Environmental noise monitoring using source classification in sensors

Citation

Year
2018

Version
Publisher's PDF (version of record)

Link to publication
TUTCRIS Portal (http://www.tut.fi/tutcris)

Published in
Applied Acoustics

DOI
10.1016/j.apacoust.2017.08.006

License
CC BY

Take down policy
If you believe that this document breaches copyright, please contact cris.tau@tuni.fi, and we will remove access to the work immediately and investigate your claim.
Environmental noise monitoring using source classification in sensors

Panu Maijala a,*, Zhao Shuyang b, Toni Heittola b, Tuomas Virtanen b

a VTT Technical Research Centre of Finland, P.O. Box 1000, FI-02044 VTT, Finland
b Tampere University of Technology, PO Box 527, FI-33101 Tampere, Finland

1. Introduction

Environmental noise, defined as unwanted or harmful outdoor sound created by human activities [1, Art. 3], can be generated by traffic, industry, construction, and recreation activities [2, p. 12]. Airports, wind power plants, rock-crushing, shooting ranges, and motorsport tracks are examples of noise sources for which sound propagation over several kilometers is relevant.

One challenge in environmental noise monitoring is how to make sufficiently comprehensive measurements both in time domain and spatially. The changes in weather conditions have a significant effect on monitored noise levels [3] and in order to obtain most of the variations the noise has to be monitored for extended periods of time [4–6]. Also, a single point noise measurement is rarely representative for a whole neighbourhood and several sensor locations are needed. Because of high costs of the equipment and the amount of human resources needed, the reliability, validity, and representativeness of environmental data is usually unsatisfactory. Only a few reported scientific experiments with uninterrupted noise data captured from each relevant location over long periods of time exist [7–10].

The typical need for measurements is to monitor the noise caused by a noise source (e.g. an airport or an industrial plant) in a residential area. However, also other noise sources exist and the captured noise level is usually a result of a combination of the target and interfering sound sources: wind-generated, cars, and birds being examples. Sound level meters used for noise monitoring either capture sound levels or time domain noise data and store the data locally – or nowadays more often – on a remote server [11]. The most common method to ensure the noise was caused by the original source is listening through all the samples afterwards. This requires a huge amount of resources because of a large amount of data due to often necessary long-term measurements. Also, if only noise levels are recorded, validation by listening is not possible.

A considerable amount of manual work can be saved by automatically validating sound sources. Furthermore, privacy issues can be avoided and required network load can be largely reduced, if the automatic validation algorithm is performed on the sensor and only the measurement result is transferred. Previous validation algorithms on sensors have been limited to hand-crafted rule-based systems [12]. However, a simple hand-crafted classifi-
A pattern recognition rule can hardly provide good accuracy in a complex environment, e.g., monitored target producing several types of sounds. As another drawback, the design of a hand-crafted classifier requires an expert for every noise monitoring scenario. The increased computational capacity has made a sensor possible to classify noise sources using a pattern classification algorithm, which is capable of learning a sophisticated noise source classifier for an arbitrary scenario, simply using relevant annotated recordings as training material.

An pattern classification algorithm typically consists of a feature extractor and a classifier. Mel-frequency cepstral coefficients (MFCCs) [13] are used as common features for a wide range of acoustic pattern classification such as speech recognition [14] and music information retrieval [15]. Gaussian mixture model (GMM) [16] has been traditionally cooperated with MFCCs to model different types of sounds. Specifically, the combination of MFCCs and GMM has been used for various noise monitoring scenarios [17,18]. The use of artificial neural network (ANN) for acoustic pattern classification has been increasing with the development of computing power and new training algorithms that allow utilizing large amounts of training data. Some recent studies have shown that ANN outperforms traditional GMM in sound event detection [19–21].

Together with the smaller and cheaper computing capacity, the breakthrough of wireless technology in the very beginning of 2000s have made possible to translate the physical world into information [22] and given reason to define concepts like Internet of Things and ubiquitous sensing [23]. The word “smart” was first used as an attribute to a sensor with an Internet access. Today, it is more closely related to a sensor with own intelligence, some computational capacity for data analysis and decision making [24].

