Tampere University

Aapo Honkakunnas

CHARACTERIZING AND DETECTING WIND NOISE IN AUDIO RECORDINGS

Master's thesis
Faculty of Engineering and Natural Sciences
Examiners: Prof. Robert Piche
D.Sc. Simo Ali-Löytty
April 2021

ABSTRACT

Aapo Honkakunnas: Characterizing and detecting wind noise in audio recordings
Master's thesis
Tampere University
Programme for Engineering and Natural Sciences
April 2021

Wind noise is a common nuisance when performing audio recording in outdoor situations. The aim of this thesis was to investigate different methods of characterizing wind noise occurring in audio recording situations, and to use these methods in wind noise detection and analyzing the behaviour of wind noise around a recording device. Four audio signal features zero-crossing rate, root mean square energy, sub-band spectral centroid and magnitude squared coherence were used in modeling the characteristics of wind noise with arguments presented for using them. Measurements were performed using a specific laboratory setup capable of measuring wind and recording audio. Recordings were performed outdoors with simultaneously recording a device in natural wind and another device inside a windshield and using devices with multiple microphones. Directly comparing the two simultaneous recordings a method for approximating absolute amount of wind noise present was suggested.

Wind detection was performed using logistic regression and Gaussian mixture model based classifiers, a Hidden Markov model was used in modelling the wind noise in different microphones around the recording device. Mathematical foundation for the methods was presented. The methods used were considered successful in characterizing and detecting the wind noise, with classifiers achieving high performance scores. The used methods also have potential to be applied in further considerations with different recording devices and data.

Keywords: wind noise, audio signal processing, Hidden Markov model, audio classification, likelihood, machine learning

TIIVISTELMÄ

Aapo Honkakunnas: Tuulimelun karakterisointi ja havaitseminen ääninauhoituksissa
Diplomityö
Tampereen yliopisto
Teknis-luonnontieteellinen DI-ohjelma
Huhtikuu 2021

Tuulimelu on yleinen ongelma ulkoilmassa suoritetuissa äänityksissä. Tämän työn tarkoituksena on tutkia erilaisia menetelmiä äänityksissä havaittavan tuulimelun karakterisointiin, sekä hyödyntää näitä menetelmiä tuulimelun havaitsemisessa sekä sen äänityslaitteen ympärillä käyttäytymisen tutkimisessa. Neljää äänisignaalin piirrettä, nollanylitysten nopeutta, neliöllistä keskiarvoenergiaa, alivyön spektrikeskusta sekä neliöityä koherenssia, käytettiin karakterisoimisessa. Perustelut käytölle esitettiin. Mittauksia suoritettiin käyttäen erityistä laboratoriojärjestelyä, joka mahdollisti tuulen mittaamisen sekä äänen nauhoittamisen. Nauhoitukset suoritettiin ulkona nauhoittaen samaan aikaan monimikrofonista laitetta luonnollisessa tuulessa ja toista samanlaista laitetta tuulisuojan sisällä. Tuulimelun absoluuttisen määrän arvioimiseen esitelttiin samanaikaisten nauhoitusten vertailua hyödyntävä menetelmä.

Tuulen havaitsemisessa käytettiin logistiseen regressioon sekä normaalisekoitemalliin perustuvia luokittelijoita. Markovin piilomallia käytettiin mallintamaan tuulimelun käyttäytymistä äänityslaitteen ympärillä. Menetelmien matemaattinen perusta esiteltiin. Käytetyt menetelmät suoriutuivat hyvin tuulimelun karakterisoimisessa ja havaitsemisessa. Luokittelijoiden arviointipisteet olivat korkeat. Käytettyjä menetelmiä voi hyödyntää myöhemmissäkin tarkasteluissa käyttäen erilaisia äänityslaitteita ja erilaista dataa.

Avainsanat: tuulimelu, äänisignaalinkäsittely, Hidden Markov- malli, äänen luokittelu, mallitodennäköisyys, koneoppiminen

PREFACE

This work is a master's thesis written for Tampere University and Nokia Technologies. I want to offer a big thank you to Nokia Technologies for an interesting topic and to my supervisors Robert Piche and Simo Ali-Löytty for giving feedback and helping to shape the work. Thank you Matti Hämäläinen, Ari Koski, Hannu Pulakka and Mikko Pekkarinen at Nokia that were a part of this project for providing opinions and also making me feel welcome in a new environment during these difficult times. Especially thanks to Miikka Vilermo at Nokia for constant guidance and discussion that really pushed this work to a next level. Also thanks to my family for supporting during this process.

Writing this thesis means that my life as a student is coming to an end. I am still overwhelmed at what a six years I have had here in Hervanta with loads of unforgettable experiences and life-long friends. I want to offer a huge thank for everyone involved in my student life. Thank you Hiukkanen, Elram, Tampereen teekkarit ry, Tampereen ylioppilaskunta and other organizations I have been privileged to be a part of. Thanks to my course mates for the experiences in academic matters and also thanks to everyone keeping me busy outside study hours, and perhaps too often during them too. Thank you TLDP, KuuNeuvosto, main_postaajat, Toto band project and all the numerous numerous big or small communities that have been around during this time. I have only a page available for thanks, so there is no way to mention everyone I would like to acknowledge, but finally thank you to everyone at Hiukkanen just for the moments of sitting at the guild room and drinking coffee. I am proud to have been a teekkari from Tampere and will carry that identity also now that I have to become an adult and quit being a student.

Tampereella, 23rd April 2021

Aapo Honkakunnas

CONTENTS

1. Introduction . . . 1
2. Theory concepts . . . 3
   2.1 Classification . . . 3
      2.1.1 Logistic regression . . . 4
      2.1.2 Gaussian Mixture Model . . . 7
      2.1.3 Evaluating classifier performance. . . . 11
   2.2 Hidden Markov Models . . . 13
3. Signal processing . . . 16
   3.1 Fundamental signal processing concepts . . . 16
   3.2 Generation of wind noise. . . . 17
   3.3 Detecting wind noise . . . 18
      3.3.1 Zero-crossing rate. . . . 20
      3.3.2 Root mean square energy . . . 21
      3.3.3 Spectral sub-band centroid . . . 22
      3.3.4 Approach for multiple microphones . . . 23
   3.4 Approximating absolute wind noise in recordings . . . 25
4. Measurements . . . 28
   4.1 The measurement setup . . . 28
   4.2 The measurement data . . . 29
      4.2.1 General . . . 29
      4.2.2 Clock issues during the measurements . . . 31
   4.3 Creating training and test datasets . . . 31
5. Findings of the study . . . 35
   5.1 Wind noise classification . . . 35
      5.1.1 Exploring the problem . . . 35
      5.1.2 Performance of the classifiers . . . 36
      5.1.3 Discussion about the classifiers . . . 38
   5.2 Wind noise in multiple microphones . . . 40
      5.2.1 Describing the model . . . 40
      5.2.2 Discussion . . . 43
6. Conclusion . . . 46

References . . . 48

LIST OF SYMBOLS AND ABBREVIATIONS

$\sum_{i=0}^{n}$	Sum of arguments with indices 0 to n sort
$A(\mu)$	Calibration coefficient matrix
$\mathcal{A}()$	Transition matrix of a HMM
acc(θ)	Accuracy of a classifier
$\mathcal{B}()$	Emission matrix of a HMM
BFGS	Broyden–Fletcher–Goldfarb–Shanno algorithm
C	Value of a response variable
$C_{12}(f)$	Magnitude squared coherence
$\chi^2(d)$	Chi-squared distribution with d degrees of freedom
$\cos()$	Cosine function
D_c	Device individual constant for wind interaction
DFT	Discrete Fourier Transform
e	Euler's number, $e \approx 2.71828$
$E(\lambda)$	Frame energy
$\mathcal{E}()$	Expected value of a probability distribution
EM	Expectation-Maximization
E_{RMS}	Root mean square energy
f	Frequency
$f()$	Loss function used for classification
f_s	Sample rate of a signal
G	Test statistic of likelihood ratio test
GMM	Gaussian mixture model
h()	Decision function
$\#()$	Cardinality of a set
HMM	Hidden Markov model
i	index
$\gamma(Y_{ij})$	Probability for a label in EM algorithm

k	Sample index of signal
$\frac{\partial F}{\partial x}$	Partial derivative of function F with respect to x
τ	Weight in a Gaussian mixture model
κ	Sample index inside a frame
$L(\theta)$	Log-loss function for a model
$\ln$	Natural logarithm
$l(\theta)$	Likelihood function for a model
Λ	Lagrange multiplier
K	Length of sequence of frames
$S()$	Discrete Fourier transformed signal in the frequency domain
λ	Frame index
μ	Frequency bin index
$\mathbb{N}$	Group of natural numbers
p	Sound pressure
$\sigma_E^2(\lambda)$	Short term energy variance
$ZCR(\lambda)$	Zero crossing rate
L_F	Frame length
M	Number of Gaussians in a GMM
m	Mean value of a gaussian distribution
$\arg\max_x$	Maximizing operator that gives the maximizing argument x
$\arg\min_x$	Minimizing operator that gives the minimizing argument x
MSC	Magnitude squared coherence
$N(\lambda, \mu)$	Frequency domain noise signal
n	Number of measurements
$n(k)$	Noise signal
$\mathcal{N}()$	Probability density function of a Gaussian
N_{tot}	Total noise difference
$P(\cdot)$	Probability
$P(x \mid y)$	Conditional probability of x given y
$\Phi_{12}(f)$	Power spectral density
$\pi(\cdot)$	Predicted probability for positive class in a classifier
$\prod_{i=0}^{n}$	Product of arguments with indices from 0 to n

prec(θ)	Precision of a classifier
PSD	Power spectral density
$Q(\theta, \theta_{r-1})$	Auxiliary function for EM algorithm
$\mathbb{R}$	Group of real numbers
r	Iteration index
rec(θ)	Recall of a classifier
RMS	Root mean square
$s()$	Logistic sigmoid function
$t(k)$	Target source signal
sgn$(x(k))$	Sign function of a signal
Σ	Covariance matrix of a Gaussian distribution
$SSC_m(\lambda)$	Spectral sub-band centroid
$T(\lambda, \mu)$	Frequency domain target signal
t	Time
θ	Classification model
Θ	Domain group for classification models
U	Wind velocity
W_i	State of a Hidden Markov model
$w(k)$	Windowing function for signal
X	Explanatory variables
$x_\lambda(k)$	Frame λ of a digital audio signal
$x(k)$	Digital audio signal
$\mathcal{X}$	Domain group for explanatory variables
$\overline{X}$	Mean value
$\lvert X \rvert$	Absolute value
Y	Response variables, labels
$\mathcal{Y}$	Domain group for labels

1. INTRODUCTION

Recording audio with different devices and in different situations has been getting more and more accessible and common during the latest years. The use of recordings has also been getting much more varied, with applications such as remote meetings, voice messages and social media content getting all the more popular and joining professional audio recordings and regular phone calls in common uses of audio. This has been largely made possible by the constant evolution of devices such as mobile phones [1], which can nowadays produce recordings with good quality even in a difficult environment with lots of background noise. The expectations for devices to be able to handle also different and difficult cases demands also more from the devices and requires constant evolving and research.

Dealing with different background noises is a fundamental part in producing quality audio recordings. In this work the focus is in wind noise, which is a common source of noise especially in recordings done outdoors and which has been found to have a particularly harmful effect in intelligibility and quality of the recording [2]. Protecting from the effects of wind noise is possible using different physical wind shields [3] but that is often not feasible in smaller devices such as mobile phones or hearing-aid devices [4].

Wind noise also has significantly different characteristics compared to most other common sources of noise, which makes it more difficult to reduce the effect of wind noise using signal processing and regular noise-canceling algorithms [5]. Instead a family of completely different noise-cancelling algorithms specifically for wind noise is required and this has been a topic of a lot of research [6]. The purpose and topic of this work is not to go through different algorithms for canceling wind noise, but to investigate the characteristics of wind noise and use them to detect its presence in recordings. That is a substantial step in the process of reducing wind noise [7]

In this work the generation and characteristics of wind noise are discussed according to what has been done in previous studies. A set of features of audio signals are presented and they are discussed in the context of using them for detecting wind noise. For these purposes a substantial amount of outdoor recordings in windy conditions are made, which is a different method compared to many other studies, where the wind noise is often added manually to samples that are studied [8].

In the recording process two recordings are done simultaneously: one with a device equipped with a windshield and one without. This approach provides a very efficient way to analyze the effects of wind compared to the reference recording. It also gives an opportunity to try to approximate the absolute amount of wind noise present with the assumption that after calibrating the microphone volumes most of the difference in the two recordings is due to the wind noise. This approach is used to investigate the performance of signal feature based classifiers in the task of detecting wind noise. In addition to this, the effects of the direction of arrival in the wind noise present are investigated using a Hidden Markov model exploiting the multiple microphones in the recording setup.

In Chapter 2 the mathematical background of the machine learning and modeling methods used in this study are discussed and in Chapter 3 the fundamental signal processing concepts needed in this study are presented. After that the reasons for the occurrence of wind noise are discussed, which leads to the investigation of the characteristics of the noise. The signal features used in this work are also presented in the chapter, as is the motivation and the process behind approximating the absolute wind noise. In Chapter 4 the measurement and recording setup is presented and the processing of measurement data is described, including the creation of training and testing datasets for the machine learning approaches. Training and performance of the wind noise classifiers is discussed and the Hidden Markov model approach is explained in Chapter 5.

