Temporally-aware algorithms for the classification of anuran sounds

Amalia Luque Corresp., 1, Javier Romero-Lemos 1, Alejandro Carrasco 2, Luis Gonzalez-Abril 3

1 Departamento de Ingeniería del Diseño, Universidad de Sevilla, Sevilla, Spain
2 Departamento de Tecnología Electrónica, Universidad de Sevilla, Sevilla, Spain
3 Departamento de Economía Aplicada I, Universidad de Sevilla, Sevilla, Spain

Corresponding Author: Amalia Luque
Email address: amalialuque@us.es

Several authors have shown that the sounds of anurans can be used as an indicator of climate change. Hence, the recording, storage and further processing of a huge number of anuran sounds, distributed over time and space, are required in order to obtain this indicator. Furthermore, it is desirable to have algorithms and tools for the automatic classification of the different classes of sounds. In this paper, six classification methods are proposed, all based on the data-mining domain, which strive to take advantage of the temporal character of the sounds. The definition and comparison of these classification methods is undertaken using several approaches. The main conclusions of this paper are that: i) the sliding window method attained the best results in the experiments presented, and even outperformed the Hidden Markov Models usually employed in similar applications; ii) noteworthy overall classification performance has been obtained, which is an especially striking result considering that the sounds analysed were affected by a highly noisy background; iii) the instance selection for the determination of the sounds in the training dataset offers better results than cross-validation techniques; and iv) the temporally-aware classifiers have revealed that they can obtain better performance than their non-temporally-aware counterparts.
Temporally-aware algorithms for the classification of anuran sounds

Amalia Luque¹, Javier Romero-Lemos¹, Alejandro Carrasco², Luis Gonzalez-Abril³

¹ Dpto. Ingeniería del Diseño, Universidad de Sevilla, Sevilla, Spain
² Dpto. Tecnología Electrónica. Universidad de Sevilla, Sevilla, Spain
³ Dpto. Economía Aplicada I. Universidad de Sevilla, Sevilla, Spain

Corresponding Author:

Amalia Luque¹
Abstract
Several authors have shown that the sounds of anurans can be used as an indicator of climate change. Hence, the recording, storage and further processing of a huge number of anuran sounds, distributed over time and space, are required in order to obtain this indicator. Furthermore, it is desirable to have algorithms and tools for the automatic classification of the different classes of sounds. In this paper, six classification methods are proposed, all based on the data-mining domain, which strive to take advantage of the temporal character of the sounds. The definition and comparison of these classification methods is undertaken using several approaches. The main conclusions of this paper are that: i) the sliding window method attained the best results in the experiments presented, and even outperformed the Hidden Markov Models usually employed in similar applications; ii) noteworthy overall classification performance has been obtained, which is an especially striking result considering that the sounds analysed were affected by a highly noisy background; iii) the instance selection for the determination of the sounds in the training dataset offers better results than cross-validation techniques; and iv) the temporally-aware classifiers have revealed that they can obtain better performance than their non-temporally-aware counterparts.

1. Introduction
Sound classification has become a major issue in numerous scientific and technical applications. Many techniques have been proposed to obtain the desired sound labelling: some for general purpose (Hinton et al., 2012) and others for specific applications (Cowling & Sitte, 2003).

Although sounds are inherently represented by a time-series of acoustic data, it is common to focus on small fragments of audio signals and attempt to classify them without considering the preceding or subsequent sound sections. For this reason, non-temporally-aware (NTA) methods are also frequently applied (Tzanetakis & Cook, 2002; Wang et al., 2006).

In order to clarify the temporal character of the sound in this paper, our interest lies in the evolution of its low-level short-duration frames and not to the sequence of acoustic units commonly used in bioacoustics (Kershenbaum et al., 2016).

The main aim of this paper is to analyze and compare temporally-aware and non-temporally-aware classifiers and to show that the consideration of temporal information clearly improves classification performance.

Let us indicate that the study presented here could eventually be applied to the study of global warming, since the sounds produced by certain animal species have been revealed as a strong indicator of temperature changes and, therefore, of the existence of climate change. Of particular interest are the results provided by anuran-sound analysis (Márquez & Bosch, 1995), and hence these kinds of sounds are analysed in this paper.
As a widely distributed taxonomic group, anurans are considered excellent indicators of biodiversity. However, frog populations have been experiencing dramatic declines over the past decade due to habitat loss, climate change, and invasive species (Xie et al., 2017). Therefore, long-term monitoring of frog populations is becoming increasingly important in the optimization of conservation policy.

It is worth noting that the sound production mechanism in ectotherms is strongly influenced by the ambient temperature (Llusia et al., 2013). Hence, the temperature may significantly affect the patterns of calling songs by modifying the beginning, duration, and intensity of calling episodes and, consequently, the anuran reproductive activity. The presence or absence of certain anuran calls in a certain region, and their evolution over time, can therefore be used as an indicator of climate change.

The first step in biological species identification involves the recording of different sounds in their natural environment, where different devices can be used. Processing of the recorded sounds can be performed either locally in real time (Aide et al., 2013), or in a remote centre requiring, in this case, a suitable communication system, usually a wireless sensor network (WSN), which generally requires information-compressing technologies (Diaz et al., 2012).

In previous work (Luque et al., 2016), a non-temporally-aware method for sound classification has been proposed. According to this procedure, the sound is split up into frames. Every frame is subsequently featured using 18 parameters (also called features). The frame features are then compared to certain frame patterns belonging to known sounds, thereby assigning a class label to each frame. Finally the sound is classified by frame voting, for which up to 9 different algorithms have been proposed (Luque et al., 2018; Romero, Luque & Carrasco, 2016). For the determination of the sounds which should be included in the training dataset, instance selection and cross-validation techniques are considered and compared.

However, sounds are inherently made up of a time series of acoustic data. Therefore, if the temporal information of the frame is added to the classification process, then better classification results should be expected. It must be borne in mind that the goal of classification is to recognize species and, more precisely, their different vocalizations.

The paper is organized as follows: Section 2 describes the anuran dataset and presents the methodology employed to compare classifiers, by depicting its general schema, the six (three frame-based, three segment-based) approaches to temporally-aware classification, the classification algorithms considered, and the performance metrics employed. The application for the classification of a set of actual anuran sounds is presented in Section 3, where the results of the six approaches are compared with each other and also with non-temporally-aware classifiers.
2. Materials & Methods

2.1. Sound dataset

For testing purposes, actual anuran sounds provided by the National Natural History Museum (Fonozoo, 2017) have been employed. The sounds correspond to 2 species, the _epidalea calamita_ (natterjack toad) and _alytes obstetricans_ (common midwife toad), with a total of 868 recordings containing 4 classes of sounds:

1. _Epidalea calamita_; mating call (369 records)
2. _Epidalea calamita_; release call (63 records)
3. _Alytes obstetricans_; mating call (419 records)
4. _Alytes obstetricans_; distress call (17 records)

A total of 4,343 seconds of recording have been analysed, with an average duration of 5 seconds. A common feature of all the recordings is that they have been taken in their natural habitat, with very significant surrounding noise (wind, water, rain, traffic, voices, etc.), which posed an additional challenge in the classification process.

To perform a supervised classification, certain sounds have to be selected as patterns (to be used in the training phase) and others are employed for validation and testing purposes. A common practice is to split the dataset into several disjoint subsets and apply a cross-validation technique (4 folds have been used in the paper). However, use of these noisy recordings as patterns may lead to a decrease in the classification performance. Hence, several other approaches arise as an alternative to cross-validation. In our case, recordings with relatively low background noise, which were carefully selected by biologists and sound engineers, have been used as patterns.

This approach, usually called instance or example selection, is recommended in order to increase the rate of learning by focusing attention on informative examples (Blum & Langley, 1997; Raman & Ioerger, 2003; Olvera-López et al., 2010; Borovicka et al., 2012). In order to determine the frame patterns, the experts listen to the recording of the anuran calls and simultaneously consider the spectrogram and the set of MPEG-7 features, and label each frame that they consider may belong to any of the possible classes.

The parameters for every classifier are determined by exclusively using the pattern records (training dataset). The remaining elements in the dataset are then divided into two approximately equal subsets used for validation and testing. The validation dataset is employed to determine the hyper-parameters of the classifiers, such as the number \( r \) of relevant features, the number \( w \) of frames to be considered in SW, RSW and HMM-SW, and the number \( T \) of recurrent inputs in RNN. On the other hand, the testing dataset, which includes none of the
patterns nor validation sounds, is employed for the evaluation of the performance of every
algorithm. Table 1 summarizes the sound and pattern dataset.

2.2. General description of the classification methodology
The general schema of the proposed procedure, depicted in Fig. 1, is based on the following
steps:

1. The sound is split up into 10-ms frames. This is the frame length recommended by MPEG-7
since it is the approximate time period for the opening and closing of the anuran vocal
cords.

2. Every frame uses D MPEG-7 features: a vector $\mathbf{x}$ in $\mathbb{R}^D$ (ISO, 2001). The series of $S$ frame
vectors $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_S]$ makes up the $\mathbf{X}$ matrix of dimension $M \times D$, which features the full
sound. Feature extraction using MPEG-7 features has been chosen because very good
results are shown in their description of sound frames for classification purposes, and these
features appear as serious competitors to MFCC features, which are widely used in many
applications (Herrera-Boyer, Peeters & Dubnov, 2003). MPEG-7 and MFCC features show
similar classification performance but, although MPEG-7 extraction could require more
computational effort, it enjoys several advantages: it is semantically richer (in the sense that
it is easier to intuitively grasp its meaning); it is fully standardized for general-purpose
applications; and it presents better performance when a reduction in the number of features
is required.

3. Temporal information is considered by using one of two approaches:

3.1. Frame-based approach: for every frame, a vector $\mathbf{y}$ of $C$ additional features is
constructed (Liu & Motoda, 1998) by applying a function $f$ to the matrix $\mathbf{X}$ of the
MPEG-7 original features and hence $\mathbf{y} = f(\mathbf{X})$. Therefore, every frame is featured using
$D + C$ features, that is, a vector $\mathbf{z} = \mathbf{x} \cup \mathbf{y}$ in $\mathbb{R}^{D+C}$. Certain forms of the function $f$, for
instance, are comprised of statistical measures of $\mathbf{X}$, or the concatenation of the vectors
corresponding to stacked frames.