The main objective of this study was to show if it would be possible to automatically capture only the noise from the original source, by adding intelligence and human hearing-like decision algorithms to the sensor. This would free the huge amount of human resources needed to validate the noise data and improve and representativeness of the results in environmental noise measurements. An implementation of a noise classification algorithms in a sensor will be introduced. The general concept of the noise monitoring system is explained in Section 2 and the pattern classification algorithms are given in Section 3. Additionally, an evaluation of the performance of the algorithms in a case study is shown (Section 4) and some discussion the requirements and the future work in Section 5.

2. Noise monitoring

The proposed noise monitoring system comprises of smart sensors which are connected through wireless uplink to the cloud service. The overview of the system is illustrated in Fig. 1. The smart sensor consist of a measurement microphone and a single-board computer with a wireless transmission unit. To alleviate the privacy issues concerning the continuous audio capturing and storage, the most of the analysis and processing is done already in the sensor and only analysed data is transferred and stored in the default setting. This approach will also lower the amount of transferred data from a sensor to the cloud service, and enables placing sensors to areas with lower quality wireless uplinks. In the sensor, A-weighted 10-min equivalent sound pressure level \( L_{A,600s} \) values are calculated continuously, and predominant noise sources are detected within the measurement time segment. This information is used to decide whether the actual acoustic signal is needed for further inspection in the cloud service. For example, segments exceeding the legal maximum allowed sound level can be saved for manual inspection. All the extracted measurements are transmitted from the smart sensor to the cloud service for further analysis. The cloud service stores the data in the measurement database, and audio segments marked for later inspection are stored in the disk server. End-users access the measurement data and analysis of the measurements through a web-based portal.

2.1. Smart sensor

For the prototype, the credit-card-sized RPi (Raspberry Pi) developed by the Raspberry Pi Foundation was selected mainly due to its excellent support network and general usability. RPi1, the first generation model was used in the prototype because it was the only available model in 2012 when the implementation was made. Additional functionality was added by an audio codec (a 24-bit multi-bit sigma delta AD converter), a smart power

![Fig. 1. Block diagram of the noise monitoring system.](image)
management board with an uninterruptible power supply feature, and mobile connectivity. The selection of the microphones ended up with two models: one covering the audible range dynamics from 14 dB to 119 dB, and another from 20 dB to 140 dB (A-weighted).

Based on preliminary tests, solar power was selected to allow totally wireless sensors. The electronics and batteries were built inside a solar panel frame (see Fig. 2). Whenever the 60 W panel gets solar energy, the batteries start charging, system is powered up, a secure cloud service connection is established, and preprocessed real-time noise data flow to the online service is initiated. It is also possible to access the sensor unit remotely through the online service. The batteries, when fully charged, will keep the system running during the dark hours. The total cost of the components is about 150 €, the solar panel being the most expensive component, but the price could be reduced in mass production, or using an external power source.

The sensor continuously monitors the noise by capturing 10-min long non-overlapping analysis segments, and the equivalent sound pressure level \( L_{p,A,600s} \) values are calculated for each segment. The sound source classification is used to find the noise source likelihoods within the analysis segment. The acoustic measurement values, noise source likelihoods and time-stamps are transmitted to the cloud service. Analysis segments having \( L_{p,A,600s} \) value over the set threshold are compressed with a lossy-audio compression (e.g. 32 kbit/s MPEG-1 Audio Layer 3) method and transmitted to the cloud service. These can be later used to verify the noise source more accurately either with automatic methods or by the users.

2.2. Accessing data and visualisation

The measurements are accessible through a web-based user interface, which combines a large amount of measurements in an easily readable format by using data visualisation and data reports. The sound pressure level (SPL) measurements can be filtered based on the sound source classification results to show measurements for assigned to particular sound source. In the service the measurement data is visualised in multiple ways: calendar heat-maps, graphs, and report tables. Example view from the portal is shown in Fig. 3.