2. THEORY CONCEPTS

2.1 Classification

In numerous statistical and machine learning applications the goal is finding the best possible model to describe the data available and then to use the model to predict some new data. In classification the goal of the analysis is to find such a model that divides the data into certain classes. In such cases every instance of data used is paired as (X_i, Y_i), where $X_i \in \mathcal{X}$ denotes an instance of input data that consists of explanatory variables that are different features of the data and $Y_i \in \mathcal{Y}$ denotes output data which is the label of the class. [9] The classification can be binary or multinomial; in binary classification the label set is $\mathcal{Y} = \{0, 1\}$. In this work the classification done is mainly binary.

Given an arbitrary model $\theta \in \Theta$ fitted to the data, the model is used in a decision function $h : \mathcal{X} \to \mathcal{Y}$ to predict the label given the features of the input data [10]. To find the best possible model to predict the labels as accurately as possible, a loss function that defines if the predicted label is correct or not is needed. In most applications the decision function and the loss function are difficult to calculate exactly and thus they can be approximated with a surrogate loss function that can be used to define the model [9].

Definition 2.1. Let $X \in \mathcal{X}^n$ and $Y \in \{0, 1\}^n$ be the input and output data and $\theta \in \Theta$ be a model that describes the data. Given an arbitrary surrogate loss function $f : \mathcal{X} \times \{0, 1\} \times \Theta \to \mathbb{R}_+$, the optimal model for binary classification is chosen by

$$\arg\min_{\theta} \frac{1}{n} \sum_{i=0}^{n-1} f(X_i, Y_i \mid \theta), \tag{2.1}$$

where n is the number of measurements in input and output data. [10]

Many different options exist for the kinds of models and decision functions chosen and also for fitting the model to the data. These are common discussions and issues in the field of machine learning and the choice of methods is related to the problem and data in question. The process of fitting the model is performed using a predetermined set of training data. If the training data contains output data in addition to the input data, the training process is called supervised learning and in the case of only input data being available the training is called unsupervised learning. [11] In both cases the goal is to find a model that is the best possible candidate in explaining the properties of the data. This is called maximizing likelihood and maximizing likelihood is identical to minimizing the loss function. [12] The likelihood that is maximized is a function of the model and describes the ability of the model to explain the data [13].

2.1.1 Logistic regression

Logistic regression is a classification method that fits a linear model to the training data. It predicts the probability of a data instance belonging in a class using the logistic function

$$s(t) = \frac{1}{1+e^{-t}}. \tag{2.2}$$

It is a commonly used method in different areas of study due to its relative simplicity and efficiency [14].

Definition 2.2 (Logit transformation)**.** Given a linear model $\boldsymbol{\theta}$ and an instance of input data X_i, the linear model gives the logarithmic odds of the corresponding response variable belonging to the class $Y_i = 1$

$$\ln\left(\frac{\pi(X_i)}{1-\pi(X_i)}\right) = \boldsymbol{\theta} X_i, \tag{2.3}$$

where $\pi(X_i)$ denotes the probability $P(Y_i = 1 \mid X_i, \theta)$. [12]

From the logit transformation it is also possible to calculate the predicted probability of class $Y_i = 1$ as

$$\pi(X_i) = \frac{1}{1+e^{-\boldsymbol{\theta} X_i}}, \tag{2.4}$$

which is the logistic function evaluated with the linear model. This shows that the infinite range of the linear model is mapped to the range $[0, 1]$ for classification purposes. In binary classification the response variable Y_i is Bernoulli distributed [10] and thus predicted probabilities for both classes can be calculated as

$$P(Y_i = C \mid X_i, \theta) = \begin{cases} \pi(X_i) & , C = 1 \\ 1 - \pi(X_i) & , C = 0. \end{cases} \tag{2.5}$$

The class which is more probable is assigned as the label for the instance of data [15].

The optimal linear model is fit using training data that has instances of input data with assigned labels. As seen in Figure 2.1, for data instances belonging in class $Y = 1$ the probability $P(Y_i = 1 \mid X_i, \theta) = 1$ and for instances belonging in class $Y = 0$, probability $P(Y_i = 1 \mid X_i, \theta) = 0$. The logistic regression model predicts the probabilities for classes and the linear model is fit with the goal of losing as little information as possible, as described in Definition 2.1.

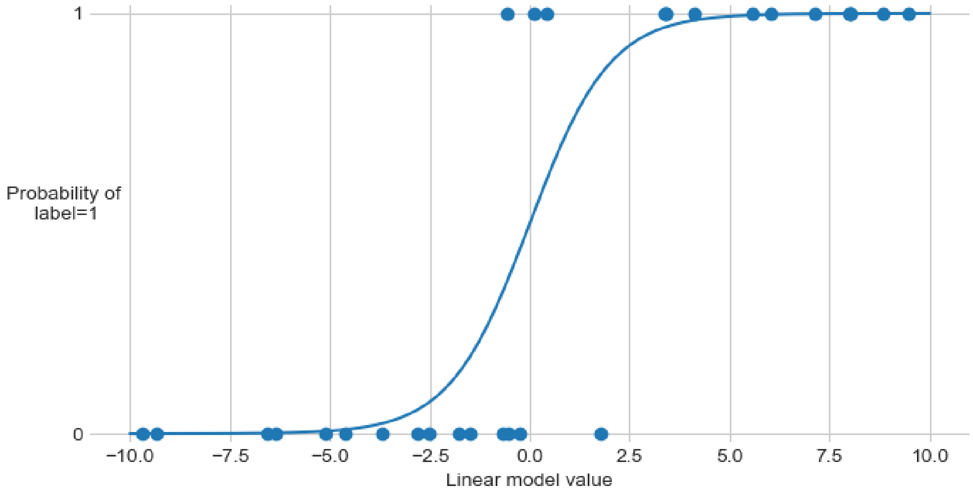


Figure 2.1. *Visualization of a logistic function that is fitted to binary training data*

Theorem 2.1. *The optimal linear model for the logistic regression is obtained by choosing the model*

$$\arg\max_{\theta} \sum_{i=0}^{n-1} Y_i \ln\left(\pi(X_i)\right) + (1 - Y_i)\ln\left(1 - \pi(X_i)\right), \tag{2.6}$$

where Y_i is the value assigned to the label and n is the amount of data instances in the training data.

Proof. Consider a linear model θ. This model predicts the probability of a training data instance X_i belonging in either class as seen in (2.5). The likelihood $l_i(\theta)$ of the model predicting the label correctly can be expressed for Bernoullian variables as [12]

$$l_i(\theta) = \pi(X_i)^{Y_i}(1 - \pi(X_i))^{1-Y_i}. \tag{2.7}$$

The data instances in the training data are assumed to be identically independently distributed [9] and thus the likelihood for the model on the course of the whole training dataset is the product

$$l(\theta) = \prod_{i=0}^{n-1} \pi(X_i)^{Y_i}(1 - \pi(X_i))^{1-Y_i}. \tag{2.8}$$

Taking logarithms, the log-likelihood is then

$$L(\theta) = \sum_{i=0}^{n-1} Y_i \ln\left(\pi(X_i)\right) + (1 - Y_i)\ln\left(1 - \pi(X_i)\right). \tag{2.9}$$

As the log-likelihood function is a sum of logarithms of likelihoods of the model predicting correctly in each instance, maximizing the log-likelihood function gives the best possible fit for the data. □

The log likelihood is usually used instead of likelihood because of numerical stability. With large datasets a lot of multiplications of numbers between zero and one are needed and thus eventually precision is lost. Maximizing the log-likelihood can be done with various optimization algorithms; quasi-Newton methods such as the Broyden–Fletcher–Goldfarb –Shanno (BFGS) algorithm are popular. [12]

The likelihood of the model can also be used to assess how well the model explains the data compared to another models. A commonly used statistical test for comparing two logistic regression models is the likelihood ratio test.

Definition 2.3 (Likelihood ratio test)**.** The likelihood ratio test between models θ_1 and θ_2 is performed using the test statistic

$$G = -2\ln\frac{l(\theta_1)}{l(\theta_2)}, \tag{2.10}$$

where $l(\theta_1)$ and $l(\theta_2)$ are likelihoods of the models and $G \sim \chi^2(d)$ with d degrees of freedom. Degrees of freedom d is calculated from the difference of dimensions between θ_1 and θ_2. [12]

Using log likelihoods instead of likelihoods, the test statistic can be written as

$$G = -2(L(\theta_1) - L(\theta_2)) \tag{2.11}$$

and thus it can be interpreted as describing the difference in the log likelihoods of the models. As the statistic is $\chi^2(d)$ distributed, the statistical significance of the likelihood difference of the two models can be determined using a hypothesis test [16]. If the test is considering whether model θ_2 is significantly better than θ_1, the p-value for the test is $P(\chi^2(d) > G)$. If the p-value is low, the null hypothesis of θ_1 is rejected and the hypothesis of θ_2 is accepted. This kind of test is often used when analyzing the features in the model and testing, whether all of the features are significant in building the model. [12]

2.1.2 Gaussian Mixture Model

Gaussian Mixture Models (GMM) are a way of representing characteristics of data by assigning the datapoints to different clusters. In a Gaussian Mixture Model, M Gaussian distributions are fitted to the data and the distribution is considered to be a superposition of all these distributions.

Definition 2.4. A Gaussian Mixture Model θ can be represented with parameters

$$\theta = \{\tau, \mathbf{m}, \mathbf{\Sigma}\}, \tag{2.12}$$

where $\tau = (\tau_1, \ldots, \tau_M)$ defines weights for each of the M Gaussians in the model, $\mathbf{m} = (m_1, \ldots, m_M)$ contains means for each Gaussian and $\mathbf{\Sigma} = (\Sigma_1, \ldots, \Sigma_M)$ contains each covariance matrix. [17]

The probability distribution of a GMM can be expressed as the linear superposition [18]

$$P(X_i) = \sum_{j=1}^{M} \tau_j \mathcal{N}(X_i \mid m_j, \Sigma_j), \tag{2.13}$$

where $\mathcal{N}(X \mid m_j, \Sigma_j)$ denotes the probability density function of the jth Gaussian distribution

$$\mathcal{N}(X \mid m_j, \Sigma_j) = \frac{1}{\sqrt{(2\pi)|\Sigma_j|}} \times e^{-\frac{1}{2}(X_i - m_j)^T \Sigma_j^{-1}(X_i - m_j)}. \tag{2.14}$$

Due to this probabilistic nature, a condition for the weights τ exists. The weights represent the probabilities of a sample belonging to a distribution and thus it is required that $\sum_{j=1}^{M} \tau_j = 1$ [17].

When a trained GMM is used for a classification problem, instead of calculating the whole mixture distribution, the probability for each of the individual distributions is calculated as

$$P_j(X_i) = \tau_j \mathcal{N}(X_i \mid m_j, \Sigma_j) \tag{2.15}$$

and is interpreted as the probability of data instance X_i originating to the jth Gaussian distribution. In this context the individual distributions are considered as response variables, i.e classes, and the component that has the highest weight for the data instance is assigned as the label. [19]

Gaussian Mixture Models are generally an unsupervised learning method, which means that for training the model only the explanatory input data is used, not the training labels that may be available for the user [9]. A maximum likelihood estimate for the model parameters is derived using an algorithm called the Expectation-Maximization algorithm.

Expectation-Maximization Algorithm

For finding the maximum likelihood model parameters, a likelihood function for one data instance X_i using an arbitrary model $\theta = \{\tau, \mathbf{m}, \mathbf{\Sigma}\}$ can be taken from Equation (2.13) as

$$l_i(\theta) = \sum_{j=1}^{M} \tau_j \mathcal{N}(X_i \mid m_j, \Sigma_j). \tag{2.16}$$

For a dataset of N instances the likelihood and the log-likelihood functions are derived similarly as in Equation (2.9) and thus the log-likelihood for training a GMM for the whole dataset is

$$L(\theta) = \sum_{i=1}^{N} \ln \left(\sum_{j=1}^{M} \tau_j \mathcal{N}(X_i \mid m_j, \Sigma_j) \right). \tag{2.17}$$

The maximum likelihood model can be then found maximizing the likelihood function with respect to the model parameters. In reality though, it is not feasible to use the log-likelihood instantly, as the summation over the M distributions inside the logarithm makes it really difficult to differentiate the function during calculating the maximum value [18].

The cause for this lies in the unsupervised nature of training a GMM. Instead of having data pairs $\{X_i, Y_i\}$ of input data and label, the training is only done with input data instances $\{X_i\}$ and it makes the likelihood function more complicated as it has one parameter less [18]. This gives the motivation for Expectation-Maximization (EM) algorithm that tries to iteratively estimate the full likelihood $L(X, Y \mid \theta)$ instead of $L(X \mid \theta)$ in Equation (2.17). The algorithm is frequently used in different kinds of mixture models, not just with the Gaussian mixture models.

Definition 2.5 (Expectation-Maximization Algorithm)**.** Expectation-Maximization algorithm for fitting parameters of a mixture model θ consists of three steps; steps 2 and 3 are iterated for $r = 1, 2, \ldots$

1. Choose initial values for model parameters θ_0
2. Estimate $Q(\theta, \theta_{r-1}) = \mathbb{E}(L(\theta) \mid X, \theta_{r-1}) = \sum_Y P(Y \mid X, \theta_{r-1}) \ln P(X, Y \mid \theta)$
3. Maximize $\theta_r = \arg\max_\theta Q(\theta, \theta_{r-1})$,

where $\mathbb{E}()$ is the expected value and $Q(\theta, \theta_{r-1})$ is an auxiliary function. Iteration is continued until convergence is obtained. [17]

In this general form steps 2 and 3 of the algorithm can be interpreted as calculating the probabilities for each of the data instances being generated by each Gaussian in step 2, and then updating the model to find the most likely parameters to produce the result of step 2 in step 3 [18]. Steps 2 and 3 are referred to as Expectation step and Maximization step. The algorithm has been proven to converge to a local likelihood maximum with each iteration step monotonically growing the likelihood $L(\theta)$ of the model [17].