3.2. Segment-based approach: every series of $N$ sound frames is represented by a model
using its $N \times D$ features.

4. Every sound fragment, either in terms of a frame or segment, is classified by using one of
two approaches:

4.1. The frame features are compared to frame patterns belonging to known species, thereby
assigning a class label to each frame. By means of the feature extraction and construction
procedures previously described, each sound frame is characterized by $D + C$ features or,
equivalently, by a point in an $\mathbb{R}^{D+C}$ space defined by its coordinate vector $\mathbf{z}$. $N$ pattern
frames are also available where the $i$-th pattern is also represented by a point in the
space with a coordinate vector $\mathbf{\pi}_t$. Each frame is labelled as belonging to a certain
class $\theta$ out of a total of $M$ classes. The set of pattern frames can be viewed as a cloud of
points in $\mathbb{R}^{D+C}$ and can be identified by a matrix $\mathbf{\Pi} = [\mathbf{\pi}_1, \mathbf{\pi}_2, \ldots, \mathbf{\pi}_N]^\top$ containing the
coordinate vectors of the $N$ points. The subset of points in $\mathbf{\Pi}$ belonging to the class $\theta$ is
denoted by its matrix $\mathbf{\Pi}_\theta$. Non-temporally-aware classifiers perform a certain type of
comparison between the frame to be classified (represented by its vector $\mathbf{y}$) and the
pattern frames (represented by its matrix $\mathbf{\Pi}$). This comparison is carried out in the space
of the $\mathbb{R}^{D+C}$ features and its result is called a supervised classification. A wide and
representative set of non-sequential supervised classifiers is considered and these are
described below.

4.2. The segment models are compared to segment patterns belonging to known species, and
a class label is thereby assigned to each segment.

5. Finally the sound is classified by means of frame or segment voting.

2.3. MPEG-7 feature extraction and selection

The task of extracting MPEG-7 features from every sound frame is accomplished by using three
different processes: spectrogram analysis, linear prediction coding (LPC), and harmonicity
analysis. Hence, $D = 18$ features are obtained by following Luque et al. (2016), which is
summarized in Table 2.

As shown below, the consideration of temporal information associated to the frames usually
leads us to significantly increase the number of features required. In order to cope with this
drawback, a reduction in the number of the 18 original MPEG-7 features is proposed by
considering the $r$ most significant features of each frame (leading to a vector in $\mathbb{R}^r$). Feature
selection procedures are employed to determine the relevance-ordered set of features and its
optimal size (Guyon et al., 2006).

The feature selection technique used in the paper is based on the Jensen-Shannon divergence
(Lin, 1991). It obtains the separability of the sound classes for every feature by applying the
following procedure:

1. Consider the set of the $N$ pattern frames represented by the matrix $\mathbf{\Pi} = [\mathbf{\pi}_1, \mathbf{\pi}_2, \ldots, \mathbf{\pi}_N]^\top$.

   Focus on the subset of elements in $\mathbf{\Pi}$ belonging to the $k$-th class $\theta_k$, which is denoted by its
   matrix $\mathbf{\Pi}_k$ and, specifically, in the $i$-th pattern frame $\mathbf{\pi}_i \in \mathbf{\Pi}_k$. The vector $\mathbf{\pi}_i$ contains $D$
elements, one for each feature. The $j$-th feature corresponding to the $i$-th pattern frame is
denoted as $\pi_{ij} | \mathbf{\pi}_i \in \mathbf{\Pi}_k$. Let us denote $\varphi_{jk}$ as the set of values of the $j$-th feature in every
frame belonging to the $k$-th class ($\theta_k$): $\varphi_{jk} = \{\pi_{ji} | \forall i | \mathbf{\pi}_i \in \mathbf{\Pi}_k\}$. 
2. Estimate the probability density function (pdf) $f_{jk}$ of the values in $\varphi_{jk}$, that is, of the $j$-th feature values for those pattern frames belonging to the $k$-th class.

3. For the $j$-th feature and every pair of classes $u$ and $v$, an indication is obtained of how separate the corresponding $f_{ju}$ and $f_{jv}$ pdfs are. For this purpose, the Jensen-Shannon divergence is used, which is given by

$$D_{JS}(f_{ju}, f_{jv}) = \frac{1}{2} \int_{-\infty}^{\infty} f_{ju} \log_2 \frac{2f_{ju}}{f_{ju} + f_{jv}} dx + \frac{1}{2} \int_{-\infty}^{\infty} f_{jv} \log_2 \frac{2f_{jv}}{f_{ju} + f_{jv}} dx.$$ (1)

4. Every value of the Jensen-Shannon divergence is transformed in the corresponding distance, which is given by

$$d_{JS}(f_{ju}, f_{jv}) \equiv \sqrt{D_{JS}(f_{ju}, f_{jv})}.$$ (2)

5. For the $j$-th feature, the separability index $\Psi_j$ is derived, in accordance with

$$\Psi_j \equiv \sqrt{\prod_{u=1}^{A-1} \prod_{v=i+1}^{A} d_{JS}(f_{ju}, f_{jv})},$$ (3)

where $A$ is the total number of classes and $B$ is the number of pairs of classes $u$ and $v$, which is given by

$$B = \frac{A(A-1)}{2}.$$ (4)

The separability index $\Psi_j$ for the $j$-th feature is an indicator of how separate the pdfs are corresponding to each class. The more separate the pdfs are, the more useful (or relevant, or significant) that feature is for classification. Hence, the value of $\Psi_j$ is used as an indicator of the relevance of the $j$-th feature.

For comparison purposes, 2 non-temporally-aware methods are also considered:

1. Non-temporally-aware classification based on 18 MPEG-7 features (NS-18).
2. Non-temporally-aware classification based on the $r$ most relevant MPEG-7 features (NS-$r$).
2.4. Feature construction

In order to consider the temporal behaviour of a sound, the frames should not be considered one by one, but the preceding and subsequent frames should also be taken into account, that is, their ordered succession should be considered. Several methods have been proposed in the literature to include this temporality (Dietterich, 2002; Esling & Agon, 2012). A number of these methods can be considered frame-based, that is, they still classify frames but now the frames are featured with additional information on the temporal context. Alternatively, other approaches are defined as segment-based as they do not classify isolated frames but instead classify a series of frames (a segment). First, three frame-based approaches are described:

1. Construction of Local InterQuartile Range (LIQR) features (Schaidnagel, Connolly & Laux, 2014). The general idea for this feature construction technique is to use the time axis to construct new temporally-aware features. These techniques are commonly based on the values of the features of the frame without considering their order, which is usually called a bag of features. Average values or other related statistics are usually employed.

In the case of the anuran calls to be classified, the typical croaking of a frog is found, while other calls are similar to the sound of a whistle. The croaking sound is produced by repeatedly opening and closing the vocal cords (roughly every 10 ms, equal to the frame length) leading to a series of frames featured with widely spread values (Fay, 2012). On the other hand, the whistle-like sounds are produced by a continuous air flow showing a narrow spread in feature values. For the incorporation of this information into the classification process, a new set of features is therefore constructed that considers the spread of the extracted feature values and not their average. Furthermore, in order to avoid the influence of outliers, the interquartile range (IQR) is selected instead of the standard deviation.

In the implementation used, first for every frame, a “window” centred on that frame is considered, using the closest neighbouring frames. For every original feature, a new derived feature is constructed. To this end, the values of the original feature for every frame in the window are considered. The interquartile range of these values is computed, and this value is considered the new derived feature. In this way, the number of constructed features is $C = D$, and hence up to $2 \times D$ features (a vector in $\mathbb{R}^{2 \times D}$) are now identifying a frame, where $C$ of these features include temporal information. In this approach, a window size of 10 frames (100 ms) has been used.

2. Sliding window (SW) (Aggarwal, 2007). In this technique, also known as frame stacking or shingling, a short window with $w$ frames, centred on each frame, is considered. An odd-numbered value is usually chosen for the window size, that is, $w = 2d + 1$, where $d$
is an integer. The class $\theta_i$ for the $i$-th frame is obtained using a classifying function $f_c$, as follows:

$$\theta_i = f_c(x_{i-d}, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{i+d}),$$

(5)

where $x_j \in \mathbb{R}^D$ represents the feature vector for the $j$-th frame. The $D$ features describing each frame are now those corresponding to all the frames under the window. Therefore, each frame is featured using $w \times D$ features (a vector in $\mathbb{R}^{w \times D}$). In this approach, the number of features describing each frame can significantly increase, thereby inflicting a major impact on the computing resources required in the classification process. For this reason, only the $r$ most relevant features have been used by applying the aforementioned feature selection techniques.

3. Recurrent sliding window (RSW) (Joshi & Dietterich, 2003). This is a method similar to the SW procedure above, except that the classifier now considers not only the features of the frame under the window, but also their previous classification results. Thus, the class $\theta_i$ for the $i$-th frame is obtained as follows:

$$\theta_i = f_c(\theta_{i-d}, \ldots, \theta_{i-1}, x_{i-d}, \ldots, x_i, \ldots, x_{i+d}),$$

(6)

2.5. Non-temporally-aware classifiers
Every frame-based approach (and also the segment-based ARIMA model, described below) relies on an underlying non-temporally-aware classifier. A broad and representative selection of these classifiers has been used throughout this paper: Minimum distance (MinDis), (Wacker & Landgrebe, 1971); Maximum likelihood (MaxLik), (Le Cam, 1990); Decision tree (DecTr), (Rokach & Maimon, 2008); k-nearest neighbours (kNN), (Cover & Hart, 1967); Support vector machine (SVM), (Vapnik, 1998); Logistic regression (LogReg), (Dobson & Barnett, 2008); Neural network (Neur), (Du & Swamy, 2013); Discriminant function (Discr), (Härdle & Simar, 2012); and Bayesian classifier (Bayes), (Hastie, Tibshirani & Friedman, 2005).

All these classifiers have been prototyped using MATLAB. The minimum distance classifier in its training phase obtains the mean value $\mu_{jk}$ for the $j$-th feature belonging to the $k$-th class. In the test phase for the $i$-th frame, the distance $d_{ik}$ between the frame features and the mean value of the $k$-th class $\theta_k$ is obtained in accordance with the expression

$$d_{ik} = \sqrt{\sum_{j=1}^{D} (x_{ji} - \mu_{jk})^2},$$

(7)

where $x_{ji}$ is the value of the $j$-th feature corresponding to the $i$-th frame. The class assigned to the frame is that with the minimum distance.
The maximum likelihood classifier is used under a Gaussian probability distribution with full
covariance. The neural network classifier is based on a feed-forward neural network with a 10-
neuron hidden layer and a 1-neuron output layer. The remaining methods and classifiers have
been coded based on built-in MATLAB functions using their default parameters, which are
reflected in Table 3.