The calendar heat-maps are used to visualise the average SPL values over certain time span (one day, one hour) with a colour of a calendar cell, an example of this is shown as measurement calendar in Fig. 3. The heat-map collapses SPL measurements within one hour into one number and decodes it into colours based on location-specific SPL limits. In preliminary studies, three colours were observed to give sufficient visualisation of measured SPL values. For the case study (see Section 4), colours are defined in following manner: green colour denotes SPL values under 45 dB, yellow denotes SPL values between 45 dB and 55 dB, and red denotes SPL value over 55 dB, the national limit for outdoor noise in residential areas. The limits shall be adjusted in accordance with the national law for each target. Only measurements associated to the targeted sound class are presented in the calendar.

The measurement graph is used to visualise the SPL values against the measurement time-stamp, an example of this is shown in the lower panel in Fig. 3. Three type of graphs are used to visualise measurement with differently assigned data: firstly showing all SPL measurements as such, secondly showing SPL measurements and sound source probability at current time interval denoted with colour intensity under the curve, and thirdly showing only SPL measurements assigned for targeted sound source. The noise monitoring location specific SPL limits (same as in calendar heat-map) are shown in the graph with horizontal lines.

In addition to the calendar and graph based visualisations, numerical measurement reports are used to show more exact values and analysis. The reports are used to show daily, weekly, monthly and yearly averages of the SPL measurements. Reports include also noise descriptors such as the day-evening-night level \( L_{DEN} \) introduced in the END [1], to give comprehensive figure of the noise levels over longer time segments. If needed, some higher level noise values like unbiased annoyance (UBA) [25] can be added to be calculated.

The portal provides different level information depending on the user account type. The monitoring site managers (system clients) can grant access for the people living close to the monitoring site (public users), and the services provides them easily approachable noise measurement summaries, and possibility to add feedback or comments on the measurement time-line, providing direct connection to the monitoring site management. The site manager or a community liaison officer can use the feedback from

![Fig. 2. A prototype version back cover opened.](image-url)
the public to react noise levels and types, and reply directly to the comments. If the commented time-stamp has a stored audio associated to it, the site manager can also audition it in this stage. Public access is important to make the noise monitoring transparent, and engage the public by giving them more active role how the monitoring results should be interpreted. This should alleviate many negative attitudes related environmental noise and noise monitoring. Administrative users like governmental authorities are presented accurate measurement reports to help to follow the average noise levels over longer time segments often used in official noise management.

2.3. Validity of the results

Standard IEC 61672-1:2002 [26] specifies three kinds of sound measuring instruments in two performance categories. Most of the commercially available sound level meters conform this standard requirements. There have been attempts to integrate sound measuring capabilities also to other instrumentation or devices, like mobile phones [27–29]. The driving force in these studies has been the need for spatially more representative data and fulfilling the accuracy requirements of the instrumentation for standardised measurements has not been addressed. The presented approach balances between these two extremes: the goal for design is to conform at least class 2 requirements, but still to keep the costs low so that the number of units in any implementation may be several times higher than using the conventional sound level meters. The calibration of the unit is performed using a conventional sound level calibrator equipped with a specially manufactured 1" adapter on the microphone of the unit.

Considering the uncertainty of the measurements, the fact is that the influence of instrumentation can be considered low [30] compared to the effect of environmental conditions [5]. The representativeness of data increases validity of an environmental noise measurement and this is achieved by both the increased spatial coverage and classified noise source data.

3. Automatic detection of target sources

In the proposed automatic target source detection system, noises are defined into two classes. Sounds propagating from the target sources belong to a target class, whereas interfering noises as well as silence belong to a background class. Examples of possible target sounds are plant noise and aircraft noise. Possible background noises may be caused by e.g. traffic, wind, rain, thunder, and birds. The activity of the target sources is detected by analysing continuous audio input and making binary classification between the background and the target. The audio input is the same as the signal used for SPL measurement, but without the A-weighting filter.

