Automatic Acoustic Target Detection and Classification off the Coast of Portugal

Razi Sabara
LARSyS
University of Algarve
8005-139 Faro, Portugal

Cristiano Soares
MarSensing Lda.
8005-139 Faro, Portugal

Friedrich Zabel
MarSensing Lda.
8005-139 Faro, Portugal

José V. Oliveira
Universidade do Algarve
8005-139 Faro, Portugal

Sérgio M. Jesus
LARSyS
Universidade do Algarve
8005-139 Faro, Portugal

***Abstract*—Ocean noise has been a topic of research for many years, for its impact in sonar detection, underwater communications and ocean acoustic observation in general. Recently, ocean sound has been designated as an Essential Ocean Variable (EOV) and is therefore, becoming increasingly recorded and monitored, along with other oceanic and meteorological variables. The research projects EMSO-PT and SUBECO aim at deploying ocean observatories along the coast of Portugal for long term ocean variables monitoring, among which ocean sound. Unlike other ocean variables, ocean sound allows for feature detection, characterisation and possibly identification with known patterns. This work shows the results obtained with current machine learning algorithms for feature detection and extraction on a two days recording of ocean noise obtained on a offshore buoy deployed under the SUBECO project, on the west coast of Portugal. Preliminary results show the possibility of improved event detection, followed by classification and clustering, that foresee a rapid and accurate analysis of large observatory acquired acoustic data sets.**

***Index Terms*—ocean sound, MSFD, machine learning, acoustic detection**

I. Introduction

In underwater acoustics there is a clear distinction between active acoustics, that encompasses transmitting and receiving sound, and passive acoustics which involves sound listening only. Recently, ocean sound has been designated as an Essential Ocean Variable (EOV) and is, therefore, becoming increasingly recorded and monitored, along with other oceanic and meteorological variables. Unwanted ocean sound is termed as noise. Ocean noise has been a topic of research for many years, for its impact in sonar detection, underwater communications and ocean acoustic observation in general.

There is significant evidence that the mean ocean noise level in the ocean has been steadily increasing in the last decades, mostly due to shipping [1], correlated with economic globalization [2]. The noise sources responsible for this increase are clearly identified: ship traffic, offshore industrial construction and seismic oil & gas exploration. The European Union was the first legal body to address the issue of ocean noise and its role on the Good Environmental Status (GES), through the Marine Strategic Framework Directive (MSFD) in 2010 [3].

Funded by projects SUBECO and EMSO-PT.

The research projects EMSO-PT[1] and SUBECO[2] aim at deploying ocean observatories along the coast of Portugal for long term ocean variables monitoring, including ocean sound, and thus for supporting the MSFD. The SUBECO project includes deploying a network of multi-parametric offshore buoys which preferred location is within or close by the ship Traffic Separation System (TSS) along the west and south coast of Portugal. These buoys have recently been fitted with arrays of continuously recording broadband acoustic receivers. It is therefore expected that the received acoustic field will be dominated by shipping noise as a mixture of both short and long range ships. The whole data are safeguarded on the buoy and only downloaded during buoy maintenance which takes place approximately once every six months. At those moments data are analyzed for noise level statistics and trends detection, if any. Besides estimating noise levels, trends and GES risk, passive acoustics also allows for identifying characteristic acoustic patterns that may be classified according to their origin that can be either natural (waves, wind, rain, ice and earthquakes), biological (marine mammals, fish, invertebrates,..) or man-made (shipping, industrial acitivity, seismic exploration, etc). It is therefore of paramount importance to perform routine identification of various acoustic signatures present in the data such as marine mammals vocalization traces, AIS and non AIS correlated shipping, fast surface vessels possibly associated with smuggling activities, and any other underwater unknown sounds. This work focuses on the post processing of the archived data for automatic analysis including event detection, identification and classification using machine learning techniques.

Our main contribution is at the feature extraction level. Specifically here we propose to use deep autoencoders evolved with genetic algorithms in order to automatically extract patterns of interest from the underwater acoustic recordings.

This paper is organized as follows: section II describes the experimental setup used for real data acquisition; section III describes the methodology for signal detection and processing algorithms; section IV describes and discusses the results

[1]European Multidisciplinary Seafloor Observatories - Portugal, funded by FCT (contract 022157).

[2]SUB-ECO Acoustic Surveillance System, funded under contract of PO-Navy (2015-2019).

obtained on the test data set; finally section V draws some conclusions of the work done so far.