2.6. Segment modelling and classification
With respect to segment-based approaches for the introduction of temporal information, the
following methods are proposed:

1. Hidden Markov Models (HMM) (Rabiner, 1989). This is a genuinely temporally-aware
classifier which takes every sound frame \( x_i \in \mathbb{R}^D \) and assigns it with a discrete label
(Linde, Buzo & Gray, 1980; Brookes, 2006), thereby obtaining an observation \( O_i \), which
is an integer number \( c_k \) (a code) in the range \([0,C-1]\). The series of observations is
assumed to be produced by an HMM made up of \( N \) connected states \( S \), where the \( S_a \) state
emits the observation code \( c_k \) with an emitting probability \( E_{ak} \), and evolves to the state \( S_b \)
with a transition probability \( T_{ab} \). For the recognition of isolated “words” (anuran calls),
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, the differences in error rate for values of \( N \) that are close to 5
are small. The structure and the value of \( N \) have been taken from (Rabiner, 1989) and
they are depicted in Fig. 2. The \( E \) and \( T \) matrices are obtained for each class \( \theta \) from their
corresponding pattern frames \( \Pi_\theta \) using the forward-backward algorithm (Baum & Eagon,
1967). Once the HMMs are identified, the algorithm takes the series of observations for
the full sound segment to be classified (and not only for a single frame), and estimates the
probability of being produced by the HMM of each class. The full sound segment is
labelled as belonging to the class with the highest probability. When a sound file has to
be classified, 3 alternatives for the determination of the segment length have been
explored:

- The full sound file (HMM-F).
- A segment with the same length as the ROI mean length (HMM-ROI). The
  Regions Of Interest (ROIs) are the segments of the sound patterns containing a
  valid sound (no silence or noise).
- A segment defined by a sliding window of a certain length (HMM-SW).

This is the classifier recommended in the MPEG-7 standard. In this technique, the \( r \) most
significant values have been used, where \( r \) is a parameter to be chosen from the
experimentation. Additionally, the HMM classifiers tested in the paper use a 256-code ($C = 256$) quantization codebook.

2. Recurrent Neural Networks (RNN) (Parascandolo, Huttenen & Virtanen, 2016). The series of frame features $x_i$ is introduced into a Neural Network with $H$ neurons in its hidden layer, which produces an intermediate output $y_i$. The previous outputs $y_{i-1}$ to $y_{i-T}$ are then introduced as new inputs of the network (Fig. 3). A value of $H = 10$ neurons in the hidden layer is used throughout the paper.

3. Autoregressive integrated moving average (ARIMA) models (Box, Jenkins & Reinsel, 2011). The series of frame features $x_i \in \mathbb{R}^D$ is considered the result of the vector ARIMA time series, $VARIMA(a,d,b)$, defined as

$$x_i^{(d)} = C_0 + \sum_{k=1}^{a} A_k x_{i-k}^{(d)} + \sum_{k=1}^{b} B_k \epsilon_{i-k} + \epsilon_i \tag{8}$$

where $a$ is the order of the autoregressive model, $d$ is the degree of differencing, and $b$ is the order of the moving-average model. The coefficient matrices $A_k$ and $B_k$ have a $D \times D$ dimension, and the $C_0$ vector, representing the time series mean, has $D$ components. In this case, the number of features describing the sound segment is $(a + b) \times D^2$. For the sake of simplicity, the stationarity of time series ($d = 0$) is assumed. On the other hand, VARMA models can be approximated by equivalent VAR models ($b = 0$). In this case, the optimum value for the remaining order of the model ($a$) is obtained using the Akaike Information Criterion (AIC) (Akaike, 1974), and the $A_k$ matrices are estimated using a maximum-likelihood technique (Hevia, 2008). Every sound segment, featured with $a \times D^2$ parameters, can now be labelled using non-temporally-aware classifiers.

In order to determine the order of the model ($a$), first the optimum order for every $k$-th ROI pattern (in the training dataset) is computed using a weighted AR mean order $\bar{a}_k$, derived as

$$\bar{a}_k = \frac{\sum_{i=1}^{O_M} i \cdot AIC_{ik}}{\sum_{i=1}^{O_M} AIC_{ik}} \tag{7}$$
where $AIC_{ik}$ is the AIC value for the $k$-th ROI pattern modelled as a VAR model of order $i$, and $O_M$ is the maximum VAR order considered ($O_M = 10$ is used). The optimum value for the VAR order model is then determined by

$$a = \frac{1}{N_{ROI}} \sum_{i} a_{ki} \tag{9}$$

where $N_{ROI}$ is the number of ROI segment patterns.

### 2.7. Classification performance metrics

The definition of the proper performance indicators constitutes a key aspect in the evaluation of procedures, and it is difficult to overstate its importance (Sturm, 2014). In order to compare the results obtained for every classifier, several metrics for the performance of a classifier can be defined (Sokolova & Lapalme, 2009), all of which are based on the confusion matrix (Table 4).

The most relevant metrics and their definitions are shown in Table 5, where they are computed for each class that is considered “positive”, as compared to the remaining classes, which are considered “negative”. Additionally, an average value per class can be defined for each metric.

Since, in the dataset, the number of instances in every class remains imbalanced (see Table 1), the use of accuracy or precision as the main performance metric can imply a significant skew (Chawla, 2005). It is therefore preferred to use sensitivity and specificity since these remain unbiased metrics even when the classes are imbalanced (Gonzalez-Abril et al., 2014; Gonzalez-Abril et al., 2017). Therefore, when a single metric is required for the comparison of classifier results (i.e. to identify “the best classifier”), the Geometric Mean or the AUC values are preferred since they combine, in a single metric, the sensitivity and the specificity which both present better behaviour in the presence of imbalanced classes. The AUC is more commonly employed and is the metric used for the selection of the best options and/or classifiers throughout the paper. When only one point is available in the ROC space, the value of the AUC is computed as the arithmetic mean of sensitivity and specificity.

### 2.8. Confidence interval of the classification performance metrics

Once the classification performance metrics are obtained, it is good practice to estimate the confidence interval of their values. To undertake this task, a bootstrap analysis is performed (Efron & Tibshirani, 1994). Firstly consider the testing dataset $T$ containing $S$ sounds. From this dataset, $S$ samples are then taken with replacement and a new $T_1$ dataset is obtained. Due to the replacement in the sampling process, certain sounds are not contained in $T_1$, while others are repeated at least once. The classification metrics vector $\mu_1$ can now be computed for the $T_1$ dataset.
This process is repeated $N_b$ times (usually a large number), thereby obtaining datasets $T_1, T_2, \ldots, T_{N_b}$ and their corresponding metrics vectors $\mu_1, \mu_2, \ldots, \mu_{N_b}$. This set of metrics vectors is employed to estimate the probability density function (pdf) of the metrics vector $f(\mu)$ and other related statistics. This procedure is commonly employed to derive the confidence interval of the classification metrics. Therefore, considering the metric $\mu_k$, which is the $k$-th metric in the $\mu$ vector, and its pdf $f_k(\mu_k)$, the confidence interval of $\mu_k$, for a given confidence level $\gamma$, is the interval between the values $u_k$ and $v_k$ such that $\Pr[u_k \leq \mu_k \leq v_k] = \gamma$. The value of $u_k$ can be estimated as the $\gamma/2$ percentile of $\mu_k$, and the value $v_k$ as the $100 - (\gamma/2)$ percentile.

Throughout this paper, bootstrap analysis with $N_b = 10000$ and a confidence level of $\gamma = 95\%$ is used. Bootstrap analysis can also be employed to estimate the probability that a certain metric outperforms another. For every $T_j$ dataset, the classification methods 1 and 2 are employed and their metric vectors $\mu_{j1}$ and $\mu_{j2}$ are computed. The difference between these metric vectors is then derived by $\delta_j = \mu_{j1} - \mu_{j2}$. The probability density function (pdf) of the differences vector $f(\delta)$ and the continuous density function (cdf), $F(\delta)$, can then be computed. Finally, considering the difference $\delta_k$, which is the $k$-th metric in the $\delta$ vector, and its cdf $F_k(\delta_k)$, the probability of outperforming, $\omega_k$, is the probability that $\delta_k > 0$, that is, $\omega_k = \Pr[\delta_k > 0] = F_k(0)$.

### 3. Results

#### 3.1. Instance selection vs. cross-validation

In Fig. 4, cross-validation and instance selection approaches are depicted for non-temporally-aware classification based on 18 MPEG-7 features (NS-18). As can be observed, most of the algorithms present a significantly better performance when the patterns are chosen using the instance selection method, with an increase of more than 7 points (in %) in the AUC metric of the centroid. Similar results are obtained for other temporally-aware and non-temporally-aware classifiers for which instance selection has been employed.

#### 3.2. Non-temporally-aware classification for a varying number of features

The results obtained by the non-temporally-aware classifiers based on 18 MPEG-7 features are compared using the ROC analysis, which is depicted in Fig. 5. The best result corresponds to the minimum distance classifier, with an AUC of 83.5%. This result is considered as the original baseline (denoted NTA-18) for future comparisons.

In order to prevent a high number of features entering the following temporally-aware algorithms, it could be convenient to reduce their number by selecting the $r$ most relevant MPEG-7 features. To determine the value of $r$, the AUC of the validation dataset is used. In Fig. 6, the AUC values for three non-temporally-aware classifiers are considered as a function of the
number of features. The classifiers in the figure are those showing the best AUC performance for values of \( r \): minimum distance, maximum likelihood, and decision tree. From this figure, it can be seen that using the 11 most relevant features \((r = 11)\), the best AUC is obtained. On the other hand, if the computing effort is a major concern and therefore the number of features becomes an important issue, selecting the 5 most relevant features \((r = 5)\) is a good balance between the AUC and the number of features. A further reduction would produce the steepest AUC decrease (below 75\%, decreasing more than 10 points) which is confirmed below (joint optimization subsection). The following subsections derive the results for both the reduced \((r = 5)\) and the optimum \((r = 11)\) number of features.

### 3.3. Classification with a reduced number of features

#### 3.3.1. Non-temporally-aware classification

For comparison purposes, the results obtained by the non-temporally-aware classifiers based on the 5 most relevant MPEG-7 features are also compared using the ROC analysis, which is depicted in Fig. 7. The best result corresponds to the decision-tree classifier with an AUC of 80.7\%. This result is considered as the reduced baseline (denoted NTA-5) for future comparisons.

