LIFE13 ENV/IT/001254

B3 - Report describing the ANED algorithms for low and high computation capacity sensors

LIFE – DYNAMAP
Dynamic Acoustic Mapping – Development of low cost sensors networks for real time noise mapping

Deliverable Number and Title:	B3 - Report describing the ANED algorithms for low and high computation capacity sensors
Action Number – Title:	B3 - Development of the ANED (Anomalous Noise Events Detection) algorithm
Dissemination Level:	R (Restricted to Beneficiaries)
Status:	Final – Version V01
Release Date:	30/09/2016
Author(s):	Joan Claudi Socoró, Francesc Alías, Rosa Maria Alsina, Xavier Sevillano
Reviewer(s):	Patrizia Bellucci
Document code:	LIFE-DYNAMAP_LA SALLE_ B3- Report describing the ANED algorithms for low and high computation capacity sensors _V01 – 30/09/2016

Contact person:	Rosa Maria Alsina
Postal address:	Funitec La Salle Quatre Camins, 30 08022 Barcelona

	SPAIN
Telephone:	+34 93 2902455
Fax:	+34 93 2902385
E-mail:	ralsina@salleurl.edu
Project Website:	http://www.life-dynamap.eu/

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF KEYWORDS

Environmental sound recognition, audio event detection, noise maps, pilot areas, acoustic sensor network, mel-frequency cepstral coefficients, gammatone cepstral coefficients, anomalous noise events, road traffic noise, recording campaign, supervised machine learning, binary classification, high and low computation capacity sensors, computational cost.

EXECUTIVE SUMMARY

The LIFE-DYNAMAP project (Dynamic Acoustic Mapping - Development of low cost sensors networks for real time noise mapping) aims at developing an automatic dynamic noise mapping system able to detect and represent in real time the acoustic impact due to road infrastructures. The scope of the project is the European Directive 2002/49/EC relating to the assessment and management of environmental noise (END), referring to the need of updating noise maps every five years, as stated in the Directive.

To ease the update of noise maps and reduce their economic impact, DYNAMAP aims at building an automatic and integrated system for data acquisition and processing able to detect and report in real time the acoustic impact due to noise sources. The system should consist of low cost sensors measuring the sound pressure levels emitted by the noise sources present in the area to be mapped; after the measure, a software tool based on a GIS platform is needed to perform real time noise maps update. Any noise coming from other sources but traffic has to be avoided in the noise level evaluation for the noise maps elaboration.

This document contains the work progress description and the results concerning the design and implementation of the Anomalous Noise Event Detection (ANED) algorithm, which has to be implemented in the low cost sensors platform. The purpose of the ANED algorithm is to produce reliable labels at real-time to classify any input audio frame into two main categories: road traffic noise (RTN) and anomalous events (ANE). The sensors will send this information together with the measured equivalent noise level at a regular basis to the server that hosts the GIS software. At that point, the measured noise levels labeled by the ANED algorithm as RTN will be used to update the real-time noise level; but the time frames labeled as ANE will be avoided to be computed for the noise maps update. This way, any other noise sources but the road traffic noise (e.g. noise of trains and tramways, airplanes, people talking, music in cars or in the street, vehicles horns, noise of birds approaching the sensors, meteorological noises like thunders, etc.) will influence the maps update.

The reported work belongs to DYNAMAP action “3 – Development of the ANED (Anomalous Noise Event Detection) algorithm” and it describes the main outcome of this action named as deliverable “3. Prototype of the ANED algorithm for high and low computational capacity sensors ready”. The name of the deliverable is “B3-Report describing the ANED algorithm for low and high computation capacity sensors”. La Salle has lead this action and the progress work has been developed with the collaboration of the following project partners:

- ANAS S.p.A., in the recording campaign performed in Rome within the task Task “B3.1 – Pilot areas onsite inspections and environmental noise recording”.
- BICOCCA - University of Milan (Italy), in the recording campaign performed in Milan within the task “Task B3.1 – Pilot areas onsite inspections and environmental noise recording”.
- BLUE WAVE (Italy), in the ANED implementation for the low cost sensors hardware, concerning to the task “Task B3.2 – Development of the ANED algorithm for high computation capacity sensors”.

The 3rd task developed within action B3 is “Task B3.3 – Development of the ANED algorithm for low computation capacity sensors” has been addressed without the definition of the requirements of a low computational capacity sensors hardware. Then, with the ANED version for the high computation capacity sensors as a reference, some simplifications have been proposed in the algorithm in the aim of obtaining a lower complexity system, assuming a reduction in the accuracy values of the approach.

The main conclusion of this deliverable is that the ANED design and validation with real audio recordings obtained in the two DYNAMAP pilot areas has been addressed, obtaining two solutions that show a different tradeoff between reliability (of its event detection capability) and computational cost. A solution for high computation capacity sensors has been designed and integrated for the sensors hardware real-time workflow, including different type of machine learning strategies (e.g. GMM) and obtaining test reliabilities of about 72-80% (F1 macro-averaged measure) for the two scenarios. A second ANED solution for low computation capacity sensors has been also explored, i.e. reducing the computational load of the detector up to a 17% of the previous version using a simple threshold detector over a signal level measure that is obtained from aggregation of specifically selected set of frequency bandwidths. However, in this case lower detection reliabilities are obtained (of about 60-68% of F1 macro-averaged measure).

The work described in this action report has produced the following scientific contributions:

- Related to Task B3.1
  - Francesc Alías, Joan Claudi Socoró, Xavier Sevillano, Luca Nencini; "Training an Anomalous Noise Event Detection Algorithm for Dynamic Road Traffic Noise Mapping: Environmental Noise Recording Campaign", TecniAcústica 2015, (ISBN 978-84-87985-26-3), pp. 345-352, 21-23 October 2015, Valencia (Spain). Link: http://www.sea-acustica.es/fileadmin/Valencia15/AAM-3%20002.pdf
- Related to Task B3.2:
  - Joan Claudi Socoró, Gerard Ribera, Xavier Sevillano, Francesc Alías; "Development of an Anomalous Noise Event Detection Algorithm for dynamic road traffic noise mapping", Proceedings of the 22nd International Congress on Sound and Vibration (ICSV22), July 2015, Florence, Italy. Link: http://iiav.org/icsv22/index.php?va=viewpage&vaid=175
  - Joan Claudi Socoró, Xavier Albiol, Xavier Sevillano, Francesc Alías; "Analysis and automatic detection of anomalous noise events in real recordings of road traffic noise for the LIFE DYNAMAP project", Inter-Noise 2016, pp. 6370-6379, August 2016, Hamburg, Germany. Link: http://pub.dega-akustik.de/IN2016/data/articles/000268.pdf

LIST OF ABBREVIATIONS

ANE: Anomalous Noise Event
ANED: Anomalous Noise Event Detection algorithm
ANN: Artificial Neural Network
BCK: Background Noise
DA: Discriminant Analysis
DCT: Discrete Cosine Transform
DTD-ANED: Direct Threshold-Detection version of the low cost sensors ANED algorithm.
DYNAMAP: DYNamic Acoustic MAPping
END: European Noise Directive
FFT: Fast Fourier Transform.
FLD: Fischer Linear Discriminant
GMM: Gaussian Mixture Models
GTCC: Gammatone Cepstral Coefficients
IDCT: Inverse Discrete Cosine Transform
KNN: K Nearest Neighbor
MFCC: Mel-Frequency Cepstrum Coefficient
PDF: Probability Density Function
RTN: Road Traffic Noise
STD-ANED: Subband Threshold-Detection version of the low cost sensors ANED algorithm.
SVM: Support Vector Machine
TANE: Anomalous Noise Event Total Duration

1. INTRODUCTION

Traffic noise is one of the main pollutants in urban and suburban areas, which affects the quality of life of their citizens. For instance, the continued exposure to high traffic noise levels has been found to cause harmful health effects, being highly correlated with cardiovascular diseases (Babisch, 2006). As cities grow in size and population, the consequent increase in traffic is making this problem even more present and evident. In order to address this issue, European authorities have driven several initiatives to study, prevent and reduce the effects of exposure of population to traffic noise. Among them, the European Noise Directive (END 2002/49/EC) is focused on the creation of noise level maps in order to inform citizens about their exposure to noise, besides drawing up appropriate action plans to reduce its negative impact (Directive, 2002). In general terms, these maps are updated every 5 years. On the one hand, this entails a time and cost consuming process that is undertaken by local and regional bodies of government, and on the other hand, the resulting action plans are only evaluated in five-year periods.

As a means to start addressing the aforementioned issues, the DYNAMAP (DYNamic Acoustic MAPping - Development of low cost sensors networks for real time noise mapping) project is aimed at developing a dynamic noise mapping system able to detect and represent the acoustic impact of road infrastructures in real time. This way, the project will develop an approach that will reduce the cost of periodically updating noise maps, as required by the END. To that end, an automatic monitoring system is being designed, based on low-cost acoustic sensors hardware. Several algorithms have to be developed to derive reliable acoustic maps through a software tool based on a geographic information system (GIS) platform performing real time noise maps updating.

In order to validate the DYNAMAP's project approach, the system will be deployed in two demonstrative pilot areas in the cities of Milan and Rome (Italy). The first one will be located inside the city of Milan, thus allowing testing the system in an urban scenario, while the second one will be located along a major road surrounding the city of Rome (the A90 highway), making it possible to validate the performance of the system in a suburban environment. The specific selection and location of the network has been defined following the process described in (Radaelli, et al., 2015), with a total of 17 critical areas located along the A90 highway in Rome for the suburban setting. Taking into account several environmental and infrastructural factors (e.g., noise levels, population density, number of dwellings, etc.), the candidate areas were ranked based on scores dependent on these factors. The final selection for Milan as the urban pilot area was district 9 (Zambon, et al., 2015).

Because of automating the road traffic noise mapping, the DYNAMAP system will inevitably have to deal with acoustic events produced by non-traffic sources that could alter the traffic noise levels (e.g., an air-craft flying over, nearby industries or railways, road works, church bells, animals, etc.). In order to increase the road traffic noise mapping robustness, the DYNAMAP system includes an Anomalous Noise Event Detection (ANED) algorithm designed to avoid biasing the traffic noise map computation with non-road traffic acoustic events. Therefore, those events should be detected and eliminated from the noise map computation to provide a reliable picture of the actual road traffic impact.

The core of the ANED algorithm is the audio signal processing by means of spectral feature extraction. These calculated features feed a machine learning algorithm that classifies between road traffic noise (RTN) and anomalous noise event (ANE), including in the first all type of vehicle noise and in the second one any other sound not belonging to RTN. The impact on the

final measure noise levels is variate depending on the duration and the saliency of all the possible ANE, but the ANED algorithm claims to detect all anomalous events found in the ground truth.

In this document the complete work devoted to the ANED design and implementation is detailed, which include:

- The development of the preliminary algorithm versions tested with synthetically mixed road traffic noise and anomalous events obtained from online databases (see Section 2).
- The recording campaign conducted within representative locations of the two selected pilot areas (Rome ring highway and Milan district 9), together with the generation of the two corresponding labelled audio databases used for subsequent studies and algorithm developments (see Section 3).
- The design, validation and implementation of ANED algorithm version designed to run on the high computational capacity sensors (see Section 4).
- The design and validation of a reduced complexity version of the ANED algorithm evaluating its performance and computational cost to run on the possibly later developed low computational capacity sensors (see Section 5).

2. PRELIMINARY STUDIES WITH SYNTHETIC MIXTURES OF ROAD TRAFFIC NOISE AND ANOMALOUS EVENTS

As a first step towards developing the ANED, a two-class classifier was designed and tested following a "detection-by-classification" approach to differentiate "road traffic noise" (RTN) from "anomalous noise events" (ANE). In a road traffic noise-monitoring context like the DYNAMAP project, the presence of anomalous noise events can be:

a. Highly local (e.g. sensors located in roads near airports will often capture aircraft noise while others will rarely be affected by this type of noise)
b. Unpredictable and highly diverse (e.g. ambulance sirens or thundering), and
c. Little likely to occur (e.g. a bird or a cricket that approaches the sensor).

For these reasons, collecting a sufficient number of anomalous noise events samples that represent this high diversity of possible noise sources to train accurately the classifier would require a great effort. To circumvent this inconvenience, a location-independent ANED algorithm based on a semi-supervised approach that avoids creating acoustic models for the minority ANE class was explored initially. Instead, distance-based classifiers were employed, optimizing a decision threshold based on distance distributions with respect to the predominant RTN class. In these preliminary experiments, we compared this approach to a classic two-class supervised classifier (used as a baseline) that creates acoustic models for both classes.

The algorithm was evaluated on a dataset of synthetic mixtures of anomalous events and road traffic noise. The experimental results showed that the proposed scheme outperformed the baseline detector in terms of recognition and detection rates especially in those scenarios in which ANE have higher-pressure levels.

2.1. System description

The initial implementation of the ANED algorithm was designed to automatically detect anomalous noise events follows a pattern recognition approach divided into two main steps: signal feature extraction and recognition. The recognition stage was tackled by supervised machine learning techniques. This requires training the system with noise samples with their corresponding labels in order to build acoustic models that allow recognizing different noise classes.

In the context of the problem at hand, it would be sufficient to train the classifier with N=2 noise categories, as our goal is to detect the presence of noise events other than road traffic noise. Figure 1 depicts the block diagram of a generic 2-way "detection-by-classification" system, referred to as the *baseline* detector hereafter. Notice that two phases are envisaged: *i)* a *training+validation* phase, in which, firstly, one acoustic model per class is built after the parameterization of labelled *training* data (through windowing and feature extraction); and secondly, internal parameters of the classifier are tuned using labelled *validation* data of both classes; and *ii)* an *operation* phase, in which the classifier assigns one of the two possible noise class labels to each frame of an unknown noise signal.

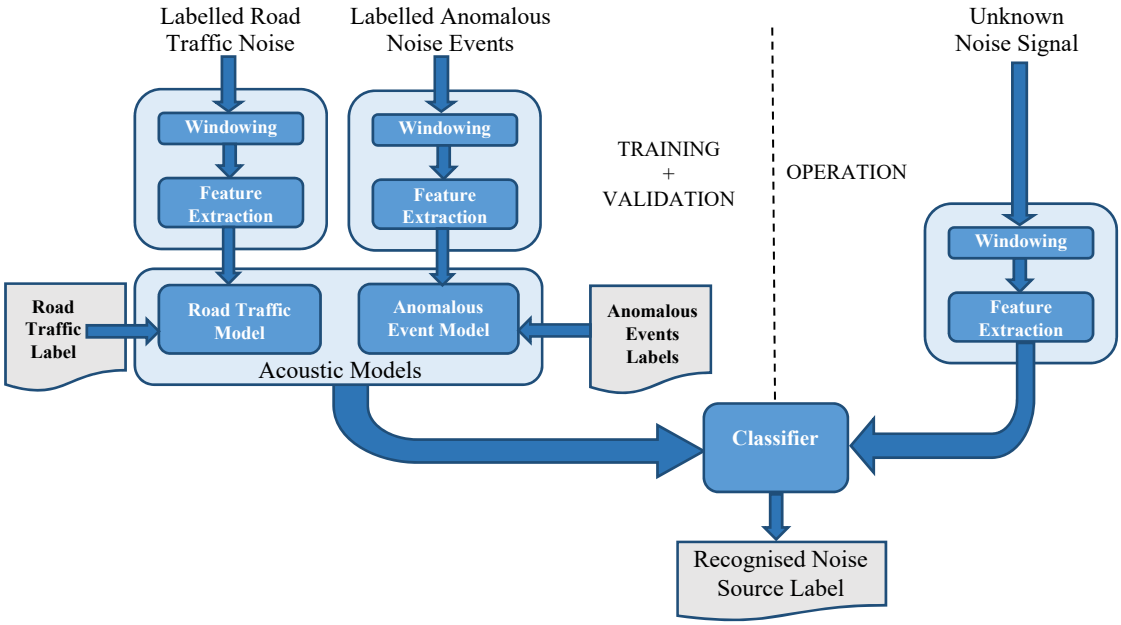


Figure 1 – Block diagram of a baseline 2-way "detection-by-classification" system based on supervised learning.

Since DYNAMAP acoustic monitoring stations will be placed in a fixed location, it would be possible to train the ANED algorithm accurately by collecting a sufficient number of samples of both noise classes at each sensor's location. However, the local, occasional, diverse and unpredictable nature of most types of anomalous noise events makes sample collection a repetitive, difficult and burdensome task. For this reason, we proposed a location-independent semi-supervised 2-way "detection-by-classification" ANED system that minimized the need for anomalous noise events samples collection, avoiding training with on-site collected noise samples.

2.1.1. Proposed semi-supervised approach

Figure 2 shows the block diagram of the proposed semi-supervised ANED approach. One of the main differences with regard to the baseline system depicted in Figure 1 is that anomalous events are used only for adjusting a decision threshold, and no acoustic model is built for this class.

From an operational perspective, the proposed system uses *training* data corresponding to road traffic noise during the training phase to build the acoustic model of the RTN class. Afterwards, a *validation* data set containing samples of both the RTN and ANE classes is employed to adjust the threshold that allows measuring the proximity of the signals of both classes to the learned RTN acoustic model. Although this decision threshold could be fixed heuristically, a more precise value can be obtained by a simple analysis of the two-class (RTN and ANE) distances distributions, obtained from the classifiers responsible for the detection. Finally, the system enters the *operation* mode, in which unseen noise samples are classified either as RTN or as ANE.

The following paragraphs describe the constituting modules and implementation details of the proposed ANED system, namely: signal parameterization, classification algorithm and decision threshold optimization.

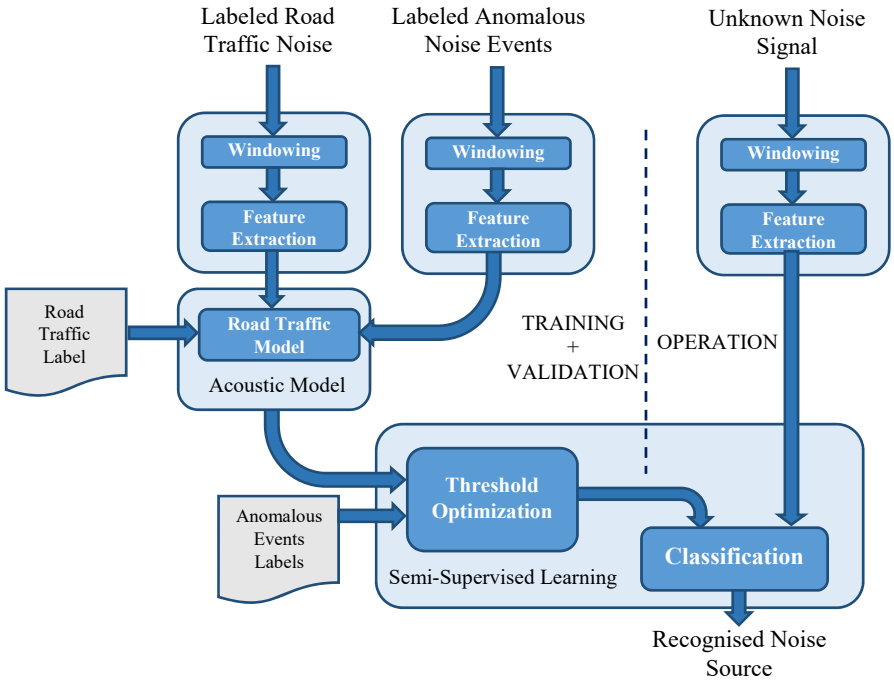


Figure 2 – Block diagram of the proposed semi-supervised ANED approach.

2.1.2. Signal parameterization

The feature extraction block of the ANED algorithm parameterises the noise signal by means of a fixed-size set of coefficients that model the spectro-temporal characteristics of the noise signals. To this end, first the input signal is segmented into 30 ms frames using a Hanning window. Subsequently, a feature set is extracted from each signal frame. In this work, we have selected the biologically-inspired Gammatone Cesptral Coefficients (GTCC), which have recently shown an improved performance in environmental sound recognition tasks (Valero, 2012). Also, Mel-Frequency Cepstral Coefficients (MFCC) are compared to GTCC in the experiments for being a classic baseline benchmark. The number of computed coefficients is 13 for both GTCC and MFCC. For more details about signal parametrization, please address (Alías, et al., 2016).

2.1.3. Classifiers

In this initial implementation, we considered two simple but effective classification techniques: K-Nearest Neighbour (KNN) and Fisher's Linear Discriminant (FLD). The choice of these two classification techniques was motivated by the fact that they both provide a certain distance measure that can be interpreted as a measure of similarity between the dominant class (i.e., RTN) and the input noise frame, which will used for adjusting the detection decision threshold. For KNN, this is the distance between the input noise frame and the K closest training examples. In case of FLD, this measure corresponds to an estimation of the log probability that road traffic noise is the source of the input frame (thus, a value close to zero shows high similarity to this class while high negative values show that the input could be an anomalous event).

As for the internal configuration of the KNN classifier, the validation set allowed deciding that the KNN would employ the Euclidean distance metric and consider the K=3 nearest neighbours.

2.1.4. Decision threshold optimization

The proposed technique for setting the decision threshold is based on obtaining an equally minimum value of both type I and type II errors (false positives and false negatives), as in (Furui, 1981), where the same criteria was adopted with the aim of obtaining an optimal speaker verification system. This threshold adjustment is performed by using samples from the validation dataset (see section 2.2.1).

2.2. Experiments

2.2.1. Audio database

The audio database used for the experiments consists of real road traffic noise (RTN) recordings of the ring road surrounding the city of Barcelona, synthetically mixed with anomalous noise events (ANE) samples (containing up to 15 noise types, like horns, ambulance sirens, car collisions, church bells, birds, crickets, rain, thunders, etc.) gathered from free online repositories. Road traffic noise free field recordings were obtained using the Brüel & Kjaer 2250 sonometer, with 48 KHz sampling rate, 4.2 Hz - 22.4 kHz broadband linear frequency range and its own microphone (Type 4189). The total length of sound samples used for training is 250 seconds of RTN and 300 seconds of ANE. Finally, to test the system in scenarios in which ANE had different degrees of relevance, the level of each type of noise in the mixtures was adjusted to obtain RTN-to-ANE ratios of -6 and -12 dB.

2.2.2. Baseline techniques and experimental setup

The evaluation process is performed following a 4-fold cross validation scheme. In each repetition, *training* + *validation* and *test* subsets are changed to obtain statistically reliable results. As regards the baseline detector, *training*+*validation* data (75% of the total available data) contains both classes (RTN and ANE). In contrast, in the proposed ANED algorithm, *training* data (12.5% of the total available data) contains only RTN class; while the *validation* set (12.5% of the data) contains RTN and ANE samples.

2.2.3. Results

Table 1 presents the results of the conducted experiments in terms of two evaluation metrics: *i)* the F1 measure of the ANE class ($F1_{ANE}$); *ii)* the total classification rate R in % (averaged percentage of the testing samples correctly classified). The $F1_{ANE}$ is computed as the harmonic mean between *precision* (ratio between the true positives and the total amount of frames classified as ANE) and *recall* (ratio between the true positives and the total amount of true ANE frames). The performance of the proposed ANED algorithm is compared to the baseline detector for RTN-to-ANE ratios of -6 and -12 dB, using the FLD and KNN classifiers, as well as GTCC and MFCC features.

Table 1 shows that the proposed ANED algorithm outperforms the baseline detector in terms of both evaluation metrics in most cases. Specifically, the proposed method attains better results in five of the eight experimental scenarios. In particular, the best recognition accuracy and $F1_{ANE}$ are obtained for the configuration that considers using the proposed ANED scheme, FLD

classifier, GTCC features and RTN-to-ANE = -12 dB (91.46% of recognition rate and 0.8976 value of $F1_{ANE}$).

		RTN-to-ANE = -12 dB				RTN-to-ANE = -6 dB			
		FLD		KNN		FLD		KNN	
		GTCC	MFCC	GTCC	MFCC	GTCC	MFCC	GTCC	MFCC
Baseline	$F1_{ANE}$	0,7877	0,7990	0,8305	**0,8266**	0.6983	**0.8233**	0.7738	**0.8478**
	R (%)	85.25	85.62	87.26	**86.42**	77.43	84.58	**83.77**	**87.12**
ANED	$F1_{ANE}$	**0,8976**	**0,8397**	**0,8440**	0,7710	**0.8252**	0.7906	**0.7810**	0.6718
	R (%)	**91.46**	**87.22**	**87.70**	83.35	**84.56**	**84.79**	79.97	76.82

Table 1 – Results of the conducted experiments. The best results for each combination of RTN-to-ANE vs. Classifier vs. Features are highlighted in bold.

It is important to note that the ANED performs better than the baseline detector especially in the most adverse scenario regards the presence of anomalous noise events (RTN-to-ANE = -12 dB), which is particularly important in the context of the DYNAMAP system.

3. PILOT AREAS ONSITE INSPECTIONS AND ENVIRONMENTAL NOISE RECORDING

This section describes the recording campaign conducted in the two pilot areas of the DYNAMAP project, Rome and Milan, aimed at collecting road traffic and anomalous noise event samples to train the algorithm.

3.1. Introduction

Given the negative effects of continued exposure to dense traffic noise levels (Babisch, 2008), European authorities have driven several initiatives to study the effects and to reduce the exposure of population to traffic noise. An example is the European Noise Directive (END 2002/49/EC), focused on the creation of noise level maps (Directive, 2002). In order distinguish changes in environmental conditions, these maps have to be updated with a 5-year periodicity, which entails a time and cost consuming process undertaken by local and regional bodies of government.

In an attempt to simplify and reduce the cost of updating traffic noise maps, the LIFE+ DYNAMAP project aims at automating the whole process. To that end, the main goal of the project is to develop a network of low cost acoustic sensors and an integrated system for data acquisition and processing able to detect, report and map the acoustic impact caused by road infrastructures in real time.

To validate the proposed approach, during the LIFE+ DYNAMAP project two demonstrative pilot systems will be implemented and tested for at least one year in the cities of Milan and Rome. The first one will be located inside the city of Milan, allowing to test the system in an urban scenario, while the second one will be located along a major road surrounding the city of Rome (the A90 highway), making it possible to validate the performance of the system in a suburban environment. More specifically, in previous stages of the project, a support tool was designed to choose the specific pilot areas where the demonstrative systems will be implemented (Radaelli, et al., 2015). Taking into account several environmental and infrastructural factors (e.g. noise levels, population density, number of dwellings, etc.), the candidate areas were ranked based on scores dependent on those attributes. The two chosen areas were eventually Milan's district 9 as the urban pilot area (Zambon, et al., 2015), and a total of 17 critical areas located along the A90 highway in Rome for the suburban setting (Radaelli, et al., 2015).

However, the automation of the RTN data gathering and analysis processes entails several consequences. One of them has to do with the fact that acoustic events produced by non-traffic sources that could alter the measured noise levels (e.g. an air-craft flying over, nearby industries or railways, road works, church bells, animals, etc.). These noises should be detected and eliminated from the noise map computation to provide a reliable picture of the actual road traffic impact. For this reason, it is necessary to devise strategies to identify automatically anomalous noise events captured by the network of sensors.

To that end, the LIFE+ DYNAMAP project includes the development of an ANED algorithm to detect such events and ensure that the noise levels represented on the dynamic maps only reflect RTN. In its current version, the ANED algorithm follows a semi-supervised machine learning approach that requires the construction of a reliable acoustic model of road traffic noise (Socoró, et al., 2015).

Given the diversity of operating scenarios (i.e. urban and suburban), it is necessary to build acoustic models that faithfully reflect the characteristics of road traffic noise in both configurations. For this reason, an environmental noise recording campaign was conducted on the pilot areas where the two demonstrative versions of the DYNAMAP system will be implemented.

Because of the recording campaign, nearly 10 hours of audio were collected, labeled and processed to train the mentioned acoustic model for subsequent development stages of the ANED algorithm. The primary goal of this paper is to describe all the procedures related to the recording campaign in the following terms. First, section 2 presents the main technical aspects of the recording campaign. Next, section 3 is devoted to the presentation of the audio post-processing tasks conducted on the recorded audio. Finally, section 4 outlines the conclusions and future steps towards the implementation of the ANED algorithm.

3.2. Recording campaign

The main goal of the recording campaign was to collect widely diverse road traffic noise samples in their actual environment conditions to train, validate and test the current version of the ANED algorithm (see (Socoró, et al., 2015) for more details). To that effect, several recordings were conducted between the 18th and 21st of May 2015 in specific locations of the two pilot areas of the LIFE+ DYNAMAP project. The selection of these locations was based on obtaining representative samples of the traffic conditions and acoustic characteristics of the pilot areas.

Moreover, it is worth noting that the monitoring network of sensors of the LIFE+ DYNAMAP system will be composed of low-cost acoustic sensors. The recordings were conducted with two measuring devices simultaneously: one low-cost sensor from Bluewave (Nencini, 2015) connected to a ZOOM H4n digital recorder, (see Figure 3.b) and a Bruel&Kjaer 2250 sonometer (see Figure 3.a), used as a reference. These dual recordings were conducted to allow the validation of the low-cost acoustic sensor performance with respect to the sonometer in the near future. The recording setup was the following:

- Situation of both measuring devices: 50 cm distance between them.
- Sampling: 48 kHz sampling rate with 24 bits/sample,
- Sensitivity verification using a 94 dB_{SPL}, 1 kHz calibration tone.
- Clapping: in order to align the audio recordings from both measuring devices, a sequence of 5 sec. of clapping was performed between both sensors with a separation that assured a very good signal to noise ratio despite the environmental noise.
- Gain adjustment: the input gain of each recorder was selected to guarantee enough room for in-site audio dynamics (no saturation).
- Installation: both recording systems were installed on a tripod and included a windscreen to protect the sensor from wind.
- Orientation: the final orientation of the DYNAMAP low-cost sensors with respect to the traffic flow is still undefined. For this reason, recordings were made with three orientations: putting the sensor in the direction of the traffic –forward orientation–, in the opposite direction –backward–, or orthogonal to the vehicles flow. Moreover, three elevation angles of the sensors positions were also employed: 0º, 45º and -45º.

(a) Bruel&Kjaer 2250 sonometer. (b) Low-cost measuring device.

Figure 3 – Recording equipment.

During the 18th and 19th of May 2015, recordings were conducted in six sites along the A90 highway in Rome (see Figure 4). They constituted a representative subset of the 17 sites in this pilot area according to the following four classes (Radaelli, et al., 2015): single road; additional crossing or parallel roads; railway lines running parallel or crossing the A90 motorway; and a complex scenario including multiple connections. In particular, the recording equipment was installed in six highway portals owned by the DYNAMAP-partner ANAS S.p.A (see Figure 5), a government-owned company under the control of the Ministry of Infrastructure and Transport in Italy. During these recordings, the weather conditions were sunny, without rain and with an average temperature of 19 ºC.

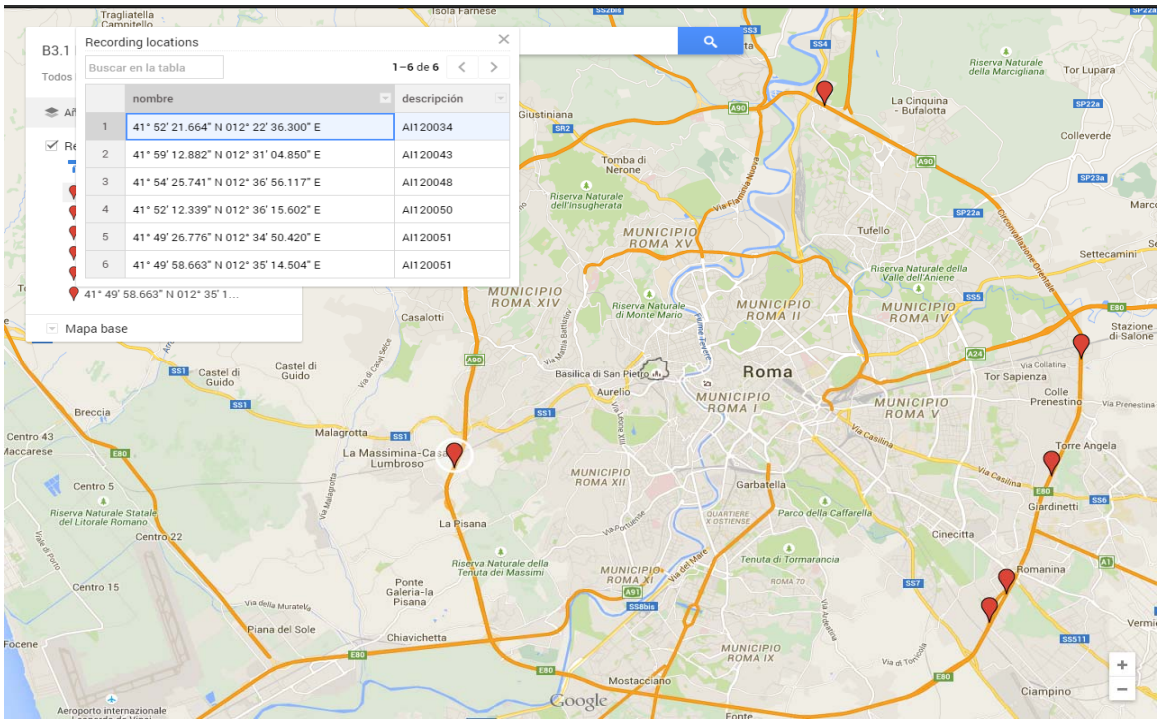


Figure 4 - Locations of the suburban recordings in the A90 highway surrounding Rome.

Figure 5 - Examples of the recording setup installed in the ANAS S.p.A. portals situated on the A90 highway surrounding Rome.

From the 20th to the 21st of May 2015, we moved to the Milan's district 9 pilot area to collect urban road traffic noise samples in twelve locations at different times of day and night (see Figure 6).