In order to apply the EM algorithm for the context of Gaussian mixture models, the two probabilities in $Q(\theta, \theta_{r-1}) = \sum_Y P(Y \mid X, \theta_{r-1}) \ln P(X, Y \mid \theta))$ need to be presented in terms of Gaussian mixture models. The conditional probability $P(Y \mid X, \theta_{r-1})$, which describes the probability of each class given the data and the model update from the last iteration, can be calculated using Bayes' theorem as

$$P(Y_i = C \mid X_i, \theta_{r-1}) = \frac{\tau_C \mathcal{N}(X_i \mid m_C, \Sigma_C)}{\sum_{j=1}^{M} \tau_j \mathcal{N}(X_i \mid m_j, \Sigma_j)} \equiv \gamma(Y_{iC}). \tag{2.18}$$

In the equation C is one of the possible classes $C \in \mathcal{Y}$ and $\gamma(Y_{ij})$ is the notation that will be used for this probability of the ith data instance being classified in the jth class. This is the expectation calculation that is performed in the expectation step for a GMM. [18] The second probability $\ln P(X, Y \mid \theta)$ denotes the total log-likelihood given a GMM θ and with labels Y known. It can be calculated as

$$\ln P(X, Y \mid \theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} P(Y_{ij})(\ln \tau_j + \ln \mathcal{N}(X_i \mid m_j, \Sigma_j)), \tag{2.19}$$

where the probability $P(Y_{ij}) = 1$ for one Gaussian of M and $P(Y_{ij}) = 0$ for other labels. [17] This likelihood function is now of a much easier form to maximize with derivatives than the previous one in Equation (2.17) without knowledge of Y. Combining the $\gamma(Y_{ij})$ and the $\ln P(X, Y \mid \theta)$ calculated earlier, the function $Q(\theta, \theta_{r-1})$ to be maximized in the maximization step is now

$$Q(\theta, \theta_{r-1}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \gamma(Y_{ij})(\ln \tau_j + \ln \mathcal{N}(X_i \mid m_j, \Sigma_j)) \tag{2.20}$$

and it can be maximized with respect to the parameters $\{\tau, \mathbf{m}, \mathbf{\Sigma}\}$ using derivatives and finding the zero point.

For m_j in the model, the update m_j^* can be found by setting

$$\frac{\partial Q(\theta, \theta_{r-1})}{\partial m_j} = 0 \tag{2.21}$$

$$\frac{\partial}{\partial m_j} \sum_{i=1}^{N} \sum_{j=1}^{M} \gamma(Y_{ij}) \ln \mathcal{N}(X_i \mid m_j, \Sigma_j) = 0 \tag{2.22}$$

$$\begin{aligned} \frac{\partial}{\partial m_j} \sum_{i=1}^{N} -\frac{\gamma(Y_{ij})}{2}(N \ln 2\pi + \ln |\Sigma_j| + \\ (X_i - m_j)^T \Sigma_j^{-1} (X_i - m_j)) = 0, \end{aligned} \tag{2.23}$$

where part of the $Q(\theta, \theta_{r-1})$ was omitted as it was only dependent on the value of τ_j. The sum over M was omitted because only the index j is considered. The m_j^* is solved from the derivative [20]

$$\sum_{i=1}^{N} \gamma(Y_{ij}) \Sigma^{-1} (X_i - m_j) = 0 \tag{2.24}$$

$$\sum_{i=1}^{N} \gamma(Y_{ij}) X_i - \sum_{i=1}^{N} \gamma(Y_{ij}) m_j = 0 \tag{2.25}$$

$$m_j = \frac{\sum_{i=1}^{N} \gamma(Y_{ij}) X_i}{\sum_{i=1}^{N} \gamma(Y_{ij})}. \tag{2.26}$$

Similarly for Σ_j, the derivative is taken with respect to Σ_j^{-1} [20] and it can be taken starting from Equation (2.23). Now the updated Σ_j^* is calculated

$$-\frac{1}{2} \sum_{i=1}^{N} \gamma(Y_{ij}) (\Sigma_j - (X_i - m_j)(X_i - m_j)^T) = 0 \tag{2.27}$$

$$\sum_{i=1}^{N} \gamma(Y_{ij}) \Sigma_j - \sum_{i=1}^{N} \gamma(Y_{ij}) (X_i - m_j)(X_i - m_j)^T = 0 \tag{2.28}$$

$$\Sigma_j = \frac{\sum_{i=1}^{N} \gamma(Y_{ij}) (X_i - m_j)(X_i - m_j)^T}{\sum_{i=1}^{N} \gamma(Y_{ij})}. \tag{2.29}$$

For calculating the weights τ, a constraint is needed to make sure the requirement $\sum_{j=1}^{M} \tau_j = 1$ is fulfilled. It is done using a Lagrange multiplier Λ and for the maximization the function $Q(\theta, \theta_{r-1})$ is now [17]

$$Q(\theta, \theta_{r-1}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \gamma(Y_{ij}) (\ln \tau_j + \ln \mathcal{N}(X_i \mid m_j, \Sigma_j)) - \Lambda(\sum_{j_1}^{M} \tau_j - 1). \tag{2.30}$$

It is differentiated with respect to τ_j and set to zero

$$\frac{\partial Q(\theta, \theta_{r-1})}{\partial \tau_j} = 0$$

$$\sum_{i=1}^{N} \frac{\gamma(Y_{ij})}{\tau_j} - \Lambda = 0 \tag{2.31}$$

$$\tau_j = \frac{\sum_{i=1}^{N} \gamma(Y_{ij})}{\Lambda}.$$

The value for Λ can be calculated by considering the summation over M on both sides of the equation. Both sums $\sum_{j=1}^{M} \gamma(Y_{ij})$ and $\sum_{j1}^{M} \tau_j$ equal to one and thus the value of $\Lambda = N$. Now the update for weight τ_j is

$$\tau_j^* = \frac{\sum_{i=1}^{N} \gamma(Y_{ij})}{N}. \tag{2.32}$$

The EM algorithm for Gaussian mixture models can thus be defined.

Definition 2.6 (EM algorithm for Gaussian mixture models)**.** The expectation-maximization algorithm for finding the maximum likelihood Gaussian mixture model $\theta = \{\tau, \mathbf{m}, \mathbf{\Sigma}\}$ consists of three steps

1. Initial value for model $\theta_0 = \{\tau_0, \mathbf{m}_0, \mathbf{\Sigma}_0\}$
2. Calculate $\gamma(Y_{ij}) = \frac{\tau_j \mathcal{N}(X_i|m_j,\Sigma_j)}{\sum_{j=1}^{M} \tau_j \mathcal{N}(X_i|m_j,\Sigma_j)}$
3. Calculate updated parameters $m_j^* = \frac{\sum_{i=1}^{N} \gamma(Y_{ij}) X_i}{\sum_{i=1}^{N} \gamma(Y_{ij})}$, $\Sigma_j^* = \frac{\sum_{i=1}^{N} \gamma(Y_{ij})(X_i - m_j)(X_i - m_j)^T}{\sum_{i=1}^{N} \gamma(Y_{ij})}$ and $\tau_j^* = \frac{\sum_{i=1}^{N} \gamma(Y_{ij})}{N}$.

Steps 2 and 3 are repeated until convergence criteria is met. [18]

All of the updated parameters can be interpreted as a weighted mean of the data with the weights being calculated in the expectation step [17]. Using the EM algorithm for training a GMM requires a certain level of knowledge about the data that is being used due to some of its drawbacks, such as issues in identifiability of individual classes. The EM algorithm does not guarantee convergence to a global maximum but only a local one and thus the result is dependent on the initial value. [18]

2.1.3 Evaluating classifier performance

Validating and evaluating the performance of the classifier is an important part of creating one. For the testing part another dataset consisting of instances of input data that has been assigned with a correct label is needed. This test data is classified using the trained model and the predicted labels are then compared to the correct labels [11]. Various metrics are used to measure the success of the predicting. [21]

Definition 2.7. Consider a sequence of n labels $\hat{Y}$ predicted by a classifier θ and a sequence of corresponding true labels Y. The *accuracy* of the classifier can be calculated as

$$\text{acc}(\theta) = \frac{\#(\hat{Y}_i = Y_i)}{\#(Y_i)}, \tag{2.33}$$

where $\#()$ notes the cardinality of the set. [22]

Accuracy gives a good indication of the success of the classification but it does not give any information about the type of errors the classifier does. More thorough evaluation about the errors made is possible by comparing the actual label $_i$ and the predicted label $\hat{Y}_i$ for each data instance. A confusion matrix, where each item of the matrix represents number of occurrences for each combination of true and predicted labels, is a common way to present this information [23]. For binary classifiers a confusion matrix is illustrated in Table 2.1.

	$\hat{Y}_i = 0$	$\hat{Y}_i = 1$
$Y_i = 0$	$\#(\hat{Y}_i = 0 \mid Y_i = 0)$	$\#(\hat{Y}_i = 1 \mid Y_i = 0)$
$Y_i = 1$	$\#(\hat{Y}_i = 0 \mid Y_i = 1)$	$\#(\hat{Y}_i = 1 \mid Y_i = 1)$

Table 2.1. *Confusion matrix for binary classifiers*

In some applications for binary classification particular interest in the performance of the classifier is related to the ability of predicting a certain class, such as finding cases that are labeled positive. The items of the confusion matrix give an opportunity to define indicators that produce information about the performance related to a specific class.

Definition 2.8. Consider a sequence of n labels $\hat{Y}$ predicted by a classifier θ and a sequence of corresponding true labels Y. The *precision* of the classifier for class $Y = 1$ can be calculated as [21]

$$\text{prec}(\theta) = \frac{\#(\hat{Y}_i = 1 \mid Y_i = 1)}{\#(\hat{Y}_i = 1)} \tag{2.34}$$

and the *recall* for class $Y = 1$ is calculated as [21]

$$\text{rec}(\theta) = \frac{\#(\hat{Y}_i = 1 \mid Y_i = 1)}{\#(Y_i = 1)}. \tag{2.35}$$

Precision can be interpreted as the ability of the classifier to predict positive labels only for instances that are true positives. Recall on the other hand describes how big part of the true positive instances the classifier predicts as positive. [24] Calculating both precision and recall gives a good indication of how the classifier is able to predict in context of the wanted class, and for a perfect classifier both prec$(\theta) =$ rec$(\theta) = 1$. In realistic situations a tradeoff situation exists between precision and recall and depending on the preferred classifier characteristics a model can be chosen [11]. The tradeoff is often described with a precision-recall-curve, as is shown in Figure 2.2. The curve is plotted by varying the values of the decision probability threshold for classification between $P(\hat{Y}_i = 1) = [0, 1]$ and calculating precision and recall for each threshold [21]. The performance can be analyzed from the curve and the north-east corner in the graph resembles a perfect classifier. The closer the curve is to that point, the better the classifier is [24].

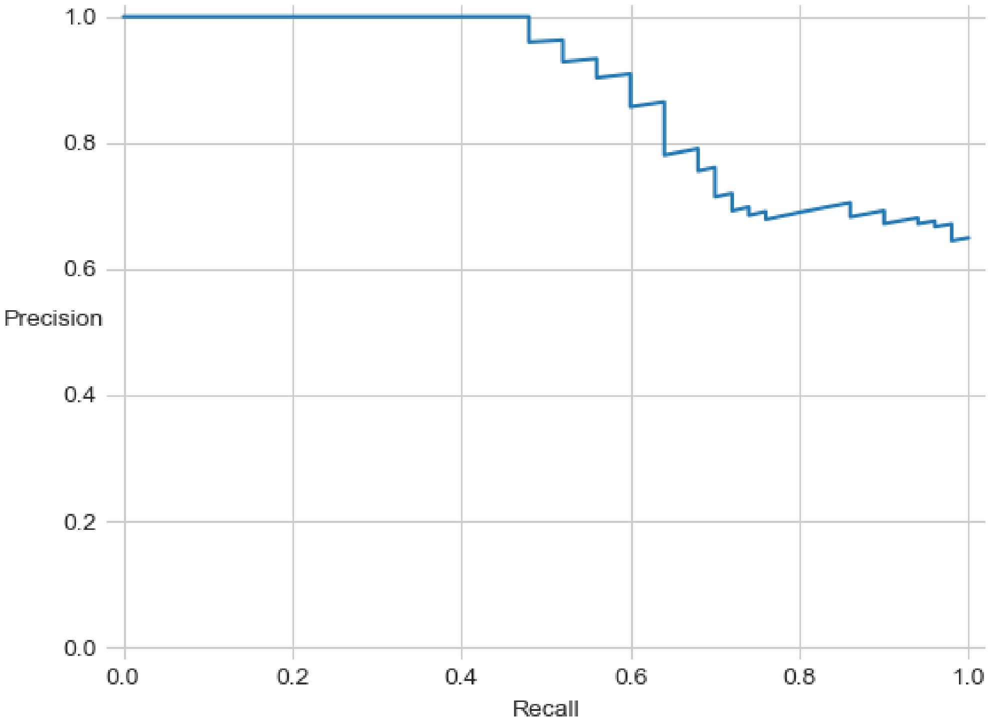


Figure 2.2. *A precision-recall curve with precision and recall computed with different thresholds*

2.2 Hidden Markov Models

In the statistical models presented in Section 2.1 the data was assumed to be independently identically distributed, so that there was no dependence between individual data instances. This however is not the case in many practical applications, where the data that is modelled is sequential in nature. Hidden Markov model (HMM) can be used to process the data in these kind of situations, where the input data $X = \{X_1, \dots X_T\}$ and response data $Y = \{Y_1, \dots Y_T\}$ are sequences and individual data instances have an effect on the others [17]. A common example of such data is a time series of measurements, where the past measurements are predictive of the future measurements. In Hidden Markov models, such stochastic processes are assumed to behave according to the Markov property, which makes the processes to be called Markovian.