II. SUBECO BUOY TEST EXPERIMENT

Figure 1 shows an artistic drawing of the SUBECO buoy. The floating device is provided by Fugro-Oceanor model Seawatch[3] with its meteorological and environmental sensors. For the SUBECO project the buoy has been fitted with an acoustic array formed by 6 broadband hydrophones at the vertices of a tetrahedron of 1.5 m side (not to scale in the figure). The

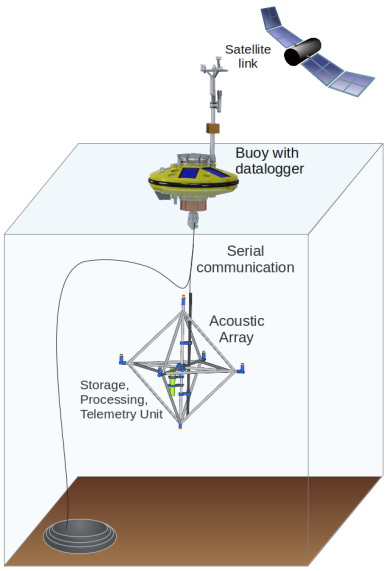

Fig. 1: SUBECO buoy artistic drawing (not to scale).

array is deployed at the end of an umbilical electromechanical cable at 80 m depth by design. The acquired data is locally stored and processed, while snippets of data and system status are cabled to the surface buoy allowing for remote monitoring and data inspection through a remote satellite link. The design, installation and testing of the acoustic recording system was provided by Marsensing Lda. (Faro, Portugal). For testing purposes, a first SUBECO buoy prototype was deployed off the coast of Portugal between mid April and mid May, 2019. Figure 2 shows the deployment location on top of a ship traffic density map using AIS data over one year back in 2014. As it can be seen the selected location is close to the ship TSS lanes, running North-South along the Portuguese coast. The water depth at the deployment location is approximately 1330 m. Data recordings were performed at three sampling frequencies and for three different duration: 60 seconds at 10 kHz, 30 seconds at 50 kHz and 10 seconds at 100 kHz and stored in separate WAV files. This sequence of 1 minute and 40 s duration is repeated with a duty cycle of 10 minutes. The data set analyzed in this work covers two

[3]for buoy model details see https://www.fugro.com/about-fugro/our-expertise/technology/seawatch-metocean-buoys-and-sensors

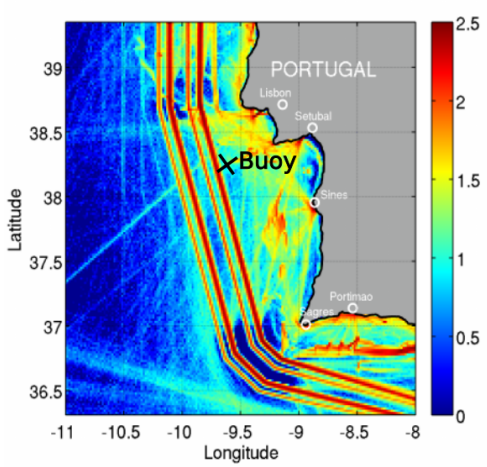

Fig. 2: AIS ship density map with SUBECO buoy deployment location.

days of data from May 5 to 7, 2019, so approximately 280 data records of 30 seconds at 50 kHz form the target data set. An overview of the data was obtained through a calibrated power spectrum density (PSD) spectrogram analysis for all the 280 data blocks. Spectrograms used the Welch periodogram with a 8k block size short time Fourier transform with 50% overlap and hamming windowing for higher sidelobe supression and estimate stabilization.

Examples are shown in Fig. 3 for (believed to be) opportunity dolphins' vocalizations (a) and a slow passing by vessel (b). Together with ocean sound features of interest to this

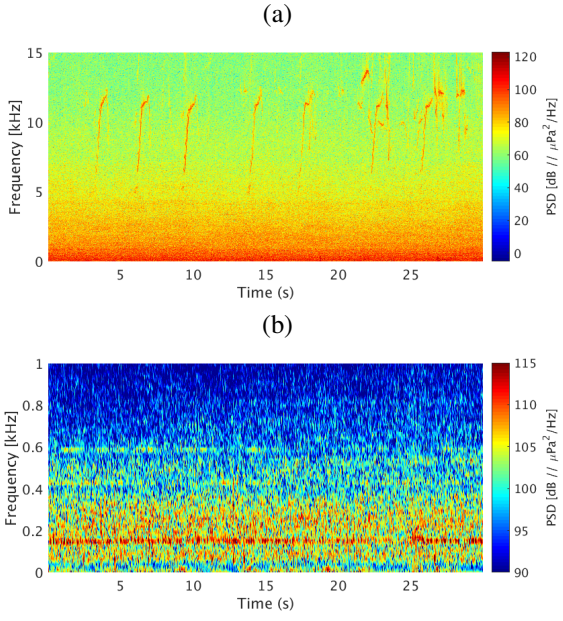

Fig. 3: spectrograms of selected acoustic data as example bioacoustics (a) and ship noise (b).

study, the data sets also contain numerous impulsive sounds, characterized by a large frequency band and high intensity

(see an example in Fig. 4). These are generated on the buoy mooring itself due to tension on cables and shackles in relation to surface waves and ocean currents. It was found that these interference are correlated with weather conditions (wind and sea state) on site. Such interference are common in offshore sound recording systems and are usually termed as self-noise or pseudo-noise [4]. In this study self-noise was treated as a