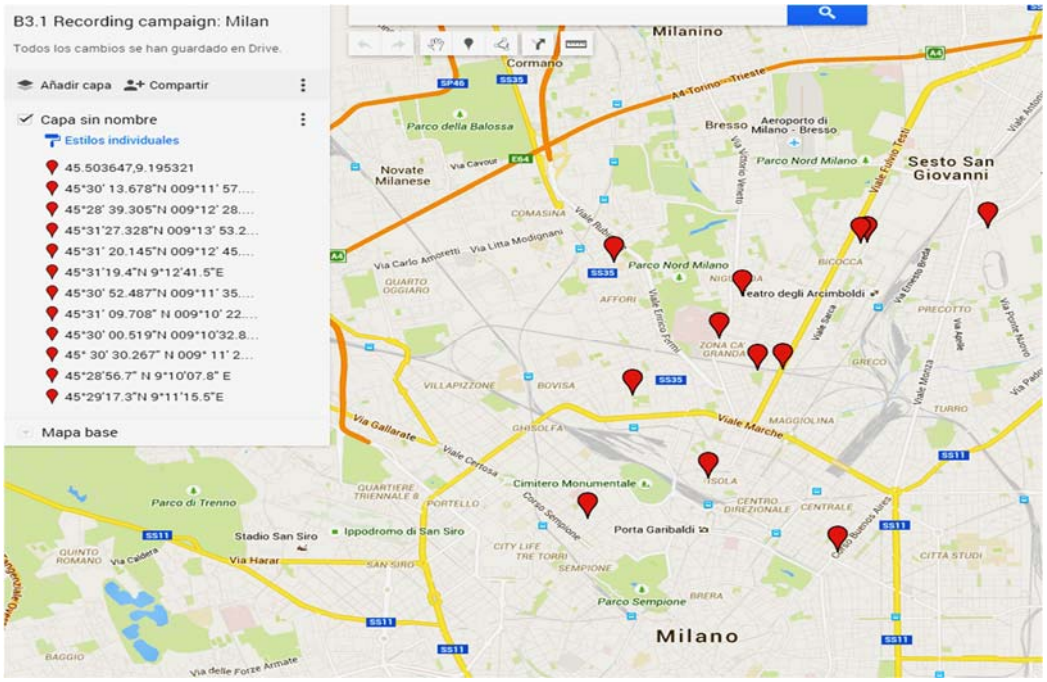


Figure 6 - Locations of the urban recordings within the Milan municipality.

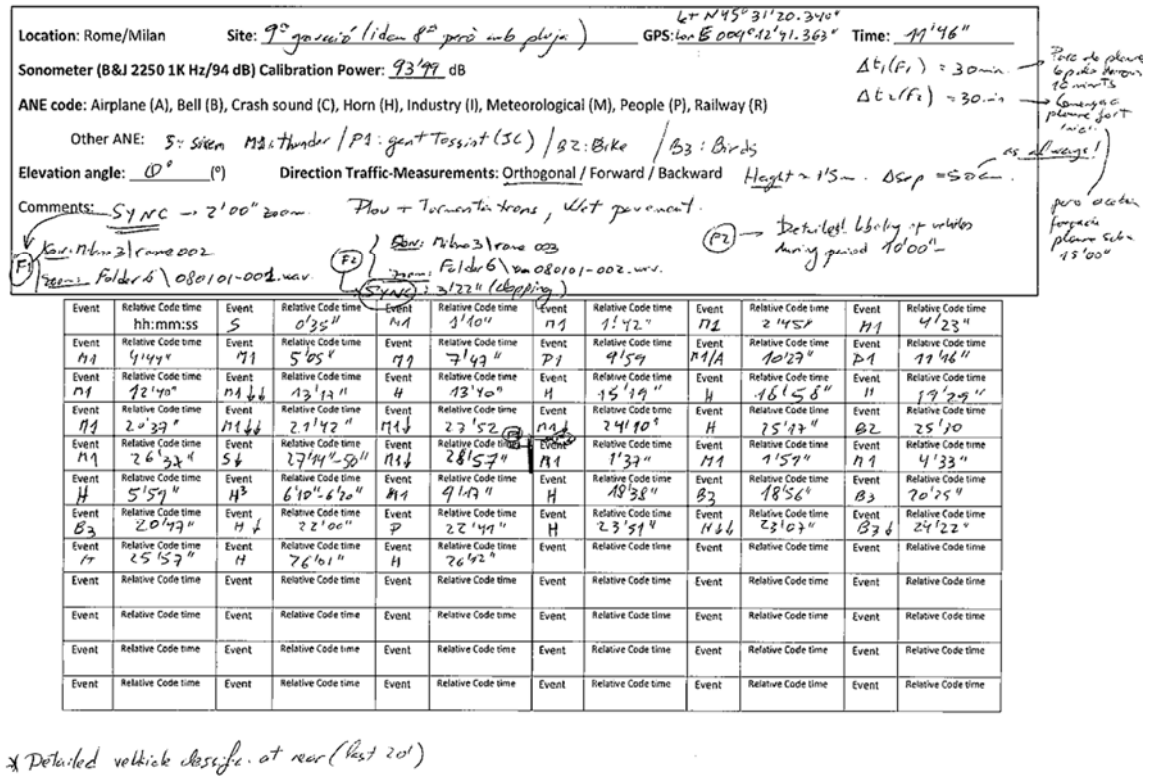
More precisely, the twelve locations correspond to the following specific locations:

1. Near hospital location, including tramways and low traffic.
2. One-way road with very-low traffic.
3. Highly-dense but slow traffic, with tramways, stone pavement, traffic lights and retentions.
4. Railways, very-low traffic.
5. Tram and railways, fluid-fast traffic (multilane).
6. City center, shopping road, crossroad with traffic lights. Wet pavement.
7. Very-low-fluid traffic two-way road at night (multilane).
8. Two-way road with fluid traffic near university (multilane).
9. The same location as number 8 but with wet pavement.
10. Narrow two-way road with fluid traffic in a residential area.
11. Narrow two-way road with very-low-density traffic near a school.
12. Low traffic, narrow one-way street near the city council.

During the Milan recordings the weather was quite sunny, except in the second day when thunderstorms occurred during one of the recordings, making possible to record the noise of thunders as well as road traffic noise with wet pavement.

Furthermore, in order to help the subsequent labeling and post-processing, the recordings were accompanied with a measurement annotation sheet including the following information (see Figure 7): the site location and time, the calibration-tone power measured by the sensors, the elevation angle, and the direction of traffic measurements (facing forwards or backwards). Moreover, the annotation sheet allowed the manual pre-labeling of anomalous events occurring during the recording periods, together with some other particular annotations of the recording context (e.g., new types of acoustic events, or weather conditions).

As a result of the four-day recording campaign between Rome and Milan, a total 10 hours and 20 minutes of audio were collected and prepared for the subsequent labeling and post-processing phase, which is described in the next Section.



* Detailed vehicle classific. at rear (last 20')

Figure 7 - Example of the measurement annotation sheet.

3.3. Audio database generation

After finishing the recording campaign, a post-processing phase was conducted in order to normalize and label all the recorded audio files and export the audio clips in the ANED-specific taxonomy. To that effect, we used the Audacity freeware software.

A total of eighteen audio projects were generated, one for each session during the recording campaign. Six projects were related to the Rome recordings, and twelve to the Milan ones. In

the next paragraphs, the two main procedures performed during the post-processing of the audio recordings are described, divided into two steps: normalization, and labeling plus audio clip export.

Normalization step: For each audio project, the audio files gathered from the two recording devices were imported. After that, the amplitude of the audio files was normalized to make all the noise recordings uniform. This is a key process to avoid biasing the performance of the ANED algorithm. To this end, the amplitude gain was set to adjust the amplitude of the 1 kHz calibration tone of 94dBSPL to -30 dB full scale in the audio signal spectrum. For that, we used the Audacity spectrogram analysis tool with a 512-point FFT and a Hanning window of 10.6 ms. On the one hand, this setting avoids the clipping of the regions of interest along the recordings, and, on the other hand, it allows obtaining a relevant audio signal amplitude to perform the posterior subjective labeling comfortably. Secondly, both parallel audios were manually aligned thanks to the clapping passages at the beginning of each recording session, using the clap impulsive signals (of 5 ms length each in average) to reliably align the parallel recordings through simple visual inspection. This process was followed by a subsequent perceptual validation stage based on assigning each audio to a different stereo channel (i.e., the signal from the low-cost sensor to the left channel and the signal recorded by the sonometer to the right channel) to ensure there was no perceived delay in the audio.

Figure 8 shows an example of an Audacity project to illustrate the processes just described. The two tracks correspond to the signals simultaneously recorded by the two devices in the same location. The amplitude normalization adjusts the amplitude of the calibration tone (corresponding to the selected area in the first track) to -30dB full scale with the help of the spectrogram in the bottom-right region of the figure. The time alignment between both tracks is conducted by zooming the clapping area (interval between seconds 35 and 38) and delaying the second track with respect the first one, as shown in the bottom-left part of the figure.

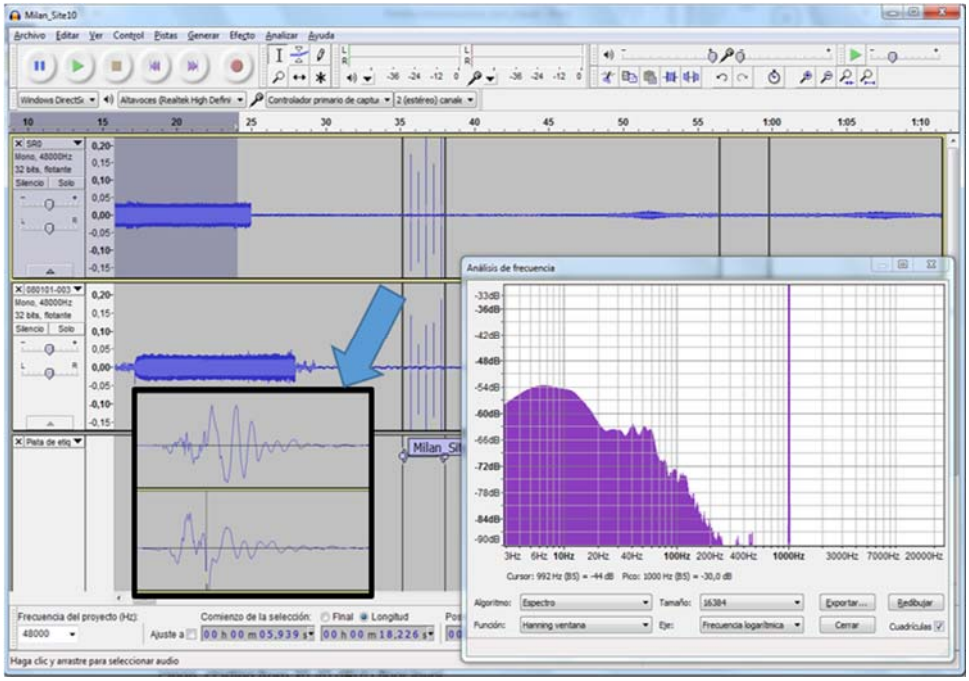


Figure 8 - Example of an Audacity audio project during the normalization process

Audio labeling and clip exporting step. The step of labeling the audio files was toilsome and time consuming, as it entailed listening to nearly 10 hours of recordings to label audio passages according to a predefined taxonomy. Our annotation system distinguishes among the following environmental noise events: road traffic noise (RTN), background noise (BCK) and anomalous events (ANE). RTN labels were assigned to all audio regions containing the pass-by of road vehicles. BCK labels were reserved to those passages where it was difficult to identify the noise coming from vehicles since they contain the background noise of the city, e.g., quiet noise in a one-way street when no vehicles are present, but some distant traffic noise is perceived. In order to ensure that the ANED algorithm detects anomalous events appropriately, it is necessary to train it using samples of both RTN and BCK classes.

In turn, anomalous events (ANE) were labeled by using different subcategories, taking into account the diversity of acoustic phenomena gathered during the environmental recording campaign. These subcategories were defined, in order to enrich the description of the occurred acoustic events, using the following labels (in alphabetical order): **airp** (airplanes), **bike** (noise of bikes), **bird** (birdsong), **brak** (noise of brake or cars' trimming belt), **busd** (opening bus or tramway door noise, or noise of pressurized air), **chai** (noise of chains), **dog** (barking of dogs), **door** (noise of house or vehicle doors, or other object blows), **horn** (horn vehicles noise), **mega** (noise of people reporting by the public address station), **musi** (music in car or in the street), **peop** (people talking), **sire** (sirens of ambulances, police, etc.), **stru** (noise of portals structure derived from its vibration, typically caused by the passing-by of very large trucks), **thun** (thunder storm), **tram** (stop, start and pass-by of tramways), **tran** (stop, start and pass-by of trains), **trck** (noise when trucks or vehicles with heavy load passed over a bump) and **wind** (noise of wind, or movement of the leaves of trees).

All the ANE labels were associated to time intervals of the audio recordings only if they were easily identified subjectively. However, when an acoustic event was perceived but with high difficulty, that time region was marked with **cmplx** label, i.e., it was hard to distinguish the event from the background road traffic noise or from other acoustic events that simultaneously occur. Furthermore, noise of tires when street bumps were considered within the RTN category.

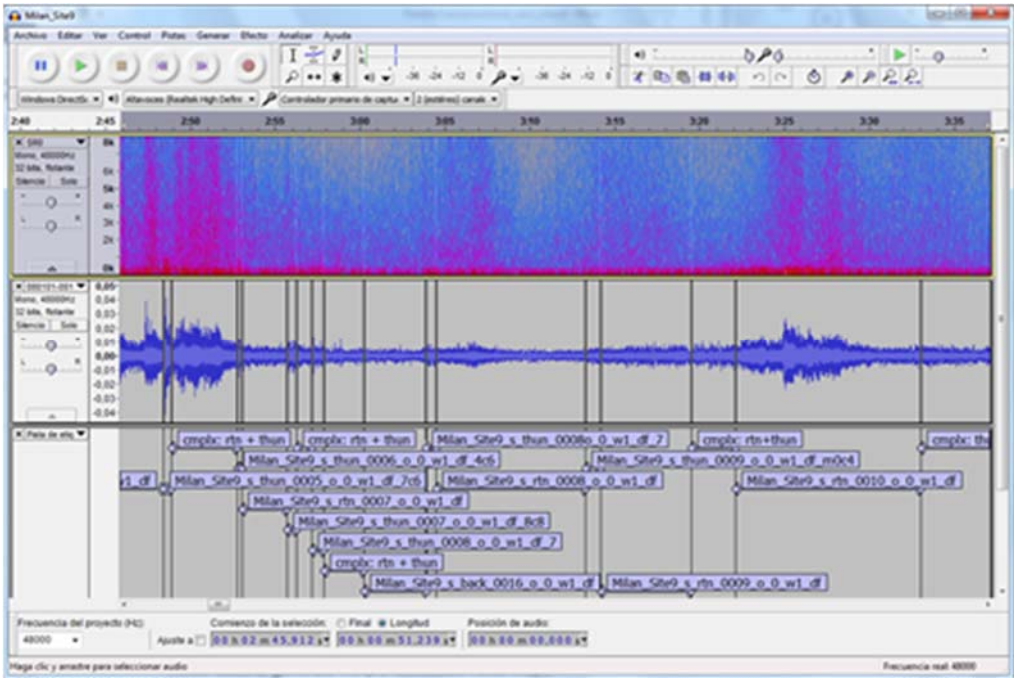


Figure 9 - Example of an Audacity audio project during the labelling process.

Figure 9 illustrates how the labeling process was conducted. To that effect, we listened to the two available audio tracks (one for each recording device), which were also visualized using spectrogram and time waveforms (see the two top tracks of Figure 9). The third track included the corresponding labels for the analyzed audio passage, where road traffic noise (rtn), background noise (bck) and thunders (thun) labels were used. Also, some regions labeled as complex scenes containing a mix of road traffic noise and thunders (cmplx: rtn + thun) are shown. Each one of the audio clips that was not labeled as a complex passage was exported as an independent .wav file (48 KHz and 16 bits/sample).

The labeled audio clips were exported as independent .wav audio files using a sampling rate of 48 KHz and 16 bits/sample. Each filename contained the following parts: type of sensor[1], type of event[2], order of appearance of this type of event in the same audio project[3], direction of measurement in relation with the traffic direction[4], elevation angle of the measurements[5], type of road[6], type of traffic[7]. Additionally, ANE audio clips were also tagged with a computation of the relative amount of ANE amplitude with respect to BCK noise in dBs manually. This computation was performed individually for each audio clip, taking into account the signal to background noise ratio along the ANE with respect to a portion of audio of at least 30 ms immediately before or after the occurrence of the anomalous event. This extra information was added to include valuable information for the training step of the ANED algorithm, i.e., for excluding from training the anomalous events of very low amplitude in relation to the background noise. However, this process was finally improved with an automatic SNR labeling that is explained in section 3.4.

After the labeling process an inventory of the number of files, frames and total durations was generated, which is shown in Table 2.

	Rome		Milan	
	#Wav Files	Total duration	#Wav Files	Total duration
RTN	238	4h 44m 10 sec.	613	4h 23m 59 sec.
BCK	0		286	
ANE	261		711	

Table 2 – Summary of the labels distribution and total durations of recorded audio databases for both scenarios (Rome and Milan).

The complete recording campaign was reported in the TecniAcústica 2015 conference (see (Alías, et al., 2015)).

3.4. From manual to automatic SNR labeling of anomalous events

As was reported previously, see (Alías, et al., 2015), ANE samples were manually tagged by computing the relative sound pressure level of road traffic noise or background noise with

[1] s: Bruel & Kjaer 2250 Sonometer, or z: Zoom H4n recorder plus Bluewave sensor
[2] rtn, bck, peop, musi, tram, sire, stru, horn, brak, thun, bird, trck, door, airp, wind, bike, mega, busd, chai, or dog
[3] from 0 to 100
[4] f : forward, b: backward, fb: in both directions, or o: orthogonal
[5] 0º, 45º or -45º
[6] h: highway, r1: two-way wide road, r2: one-way wide road, r3: two way narrow road, r4: one way narrow road, w1: two-way wide road with wet pavement, w3: two-way narrow road with wet pavement
[7] df: dense and fluid, dr: dense with retentions, l: low, or vl: very low

respect to the ANE level in dBs. However, it is worth noting that we finally defined the signal to noise ratio or SNR the other way round: the relation between ANE and road traffic (or background noise level) in dBs as this definition allows for a better interpretation of the ANE saliency level with respect to the background or road traffic noise levels. This additional information is considered very useful for the subsequent learning stage of the proposed noise classification schemes. The second difference with the approach introduced in (Alías, et al., 2015) is the inclusion of a perceptual-based measure of saliency to consider the human sensitivity to frequency. Finally, the third difference is moving the approach from a manual to an automatic paradigm, which yields to a dramatic reduction of supervision effort.

Specifically, the SNR is estimated using an A-weighted L_{eq} computed with the free Matlab "Continuous Sound and Vibration Analysis" toolbox developed by Edward L. Zechman (Zechman), using a 30 ms integration time. An automatic methodology for computing the contextual SNR is subsequently applied to the obtained L_{eq} profiles of the audio signal obtained for each recording session, which is summarized next. The main goal of this process is obtaining two equivalent noise levels in order to compute what we call the contextual SNR for each anomalous event present in the recorded masters: a median L_{eq} level within the ANE region (which has a duration of T_{ane} seconds, given the start and end marks obtained in the database labelling process) and a median L_{eq} level within the surrounding but closer region to the labeled ANE. Once these two equivalent noise levels are obtained, the computation of the contextual SNR is straightforward.

Contextual SNR is the proposed solution for obtaining estimations of ANE and RTN or BCK levels in the time region where the anomalous event occurs. Obviously, this approach has its own limitations. RTN has a strong a non-stationary behavior and then approximating its level during the ANE time interval using samples from its surroundings is a naive approach. In addition, during the ANE time interval, the measured L_{eq} is also influenced by the background noise (this is especially true for little salient ANE, i.e. those with low SNR). However, assuming that the estimation of SNR using the aforementioned levels has a limited accuracy, we think that the obtained values can be useful enough for the subsequent learning stages of any classification scheme that aims at detecting ANE with the required reliability.

To compute the median L_{eq} level within the closest surroundings of the analyzed ANE, two measurements are made: one before the start of the anomalous event (referred to as left measurement), and another after the event has finished (called right measurement). When possible, the sum of the lengths of the intervals where these two measurements are made should equal the duration of the anomalous noise event.

For illustration purposes, let us define T_L as the duration of the closest background or RTN region before the beginning of the anomalous event. Analogously, we define T_R as the duration of the closest background or RTN region after the end of the anomalous event. Let us also define T_1 and T_2 as the time durations of the two backgrounds or RTN regions considered to compute the corresponding median L_{eq} (T_1 for the background or traffic noise before the ANE start and T_2 for the background or traffic noise after its end). In all the defined time periods the L_{eq} integration time of 30 ms has been subtracted in order to obtain time regions not affected by transients (this time is shown as T_a in next figures). Following the previous definitions, it is clear that $T_L \geq T_1$ and $T_R \geq T_2$. The general aim of the proposed approach is obtaining two background or RTN measurement regions such that T1 + T2 is equal to the anomalous event total duration (TANE). This way, the averaged L_{eq} measure is computed upon similar

conditions. When this condition cannot be accomplished, only the available samples of background and/or road traffic noise are used.

Two case studies were taken into account:

- **An anomalous noise event is surrounded by road traffic or background noise**. This represents the majority of cases in the recorded databases. Within this case study four possibilities exist:
  i. when $T_R \geq T_{ane}/2$ and $TL \geq T_{ane}/2$ then $T_1 = T_2 = T_{ane}/2$ (there are available a half of the ANE duration samples of background or road traffic noise in both sides of the event);
  ii. when $T_R \geq T_{ane}/2$ and $TL < T_{ane}/2$ then $T_1 = T_L$ and $T_2 = \max(T_{ane} - T_1, T_R)$ (less samples of background or road traffic noise are available before the anomalous event start than after its end);
  iii. when $T_R < T_{ane}/2$ and $T_L \geq T_{ane}/2$ then $T_2 = T_R$ and $T_1 = \max(T_{ane} - T_2, T_L)$ (less samples of background or road traffic noise are available after the anomalous event end than before its start);
  iv. when $T_R < T_{ane}/2$ and $T_L < T_{ane}/2$ then $T_2 = T_R$ and $T_1 = T_L$ (there are less samples of background or road traffic noise than the half of the noise event duration at both sides).

  We search for the closest time regions to the current anomalous noise event (ANE_i) measuring the contextual SNR with a proximity criterion, and trying to obtain as many samples of background or road traffic noise as the samples contained in the anomalous noise event duration ($T_1 + T_2 = T_{ane}$). Moreover, integration time (used for the L_{eq} computation) is not considered to avoid transients caused by the start and end of the ANE, obtaining more accurate estimations. In Figure 10 a schematic representation example of a situation with a balanced set of regions is shown.

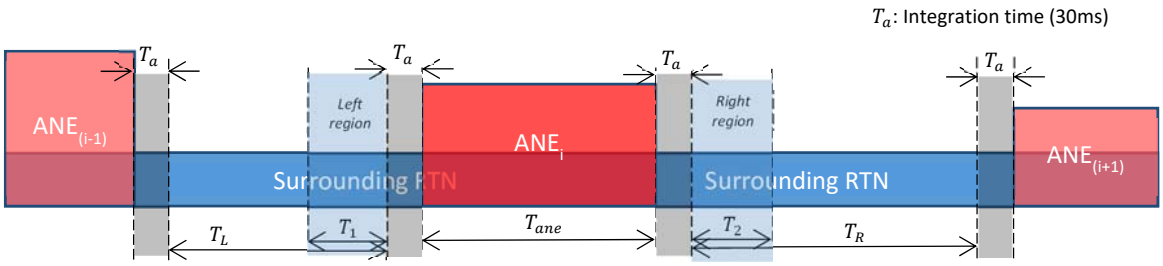


Figure 10 - An anomalous noise event is surrounded by road traffic or background noise. In this case both regions of RTN are balanced.

Contrary to the previous example, in Figure 11 an example is shown where the right region does not contain enough samples to obtain a balanced set of time regions of surrounding RTN, and more samples from the left region are here considered for the SNR computation.

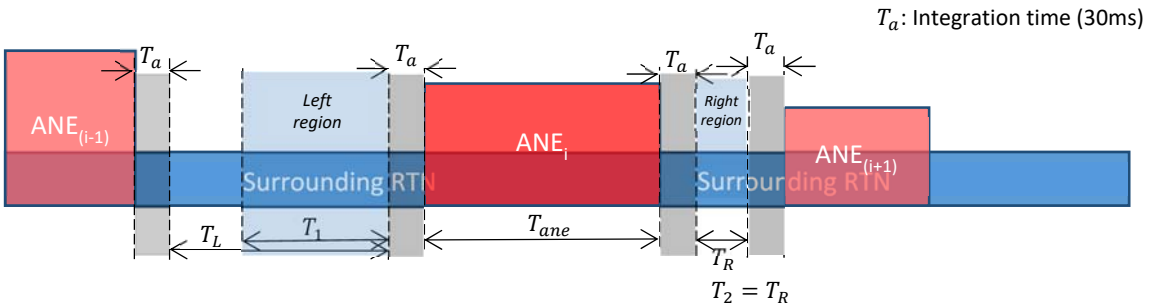


Figure 11 - An anomalous noise event surrounded by road traffic or background noise. In this second case, left region contains more samples than right region.

- **Other noise events occur just before and/or after the analyzed anomalous noise event**. In this less-frequent scenario, the selection of the time regions where the background or the RTN level is computed is a little trickier. We look for the closest time regions to the current anomalous noise event following a global idea of measuring the contextual SNR with a proximity criterion, and trying to obtain as many samples of background or road traffic noise as the samples contained in the anomalous noise event duration (T_{ane}). In this case, we analyze first the background or RTN region that is closer to the analyzed noise event (e.g. let us suppose that this is the road traffic noise time region that is closer but before the analyzed noise event start). When this region contains a time interval greater than T_{ane} in which all the samples are closer to the anomalous noise event than any sample of the opposite side (e.g. after the analyzed noise end), or any sample within this region is closest to the analyzed noise event than any sample of the opposite region, then the interval of duration *min*(T_{ane},T) closest to the analyzed event is selected within this region (being T the duration of this time region). Otherwise, when it is possible to obtain samples of background and/or road traffic noise from both sides of the anomalous noise event with the general criterion that none of these two time regions are strictly closer than the other, i.e. they contain samples equally distant from the closer ANE sample, then samples from both sides are used to compute the road traffic noise or background noise level.

  In Figure 12 a specific example of this situation where all the samples considered for the SNR computation come from the left region is shown. This is because the RTN samples in the right region are further away than the farthest sample of the RTN left region (then, $T_2 = 0$).

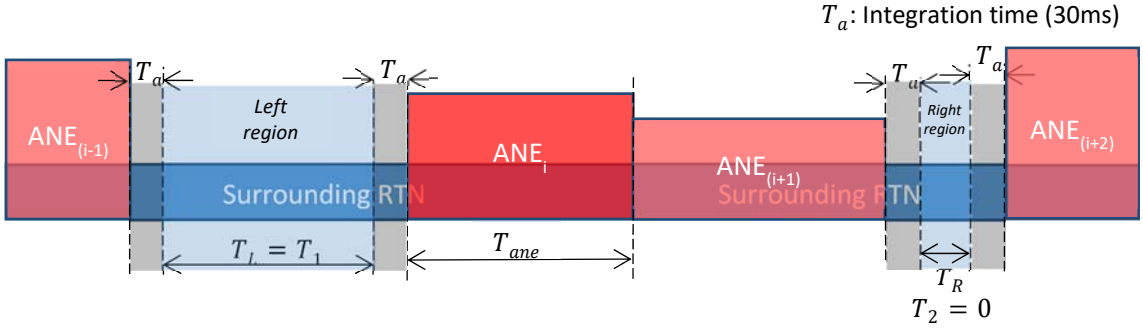


Figure 12 - Other noise events occur just before and/or after the analyzed anomalous noise event. Example where all the samples considered for the SNR computation come from the left region.

In Figure 13 three examples of the most predominant study case (anomalous noise event surrounded by road traffic or background noise) are shown. The L_{eq} curve is highlighted in different color depending on if the time region is attributed to an anomalous noise event (in red) or to background or road traffic noise (in blue). A time period equal to the integration time for the L_{eq} computation is located in black at both sides of the anomalous event (hard to see in these examples), in order to avoid transients affect the SNR measurements. The median L_{eq} levels for each time region are shown as magenta horizontal lines (for background and RTN at both sides) and horizontal red lines (for the ANE).

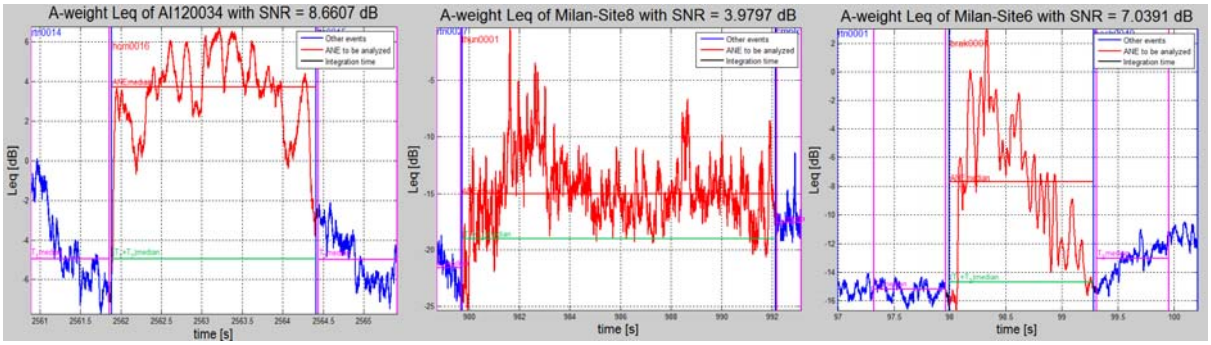


Figure 13 - Examples of anomalous events SNR labeling. From left to right: horn (measured along the A30 motorway in Rome, and with SNR = 8.66 dB), thunder (measured in a Milan road, and with SNR = 3.98 dB) and sound of a brake (also measured in the Milan city, and with SNR = 7.04 dB). Colour code: in red the ANE L_{eq} region and the computed median as a horizontal line, surrounding background or road traffic noise regions are highlighted in blue and its median L_{eq} levels for each side are in magenta, and finally the median L_{eq} of surrounding background or road traffic noise considering both sides is depicted as a green horizontal line within the ANE time region. The SNR is computed as the differences between the median L_{eq} of ANE and RTN. X axis correspond to time in seconds referenced to the start of the recording.

3.5. Analysis of the recorded database

The main purpose of this study is to analyze the distribution and durations of the ANE subcategories and their SNR distribution in a real case scenario. To that end, an in-depth analysis of the recordings of our database was performed. These recordings contain:

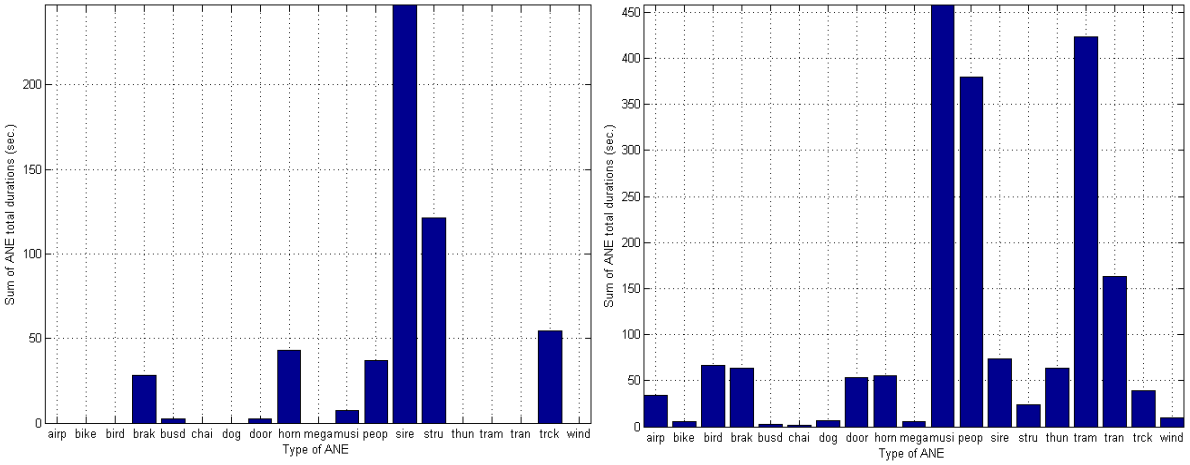
- For the Rome database: 4 hours, 34 minutes and 55 seconds of RTN, and 9 minutes and 3 seconds of ANE, which shows the occasional nature of ANE (i.e., it only represents the 3.2% of the total recorded audio).
- For the Milan database: 3 hours, 51 minutes and 47 seconds of RTN plus BCK, and 32 minutes and 12 seconds of ANE, which in this case represents the 9% of the total recorded audio.

In the next three subsections the analysis of the database is presented taking into account both locations (Rome and Milan) in a separate way.

3.5.1. ANE distributions

This first part of the study aims at determining the predominant types of ANE in each of the two locations recordings (Rome and Milan). To that end, the distribution of ANEs has been analysed by computing the total duration for each ANE subcategory within the recorded database. In figure 2, the distributions of the sum of ANE durations are depicted. As it can be observed, sirens and sound of portals structures (stru) followed by the noise of vehicle horns, people, trucks and car brakes are the most observed subcategories of anomalous events, being the rest significantly-less probable. As the list of anomalous events subcategories were defined

(see section 3.3) to account for all the ANEs recorded both in Rome and in Milan, it can be also observed from Figure 14 that some ANE subcategories do not occur in Rome. For instance, no samples of airplanes, bikes, birds, chains, dogs, mega, thunder, tramways or wind were recorded during the Rome recordings. Finally, it is worth noting that some types of anomalous events were collected unexpectedly. For instance, we recorded highway operators talking while doing maintenance works (i.e., peop ANE subcategory), which is a somewhat rare event to collect in the surroundings of a highway.



(a) Rome recordings

(b) Milan recordings

Figure 14 - Sum of total ANE durations for each type of ANE for the both locations (Rome and Milan). The X axis show the type of ANE.

The recording campaign was long enough to observe the main possible types of anomalous noise events but not all of them (which, by definition are any of those that are different from the road traffic noise). For instance, in the context of a highway like the one studied in the Rome ring it could be difficult that a birdsong distorted the traffic noise during audio measurements, but there is also the possibility that a bird approaches to the sensor and its noise exceeds the background traffic noise. Of course, other type of anomalous noise events, which can eventually be more intense than the background traffic noise, could be observed during longer recording campaigns (e.g. airplanes, not reported during Rome recordings). As can be seen from Figure 14, in Milan a more diverse type of anomalous events where observed than in Rome. This can be explained because in Milan a more diverse type of roads where explored as has been reported in section 3.2 (i.e. roads with different traffic density and city environment) while in Rome all recordings where obtained on a highway with very similar traffic and environment conditions.