Definition 2.9 (Markov property)**.** The Markov property for the conditional probability of sequential data holds, if

$$P(Y_t \mid Y_{t-1}, Y_{t-2}, \dots, Y_1) = P(Y_t \mid Y_{t-1}). \tag{2.36}$$

In Markovian processes the future measurements are only affected by the current state and the process is time-invariant. [18]

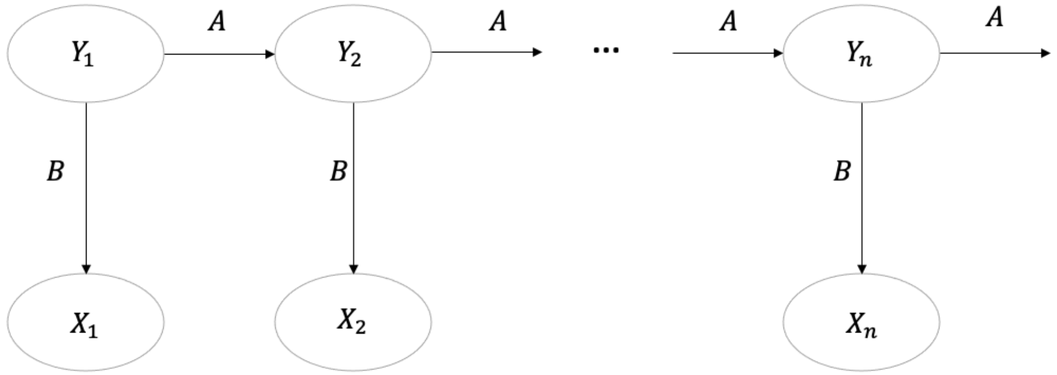


Figure 2.3. *The structure of a Hidden Markov model, where the states Y_i are sequential and observations X_i are emitted by the states*

A Hidden Markov model consists of a Markovian sequence of states $Y_{1:T}$ and a sequence of observations $X_{1:T}$ over a time interval of length T. The architecture of a HMM is illustrated in Figure 2.3. The states Y are called hidden states due to their nature of being unobservable. In a Hidden Markov model the values of the states are discrete and $Y_i \in \mathcal{Y}$ [9]. For the observations a model specific observation model is used. And as seen in Figure 2.3, each observation is conditionally independent of other observations, given the current state. Thus the joint probability distribution of the state and observation sequences is

$$\begin{aligned} P(Y_{1:T}, X_{1:T}) &= P(Y_{1:T})P(X_{1:T} \mid Y_{1:T}) \\ &= \left(P(Y_1) \prod_{t=2}^{T} P(Y_t \mid Y_{t-1}) \right) \left(\prod_{t=1}^{T} P(X_t \mid Y_t) \right), \end{aligned} \tag{2.37}$$

where $P(Y_1)$ is the prior probability of the states, $P(Y_t \mid Y_{t-1})$ is the Markovian conditional probability of state at time t given the previous state and $P(X_t \mid Y_t)$ is the conditional probability of the observation given the state [17].

Definition 2.10. A time-invariant Hidden Markov model θ can be defined with parameters

$$\theta = \{\tau, A, B\}, \tag{2.38}$$

where τ is the prior probability distribution of state Y_1, A is the time-invariant state transition matrix and B is the emission matrix for observations. [18]

The transition matrix A consists of probabilities of transitioning from each possible state to each possible state, such as $A_{ij} = P(Y_t = j \mid Y_{t-1} = i)$. The emission matrix consists of conditions for expected observations given each state and it depends on the observation model used. A common choice is to use Gaussian emissions and in these cases $B_j = \mathcal{N}(X_t \mid \mu_j, \Sigma_j)$. [17] A HMM with Gaussian emissions can be considered as a Gaussian mixture model with the states being sequential.

The parameters $\{\tau, A, B\}$ for Gaussian emissions can be learned using both unsupervised and supervised learning methods, depending on if the training data is labeled. For unsupervised learning, the parameters are learned using the Baum-Welch algorithm, which is a special case of the EM algorithm presented in Section 2.5 [18]. For supervised learning, the transition matrix A and the other parameters can be calculated using the training labels Y_i. Transition matrix is calculated by counting the occurrences for each transition as

$$A_{ab} = \frac{\#(Y_{i+1} = b \mid Y_i = a)}{\#(Y_i = a)}, \tag{2.39}$$

where a and b are states of the model. [25] The prior probabilities τ can be determined from the relative occurrences of each label in the training data set as

$$\tau_a = \frac{\#(Y_i = a)}{\#(Y)}. \tag{2.40}$$

If the HMM has Gaussian emissions, the emission distributions for each state are determined from the sample means and covariances of the data instances belonging to each label. Thus [25]

$$B_a = \mathcal{N}(m_a, \Sigma_a). \tag{2.41}$$

The trained parameters can give a lot of information about the process that is being modelled with the HMM and defining the parameters by training a model is one of the important questions that can be answered using Hidden Markov models. Another important use of Hidden Markov models is a decoding problem, where given the parameters of the model, the goal is to find the most likely state sequence Y given a sequence of observations X. A common method of solving this problem is using the Viterbi algorithm. [26]

3. SIGNAL PROCESSING

3.1 Fundamental signal processing concepts

An audio signal is created when a microphone or a similar transducer device senses vibrations of pressure in a medium and produces an electric signal $x(t)$. This signal is a continuous analog signal and it can be transformed into a digital signal for contemporary signal processing using an analog-digital conversion. In this process the signal is filtered, quantized and sampled to the wanted sample rate f_s. The end product is a discrete signal $x(k)$ that consists of samples of the original waveform. The number of samples per second is called the sample rate. [9]

The recording device usually captures audio energy from several different sources. Depending on the purpose of the recording, one or several of the sources can be classified as the wanted target sources and the rest of them are noise sources. Possible sources for noise can be different background events and processes, such as wind being present near the recording device. [7] In this work, two recordings are made simultaneously and with nearly identical placements related to the target source and background noise sources. One of the recordings is made inside a wind shield that is assumed acoustically invisible subject to slight calibrations, which leads to the assumption that the sole difference between the two recordings is the wind noise. With this assumption, the two recorded signals x_1 and x_2 can be modeled as

$$\begin{aligned} x_1(k) &= t(k) \\ x_2(k) &= t(k) + n(k) = x_1(k) + n(k), \end{aligned} \tag{3.1}$$

where $t(k)$ consists of the target source and different background noises and $n(k)$ is the wind noise signal.

The signal representation $x(k)$ describes the amplitude of the signal at a specific time and thus gives information about the signal in the time domain. The time domain gives some information about the properties of the signal in question, but for purposes of analyzing the signal, it is usually much more useful to consider the signal in the frequency domain. [27] This transform from the time domain to the frequency domain can be done by dividing the time domain signal into frames and then using the Fourier transform.

The framing is done by taking a frame length L_F of samples and then multiplying the sequence with a window function. This windowing is done in such a way to reduce spectral leakage and other unwanted effects. A commonly used window function is the Hann window

$$w(\kappa) = \frac{1}{2} - \frac{1}{2} \cdot \cos\left(\frac{2\pi\kappa}{L_F}\right), \tag{3.2}$$

where $\kappa = 0, \ldots, L_F - 1$. [28] Thus the frame with index λ can be obtained with

$$x_\lambda = \sum_{\kappa=0}^{L_F-1} w(\kappa) \cdot x(\lambda \cdot \frac{L_F}{4} + \kappa), \tag{3.3}$$

where $\frac{L_F}{4}$ is the windowing step size used. For these frames the Fourier transform can then be used.

Definition 3.1. The Discrete Fourier Transform (DFT) is the discrete version of the Fourier transform for converting a function from time domain into the frequency domain. For a sequence of L_F samples, it can be calculated as

$$S(\lambda, \mu) = \sum_{\kappa=0}^{L_F-1} x_\lambda(\kappa) \cdot e^{\frac{-i2\pi\mu\kappa}{L_F}}, \tag{3.4}$$

with $\mu = 0, \ldots L_F - 1$ being the discrete frequency bin indices and κ being the sample index in a single frame. The value $S(\lambda, \mu)$ is a complex number that represents the magnitude and phase of the given frequency μ being present in the frame λ. [9]

When the DFT is performed for a sequence of consecutive frames, the output is a representation of the signal in the time-frequency domain. This representation gives information about the spectral components given by the DFT and about the timestamp of the given frame. This gives the opportunity to construct different signal features and visualizations such as spectrograms [29].

In this work the computation of DFTs and frames is done with the Python library `librosa`, which uses the FFT algorithm for computing the DFT. It requires the frame lengths to be of the form $2^n, n \in \mathbb{N}$ and unless stated otherwise, in this work the used frame size is $L_F = 2048$.

3.2 Generation of wind noise

The process of generation of wind noise in microphones has been a subject of research for a long time. The main reason for wind noise in recordings is the turbulent pressure fluctuations present in the air flow beyond the microphone [30]. These fluctuations interact with the microphone membrane and induce a noise signal in the same way the microphone would interact with the sound pressure created by the wanted sound source. [7]

The turbulences experienced by the recording device can be roughly divided into two categories: the intrinsic turbulences that occur in the air flow and the eddies and vortices that are created when the air flow encounters the edges of the device. The intrinsic turbulences are created in a much larger scale than the recording situation and they occur when the wind stream encounters obstacles such as trees, buildings or vehicles in its boundary layer. [31] In multiple experiments it has been seen that in outdoor measurements the main source of wind noise are the intrinsic turbulences. That does not mean that vortices are not created when the air flow hits the device: the air flow that creates these vortices is just not constant and therefore also the frequency and direction of the vortices is very inconsistent. This results in the vortices being unable to create a clear and strong noise spectrum, as the vortice effects often cancel each other. [6, 32]

The characteristics of induced wind noise in different conditions have been investigated extensively and a strong correlation between the wind velocity and level of wind noise has been noticed. [32] In one representation, the sound pressure $p(f)$ of the wind noise in different frequencies can be modeled as

$$p(f) \propto U^{3.15} f^{1.65}, \tag{3.5}$$

where U is the wind speed and f is the frequency [33]. The model is experimental in nature and it may be valid only for the microphones and devices used to derive it, but it shows correlation between wind noise and the wind speed. Finding relationships between wind noise and the direction of the wind is much more difficult because the wind direction varies heavily due to the turbulent nature of the windy air flow in outdoor measurements.

The nature of the vortices that are shed by the recording device depend heavily on the geometry and material of the device. That means that it is a good practice to look for correlations in induced wind noise and wind conditions is by performing measurements with the single device in question. This also aids in taking the effects of microphone placements in devices into account. In many devices the microphones are mounted inside the outer shell of the device and the slits above the microphones can also generate additional vortices that generate noise. [7] In this work data is obtained from multiple microphones located all over the device and this gives a possibility to compare the generation of wind noise in different conditions depending on microphone placement.

3.3 Detecting wind noise

Detecting wind noise and reducing its effect in recordings is an important topic of study in the audio processing community. The ability to do it effectively requires knowledge and understanding of the characteristics of wind noise. For a listener the wind noise is easily identifiable even among other possible noise in the recording. The rapidly changing low-frequency whooshing sound is characteristic only to wind noise and these properties can also be used to identify if a recording has wind noise present. Both of these clear properties arise from the mechanics of generation of wind noise, discussed in section 3.2.

Definition 3.2. When an audio signal is divided to frames of size L_F, the frame energy can be calculated as

$$E(\lambda) = \sum_{k=\lambda \cdot L_F + 1}^{\lambda \cdot (L_F + 1)} x(k)^2, \tag{3.6}$$

where λ is the frame index. From a sequence of K frames, the short term energy variance can be defined

$$\sigma_E^2(\lambda) = \frac{1}{K} \sum_{i=\lambda-(K-1)/2}^{\lambda+(K-1)/2} (E(i) - \overline{E(\lambda)})^2, \tag{3.7}$$

where $\overline{E(\lambda)}$ is the mean of the frame energies in the sequence. [6]

When compared to other common types of encountered noises, such as pub noise [34], it can be noticed that wind noise has much larger short term energy variance [8]. It means that a commonly used assumption of background noise level being constant cannot be applied with wind noise. This temporal variance is what can be heard as constantly fluctuating noise level in recordings with wind noise.

In addition to the variance, another identifiable property of wind noise is its frequency spectrum that is concentrated heavily on the lower frequencies [35]. Visualizing the issue, from the spectrogram in Figure 3.1 it can be seen that most of the energy of the windy portion is concentrated below 500 Hz, while in the less windy part the energy is distributed more evenly.

Analysing and detecting the described temporal and spectral characteristics of wind noise is possible using various signal processing techniques.[6] A common course of action is to use different feature extraction methods that are more efficient to compute than full spectrograms or variance analyses but still describe the same properties. Due to the highly non-stationary characteristic of wind noise, these features must also be computable in short time intervals. A collection of commonly used and well-working audio features are discussed in the following sections.