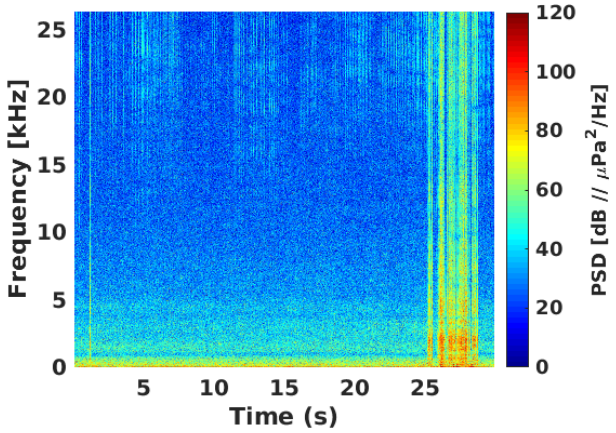

Fig. 4: Illustration of mooring noise spectrogram.

category of sounds to be detected and identified, along with other categories of interest.

III. Methodology and Signal Analysis

The architecture used for acoustic data processing is described in Fig. 5.

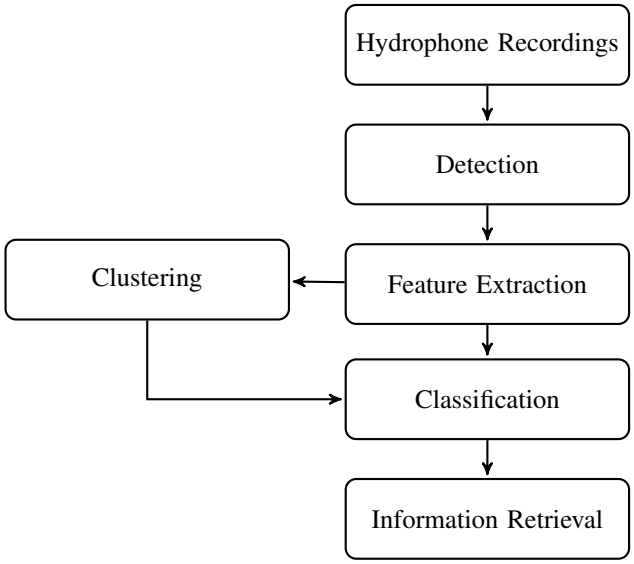

Fig. 5: Passive acoustic monitoring signal analysis methodology.

Each of the various blocks on the diagram of Fig. 5 perform a specific operation on the data in order to retrieve valuable information. For instance the goal of the detection stage is to identify events of interest. At the feature extraction level various mathematical rules were applied in order to capture particular signal characteristics. This block is very important since it controls learning algorithms performance. The following sections briefly describe each one of these blocks.

A. Signal detection

Signal detection is a crucial step within a passive acoustic monitoring system. The detector should correctly time-pinpoint relevant signals with a low false alarm rate. For a time-varying underwater channel, the observation $y(n)$ may be given as

$$y(n) = \begin{cases} \eta(n) & \mathcal{H}_0 \\ x(n) + \eta(n), & \mathcal{H}_1 \end{cases} \quad (1)$$

where $\eta(n)$ is the diffuse ocean environmental sound (rain, waves, bubbles,..) and $x(n)$ represents the signal of interest which could be of anthropogenic (ships, offshore activity,..), biological (marine mammals,..) or geological (seismic,..) origin. $\mathcal{H}_1$ denotes the hypothesis under which an event of interest is present and $\mathcal{H}_0$ otherwise.

The adopted detector was based on the energy detector which decision statistic is given as

$$\mathbb{T}_{ED} = \sum_{n=1}^{N} |y[n]|^2, \quad (2)$$

where N denotes the window frames within a recording. The decision statistic is compared against a threshold ζ and the detection is considered positive if the statistic is larger than the threshold and negative otherwise. The energy detector is known to be robust for the detection of random signals in white noise [5].

B. Feature extraction and dimensionality reduction

Directly feeding raw audio signals in learning algorithms usually require a long time for processing and the obtained results are often poor. A practical approach consists in extracting relevant features before training the algorithms.

A feature could be defined as a particular characteristic or description of the data. In order to successfully perform the clustering and classification tasks, a set of relevant features are required. Specifically, this set captures the hidden patterns within the data such as harmonic structure, pitch or energy distribution among others.

We have considered extracting multiple features from the time domain (RMS, ZCR), frequency domain (MFCC, Centroid, Chroma, Rolloff, Contrast, Tonnetz), as well as the time-frequency domain (Mel-frequency Spectrogram) in order to capture different information from the raw data. These are briefly described below.