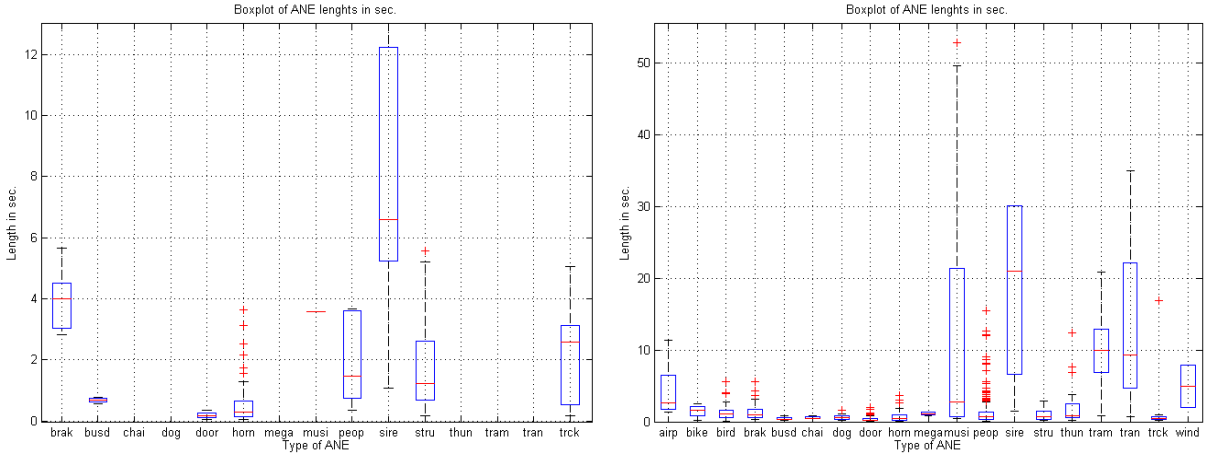
In regard the probability of the recorded ANEs, it can be concluded from Figure 14 that sirens and the sound of portal's structures where the most frequent in Rome while music, people talking and sounds of tramways and trains where predominant in Milan. In a second order of occurrence we found sounds of brakes, horns, people talking and trucks in Rome, while in Milan, sounds of airplanes, birdsongs, car brakes, doors, horns, sirens, structures, thunders (during a rainy day) and trucks were observed. Finally, the less frequent sounds but still present were sound of pressurized air (busd), door or impulsive-like sounds, and music in cars in the city of Rome, and sounds of bikes, opening bus or tramway door noises (busd), chains, dog's barks, noise of people reporting by the public address station (mega), and wind.

3.5.2. ANE durations

In this section, we study the durations of the observed ANE subcategories in both site locations (Rome and Milan).

In Figure 15 the boxplots of ANE durations are shown for each type of ANE subcategory. As it can be observed, in Rome the sounds of sirens constitute the longest ANE, while the shortest ones are very short and impulsive-like noises (which were assigned to "door" label) in the highway ring of Rome. However, in these recordings also brake noises, people, sounds of trucks and noise coming from the structure of portals have significant durations.

Regarding the audio recorded in the district 9 within the Milan city, in Figure 15 we can see that music, sirens and tramways and trains, airplanes and wind sound are the ones with higher time durations. It is worth to note that in the context of an urban environment like the one in Milan, events like sirens exhibit larger durations than in the highway in Rome, because vehicles speeds are lower and rapprochement and remoteness are longer in time.



(a) Rome recordings

(b) Milan recordings

Figure 15 - Boxplots of the durations of each ANE type.

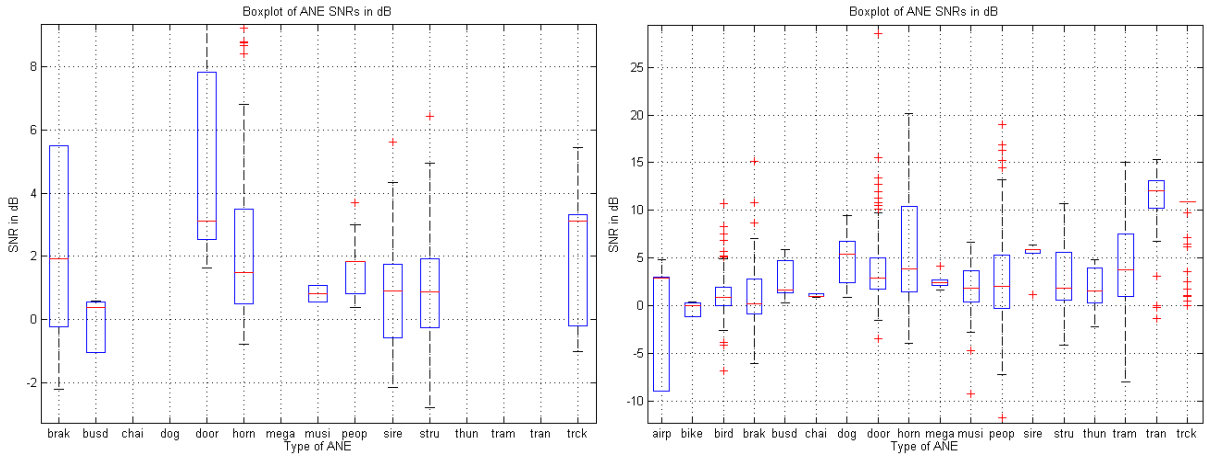
3.5.3. SNR distributions

In this section, the analysis of the contextual SNR (see section 3.4) is described with the aim of obtaining a more accurate description of the saliency of anomalous noise events with respect to the surrounding road traffic noise in real scenarios. The boxplots of the measured SNR for each type of ANE collected in both site locations (Rome and Milan) are depicted in Figure 16.

As it can be observed, the median SNR values are located in the range between 0 and 5 dB, except for dog's barks and sirens in Milan. Hence, anomalous events observed in Rome show a lower saliency than the synthetic mixtures studied in section 2. The traffic density during the recordings in Rome was generally high or very high, which made difficult to obtain audio passages where other types of noises surpassed significantly the background traffic noise. The mean value of SNR for the observed anomalous events is 1.5 dB in Rome and 2.8 dB in Milan.

In Rome door or impulsive-like sounds (door) are the anomalous events presenting a higher SNR, while in Milan there is a more uniform distribution of higher saliencies (i.e. barks of dogs,

horns, people, sirens, trams and trains and sounds of trucks). It is worth to mention that in Milan, anomalous events occurred sometimes within a quiet background conditions (what we labelled with BCK label – background), and that is the reason that for example dog's barks where salient ANEs (while being in one direction road with very low traffic conditions).



(a) Rome recordings (b) Milan recordings

Figure 16 - Boxplots of contextual SNR for each ANE type in each of two site locations (Rome and Milan).

3.6. Corpus parameterization

Two type of cepstral-based features have been computed in order to feed the ANED algorithm: MFCC (*Mel Frequency Cepstral Coefficients*) and GTCC (*Gammatone Cepstral Coefficients*).

MFCC have been largely employed in the speech recognition field but also in the field of audio content classification, due to the fact that their computation is based on perceptually-based frequency scale in the first stage (the human auditory model in which is inspired the frequency Mel-scale). After obtaining the frame-based Fourier transform, outputs of a Mel-scale filter bank are logarithmized and finally they are decorrelated by means of the Discrete Cosine Transform (DCT). Only first DCT coefficients (usually from 8 to 13) are used to gather information that represents the low frequency component of the signal's spectral envelope (mainly related to timbre). In this work 13 coefficients have been employed.

However GTCC is based on Gammatone function that predicts human masking data accurately, and Gammatone filters were originally designed to model the human auditory spectral response, given their good approximation in terms of impulse response, magnitude response and filter bandwidth. Efficient implementation of a Gammatone filter bank (using the 4th-order linear Gammatone filter) has been proposed in the literature for the frequency analysis and resynthesis of audio signals. Here we have used an implementation of the Gammatone cepstral coefficients features by maintaining the effective computation scheme from MFCC but changing the Mel filter bank by a Gammatone filter bank.

In Figure 17, a schematic diagram of the procedure for MFCC and GTCC computation is depicted. First, the audio signal is windowed using a Hanning window of 30 ms length using the original sampling rate of 48000 Hz, which corresponds to an audio frame of 1440 samples. The framing is performed using an overlap of 50% between successive frames, in order to do not neglect any important part of the incoming audio signal. Each audio frame is then

transformed to the frequency domain applying a Fast Fourier Transform (FFT) using the same number of frequency bins as temporal samples. Next step is the filter bank filtering, in which the frequency vector is transformed in a filter bank output by means of simple matrix-vector product. The matrix is a filter bank matrix which rows contain the module of the frequency response of each of the filters of the used filter bank. Then, this matrix is of size 48 by 1440, being 48 the number of selected subbands to account for the cepstral analysis of the signal (or also the total number of filters of the analysis filter bank). When using MFCC filters with triangular-shaped frequency responses are defined, while Gammatone filters are used for GTCC. After the filter bank analysis, the logarithm of the filter output module is evaluated and the Inverse Discrete Cosine Transform (IDCT) is computed, in order to decorrelate the vector samples of the analysis. The final step consists of selecting a reduced set of the output IDCT coefficients (usually those that are in the first positions and that represent the lower frequency content of the cepstral shape) in order to reduce information without an important reduction of accuracy.

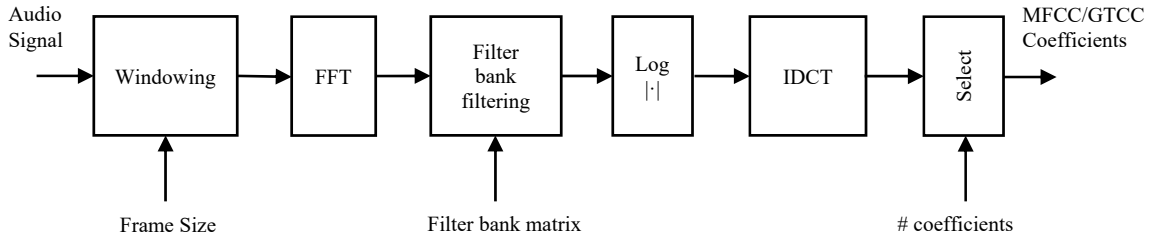


Figure 17 – Schematic diagram of the procedure for MFCC and GTCC computation.

3.6.1. Final audio database

After the parameterization process of the recorded audio database using the two cepstral parameters (MFCC and GTCC) a set of parameterized audio frames were obtained for each scenario (Rome and Milan), shown in the next table. As can be seen, the background (BCK) and the road traffic noise (RTN) labels were considered part of a joint category named RTN&BCK for the purpose of training and validation of the ANED system in the Milan scenario. However, as no background noise was observed in Rome, only RTN label was considered as the main category representing the road traffic noise.

	Rome		Milan	
	#Wav Files	#frames	#Wav Files	#frames # sec
RTN&BCK	238	1099108	613	971071
ANE	162	36349	856	141614

Table 3 – Summary of the labels distribution and total durations of recorded audio databases for both scenarios (Rome and Milan).

Only the recorded audio from the low-cost sensor has been used for training and validation of the ANED algorithm. The other recorded audios (from the sonometer) have been left for further studies.

3.7. Conclusions

In this section, we have described the environmental noise recording campaign performed in May 2015, in the two pilot areas of the LIFE+ DYNAMAP project: Rome and Milan. The main goal of the campaign has been collecting enough representative acoustic data to train, validate and test the ANED algorithm included in the project to avoid including non-traffic noise sources when computing noise maps dynamically. After obtaining nearly 10 hours of audio, subsequent labeling and post-processing has led to 7 hours, 48 minutes and 38 seconds of RTN, 38 minutes and 37 seconds of BCK, and 25 minutes and 54 seconds of ANE. The rest of the recorded audio was labeled as complex audio passages. During this work, we have realized that the latter passages will need further analysis. Next sections will be focused on training the ANED with the obtained acoustic database and validating its performance with respect to the results obtained previously using the synthetic databases.

4. DEVELOPMENT OF THE ANED ALGORITHM FOR THE HIGH COMPUTATION CAPACITY SENSORS

In this section the design and assessment of the ANED algorithm designed for high computational capacity sensors is described. The main goal is obtaining a system capable of attaining reliable results in a real-world scenario, that is, working appropriately with the acoustic data obtained from the recording campaign explained in Section 3 (Alías, et al., 2015): the Rome dataset (Socoró, et al., 2016) and the Milan dataset. As a final output, the ANED algorithm is implemented taking into account the results from the previous studies, considering classification reliability but also computational costs. The ANED algorithm is implemented to be integrated in the hardware platform of the DYNAMAP project, and some preliminary experiments of the computational load are performed to validate its real-time operation.

4.1. Introduction

In Section 2 a first approach of the ANED algorithm based on both semi-supervised and supervised machine learning approaches was evaluated considering synthetic mixtures of recorded road traffic noise and anomalous events gathered from Internet repositories. However, this laboratory generated data lacked on dimensionality and accuracy (in terms of artificially generated SNR ratios).

The rest of this section is organized as follows. Section 4.2 is devoted to the design, training and validation of the ANED algorithm for high computational capacity sensors with the audio datasets obtained from the recording campaign. In this section, the ANED algorithm is implemented using the Matlab© software, and the results obtained are used to select the most appropriate ANED configuration to be implemented in the real-time operation prototype within the DYNAMAP project. In Section 4.3 the work concerning the real-time implementation of the ANED algorithm for the DYNAMAP high computational capacity sensors is detailed, and, finally, the conclusions are described in Section 4.4.

4.2. Design, training and validation of the ANED algorithm for high computational capacity sensors with real data.

This section describes the development of the proposal to discriminate between road traffic noise and anomalous noise events for the high computational capacity sensors. First, the classification schemes explored in Section 2 are analyzed in Section 0 considering the real recordings obtained from the recording campaign (see Section 3). In Section 4.2.2, data reduction is applied through clustering the road traffic noise recordings in order to overcome the performance limitations detected in the previous section: working with balanced datasets of the two main audio classes to distinguish (RTN and ANE). Two different case studies regarding the definition of the training acoustic databases are considered and explained in Section 4.2.3, which constitutes one of the optimization steps towards maximizing the ANED performance. Thanks to having obtained balanced audio datasets of reduced size (in terms of the major class RTN), new classifiers can be considered and its optimization is detailed in Section 4.2.4. To end this section, a computational cost analysis is performed in section 4.2.5.

4.2.1. Learning and testing results with the original dataset

The proposed schemes explored in Section 2 are here revisited with the aim of evaluating their performance with the real recordings collected from Rome and Milan cities. The considered classification schemes are based on two machine learners (K-Nearest Neighbor and Fischer Linear Discriminant) and two machine learning variants (supervised and semi-supervised). However, in the context of the real recordings only the FLD classifier has been able to run with the limited resources of a desktop PC (main features: Intel(R) Core(TM) i7-Q720 CPU @1.60GHz, 4 GB of RAM and O.S. Windows 7). KNN classifier uses the complete labelled database during the test phase, so it demands for huge memory resources and its simulation fails using the aforementioned computational resources. For this reason, only FLD has been assessed in this preliminary set of studies. Moreover, both learning strategies (supervised and semi-supervised) and also two audio parameterizations (MFCC and GTCC) have been compared.

The evaluation process is performed following a 4-fold cross validation scheme following the procedure explained in Section 2. In each fold, training + validation and test subsets are changed so as to obtain statistically reliable results. As regards the supervised version of the ANED algorithm, training plus validation data (75% of the total available data) contains both classes (RTN and ANE). In contrast, in the semi-supervised version, training data (37.5% of the total available data) contains only the RTN class, while the validation set used for the threshold optimization (37.5% of the data) contains both RTN and ANE samples. All the assessments are obtained at the frame level, i.e. every 30ms the test data is assigned to a specific class label (ANE or RTN) by each version of the ANED classifier. Finally, the evaluation measures are computed with respect to the manual labels obtained from the labelling process (i.e., the so called ground truth).

With the aim of providing the classifier with the proper diversity of training data, the ANE frames are randomly selected and distributed into the 4-fold cross validation scheme, assuring that all types of ANEs are present in each fold for the learning+validation and the test partitions in a similar proportion. This process is particularly important to guarantee a fair ANED evaluation, by considering the unbalanced distribution of ANE and RTN within the collected databases in Rome and Milan (see Section 3), which it was not the case in the artificially generated database described in Section 2, designed as balanced.

In Figure 18 and Figure 19 the results obtained with the FLD classifier using the two machine learning strategies (supervised and semi-supervised) and the two types of audio parameterizations (GTCC and MFCC) are shown for the two scenarios (Rome and Milan) respectively. For comparison purposes with previous research conducted using synthetic mixtures, the same two types of evaluation measures are considered: *(i)* the global accuracy of the system, which accounts for the number of correct classifications with respect to the total number of evaluations; and *(ii)* the Macro-averaged F1 measure, which is defined as the harmonic mean of the macro-averaged recall and precision (both values are computed as a mean of the recall and precision of both categories – ANE and RTN):

$$F1\ macro-averaged\ (\%) = 100\frac{2Prec_{MAV}Rec_{MAV}}{Prec_{MAV}+Rec_{MAV}}$$

$$Prec_{MAV} = \frac{Prec_{ANE}+Prec_{RTN}}{2}$$

$$Rec_{MAV} = \frac{Rec_{ANE} + Rec_{RTN}}{2}$$

being $Prec_{ANE}$ and $Prec_{RTN}$ the precisions of ANE and RTN categories (i.e. the number of true positives divided by the sum of true and false positives), and Rec_{ANE} and Rec_{RTN} the recalls of ANE and RTN categories (i.e. the number of true positives divided by the sum of true positives and false negatives).

Contrary to the initial evaluation approach described in Section 2, the F1 measure is macro-averaged across the two categories in order to better consider the unbalanced nature of the RTN and ANE classes. In Section 2 the F1 measure value of the ANE class was computed, which was appropriate in the context of evaluating balanced classes.

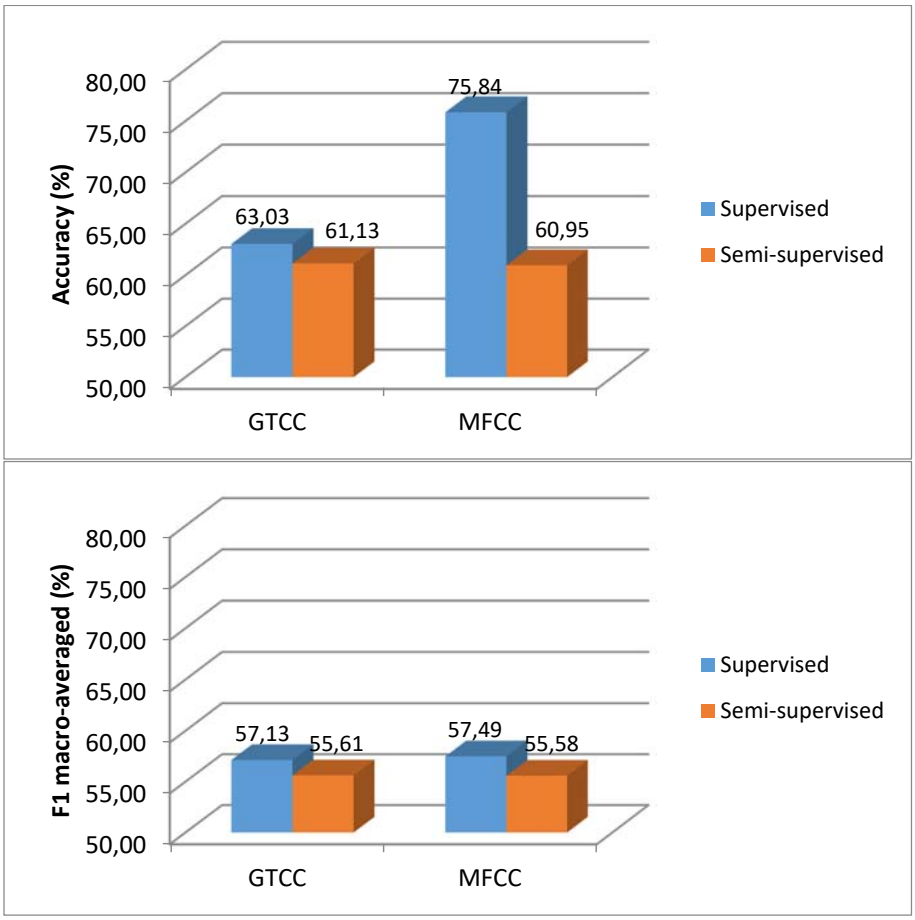


Figure 18 – Results of ANED algorithm considering the Rome pilot area real recordings and with the FLD classifier. Global accuracy is depicted in the upper side while at bottom side the macro-averaged F1 measure is shown, both in %.

As it can be observed from Figure 18 and Figure 19, the supervised version of the ANED algorithm shows the best results both in terms of accuracy and F1 macro-averaged, showing an opposite pattern to what was observed in Section 2 when considering synthetic mixtures of ANE and RTN with 6 and 12 SNRs (ANE-to-RTN SNRs) and balanced classes. Additionally, the averaged results show better performances when using MFCC than GTCC, which also draws a different picture that the results obtained with synthetic mixtures.

In particular, in Rome it can be observed that the accuracies obtained with MFCC are higher than the ones obtained with GTCC for the supervised approach (i.e., 75.84% with MFCC compared to 63.03% with GTCC) (Socoró, et al., 2016). However, this improvement is significantly reduced when comparing both F1 macro-averaged values. This can be explained because the recall of RTN improves in 21% when using MFCC instead of GTCC, while the ANE recall is reduced in 20%. Moreover, the ANE precision improves in 25% when using MFCC rather than GTCC, while the RTN precision remains nearly constant. As RTN is the dominant class (i.e., it represents the 97.5% of the total number of samples), the improvement of RTN recall is significantly biasing the computed accuracy. In contrast, as the F1 macro-averaged measure is designed to better represent the response of a classifier when dealing with significantly unbalanced classes, their values show a different behaviour. Specifically, no clear preference between both types of audio features is observed, since the differences in F1 macro-averaged values are lower than 0.5% (i.e., no statistically significant differences are obtained).

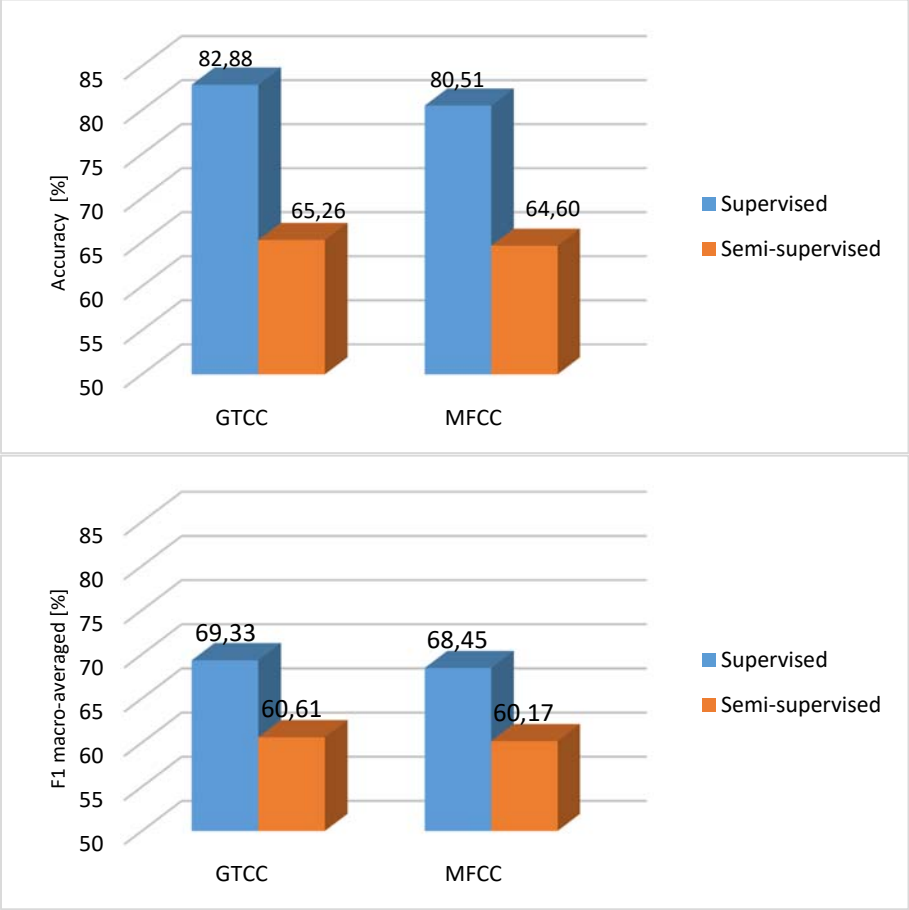


Figure 19 – Results of ANED algorithm considering the Milan pilot area real recordings and with the FLD classifier. Global accuracy is depicted in the upper side while at bottom side the macro-averaged F1 measure is shown, both in %.

Concerning the results for the Milan database, the global performance of the system is very similar regarding the type of audio parameterization used, but showing slightly better results when considering GTCC. However, a similar reduction in performance when comparing accuracy and F1 macro-averaged values is also observed in Milan with respect Rome experiments: accuracy values show good reliability, and especially for the supervised approach, while F1 macro-averaged values are significantly lower. This fact can be explained by the same

previous analysis done on the Rome database related to the unbalanced nature of the audio database (RTN class contains much more instances than the ANE class).

Therefore, while the accuracy is a standard performance measure for evaluating classification schemes when the considered training and test datasets are balanced (there are similar proportion of samples for each class), the F1 macro-averaged is best suited for evaluating classification schemes on unbalanced datasets as the one at hand.

Additionally, it is worth noting that the obtained F1 values are significantly lower than the ones obtained with the artificially generated database (around 25% lower in average), and they are far from the desired values for a competent classifier. These results shows the complexity of dealing with real life audio data and it makes necessary to develop further studies to improve the ANED algorithm by addressing the observed challenges of these data, such as the unbalanced nature of the database and the high diversity of ANEs and SNR values.

For instance, if the experiments in Section 2 are repeated by setting the SNR to the mean value observed in the Rome database, which is 1.5 dB (see Section 3.5.3**Error! No s'ha trobat l'origen de la referència.**), the obtained performance of the classifiers decreases 15% in terms of accuracy in comparison with the results reported with 6dB artificially fixed SNR. However, as in the artificially mixed experiments the datasets are better balanced, both F1 and accuracy measures achieve similar values (with a mean value of approximately 70% across different audio parameters and learning strategies).

The unbalanced nature of the data can be tackled through data reduction techniques like clustering (see the next section), which in turn will make it feasible to train and evaluate other classification techniques like KNN, whose computation was unfeasible with the aforementioned computational resources.

4.2.2. Data clustering and reduction

Recording many hours of on-site audio offers the possibility of capturing traffic noise under different meteorological and traffic density conditions, as well as the chance to record occurrences of anomalous events. On the flip side, however, the gathered recordings contain a large degree of redundancy given the repetitive and rather uniform nature of noise traffic. For this reason, and after completing the labeling process, it makes sense to try to summarize and reduce the road traffic noise samples to reduce the training data to a suitable and representative amount. This stage will allow, in turn, reducing the computational complexity of the ANED algorithm training process.

As discussed in the previous section, training the classifier with the complete dataset, which contains a highly unbalanced proportion of road traffic noise vs. anomalous events, can lead to an overtraining of the majority class. Consequently, the classification system reduces its ability to detect the minority ANE class. To minimize this drawback, a reduction process based on the clustering of the RTN class in both scenarios (Rome and Milan) has been addressed. In the Milan scenario, this reduction has been performed over the RTN+BCK class, let's say, the class that aggregates road traffic noise (RTN) and city background noise (BCK). The latter was only observed within the context of Milan recordings, in an urban environment. The clustering process has been applied to the parameterized audio frames, for both MFCC and GTCC features, deriving different reduced datasets of RTN or RTN+BCK (for Rome and Milan). The

details about the clustering-based reduction process are detailed for the Rome database in the following paragraphs.

Provided that the recognition process is based on the parameterized version of the audio segmented into frames, one possible criterion to perform road traffic noise summarization consists in analyzing the similarity between audio frames. To achieve this goal, cluster analysis offers a series of well-established tools and techniques (Jain, et al., 1999).

Cluster analysis (also known as data clustering or simply clustering) is defined as the task of separating a finite unlabeled data set into a finite and discrete set of natural groups based on similarity criteria (Xu, et al., 2005). After a successful clustering process, the presumably high number of objects contained in the data set can be represented by means of a comparatively lower number of meaningful clusters. This necessarily implies a loss of certain fine details, but yields a simplified and more usable cluster-based data model. We use this idea to obtain a reduced but meaningful version of the original training database after converting it into a cluster-based data model.

To that end, the audio data frames of the original training set have first been clustered using k-means, a classic squared error based clustering algorithm that minimizes the sum of squared distances between the objects and the centroid of the cluster they are assigned to (Jain, et al., 1999).

Usually, it is difficult to ascertain what is the optimal number of clusters a specific dataset should be divided into (Xu, et al., 2005). For this reason, k-means has been run on the RTN data to obtain partitions with different number of clusters. Subsequently, the suitability of each partition has been evaluated using the Davies-Bouldin (DB) index, which is based on minimizing the ratio of within-to-between cluster distance (Davies, et al.), as described by the next equation.

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\{D_{i,j}\}$$

where $D_{i,j}$ is the within-to-between cluster distance ratio for the ith and jth clusters, or equivalently:

$$D_{i,j} = \frac{(\bar{d}_i + \bar{d}_j)}{d_{i,j}}$$

with $\bar{d}_i$ being the average distance between each point in the ith cluster and the centroid of this same cluster, $\bar{d}_j$ the average distance between each point in the jth cluster and the centroid of the jth cluster, and $d_{i,j}$ the Euclidean distance between the centroids of the ith and jth clusters.

As a result, we are able to know which is the optimal number of clusters for the data at hand, as the clustering solution with the minimum DB index is the one with an optimal number of clusters, corresponding to a partition with clusters which are maximally compact and separated.

Finally, the audio data reduction process is completed by selecting, from each cluster, a user-defined percentage of the audio frames weighted by their probability according to their distance to the cluster centroid. To that end, the normalized histogram of distances with respect to each

cluster centroid is computed and employed to select, for consecutive distance intervals, a proportional number of audio frames to end up with a reduced but representative sample of the population of each cluster.

The following paragraphs illustrate the results of the data reduction process. After the audio labeling process, we have obtained 603.680 parameterized audio frames containing only RTN+BCK samples. These frames are subject to clustering, obtaining a set of partitions of 2 to 10 clusters each, which are evaluated using the DB index. Figure 20 depicts the value of the DB index (corresponding to the ratio of within-to-between cluster distances) of the evaluated clustering solutions in the case of the RTN frames of the Rome database. It can be observed that the index shows a monotonically increasing behavior, indicating that the more clusters the data is grouped in, the less compact and separated clusters are obtained. In other words, the DB index value is minimized by the 2-way partition, which indicates that the parameterized audio RTN+BCK signal tends to group naturally into two clusters.

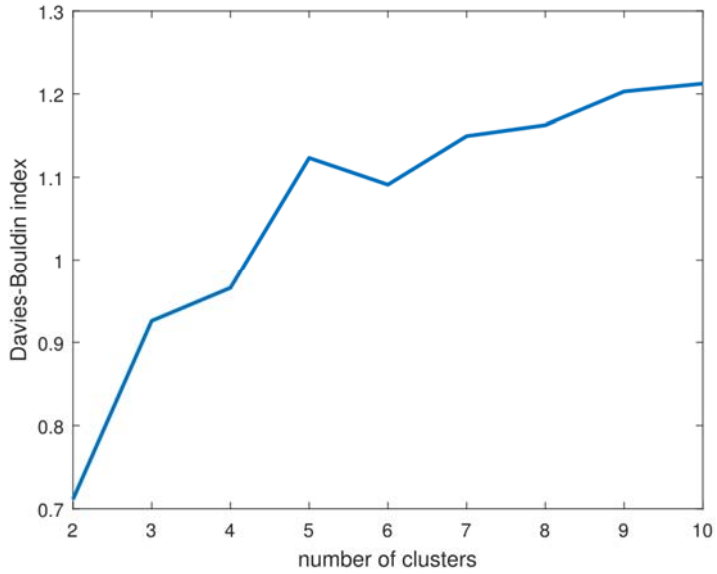


Figure 20 - Values of the Davies-Bouldin cluster evaluation index for the of 2 to 10 clusters partitions of the RTN parameterized audio in the Rome scenario.

Next, Figure 21 presents the normalized histograms of the distances of all frames to each cluster centroid. It can be observed that these distance distributions present different shapes for each cluster (positively skewed for cluster 1 and slightly bimodal for cluster 2). For this reason, to obtain a smaller surrogate of the original population it makes sense to sample each cluster following its distance-to-centroid distribution, obtaining a training audio data set that faithfully represents and summarizes the characteristics of the original audio data.

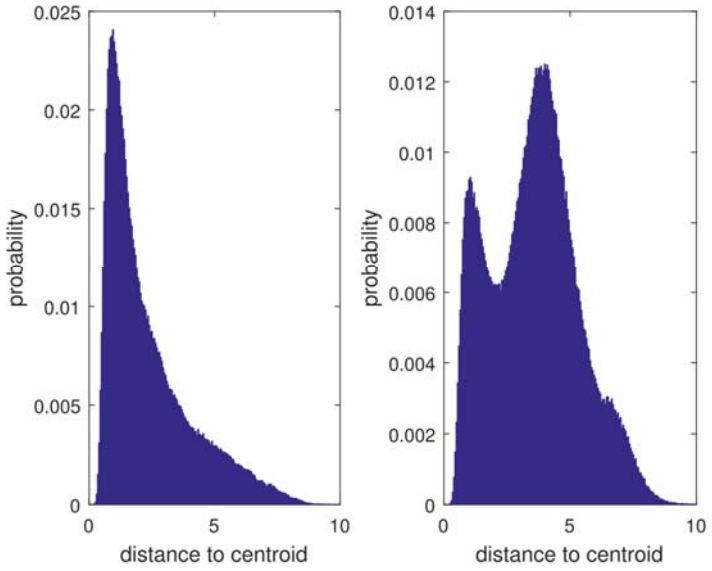


Figure 21 – Normalized histograms of the distances between audio frames and the cluster centroids for the RTN class of the Rome database and using GTCC features.

