27] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[28] N. Christianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.

[29] A. J. Dobson and A. G. Barnett, *An Introduction to Generalized Linear Models*, Texts in Statistical Science Series, CRC Press, Boca Raton, Fla, USA, 3rd edition, 2008.

[30] K.-L. Du and M. N. S. Swamy, *Neural Networks and Statistical Learning*, Springer Nature, New York, NY, USA, 2014.

[31] W. Härdle and L. Simar, *Applied Multivariate Statistical Analysis*, Springer Science & Business Media, New York, NY, USA, 2012.

[32] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, New York, NY, USA, 2009.

[33] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, article 12, 2012.

[34] Fonozoo.com, http://www.fonozoo.com/.