- **Tonnetz** The harmonic network or Tonnetz [6] detect harmonic changes by using a mapping from 12-bin chroma vectors to the interior space of a 6-D polytope. Concretely the six dimensional tonal centroid vector is given by the multiplication of the chroma vector with the transformation matrix Φ.

$$\text{Tonnetz[d]} = \frac{1}{|c|}\sum_{l=0}^{11}\Phi(d,l)c(l) \tag{3}$$

∗**Mel-frequency Spectrogram** The mel scale was proposed to describe the non-linearity in pitch perception [7]. The mapping from the Hertz scale to the Mel scale is based on the following

$$\text{Mel[f]} = \frac{1000}{\log 2}\log(1+\frac{f}{1000}) \tag{4}$$

∗**ZCR** The intuition behind Zero-Crossing Rate features consists in evaluating the frequency content of a signal by measuring its rate of change from positive to negative [8]. Concretely this is performed using the following formula

$$\text{ZCR} = \frac{1}{2}\sum_{n=1}^{N}|\text{sign}(y[n]) - \text{sign}(y[n-1])| \tag{5}$$

∗**Rolloff** The key insight behind Rolloff features relies on the measure of spectrum magnitude concentration [9]. They were at first proposed to detect the presence of speech.

$$\text{Rolloff} = \sum_{n=0}^{N}|Y_r[n]| \tag{6}$$

where $Y_r[n]$ represents the STFT of frame r.

∗**MFCC** Mel frequency cepstrum coefficients [10] have been particularly successful in speech processing and audio tasks among others. The calculation process of MFCC starts by computing the logarithm of mel-band energies then applying the discrete cosine transform (DCT) to decorrelate the overlapping filter-banks.

∗**Contrast** Octave-based spectral contrast features [11] have been proposed to represent the relative spectral characteristics of audio signals. Concretely this is performed by considering the strength of spectral peaks and spectral valleys in each sub-band separately.

$$\text{Contrast}_k = \text{Peak}_k - \text{Valley}_k \tag{7}$$

$$\text{Peak}_k = \log(\frac{1}{N}\sum_{i=1}^{N}y_k[i])$$

$$\text{Valley}_k = \log(\frac{1}{N}\sum_{i=1}^{N}y_k[N-i+1])$$

∗**RMS** Root Mean Square represents the time domain envelope within which the signal is contained. It is calculated by considering the square root of the average power

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|y[n]|^2} \tag{8}$$

∗**Centroid** The spectral centroid represents the spectrum center of gravity [12]. It is calculated as the following

$$\text{Centroid}_r = \frac{\sum_{n=1}^{N/2}f[n]|Y_r[n]|}{\sum_{n=1}^{N/2}|Y_r[n]|} \tag{9}$$

where $f[n] = \frac{nf_s}{N}$ is the frequency at bin n and $Y_r[n]$ is the STFT of frame r.

∗**Chroma** The goal of chroma is to capture harmonic informations [13]. It is calculated by summing the log-frequency magnitude spectrum over the octaves. The chroma vectors are given by

$$\text{Chroma[b]} = \sum_{z=0}^{Z-1}Y_{lf}|b+z\beta| \tag{10}$$

where Y_{lf} represents the log-frequency spectrum, b the pitch class index, z the octave index and β the bins per octave.

For dimensionality reduction we used both PCA – principal component analysis – and convolutional autoencoders. These are briefly described next.

1) Principal Component Analysis: Principal Component Analysis [14] is a non-parametric technique which has been successfully applied in numerous applications ranging from outlier removal to data compression. This particular property is mostly interesting when the data matrix X lies in a high dimensional space. To that end PCA could efficiently get the best low rank approximation of the data. Concretely, this is performed by a mapping to a subspace in which the components with highest variance are retained and the remaining ones discarded. The PCA problem is formulated as the following

$$\begin{aligned}&\text{maximize} \quad \text{Var}(\mathbf{w}^T\mathbf{x}) = \mathbf{w}^T\Sigma\mathbf{w}\\ &\text{subject to} \quad\quad ||\mathbf{w}|| = 1,\end{aligned} \tag{11}$$

where $\Sigma = E[(x-\mu)(x-\mu)^T]$. Henceforth, the first and second principal components $||\mathbf{w}_1||$ and $||\mathbf{w}_2||$ are found by solving

$$\begin{aligned}&\underset{\mathbf{w}_1}{\text{argmin}}\ \mathbf{w}_1^T\Sigma\mathbf{w}_1 - \alpha(\mathbf{w}_1^T\mathbf{w}_1 - 1)\\ &\underset{\mathbf{w}_2}{\text{argmin}}\ \mathbf{w}_2^T\Sigma\mathbf{w}_2 - \alpha(\mathbf{w}_2^T\mathbf{w}_2 - 1) - \beta(\mathbf{w}_2^T\mathbf{w}_1 - 0),\end{aligned} \tag{12}$$

where α and β are Lagrange multipliers. Under the condition where $\mu = 0$, the covariance matrix reduces to $\Sigma = X^TX$.