#### 3.3.2. Determining the number of frames

In four of the proposed temporally-aware methods (SW, RSW, HMM-SW, and RNN) several consecutive frames have to be considered. The first issue is therefore to determine the optimum number of frames (also called window size \( w \)). For this purpose, the AUC of the validation dataset is used, which is represented in Fig. 8 for several temporally-aware methods (using the best underlying NTA classifiers) as a function of the number of frames. In that figure, instead of the AUC absolute value, the increase in the AUC is depicted, compared to the \( w = 1 \) case. This graphical approach clearly shows the advantage of using temporally-aware classifiers. In all the methods, except in the RNN, only an odd number of frames have been considered because they are preferred in those algorithms.

From this figure, by using a window size between 3 and 9 in the sliding window method, the AUC value can be enhanced by more than 6 points (in \%). With these considerations, a 7-frame sliding window \((w = 5)\) has been selected (its optimum value is denoted in the figure by a filled blue marker). This means a duration of 70 ms which roughly corresponds to 7 opening periods of the anuran vocal cords. Similarly, the optimum values of the number of frames for the remaining methods are RSW: 11; HMM-SW: 1; and RNN: 15.

#### 3.3.3. Local Interquartile Range classification

The first frame-based approach to temporally-aware classification is now considered: that of the construction of the local interquartile range (LIQR) features. The results corresponding to the ROC analysis are depicted in Fig. 9. The best result corresponds to the decision-tree classifier that has an AUC of 85.2\%. For most of the classifiers, the LIQR approach attains slightly better
results than does the equivalent non-temporally-aware classifier: a mean enhancement of about 5 points (in %) is achieved in the AUC value compared to the reduced baseline (NTA-5).

3.3.4. Sliding window classifiers
By considering the 5 most relevant features and a 7-frame window size, the sliding window method (SW7-5) is examined and its results compared through the ROC analysis, as presented in Fig. 10. The best result corresponds to the decision-tree classifier, with an AUC of 86.7%, which means an enhancement of about 6 points (in %) compared to the reduced baseline (NTA-5), and an enhancement of about 3 points (in %) compared to the original baseline (NTA-18).

The third frame-based approach to temporally-aware classification is now considered: the recurrent sliding window method. The results corresponding to the ROC analysis are depicted in Fig. 11, when 5 features \( r = 5 \) and an 11-frame window size \( w = 11 \) are considered (RSW11-5). The best result corresponds to the decision-tree classifier, which presents an AUC of 72.7%. For most of the classifiers, the recurrent sliding window approach obtains worse results than the equivalent non-temporally-aware classifier, with a mean decrease of about 13 points (in %) in the AUC.

3.3.5. Segment-based classifiers
The HMM is the first segment-based approach to the introduction of the temporal information into the classification process. The HMM takes a sound segment and attempts to classify it as a whole, without any framing. The results corresponding to its ROC analysis are depicted in Fig. 12. In this figure, the 5 most relevant features \( r = 5 \) are considered. The HMM over a segment defined by a sliding window (HMM-SW) of size \( w = 1 \), that is, over a single frame, obtains the best results among the HMM classifiers, with an AUC of 63.2% which, comparatively, is a poor result. Although the HMM is the classifier recommended in the MPEG-7 standard, it is clearly superseded by other non-temporally-aware techniques.

The second segment-based approach to temporally-aware classification is now considered: the recurrent neural network. Using a hidden layer with \( H = 10 \) neurons, 5 features \( r = 5 \), and 15 frames \( w = 15 \), that is, a number of \( T = 14 \) previous intermediate outputs, an AUC of 61.0% has been obtained. This result is also depicted in Fig. 12.

Finally, the ARIMA segment-based approach is considered. As stated before, a vector AR (VAR) simplified model is considered, where the 5 most relevant features are used \( r = 5 \). The first step is to determine the order of the VAR model \( a \) using the AIC criterion on the training dataset, as was described in the method section. The results are depicted in Fig. 13, where the AIC values have been normalized to the [0,1] interval. A white point is drawn at every \( k \)-th row indicating the weighted AR mean order \( a_k \) for the \( k \)-th ROI pattern. The optimum value for the
VAR order model is represented in the figure with a vertical white line, and has the value $\alpha = 3.36$. Its closest integer is used as the VAR order model, $\alpha = 3$.

Once the ARIMA models are determined, their parameters are classified using non-temporally-aware classifiers and their performances are also compared using the ROC analysis, as illustrated in Fig. 14. The best result corresponds to the decision-tree classifier with an AUC of 62.0%.

### 3.3.6. Comparing classifiers

Hitherto, partial results have been presented for every temporally-aware method. In order to obtain an overall perspective, a comparison of the six different methods proposed for temporally-aware classifiers is presented in Fig. 15 and Table 6, where the non-temporally-aware classifiers (original and reduced baselines) are also considered for reasons of contrast.

Additionally, a ROC analysis has also been accomplished for every method and its results are depicted in Fig. 16. From these results, it can be observed that the best performance corresponds to the sliding window approach (with an underlying decision-tree classifier). It shows the best AUC metric with a value of 86.7%. The sliding window method also has the best values for almost every performance metric. The only exceptions are the precision and the $F_1$ score (which depends on precision) which, although they present the highest values for the LIQR method, also present good figures for the sliding window method.

### 3.3.7. Bootstrap analysis

Using bootstrap analysis on the testing dataset, the probability density function (pdf) of the classification performance metrics can be obtained. The results, focusing on the best temporally-aware method (sliding window) and considering the AUC pdfs for different window sizes, are shown as a colour map in Fig. 17. The colours represent the probability density for every AUC given a certain window size.

Let us now centre on the sliding window method using the 7-frame optimum window size (as obtained using the validation dataset) and the reduced number of 5 features. This case, which is denoted as SW7-5, is now compared to the two non-temporally-aware baselines: one with the original number of features (NTA-18), and the second with the reduced number of features (NTA-5). The pdfs for the AUC in these 3 cases are depicted in Fig. 18.

Not only can Bootstrap analysis offer the confidence interval for every classification performance metric, but it can also, even more importantly, show how much the optimum temporally-aware classification method (sliding window SW7-5) improves the results above the two non-temporally-aware baselines (NTA-5 and NTA-18). The results for a 95% confidence level are shown in Table 7.

The AUC improvement over the two mentioned baselines obtained via the sliding window method for various window sizes is depicted in Fig. 19. In this figure, the 95% confidence interval is also shown.
3.4. Classification with the optimum number of features

3.4.1. Separate optimization

It is now time to turn our attention to the cases when the number of features is not such an important issue and it is affordable to use the $r = 11$ most relevant features. This number was obtained through an optimization procedure presented above.

Now, again, the next issue is to run a second and separate optimization process to determine the optimum number of frames for the methods requiring such a parameter. For this purpose, the AUC on the validation dataset is used, which is represented in Fig. 20 for several temporally-aware methods (using the best underlying NTA classifiers) as a function of the number of frames. In this figure, instead of the AUC absolute value, the increase of the AUC compared to the $w = 1$ case is depicted.

From this figure, by using a window size between 3 and 9 in the sliding window method, the AUC value can be enhanced by about 3 points (in %). With these considerations, a 3-frame sliding window ($w = 3$) has been selected (its optimum value, denoted in the figure by a filled blue marker). Similarly, the optimum number of frames for the remaining methods are RSW: 3; HMM-SW: 11; and RNN: 3.

Repeating the analysis of the various temporally-aware methods on the testing dataset, now using 11 features, the results obtained are presented in Fig. 21 and Table 8, where the non-temporally-aware classifiers (original and optimum baselines) are also considered for reasons of contrast.

Additionally, an ROC analysis has also been accomplished for every method and its results are depicted in Fig. 22. From these results, it can be observed that the best performance corresponds to the sliding window (and to the RSW) approach (with an underlying minimum distance). It shows the best AUC metric with a value of 88.4%. The sliding window (and RSW) method also has the best values for every performance metric. Although SW and RSW methods show exactly the same performance metrics on the testing dataset, the SW has been chosen as the best method for two reasons: it offers slightly better AUC on the validation dataset; and it provides better performance for non-optimum window sizes (see Fig. 20).

Bootstrap analysis can now offer the confidence interval on how much the optimum temporally-aware classification method (sliding window SW3-11) improves the results above the two non-temporally-aware baselines (NTA-11 and NTA-18). The results for a 95% confidence level are shown in Table 9.

3.4.2. Joint optimization

In the previous section the numbers of features and frames were separately optimized, that is, firstly the optimum number of features was determined and, subsequently, the optimum window size for that value was derived.
However, it is also possible to run a joint optimization process to simultaneously seek the optimum values for both parameters. By running this process on the validation set for the best temporal-aware method (sliding window), a set of AUC values for each pair of values of the parameters (number of features and window size) is obtained. The result is shown in Fig. 23 in the form of several lines (one per window size) that depict the increase of the AUC compared to the $w = 1$ case. In this figure, the maximum values (optimums in the number of features dimension) are represented by small filled circles. This figure also confirms that the selection of the 5 most relevant features ($r = 5$) provides a good balance between the AUC and the number of features (a result previously derived from Fig. 6).

An alternative way to represent the joint optimization process is to employ a bidimensional colour map, as in Fig. 24, which depicts the increase of AUC for every pair of values (number of features, window size). The optimums in the number of features dimension are marked with a black spot, while the optimums in the window size dimension are denoted by an empty circle. The overall optimum value, which is indicated with a cyan filled circle, is reached for 8 features and a 7-frame window (SW7-8).

In order to ascertain the impact of optimizing in each direction, Fig. 25 has been constructed. Given a certain number of features (x-coordinate for the corresponding blue point), the AUC can be optimized by changing the window size, given by a vertical movement in Fig. 24, and the maximum value is the y-coordinate for that blue point. Alternatively, given a certain window size (x-coordinate for the corresponding green point), the AUC can then be optimized through the selection of the proper number of features, given by a horizontal movement in Fig. 24, and the maximum value is the y-coordinate for that green point. It can be seen that, in almost every case, the optimization of the window size offers greater improvement than the optimization of the number of features.

In Fig. 23, the AUC is plotted as a function of the number of extracted (primary) features, which has been denoted as $D$. However, the sliding window method adds other $C = w \cdot D$ constructed (secondary or derived) features. Therefore, the total number of features defining the dimension of the space for classification purposes is $D + C$, and this is the value which has to be considered and kept as low as possible in order to reduce the computing requirements. For this reason, it is worth redrawing this figure in terms of the total number of features. The result is shown in Fig. 26. It can be seen that, if the total number of features is a concern, then the green line (corresponding to $w = 3$ frames) has a suboptimum (a secondary peak identified with an empty green circle) corresponding to 5 extracted features, that is, 15 total features. The corresponding classifier (SW3-5) should also be considered as a possible alternative.