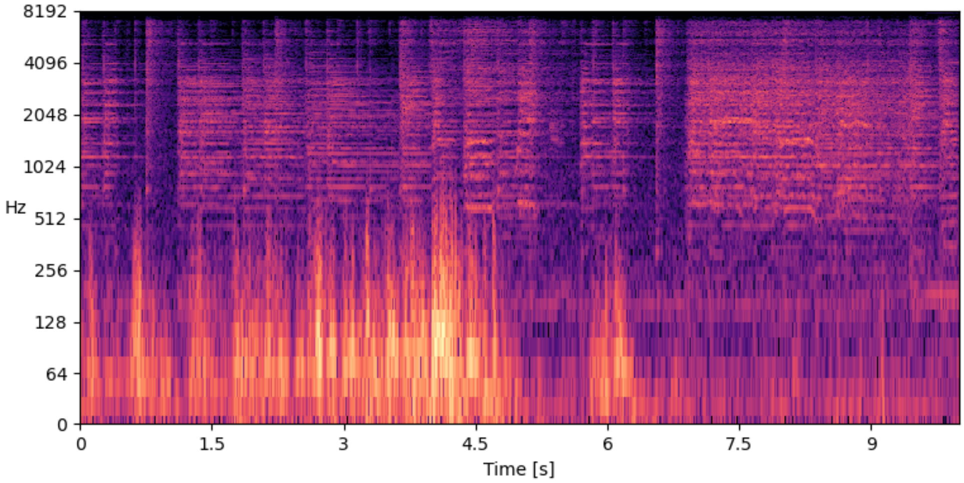


Figure 3.1. *A spectrogram of a 10 seconds long sample of the recordings. A wind gust with speed $U = 1.9$ m/s takes place during the first half of the snippet and the second half has less than 0.5 m/s of wind present.*

3.3.1 Zero-crossing rate

Zero-crossing rate (ZCR) is a commonly used and simple feature of audio signal. It describes how rapidly the signal changes its sign, i.e. crosses zero.

Definition 3.3. Zero-crossing rate of a frame λ of audio signal is

$$ZCR(\lambda) = \frac{1}{L_f} \sum_{k=0}^{L_f-1} |\text{sgn}(x_\lambda(k)) - \text{sgn}(x_\lambda(k-1))|, \tag{3.8}$$

where the function $\text{sgn}(\cdot) = \begin{cases} 1 & , x(k) \geq 0 \\ -1 & , x(k) < 0 \end{cases}$ denotes the sign of the signal. [27]

The rate of signal changing its sign is heavily related to the frequency of the signal, which makes it a useful feature in getting coarse information about the frequency components of the signal. Its simplicity and computational feasibility make it attractive, even though it doesn't have as much explanation power as some other more complicated features. Common use cases for zero-crossing rate are voice activity detectors, where it is often used together with short term energy [36].

In the wind detection context, zero-crossing rate is potentially useful, because of its ability to give information about the spectral characteristics of the noisy signal. Wind noise is more active in lower frequencies than the recorded target signal and this means that we can expect zero-crossing rates to be small in the windy parts and larger in parts where wind noise is not present. This is also seen in Figure 3.2, where the zero-crossing rates of same snippet of recording as in Figure 3.1 are plotted along with the ZCRs of the wind

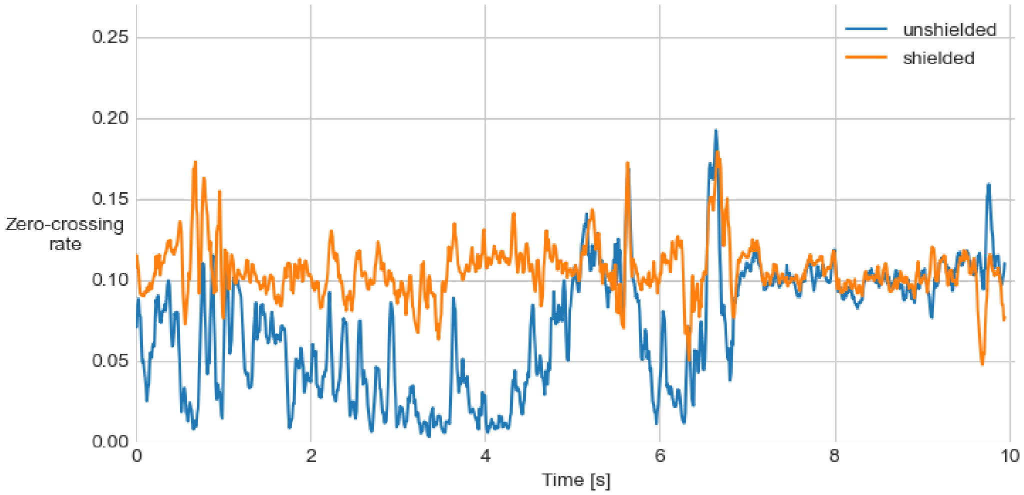


Figure 3.2. *Zero-crossing rates of two signals recorded at the same time and place with and without wind shielding.*

protected audio from the same timestamp. This property can be used in detecting the presence of wind noise. [6]

3.3.2 Root mean square energy

Another useful temporal feature of a signal is its energy, which can provide information about the energy and loudness of the signal. Audio signals are waveforms that oscillate around zero with both negative and positive amplitudes with both contributing to the energy of the signal with their magnitude, not their sign. Using a root mean square calculation gives information about the energy in both positive and negative amplitudes and thus the root mean square energy value is a useful feature in assessing the signal energy during a frame [37].

Definition 3.4. The root mean square (RMS) energy of a frame λ of audio signal is calculated as [29]

$$E_{RMS}(\lambda) = \sqrt{\frac{1}{L_f} \sum_{k=0}^{L_f - 1} x_\lambda(k)^2}. \tag{3.9}$$

Using the RMS energy as a feature it is possible to assess the loudness of the signal during each frame. It is useful information in the context of detecting wind noise because the noise created by wind gusts can often be louder than the target signal. That occurs especially in situations where the wind noise disturbs the recording dominantly [8]. As also seen in Figure 3.3, the RMS energy in the unprotected microphone rises significantly when wind occurs.

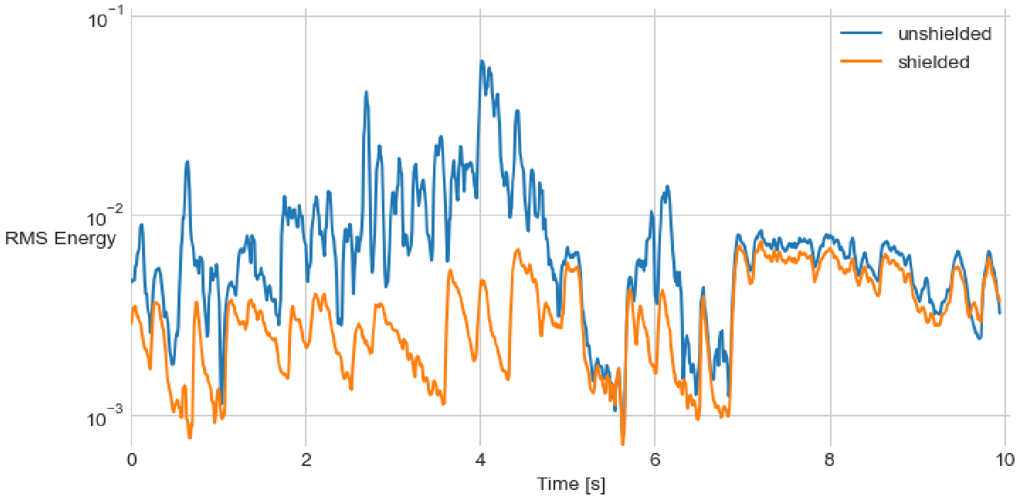


Figure 3.3. *RMS energies of two signals recorded at the same time with and without wind shielding.*

3.3.3 Spectral sub-band centroid

Spectral centroid is a feature that provides information about the distribution of signal energy in different frequencies. It can be stated to be the center of the mass of the spectrum. [29] The spectral sub-band centroid (SSC) that is in question in this work is similar, but instead of using the full frequency domain, it is divided into smaller pieces that are called sub-bands. For sub-band spectral centroids the centroid calculation is performed for only the necessary sub-band in order to get information about that area of the frequency domain. In order to calculate spectral centroids for a signal, substantial information about the frequencies of the signal is required. This is obtained by transforming the signal from the time domain to the frequency domain using the DFT defined in Equation (3.4).

Definition 3.5. The spectral centroid of the frame λ for the ith sub-band of the signal frequency domain can be calculated as

$$SSC_i(\lambda) = \frac{f_s}{L_F} \frac{\sum_{\mu=\mu_{i-1}}^{\mu_i - 1} \mu \cdot |S(\lambda, \mu)|}{\sum_{\mu=\mu_{i-1}}^{\mu_i - 1} |S(\lambda, \mu)|}, \tag{3.10}$$

where f_s is the sample rate of the signal, L_F is the frame length, μ is the central frequency of the frequency bin, $S(\lambda, \mu)$ is the DFT value of the frame and frequency bin in question and μ_i and μ_{i-1} represent the edges of the sub-band. [38]

The ability to give information where the energy in concentrated in the frequency spectrum makes spectral centroid a successful method in analyzing the timbre of the audio signal. This has made it a frequently used feature in different music classification and detection tasks. [27] It is also used in automatic speech recognition solutions to help classification between voiced and unvoiced speech [38].

In the wind noise detection context, using spectral centroids is useful because we know that the wind noise spectrum is heavily concentrated in the low frequencies, while the target signal consists usually of higher frequency components. This also motivates dividing the spectrum into sub-bands and only using the lowest frequency sub-band centroid, because the higher sub-bands would not be affected by the wind anyway, so trying to detect it there would also be more difficult. As most of the wind noise is present in very low frequencies, the sub-band for detecting wind is set to start from $\mu = 0$ Hz and end in $\mu_1 = 3000$ Hz.

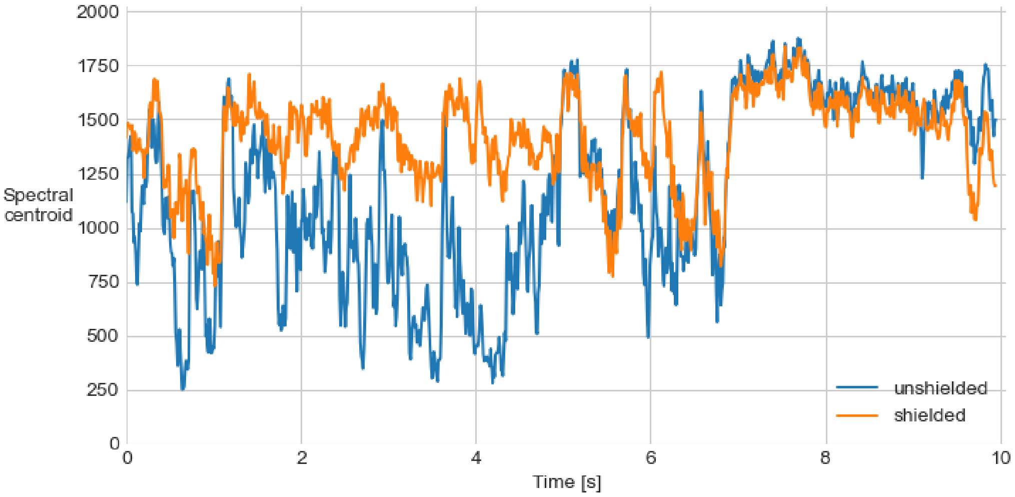


Figure 3.4. *Sub-band spectral centroids of two signals recorded at the same time with and without wind shielding.*

The low-frequency characteristics of wind noise, as also seen in Figure 3.4, force the spectral centroid to occur in a significantly lower frequence in the presence of wind noise. This property can be used to detect wind noise. [6]

3.3.4 Approach for multiple microphones

While the previously addressed features are suitable for detecting wind noise with information from only a single microphone, many contemporary devices are equipped with multiple microphones and this gives an opportunity to use different methods for detecting wind noise. This approach is important in devices with little computing capacity such as hearing-aid devices [39]. The methods for using multiple microphones is based on comparing simultaneously recorded signals from different microphones and their similarity. The similarity can be assessed using signal coherence.

Definition 3.6. The magnitude squared coherence (MSC) C_{12} between signals $x_1(k)$ and

$x_2(k)$ can be computed as

$$C_{12}(f) = \frac{|\Phi_{12}(f)|^2}{\Phi_{11}(f) \cdot \Phi_{22}(f)}, \tag{3.11}$$

where $\Phi_{12}(f), \Phi_{11}(f)$ and $\Phi_{22}(f)$ are the auto- and cross power spectral densities (PSD) of signals. The PSD values describe the distribution of power as a function of frequency f and are approximated using the Welch method [40]. By the Cauchy-Schwartz inequality, the values of MSC are between $0 \leq C_{12}(f) \leq 1$. [41]

MSC thus measures how well the power distributions of the signals match in different frequencies. If the coherence is close to 1, the result can be interpreted as the signals having a strong relationship between each other and similarly if the result is close to 0, the signals have no relationship at all. Ideally in the process of recording sound from a single source, the value of coherence is 1, but in real situations the coherence is lowered by the distance between microphones and the presence of noise sources. [42]

While the sound field produced by a single sound source such as a person speaking can be considered coherent, the sound field produced by wind noise is incoherent. That is due to the wind noise being generated in the turbulences that occur very near the device and thus different microphones will sense the turbulences differently [6]. In the past, several models have been created to represent the coherence in such cases, such as the Corcos model [43] that has also been shown to approximate the effects in recordings fairly well [44]. The Corcos model predicts exponential decay in coherence with growing frequencies which means that coherence will approach zero everywhere except at low frequencies [42]. In some approaches the coherence is also assumed to be fully approaching zero when wind is affecting the recording [7].