To evaluate the soundness of this proposal, we conducted the mentioned process to reduce the original training data set size by a 75%, 90% and 99%. To validate the data reduction process results, we run again the k-means clustering algorithm on the reduced data sets, searching from 2 to 10 clusters. In all cases, the DB index chose the partition with two clusters as the optimal one, proving the consistence of the method.

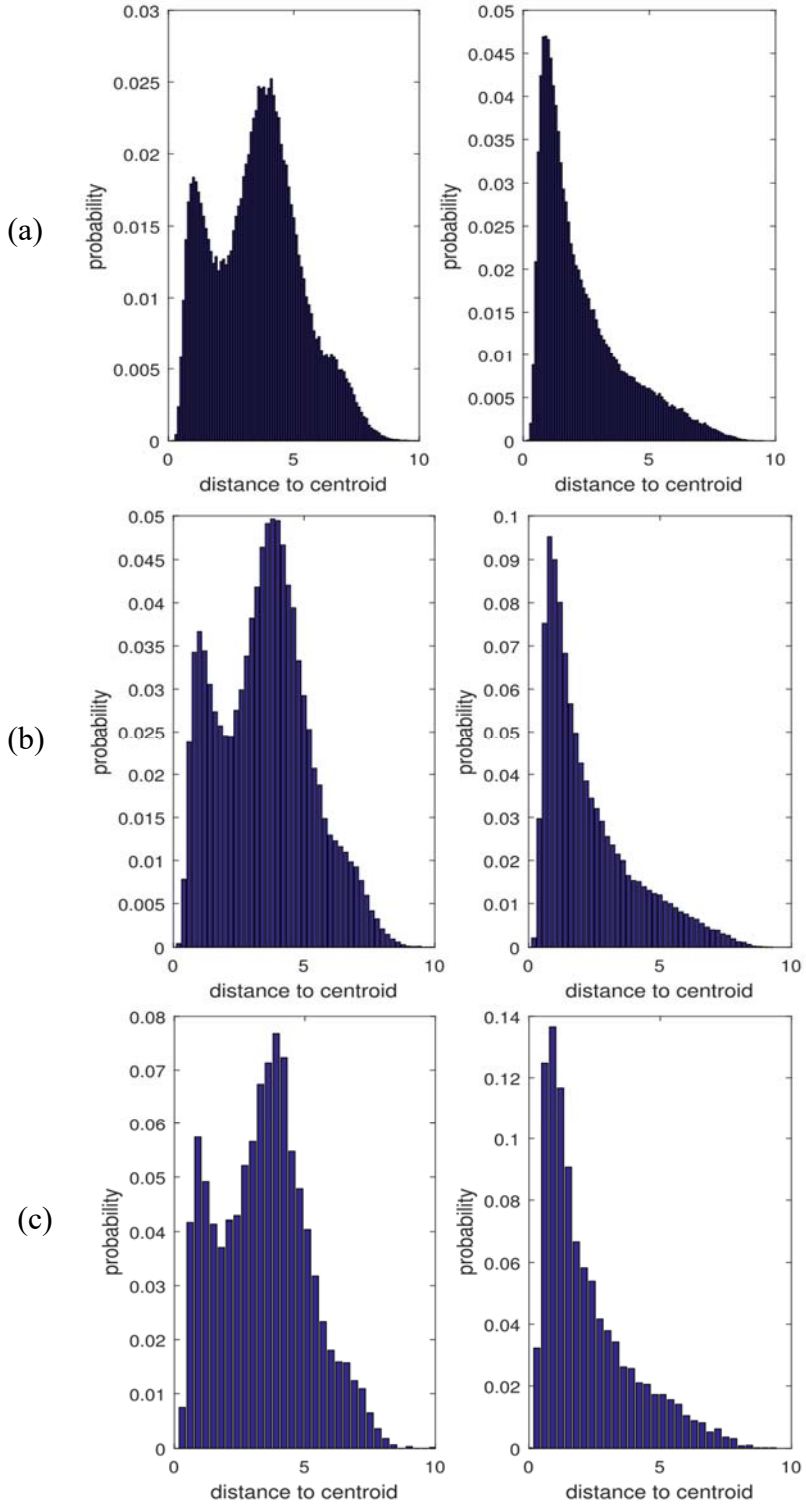


Figure 22 – Normalized histograms of the distances between audio frames and the cluster centroids after the data reduction process with reduction percentages of 75, 90 and 99% for the RTN class in Rome recordings, and using GTCC features: (a) Data reduction of 75% (from 603.680 to 150.920 frames); (b) Data reduction of 90% (from 603.680 to 60.368 frames); (c) Data reduction of 99% (from 603.680 to 6.037 frames).

To further validate the data reduction process results, the normalized histograms of the distances of all frames to each cluster centroid are presented in Figure 22. Comparing these histograms to those depicted in Figure 21, it can be observed that the distance distributions of the clusters

are faithfully preserved, the only difference being the ordering of the clusters caused by the unsupervised nature of clustering. In conclusion, the proposed training data reduction process allows adjusting the size of the training data set as desired, while preserving the structure of the original data.

In order to validate to what extent the clustering-based reduction process of the majority class (RTN) affect the reliability results of the classifier learning process a study was conducted using FLD classifier with supervised learning and changing the % of the reduced RTN+BCK class in comparison with the original data subset within the values {1%, 3%, 5%, 10%, 20%}. In Figure 23 and Figure 24 the assessment results in Rome and Milan scenarios are shown, respectively. With Rome audio dataset, the original number of signal frames of both RTN and ANE classes are 1.099.108 and 36.349, respectively (see table Table 3). Thus, ANE feature vectors represent 3.3% of the RTN feature vectors.

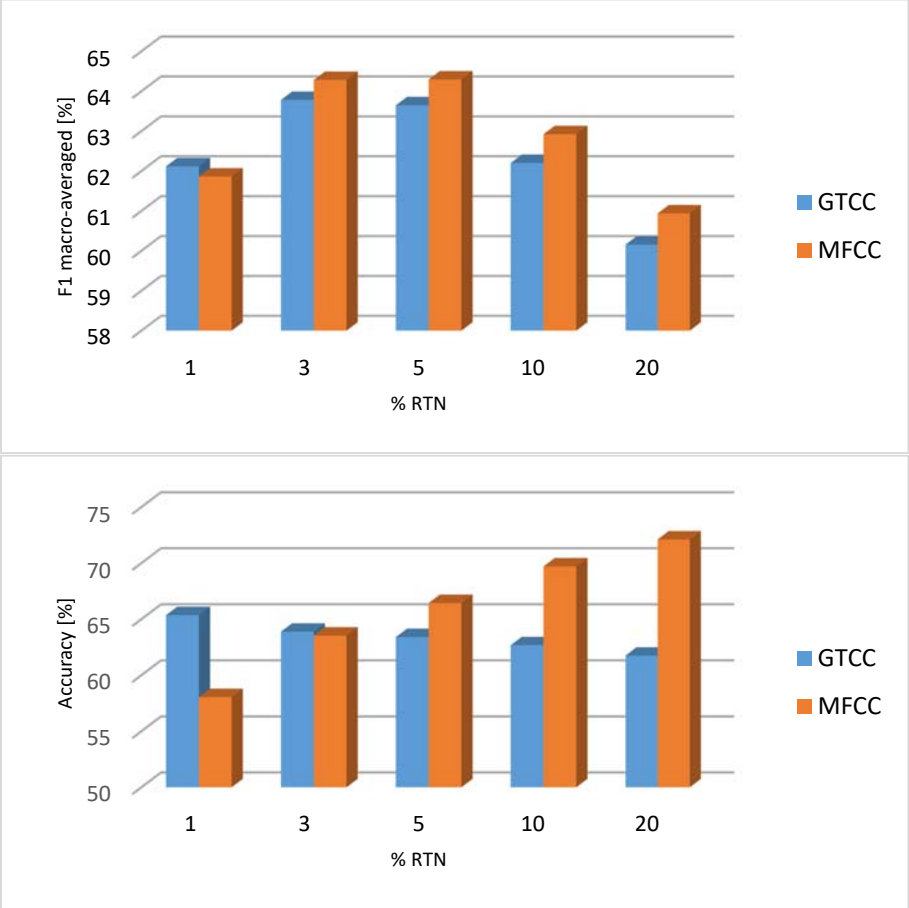


Figure 23 – Results of the FLD classifier over the Rome scenario trained with different reduced RTN class subset, for the two features (MFCC and GTCC). The % of the reduced RTN subset is shown in the X axis. F1 macro-averaged measure is shown on the upper figure while accuracy in the bottom figure.

As we can see in Figure 23, the optimum results from the point of view if the F1 macro-averaged measure are obtained when RTN class is reduced up to 3% of its original size. However, the behavior in terms of accuracy is different depending on the type of audio features used in the classification scheme. The better accuracy results obtained with MFCC features for lower reduction percentages of the RTN class can be explained because the majority class is overtrained, so the classifier tens to better recognize this class and the global accuracy results are better. Then, from here on the 3% reduced RTN class has been chosen as the one more

suitable for further studies within the ANED design and optimization process, because this way the two categories are equally represented in the feature database and no bias is induced in the learning of the classifier. The reduced database has been computed for both MFCC and GTCC features, in which ANE class in maintained as the original one, while the RTN class contains only the subset of audio feature vectors selected by the aforementioned clustering procedure.

Regarding the Milan case, the original database contains 971.071 RTN feature vectors and 141.614 ANE feature vectors, so the ANE class represents the 14.6% of the RTN class. As we can see from Figure 24, when RTN class approaches 10% of its original size, the F1 macro-averaged measure attains almost its maximum value, while the accuracy still shows an increasing pattern that seems to continue above the explored range of data reduction percentage. Following similar reasons as the ones exposed for the Rome database, from here on the ANED design and optimizations have been performed using the labelled audio dataset that contains 10% of the original RTN+BCK class keeping the complete ANE class. In this case, the selected value of 10% is also supported for the fact that when considering larger percentages, the learning process of the different classifiers explored in next sections were almost unaffordable with the available computational resources.

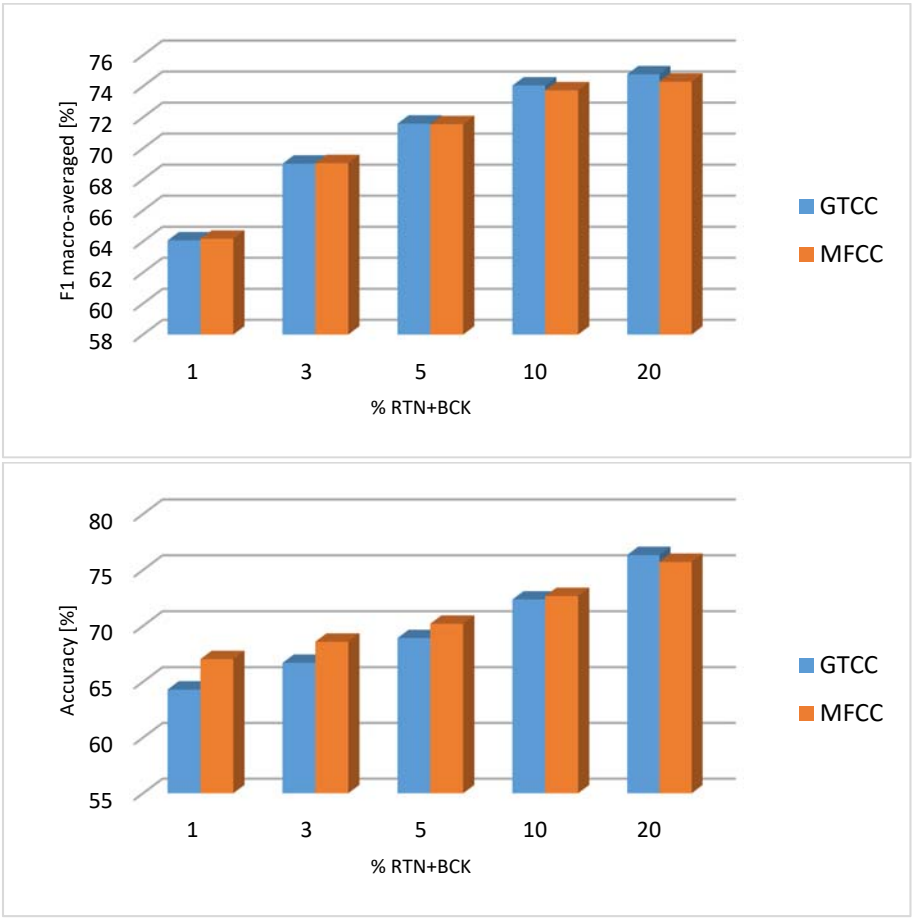


Figure 24 – Results of the FLD classifier over the Milan scenario trained with different reduced RTN class subset, for the two features (MFCC and GTCC). The % of the reduced RTN subset is shown in the X axis. F1 macro-averaged measure is shown on the upper figure while accuracy in the bottom figure.

As we have seen from the results in this section, when classes are well balanced, the accuracy and F1 macro-averaged measures tend to similar values. However, as this balance is not totally perfect, F1 measure still better reflects the behavior of the classifier. For these reasons, hereafter

we only consider the F1 macro-averaged measure to evaluate the performance of subsequent studies regarding ANED algorithm design and optimization.

4.2.3. Cases of study for the classifiers learning stage

As described in the previous section, the classifier performance is highly influenced by the database used for learning. Another important observed issue is to what extent the feature vectors included within the ANE class are faithfully represented by the attributed label. For example, during the database generation process some anomalous noise events labelled were manually attributed to the ANE class without taking into account their saliency with respect to background or the surrounding road traffic noise (see Section 3). However, in the labelling process a SNR measure was computed and attributed to each labelled anomalous event (see Section 3.4). This measure has been used in order to assess the different classifiers when evaluated on the reduced datasets. These criteria are set according to the following case of studies:

- Case I: all the feature vectors form the ANE class are considered without differentiating their SNR. Then, if an specific feature vector within this class has negative SNR values (e.g. ANE-to-RTN_SNR=-6 dB) is considered as an ANE frame, in spite of its negative saliency and, logically, resembling road traffic noise.

- Case II: only ANE feature vectors that have a SNR equal or over 0 dB are considered as ANE, while the remaining ANE feature vectors with SNR below 0 dB are considered as RTN.

In the following sections, the optimization of different classification algorithms is addressed thanks to having reduced the original dataset size (the RTN or RTN+BCK class), and their comparison is performed by considering both types of cepstral-based features (MFCC and GTCC) and the two considered cases of study (case I and case II).

4.2.4. Optimization of classifiers

In this section the optimization of the different machine learning approaches is tackled with the aim of obtaining a fair comparison between classifiers. To that end, a set of simulations has been conducted by performing a sweep of values of several parameters and options that are related to the internal structure of each classification technique. Also, the two sets of recordings (those gathered in Rome and the others corresponding to the city of Milan) and the two cases of study (Case I and Case II) are assessed in the study in order to establish the most convenient scenario for the final choice.

In regard the type of cepstral parameters to feed the classifier, another simplification has been addressed in order to reduce the computational complexity of an exhaustive considering all the variables simultaneously: classifiers parameters, cases of study and type of cepstral parameters. During the exploration of optimal configuration parameters of each classifier and each case of study, MFCC parameters have been used for analysis purposes. Once the optimal configuration for each classifier has been set a further study has been addressed considering both type of cepstral parameters (MFCC and GTCC) in order to explore if an improvement of results can be obtained.

Each study has been set in a range of values which allows to run simulations using Matlab © software on a standard desktop PC (SO Windows 64 bit, Intel i5 CPU 3.20 GHz, 4 MB RAM) in a reasonable period of time (less than 1 day per study). For that reason, finding the optimum values of configuration parameters for each classifier is not always guaranteed. Instead, the best configuration value is finally selected within the exploration interval of the study. It is worth to mention that when multiple configuration parameters are studied for a certain classifier (e.g. ANN), each parameter is studied after fixing the other parameters to a specific value, usually using intermediate values within the range of values of each parameter. In this case, an exhaustive multidimensional search approach is discarded since it would demand excessive computational resources.

RTN data reduction through clustering using 10% of reduction for Milan recordings while 3% for the Rome recordings is also considered. As previously observed in Section 4.2.2, these values yield to quite well balanced classes (RTN and ANE). This solution constitutes a good framework for an efficient operational scheme for the ANED algorithm. Finally, only F1 macro-averaged measure is computed, as previously argued.

4.3.4.1. Optimization of the kNN

The KNN classifier has been optimized ranging parameter *k* (the number of neighbors used for taking the final decision) within the set of values {1, 3, 5 and 7}. In Figure 25 and Figure 26 the results of the KNN optimization are shown using Rome and Milan databases, respectively.

As it can be observed from the results, the behavior of the classifier performance is completely different in the two studied scenarios. In Rome, $k = 1$ is the optimal value while it is $k = 7$ for the Milan recordings. On the other hand, Case II is the one that attains better results in Rome, while in Milan the optimum case is I.

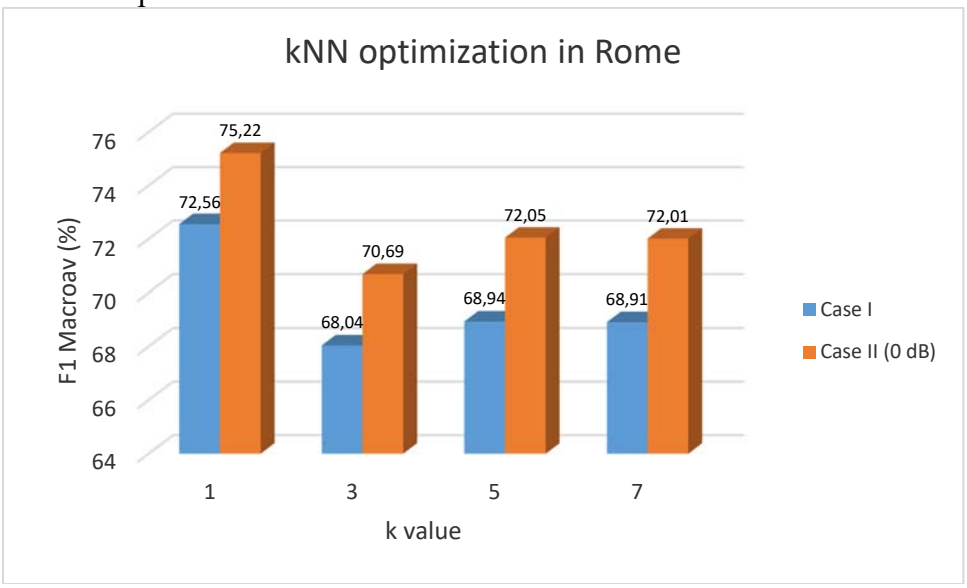


Figure 25 - Optimization results of the kNN for the Rome recordings.

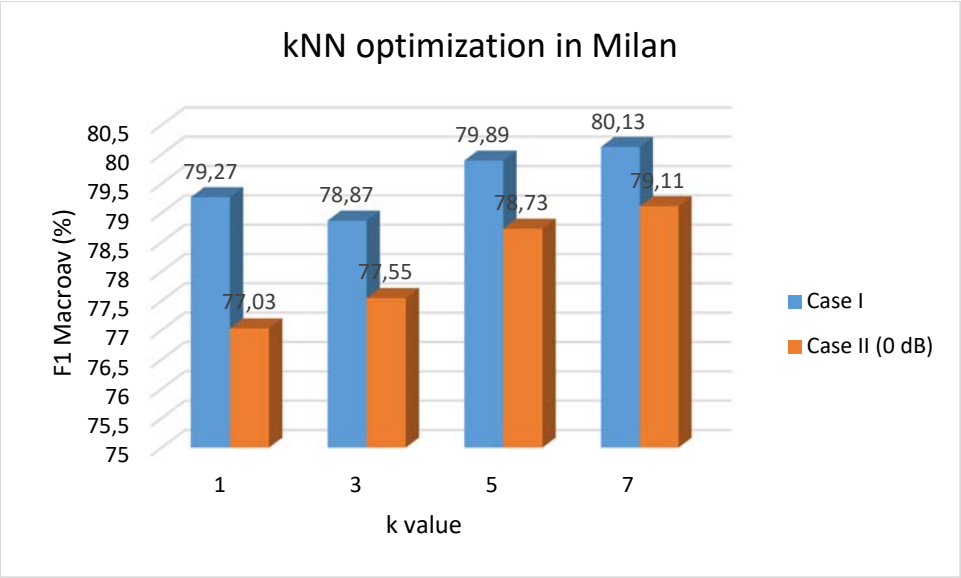


Figure 26 – Optimization results of the kNN for the Milan recordings.

4.3.4.2. Optimization of the discriminant analysis

Different discriminant functions have been studied:

- linear (fits a multivariate normal density to each group, with a pooled estimate of covariance and this is in fact the Fischer Linear Discriminant classifier),
- diaglinear (similar to linear, but with a diagonal covariance matrix estimate – naive Bayes classifiers –),
- quadratic (fits multivariate normal densities with covariance estimates stratified by group),
- diagquadratic (similar to quadratic, but with a diagonal covariance matrix estimate – naive Bayes classifiers –),
- mahalanobis (uses Mahalanobis distances with stratified covariance estimates).

In Figure 27 and Figure 28, the results of the discriminant analysis are shown. As it can be seen, the "quadratic" discriminant function is the one that attains the highest F1 macro-averaged value for both cities recordings, being the "linear" (Fischer Linear Discriminant) the second better result. Similarly to the KNN optimization, also Case I outperforms Case II in Milan while the other way around is observed in Rome.

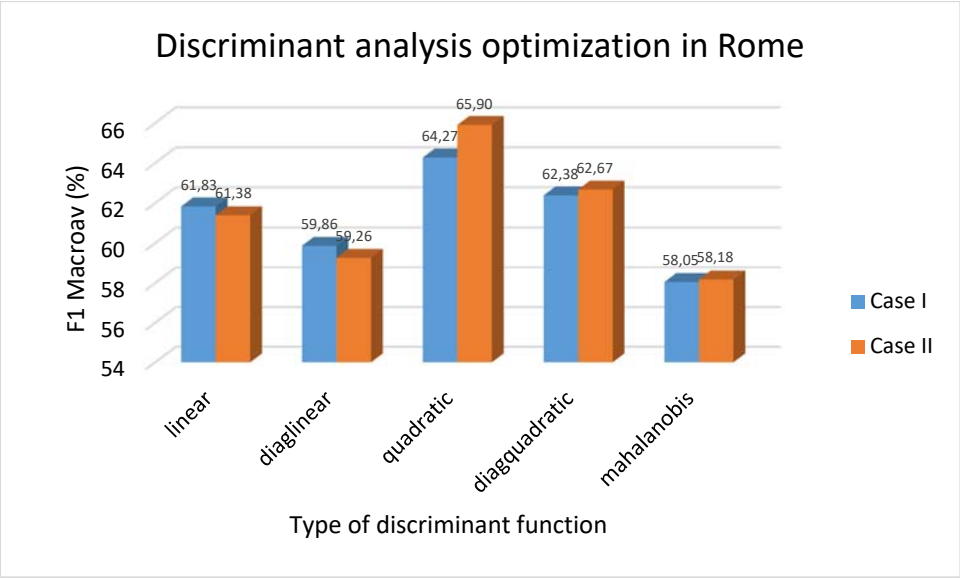


Figure 27 - Optimization results of the Discriminant analysis for the Rome recordings.

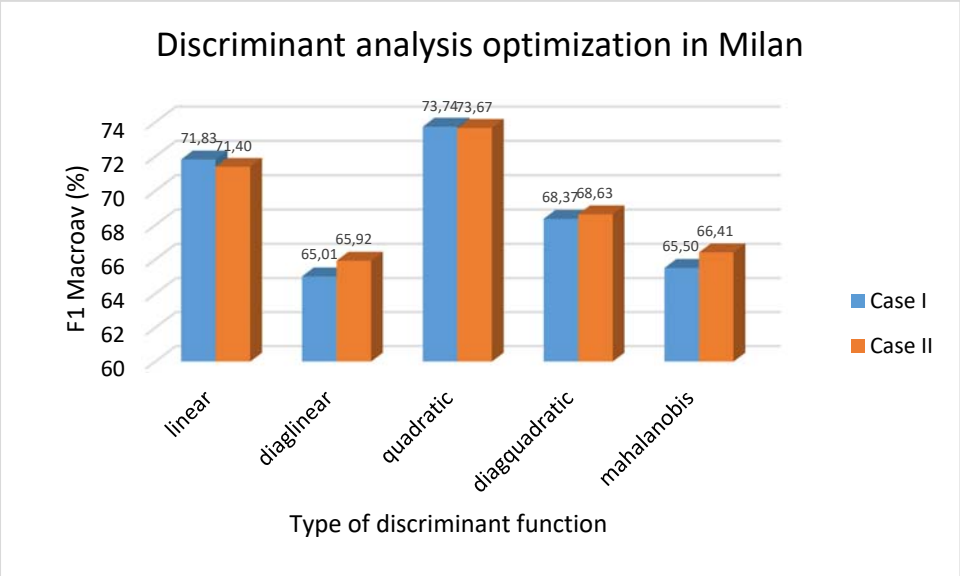


Figure 28 - Optimization results of the Discriminant analysis for the Milan recordings.

4.3.4.3. Optimization of the Gaussian Mixture Model

GMM optimization has been addressed by modifying the order of the model (i.e., the number of Gaussian components within the probabilistic model), considering the values within the set {5, 10, 20 and 30}.

In Figure 29 and Figure 30, we can see that GMM performance improves with order model, being the maximum tested order model the best one for both cities' databases. Similarly as it has been observed in previous studies, Case I is better suited for Milan recordings while Case II is better for Rome.

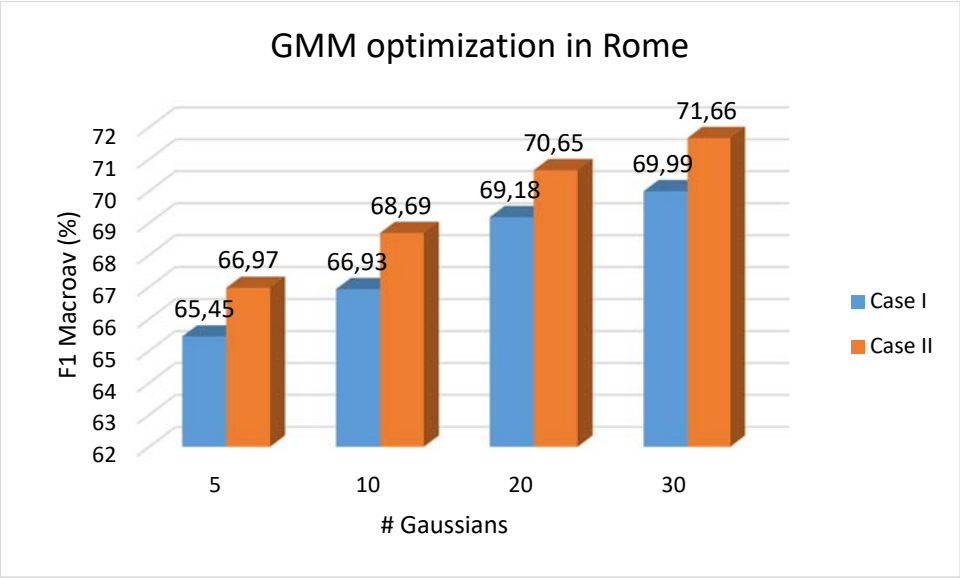


Figure 29 – Optimization results of the GMM for the Milan recordings.

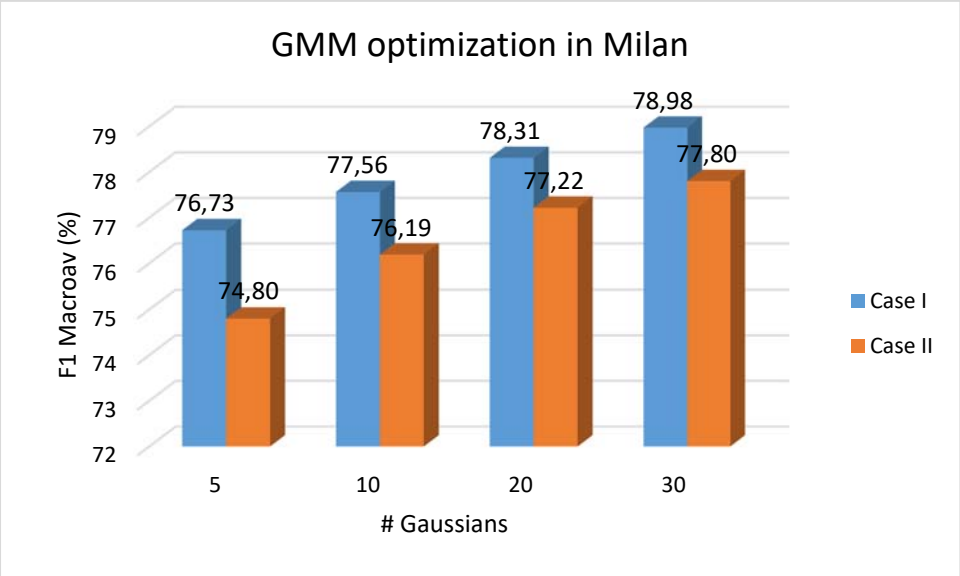


Figure 30 - Optimization results of the GMM for the Milan recordings.

4.3.4.4. Optimization of the Artificial Neural Network

The optimization of the feed-forward artificial neural network (ANN) has been addressed through modification of four parameters: number of hidden layers (#layers), number of neurons per hidden layer (#neurons), number of epochs (#epochs) and the training function (learning fcn). An exhaustive search of the optimum configuration parameters requires huge computational resources. For this reason, a simplification based on a single sweep of each individual parameter while fixing the other parameters to a constant value is addressed. This strategy is validated through a final experiment where the optimum set of parameters are used, being the obtained results compared to the rest of simulations of the ANN. In this regard, the following sweep of values and configurations have been explored:

- Optimization of the number of neurons per hidden layer: values explored are {30,60,120,240}
  - #layers = 1
  - #epochs = 200
  - Learning fcn = "trainrp"
- Optimization of number of hidden layers: values explored are {1,2,4,8}
  - #neurons = 30
  - #epochs = 200
  - Training fcn = "trainrp"
- Optimization of number of epochs: values explored are {50,100,200,400}
  - #neurons = 30
  - #layers = 1
  - Training fcn = "trainrp"
- Optimization of number of training function: values explored are "trainrp" (Resilient Backpropagation), "traingd" (Gradient Descent), "trainbfg" (BFGS Quasi-Newton) and "trainlm" (Levenberg-Marquardt).
  - #neurons = 30
  - #epochs = 200
  - #layers = 1

In Figure 31 and Figure 32, the optimization results for the number of neurons per hidden layer are shown, for Milan and Rome cities respectively. As it can be seen, working with 240 neurons attains the best results. However, Case I is the one yielding the best performance in Milan while in Rome the one that attains a highest F1 macro-averaged value is Case II.

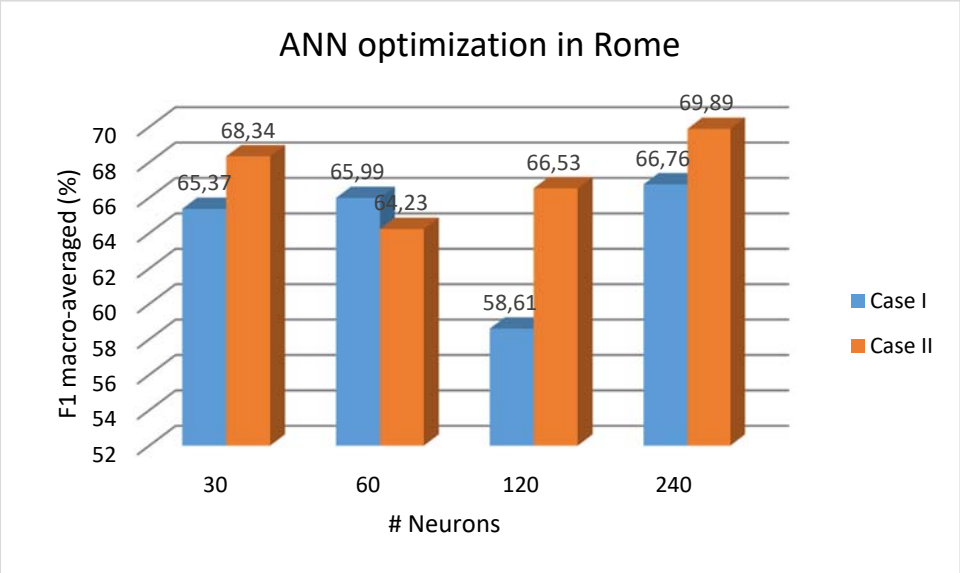


Figure 31 – Optimization results of the ANN (number of neurons in the hidden layer) for the Rome recordings.

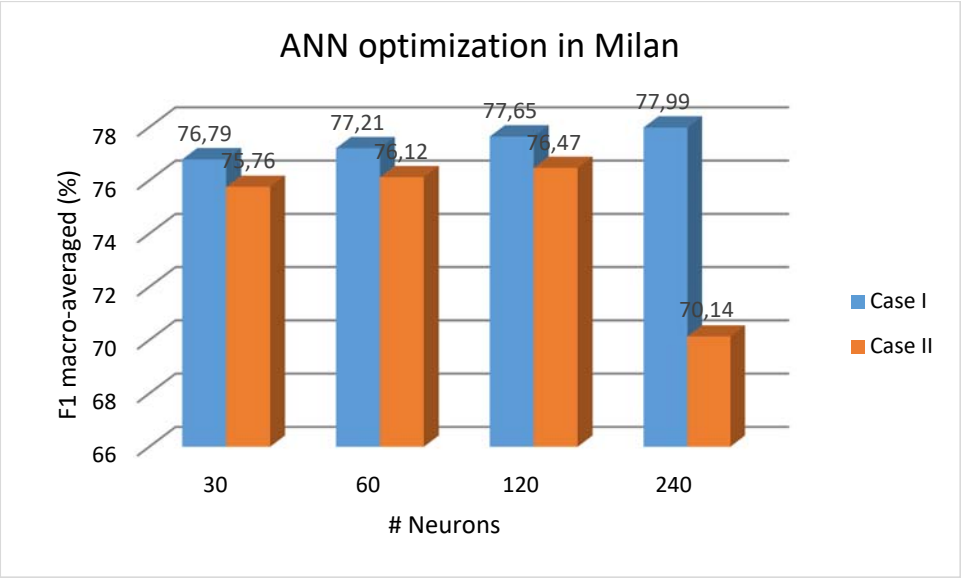


Figure 32 – Optimization results of the ANN (number of neurons in the hidden layer) for the Milan recordings.