### 3.5. Summary of results

Throughout the previous subsections, several classification methods have been identified. Firstly, there are 3 non-temporally-aware classifiers using: the original number of features (NTA-18); the reduced or balanced number of features (NTA-5), and the optimum number of features (NTA-
These 3 classifiers have been used as the baselines to determine the improvement achieved using other procedures. Later, when considering temporally-aware methods, the sliding window method has shown itself to be the most efficient. The determination of the window size in a separate optimization process identifies a classifier for a balanced number of (primary) features (SW7-5) and another classifier for the optimum number of features (SW3-11). On the other hand, the joint optimization of the number of features and frames leads to the detection of an optimum method (SW7-8) or of a classifier that balances the performance metric and the dimension of feature space (SW3-5). Table 10 summarizes these seven classification methods.

Using bootstrap analysis, the probability density function of each performance metric for each classification method can be estimated. The results regarding AUC are shown in Fig. 27 with the classification methods ordered in terms of the increasing number of total features. It can be seen that all the sliding window classifiers (except the SW3-5) obtain very similar results and improve the original baseline by about 5 points (NTA-18).

However, by considering another classification performance metrics, different results can be obtained. For instance, Fig. 28 depicts the comparison of the bootstrap analysis when the accuracy (ACC) is considered. Now all the sliding window classifiers clearly outperform the non-temporally-aware methods. The best classifier (SW7-8), which was obtained by a joint optimization process, increases the original baseline (NTA-18) by more than 10 points.

In order to compare each classification method using various performance metrics, a box plot has been drawn (Fig. 29). For each metric and each method, four elements are drawn: a filled box from the 25% to 75% percentiles; an upper vertical line from the 75% percentile to the upper limit of the confidence interval; a lower vertical line from the 25% percentile to the lower limit of the confidence interval; and a horizontal black line corresponding to the median value.

This same information is also presented in Table 11. Using AUC as the single performance metric, the overall best classifier is the SW3-11 which outperforms the baseline by about 5 points (requiring 33 features instead of 18). However, the SW7-8 classifier (requiring 56 features) outperforms SW3-11 in terms of accuracy, precision, specificity and F_1 score, and stands as the second best in terms of AUC. On the other hand, if the number of features is the greatest concern, then the NTA-11 classifier (requiring 11 features), still outperforms the original baseline with a much reduced number of features.

By considering not only the mean value of the improvements but also their statistical distribution, the confidence interval for each metric and method can be derived. These results are shown in Table 12, where the probability that the chosen method outperforms the original baseline (NTA-18) is also presented. It can be seen that for almost every metric, the selected method outperforms the original non-temporally-aware classifier with a high probability.
4. Discussion

We show that instance selection for the classification of the sounds in the training dataset offers better results than do cross-validation techniques. This is consistent with several other studies that have shown that selective learning helps reduce the effect of the noise in the data (Raman & Ioerger, 2003; Olvera-López et al., 2010). As has been addressed in (Borovicka et al., 2012) many instances in the training set may prove to be useless for classification purposes and they commonly do not improve the predictive performance of the model and may even degrade it.

Despite the noise in the data, certain researchers (Blum & Langley, 1997) have proposed a further two reasons for instance selection. The first reason arises when the learning algorithm is computationally intensive; in this case, if sufficient training data is available, it makes sense to learn from only a limited number of examples for purposes of computational efficiency. Another reason arises when the cost of labelling is high (e.g., when labels must be obtained from experts). In our case, the identification of the first and final frames of the ROIs is a burdensome task which can be minimized by using fewer examples in the training dataset.

Furthermore, from the results, the decision-tree method appears as one of the best classifiers in many temporally-aware methods. This fact is consistent with other studies where non-speech sounds (Pavlopoulos, Stasis & Loukis, 2004), or more specifically, environmental sounds (Bravo, Berrios & Aide, 2017) are considered.

Additionally, the temporally aware classifiers have revealed that they can outperform their non-temporally-aware counterparts. Several authors (Dietrich, Palm & Schwenker, 2003; Salamon et al., 2016) have reached similar results in the field of bioacoustics and argue that the constructed features can better capture spectro-temporal shapes that are representative of the various sound classes.

The sliding window method attained the best results in our tests, which is also consistent with other works (Salamon & Bello, 2015) that shows that feature learning is more effective when the learning is performed jointly on groups of frames. In their study the authors have reported very similar results for various window sizes. Our study, however, which has comprised a larger set of values for window size, concludes that there is an optimum region for the optimum number of frames and that overly large values of this parameter can even degrade the classification performance. The results attained by the sliding window method even outperformed the Hidden Markov Models usually employed in speech recognition applications. This result is mainly due to the fact that the HMM is a classifier that uses sub-word features, which are not suitable for non-speech sound identification since environmental sounds lack the phonetic structure that speech possesses (Cowling & Sitte, 2003). It has been found that the optimum classifier (SW3-11) increases the AUC by approximately 5 points and obtains a noteworthy overall accuracy of 90.5% (6 points higher than the baseline). Since the level of background noise in the recordings is high, this can be considered a remarkable result. In (Salamon & Bello, 2015) an increase of 1.5 points in the AUC and 5 points in accuracy were reported for an 8-frame window size.
The outperformance using these methods may only be moderate but it is reliably consistent. The probability that the selected temporally-aware methods improve their non-temporally-aware counterparts is extremely high (more than 90% in most cases).

Conversely, the cost of more complex computing due to the higher number of features required in the optimum SW3-11 method has been considered in detail (Luque et al., 2017). Using 33 (3 x 11) features almost double the number of the original 18 parameters which affects processing efforts in 3 different aspects. The first issue involves the time required for the construction of the new features that, for the 3-frames sliding window optimum method, is approximately 10 microseconds measured on a conventional desktop computer. This time is negligible compared to the classification time (detailed below) and to the frame length (10 ms, 1000 times higher).

Additionally employing a greater number of features leads to higher processing requirements in the task of training classifiers. By doubling the number of features, the time needed to train classifiers is also approximately doubled, with values of 30 ms for the minimum distance and of 800 ms for the decision tree. Although these values are greater than the 10 ms window length they have a limited effect on the overall classification process because classifiers are trained off-line only once and, therefore, they do not affect real-time performances.

A third issue regarding processing efforts is the effect of employing a greater number of features on classification times. Using 33 features instead of the original 18 parameters approximately increases the classification time of a frame by 5% with absolute values of approximately 2 ms for the minimum distance and 0.8 ms for the decision tree. Hence, a very limited rise in the computing effort is demanded when the temporally-aware methods are applied.

Another issue to be considered is the ability of the proposed method to identify when within each audio recording the call is located. It could be thought that sliding-window classifiers are going to blur the edges of audio events by using features obtained over a wider time span. However, these classifiers and certain other temporally-aware methods can still sharply identify the events. The sliding-window method features a frame considering preceding and subsequent frames, but it still independently classifies every frame, thereby allowing the precise identification of calls as has been shown in several independent studies (Mesaros et al. 2016; Stowell & Clayton, 2015; Foggia et al., 2015).

5. Conclusions

Changes in the sounds of anurans can be used as an indicator of climate change. Algorithms and tools for the automatic classification of the different classes of sounds could be developed for this purpose. In this paper, six different classification methods based on the data-mining domain have been proposed, which try to take advantage of the temporal behaviour of sound. The definition and comparison of this behaviour is undertaken using several approaches.
A detailed analysis of the classification errors shows that most errors occur when the recordings are very noisy. Additionally, other misclassifications appear when a recording, labelled as belonging to a certain class, is in fact made up of two or more overlapping sounds: one belonging to the true class and the others to a false class.

Firstly, it has been shown that instance selection for the determination of the sounds in the training dataset offers better results than do cross-validation techniques.

Additionally, the temporally-aware classifiers have revealed that they can obtain a better performance than their non-temporally-aware counterparts. The sliding window method attained the best results in our tests, and even outperformed the Hidden Markov Models usually employed in speech recognition applications.

For classifiers based on a given number of features, the optimization of the window size can increase the AUC value by up to 12 points (in %), while the optimization of the number of features only leads to an AUC increase of fewer than 3 points.

If the number of total features is of no great concern, then the optimum classifier for our dataset is based on 11 original features and a window with 3 frames (SW3-11), which increases the AUC by about 5 points and obtains a noteworthy overall accuracy of 90.5%: a result even more significant when one considers the high level of background noise affecting the sounds under analysis.

On the other hand, if the number of features has to be minimized due to low computing capacity then the optimization of the number of features in non-temporally-aware classifiers presents the best method, with an optimum for 11 features (NTA-11) thereby achieving an increase in the AUC of 3 points. If a further reduction in the number of features is required, a good compromise is found in the use of only 5 features (NTA-5) instead of the original 18, which reduces the number of parameters to less than one third while it reduces the AUC performance by only 3 points.

Acknowledgements
The authors would like to thank Rafael Ignacio Marquez Martínez de Orense (Museo Nacional de Ciencias Naturales) and Juan Francisco Beltrán Gala (Faculty of Biology, University of Seville) for their collaboration and support.

References
Aggarwal, C. C. 2007. Data streams: models and algorithms, 31. Springer Science and Business Media.
Aide, T. M., Corrada-Bravo, C., Campos-Cerqueira, M., Milan, C., Vega, G., & Alvarez, R. 2013. Real-time bioacoustics monitoring and automated species identification. *PeerJ*, 1, e103. DOI: 10.7717/peerj.103.

Akaike, H. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716-723. DOI: 10.1109/TAC.1974.1100705

Baum, L. E., & Eagon, J. A. 1967. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc*, 73(3), 360-363.

Blum, A. L., & Langley, P. 1997. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1), 245-271. DOI: 10.1016/S0004-3702(97)00063-5

Borovicka, T., Jirina Jr, M., Kordik, P., & Jirina, M. 2012. Selecting representative data sets. In *Advances in data mining knowledge discovery and applications*. InTech. DOI: 10.5772/50787

Box, G. E., Jenkins, G. M., Reinsel, G. C. 2011. *Time series analysis: forecasting and control* (Vol. 734). John Wiley and Sons.