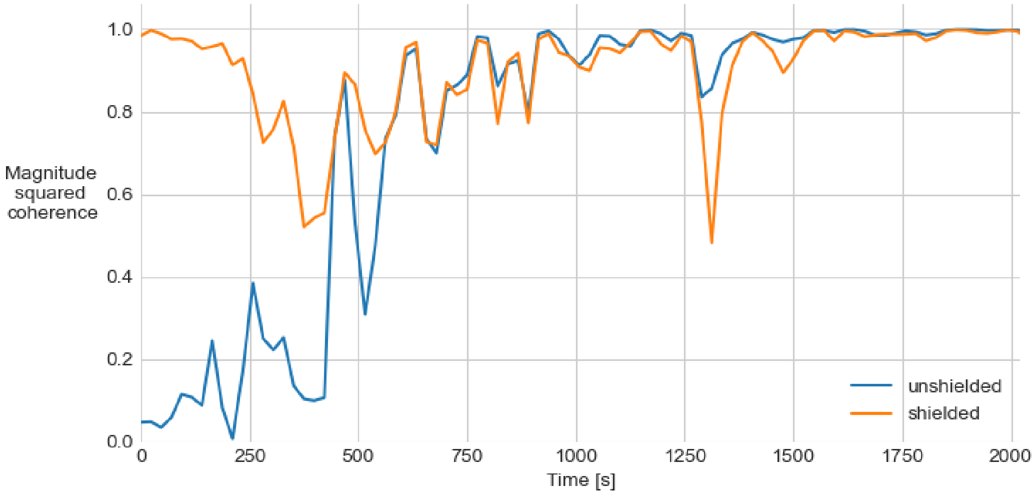


Figure 3.5. *Coherence between two microphones* 2 *cm apart and with and without wind noise present*

When recording audio from a coherent sound source with incoherent wind noise present using multiple microphones, it can be expected that the coherence between the two microphone signals has properties of both coherent and incoherent sound sources. In Figure 3.5 the coherence between two microphones with short distance $d = 2$ cm between them is plotted for the case with windshield and without shielding. It can be seen that coherence approaches 1 at frequencies above 800 Hz but in lower frequencies the value of the windy case is closer to 0. The behavior can be explained with the low frequency characteristics of wind noise and thus the wind noise lowers the coherence in the low frequencies but with higher frequencies it has much smaller effect. This implies that high coherence in the low frequencies indicates the presence of wind noise.

A good way of quantifying the coherence calculated for the lowest frequency bins of a frame is to take the mean

$$\overline{C_{12}}(\lambda) = \frac{\sum_{\mu=1}^{\mu_{\text{max}}} C_{12,\lambda}(\mu)}{\mu_{\text{max}}}, \tag{3.12}$$

where μ_{max} denotes the index of the highest frequency bin taken into calculation. When the mean coherence is close to 1, it is inferred that wind is not affecting the recording and on the other hand a mean coherence close to 0 suggests the presence of wind noise.

3.4 Approximating absolute wind noise in recordings

The measurements in this work are done simultaneously with a device that is shielded from wind and a device that is not shielded. As described earlier in Equation (3.1), this gives an opportunity to see the quantative differences in a signal with and without wind noise. Obviously there are always differences in the dynamic responses of different microphones of different devices and the two recording devices will also be slightly differently aligned towards the target sound source, so a simple subtraction as suggested by Equation (3.1) will not be free of those systematic errors described.

This issue will be tackled with calibrating the recording setup with measurements from time periods that did not have wind present while recording. The calibration is done in the frequency domain and with calibration coefficients $A(\mu)$ we can represent the two signals in the frequency domain as

$$\begin{aligned} S_1(\lambda,\mu) &= T(\lambda,\mu) \\ A(\mu)S_2(\lambda,\mu) &= T(\lambda,\mu) + N(\lambda,\mu), \end{aligned} \tag{3.13}$$

where $T(\lambda,\mu)$ and $N(\lambda,\mu)$ are the frequency domain representations of the target signal including non-wind background noises and the wind noise similarly to Equation (3.1).

For the calculation of the calibration coefficients $A(\mu)$, a subset of the recording data is created from periods where according to the anemometer measurements the wind speed is lower than 1 m/s. In this subset the wind noise is considered negligible and thus it can be assumed that the magnitude difference in X_1 and X_2 is due to the systematic bias considered earlier. A mean is calculated from all of the frames for each frequency bin and from this the calibration coefficients for each bin are calculated as

$$A(\mu) = \frac{\overline{|S_{1,\mu}(\lambda)|}}{\overline{|S_{2,\mu}(\lambda)|}}. \tag{3.14}$$

In this calculation the frequency spectra of signals X_1 and X_2 are divided into $1/3$-octave frequency bands [45] and the used DFT frame size is 2048. The used frequency range is $200 - 8000$ Hz. The calibration subset consisted of approximately 30 minutes of data. With the calibration coefficients calculated, the approximation of pure wind noise present in each frequency band and each frame can be calculated as

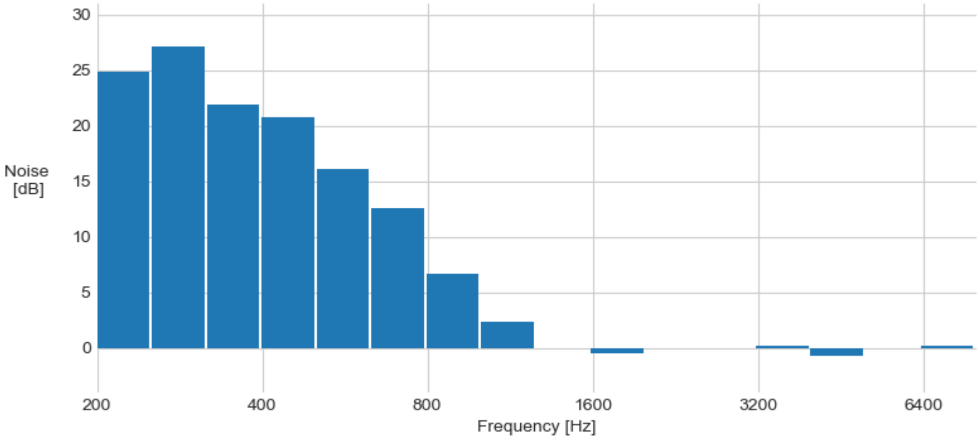


Figure 3.6. *Level difference between a windy recording and a recording shielded from wind over frequency bins with 1/3-octave bands*

$$N(\lambda, \mu) = A(\mu)S_2(\lambda, \mu) - S_1(\lambda, \mu) \tag{3.15}$$

and the matrix $N(\lambda, \mu)$ consists of level differences between the two signals. An example result of this approach is shown in Figure 3.6, where the level of wind noise present during the same example snippet used earlier in this chapter is visualized. Eight consecutive frames from the wind gust seen in Figure 3.1 are taken and averaged for the visualization. As it can be seen, large differences occur in the lower frequency bands below 800 Hz, which is a clear sign of low-frequency wind noise present in the recording. The fluctuating differences in the higher frequency bands suggest that despite the calibration, there will be differences of a few decibels between the levels even when wind is not present, as is seen in Figure 3.7. The visible effect of large differences in low frequencies is however significant compared to those fluctuations.

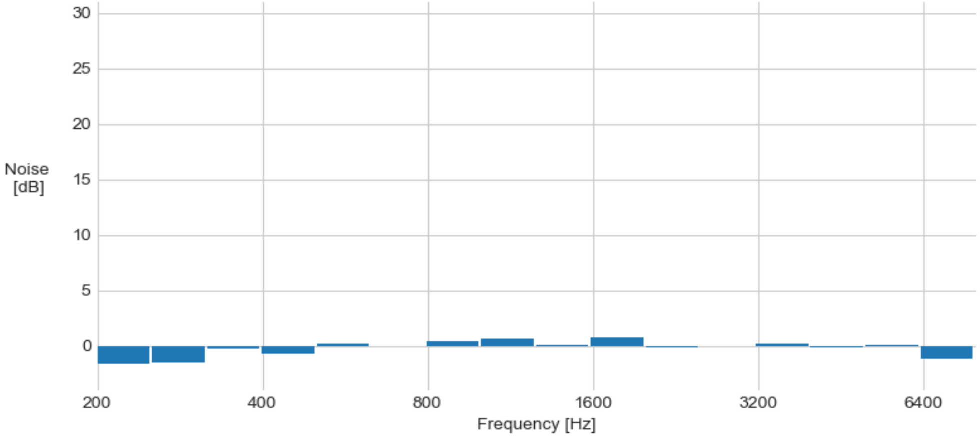


Figure 3.7. *Level difference between a windy recording and a recording shielded from wind over frequency bins with 1/3-octave bands in calm wind*

The noise differences in different frequency bins during a frame of audio can be quantified by summing the differences through the j considered bins

$$N_{\text{tot}}(\lambda) = \sum_{\mu=1}^{j} N(\lambda, \mu). \tag{3.16}$$

The higher the value of the total difference N_{tot} is, the more wind noise is affecting the recording.

4. MEASUREMENTS

The measurements and data for this work were collected using a specific laboratory setting that was designed especially for collecting wind noise measurements. The implementing of the laboratory setup was done by the author as a summer project previous to this research project. The key principle was for the setup to be fit for different purposes and to be movable in order to record outdoors in presence of natural wind instead of indoor measurement in wind tunnels or using an ordinary fan. Indeed, the whole setup was build on a portable platform and it was also possible to put the plane in a cargo bike to make measurement trips in different windy locations.

4.1 The measurement setup

The used measurement setup was specifically designed to be used in different applications of measuring wind noise in recordings. It allows recording of multiple devices in presence of wind while constantly measuring the wind present. The wind measurement is done by a Vaisala WMT700 ultrasound anemometer, and in this measurement setting the audio recording devices are two mobile phone shaped prototypes with 8 microphones in different positions as described by Figure 4.1. One of the prototypes was put inside a Rode Blimp MkII windshield to be completely protected from the effects of the wind noise and the other was placed right below the windshield. A regular Bluetooth speaker was mounted 40 cm away from the recording devices. The devices and the speakers were fixed to a Mark Roberts Motion Control SFH-30 camera head, which allowed the devices to be rotated in order to get wind from different directions. The measurement setup in action is pictured in Figure 4.2 and the components are highlighted in Figure 4.3.

The measurement software was implemented using Python. The software took care of the recording and playback of audio, rotating of the camera head and the gathering of the anemometer data. The python library `pyaudio` was used in handling the recording and playback of audio.

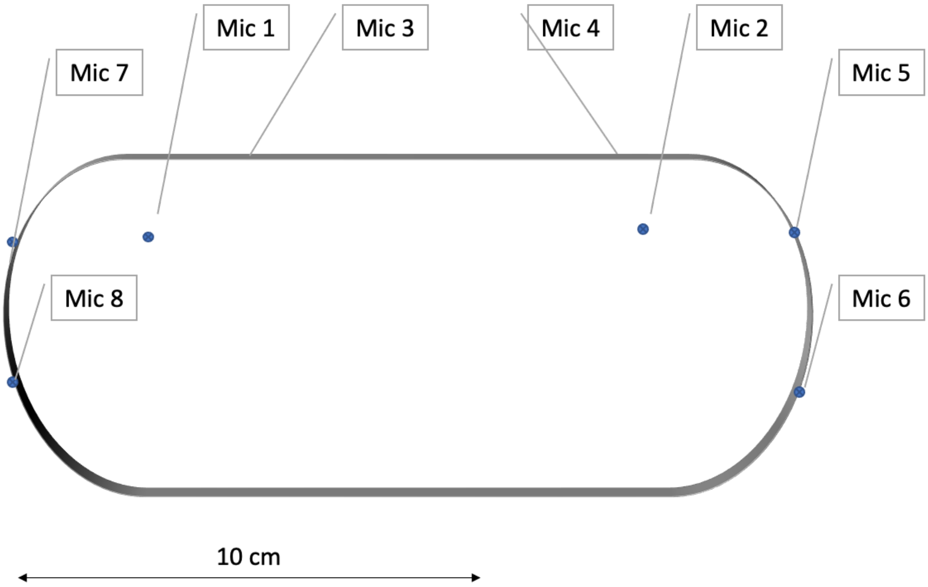


Figure 4.1. *An illustration of the recording device used with its microphone locations described*

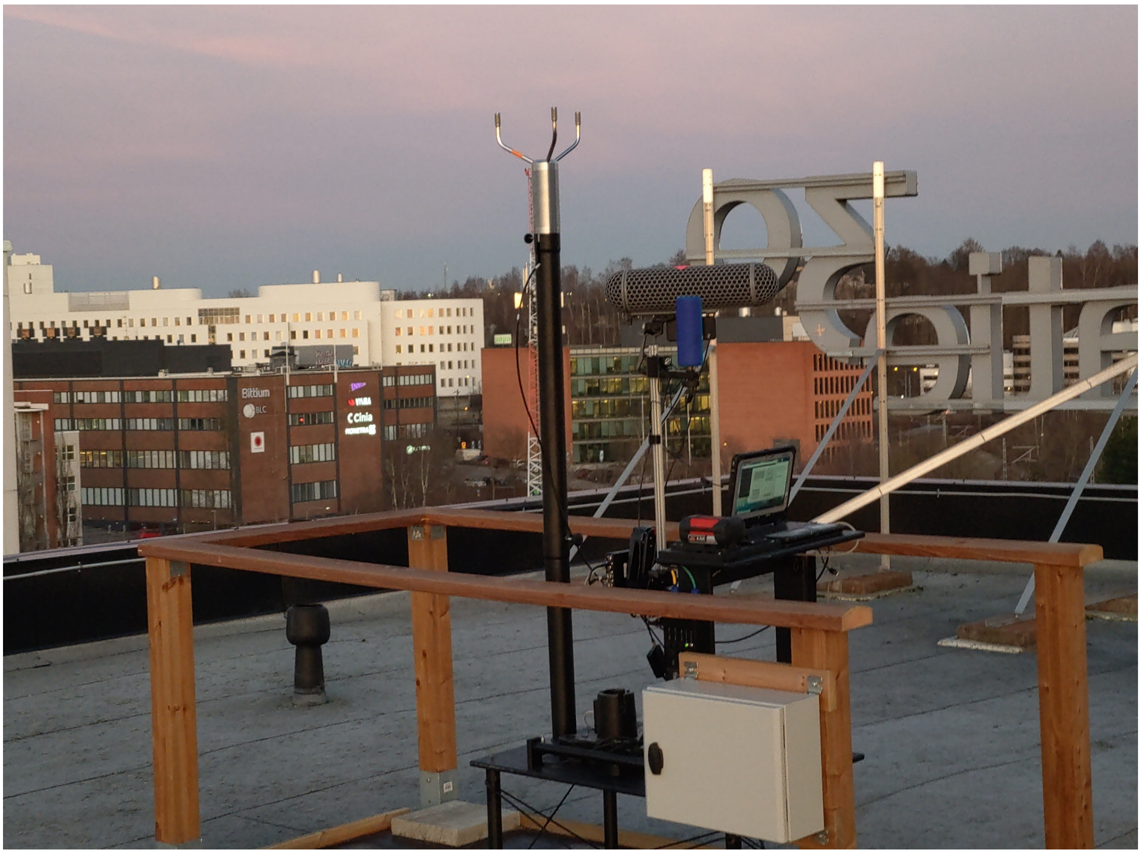


Figure 4.2. *The measurement setup in action at the office rooftop.*

4.2 The measurement data

4.2.1 General

For the purpose of this study measurements in different locations were made. Each measurement lasted for several hours and produced 16 channels of audio, data from wind