Optimization results for the number of hidden layers of the ANN in Rome and in Milan are shown in Figure 33 and Figure 34, respectively. As it can be seen, while one layer and Case II is best suited for the Rome recordings, the best result for Milan is obtained with two hidden layers and Case I.

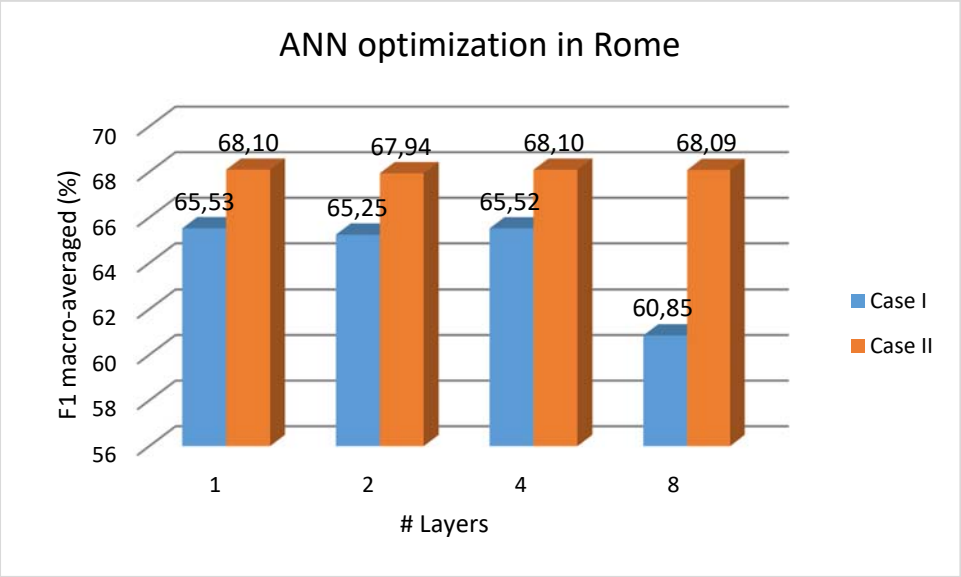


Figure 33 – Optimization results of the ANN (number of hidden layers) for the Rome recordings.

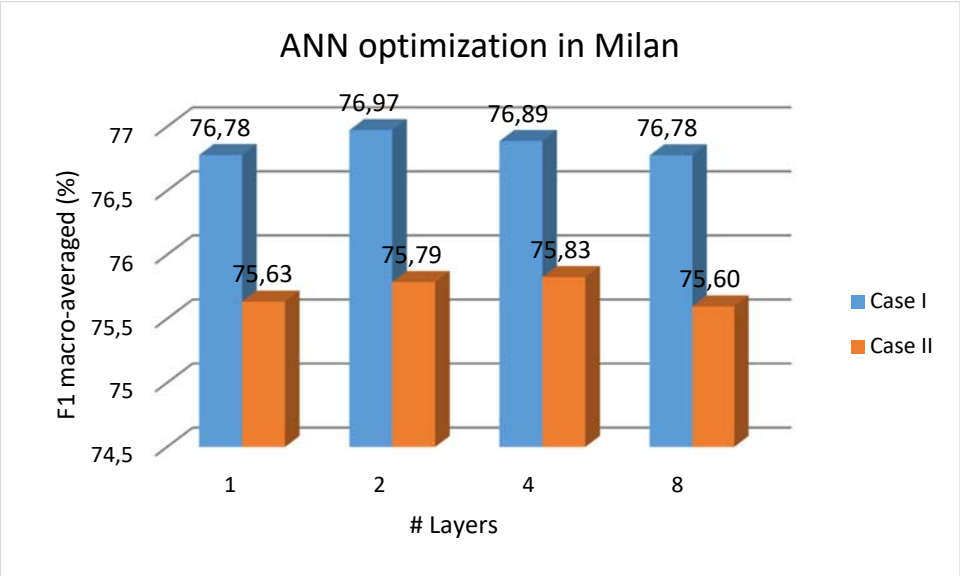


Figure 34 – Optimization results of the ANN (number of hidden layers) for the Milan recordings.

Next, the study of the number of epochs necessary to train the feed-forward neural network has been performed, considering an epoch the training of the ANN with one round of the complete training data within the 4-fold cross-validation scheme. Figure 35 and Figure 36 depict the increase performance of the ANN with the number of epochs, being 400 the value that attains a greater F1 macro-averaged for both databases (Rome and Milan). However, following the similar previous patterns in later studies, training it with audio database considering all ANE as they are without regarding its SNR (Case I) obtains better results in Milan while relabelling all ANE samples with SNR below 0 dB as RTN (Case II) is the best option for the Rome database.

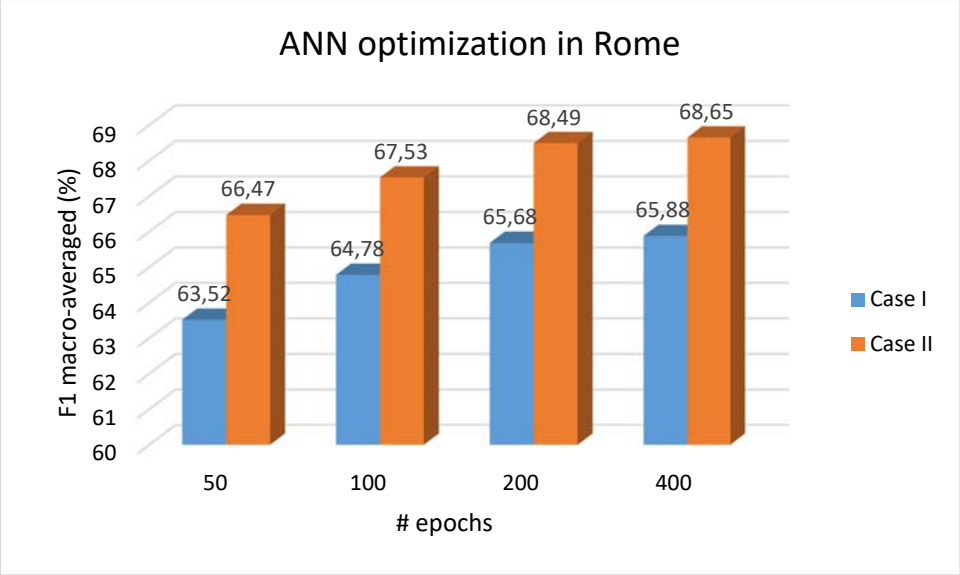


Figure 35 – Optimization results of the ANN (number of epochs for training) for the Rome recordings.

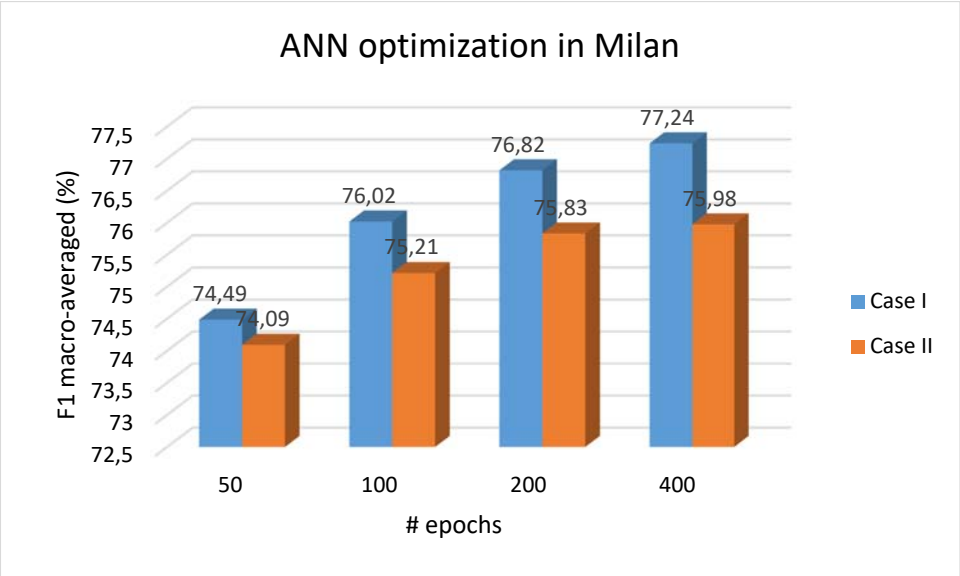


Figure 36 – Optimization results of the ANN (number of epochs for training) for the Milan recordings.

Finally, a study regarding the best suited training function for training the ANN has been carried out. As it can be seen from Figure 37 and Figure 38, 'trainrp' is the training function that attains better F1 measure in both city recordings, while the same pattern in regards the ANE samples labelling of the audio database is again observed: Case I better fits the Milan data while Case II is better adjusted for Rome.

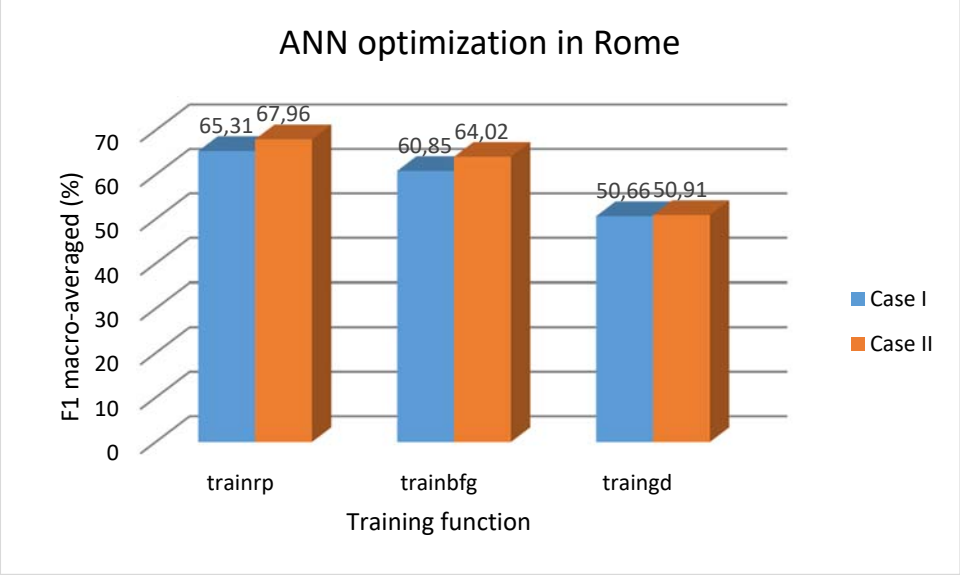


Figure 37 – Optimization results of the ANN (training function) for the Rome recordings.

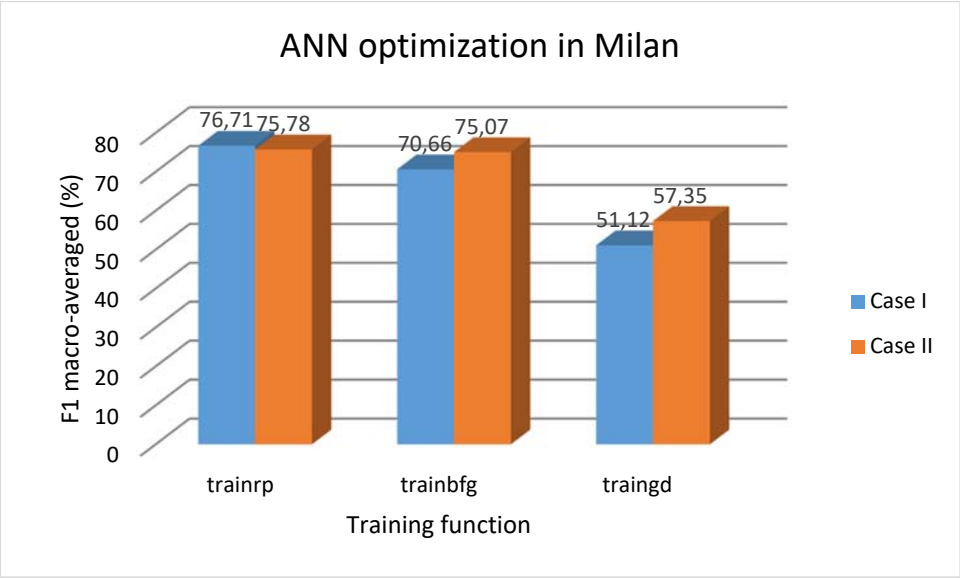


Figure 38 - Optimization results of the ANN (training function) for the Milan recordings.

As reported in the previous experiments, the optimization of the ANN show that the best configuration in each of the two city recordings are the following:

- Rome: Case II, 240 neurons per hidden layer, 1 hidden layer, 'trainrp' training function, 400 epochs. The maximum F1 macro-averaged value is 69.89%.
- Milan: Case I, 240 neurons per hidden layer, 2 hidden layers, 'trainrp' training function, 400 epochs. The maximum F1 macro-averaged value obtained is 77.99%.

To end the studies to optimize the ANN for the DYNAMAP pilot areas, a final study has been performed in which the best parameters of the individual studies have been found. Moreover, this final study has been used as a validation step to determine if the final configuration for each city obtains better results than the ones obtained in the individual adjustment of parameters.

In Figure 39 the results of this final study of the ANN are shown, in which both audio parameterizations (MFCC and GTCC) are also included. It must be highlighted that GTCC attain the best results in both cities' recordings. It must be pointed out that, due to the stochastic behavior of the ANN training, the results of the optimized ANN configuration in Rome with MFCC attain a slightly lower F1 value that the maximum value obtained in the individual sweep studies, i.e., a value of 69.36% is obtained with the optimal configuration while a value of 69.89% was obtained before, and more precisely in the study of the #neurons per hidden layer. This fact is not relevant at all, as it represents a decrease of less than 1% in performance. However, it can be seen that the final optimal configurations outperform the F1 values measured during the individual sweep of parameters, and the optimized values increase are about 1.6% in Milan and 2.17% in Rome.

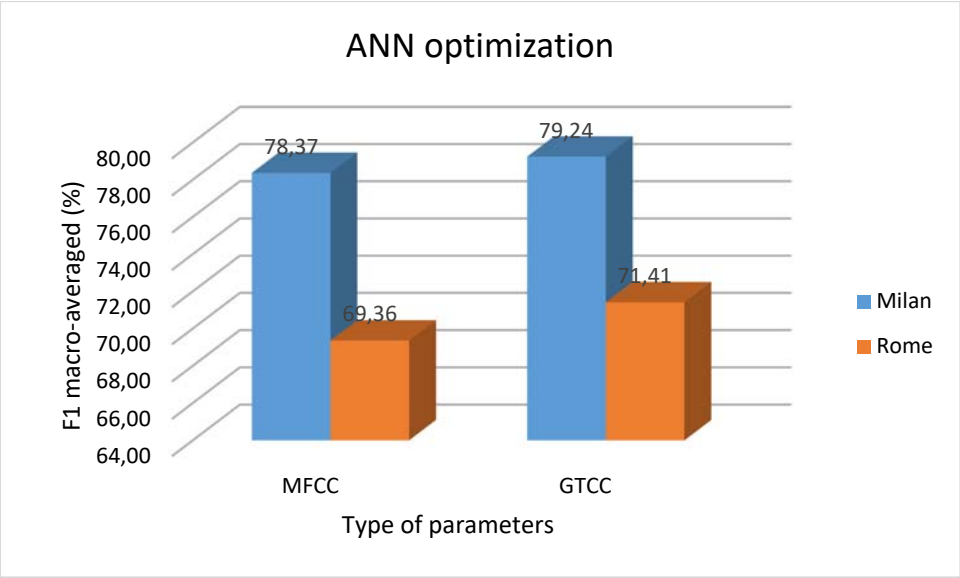


Figure 39 – Optimization results of the ANN for the best configuration explored in previous studies. Rome: case I, 240 neurons per hidden layer, 1 hidden layer, 'trainrp' training function, 400 epochs. Milan: case II, 240 neurons per hidden layer, 2 hidden layers, 'trainrp' training function, 400 epochs.

To summarize, the best configuration of the ANN before the optimization process are the following:

- Rome: case II with GTCC coefficients, and an ANN with 240 neurons per hidden layer, 1 hidden layer, 'trainrp' training function, 400 epochs.
- Milan: case I with GTCC coefficients, and an ANN with 240 neurons per hidden layer, 2 hidden layers, 'trainrp' training function, 400 epochs.

4.3.4.5. Optimization of the Support Vector Machine

The ANED based on Support Vector Machine has been optimized in terms of its kernel function, as it represents a crucial parameter for a successful training. The kernel function has been evaluated within the values of 'linear' (a linear kernel), 'rbf' (Gaussian or Radial Basis Function kernel) or 'polynomial' (3-order polynomial kernel).

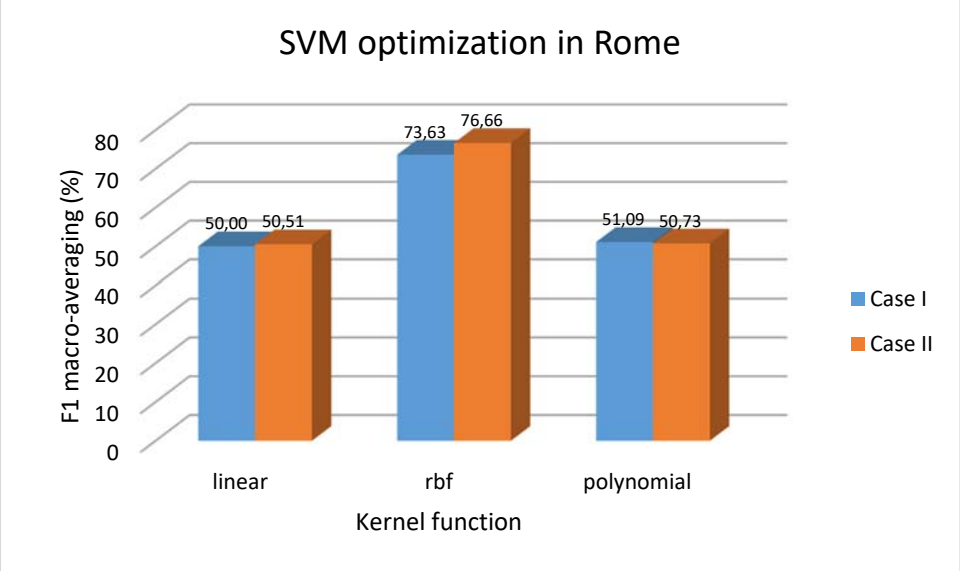


Figure 40 – Optimization results of the SVM (Kernel function) with Rome recordings.

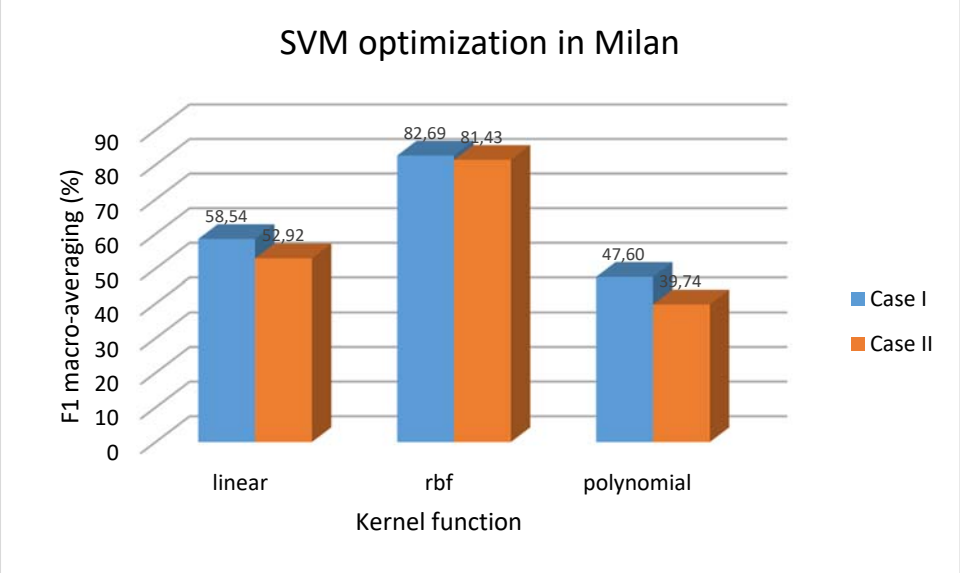


Figure 41 - Optimization results of the SVM (Kernel function) with Milan recordings.

In Figure 40 and Figure 41 the results of the SVM optimization with Rome and Milan databases are shown, respectively. As it can be observed from these figures, the Gaussian kernel obtains the best performances in both scenarios, attaining F1 macro-averaged values significantly higher than the other two type of kernels. Moreover, the same previous behavior observed for the rest of classification techniques is also obtained in regard the best case of study for each city: case II is better than case I in Rome while the other way around for Milan.

4.3.4.6. Concluding remarks

To summarize the work reported in this section, the optimum values adjusted for the different machine learners for each of the cases of study are shown in the following table.

	Rome (Case II)	**Milan (Case I)**
kNN	k = 1	k = 7
Discriminant analysis	Discriminant Fcn = quadratic	Discriminant Fcn = quadratic
GMM	#gaussians = 30	#gaussians = 30
ANN	#layers = 1 #neurons = 240 # epochs = 400 Training Fcn = trainrp	#layers = 2 #neurons = 240 # epochs = 400 Training Fcn = trainrp
SVM	Kernel Fcn = rbf	Kernel Fcn = rbf

Table 4 – Summary of the optimal configuration study for classifiers optimization.

4.2.4. Comparison of classification schemes with reduced datasets

In this section a comparison of different optimally adjusted classifiers (k-Nearest Neighbor, Discriminant Analysis, Gaussian Mixture Model, etc.) is introduced in order to evaluate their performance for each of the application scenarios (Rome and Milan databases). As previously reported, in Rome Case II is used, which means that all ANE clips with a SNR below 0 dB are considered as RTN. Otherwise, in the simulations carried out with Milan database the best solution used is Case I, which means that no ANE samples are relabeled as RTN regarding their SNR. In this study, the best configuration from the reported experiments in the previous section has been tested using MFCC and GTCC features, in order to explore which are the best combination of features and classifier. In further paragraphs, a combination of features and classifier is denoted by *Features–Classifier*, being *Features* the MFCC or GTCC while *Classifier* one of the five solutions explored.

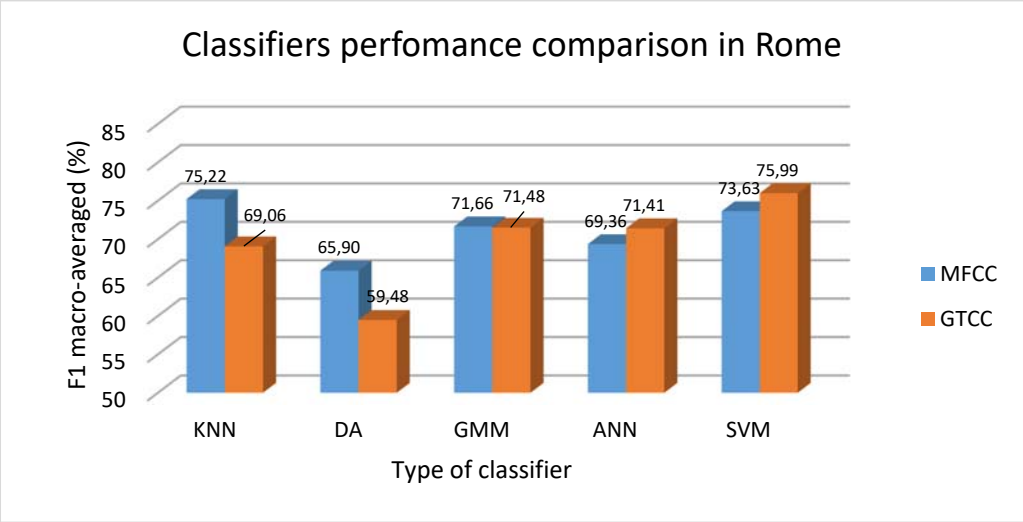


Figure 42 – Results of different optimized classifiers in the best conditions studied with Rome database (case II).

In Figure 42 the results of the different optimized classifiers is compared in the Rome scenario. GTCC–SVM obtains the best reliability with a macro-averaged F1 measure of 75.99%, followed by MFCC–kNN (75.22%), MFCC–SVM (73.63%), MFCC–GMM (71.66%) and GTCC–GMM (71.48%). Artificial neural network attains scores that, using GTCC, are very close to GMM classifier. DA obtains the worst scores for both audio features.

Figure 43 shows the results of the classifiers comparison using Milan database. In this case, SVM also outperforms the rest of classifiers, obtaining the two highest scores of the set with GTCC (83.02%) and MFCC (82.69%). Next, MFCC–kNN scheme obtains an F1 value of 80.13%, followed by GTCC-GMM (79.27%). As in the Rome studies, ANN obtains slightly lower scores than GMM with the equivalent audio features for the Milan database, and DA classifier is the one attaining the lowest F1 values.

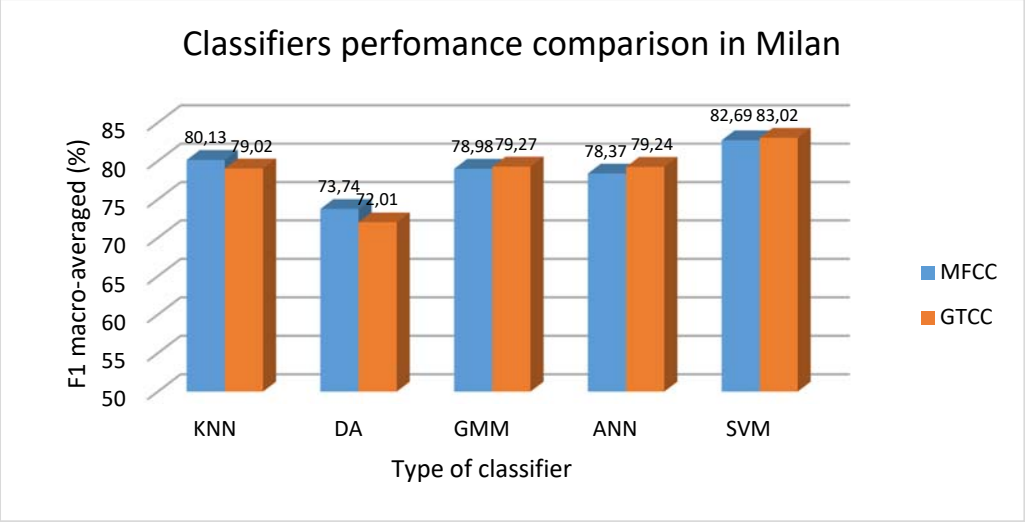


Figure 43 – Results of different optimized classifiers in the best conditions studied with Milan database (case I).

As shown in Figure 42 and Figure 43, the SVM classifier obtains the best F1 macro-averaged values in both scenarios (Rome and Milan), while Discriminant Analysis obtains the poorest ones. Otherwise, K-Nearest Neighbor, Gaussian Mixture Model and Artificial Neural Network obtain a slightly lower performance than GMM when using GTCC in both city recordings, while Discriminant Analysis obtain the lowest performances. However, MFCC-DA attains significantly better F1 values than GTCC-DA (65.9% for Rome and 73.74% for Milan).

Regarding the range of F1 values for the two application scenarios, we can see that Rome results (F1 macro-averaged values from 59.48% to 75.99%) are lower than the Milan ones (F1 macro-averaged values from 72.01% to 83.02%). This is probably due to the different nature or environmental conditions of the two scenarios, one (Rome) with road traffic noise in a highway and with barely salient anomalous events, and the other (Milan) with different traffic conditions (from low density to dense traffic conditions) within an urban environment and with anomalous events with a more diversity of saliency.

4.2.5. Computational cost analysis

One of the key aspects of the ANED implementation for the high capacity sensors is that the implemented system is subject to real-time performance. For this reason, a study of the time consumption with a desktop PC (Intel(R) Core(TM) i7-Q720 CPU @1.60GHz, 4 GB of RAM and O.S. Windows 7) has been conducted.

As the studies provided here have been performed using Matlab© software, the computational load of the different classification techniques have been evaluated using difference of time stamps using the time monitoring functions *tic* and *toc*. The total time need for learning and for testing in the 4-fold cross-validation scheme for the assessment of each configuration has been retrieved during the simulations, which has been averaged across parameters (MFCC and GTCC) and scenarios (Rome and Milan databases).

Classifier	Training time (s)	Testing time (s)	% Testing time/ DA testing time
KNN	6,63	1235.41	165252.8%
DA	1.74	0.75	100%
GMM	2436.39	3.36	449.7%
ANN	63333.39	7.89	1055.9%
SVM	5798.03	27.60	3691.7%

Table 5 – Comparison of computational load of classifiers in terms of time consumption in a Windows-based desktop PC platform and using Matlab © software.

The results of the time consumption performance is shown in Table 5. As it can be seen, Discriminant Analysis classifier obtains the best performance both for training and testing. K-Nearest Neighbor classifier attains the second position in the training time consumption, because in fact only a simple database preparation process is needed, while it demands for extremely high resources at classification time (testing). Gaussian Mixture Models obtain the second best position for testing, while being 4.5 times slower than the Discriminant Analysis counterpart. Artificial Neural Network and Support Vector Machine obtain the lowest scores, being the first the slowest when training, and the second the most demanded time consumption algorithm during the testing phase.

As a conclusion, GMM is the classifier that attains a good tradeoff between accuracy and computational cost while DA is the most competent in terms of computational cost in the testing stage. ANN can be discarded for the subsequent implementation, as it is a classifier that obtains significantly lower classification accuracies and computational performance. Finally SVM is the best solution in terms of classification reliability but it demands for almost 40 times more computational resources than the DA classifier.

4.3. Implementation an ANED prototype for real-time performance.

Once the Matlab © simulations have been shown and the respective results have been analyzed, the explanation of a first ANED implementation prototype for the high capacity sensor is hereafter described. The implementation has been performed with the support of the DYNAMAP project partner Blue Wave, who has shared with us the development framework in order to perform a soft integration process, minimizing risks and communication issues.

The implementation has been developed using the Eclipse IDE running on a GNU/Linux Debian distribution. To work in a similar environment, the OS is hosted in a Vmware-powered virtual machine, but still in absence of the acoustic sensors (were not available during the development of this work). The hardware details of the system are given in Table 6.

Hardware framework configuration	
Processor	Intel(R) Core(TM)2 Duo CPU E6750 @ 2.66GHz
Memory	3 GB
	2 GB (Allocated in the virtual machine)
System cache	64 bits

Table 6 – Hardware framework configuration characteristics.

The parameterization and classification code has been entirely programmed in C language. Nevertheless, the main routine and one of the libraries (the one that evaluates operation of the algorithm) are programmed in C++.

As far as the algorithm is concerned, it is composed by a set of libraries allowing the classification task of an audio file stored in the hard disk drive. Some of them are obtained from an external source and some others have been explicitly developed for the ANED prototyping. A list of the used libraries followed by a brief explanation is described below.

- **Synthesis Toolkit (STK)**: a set of processing and audio synthesis algorithms implemented in C++ (see (Cook, y otros, 1995)). Only the extension FileRead is used in this project for the task of reading an audio file and importing its properties (channel number, sampling rate, etc.).
- **fftw3**: programmed in C-language, carries out an efficient calculation of the DFT (Discrete-Fourier Transform), see (Frigo, y otros, 2005). It is used in the parameterization of new input frames in the ANED system. This library has two main advantages. Firstly, there is no need for the frame to be a power of two, in respect to other similar libraries that would require a previous zero-padding. Also, the dynamic adaptation of the algorithm depending on the hardware and the nature of the calculation allows to run the most efficient implementation of the DFT algorithm.
- **readingFiles**: it reads a set of text files used to configure the system (frame duration, overlapping, etc.) and stores values that need to be calculated once. These files pretend to reduce computational costs. Examples of these files are the models trained in Matlab © environment and the filter banks implemented during the GTCC or MFCC calculation. This library has been implemented by our team.
- **classify**: it contains the algorithms in charge of the frame classification. In this first prototype, the methods implemented are KNN, GMM and FLD. This library also includes the subroutines ensuring the proper calculation of the method itself, e.g. the Euclidian distance in the case of KNN and the PDF calculation in the case of GMM. This library has been implemented within the ANED prototyping.

Apart from these libraries, the main procedure (readingwav2_gtcc), besides reading the input acoustic data, it also implements the function that calculates the GTCC and MFCC parameters of the input frames.

In order to have a first insight of which methods are more efficient in terms of its computational cost, C++ standard library *chrono* allows timing the seconds spent in each part of the algorithm; i.e. the parameterization and classification of each frame, in our case. This information is useful for assessing the possible real-time behavior of the ANED algorithm.

Another aspect to take into account is the decision time of the classifying algorithm. The system has been implemented to classify each input frame at every 30ms. However, the DYNAMAP project has been stipulated that the decision time of the system (regarding accepting or discarding a L_{eq} measure depending on the ANED output decision) will be set to 1s, so a number of 33 consecutive frames without overlapping are needed to complete this analysis time. The final system outputs a majority vote taking into account the frame-based decisions (one decision per frame) contained within 1 second of audio.

Figure 44 shows the flow diagram of the implemented ANED prototype.