Bravo, C. J. C., Berríos, R. Á., & Aide, T. M. 2017. Species-specific audio detection: a comparison of three template-based detection algorithms using random forests. *PeerJ Computer Science*, 3, e113. DOI: 10.7717/peerj-cs.113

Brookes, M. 2006. VOICEBOX: A speech processing toolbox for MATLAB. URL http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html. Available online.

Chawla, N. V. 2005. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook* (pp. 853-867). Springer US. DOI 10.1007/978-0-387-09823-4_45

Cover, T. M., Hart, P. E. 1967. *Nearest neighbour pattern classification*. Information Theory, IEEE Transactions on, 13(1), 21-27. DOI: 10.1109/TIT.1967.1053964

Cowling, M., & Sitte, R. 2003. Comparison of techniques for environmental sound recognition. *Pattern recognition letters* 24(15): 2895-2907. DOI: 10.1016/S0167-8655(03)00147-8.

Diaz, J. J., Nakamura, E. F., Yehia, H. C., Salles, J., & Loureiro, A. 2012. On the Use of Compressive Sensing for the Reconstruction of Anuran Sounds in a Wireless Sensor Network. In *IEEE International Conference on Green Computing and Communications* (GreenCom) (pp. 394-399). IEEE. DOI: 10.1109/GreenCom.2012.64.

Dietrich, C., Palm, G., & Schwenker, F. 2003. Decision templates for the classification of bioacoustic time series. *Information Fusion*, 4(2), 101-109.
Dietterich, T. G. 2002. Machine learning for sequential data: A review. In *Structural, syntactic, and statistical pattern recognition*, 15-30. Springer Berlin Heidelberg. DOI: 10.1007/3-540-70659-3_2.

Dobson, A. J., Barnett, A. 2008. *An introduction to generalized linear models*. CRC press.

Du, K. L., Swamy, M. N. S. 2013. *Neural Networks and Statistical Learning*. Springer Science and Business Media.

Efron, B., & Tibshirani, R. J. 1994. *An introduction to the bootstrap*. CRC press.

Esling, P., & Agon, C. 2012. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1): 12. DOI: 10.1145/2379776.2379788

Fay, R. R. (Ed.). 2012. *Comparative hearing: fish and amphibians*, 11. Springer Science & Business Media.

Foggia, P., Petkov, N., Saggese, A., Strisciuglio, N., & Vento, M. 2015. Reliable detection of audio events in highly noisy environments. *Pattern Recognition Letters*, 65, 22-28. DOI: 10.1016/j.patrec.2015.06.026.

Fonozoo.com. 2017. Retrieved from [http://www.fonozoo.com/](http://www.fonozoo.com/)

Gonzalez-Abril, L., Nuñez, H., Angulo, C., & Velasco, F. 2014. GSVM: An SVM for handling imbalanced accuracy between classes in bi-classification problems. *Applied Soft Computing*, 17, 23-31. DOI: 10.1016/j.asoc.2013.12.013

Gonzalez-Abril, L., Angulo, C., Nuñez, H., & Leal, Y. (2017). Handling binary classification problems with a priority class by using Support Vector Machines. *Applied Soft Computing*, 61, 661-669.

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (Eds.). 2006. *Feature extraction: foundations and applications* (Vol. 207). Springer.

Härdle, W. K., Simar, L. 2012. *Applied multivariate statistical analysis*. Springer Science and Business Media.

Hastie, T., Tibshirani, R., Friedman, J. 2005. *The elements of statistical learning: data mining, inference and prediction*. Springer-Verlag.

Herrera-Boyer, P., Peeters, G., & Dubnov, S. 2003. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1): 3-21.

Hevia, C. 2008. *Maximum likelihood estimation of an ARMA (p, q) model*. The World Bank, DECRG.
Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29(6): 82-97. DOI: 10.1109/MSP.2012.2205597.

ISO. 2001. *ISO/IEC 15938-4:2001 (MPEG-7: Multimedia Content Description Interface), Part 4: Audio*. ISO/IEC JTC 1.

Joshi, S. S. & Dietterich, T. G. 2003. *Calibrating recurrent sliding window classifiers for sequential supervised learning*. Oregon State University. Dept. of Computer Science. Retrieved from [http://hdl.handle.net/1957/31863](http://hdl.handle.net/1957/31863)

Kershenbaum, A., Blumstein, D. T., Roch, M. A., Akçay, Ç., Backus, G., Bee, M. A., ... & Coen, M. 2016. Acoustic sequences in non-human animals: a tutorial review and prospectus. *Biological Reviews*, 91(1), 13-52. DOI: 10.1111/brv.12160

Le Cam, L. 1990. Maximum likelihood: an introduction. *International Statistical Review/Revue Internationale de Statistique*, 153-171. DOI: 10.2307/1403464

Lin, J. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory*, 37(1): 145-151. DOI: 10.1109/18.61115.

Linde, Y., Buzo, A., & Gray, R. 1980. An algorithm for vector quantizer design. *IEEE Transactions on communications*, 28(1): 84-95. DOI: 10.1109/TCOM.1980.1094577

Liu, H., & Motoda, H. (Eds.). 1998. *Feature extraction, construction and selection: A data mining perspective* (Vol. 453). Springer Science & Business Media. ISBN 978-1-4615-5725-8

Llusia, D., Márquez, R., Beltrán, J. F., Benitez, M., Do Amaral, J. P. 2013. Calling behaviour under climate change: geographical and seasonal variation of calling temperatures in ectotherms. *Global change biology*, 19(9): 2655-2674. DOI: 10.1111/gcb.12267.

Luque, J., Larios, D. F., Personal, E., Barbancho, J., León, C. 2016. Evaluation of MPEG-7-Based Audio Descriptors for Animal Voice Recognition over Wireless Acoustic Sensor Networks. *Sensors*, 16(5): 717. DOI: 10.3390/s16050717.

Luque, A., Gómez-Bellido, J., Carrasco, A., Personal, E., & Leon, C. 2017. Evaluation of the Processing Times in Anuran Sound Classification. *Wireless Communications and Mobile Computing*, 2017. DOI: 10.1155/2017/8079846.

Luque, A., Romero-Lemos, J., Carrasco, A., & Barbancho, J. 2018. Non-sequential automatic classification of anuran sounds for the estimation of climate-change indicators. *Expert Systems with Applications*, 95, 248-260. DOI: 10.1016/j.eswa.2017.11.0.16
Márquez, R., Bosch, J. 1995. Advertisement calls of the midwife toads Alytes (Amphibia, Anura, Discoglossidae) in continental Spain. *Journal of Zoological Systematics and Evolutionary Research*, 33(3-4): 185-192. DOI: 10.1111/j.1439-0469.1995.tb00971.x.

Mesaros, A., Heittola, T., & Virtanen, T. 2016. TUT database for acoustic scene classification and sound event detection. In *Signal Processing Conference (EUSIPCO), 2016 24th European* (pp. 1128-1132). IEEE. DOI: 10.1109/EUSIPCO.2016.7760424.

Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., & Kittler, J. 2010. A review of instance selection methods. *Artificial Intelligence Review*, 34(2), 133-143. DOI 10.1007/s10462-010-9165-y.

Parascandolo, G., Huttunen, H., & Virtanen, T. 2016. Recurrent neural networks for polyphonic sound event detection in real life recordings. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on* (pp. 6440-6444). IEEE. DOI: 10.1109/ICASSP.2016.7472917

Pavlopoulos, S. A., Stasis, A. C., & Loukis, E. N. 2004. A decision tree–based method for the differential diagnosis of Aortic Stenosis from Mitral Regurgitation using heart sounds. *Biomedical engineering online*, 3(1), 21. DOI:10.1186/1475-925X-3-21

Powers, D. M. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63

Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2): 257-286. DOI: 10.1109/5.18626.

Raman, B., & Ioerger, T. R. 2003. Enhancing learning using feature and example selection. *Texas A&M University, College Station*, TX, USA.

Rokach, Lior, Maimon, O. 2008. *Data mining with decision trees: theory and applications*. World Scientific Pub Co Inc

Romero, J., Luque, A., Carrasco, A. 2016. Anuran sound classification using MPEG-7 frame descriptors. *XVII Conferencia de la Asociación Española para la Inteligencia Artificial (CAEP A)*, 801-810.

Salamon, J., & Bello, J. P. 2015. Unsupervised feature learning for urban sound classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on* (pp. 171-175). IEEE. DOI: 10.1109/ICASSP.2015.7177954.

Salamon, J., Bello, J. P., Farnsworth, A., Robbins, M., Keen, S., Klinck, H., & Kelling, S. 2016. Towards the automatic classification of avian flight calls for bioacoustic monitoring. *PloS one*, 11(11), e0166866. DOI: 10.1371/journal.pone.0166866.
Schaidnagel, M., Connolly, T., Laux, F. 2014. Automated feature construction for classification of time ordered data sequences. *International Journal on Advances in Software*, 7(3 and 4): 632-641.

Sokolova, M., Lapalme, G. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4), 427-437. DOI: 10.1016/j.ipm.2009.03.002

Stowell, D., & Clayton, D. 2015. Acoustic event detection for multiple overlapping similar sources. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on* (pp. 1-5). IEEE. DOI: 10.1109/WASPAA.2015.7336885.

Sturm, B. L. 2014. A simple method to determine if a music information retrieval system is a “horse”. *IEEE Transactions on Multimedia*, 16(6), 1636-1644. DOI: 10.1109/TMM.2014.2330697

Tzanetakis, G., & Cook, P. 2002. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing* 10(5): 293-302. DOI: 10.1109/TSA.2002.800560.

Vapnik, V. (1998). *Statistical learning theory*. 1998. Wiley, New York.

Wacker, A. G. & Landgrebe, D. A. 1971. *The minimum distance approach to classification*. Purdue University. Information Note 100771.

Wang, J. C., Wang, J. F., He, K. W., & Hsu, C. S. 2006. Environmental sound classification using hybrid SVM/KNN classifier and MPEG-7 audio low-level descriptor. In *IEEE International Joint Conference on Neural Networks IJCNN'06* 1731-1735. DOI: 10.1109/IJCNN.2006.246644.

Xie, J., Towsey, M., Zhu, M., Zhang, J., & Roe, P. (2017). An intelligent system for estimating frog community calling activity and species richness. *Ecological Indicators*, 82, 13-22.
Figure 1

General schema for the classification procedure.
1. Sound framing

2. Feature extraction of the frames

3.1. Feature construction of the frames

3.2. Segment modelling

4.1. Frame classification

4.2. Segment classification

5. Sound classification
Figure 2

Hidden Markov Model structure for each anuran call as proposed in Rabiner (1989).
Figure 3

Recurrent Neural Network structure.