2) Convolutional Autoencoders: Convolutional Autoencoders (CAE) are composed by different types of layers which perform specific operations to their input. The first block of a CAE is called the Convolution Network or Encoder and it includes Convolutions and Max-Pooling layers while the second block is denoted the Deconvolution Network or Decoder and is composed of Deconvolutions and Upsampling layers. Using Max-Pooling layers allows to reduce the resolution of the grid space while preserving the most important informations. The output of a Max-Pooling operation is a layer of lower dimension. Activation functions such as the ReLU, Tanh or the logistic function are usually applied at the output for non linear decision boundaries.

Training of CAE is usually performed using the backpropagation algorithm, associated with the following cost function

$$\mathfrak{L}(\mathbf{x},\hat{\mathbf{x}}) = ||\mathbf{x}-\hat{\mathbf{x}}||_2^2. \tag{13}$$

At the encoder the input $\mathbf{x}$ is compressed into a lower representation denoted $\mathbf{z} = f(\mathbf{W}_e * \mathbf{x} + \mathbf{b}_e)$, where $*$ is the convolution operation and $(\mathbf{W}_e, \mathbf{b}_e) = (W^1, b^1, \ldots, W^l, b^l)$ are the parameters associated with the l layers of the encoding network. During this process irrelevant features disappear and the dimensionality of the feature vector is reduced. At the decoder, the output $\hat{\mathbf{x}} = g(\mathbf{W}_d * \mathbf{z} + \mathbf{b}_d)$ tries the reconstruct the input from the bottleneck layer.

Representation learning is performed using filters convolved with the feature maps, using the following formula

$$f[x, y] * g[x, y] = \sum_i \sum_j f[i, j] g[x - i, y - j], \tag{14}$$

where f represents the input image, g the filter, x and y the horizontal and vertical axis.

Concretely, the filter is placed at the top left corner of the image then shifted to the right by one pixel until it reaches the right corner, afterwards it moves down. This process is repeated until the filter reaches the bottom right corner of the image. At each step, the values at each spatial region of the filter and feature map are multiplied element wise then summed. The result of this operation produces a new feature map. In order to learn different kinds of features, the depth of the feature map is increased by using multiple filters.

C. Clustering

The main goal of clustering it to divide an unlabelled heterogeneous data set into a partition of homogeneous subgroups (or clusters). Although well-known we briefly describe for easy reference the algorithms employed in this work.

1) Gaussian Mixture Models: Gaussian Mixture Models (GMM) is a probabilistic method which assumes that the data $\mathbf{x} \in \mathcal{R}^d$ is drawn from a mixture of gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ weighted by π_k in order to build predictions around the number of clusters k.

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{15}$$

$$\text{subject to} \quad \sum_{k=1}^{K} \pi_k = 1$$

$$p_k(\mathbf{x}_i | \theta_k) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k)\right).$$

The learning process of GMM relies on the iterative algorithms Expectation-Maximization (EM) or Maximum A Posteriori (MAP) in order to find the parameter vector $\boldsymbol{\theta}_k = (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

2) Hierarchical Agglomerative Clustering: Ward based Hierachical Agglomerative Clustering (HAC) [15] is a powerful tool for analysing and identifying similar groups within large data sets. Concretely, the clusters are built by iteratively merging the clusters G_u and G_v with minimal linkage δ.

$$\delta(G_u, G_v) = \frac{|G_u||G_v|}{|G_u| + |G_v|} ||x_{G_u} - x_{G_v}||^2, \tag{16}$$

where $x_G = \frac{1}{n} \sum_{i=1}^{n} x_i$ is the center of gravity of G. The results produced by HAC could be displayed by dendrograms which have an inverse tree structure. In order to find the number of clusters using the dendrogram we perform a horizontal cut at a specific height.

3) k-means: k-means is one of the simplest algorithms for clustering which has been successfully applied in numerous fields ranging from computer vision to medical imaging and speech recognition among others. The underlying principle behind k-Means consists in finding k clusters centroids $\mu_j (j = 1, \ldots, k)$ using an iterative approach which starting from initial centroids guesses update them until the distance between the data points and their respective centroid is minimized.

D. Classification

1) Support Vector Machines: Support Vector Machines (SVM), introduced by Vapnik [16], implement a discriminative classification algorithm which aims to separate the data instances $\{(\mathbf{x}_i, y_i)\ i = 1, \ldots, m; \mathbf{x}_i \in \mathcal{R}^d\}$ belonging to different classes $y_i \in \{+1, -1\}$ using a hyperplane $\mathbf{w}^T \mathbf{x}_i + b$ such that the margin which separates the nearest data points is maximized. The decision boundary could be found by solving the following constrained optimization problem

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{m} \xi_i \quad \text{(cost function)} \tag{17}$$

$$\text{subject to} \quad \xi_i \geq 0, \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{for i=1,...,m}$$