The detection system consists of two stages: the training stage and the monitoring stage (see Fig. 4). Acoustic models are learned from training examples, captured audio with manual annotation, in the training stage. The learned acoustic models are used to classify audio captured on a sensor, to detect the activity of target, in the monitoring stage. An example of the system output is given in Fig. 5. The training algorithm needs only annotation of target sounds. Traffic sounds, regarded as background in 5 are annotated to help understand the system output.

3.1. Acoustic features

Feature extraction transforms an audio signal into reduced representation. MFCCs are used as features in the proposed system. Mel-frequency cepstral coefficients (MFCCs) [13] have been originally proposed and widely used in speech recognition [14]. Afterwards, MFCCs have been proved to be effective in a wide range of audio processing applications such as sound event detection.
An audio signal is analysed within short frames (e.g. 100 ms) with 50% overlap. Every frame of signal is windowed with a Hamming window. Discrete Fourier transformation is performed on the windowed frames to obtain spectrum and the spectrum is wrapped into Mel scale. Logarithm of Mel-spectrum is performed with discrete cosine transformation to obtain Mel-cepstrum. Coefficients taken from Mel-cepstrum are called MFCCs. The proposed method uses the same classifier for sensors in different locations. However, the audio amplitude changes with the distance between a source and a microphone, which is reflected in the 0th coefficient. The 0th coefficient is usually excluded [14] to keep the features amplitude invariant.

In order to provide temporal dynamic information across adjacent frames, deltas of MFCCs [34] are used in addition to static MFCCs. The first-order delta (Δ) is called differential of MFCCs and the second-order delta (ΔΔ) is called acceleration.

3.2. Supervised classifiers

Two types of supervised classifiers are investigated: Gaussian mixture model (GMM) as a representative of generative classifiers and artificial neural networks (ANN) as a representative of discriminative classifiers. A GMM represents a class by a distribution of its correspondent feature vectors [16]. The probability density func-

---

**Fig. 4.** Block diagram of the automatic target sound detection system.

**Fig. 5.** Example of the target detection output using the GMM classifier. The top panel shows the spectrogram and the second panel illustrates the corresponding annotation. Traffic is regarded as background sound, whereas crusher and alarm are the target sounds. The third panel illustrates the target likelihood and decision threshold. The bottom panel illustrates the detection result as the system output.
function of a GMM for an observation $x$ is the weighted average of its multi-variate Gaussian distribution components as:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i (2\pi)^{-\frac{d}{2}} |\Sigma_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)},$$

where $M$ is the number of Gaussian components. The parameters of the density model are collectively denoted as $\lambda = [w_i, \mu_i, \Sigma_i; i = 1 \ldots M]$. The weight, mean and covariance matrix of $i$th Gaussian component are denoted as $w_i$, $\mu_i$, $\Sigma_i$, respectively, satisfying $\sum_{i=1}^{M} w_i = 1$. The GMM parameters of a class are iteratively estimated using the training data with the expectation maximisation (EM) algorithm. Classification can be made using GMMs by outputting the class whose GMM gives the highest likelihood on a input vector $x$.

An ANN is used to estimate a function that yields desired outputs with given inputs [35]. The parameters of an ANN are estimated using training examples. A training example consists of an input feature vector $x$ and a target vector $y$. When an ANN is used as a classifier [21], the target output is typically a vector $y$ with the size of $C$, the number of classes. Given the feature vector $x$ from class $i$, the target vector entry $y_i$ is set to 1, whereas other elements in target vector $y$ are set to 0. Thus, the output of an optimised ANN classifier is interpreted as the activity indications of $C$ classes of sound events. The activity indication is later called likelihood, since it is used in the same way as estimated likelihood in the GMM, though the indication is not a probability measurement. In the proposed system the multilayer perceptron (MLP) [36], which is a basic type of ANN, was used.