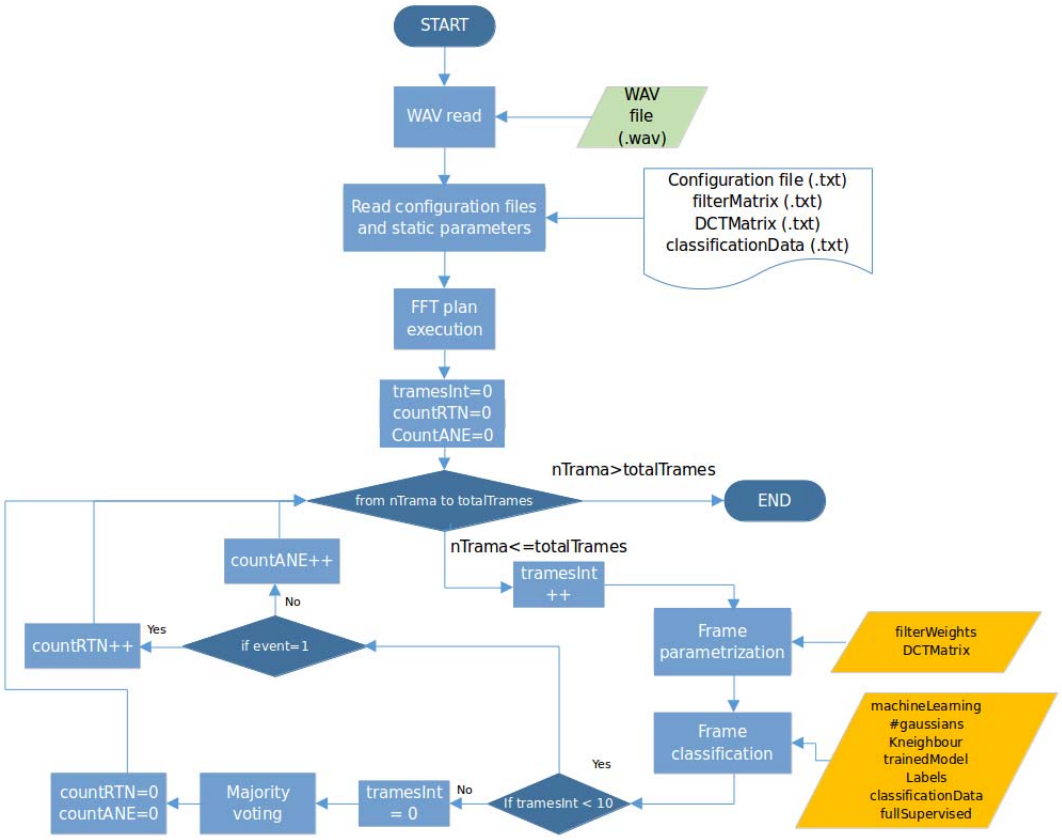


Figure 44 – Flow diagram of the designed ANED prototype.

The algorithm goes through the following steps:

1. The audio file is read and stored in an object of the FileRead class, using the STK library. Also the sampling rate, channel number and sample length are stored in the new object.
2. The configuration files are read, and their content determine the following points: concerning classification, the type of classifier to be used (KNN, GMM or FLD have been included in this version); concerning parameterization, the type of computed audio features (GTCC or MFCC). Also other parameters can be set through configuration files (e.g., frame length, the number of nearer neighbors in the KNN method, etc.).
3. Once the configuration is loaded, depending on the chosen parameterization and classification method, a supplementary file can be loaded containing the static data needed (e.g., the acoustic model associated to the selected classifier). All the needed methods belong to the readingFiles and are called inside the readStuff function located in the main procedure.
4. The FFT (more efficient than the DFT) plan is called to start the hardware adaptation and the needs of the project. Previously to its execution, several parameters must be configured and saved to a fftinfo struct containing the following fields:
    - **insize** [integer]: number of samples of the input filter (non-processed samples).
    - **outsize** [integer]: number of samples of the output filter (processed samples).
    - **usedsize** [integer]: window sample length applied to the input signal.
    - **inbuf*** [float pointer]: points to the input buffer.

- **outbuf*** [complex pointer]: points to the output buffer, both real and imaginary components belong to the float type.
- **win*** [float pointer]: points to the coefficient vector of the window.
- **plan** [fftwf_plan (library defined)]: it is needed to start the FFT plan.

The plan is called with the setplan method defined in the main procedure.

5. The variables needed accounting for the number of frames classified as RTN (countRTN) and ANE (countANE) and the number of frames used by the majority vote (tramesInt) are all initialized to zero; i.e. being 30ms the frame length and the decision being taken every second, the vote comparison will be carried out every 33 frames.
6. We enter the loop starting at nTrama=1 until the total number of frames (totalTrames). At each iteration, the counter tramesInt is incremented, the frame is parameterized and classified. While the counter of frames is still below ten, either the countRTN or the countANE is incremented. When the tenth iteration is reached, a decision is taken depending on the number of frames labeled as RTN or ANE, afterwards, the counting variables are reset and the loop restarted until the end of the audio is reached (totalTrames).

The next section details the implementation of the parameterization and classification methods, emphasizing the optimizations allowing a posterior real-time execution. In addition, data concerning the execution time of the different methods are exposed.

4.3.1. Frame-by-frame Parameterization and Code Optimization

The parameterization function is called cepstralCoefficients and it is located in the main procedure, readingwav2_gtcc. This has been implemented based on the two type of audio features explained in Section 3.6. However, unlike the Matlab© implementation, several optimization considerations have been taken into account to speed up the execution time.

As mentioned above, using a set of configuration parameters, a set of data is calculated to model the respective cepstral coefficients (MFCC or GTCC). Amongst all the needed data, some will remain static once the configuration parameters are definitely defined, so repeating these calculations should be a waste of time and a loss of efficiency. These data belong to the filter-bank transfer function and the matrix allowing the DCT calculation following the next structure:

- **Filter banks**: the size of this matrix is nFFT x BandsN, where nFFT is the number of points containing the transfer function of each filter of the MFCC or GTCC filter bank as well as the FFT of each audio frame prepared to be analyzed, and BandsN is set to 48, the number of bands considered for both MFCC and GTCC implementations.

- **DCT matrix**: the size of this matrix is CoefsN x BandsN, where CoefsN is the number of cepstral coefficients obtained at the output (set to 13 in all the previous studies) and BandsN is the number of bands of the filter-bank (48).

In order to optimize the computational cost, these matrices are stored in text files. This way, the final algorithm will implement two functions reading these two files and saving in memory the above-mentioned static variables. This will be carried once during the software execution, just before the real-time operation of the ANED algorithm.

Besides, to increase even more the efficiency of the algorithm, filter bank information has been compressed deleting redundant data offering non-useful information during the cepstral coefficients calculation. This compression process is carried out by means of a predefined threshold from which the data structure of the transfer function is stored. Also the inferior and superior indices where the transfer functions overpass this threshold are stored for each filter bank. That way, when the FFT is applied to a frame, only those calculations that really contribute to the recognition performance are performed.

Additionally, a function to calculate the compressed computational cost in relation to the non-compressed has been implemented. Taking into account the fact that the dimension of the non-compressed matrix would be nFFT x BandsN, when the aforementioned compression process is applied to the matrix, the next computational cost save value is obtained:

$$Comp.Cost\ (\%)_{comp.} = \frac{\sum_{i=1}^{BandsN} nSamples_i}{nFFT\ \ BandsN} 100$$

being $nSamples_i$ the total frequency samples belonging to the i subband frequency band taking into account the predefined threshold.

In the next table, a study of the computational cost optimization related to the filter bank operation within the cepstral feature computation is shown.

Compression	No	Yes	Yes	Yes
Threshold	-	0.00001	0.0001	0.001
Computational Cost (%)	100	22.27	18.45	12.46
$F1_{ANE}$ (%)	68.97	69.30	69.17	69.03
$F1_{RTN}$ (%)	77.85	78.96	78.80	78.42
$F1_{MACROAV}$ (%)	73.41	74.13	73.98	73.72

Table 7 – Computational cost reduction study of the filter bank matrix optimization in function of the predefined threshold. Simulation carried out with Milan data using 10% RTN, GTCC parameterization, FLD classification method, 30ms frame length and 50% overlapping. As shown, the best results are marked in green, with a 0.00001 threshold, obtaining a saving of 77.73% in relation to the computational cost of the non-compressed data and improving the F1.

As it can be observed, compression of the filter bank data does not affect significantly the recognition rate, so we confirm that it can used to optimize the algorithm for its real-time operation. Once the filter bank matrix is compressed, both the filter bank and the DCT matrices are stored in a text file using a specific Matlab function (createFiles).

4.3.2. Frame-by-frame Classification

As aforementioned, classification is carried out using a set of methods belonging to the classify library, programmed in C-language. In the first ANED prototype three classification schemes have been incorporated: KNN, FLD and GMM.

Concerning the methods coded in C language, the implementation has been translating the main routines from Matlab © but avoiding some calculations that can be performed out of the real-time ANED execution through storing into a text file those variables that will remain static for all the frames. These files refer to several training models created in the Matlab IDE, i.e., average vectors and matrices, covariances, GMM weighting coefficients, KNN and FLD matrices and, finally, the label vectors.

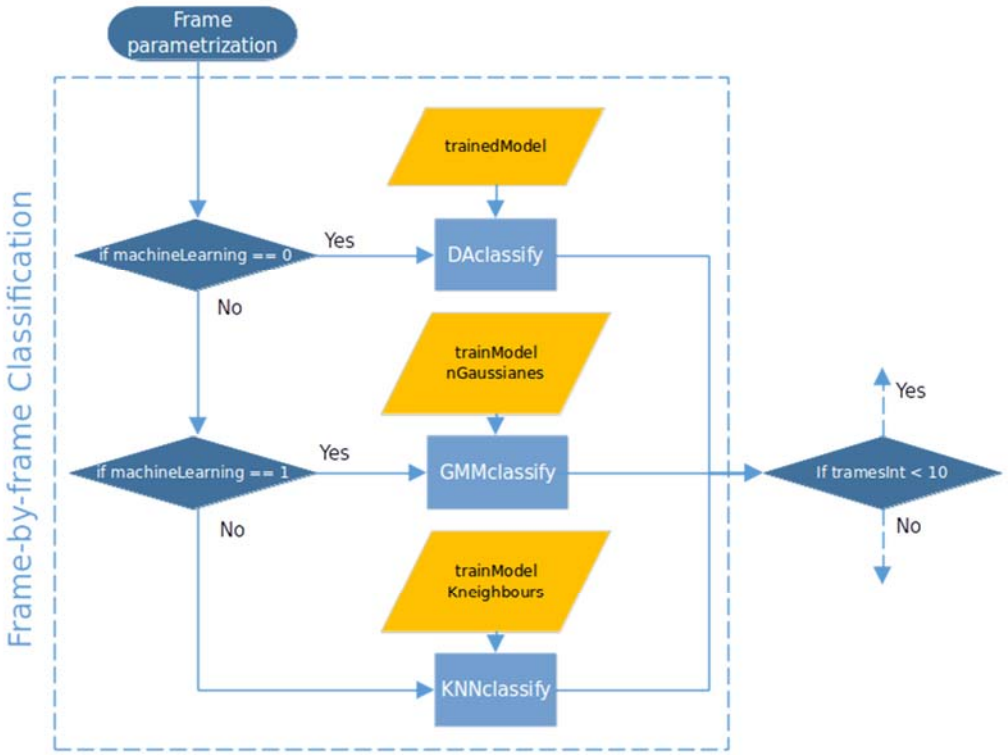


Figure 45 – Flow diagram of a frame classification.

Figure 45 shows the generic flow diagram of a frame classification. The machineLearning variable is one of parameters extracted from the configuration file. This will determine the executed classification method, being 0 for the FLD (DAclassify), 1 for the GMM (GMMclassify) and 2 in case of the KNN (KNNclassify). The other parameters belong to the training method and to particular data sets for each classification method.

4.3.3. Computational Cost Analysis

In this section, a computational cost analysis of the ANED prototype is presented. This study has been conducted in the virtual machine, so this preliminary study may differ from the final results executed in the sensor, which are not available during the writing of this deliverable.

The C++ library *chrono* was used in this study, which stores the system clock just before the classification process. When the frame-by-frame classification is finished, the clock is stored again and the initial time is subtracted from the final one. This operation returns the elapsed time of all the process in seconds, whose division by the frame number gives the average time spend in parameterization and classification in each frame.

The next table shows the classification result of a 3-second audio file using an analysis frame length of 30ms sampled at 48 kHz sample rate. All classification methods (FLD, GMM and

KNN) and types of audio features (MFCC and GTCC) are evaluated. As shown in Table 8, the most efficient configuration, concerning computational cost, is the green-highlighted row: FLD classifier with MFCC parameters. This combination needs an average of 0.13 milliseconds per frame resulting the total time 3.9ms for the entire audio file, which meets real-time requirements.

Classifier	Param.	Number of frames	Elapsed time (ms)	Elapsed Time / Frame (ms)
FLD	MFCC	30	3.9	0.13
FLD	GTCC	30	11	0.36
GMM	MFCC	30	4	0.13
GMM	GTCC	30	12	0.40
KNN	MFCC	30	2.557	85.23
KNN	GTCC	30	2.263	75.43

Table 8 – Computational cost of the algorithms taking part in the classification process. The GMM algorithm is using 30 Gaussian functions. The KNN classifier is using the three nearest neighbours for the final decision.

Although the best combination is MFCC parameters in FLD learning algorithm, the final implementation will require a proper balance between computational cost and recognition accuracy. As has been shown from studies presented in Section 4.2.4, GMM obtains good performances with 30 Gaussian components (F1 macro-averaged values of 71.48% in Rome, and 79.27% in Milan, both using GTCC audio features). From the computational cost analysis provided in Table 8 we can see that the GMM still obtains pretty good results, which are very close to the FLD classifier. Regarding the parameterization, the elapsed time per frame is less or equal to 0.4ms, which is much lower than the frame length (30 ms). KNN classifier obtains the worst results but still the time needed for processing a frame is under the frame length. However, the operation of the final system must take into account other important tasks that have been not evaluated in this study nor the real hardware platform where the system will run.

4.4. Conclusions

In this section the design, implementation and assessment of the ANED algorithm version developed for the high computation capacity sensors has been addressed. From the preliminary studies two learning strategies were proposed and assessed (supervised and semi-supervised) with synthetic mixtures of road traffic noise (RTN) and anomalous noise events (ANE). This previous studies were subsequently confronted using one classification approach (FLD) with real audio data obtained from the recording campaign. The obtained results revealed the complexity of working on real-life scenarios (Rome and Milan), obtaining significantly lower performances. Furthermore, the superiority of the semi-supervised approach with synthetic data was not revalidated in the real-case scenario, and then the supervised approach was selected in further studies. In addition, the analyses of the results obtained for the Rome and Milan recordings led to the conclusion that the highly unbalanced nature of the training audio dataset together with the diversity of ANE and SNR values were the main causes of that performance decrease.

A selection and reduction process carried over the major class (RTN) was designed based on clustering techniques in order to address on the main issues detected when training the ANED classifier with real-life recordings. This process was designed to optimally choose the number

of clusters that better represent the feature dataset distributions after reducing the data significantly. Having detected the relevance of the learning dataset structure in the final classification performance, two cases of study (Case I and Case II) were also defined in order to asses two criteria about defining what is labelled as ANE as a function of its saliency (measured in terms the contextual SNR). While Case I considers all feature vectors labelled as ANE to be included in the classifier as ANE disregarding their SNR, in Case II only those feature vectors with SNR ≥ 0 dB (ANE-to-RTN SNR) are considered as ANE, while the rest are relabeled as RTN.

The optimization of the ANED was subsequently addressed with the aim of having reduced the dimension of the RTN or RTN+BCK class (for Rome and Milan, respectively), and up to five different classifiers (KNN, DA, GMM, ANN and SVM) have been assessed, optimized and compared in terms of classification reliability and computational cost with the real-life recordings obtained. The complete study was addressed using Matlab © software and run on an advanced desktop PC platform. Although being SVM the best classifier in terms of reliability in both scenarios, the two following classifiers with highest scores were finally selected for the real-time implementation, besides an FLD classifier.

As a final step, a real-time ANED algorithm prototype has been implemented in C language using the proper development environment (a virtual machine) to assure the correct integration within the acoustic sensor platform with high capacity computation developed by Blue Wave. The procedures of audio parameterization and classification have been implemented by using optimization techniques taking into account the real-time requirements. The developed code has been assessed in terms of computational cost on the virtual machine. The system can be configured according to the training stage from the Matlab code. However, the approach allows for enough flexibility to include future improvements that could incorporate further labelled databases or new classification schemes, to name a few. Finally, the ANED algorithm implementation can be tuned for each specific application scenario, which in this project are the two pilot systems located in Rome and Milan.

5. DEVELOPMENT OF THE ANED ALGORITHM FOR THE LOW COMPUTATION CAPACITY SENSORS

In this section, an alternative version of the ANED algorithm is explored with the aim of analyzing the viability to adapt the previously detailed version of the algorithm to be able to run with low computation capacity sensors of the DYNAMAP project by means of implementing a reduced computational complexity approach. The complexity reduction is attained by proposing a simplified classification scheme based on a signal level threshold obtained directly from the L_{eq} of the acoustic signal or through a basic band energy filtering process. However, the hardware requirements of low computational capacity sensors were not available during the work performed presented in this section. Then, there is a lack of information regarding if the obtained results can suit the hardware requirements of this lower complexity ANED solution. Instead, a comparison between the obtained reduced complexity ANED version and two selected versions of the high computational capacity sensors ANED algorithm is performed at the end of this section.

In Figure 46, the block diagram of the version of the ANED for the low computation capacity sensors is depicted. The Signal Level Computation block obtains a variable that is directly related to the acoustic signal level at the global level or per bands, by including a signal-filtering step. The classification process consists of a simple comparison between the signal level measure and the optimized threshold γ. With the aim of obtaining reliable output RTN/ANE labels the threshold (γ) is computed with an optimization process based on audio databases analysis.

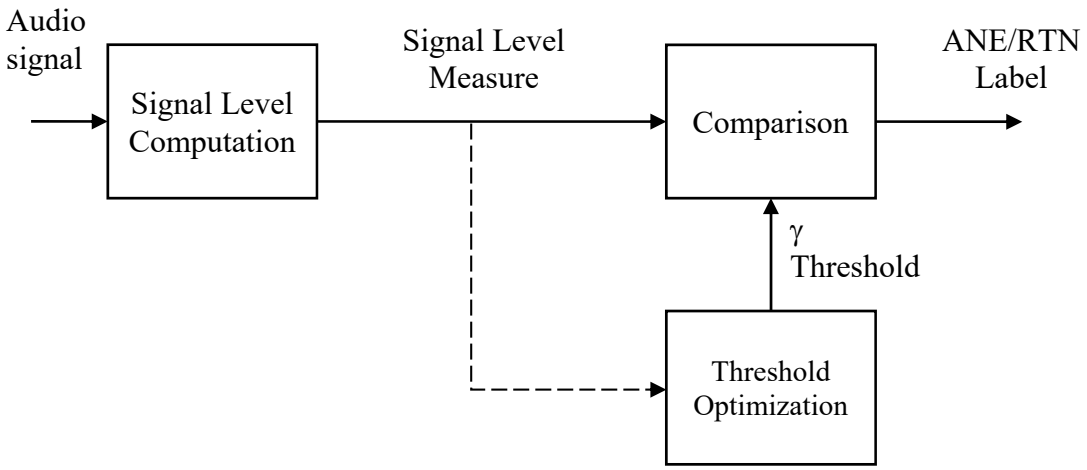


Figure 46 – Block diagram of the ANEDlite version for the low computation capacity sensors.

The rest of this section is organized as follows. In section 5.1, a study to detect significant differences between road traffic noise and anomalous events in both pilot areas (Rome and Milan) is performed by using a clustering analysis. The study is conducted as a previous analysis step before proceeding with the design of the ANED version for the low capacity computation sensors, especially when considering signal filtering before computing the signal level. In section 5.2, the investigations related to the first ANED approach for the low capacity computation sensors based on directly comparing the A-weighted equivalent noise level (L_{eq} (A)) with an optimized reference threshold are described. The obtained results show poor reliability, and then its feasibility for a real prototype is not subsequently considered. Then,

section 5.3 presents the version of the algorithm that incorporates a filtering stage prior to the signal level computation, and improves significantly the results of the previous section. In section 5.4 the results of the assessment of the selected (filter-based) configurations with the recorded audio databases are discussed. To resume, a computational cost analysis of the feasible solutions is performed in section 5.5 and the conclusions derived from the analysis of the considered two ANED variants for the low capacity computation sensors are explained in section 5.6.

5.1. Clustering analysis at spectral level

A clustering study based on the computation of the spectral envelopes of the recorded audio signals windowed at 30ms has been conducted. The purpose of this study is to evaluate to what extent there are significant spectral differences between RTN and ANE for each city database for specific frequency bands.

Two type of analyses have been conducted, both based on the spectral envelope parameterization using the gain coefficients of the 48 subbands of the Gammatone Cepstral Coefficients (GTCC):

- RTN: automatic clustering using a Gaussian Mixture Model, sweeping the number of clusters from 1 to 6.
- ANE: supervised clustering based on the subcategories defined during the audio corpus labelling process (see section 3.2). In this case, the ANE subcategories have been grouped by manual inspection and comparison of their spectral envelopes.

In Figure 47, the results of the clustering analysis for the RTN of the Rome database for 1, 2, 4 and 6 clusters are shown, where the mean frequency envelopes are drawn in solid lines while the mean ± std are depicted using dashed lines. Different colors are used for different clusters. In Figure 48, the same analysis for the Milan city database is depicted. As it can be observed, RTN exhibits different patterns in Rome than in Milan, which is clearer in Figure 49.

In Figure 50 and Figure 51, the cluster analysis for the ANE class is depicted for Rome and Milan database, respectively. In Rome, four clusters have been defined: cluster 1 (including “brak”, “stru” and “trck” ANE subcategory labels), cluster 2 (including “busd”, “door”, and “musi” ANE subcategory labels), cluster 3 (including “horn” and “sire” ANE subcategory labels) and cluster 4 (including only the ANE subcategory label named “peop”). Like in the previous figures derived from the RTN analyses, here also mean frequency envelopes are drawn in solid lines while the mean ± std are depicted using dashed lines, besides using different colors for different ANE subcategories.

From these results, on the one hand, it is worth noting that ANE and RTN exhibit similar patterns for each city database. On the other hand, subtle but not negligible differences can be observed when comparing statistics of the RTN and ANE clustered spectral envelopes across the 48 frequency bands.

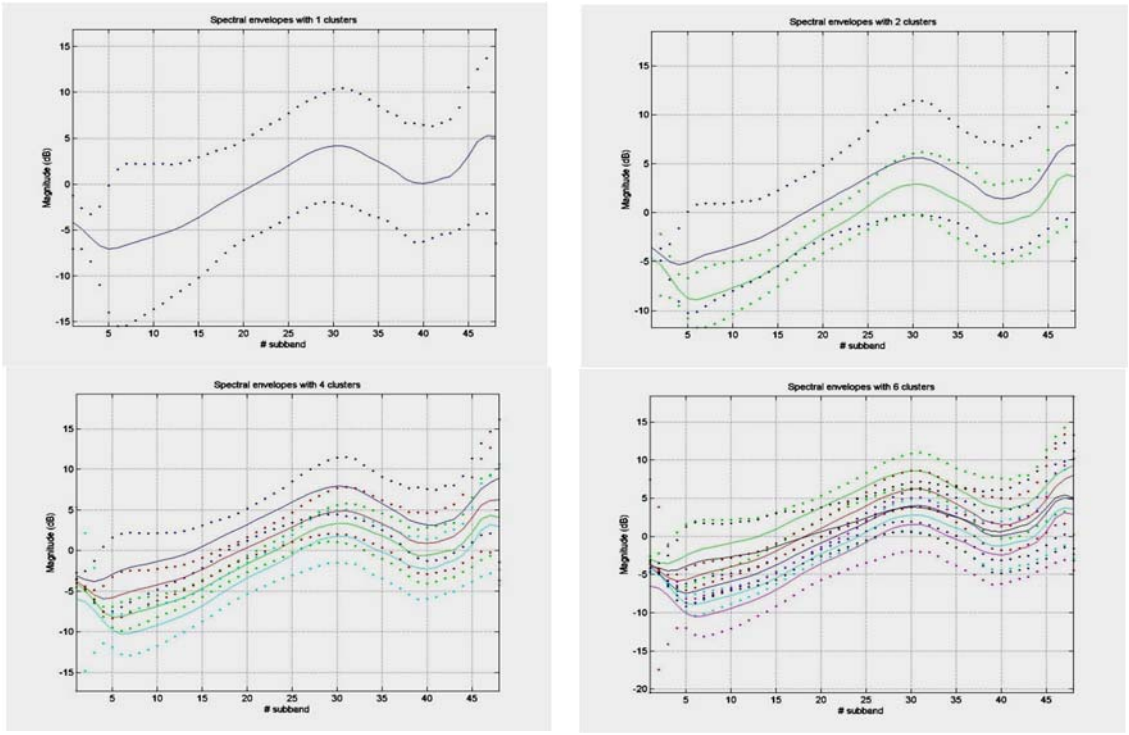


Figure 47 – Clustering analysis at spectral level of the RTN class in the Rome recordings

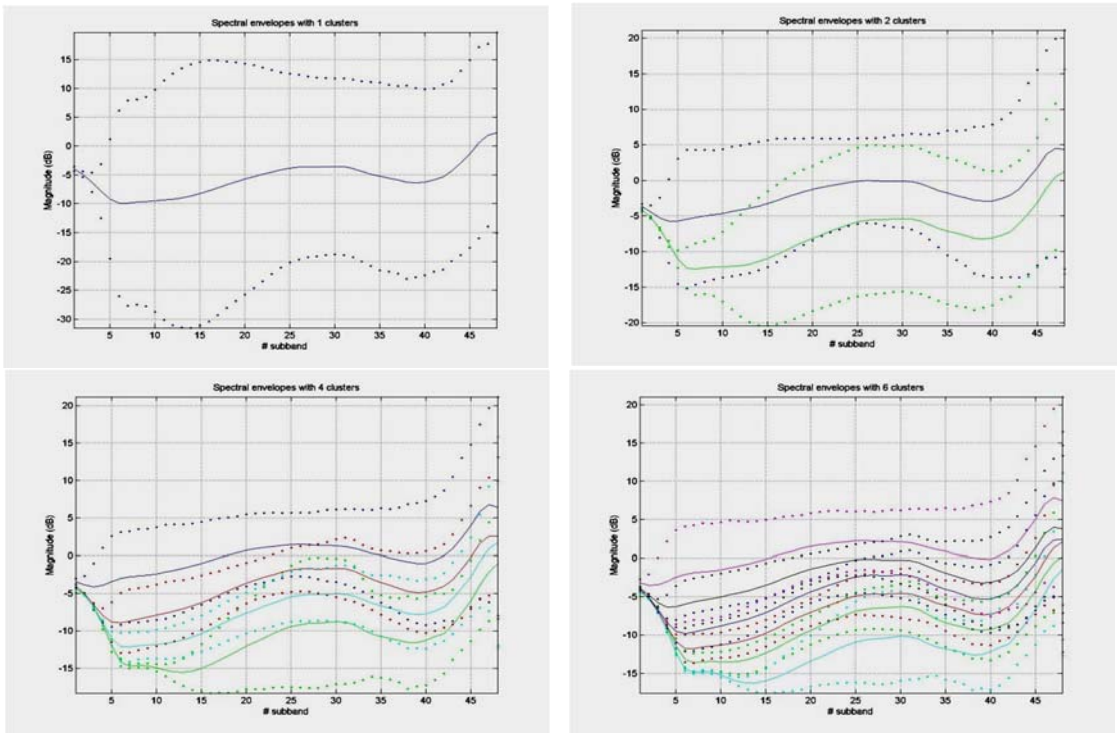


Figure 48 – Clustering analysis at spectral level of the RTN class in the Milan recordings

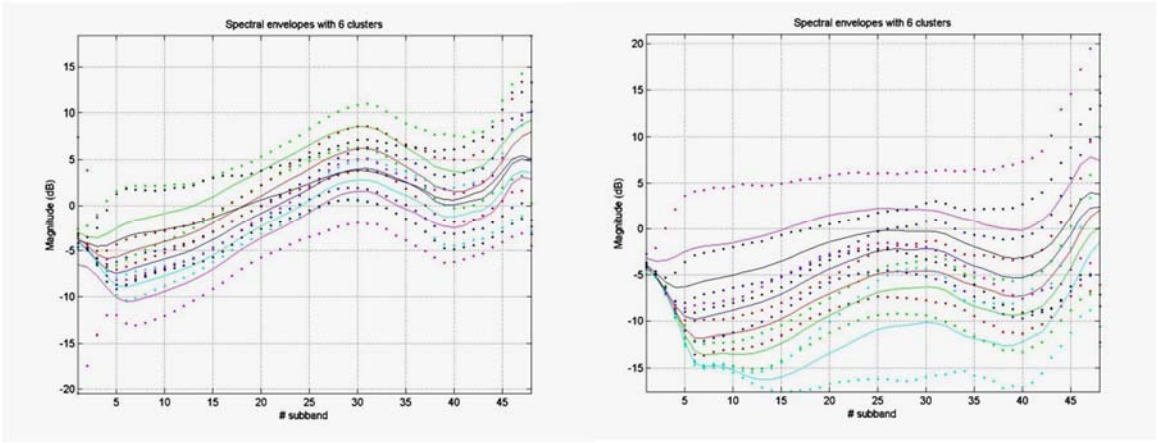


Figure 49 – Comparison of RTN 6-cluster analysis at spectral level for the two city recordings.

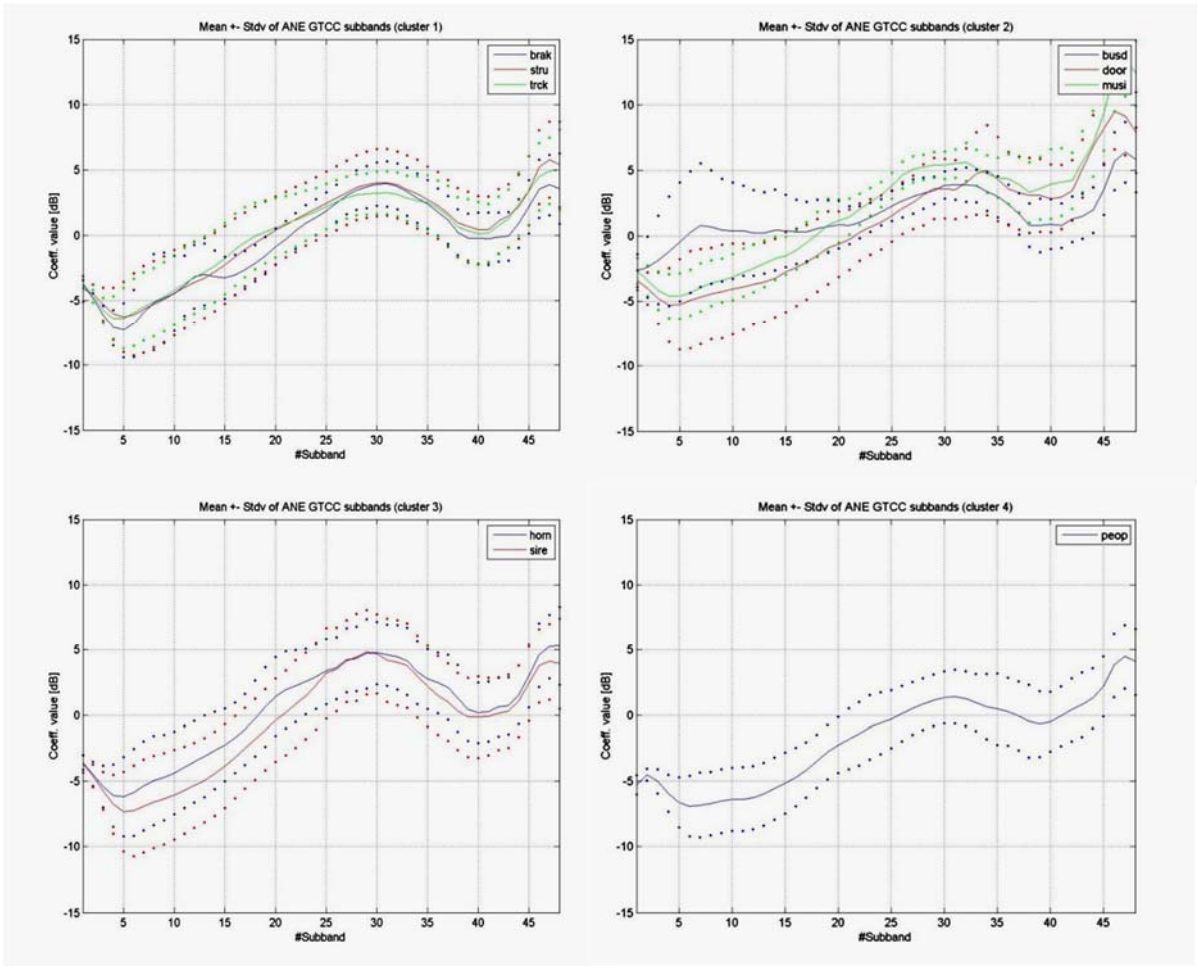


Figure 50 – Clustering analysis at spectral level of the ANE class in the Rome recordings.

However, it is hard to determine which are the GTCC bands where spectral differences between RTN and ANE that should be selected to derive a simple yet effective classifier (a low-cost solution with higher reliability than using only L_{eq} computed from the wide-band audio signal spectrum). For this reason, an in-depth quantitative analysis based on the identification of the frequency bands yielding lower error probability of classification is faced in the next section.