Introducing the Lagrange multipliers α_i, the constrained optimization problem could be formulated as a dual optimization problem yielding

$$\text{maximize} \quad \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to} \quad \alpha_i \geq 0, \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \quad \text{for i=1,...,m}$$

If the data points are not linearly separable, the kernel trick allows to transform the original problem into a linear one by mapping data into a higher dimensional space. Concretely this is performed by introducing the kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$ and then solving for

$$\text{maximize} \quad \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to} \quad \alpha_i \geq 0, \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \quad \text{for i=1,...,m.}$$

2) k-Nearest Neighbors: k-Nearest Neighbors is among the simplest algorithms for predicting an unknown target class label. Its intuitive approach consists in calculating the distance separating the closest neighbors y_i to the unknown target class x_i using for example the Minkowski metric. Afterwards a voting among the k observations is performed and the class with the highest number of representatives is selected as the predicted class. We have adopted the Euclidean metric.

3) Random Forests: In a Decision tree each node corresponds to a decision point. While constructing the Decision Tree model, the splits at each node could be performed by minimizing a purity index given by

$$G = \sum_{i=1}^{C} p(i)(1 - p(i)) \tag{18}$$

where G represents the Gini index, C the number of classes and $p(i)$ the class probability. A similar criterion is the Information gain, IG, based on Entropy H, i.e.,

$$IG(y|x_i) = H(y) - H(y|x_i). \tag{19}$$

Random Forests [17] is an ensemble method based on Decision Trees. The latter suffers from over-fitting problems so this method aims to overcome this limitation by splitting randomly the data among multiple trees. Concretely, given the data matrix with m examples and d features, each tree is constructed by randomly selecting a subset $l < d$ from the features and a subset $k < m$ from the examples.

E. Hyperparameters setting

The different hyperparameters used for tuning the learning algorithms are reported in Table. I. For each algorithm we carried out different tests using a selected list of candidates. Hyperparameters could be considered as regularizers which fine tune the model by restricting its freedom. The choice of a particular candidate has therefore a strong influence on the performance of the model.

For instance the value of C in support vector machines controls the points that lie inside the margin. A low value of C softens the constraints on the allowed points while a large value of C hardens the constraints.

The Radial Basis Function (RBF) Kernel is given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||_2^2) \tag{20}$$

where the parameter $\gamma = \frac{1}{2\sigma^2}$ controls the euclidian distance between the data points.

TABLE I: Hyperparameter Settings.

Algorithm	Hyperparameters
Hierarchical Agglomerative Clustering	Linkage: Ward
k-Nearest Neighbors	Number of Neighbors: 1
Random Forests	Number of Estimators: 400 Maximum Depth : 5
Support Vector Machines	Kernel: Linear, RBF Gamma : 0.1,1 C : 10
Principal Component Analysis	Kernel: Linear

For hyper-parameter tuning the following methods were used.

1) k-fold Cross Validation: One of the major objectives of machine learning is to generalize well on unseen data. For instance if the model was trained on a particular data-set, we are not sure whether it will perform well on new data. For this purpose several approaches were proposed to mitigate this issue. For instance if the data size is small, k-fold Cross-Validation has been proven to be very efficient. k-fold cross validation is a method which consists in dividing randomly the data instances in k folders. One folder is used for testing and the remaining ones for training. Then at each evaluation step a new folder is selected for testing. This process is repeated k times. In this work we have used 4 fold cross-validation.

2) Genetic Algorithms: The principle of Genetic Algorithms is the following; A randomly initialized population $\mathcal{P}$ of individuals where each element represents a potential solution is evolved with the aim of producing better individuals at each new generation. Evolution is carried out by crossover and mutation thus increasing the likelihood of producing successful off-springs.

Here we explored GA to optimize the time of design of Deep Autoencoders. The different individuals in the population were encoded according to a hexadecimal scheme. Under this setting every individual is represented as a string made up of hexadecimal numbers (0-9,A-F). In order to ensure that better individuals are produced after each breeding, the parents were selected based on a fitness function which represents the objective to be maximized. Guess refining was steered by the classification accuracy which the algorithm aims to maximize at each iteration. Using this policy we constraint the algorithm to evolve the hyperparameters (activation functions, filter size, number of filters) which produce successfull Deep Autoencoders architecture.

TABLE II: Autoencoder Evolved Hyperparameters

Hyperparameter	Range
Activation function	ReLu, Elu, Tanh, Logistic Hard Sigmoid, Softplus, Linear
Number of filters	2,4,8,16
Filter size	1,2,3

IV. Results and Discussion

A. Detection

Event detection is an important step within a passive acoustic monitoring system. In order to extract from the data only events of interest, we applied the detector to SUBECO data-set. Fig. 6 shows examples of mooring sound (a) and dolphin whistles (b) detection. We can notice that the energy detector performance drops when the noise level is high which suggests further improvements of the detection process.