Let us denote input layer as $h^1 = x$ and the values of $k$th layer as $h^k$. The values of the next layer $h^{k+1}$ is calculated as:

$$g^k = W^k h^k + b^k, \quad 2 \leq k < L$$

$$h^{k+1} = \mathcal{F}(g^k),$$

where $W^k \in \mathbb{R}^{h^k \times h^{k+1}}$ is the weight matrix between layer $k$ and layer $k+1$, $S_k$ being the number of neurons in layer $k$. The bias vector of layer $k$ is denoted as $b^k$, which can be considered as the weights for an additional all one's input vector. An activation function $\mathcal{F}$ is the applied element-wise on the linear transformation output. $L$ is the total number of layers in the ANN. In the developed system, maxout function as activation function for hidden layers and logistic sigmoid function for output layer was used. Maxout is an unbounded function whereas sigmoid function ranges between 0 and 1. It has been shown that using two maxout layers with enough neurons can approximate any continuous functions [37]. In the optimisation, a cost function is a measure of difference between the obtained neural network outputs and target outputs. Kull-Leibler divergence is used as cost function and the parameters, weight matrices ($W^k$) and bias vectors ($b^k$), are optimised using the stochastic gradient descent algorithm.

### 3.3. Training and monitoring

Supervised learning requires a set of training examples, i.e., audio signals with manual annotations, at the training stage. Feature vectors of target class are derived from the time segments annotated as target sounds, whereas all other frames are used to represent background class. The extracted features (MFCCs) are collected for each class according to the annotations. When GMM is used, the features are used to estimate the feature distributions of each class. When ANN is used, the target outputs are $[1, 0]$ and $[0, 1]$ for feature vectors corresponding to the background class and the target class, respectively.

At the monitoring stage, a detection is made in one second non-overlapping segments. For each class, a score is computed as the sum of log-likelihoods (the logarithm of the likelihoods) of each frame in the corresponding second. The target likelihood in Fig. 5 is calculated as the score of target class divided by the sum score from all classes. The target sound source is detected as being active when the target likelihood is over a threshold (default value 0.5), otherwise inactive. The threshold can be tuned in case that precision is more important than recall, or vice versa. The precision and recall are later introduced in Section 4.3. Fig. 3 illustrates the noise portal that represents the estimated target activity in long term (1 h), taking majority vote from the activity outputs of corresponding seconds.

### 4. A case study: rock-crushing plant

A case study was made on the noise measurement of a rock crushing plant – a typical environmental noise assessment with nearby habitation. The feasibility of the proposed concept was evaluated with one sensor node next to the plant. The plant has regular working hours, thus the reliability of the target activity detection could be easily verified.

#### 4.1. Measurement setup

The audio data was captured near a rock crushing plant (Fig. 6). The location of the sensor is indicated by a red triangle. The location of the nearest habitation house is indicated by the blue square. The most prominent sound sources in the plant are two rock-crushers denoted as red circles: a fixed rock-crusher and a mobile rock-crusher. The distance between the sensor and the fixed rock-crusher was about 280 m measured from their GPS coordinates and the distance between the sensor and the mobile rock-crusher was about 500 m. Even though the mobile rock-crusher is able to change its position, it was stationary during the case study. Beside the rock-crushers, another significant type of a target sound was made by lift-trucks, which fed rocks to the crushers and distributed the produced stones. The sensor was located close to a road, near a forest.

#### 4.2. Captured noise data

Three minutes of audio was continuously captured every 10 min, making a total of 432 min for each day. All types of noise generated by the working activity of the plant was collectively defined as the target class, including rock crushing, lifting-truck sounds, and alarm sounds from the machinery. On the contrary, traffic noise coming from the road and the noise generated by the wind and the trees were two significant types of background sound sources. Example sound spectra of rock crushing, a car passing, and wind is given in Fig. 7.

Two days of audio data were annotated and used to develop and evaluate the target detection system. The data was manually annotated (like in Fig. 5). The rock crushing activity is rather continuous and long-lasing, which made the annotation easy in most cases. In a few cases, the onset and offset of the target sound were hard to determine due to overlapping sound sources. In these cases, a 0.4 s uncertainty was associated to the onset or offset.