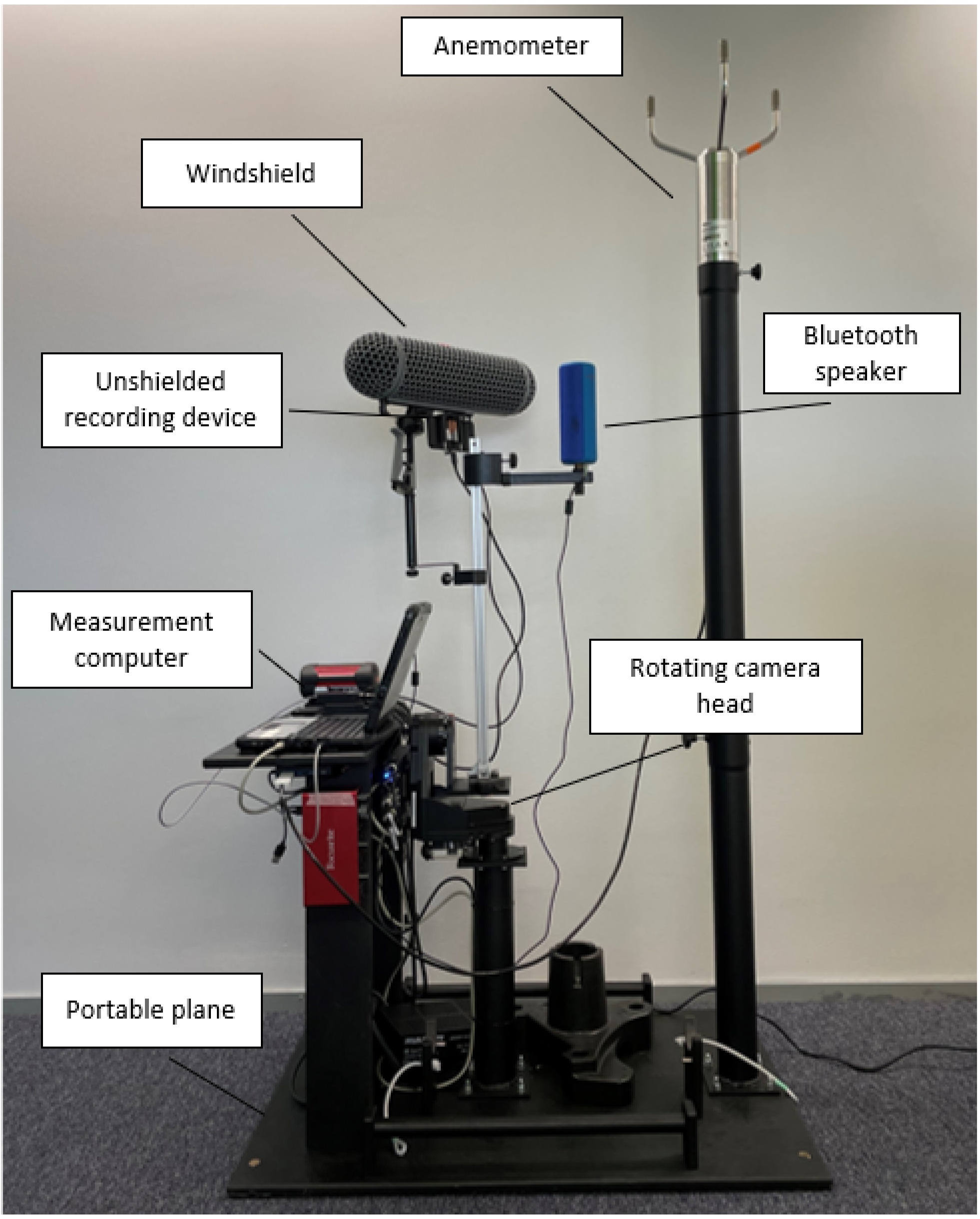


Figure 4.3. *The components of the measurement setup*

speed and direction and data about the orientation of the camera head. The recorded audio was sampled at 48000 Hz and the measurement rate of the anemometer is 4 Hz. The playback signal played from the speaker was a 7 minute sample containing speech and music and it was looped throughout the measurement. The speech consisted of sentences gathered from the SPEECON database [46] and the music was a sample of the song Rosanna by Toto. This kind of playback was used in order to get data from recording of different types of audio. The extensive amount and the rotating of the devices of data ensured that a variety of different wind conditions for every microphone was obtained.

In order to get all the data synchronized in time, a pseudo-random signal was played during the start of the measurement. During the processing of the data every audio file was then cross-correlated with the synchronization signal and because the time stamp of the start of the signal was known, all the audio files were cut from the beginning to have the signal start from the same timestamp [47]. Also the wind direction data was processed after the measurements. The position of the microphones in the devices was known and the angle of wind direction was corrected to be relative to each microphone instead of just the device.

4.2.2 Clock issues during the measurements

In the initial phase of the measurements it was noticed that the cross-correlation of the pseudo-random signal did not perform as expected and the correlation did not have a single spike as theoretically would have been expected [48]. The issue only came up with one of the recording devices and it was hypothesized that the reason could be clock inaccuracy in the devices. The issue was investigated further by conducting a measurement where the same synchronization signal was played multiple times during a course of 15 minutes and from both devices it was checked if the amount of samples recorded between the synchronizations would be different between devices. Different combinations of prototypes and sound cards specifically built for them were tried.

As from Figure 4.4 can be seen, the difference between the samples recorded at the synchronization points did not stay constant in most cases that involved the sound card referred as Box1. The linear growth in the difference would suggest that the devices were recording with different intrinsic sample rates and that was deduced to be the case. The problematic Box1 was a device that was wanted to be applied here because it had a software that handled possible microphone clipping differently to the other sound cards. The issue was solved by modifying the setup a bit and using the sound card Box1 with a slight update for it to be able to handle two input devices simultaneously. That procedure ensured that both recordings had the same internal clock.

4.3 Creating training and test datasets

The purpose for the measured data is to use it in performing machine learning tasks described in Chapter 2. Thus the measurements were processed by performing feature extraction and labeling for the data. For the feature-label pairs $\{X_i, Y_i\}$ the used frame size was 2048 samples with overlap of 512 samples.

The labeling was done using the absolute wind approximation method described in Section 3.4. For each frame from both of the input devices in the measurement system, the total noise difference N_{tot} was calculated according to Equation (3.16) using the 6 lowest frequency bins, that is the first two octaves from 200 Hz to 800 Hz. After that each frame was assigned with a label according to the N_{tot}. In this approach, a binary classification was considered with labels referring to case considered without wind and a case with

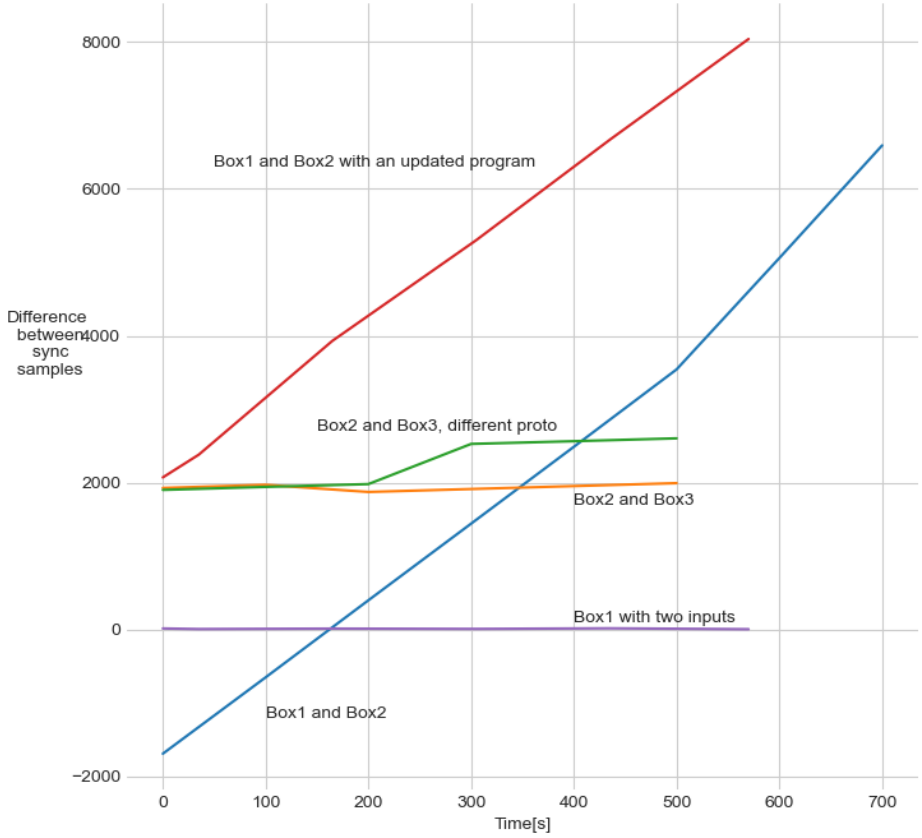


Figure 4.4. *Overview of the clock tests with different combinations of sound cards*

wind. A threshold was set to $N_{\text{tot}} = 10$ and thus the labeling was done as

$$Y_i = \begin{cases} 0, & N_{\text{tot}} < 10 \\ 1, & N_{\text{tot}} \geq 10, \end{cases} \tag{4.1}$$

where $Y_i = 0$ refers to a frame with no wind and $Y_i = 1$ to a case with wind.

Assigning the labels is a big part in training a machine learning model and a lot of the performance of the model depends on the quality of the labels [49], and thus assigning the labels and choosing the threshold for wind and no wind is an important step. The chosen threshold of $N_{\text{tot}} = 10$ refers to a case where each of the frequency bins has $N_\mu = 1.67$ dB of wind noise on average. Realistically the amount per frequency bin is higher as the performed calibration suppresses slightly too much and thus pushes the cases with no wind to have a negative noise difference. Regardless, the chosen threshold assigns windy labels for frames also with relatively little wind noise, which was desired. Detecting high wind noises is relatively easy, but in those cases the noise has corrupted the original signal relatively badly and thus not much can be done in order to improve the signal [8]. For this reason a model that attempts to detect low wind noise was desired and the choice of the threshold was done accordingly.

The features used for the explanatory data are described in Sections 3.3.1, 3.3.2, 3.3.3 and 3.3.4, and thus every data instance $X_i = (\text{ZCR}_i, E_{\text{RMS}i}, \text{SSC}_i, \overline{C_{xy_i}})$. Zero-crossing rates, root mean square energies and sub-band spectral centroids were calculated according to equations (3.2), (3.9), (3.10) respectively. For calculating the coherence values each microphone was assigned a pair with which the coherences were calculated. The pairs used were microphones $(1,2)$,$(3,4)$,$(5,6)$ and $(7,8)$, labelled as described in Figure 4.1

For each pair the coherences were calculated using the Equation (3.11) and then averaged using Equation (3.12) assigning $\mu_{\text{max}} = 800$ Hz as was done in the labeling phase. The prior knowledge about the features suggests that they are all relevant in the context of wind detection and thus eligible to be included in the model [10]. Feature extraction and labeling was done separately for the data from all of the input microphones of the unprotected prototype. This leads to a classification model for a single microphone wind detection.

Class	Number of frames
Total frames	5173016
$Y_i = 0$	2625054
$Y_i = 1$	2547962

Table 4.1. *Distribution of the training data instances between labels*

For creating a well-generalized classification model, it is important that the training data used is also as general as possible and contains a lot of different examples [49]. In this case this was ensured by sampling the training dataset evenly from all the individual recordings performed, from different wind conditions and using all of the eight microphones evenly in order to ensure that the same wind is captured from all the directions. A total of 5173016 of frames were assigned to the training set, which corresponds to 15 hours of audio. The training set was labeled and the distribution between labels is described in Table 4.1. As can be seen the training set is evenly distributed between the two classes.

The distributions of classes for each feature are shown as histograms in Figure 4.5, and as can be seen, in all of the features the distributions are mostly separable with some overlapping. This kind of situation leads to a training set that is not linearly separable and thus slightly less likely to overfit [10].