B. Feature Selection

Given a training data set denoted with $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$ where the elements of Y are the different waveforms of length

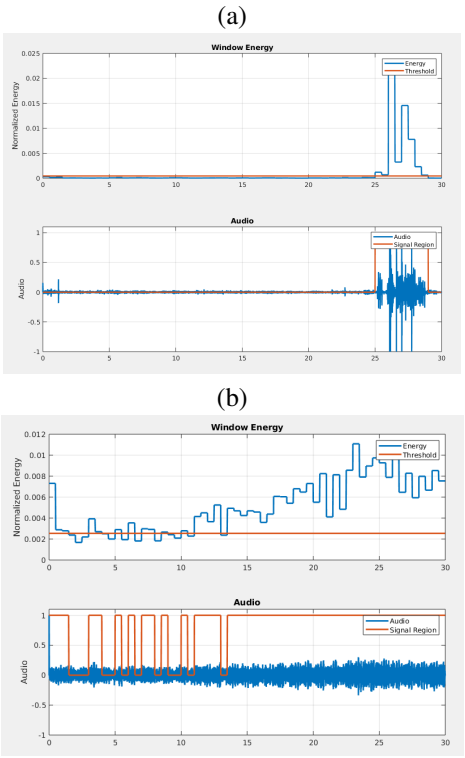

Fig. 6: *Mooring sound detection on DATB0188.wav recording (a), Dolphin whistles detection on DATB0075.wav recording (b) .*

l, a transformation which captures the hidden patterns within l is desirable to optimize learning algorithms performance.

The resulting transformation yield a new vector f_n of reduced length, where the different elements of f_n are denoted features. In order to ensure optimal transformation, a set of requirements arise including feature nature, scaling and dimension among others. Particularly, the choice should take into consideration over-fitting problems. We have conducted feature importance analysis using Random Forests and found that the best performance is achieved by the frequency domain feature MFCC.

C. Clustering

A big challenge encountered during this project was related to the manual labelling of the events. In fact for 2 days of recording, the detection process generated 770 events. We can imagine the tremendous time and effort required for 6 months of recording. For this purpose we have considered using clustering in order to optimize events partitioning into different classes. Fig. 7 shows labels prediction process using three different methods corresponding to Hierarchical Agglomerative Clustering (a), k-Means (b) and Gaussian Mixture Models (c).

In order to get accurate results of clustering, a good choice of the selected features is required. From the threshold in Fig. 7(a) we can clearly notice that the hierarchical algorithm correctly identifies the 4 classes within the data. Labels predictions by GMM reached 74% followed by k-Means with 70%.

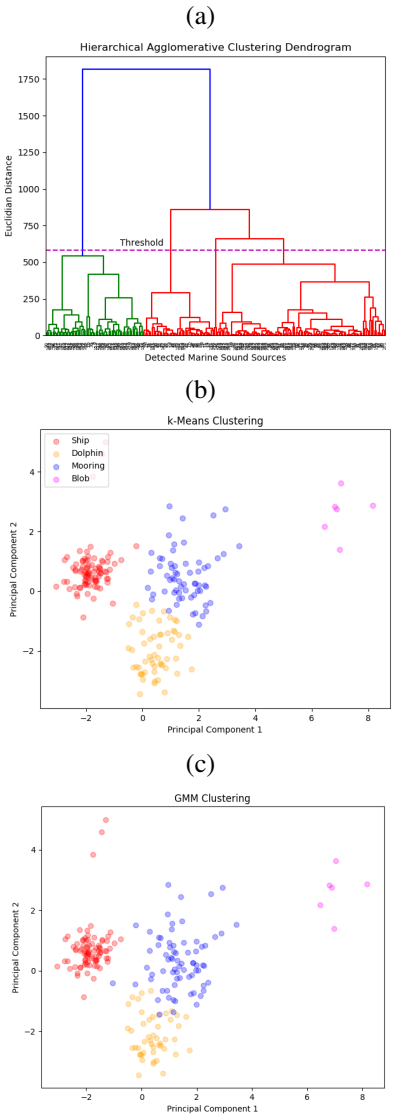

Fig. 7: *Hierarchical Agglomerative Clustering (a), k-Means Clustering (b) and Gaussian Mixture Models Clustering.*

For external validation purposes we have manually labeled the data.

D. Classification

The different events were partioned as the following; 89 ship samples, 47 dolphins, 48 mooring noise and 28 blob (unidentified) samples.

In Table III we present the results from 4 fold cross-validation. The best accuracy rate was achieved by Random Forests with 90.87%, followed by Support Vector Machines 87.36% and k-Nearest Neighbors 86.43%. These results could be explained by the selected features which contributed at a large extent to the derivation of an efficient learning rule. Although Deep Autoencoders performed well, the noticed lower rate suggest further investigation of the network optimization process.

TABLE III: Comparison of performance of different algorithms.