#### 4.3. Evaluation setup

A quantitative evaluation was made on the target noise detection performance with temporal resolution of one second. A two-folded validation, swapping the data of day 1 and day 2 for training and testing, was used. A detection output, either active or inactive,
was obtained through the proposed system for every one-second segment. The ground truth of a one second segment was seen as a target, when the target sound lasted longer than 0.1 s according to the annotation, otherwise judged as background noise.

The target source detection performance was evaluated using F-score [38], which is often used to evaluate binary classification performance. The F-score is calculated as a harmonic mean of precision and recall. Precision is the fraction of the predicted target activities that are correct, whereas recall is the fraction of the actual target activities that are predicted.

In order to study the feasibility of real-time execution and to find the most relevant factor to computation time, the computation time was evaluated for feature extraction and classification. The target detection algorithm was implemented in C++ and was run in a sensor node. The file read/write, SPL measurement, feature extraction, and classification process takes 51 s for a one minute signal, fast enough for real-time execution (85%). The sensor implementation was used as a benchmark and computation time of other feature extractors and classifiers were estimated using a Python implementation, assuming that the computation time had always the same proportion between the implementation in the sensor and in any other computer.

The developed classifier was imported to the sensor and it continuously performed noise measurement and source classification for 50 days. A reliability evaluation was made by examining the results transmitted to the web portal.

4.4. Evaluated classification systems

The acoustic features (MFCCs) from an audio signal which was sampled at 22,055 Hz. The audio signal was further analysed at a frame length of 100 ms with a 50% overlap between neighbouring frames and windowed with a Hamming window. 4096 point discrete Fourier transform and 40 Mel bands were used. Mel cepstral coefficients from the first to 8th were used as features. The number of coefficients (k) was studied as a variable. In addition to static MFCCs, the deltas were calculated using four preceding frames and four succeeding frames to represent temporal dynamics. The features were normalised to zero mean and unit variance was based on the statistics of the training dataset.

A single variable was changed at a time from the default setup to test the variable. Four variables were tested: the number of coefficients, the temporal dynamic features (deltas), the time-domain filters, and the frame length. The variable value that achieved the best performance was used to determine the next variable. To evaluate the performance of the feature extraction variables, a GMM with $M = 16$ components was used as a classifier.

The best achieving feature extraction setup was used to evaluate the classifiers. GMMs with a different number of Gaussian components $M = \{1, 2, 4, 8, 16, 32\}$ in Eq. (1) using diagonal covariance matrices and ANNs with two hidden layers, each having $\{10, 30, 50, 100\}$ neurons, were tested. Python toolboxes scikit-learn and pylearn2 were used in the implementation of the GMM classifier and the ANN classifier, respectively.

4.5. Results

The parts of the system were evaluated to select the features and the classifiers. A quantitative evaluation of the results is shown in Table 1. The selected values are shown in bold font and the computational requirements for the feature extraction is expressed as a time ratio to the estimate of real-time. Besides the detection
has to be taken into account when choosing the features and the classifier. The leading factor of the computation time in the source classification algorithm is the audio analysis frame length, which determines the number of frames to process. In comparison, the computation time is not much affected by other factors such as the neural network topology.

To evaluate the reliability of the proposed system, the hour-level measurement and detection results visualised in the web portal (Fig. 3) were examined. The sensors continuously performed noise measurement and source classification for 50 days and were able to transmit the results of every single hour, though some results were received with a delay of hours. It was assumed that the work in the plant could begin one hour later and end one hour earlier than the regular working hours (Mon–Fri, 8:00–15:00). With this assumption, almost all the target detection results were correct (1198/1200).

### 4.6. Required amount of annotated recordings

In the training material, about four hours of audio, the total number of feature vectors was about 300 000. A reduced size of the training material was tested by using every second or every fourth recording in time order. The learned classifiers achieved F-scores 0.913 and 0.924. Thus, it is sufficient to use about one hour of annotated recordings to achieve a decent classifier. When using the first half or the second half of a day data for training, the learned classifier achieved less than 0.8 F-score. This suggests that the training material should contain recordings from different times of a day to cover the most of the variability of the environmental sounds.