![Recurrent Neural Network Diagram](image-url)
Figure 4

Cross-validation and instance selection ROC analysis for non-temporally-aware classifiers based on 18 MPEG-7 features.

| Classifier | xValid. (%) | Inst. Sel. (%) |
|------------|-------------|----------------|
| MinDis     | 71.10       | 83.98          |
| MaxLik     | 75.81       | 69.86          |
| DecTr      | 73.79       | 80.13          |
| kNN        | 62.59       | 62.94          |
| SVM        | 44.66       | 58.88          |
| LogReg     | 55.91       | 60.36          |
| Neur       | 55.52       | 63.17          |
| Discr      | 60.83       | 61.31          |
| Bayes      | 48.52       | 72.46          |
| Centroid   | 60.97       | 68.12          |
Figure 5

ROC analysis for non-temporally-aware classifiers based on 18 MPEG-7 features.
Figure 6

AUC vs. the number of features for the three best non-temporally-aware classifiers.

| MinDis | MaxLik | DecTr |
|--------|--------|-------|
| 1 59.60 % | 72.73 % | 59.12 % |
| 2 69.17 % | 71.06 % | 67.55 % |
| 3 64.20 % | 73.49 % | 68.28 % |
| 4 62.67 % | 70.71 % | 72.88 % |
| 5 74.13 % | 65.76 % | 78.74 % |
| 6 77.13 % | 67.94 % | 77.48 % |
| 7 77.84 % | 67.85 % | 79.36 % |
| 8 81.64 % | 72.07 % | 79.65 % |
| 9 81.61 % | **82.49 %** | 82.01 % |
| 10 83.53 % | 72.17 % | 80.83 % |
| 11 **85.87 %** | 75.14 % | 81.57 % |
| 12 85.65 % | 76.70 % | 80.81 % |
| 13 83.97 % | 75.94 % | 81.03 % |
| 14 83.97 % | 74.26 % | **82.17 %** |
| 15 83.86 % | 73.55 % | 80.48 % |
| 16 82.81 % | 72.73 % | 80.37 % |
| 17 84.21 % | 72.40 % | 80.47 % |
| 18 84.47 % | 71.34 % | 81.03 % |
Figure 7

ROC analysis for non-temporally-aware classifiers based on the 5 most relevant MPEG-7 features.

| Classifier | AUC  |
|------------|------|
| MinDis     | 75.02 % |
| MaxLik     | 66.42 % |
| DecTr      | 80.68 % |
| kNN        | 69.06 % |
| SVM        | 53.68 % |
| LogReg     | 55.53 % |
| Neur       | 61.70 % |
| Discr      | 53.70 % |
| Bayes      | 68.84 % |
Figure 8

AUC vs. the number of frames for several non-temporally-aware classifiers (5 features).
Figure 9

ROC analysis for non-temporally-aware classifiers using LIQR feature construction.

|       | AUC     |
|-------|---------|
| MinDis| 73.67 % |
| MaxLik| 78.09 % |
| DecTr | 85.22 % |
| kNN   | 58.80 % |
| SVM   | 59.11 % |
| LogReg| 63.02 % |
| Neur  | 62.47 % |
| Discr | 60.57 % |
| Bayes | 67.38 % |
Figure 10

ROC analysis for non-temporally-aware classifiers using the sliding window method.

| Classifier | AUC    |
|------------|--------|
| MinDis     | 74.97% |
| MaxLik     | 62.58% |
| DecTr      | 86.67% |
| kNN        | 49.81% |
| SVM        | 55.27% |
| LogReg     | 56.63% |
| Neur       | 53.65% |
| Discr      | 56.24% |
| Bayes      | 63.86% |
Figure 11

ROC analysis for non-temporally-aware classifiers using the recurrent sliding window method.

| Classifier | AUC   |
|------------|-------|
| MinDis     | 64.87%|
| MaxLik     | 50.00%|
| DecTr      | 72.74%|
| kNN        | 48.86%|
| SVM        | 50.00%|
| LogReg     | 39.27%|
| Neur       | 49.81%|
| Discr      | 42.31%|
| Bayes      | 50.00%|
Figure 12

ROC analysis for HMM and RNN classifiers.

|          | AUC   |
|----------|-------|
| HMM-F    | 61.70 % |
| HMM-ROI  | 59.25 % |
| HMM-SW   | 63.18 % |
| RNN      | 60.98 % |
Figure 13

AIC values for ROI segment patterns.
Figure 14

ROC analysis for temporally-aware classifiers using ARIMA models.
Figure 15

AUC values for temporally-aware methods (5 features).
Figure 16

ROC analysis for temporally-aware methods.

| Method   | AUC   |
|----------|-------|
| LIQR     | 85.22 %|
| SW       | 86.67 %|
| RSW      | 72.74 %|
| HMM-F    | 61.70 %|
| HMM-ROI  | 59.25 %|
| HMM-SW   | 63.18 %|
| RNN      | 60.98 %|
| ARIMA    | 61.98 %|
Figure 17

Colour map for the probability density function of the AUC vs. window size. Results obtained using bootstrap analysis of the sliding window method.
Figure 18

Probability density function of the AUC for the optimum sliding window case (with reduced number of features). Comparison to the original and reduced baselines.
Figure 19

AUC improvement for the sliding window method with reduced number of features. Comparison to the reduced (A) and original (B) baselines.
Figure 20

AUC vs. the number of frames for several non-temporally-aware classifiers (11 features).

|   | SW   | RSW   | HMM-SW | RNN  |
|---|------|-------|--------|------|
| 1 | 0.00%| 0.00% | 0.00%  | 0.00%|
| 2 | -    | -     | -      | -2.95%|
| 3 | 2.69%| 2.11% | -0.47% | 0.62%|
| 4 | -    | -     | -1.62% | -1.52%|
| 5 | 1.24%| -1.64%| -1.62% | -2.97%|
| 6 | -    | -     | -      | -3.58%|
| 7 | 2.09%| -0.19%| -1.78% | -1.21%|
| 8 | -    | -     | -      | -1.22%|
| 9 | 2.66%| -1.61%| -1.49% | -3.58%|
| 10| -    | -     | -      | -0.30%|
| 11| -3.25%| -1.32%| 1.74%  | -2.75%|
| 12| -    | -     | -      | -3.00%|
| 13| -6.43%| -16.32%| -0.56%| -3.79%|
| 14| -    | -     | -      | -2.51%|
| 15| -2.96%| -13.80%| -0.04%| -2.27%|
Figure 21

AUC values for temporally-aware methods (11 features).
Figure 22

ROC analysis for temporally-aware methods (11 features).

| Method   | AUC       |
|----------|-----------|
| LIQR     | 83.04 %   |
| SW       | 88.39 %   |
| RSW      | 88.39 %   |
| HMM-F    | 67.81 %   |
| HMM-ROI  | 62.21 %   |
| HMM-SW   | 60.42 %   |
| RNN      | 66.00 %   |
| ARIMA    | 69.34 %   |
Figure 23

Increasing the AUC values for the sliding window method with a varying number of features and window sizes.
Figure 24

Colour map of the increase in the AUC values for the sliding window method.
Figure 25

Impact of optimizing the AUC by selecting the window size or the number of features.
Figure 26

Increase of the AUC values for the sliding window method as a function of the dimension of feature space (D+C).
Figure 27

Probability density function of the AUC.
Figure 28

Probability density function of the ACC.
Figure 29

Box plot for each performance metric.
**Table 1** (on next page)

Sound and pattern datasets
| Sound class          | Sound Records | Pattern Records | Pattern Records |
|---------------------|---------------|----------------|----------------|
|                     | Number        | Seconds        | Number (pattern section) | Number (total record) |
| Ep. cal. mating call| 369 (43%)     | 1,853          | 4               | 13.89          | 20.39          |
| Ep. cal. release call| 63 (7%)       | 311            | 3               | 0.99           | 14.56          |
| Al. ob. mating call | 419 (48%)     | 2,096          | 4               | 1.09           | 19.72          |
| Al. ob. distress call| 17 (2%)       | 83             | 2               | 3.30           | 9.80           |
| Silence/Noise       | -             | -              | -               | 45.20          | -              |
| Total               | 868           | 4,343          | 13              | 64.47          | 64.47          |
Table 2 (on next page)

MPEG-7 features and the processes for their extraction.
| Feature description | Extracting process |
|---------------------|--------------------|
| 1 Total power       | Spectrogram analysis |
| 2 Relevant power    |                     |
| (power in a certain frequency band) |                     |
| 3 Power centroid    |                     |
| 4 Spectral dispersion |                   |
| 5 Spectrum flatness |                     |
| 6, 7, 8 Frequency of the formants (x3) |   |
| (The first three formants are considered) |  |
| 9, 10, 11 Bandwidth of the formants (x3) |   |
| (The first three formants are considered) |  |
| 12 Pitch            | Linear prediction coding (LPC) |
| 13 Harmonic centroid |                     |
| 14 Harmonic spectral deviation |                |
| 15 Harmonic spectral spread |               |
| 16 Harmonic spectral variation |              |
| 17 Harmonicity ratio | Harmonicity analysis |
| 18 Upper limit of harmonicity |                  |
Table 3 (on next page)

MATLAB functions supporting the various classifiers.
| Classif. | Training function | Test function | Additional function |
|---------|-------------------|---------------|---------------------|
| MinDis  | fitgmdist         | -             | mvnpdf              |
| MaxLik  | -                 | -             |                     |
| DecTr   | fitctree          | predict       |                     |
| kNN     | fitcknn           | predict       |                     |
| SVM     | fitcsvm           | predict       |                     |
| LogReg  | mnrfit            | mnrval        |                     |
| Neur    | feedforwardnet;   | net           |                     |
|         | train             |               |                     |
| Discr   | fitcdiscr         | predict       |                     |
| Bayes   | fitNaiveBayes    | posterior     |                     |
| HMM     | hmmtrain          | hmmdecode     | kmeanlbg; disteusq |
|         |                   |               | aicbic              |
| ARIMA   | vgxset;           | NTA           |                     |
|         | vgxvarx           | classifiers   |                     |
| RNN     | layrecnet; train  | net           |                     |
Table 4 (on next page)

Confusion matrix.
| Classification class | Classified as positive | Classified as negative |
|----------------------|------------------------|------------------------|
| Data class           | Positive               | FN (false negative)    |
|                      | Negative               | FP (false positive)    |