Feature Dimension	Algorithm Name	Recognition Rate
MFCC :: 8	k-NN	83.42%
	RF	90.18%
	SVM	86.14%
Autoencoder :: 8	k-NN	82.96%
	RF	84.30%
	SVM	83.50%
PCA :: 8	k-NN	86.43%
	RF	90.87%
	SVM	87.36%

In order to get a deep insight of the internals of the algorithms we decided to draw the constructed decision rules. In Fig. 8(a) we can visualize how Support Vector Machines divides the feature space using a set of hyperplanes. Although the model (a) misclassifies a set of data points (blue circles), it generalizes better than the model (b) as could be noticed. The best performance is achieved by the RBF kernel with gamma=0.1, since it constructs the optimal prediction surface.

V. Conclusion

In this work we have studied the problem of identification of underwater ambient sounds from passive hydrophone recordings. This was performed using a system which processes the gathered data autonomously. We have considered identifying various sounds through the use of an energy detector. The different features extracted from the detected events allowed to partition the large amount of data for further classification by machine learning algorithms. Our study revealed that deep autoencoders evolved with genetic algorithms offers promising accuracy levels. Besides, we have noticed that labels predictions by the clustering algorithms needs to be improved in order to minimize the penalization of the classification process.

Underwater acoustic signals travelling across large distances are heavily distorded by the channel medium due to complex phenomena which adds Doppler effects and multipath to signal characteristics, consequently we aim as a future work to further analyse the effect of environmental disturbances.

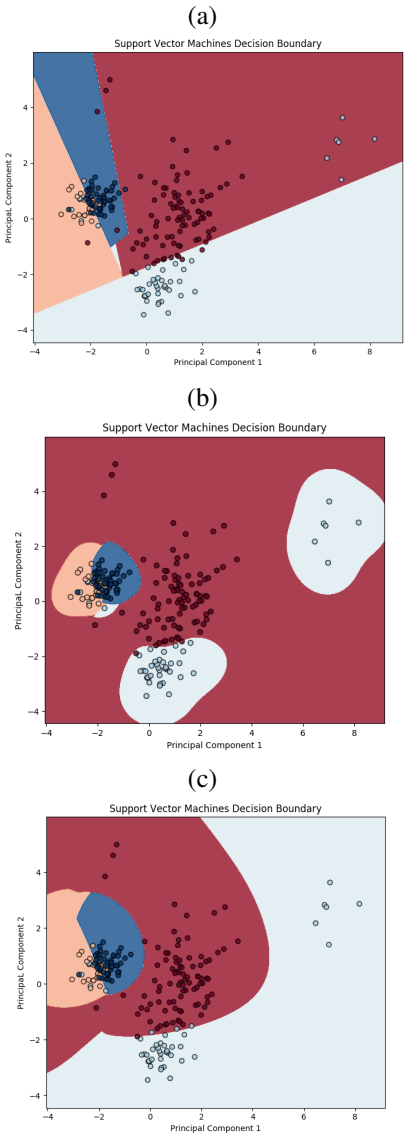

Fig. 8: *Decision boundary plot for Support Vector Machines with a linear kernel (a), Radial Basis Function kernel with gamma=1 (b) and Radial Basis Function kernel with gamma=0.1 (c).*

References

[1] R.K. Andrew, B.M. Howe, J.A. Mercer, and M.A. Dzieciuch. Ocean ambient sounds: Comparing the 1960's with the 1990's for a receiver off the California coast. *ARLO*, (3):65–70, 2002.

[2] G.V. Frisk. Noiseonomics: The relationship between ambient noise levels in the sea and global economic trends. *Sci.Rep.*, 2((437)), 2012.

[3] European Commission. On criteria and methodological standards on good environmental status of marine waters. In *Official journal of the european union*, Brussels, Belgium, 2010. tex.number: 2010/477/EU.

[4] G. Bazile Kinda, Florent Le Courtois, and Yann Stéphan. Ambient noise dynamics in a heavy shipping area. *Marine Pollution Bulletin*, 124(1):535–546, November 2017.

[5] S.M. Kay. *Fundamentals of statistical signal processing: detection theory*, volume 2. Prentice-Hall, New Jersey, USA, 1998.

[6] Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, AMCMM '06, page 21–26, New York, NY, USA, 2006. Association for Computing Machinery.

[7] S. S. Stevens and J. Volkmann. The relation of pitch to frequency: A revised scale. *The American Journal of Psychology*, 53(3):329–353, 1940.

[8] B. Kedem. Spectral analysis and discrimination by zero-crossings. *Proceedings of the IEEE*, 74(11):1477–1493, Nov 1986.

[9] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[10] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, August 1980.

[11] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 113–116 vol.1, Aug 2002.

[12] G. von Bismarck. Sharpness as an attribute of the timbre of steady sounds. *Acta Acustica united with Acustica*, 30(3), 1974.

[13] Roger N. Shepard. Circularity in judgments of relative pitch. *The*

Journal of the Acoustical Society of America, 36(12):2346–2353, 1964.
[14] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.
[15] Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.
[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.
[17] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.