In environmental sound classification, a training set with a few hours can currently be regarded as a large dataset. For example, UrbanSound8K [40] contains one hour for each of 10 classes. As an example of small datasets, ESC-10 [41] contains at most 200 s audio for each of 10 classes.

In the reliability evaluation, it was shown that the system was able to do accurate hour-level classification in varying weather conditions by using annotated data of one day captured in good weather conditions. However, annotated recordings in more diverse conditions are typically required to achieve a similar accuracy as obtained in the quantitative evaluation with one second temporal resolution.

### 5. Further analysis and future work

#### 5.1. Selection of classifier

The performance of the classifiers GMM and ANN was practically the same in the evaluation. The selection between GMM and ANN should be based on other aspects. Adding a new class to the GMM classifier is easier, since statistics of the existing classes stay unchanged when a new class is added. In contrast to the GMM, the ANN has to re-estimate all the parameters to introduce a new class. Another benefit of using the GMM is that it is easier to adapt so that the classifier could adapt to small changes of the environment over time, using maximum a posteriori [16] algorithm. Typically ANN outperforms GMM when the number of classes is large. The number of the ANN parameters does not significantly increase with the number of classes, whereas the number of GMMs depends linearly from the number of classes. For example, ANN and GMM used approximately the same time in the computational time test for the binary classification. If there were ten classes in the same setup, the classification with ANN would have been about five times faster than GMM. In the one-minute audio test on a ten class case, it took 0.32 s for the ANN classifier and...
1.48 s for the GMM classifier using a Python implementation in a desktop computer.

5.2. Extension to monitoring multiple target classes

The same algorithm could be used in noise measurement scenarios involving multiple noise types. In addition to the rock crushing case study described above, a preliminary study on a set of noise samples from the port of Dublin was made. In this case, a classification system with multiple noise source classes was built. There are many kind of sound sources in the port area, some of them being also present in the neighbouring environment. The noise data was annotated with an interactive clustering method, by which a cluster of sounds were annotated or skipped at once. With this method, the annotation was fast but less accurate.

Ten classes of sound sources were present in the evaluation and the average recognition rate was 81%. The ten classes were alarm sounds, bird chirping, mild fans, strong fans, traffic noise, engine noise, footsteps, musical concert, raining, and wind blowing the microphone. It should be noted that the results might be optimistic since the segments not clearly belonging to any of the classes might have been skipped in the annotation with interactive clustering.

5.3. Sensor network

In the future, all the data from a large number of various networked sources, already available or from the autonomous smart sensors, will be centralised to a cloud service, where the data is accessible to a various groups of people: public, authorities, and to the dedicated users. The data will be made available for all the purposes it is needed: mapping and monitoring of emissions, noise, aerosols etc. It is possible to get accumulated standardised descriptors and conventional reports for various purposes. Also, it is possible to comment the visualised, and, possibly auralised, results on a time line to make feedback possible to the responsible party.

To increase the validity of the classification, multiple sensors could be used to also analyse the direction of arrival of sounds [42]. In future, the final outcome of an environmental noise assessment will be an annotation map of an area, reported with the level of uncertainty. Further, when the needs go beyond the current legislative values and limits, it is possible to calculate higher level of uncertainty. Further, when the needs go beyond the current legislation it is possible to calculate higher level of uncertainty.

The performance of the developed methods was evaluated in a rock crushing plant case study. The quantitative evaluation showed that the noise source classification using the proposed approach was accurate enough: on a temporal resolution of one second, F-score of 0.938 with the best investigated classifier was achieved. The system was run for 50 days and the results of the developed classifier matched well with the regular working hours of the rock crushing plant.

Also, a cloud service and a noise portal were introduced. The sensors transmitted the results to the cloud service and the portal was for visualisation of the results, statistical analysis, and data archiving. This approach makes it possible to extend the system towards noise management and, due to the minimal cost per sensor unit, towards real-time noise mapping with real measured data. By using this approach, the reliability, validity, and the spatial coverage of environmental noise monitoring will be increased.

References