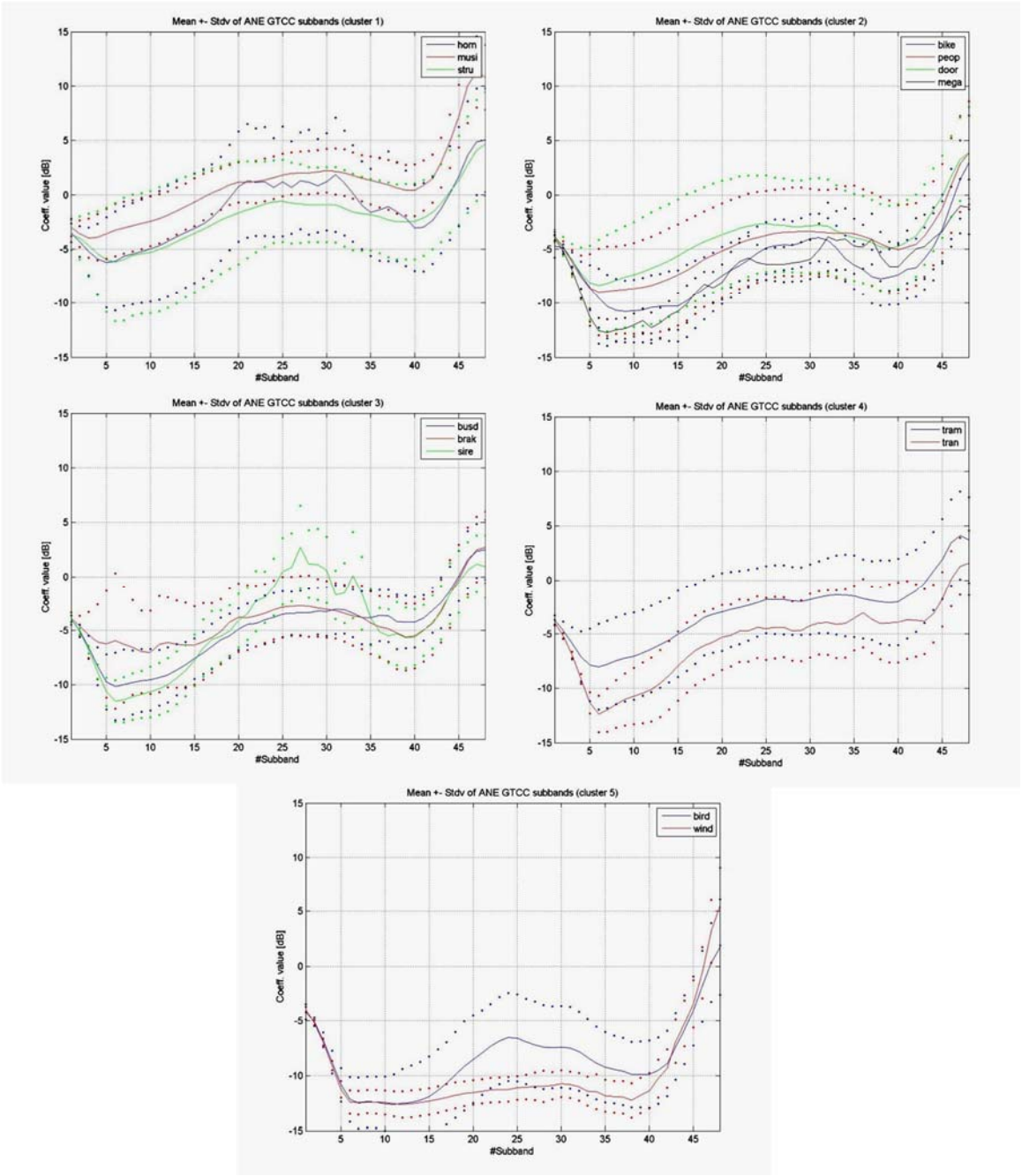


Figure 51 – Clustering analysis at spectral level of the ANE class in the Milan recordings.

5.2. Leq analysis with real data from onsite recordings

In this section, we describe the results from the analysis of a first ANED proposal for the low capacity computation sensors based on the computation of L_{eq} (A) to discriminate between road traffic noise and anomalous events. This approach is denoted as the Direct Threshold-Detector version of the ANED (DTD-ANED), in which the Signal Level Computation block of Figure 46 is composed of the A-weighting process followed by a simple equivalent noise level computation using a moving average FIR filter of 30 ms length. The A-weighted signal level is then converted to decibels using the $10\log_{10}(x)$ function.

The whole recorded database during task "B.3.1. Pilot areas inspection and recording", including Rome and Milan recordings, has been used for learning and testing. In order to reduce the computational load of the ANED algorithm instead of using the parameterized database (with MFCC or GTCC), the level-based ANED is feed with $L_{eq}(A)$[8] considering an integration time of 30 ms (the same value of the frame size used for audio parameterization in the high capacity sensors version of the ANED). As reported in section 3.3, the complete database was labeled considering two classes: road traffic noise (RTN) and anomalous event (ANE). Then, the computed L_{eq} (A) signals have been labelled according to the same time references obtained from the raw audio data, in order to design and validate the proposed low capacity sensors version of the ANED

In the following study, we analyze to what extent the L_{eq} (A) threshold can be adjusted locally for each pilot site location (the reader is referred to Section 3.2 for the description of each of the specific recording locations for Rome and Milan). This approach can be considered as an optimistic approach, because optimizing locally the threshold will provide a solution better matched to a specific recording conditions (i.e., distances from noise sources to sensor can be different in each location, and so the measured noise levels). However, also global analyses have been conducted considering *(i)* all data coming from the Rome recordings, *(ii)* all data coming from Milan recordings, or *(iii)* the full audio recorded database.

Two main studies have been conducted:

- **Statistical-based analysis of the level-based ANED**. In this first study, we have analyzed the performance of this ANED variant at different levels (from local to global) from a statistical approach in which PDFs are estimated from the labeled L_{eq} (A) values. Accuracies are directly computed from the obtained error probabilities.

- **DTD-ANED results**. In this second study, the direct threshold-detection version of the ANED algorithm is assessed when running the threshold detection process across the different project locations and obtaining accuracies and F1 Macro-averaged (averaging RTN and ANE precision and recalls) measures.

5.2.1 Statistical-based analysis of the level-based ANED

A binary detector is designed through computing the L_{eq} (A) threshold that minimizes type I (detecting ANE when RTN) and type II (detecting RTN when ANE) errors jointly. Then, type I and type II density functions have been computed, and the optimum threshold is obtained by determining the crossover point between the two functions. The threshold is computed at three different levels:

- Locally for each pilot site location.
- Globally for all locations in Rome recording campaign.
- Globally for all locations in Milan recording campaign.
- Globally for all the recordings coming from Rome and Milan.

The total error probability is obtained as the mean value of type I and type II error probabilities, assuming balanced RTN and ANE categories. This way, the possible mismatch of the results

[8] The $L_{eq}(A)$ value is computed using the free Matlab Continuous Sound and Vibration Analysis toolbox written by Edward L. Zechman. Link: http://www.mathworks.com/matlabcentral/fileexchange/21384-continuou

due to unbalanced datasets is avoided. The accuracy is computed for each case of analysis as the complement of the total error probability, taking into account both error I and II types.

In Figure 52, the type I and type II error probability density functions for the two pilot site recordings are shown. The optimum L_{eq} (A) level threshold is depicted as a straight vertical line, just in the crossover point between the two error probability functions.

Figure 52 –Example of the type I and type II error probability density functions for two pilot site recordings.

The results of the level-based ANED in which the decision threshold is locally optimized for each specific site location is shown in Table 9. As it can be observed, some locations show quite good accuracies (e.g., site Rome 12) while many others exhibit lower values (e.g., there are 12 out of 24 projects with accuracies lower than 60%).

Site location	P total	Accuracy
Rome 1	0,31024	68,98%
Rome 2	0,37782	62,22%
Rome 3	0,43337	56,66%
Rome 4	0,29081	70,92%
Rome 5	0,3476	65,24%
Rome 6	0,38676	61,32%
Rome 7	0,43701	56,30%
Rome 8	0,31434	68,57%
Rome 9	0,71129	28,87%
Rome 10	0,45534	54,47%
Rome 11	0,53388	46,61%
Rome 12	0,18409	81,59%
Milan 1	0,61282	38,72%
Milan 2	0,8372	16,28%
Milan 3	0,31239	68,76%
Milan 4	0,50155	49,84%
Milan 5	0,52845	47,16%
Milan 6	0,47399	52,60%
Milan 7	0,22192	77,81%

Milan 8	0,30579	69,42%
Milan 9	0,50743	49,26%
Milan 10	0,51929	48,07%
Milan 11	0,42736	57,26%
Milan 12	0,58981	41,02%

Table 9 – Results of statistical-based analysis of the level-based ANED optimized specifically for each site location in Rome and Milan cities.

Table 10 shows the results of the level-based ANED in which the decision threshold is optimized for each specific city (Milan or Rome) by considering all its corresponding recordings, and for the global optimization considering the complete set of recordings. It can be observed that global-based approaches attain lower accuracies, while Rome results outperform those obtained in Milan, which is precisely the opposite result obtained with the high capacity version of the ANED.

	P total	Accuracy (%)
Rome	0,4248	**57,52**
Milan	0,5190	48,10
Global	0,4917	50,83

Table 10 – Results of the level-based ANED optimized globally for each city recordings (Rome and Milan) and globally for all the recordings coming from Rome and Milan.

5.2.2 Direct Threshold-Detector version of the ANED

In this second study, the DTD-ANED version developed for the low capacity sensors (see Figure 46) is assessed. To that effect, the threshold detection process is computed across the different project locations. The performance of the system is shown in terms of accuracies and F1 Macro-averaged (averaging RTN and ANE precision and recall values). Following a similar approach of the high capacity version of the ANED, both Case I (all ANEs are considered regardless their SNR) and Case II (only ANEs with SNR lower 0 dBs are considered to be RTN) are taken into account.

γ (dB)	Case I	Case II	Case I	Case II
Project Name	F1 Macro-averaged (%)		Accuracy (%)	
Rome 1	58,18	**60,68**	69,07	**69,11**
Rome 2	56,10	**57,31**	62,14	**62,28**
Rome 3	54,07	**55,79**	**56,66**	**56,66**
Rome 4	**61,77**	**61,77**	**70,95**	**70,95**
Rome 5	**56,93**	**56,93**	**65,28**	**65,28**
Rome 6	56,19	**56,77**	61,36	**61,45**
Rome 7	52,96	**53,18**	56,21	**56,23**
Rome 8	**59,08**	59,01	**68,74**	68,67
Rome 9	48,06	**49,33**	46,30	**46,58**
Rome 10	63,07	**64,46**	81,90	**81,92**

Rome 11	41,28	**55,14**	38,61	**44,01**
Rome 12	47,25	**59,85**	47,18	**51,76**
Milan 1	**52,33**	52,07	**52,57**	52,35
Milan 2	**74,79**	74,05	**77,83**	77,19
Milan 3	62,51	**62,86**	69,35	**69,49**
Milan 4	**49,65**	49,62	**49,23**	49,21
Milan 5	**49,02**	46,26	**47,99**	47,74
Milan 6	**53,99**	52,97	**57,25**	56,88
Milan 7	44,38	**46,45**	41,00	**41,63**
Milan 8	23,91	**24,78**	16,28	**16,47**
Milan 9	**62,12**	61,85	**68,75**	68,50
Milan 10	**49,90**	49,52	**49,78**	49,59

Table 11 – Results of the DTD-ANED specifically optimized for each pilot site location and for each of the cases (Case I: all ANEs are considered regardless their SNR, and Case II: ANEs with SNR below 0 dBs are considered to be RTN).

Table 11 shows the results of the threshold-detector version of the ANED optimized specifically for each site location and for each of the cases. As it can be observed, the results obtained from the Rome recordings exhibit better results for Case II, while Case I show the higher results with Milan recordings (rows 13 to 24).

In Figure 53 a global perspective of the previous results averaging the Macro-averaged F1 values for each city (Rome and Milan) is shown. The global results draw a different picture from the one obtained by the ANED version for the high capacity sensors. In this case, Case II obtains better results than Case I for the two scenarios, showing those from the Rome site show better performance than the ones from Milan. However, the differences are not significant being the obtained Macro-averaged F1values quite low.

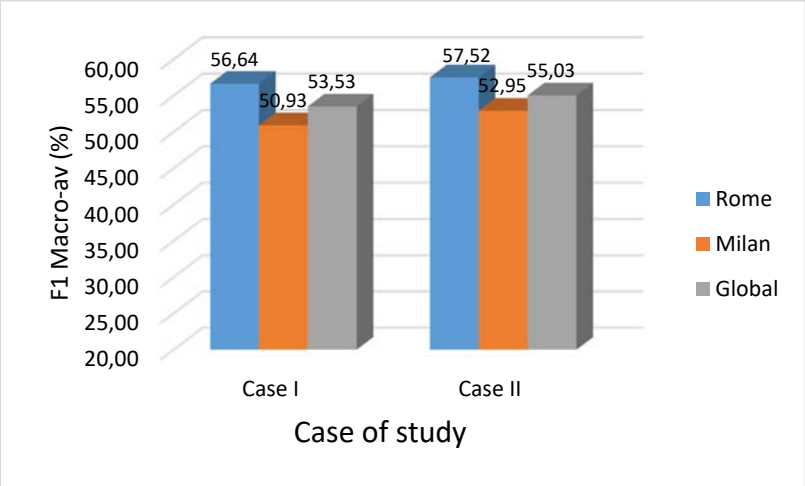


Figure 53 – Global perspective of the quantitative results for the DTD-ANED.

The main conclusion of the study of the threshold-detector version of the ANED is that conducting audio classification purely based in signal levels, only accuracies barely higher than 50% (i.e., random chance probability in a binary classification) are obtained. From these results, we discard this approach for further analyses.

In the next sections we study if exploiting spectral difference between the audio signals could be considered to increase these figures with an affordable computation complexity for the low computation capacity sensors.

5.3. Frequency subband analysis with real data from onsite recordings

In this section, a new version of the ANED algorithm for the low capacity sensors is explored by considering a filtering process previous to the signal level computation (see Figure 46, where the filtering process is included in the Signal Level Computation block). The aim of this approach is improving the results obtained with the previous version where the full frequency band signal has been used for the A-weighted equivalent noise level computation. It is named Subband Threshold-Detection version of the ANED, or shortly STD-ANED.

As a proof-of-concept, the proposed STD-ANED has been conceived taking advantage of the parametrized databases (see Section 3.6) which consists of cepstral analysis that include subband signal level computation before the final transformation to the time domain (using the DCT). Therefore, the STD-ANED could take advantage of the information about the audio signal spectral components, which in turn is represented in terms of a bio-inspired approach. This fact has been reused in the studies presented in this section, so as to obtain valuable information about the frequency subbands that could attain better discriminative properties between road traffic noise (RTN) and anomalous noise events (ANE).

First, a preliminary study based on synthetic mixtures has been performed to set the general approach, and then, the analysis of the two audio databases (Rome and Milan) has been conducted to investigate which are the most appropriate frequency subbands to adjust the optimized STD-ANED version. To end with these studies, different simulations have been run to assess the performance of the best configurations found for each scenario (Rome and Milan).

5.3.1. Threshold optimization and preliminary synthetic study

First, a preliminary study based on synthetic mixtures has been performed in order to establish the general criterion to define the separability of each subband from the previous statistical analysis of both RTN and ANE in each of the proposed scenarios (Milan and Rome). The defined criterion will be the base of the Threshold Optimization block of Figure 46.

The proposed analysis is designed in order to obtain a general picture of the approach before defining which frequency subbands are the ones that contain the most discriminant information to differentiate the input audio. Once the separability is obtained at the subband level, a higher level of decision can be established merging contiguous subbands that contain the most significant discriminant information between RTN and ANE.

Given a pair of *Probability Density Functions* (PDF) distributions, $pdf_1(x) = p(x|class\ 1)$ for Class 1 and $pdf_2(x) = p(x|class\ 2)$ for Class 2, which describe the signal level at a certain frequency band B of both RTN and ANE respectively, two types of total error functions are computed based on two hypotheses:

- *Hypothesis 1*: class 1 has signal levels that are above the ones of class 2. For this hypothesis, the total error function is defined as $p_e^{c1>c2}(x)$, being *x* the sound level at the frequency band *B*.

- *Hypothesis 2*: class 2 has signal levels that are above the ones of class 1. For this hypothesis the total error function is defined as $p_e^{c1<c2}(x)$, being *x* the sound level at the frequency band *B*.

Both error functions $p_e^{c1>c2}(x)$ and $p_e^{c1<c2}(x)$ depend of the signal level threshold x that will be used to discriminate between two classes, and are defined as follows:

$$p_e^{c1>c2}(x) = P_1 \int_{-x_{min}}^{x} pdf_1(x)dx + P_2 \int_{x}^{x_{max}} pdf_2(x)dx$$

$$p_e^{c1<c2}(x) = P_2 \int_{-x_{min}}^{x} pdf_2(x)dx + P_1 \int_{x}^{x_{max}} pdf_1(x)dx$$

where P_1 and P_2 are the a priori probabilities of classes 1 and 2, respectively.

As it can be seen in the previous equation, the total error functions $p_e^{c1>c2}(x)$ and $p_e^{c1<c2}(x)$ are computed as the sum of two type errors (type I: classify 1 when class 2, type II: classify 2 when class 1), and their computation differs in the way the PDFs are integrated (from the minimum signal level to the current x signal level or from this point to the highest observed signal level).

For each frequency band *B*, once the two functions are computed, the minimum error is computed for both functions and a final decision about which is the most feasible hypothesis:

- Accept Hypothesis 1 if $\min\left(p_e^{c1>c2}(x)\right) < \min\left(p_e^{c1<c2}(x)\right)$

- Accept Hypothesis 2 otherwise.

Moreover, the reliability of the final decision can be defined as the minimum value $\min\left(p_e^{c1>c2}(x), p_e^{c1<c2}(x)\right)$, and the optimal signal level threshold for the analyzed subband can be defined as the signal level which attains the given minimum in the corresponding total error function that attains that minimum, that is:

$$x_{th} = \begin{cases} Arg\min\{p_e^{c1<c2}(x)\} & if\ \min\left(p_e^{c1>c2}(x)\right) > \min\left(p_e^{c1<c2}(x)\right) \\ Arg\min\{p_e^{c1>c2}(x)\} & otherwise \end{cases}$$

Synthetic examples

A simple Matlab script has been implemented in which the PDF of two categories are defined, following a 1-D Gaussian mixture model with two Gaussian components each. The selection of two Gaussian components is based on a simple model that allows a simple manipulation of the shape of probability density functions with the aim of evaluating different synthetic

examples. The dimensionality of the model (1D) has been chosen because we want to study separability of both classes for each frequency subband, and then only one signal level dimension is here analyzed.

In the following figure we can see an example in which *Hypothesis 1* is the most feasible attaining a total error probability of 0.28333. In this case, a significant overlap between class 1 and 2 explains the high value of the error probability. The *Cumulative Density Functions* (CDF) depicted in Figure 54 correspond to the integral functions in the definition of the error functions. As can be seen, the total error function (shown in red color) that attains the minimum error probability value is the one attributed to *Hypothesis 1*, e.g., $p_e^{c1>c2}(x)$. This also makes sense from the perspective of the visualization of the two PDF functions (in blue and green colours), because the function of class 1 is clearly displaced to the right while that of the class 2 is more placed at the left side. This says that class 1 has level values above the class 2 levels.

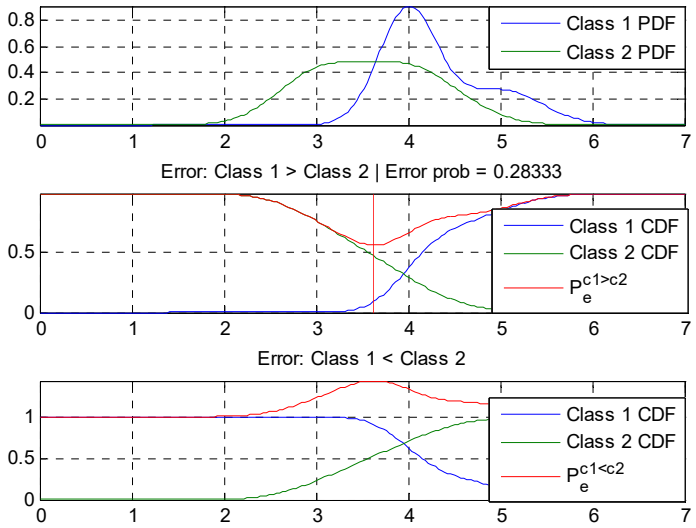


Figure 54 – Synthetic example 1.

In the next figure, we can see another synthetic example where *Hypothesis 2* is the most feasible attaining a total error probability of 0.078254. In this case, the lower overlap between class 1 and 2 with respect to the previous example explains the lower value of the error probability compared to the previous synthetic example.

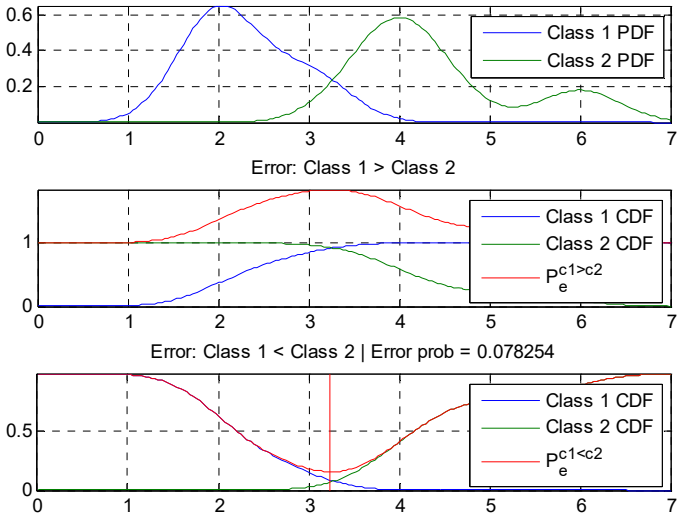


Figure 55 – Synthetic example 2.

5.3.2. PDF computation

In this section the resultant PDF functions for both classes (RTN and ANE) are analyzed using data from the two pilot site recordings (Rome and Milan). The databases parameterized using GTCC features have been used as the main input data for the analyses. PDF functions have been estimated for each of the 48 frequency bands of the GTCC, obtaining two-dimensional functions (2D-PDF) or matrices of dimensions 200x48. For each spectral band, a vector of 200 samples of the corresponding PDF function is obtained (200 signal levels within the range of observed values of the GTCC filterbank outputs, considering both Rome and Milan recordings). Following the same approach as for the ANED version designed for the high capacity sensors, the two cases of study (case I and case II) is also considered hereafter (see Section 4.2.3 for more details).

In Figure 56 and Figure 57 the 2D-PDF matrices corresponding to the Rome recordings are depicted, for Case I and Case II respectively. In each figure, the frequency subband index corresponding to the GTCC filterbank is labelled in the X-axis of the figure, while the output signal level at each subband is depicted in the Y-axis in dBs. The figures colormap is bluer for lower probabilities while tend to warm colors (red) for higher probabilities. For aesthetic reasons, 2D-PDF visualizations have been scaled at each frequency band dividing each PDF by its maximum value in order to obtain a more comprehensible plot. It is worth noting the high similarity of 2D-PDFs of both ANE and RTN plots. However, slight differences can be also noted at the lower half of the subbands dynamic range (e.g. from index 5 to 25), while the higher frequency region is more similar at both plots. In regard the differences along the case of study, it can be seen that differences at the lower frequency region between ANE and RTN are still more evident in Case II than in Case I. This fact can be explained because the ANEs recorded in Rome with low SNR (e.g. with noise levels lower than the noise level of the surrounding RTN) present frequency patterns that are closer to the RTN.

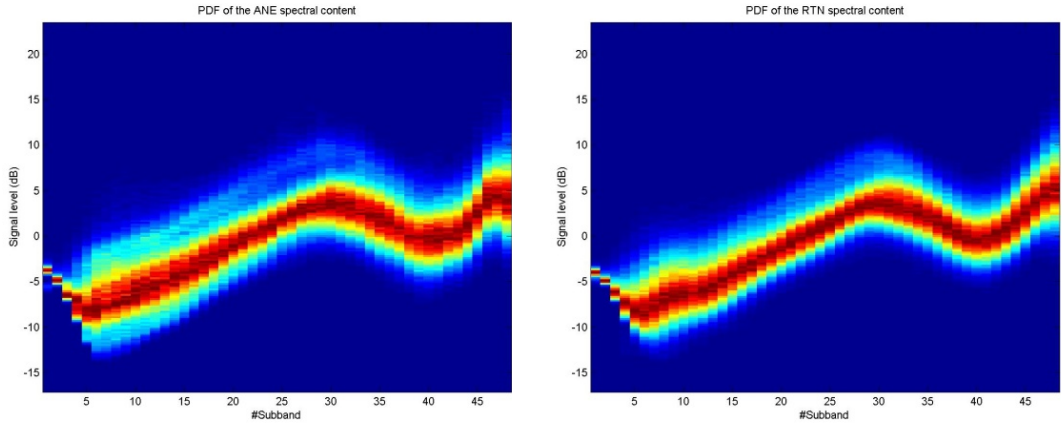


Figure 56 – 2D-PDF of the ANE (left) and the RTN (right) of the Rome database using case I.

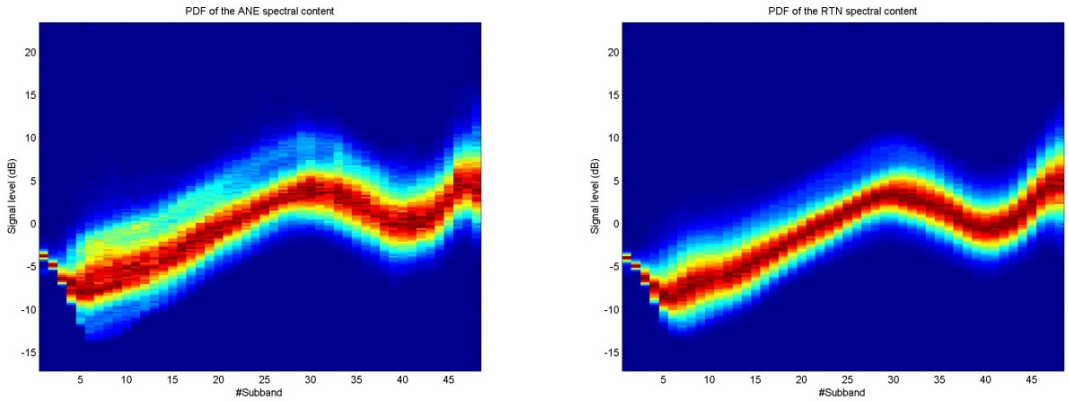


Figure 57 – 2D-PDF of the ANE (left) and the RTN (right) of the Rome database using case II.

In Figure 58 and Figure 59 the 2D-PDF matrices corresponding to Milan recordings are depicted, for Case I and Case II respectively, following the same configuration of labels and colors than the two previous figures. In this scenario, clear differences between ANE and RTN are found in the 2D-PDF functions. Moreover, evidences of two clear spectral patterns or clusters can be observed in the ANE 2D-PDF: one that is located at high signal levels, and another that draws envelopes at lower signal levels.

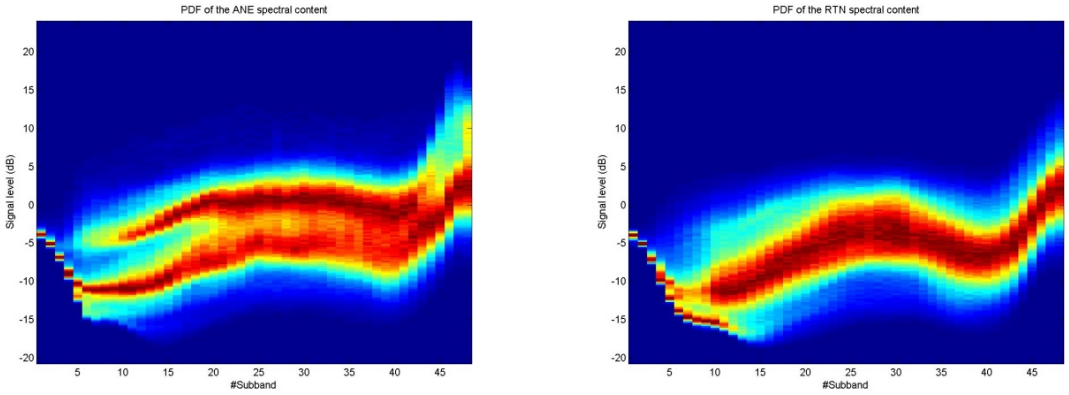


Figure 58 – 2D-PDF of the ANE (left) and the RTN (right) of the Milan database using case I.

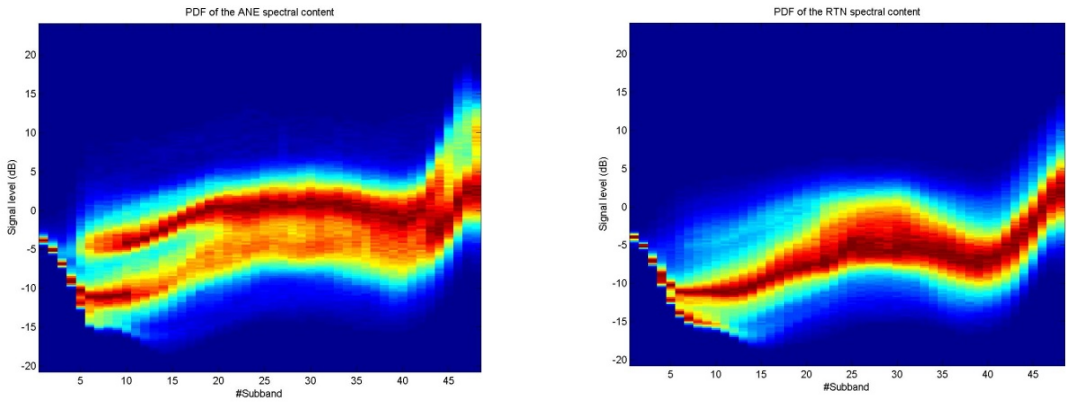


Figure 59 – 2D-PDF of the ANE (left) and the RTN (right) of the Milan database using case II.

In addition, differences between ANE and RTN are more evident in the lower frequency band (e.g. GTCC bands between 5 and 25). These differences seem to be more accentuated in Case II than in Case I, where only anomalous events with SNR greater or equal to 0 dB are still labelled as such.

5.3.3. Subbands optimization

In this section, we describe the results of the process followed for optimizing the selection of the most appropriate signal subbands (following the subband decomposition of the Gammatone Cepstral Coefficients described in Section 3.6) for each of the studied scenarios (Rome and Milan). After the 2D-PDF computation for each case of study and scenario (see Section 5.3.2), a threshold optimization is performed for each of the 48 frequency subbands following the process explained in Section 5.3.1. The obtained results will be relevant for selecting the most appropriate subband with the aim of providing a more reliable signal level measure than only considering the full band audio signal for the A-weighted equivalent noise level computation.

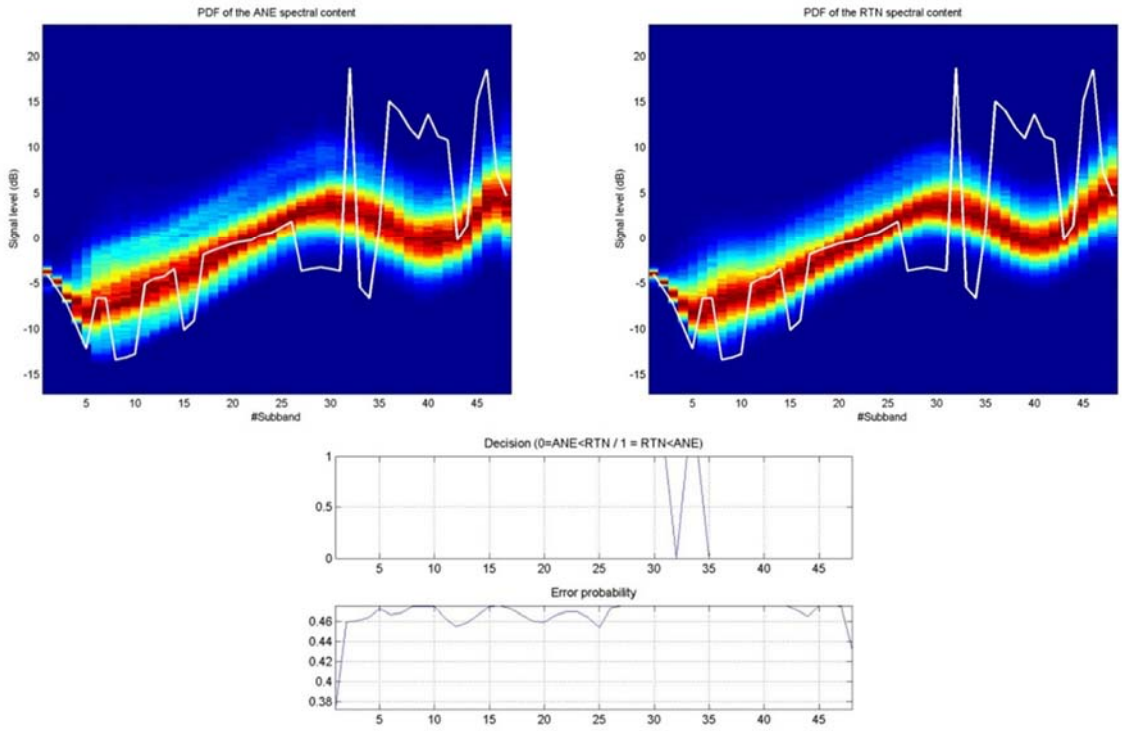


Figure 60 – Subbands optimization results with Rome recordings and case I.

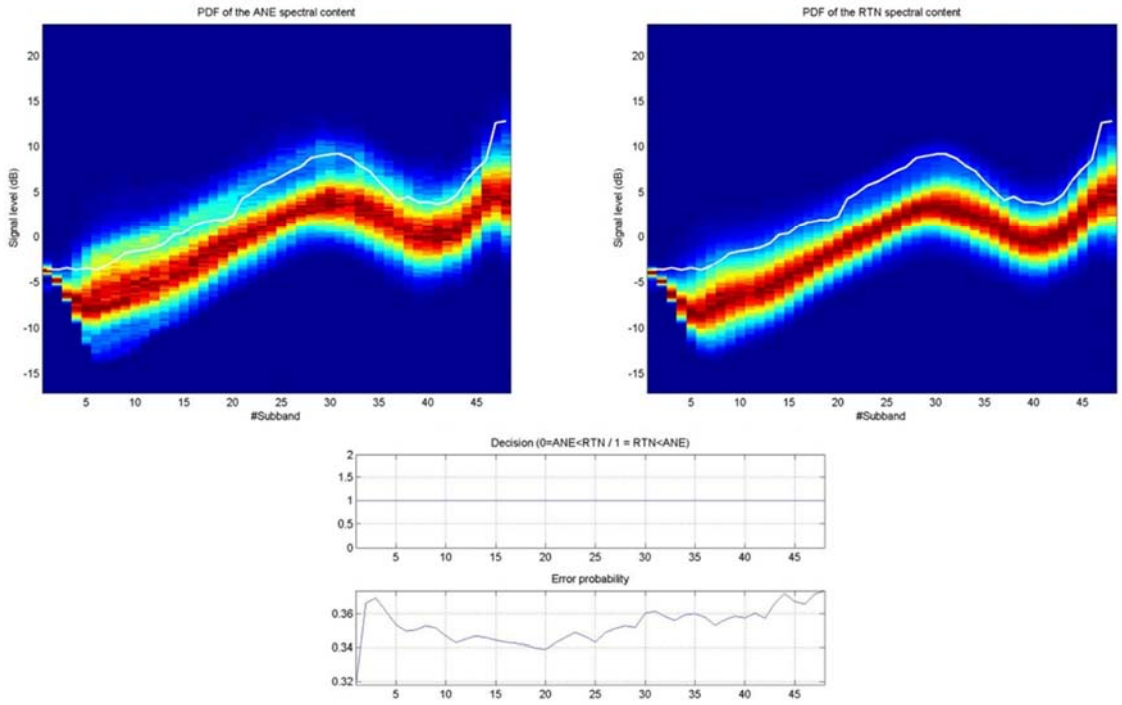


Figure 61 – Subbands optimization results with Rome recordings and case II.