TP (true positive)  
FN (false negative)  
FP (false positive)  
TN (true negative)
Table 5 (on next page)

Classification performance metrics based on the confusion matrix.
| Metric               | Formula                       | Evaluation focus                                                                 |
|---------------------|-------------------------------|----------------------------------------------------------------------------------|
| Accuracy            | $\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN}$ | Overall effectiveness of a classifier                                             |
| Precision           | $\text{PRC} = \frac{TP}{TP + FP}$                               | Class agreement of the data labels with the positive labels given by the classifier |
| Sensitivity         | $\text{SNS} = \frac{TP}{TP + FN}$                               | Effectiveness of a classifier to identify positive labels. Also called True Positive Rate (TPR) |
| Specificity         | $\text{SPC} = \frac{TN}{TN + FP}$                               | How effectively a classifier identifies negative labels. Also called True Negative Rate (TNR) |
| $F_1$ score         | $F_1 = 2 \frac{\text{PRC} \cdot \text{SNS}}{\text{PRC} + \text{SNS}}$ | Combination of precision ($\text{PRC}$) and sensitivity ($\text{SNS}$) in a single metric |
| Geometric Mean      | $\text{GM} = \sqrt{\text{SNS} \cdot \text{SPC}}$             | Combination of sensitivity ($\text{SNS}$) and specificity ($\text{SPC}$) in a single metric |
| Area Under (ROC) Curve | $\text{AUC} = \int_0^1 \text{SNS} \cdot d\text{SPC}$ | Combined metric based on the Receiver Operating Characteristic (ROC) space (Powers, 2011) |
Table 6 (on next page)

Summary of performance metrics (5 features).
| Method                  | Features | Frames | Best Classifier | ACC  | PRC   | SNS  | SPC  | F₁   | GM   | AUC  |
|------------------------|----------|--------|-----------------|------|-------|------|------|------|------|------|
| NTA-18 (original baseline) | 18       | -      | MinDis          | 84.12| 55.37 | 76.95| 90.03| 64.40| 83.23| 83.49|
| NTA-5 (reduced baseline)   | 5        | -      | DecTr           | 86.82| 76.45 | 72.60| 88.77| 74.47| 80.28| 80.68|
| LIQR                   | 10 (2·5) | -      | DecTr           | 91.88| 79.25 | 77.49| 92.94| 78.36| 84.86| 85.22|
| SW                    | 35 (7·5) | 7      | DecTr           | 92.59| 74.11 | 79.07| 94.28| 76.51| 86.34| 86.67|
| RSW                   | 55 (11·5)| 11     | DecTr           | 83.41| 57.74 | 58.52| 86.96| 58.13| 71.34| 72.74|
| HMM-F                 | 5        | -      |                 | 75.41| 44.64 | 40.62| 82.78| 42.53| 57.99| 61.70|
| HMM-ROI               | 5        | -      |                 | 71.88| 44.87 | 42.69| 75.80| 43.75| 56.88| 59.25|
| HMM-SW                | 5        | 1      |                 | 72.35| 47.08 | 45.09| 81.26| 46.06| 60.53| 63.18|
| RNN                   | 75 (15·5)| 15     |                 | 66.59| 47.81 | 40.01| 81.91| 43.58| 57.27| 60.98|
| ARIMA                 | 75 (3·5) | -      | DecTr           | 80.94| 38.75 | 38.47| 85.50| 38.61| 57.35| 61.98|
Table 7 (on next page)

Performance improvement of the sliding window method (5 features, 7 frames).
| Performance improvement | ACC  | PRC  | SNS  | SPC  | F1  | GM   | AUC  |
|-------------------------|------|------|------|------|-----|------|------|
| Baseline NTA-5          |      |      |      |      |     |      |      |
| Mean                    | 5.77 | -2.33| 6.50 | 5.51 | 2.06| 6.08 | 6.01 |
| Conf. Int.              | ±2.70| ±10.3| ±12.3| ±2.08| ±9.94| ±7.38| ±6.72|
| Baseline NTA-18         |      |      |      |      |     |      |      |
| Mean                    | 8.46 | 18.73| 2.20 | 4.25 | 12.12| 3.15 | 3.22 |
| Conf. Int.              | ±2.82| ±6.94| ±11.9| ±1.93| ±7.98| ±7.05| ±6.51|
Table 8 (on next page)

Summary of performance metrics (11 features).
| Method          | Features | Frames | Best Classifier | ACC | PRC | SNS | SPC | F₁ | GM | AUC   |
|-----------------|----------|--------|-----------------|-----|-----|-----|-----|----|----|-------|
| NTA-18 (original baseline) | 18       | -      | MinDis          | 84.12 | 55.37 | 76.95 | 90.03 | 64.40 | 83.23 | 83.49 |
| NTA-11 (reduced baseline)    | 11       | -      | MinDis          | 88.00 | 77.67 | 81.03 | 92.50 | 79.31 | 86.58 | 86.77 |
| LIQR            | 22 (2·11) | -      | DecTr           | 89.77 | 75.89 | 74.59 | 91.49 | 75.23 | 82.61 | 83.04 |
| SW              | 33 (3·11) | 3      | MinDis          | 90.47 | 79.30 | 82.85 | 93.93 | 81.03 | 88.21 | 88.39 |
| RSW             | 33 (3·11) | 3      | MinDis          | 90.47 | 79.30 | 82.85 | 93.93 | 81.03 | 88.21 | 88.39 |
| HMM-F           | 11       | -      | -               | 78.24 | 21.77 | 50.42 | 85.20 | 48.13 | 65.54 | 67.81 |
| HMM-ROI         | 11       | -      | -               | 72.59 | 85.96 | 48.39 | 76.03 | 61.92 | 60.66 | 62.21 |
| HMM-SW          | 11       | 11     | -               | 75.29 | 45.56 | 37.02 | 83.82 | 39.16 | 55.70 | 60.42 |
| RNN             | 33 (3·11) | 3      | -               | 71.06 | 48.69 | 47.44 | 84.56 | 48.06 | 63.34 | 66.00 |
| ARIMA           | 363 (3·11²) | -     | MinDis          | 89.29 | 48.03 | 47.88 | 90.81 | 47.96 | 65.94 | 69.34 |
Table 9 (on next page)

Performance improvement of the sliding window method (11 features, 3 frames).
| Performance improvement | ACC  | PRC  | SNS  | SPC  | F₁   | GM   | AUC  |
|-------------------------|------|------|------|------|------|------|------|
| Baseline NTA-5          | Mean | 2.47 | 1.64 | 1.85 | 1.43 | 1.74 | 1.66 | 1.64 |
|                         | ±2.76| ±3.76| ±11.4| ±1.76| ±6.63| ±6.62| ±6.20|
| Baseline NTA-18         | Mean | 6.36 | 23.94| 6.06 | 3.91 | 16.70| 5.08 | 4.98 |
|                         | ±2.94| ±5.28| ±11.6| ±1.85| ±7.02| ±6.71| ±6.25|
Table 10 (on next page)

Summary of the best classification methods.
| Method   | Best classifier | Features | Temporally aware | Optimization | Features concern |
|----------|-----------------|----------|------------------|--------------|-----------------|
| NTA-18   | MinDis          | 18       | No               | No           | Original        |
| NTA-5    | DecTr           | 5        | No               | No           | Balanced        |
| NTA-11   | MinDis          | 11       | No               | Only features| No              |
| SW7-5    | DecTr           | 35 (7·5) | Yes              | Separate     | Balanced        |
| SW3-11   | MinDis          | 33 (3·11)| Yes              | Separate     | No              |
| SW3-5    | DecTr           | 15 (3·5) | Yes              | Joint        | Balanced        |
| SW7-8    | DecTr           | 56 (7·8) | Yes              | Joint        | No              |
Table 11 (on next page)

Performance improvement (%) over the original baseline (NTA-18).
| Method  | Features | Statistic | ACC  | PRC   | SNS   | SPC   | F_L | GM   | AUC   |
|---------|----------|-----------|------|-------|-------|-------|-----|------|-------|
| NTA-5   | 5        | Mean      | 2.69 | 21.06 | -4.31 | -1.27 | 10.06 | -2.94 | -2.79 |
| NTA-11  | 11       | Mean      | 3.89 | 22.30 | 4.21  | 2.48  | 14.96 | 3.42  | 3.34  |
| SW3-5   | 15 (3·5) | Mean      | 7.05 | 20.60 | -2.13 | 2.44  | 10.95 | -0.06 | 0.16  |
| SW3-11  | 33 (3·11)| Mean      | 6.36 | 23.94 | 6.06  | 3.91  | 16.70 | 5.08  | 4.98  |
| SW7-5   | 35 (7·5) | Mean      | 8.46 | 18.71 | 2.20  | 4.25  | 12.12 | 3.15  | 3.22  |
| SW7-8   | 56 (7·8) | Mean      | 9.29 | 28.55 | 3.83  | 4.32  | 17.87 | 4.07  | 4.07  |
**Table 12** (on next page)

Performance improvement (%) over the original baseline (NTA-18) with confidence interval.
| Method  | Features | Statistic | ACC   | PRC   | SNS   | SPC   | F₁    | GM    | AUC  |
|---------|----------|-----------|-------|-------|-------|-------|-------|-------|------|
| NTA-11  | 11       | Mean      | 3.89  | 22.30 | 4.21  | 2.48  | 14.96 | 3.42  | 3.34 |
|         |          | ±2.94     | ±5.12 | ±11.5 | ±1.90 | ±6.94 | ±6.72 | ±6.29 |
|         |          | Pr. Outperf. | 99.42 | 99.83 | 77.52 | 99.38 | 99.82 | 85.22 | 86.23 |
| SW3-11  | 33 (3·11)| Mean      | 6.36  | 23.94 | 6.06  | 3.91  | 16.70 | 5.08  | 4.98 |
|         |          | ±2.94     | ±5.28 | ±11.6 | ±1.85 | ±7.02 | ±6.71 | ±6.25 |
|         |          | Pr. Outperf. | 100  | 99.97 | 86.08 | 100  | 99.91 | 93.81 | 94.56 |
| SW7-8   | 56 (7·8) | Mean      | 9.29  | 28.55 | 3.83  | 4.32  | 17.87 | 4.07  | 4.07 |
|         |          | ±2.71     | ±10.3 | ±12.0 | ±1.91 | ±9.01 | ±7.07 | ±6.54 |
|         |          | Pr. Outperf. | 100  | 100  | 74.67 | 99.99 | 99.86 | 88.02 | 89.51 |