In Figure 60 and Figure 61, we can see the threshold optimization results for the Rome audio database, being the first for the Case I and the second for the Case II of study. At the top of the figure, we can see the 2D-PDF matrices for the ANE (left) and the RTN (right) classes where the optimized threshold is superimposed as a white line. In the bottom of the figure the decision

criteria and the final error probability for each frequency subband are depicted. As it can be seen, the high overlap between the two PDF functions makes difficult to obtain a stable threshold function (i.e., showing a smooth behavior across the 48 frequency subbands), especially in Case I. Moreover, from the results it can be concluded that ANE class only exhibits higher signal levels than RTN only in frequency subbands between 1 and 31 for Case I, while being the other way around for bands from 35 to 48. However, in Case I the probability of error is quite high and stable and the lower value ($\min\left(p_e^{ANE>RTN}(x)\right) = p_e^{ANE>RTN}(1) = 0.3724$) is obtained at band 1 of GTCC.

However, Case II for the Rome database obtains better performance than Case I (see Figure 60). In this case the obtained threshold function shows a smoother behavior than in Case I, and the minimum error probability is also attained at frequency band 1 ($\min\left(p_e^{ANE>RTN}(x)\right) = p_e^{ANE>RTN}(1) = 0.3179$). The error probability exhibits also a frequency region with second minima around subband 20, obtaining lower error probabilities than Case I results. As a general conclusion, Case II obtains the best results for configuring the STD-ANED version for the Rome recordings. In the next section, these results are used to set different configurations of this ANED version, based on summing the signal level at certain frequency bands of the subband decomposition performed in the GTCC computation.

Figure 62 and Figure 63 show the results of the subbands optimization process for the Milan audio database, again considering the Case I and Case II variants following the same presentation scheme as for the Rome database. As it can be seen from the figures, Case II obtains better performance than Case I in Milan, which draws a different picture compared the results of the ANED version for the high capacity sensors (see Section 4.2). The minimum error probability in Case I is attained at the subband 4, and its value is $p_e^{ANE>RTN}(4) = 0.3374$, while in Case II subbands from 35 to 42 obtain a lower probability error, presenting its minimum value (0.3191) at the 41^{th} subband.

From these results, Case II is chosen case for further studies of the STD-ANED for the Milan recordings.

In the next section, these results will be used to set different configurations of this ANED version, based on summing the signal level at certain frequency bands of the subband decomposition performed in the GTCC computation.

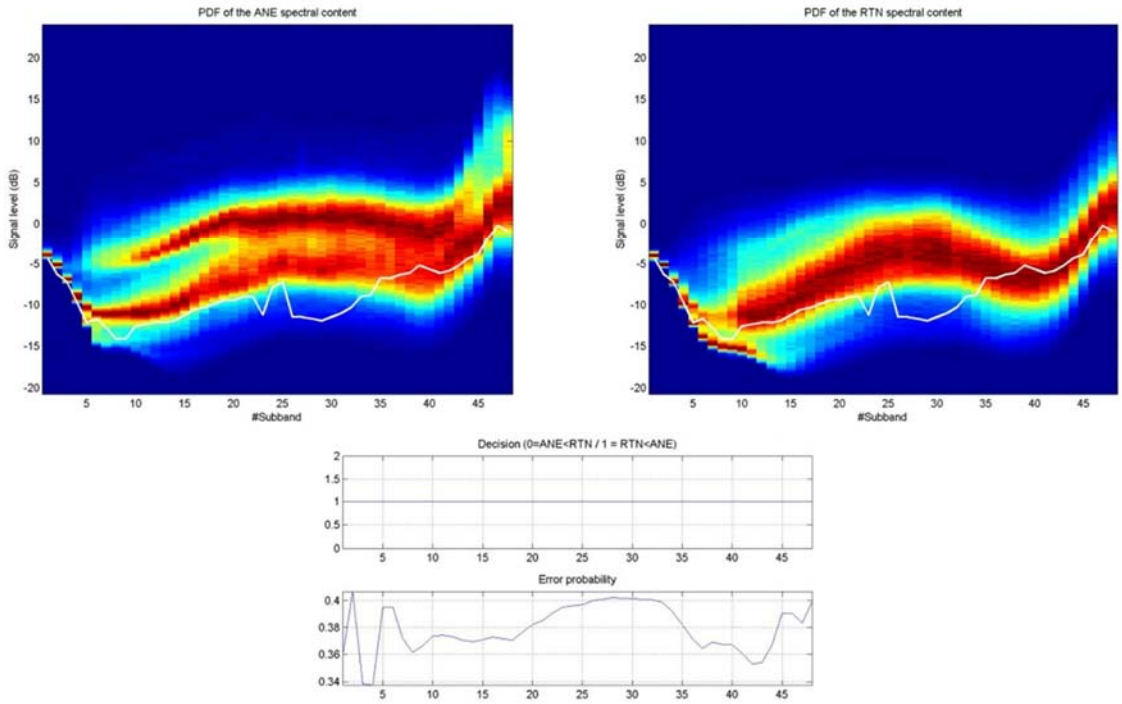


Figure 62 – Subbands optimization results with Milan recordings and case I.

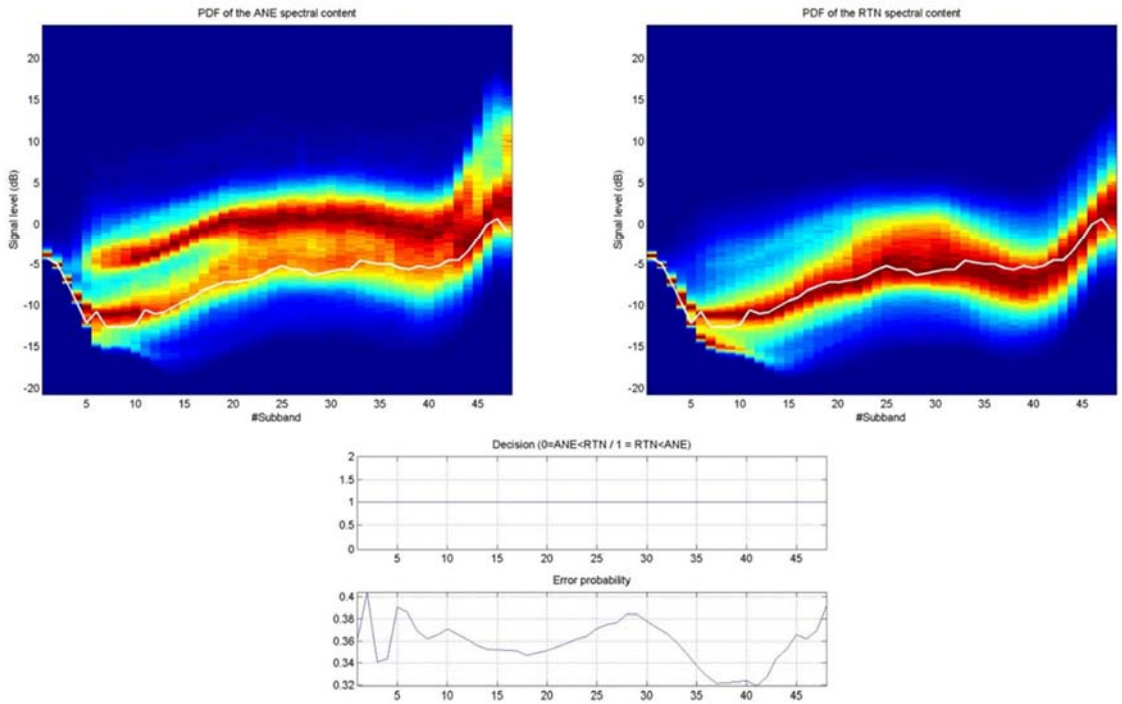


Figure 63 – Subbands optimization results with Milan recordings and case II.

5.4. Design, training and validation of the ANED algorithm for low capacity sensors with real data.

From the results detailed in the two previous sections, we can conclude that an ANED version for low capacity sensors can obtain better results if the algorithm consider those noise levels limited to specific frequency bands where RTN and ANE exhibit larger differences. For this reason, a new study has been conducted to test the reliability of the classification based on integrating the output levels from the subbands parameterized with GTTCs. In this integration (based on sum of the different selected subbands outputs in the linear domain, and then computing the $10\log_{10}(x)$ function of the result), only the outputs from a subset of specific frequency bands are considered (i.e., only the most discriminative ones). The results obtained through these studies can be used for designing an ANED version that tries to obtain reasonable accuracies but demanding for lower computational resources than the ANED version developed for the high capacity sensors accuracies (e.g., designing specific low complexity Infinite-Impulse Response – IIR – subband filters that match the selected frequency subbands).

Given the selected frequency subbands from the studies undertaken in Section 5.3.3, in this section the process of threshold computation has been repeated but following a 4-fold cross validation scheme using the GTCC parameterized databases, following the same approach undertaken in section 4.2. In this case, the threshold optimization has been performed only using the training data in each of the cross validation partition, and the system performance has been assessed using the validation data and the previous computed threshold (as for the learning of classifiers in section 4.2). This way, the obtained results exhibit a larger reliability than the ones reported until now in this chapter, and the derived conclusions will be also more relevant.

In regard the audio databases used in the experiments of this section, the reduced datasets of RTN class (together with BCK in Milan) with optimal balance have been used: a reduction of 3% of RTN and a reduction of 10% for Milan (see section 4.2.2).This option has been chosen in order to compare the computational complexity of the designed low computational capacity sensors ANED configurations with a selected subset of the high computational capacity sensors ANED versions.

Table 12 shows the set of subbands considered for the audio signal level computation before the thresholding process for the Rome database, defining a configuration number for each case of study. These configurations have been defined taking into account the studies performed in the preceding section, specifically those related to Figure 61. Apart from the first configuration (number 1), in which the signal level is computed considering all the available subbands from the GTCC analysis (1 to 48), the rest are devoted to find the best combination of subbands that attain the lowest possible probability error. In Figure 64 the assessment results of the STD-ANED with frequency subband selection for the Rome scenario (Case II) are shown. Compared with the results obtained with the equivalent noise-based threshold-detector (see Figure 53 in Section 5.2.2), we can see that all configurations of the frequency selective ANED version outperforms the previous results. Specifically, configuration 1 obtains a Macro-averaged F1 value of 58.94%, which is above the 57.52% value obtained in the previous study. In fact, configuration 1 is not frequency selective as it includes the sum of all the GTCC subbands in the signal level computation, but it is evaluated in order to assess the improvements obtained when less frequency subbands are considered. Moreover, we can see how all the other tested configurations obtain a higher F1 value than the non-selective configuration (number 1), which informs that the system is approaching a better performance. As we can see, the 9th configuration, which includes only the 1st and the 20th subbands, obtain the maximum Macro-averaged F1 value of 60.60%, which in turn is quite similar to the worst reliability obtained

with the Discriminant Analysis classifier in Section 4.2.4 (see Figure 42), a value that is 20% lower than the best ANED version for the high computation capacity sensors (SVM with GTCC features). However, the obtained results with subband selection prior to signal level computation are in the mid pathway between the ANED version based on simple full-band equivalent noise level threshold-detection and the ANED version for the high computation capacity sensors.

Configuration number	Considered Subbands
1	[1-48]
2	[1] +[11]+[17-20]
3	[10-23]+[25]
4	[10-23]
5	[1]+[10-23]
6	[5-29]
7	[1]+[5-29]
8	[1]+[17-20]
9	[1]+[20]

Table 12 – Configuration details of the ANED frequency subband analysis for the Rome city recordings with case II.

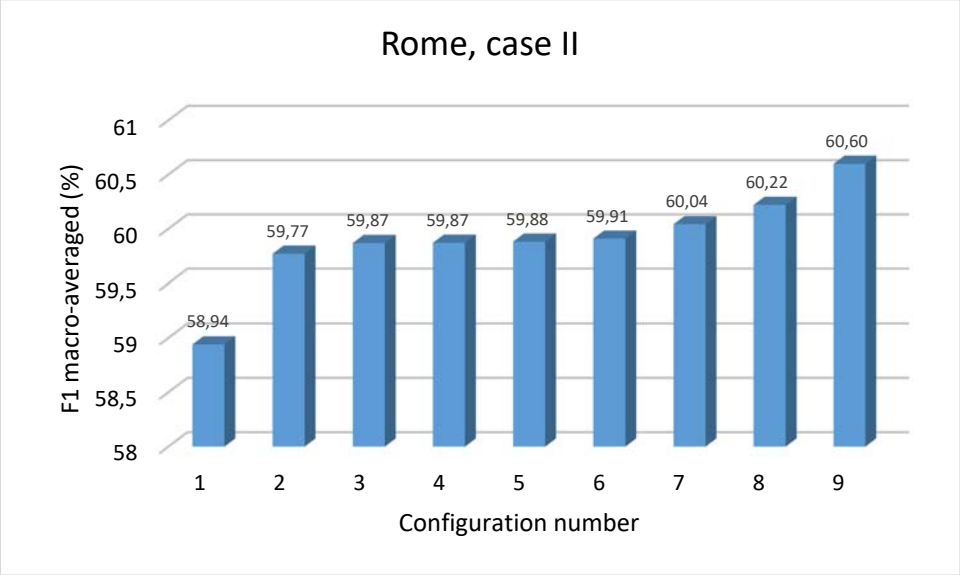


Figure 64 – Assessment results of the low capacity sensors version of the ANED with frequency subband selection for the Rome scenario (case II).

Concerning to the studies performed on the Milan database, Table 13 shows the explored configurations regarding the selected subbands within the 48 of the GTCC parameterization. The selection is based on the studies performed in Section 5.3.3, and specifically on the results shown in Figure 63. The first configuration does not select any specific subband. Instead, all subbands are used for the signal level computation that will be compared to the optimized threshold of the classifier. However, the rest of the six explored configurations reduce this set of selected subbands based on the probability error obtained in Figure 63 (see the bottom plot).

Different approaches are based on selecting subbands located around indexes 3 and 4, but also those located in the region around 37 to 43 subbands, which exhibit lower probability errors.

Figure 65 shows the performances of the STD-ANED that incorporates frequency subband selection for the Milan scenario (Case II). The obtained Macro-averaged F1values range from 63.64% (without specific subband selection) to 68.25% (selecting only subbands from 37 to 41). Compared to the results of the first threshold-detector version of the ANED (see Figure 53) the best configuration (number 7) obtains an improvement of almost 29% in the Milan scenario, although its performance is 17.8% lower than the best performance obtained with the best configuration of the ANED in Milan (see GTCC-SVM performance in Figure 43). Moreover, the non-selective configuration (number 1) obtains also better performance (10.6% greater) than the best case for Rome in the previous low cost sensors ANED proposal (see case II bar of Figure 53). This difference can be explained by the fact that in the previous version the noise level included A-weighting process, while in this version it is not included.

Configuration number	Considered Subbands
1	[1-48]
2	[3-4]+[35-43]
3	[41]
4	[35-43]
5	[36-42]
6	[37-38]+[41]
7	[37-41]

Table 13 – Configuration details of the ANED frequency subband analysis for the Milan city recordings with case II.

Finally, it can be noted that the best performance of the ANED versions designed for the low-capacity sensors is the one that incorporates frequency subband selection for the Milan scenario. It obtains a performance that is 12.6% higher than the best configuration obtained in Rome, which is near to the improvement obtained the ANED version designed for the high capacity sensors (around the 9.3%).

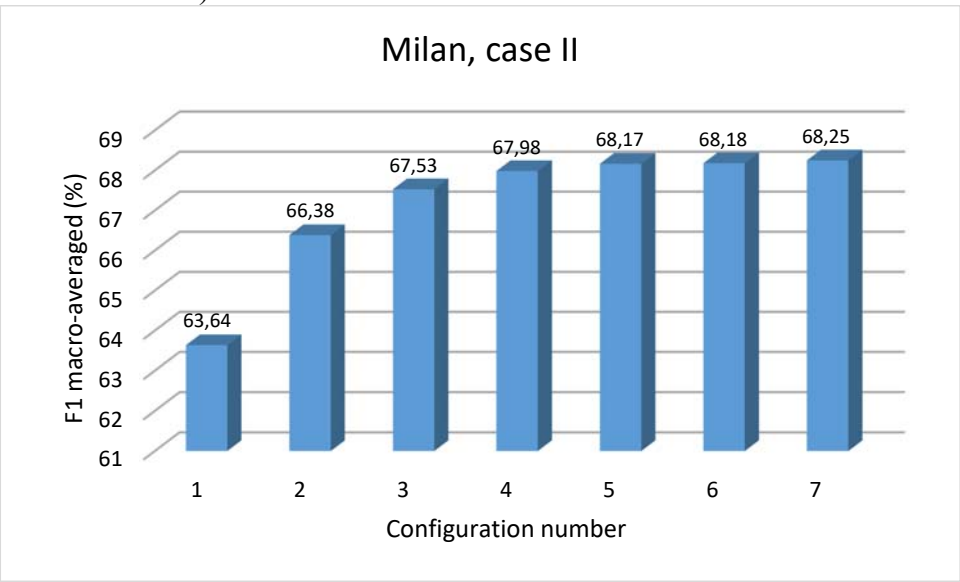


Figure 65 - Assessment results of the low capacity sensors version of the ANED with frequency subband selection for the Milan scenario (case II).

5.5. Computational cost analysis

In this section, the computational performance of the STD-ANED in each scenario is performed and compared to the ANED version developed for the high capacity sensors. The study is based on monitoring the time consumption of the algorithm running on a desktop PC (Intel(R) Core(TM) i7-Q720 CPU @1.60GHz, 4 GB of RAM and O.S. Windows 7). Time consumption of training and testing is assessed, being the second the one more crucial for a comparison of the performance of ANED versions in real-time operation (notice that the training stage is an offline process that is launched once using the available labelled audio databases).

The training and testing times have been computed by averaging the time consumption results after running each ANED version in the two scenarios (Rome and Milan). However, two of the feasible variants of the ANED designed for the high computation capacity sensors adjusted with their corresponding optimal configurations have been compared with the STD-ANED: the Discriminant Analysis (DA-ANED) and the Gaussian Mixture Model (GMM-ANED) classifiers.

As these studies have been performed using software Matlab©, the computational load of the different classification techniques have been evaluated using difference of time stamps using the time monitoring functions *tic* and *toc*. The total time need for learning and for testing in the 4-fold cross-validation scheme for the assessment of each configuration has been retrieved during the simulations, and it has been averaged across scenarios (Rome and Milan databases).

Table 14 shows the results of the computational cost analysis. As it can be observed, the STD-ANED version obtains lower time consumption in the testing than the one obtained by the most computational cost efficient machine learning approach (DA-ANED). However, the computational cost improvement of STD-ANED over DA-ANED is about 32%.

Classifier	Training time (s)	Testing time (s)	Testing time/ ANN testing time
DA-ANED	1.74	0.75	131.6%
GMM-ANED	2436.39	3.36	589.5%
STD-ANED	135.97	0.57	100%

Table 14 – Comparison of computational load of classifiers in terms of time consumption.

5.6. Conclusions

Several approaches to adapt the ANED to the low capacity sensors have been explored and validated using the audio databases built from the recording campaigns in Rome and Milan (see Section 3). Two main approaches have been investigated. The first approach is based on a simple comparison of the A-weighted equivalent noise level computed directly from the raw audio recordings (DTD-ANED), while in the second proposal a previous filtering process is applied before computing the signal level and the subsequent thresholding process (STD-ANED). The two versions have been properly assessed comparing their performance in terms of reliability (F1 measure averaged across the two main audio categories, RTN and ANE). A study with the most reliable obtained configuration of the STD-ANED has been finally

performed in order to compare the computational cost compared with two selected high computational capacity sensors versions of the ANED.

From the obtained results, it can be concluded that there is a clear advantage in the STD-ANED version comparing to the simpler DTD-ANED approach. While the DTD-ANED obtains very low performances (the best obtained Macro-averaged F1 values for Rome and Milan scenarios are 57.52% and 52.95%, respectively – see Figure 53), the STD-ANED increases these performances up to 60.60% and 68.25%, respectively. Curiously, the Milan scenario attains a lower performance than Rome scenario with DTD-ANED, which is just the opposite observed for the STD-ANED version. This fact could be explained because the STD-ANED exploits specific frequency regions, which are the ones where more significant differences have been found between RTN and ANE classes (see Section 5.3). Hence, differences in spectral content help to discriminate between road traffic noise and anomalous events observed during the recording campaign. However, the obtained performances with STD-ANED are still significantly lower than the ones obtained with the ANED designed for high computation capacity sensors (see Section 4), specifically, they are 20% and 17.8% lower than the performance obtained with the best configuration of the ANED in Rome and Milan, respectively (both using the SVM classifier).

As a conclusion, the ANED algorithm conceived for low computation capacity sensors can be seen as a solution that is able to work with lower computational resources, but obtaining a significantly lower degree of confidence in the task of discriminating between road traffic noise and anomalous noise events. In fact, when comparing the STD-ANED with DA-ANED in terms of complexity, we can see that the later need 32% of more computational resources than the former, but instead its reliability is about 8.4% greater (the Macro-averaged F1 value averaged for the two scenarios is 69.82% for the DA-ANED and 64.42% for the STD-ANED).

However, it is worth to mention that at the time of writing this report the requirements of the low computation capacity sensors were not defined, and then, no conclusions about the feasibility of including these low complexity solutions in a the DYNAMAP pilots were obtained.

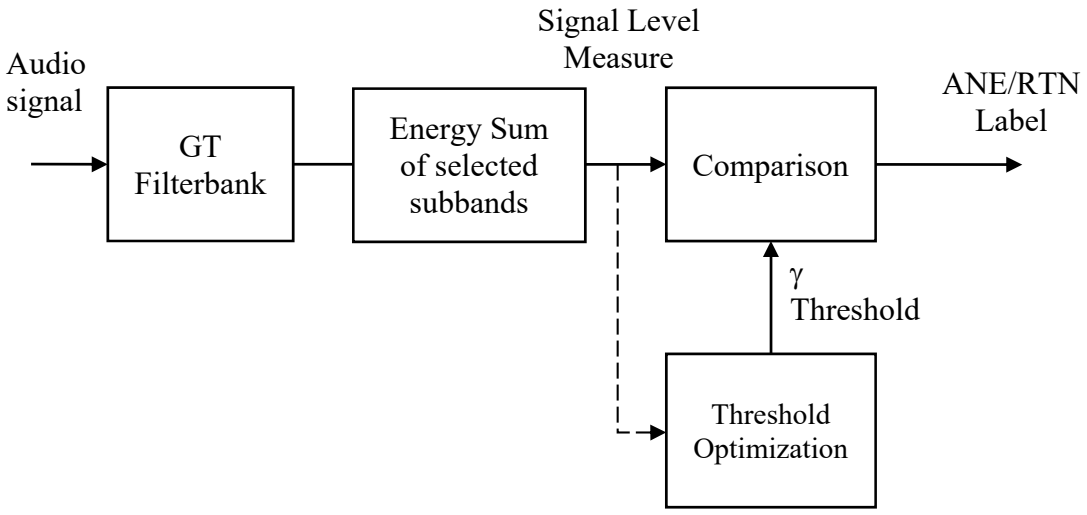


Figure 66 – Block diagram of the STD-ANED solution for the for low computation capacity sensors ANED version.

In Figure 66 a block diagram of the final proposed solution (STD-ANED) is depicted, which is based on obtaining a signal level measure based on the sum of a predefined set of subbands of

the Gammatone Filterbank used in the GTCC parameterization. However, this solution could be properly optimized in terms of computational cost by designing a low complexity filter in the front-end, substituting the GT filterbank analysis stage.

6. CONCLUSIONS

In this document, we report the main work and results derived from the development the anomalous noise event detection algorithm (ANED) that will run on the sensor network of the DYNAMAP pilots. An ANED version for high computational capacity sensors has been designed, validated and implemented to work within the hardware sensor platform, while an ANED version for low computational capacity sensors has been designed and evaluated; it works with lower computational requirements at the price of reducing its classification reliability. The final version of ANED low capacity is not implemented by today, since in the framework of the project the hardware platform that will respond to these restrictions is not yet chosen. Therefore, the real computing capacity of the low computational sensor node is not available to adjust the algorithm.

The first ANED approach was tested using synthetic mixtures of real RTN and ANE gathered from online repositories before conducting the recording campaign (Socoró, et al., 2015). Two machine learning strategies (supervised and semi-supervised), two classification algorithms (FLD and KNN) using two cepstral parameterization schemes (MFCC and GTCC) were evaluated. The obtained results showed promising results for salient ANEs (with ANE-to-RTN SNR of 6 and 12 dB) on a balanced database for the semi-supervised machine learning approach.

Next, in order to validate those preliminary results in real life scenarios, a recording campaign was conducted on May 2015 in the two selected pilot areas of the DYNAMAP project (Rome highway and district 9 of Milan) (Alías, et al., 2015). This way, we collected more than 10 hours of audio data to validate the initial ANED approach by considering databases with real ANEs (e.g., distribution, saliency, duration, etc.). An exhaustive labelling and subsequent analysis of the recordings was performed in order to generate the proper audio datasets to train and validate the ANED proposals, taking special attention to the segmentation and SNR computation of the ANEs.

Once tested the initial ANED optimal configuration with the new data, we realized it was necessary to conduct further research to obtain reliable designs with real-life audio data since the obtained results contradicted our previous conclusions and showed performances 25% lower in average. After detailed analyses, we concluded that the main reasons for this change of behavior were the following:

i. the two pilot areas represent significantly different classifications, e.g., being the ANEs collected from Rome and Milan, thus, being the synthetic database derived results only somehow generalizable to the Rome scenario,
ii. the dramatically unbalanced nature of the two-class classification problem at hand (e.g. in Rome RTN represents the 97.5% of the total number of samples), and,
iii. the greater diversity of the ANEs, showing large variability in terms of duration and saliency (e.g. with SNR ranging from < 0 dB to >20 dBs values, and durations from several milliseconds to dozens of seconds).

To address this new research framework, and taking the most of its thorough analysis, we considered the following strategies in a change of paradigm of the design:

i. moving to the supervised from the semi-supervised machine learning approach,

ii. reducing the RTN class versions instead of the complete original datasets to work with balanced databases eliminating redundant data, and,
iii. considering two cases of study:
    i. Case I - all ANEs are considered regardless their SNR, and
    ii. Case II (ANEs with SNR below 0 dBs are considered to be RTN) as an optimization parameter to configure the datasets used for training and testing the algorithm performance.

Furthermore, the initial set of classifiers (FLD and KNN) is extended with supervised approaches (DA, GMM, ANN and SVM) and a complete study regarding the optimal configuration of their tuning parameters is performed including the two audio feature parameterizations (MFCC and GTCC). The reliability of the optimal configurations of each classifier were compared, obtaining different solutions for each of the two analyzed scenarios, concluding the highest performances when using SVM and GTCC, with F1 macro-averaged values of around 76% for Rome (Socoró, et al., 2016) and 83% for Milan databases. The averaged 7% of global F1 macro-averaged difference confirms the specific particularities of the two acoustic environments evidenced within the database analyses. To conclude the review of the accuracy of the ANE detectors, it should be noted that the change of paradigm as an answer for the decrease of 25% of accuracy in the first's real-data tests has been a successful approach, reaching nearly similar performance values as we got in the synthetic mixtures.

From the time consumption analyses, we conclude that GMM is the classifier that attains a good tradeoff between accuracy (around 72% in Rome and 80% in Milan) and computational cost, while DA is the most competent in terms of computational cost in the testing stage. From these results, a solution that incorporates those classifiers with low computational load and performance tradeoff (KNN, DA and GMM) has been implemented for its real-time operation on the hardware platform. The implementation includes the tuning of some important configuration parameters, allowing for the adaptation of the ANED algorithm to each specific pilot scenario (Rome and Milan). However, the real-time operational tests with the final low capacity sensors hardware have not been included because they were not available at the time of writing this report.

In this sense, also the voting consensus presents a preliminary approximation at the time of closing action B3. A first approach has been implemented in order to label every 1 sec. time frame of audio signal, as the project demands and the hardware high capacity platform will be able to compute (for more details, see Section 4). In the development of actions B5 and B6 we will address this problem with more precision, thanks to the real data audio measurements and the accuracy of the ANED algorithm to detect the anomalous events that impact most on the calculation of the noise level. In that point, a more precise voting consensus will be proposed if needed, taking into account possible differences of performance between Rome and Milan locations.

The design of the ANED version for the low computational capacity sensors has been addressed without having any definite hardware specification. However, we have evaluated the viability of this version in terms of performance and computational complexity, considering two main approaches. The first one is based on a simple equivalent noise level comparison with a predefined threshold locally optimized to minimize the error classifications between RTN and ANE. This extremely simplified version of the ANED yield very low classification accuracies. Since they were barely higher than the chance probability in a binary classification, this approach is discarded in any further implementation. The second approach is based on applying

a signal filtering before the noise level computation (named STD-ANED). This way, the spectral differences between the two main audio categories (RTN and ANE) can be exploited. After studying which frequency subbands (based on the GTCC audio features computation) provide lower error classification probabilities considering the audio datasets obtained from the onsite recordings, an assessment of different configurations of the STD-ANED was performed to obtain the highest accuracies for both pilot areas. The results show a general computational cost reduction of around 30% regarding the most efficient machine learning approach (DA-ANED) at the price of around 20% of reliability reduction (measured using the Macro-averaged F1 measure). Nevertheless, the viability of this reasonable tradeoff should be validated on the low capacity computation sensor platform when available.

In future actions B5 and B6 any inaccuracy in the performing of the algorithms in Milan and Rome, respectively, will be corrected using the feedback of the operation of the algorithm itself working with real-world signals and evaluating the results of the classification in real time.

7. ACKNOWLEDGMENTS

The authors would like thank our colleagues in ANAS S.p.A. and Università de Milano-Biccoca for their support in the recording campaign in Rome, and Milan, respectively (Task B3.1 – Pilot areas onsite inspections and environmental noise recording). The authors would also like to thank our colleagues in BlueWave for their support in the ANED implementation for the low cost sensors hardware, concerning the task Task B3.2 – Development of the ANED algorithm for high computation capacity sensors.

This research has been partially funded by Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement (Generalitat de Catalunya) under grant refs. 2014-SGR-0590 and 2015-URL-Proj-046.

8. BIBLIOGRAPHY

Alías F. [et al.] Training an Anomalous Noise Event Detection Algorithm for Dynamic Road Traffic Noise Mapping: Environmental Noise Recording Campaign [Conference] // Proc. TecniAcústica. - Valencia (Spain) : [s.n.], 2015. - pp. pp. 345-352, .

Alías F., Socoró J.C. and Sevillano X A Review of Physical and Perceptual Feature Extraction Techniques for Speech, Music and Environmental Sounds [Journal]. - Basel (Switzerland) : MDPI, 2016. - 143 : Vol. 6.

Babisch W. Transportation noise and cardiovascular risk: Updated review and synthesis of epidemiological studies [Journal] // Noise&Health. - 2006. - 30 : Vol. 8. - pp. 1-29.

Cook R. P. and Scavone G. P. The Synthesis TooKit in C++ (STK) [Book]. - Standford, California. : [s.n.], 1995.

Davies D.L. and Bouldin D. W. A Cluster Separation Measure [Journal] // IEEE Transactions on Pattern Analysis and Machine Intelligence. - 1999 : [s.n.]. - 2 : Vol. 1. - pp. 224-227.

Directive EU Directive 2002/49/EC of the European parliament and the Council of 25 June 2002 relating to the assessment and management of environmental noise, Official Journal of the European Communities [Report]. - [s.l.] : Official Journal of the European Communities, L 189/12, 2002.

Frigo M. and Johnson S. G. FFTW [Book]. - Massachusetts Intitut of Technology (MIT), Massachusetts : [s.n.], 2005.

Furui S. Cepstral analysis technique for automatic speaker verification [Journal] // IEEE Transactions on Acoustics, Speech and Signal Processing. - 1981. - 2 : Vol. 29. - pp. 254-272.

Jain A.K., Murty M.N. and Flynn P.J. Data clustering: a review [Journal] // ACM Computing Surveys. - 1999. - 3 : Vol. 31. - pp. 264-323.

Nencini L. DYNAMAP monitoring network hardware development [Conference] // Proc. 22nd International Congress on Sound and Vibration. - 2015.

Radaelli S. [et al.] The LIFE DYNAMAP project: automating the process for pilot areas location [Conference]. - 2015.

Socoró J.C. [et al.] Analysis and automatic detection of anomalous noise events in real recordings of road traffic noise for the LIFE DYNAMAP project [Conference] // INTERNOISE 2016. - Hamburg (Germany) : [s.n.], 2016.

Socoró J.C. [et al.] Development of an anomalous noise event detection algorithm for dynamic road traffic noise mapping [Conference] // Proc. 22nd International Congress on Sound and Vibration. - 2015.

Valero X., Alías, F. Gammatone Cepstral Coefficients: Biologically-Inspired Features for Non-Speech Audio Classification [Journal] // IEEE Transactions on Multimedia, vol. 14, no. 6.. - December 2012. - pp. 1684 - 1689.

Xu R. and Wunsch D. Survey of clustering algorithms [Journal] // IEEE Transactions on Neural Networks. - 2005. - 2 : Vol. 16. - pp. 645-678.

Zambon G., Benocci R. and Bisceglie A. Development of optimized algorithms for the classification of networks of road stretches into homogeneous clusters in urban areas [Conference] // Proc. 22nd International Congress on Sound and Vibration. - 2015.

Zechman E. L. Matlab Continuous Sound and Vibration Analysis toolbox [Report].