The test data was picked and processed similarly to the training data using data from all

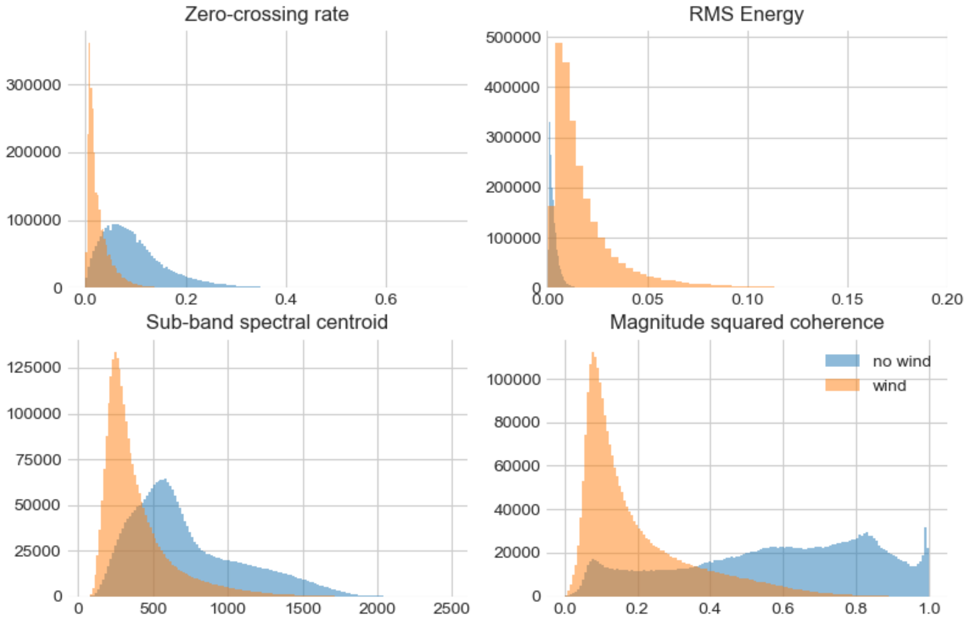


Figure 4.5. *Distributions of training data feature values with each label*

of the individual recordings and all of the microphones. The test data was also a balanced set of frames from both classes, as described in Table 4.2, and consisted of 56 minutes of audio.

Class	Number of frames
Total frames	314256
$Y_i = 0$	170258
$Y_i = 1$	143998

Table 4.2. *Distribution of the test data instances between labels*

5. FINDINGS OF THE STUDY

5.1 Wind noise classification

One of the purposes in this study is to be able to explore signal properties that are characteristic for wind noise and to use them in the task of detecting the presence of wind noise in recordings. The detecting is performed using the two different classifiers described in Section 2.1 and the performance of those different models used for classification is compared and discussed.

5.1.1 Exploring the problem

The two models that are to be used in this classification task have a different way of performing the assignment of the label. Logistic regression is a linear model that can be understood as increasing or decreasing the odds of a positive label assignment as the value of the feature increases [12] and a GMM tries to fit the best possible multivariate Gaussian for the data and use that for the classification decision [18]. Considering the distribution of the training data described in Figure 4.5, it can be argued that both approaches would be applicable and relevant with this data. The linear approach seems suitable as each of the features have a clear single threshold between either class being the most probable one, while most of the distributions are more or less bell-shaped which would suggest a Gaussian distribution being a good way to model the data.

The purpose is to train both models with identical training data and then compare the performance of the classifiers over the same test set, with both of the used datasets described in Section 4.3. Out of the performance indicators described in Section 2.1.3, in the context of wind detection the most interesting indicator is recall, that describes the relative amount of detected frames from the frames with wind present. Thus in the tradeoff between precision and recall it is of interest to get more recall and then evaluate what the precision is with high recall.

The training of both of the models was performed using a Python library `sklearn` that uses the L-BFGS algorithm for maximizing the likelihood of the logistic regression model and the EM algorithm while fitting the Gaussian mixture model. For the GMM initial values for means and covariances used were obtained using the common method of performing a K-means clustering [17] as the initial step. The covariance matrices for the model were assigned to be diagonal and thus the different features are considered to be independent [18].

The training of the logistic regression classifier took 36 seconds and the linear model $\theta = \theta_0 + \sum_{n=1}^{N} \theta_n X_n$ fit to the data is shown in Table 5.1. As seen in Equation (2.3), the linear model corresponds to the log-odds of the label being $Y_i = 1$ and thus each parameter of the model can be assessed in relation to the odds. Generally, for the features with negative coefficients in the model, the odds decrease as the feature value increases and vice versa. Comparing the values in Table 5.1 and the histograms in Figure 4.5, the model seems to agree with the data in this sense.

θ_0	θ_{ZCR}	θ_{RMS}	θ_{SSC}	θ_{MSC}
3.3458	-0.0008	-27.6467	8.2299	-4.6727

Table 5.1. *The trained linear model of the logistic regression classifier*

The training of the GMM took 23.45 seconds to reach convergence. The weights for the different gaussians were $\tau = (0.57778, 0.4222)$ for classes $Y_i = 0$ and $Y_i = 1$ respectively and the means and covariances for each components are collected in Table 5.2. Considering the weights of classes and comparing them to the initial distribution of the training data described in Table 4.1, it can be assumed that the training process has classified some of the windy frames as non-windy. This is expected as the Gaussian that are fit for both classes are expected to overlap, as shown in Figure 4.5. Further discussion about the parameters of both classifiers trained here will occur while analyzing the performance of the classifiers.

Component	ZCR	RMS	SSC	MSC
$[\mu_0, \Sigma_0]$	$[0.097, 0.004]$	$[0.004, 7.6 \cdot 10^{-6}]$	$[720, 131900]$	$[0.566, 0.062]$
$[\mu_1, \Sigma_1]$	$[0.018, 0.0001]$	$[0.021, 0.0004]$	$[334, 21630]$	$[0.150, 0.009]$

Table 5.2. *The trained parameters for the GMM classifier*

5.1.2 Performance of the classifiers

The performance of the classifiers was evaluated by using the test dataset described in Section 4.3 and the performance metrics presented in Section 2.1.3. For the logistic regression classifier the probability $\pi(X_i)$ for each test data instance was calculated as described in Equation (2.5) and the predicted label was decided from that. Confusion matrix showing the true and predicted labels for each instance of the test data is described in Table 5.3 and the precision-recall curve describing characteristics of the classifier using different decision thresholds is shown in Figure 5.1. Accuracy score for the classifier is also shown in Table 5.3.

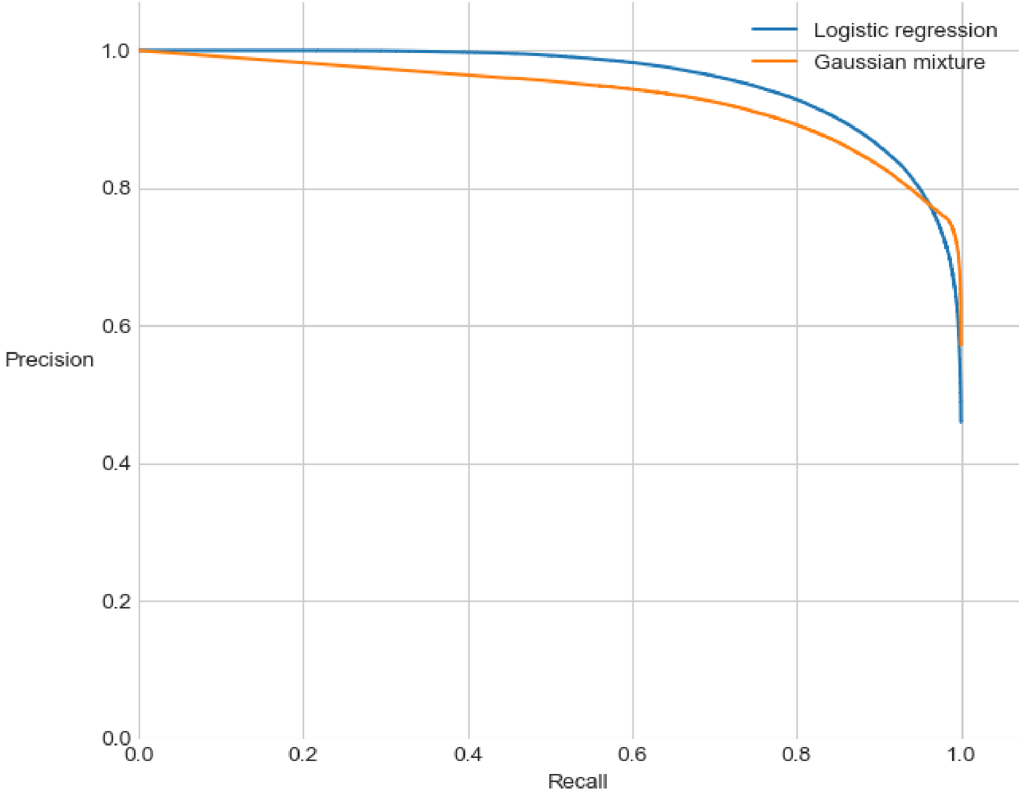


Figure 5.1. *Precision-recall curve for the classifiers*

	Predicted negative	Predicted positive
Actual negative	156254	14004
Actual positive	20890	123108
Accuracy	0.89	

Table 5.3. *Confusion matrix of the logistic regression classifier*

For the Gaussian mixture model classifier the probabilities for each instance belonging to either class were calculated as shown in Equation (2.15). The confusion matrix and the accuracy are shown in Table 5.4 and precision-recall curve is shown in Figure 5.1.

	Predicted negative	Predicted positive
Actual negative	157441	12817
Actual positive	36515	107483
Accuracy	0.84	

Table 5.4. *Confusion matrix of the Gaussian mixture model classifier*

5.1.3 Discussion about the classifiers

The performance of both classifiers with the test set was relatively good and satisfying, with the accuracy of both classifiers being well over 80%. As the labeling for positive cases was performed with a low threshold, it was to be expected that false negative predictions would occur near the threshold. This effect can also be seen in the overlap of each feature between classes shown in Figure 4.5. As seen from the confusion matrices, the performance of the Gaussian mixture model classifier suffers more heavily from the false negative predictions, as both classifiers perform almost identically with the negative test data instances, but differences occur with the positive instances.

As seen from the precision-recall curve of the logistic regression classifier, it performs for a large amount of the test data with near-perfect precision and the precision only starts falling after a decision threshold with recall around 0.6. Thus 60% of the positive test instances can be detected correctly with a negligible amount of false positives. With maximum recall, the precision falls under 0.5 and thus using a decision threshold that allows the classifier to predict all of the positive test instances correctly, over half of the predicted instances would be false positives. [21] Using the GMM classifier, the precision starts to decrease already with low recall values, although with high recall values the decrease in precision is much slower than in the case of logistic regression. With maximum recall the the precision of the GMM classifier is larger than that of the logistic regression classifier.

The difference in the behaviour of both classifiers is explained with the differences of the models used. Logistic regression uses a linear model and thus the decision for classifying is one-dimensional for each feature. This kind of behaviour leads to near-perfect precision in classifying data instances with feature values that had little overlap between classes in training data. This is a characteristic that could be useful in detecting solely higher wind instead of trying to find all of the wind, as was done in this work. On the other hand the Gaussian distributions used by the GMM classifier lead to some probability of either class existing for even the most extreme feature values and thus the precision is not perfect for even the highest winds. This Gaussian characteristic leads to better performance closer to the threshold with the overlapping feature values and has its advantages in those kind of approaches. Overall, the feature distributions described in Figure 4.5 implied that both linear approach and fitting Gaussian models are viable options in modelling the data. The good performance of the classifiers supports this observation.

Another thing that was studied about the classification was the importance of each feature used. Feature importance was assessed using the logistic regression classifier and the likelihood ratio approach described in Section 2.1.1. It was used to find, whether all of the four features were necessary for the performance of the classifier. The comparison was performed calculating the log likelihood of the full model described in Table 5.1 and then fitting comparison models with omitting one of the features at a time. Log likelihoods of the comparison models were compared to the full model value and the decreases of likelihood caused by leaving each feature out of the model was calculated. The values are shown in Figure 5.2.

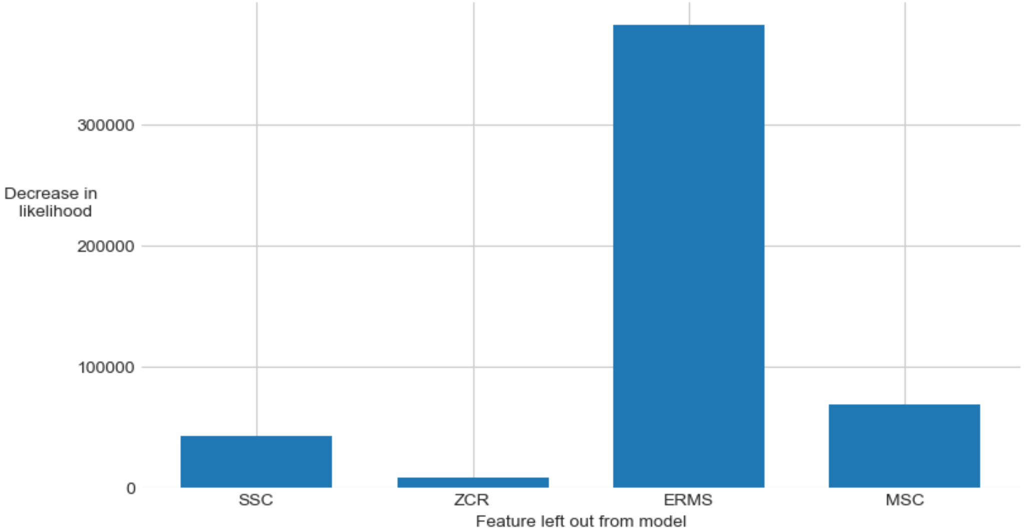


Figure 5.2. *Decrease of likelihood for logistic regression model when a feature is omitted*

The amount of decrease in model likelihood when omitting a feature can be interpreted as the importance of the feature for the model. As can be seen, RMS energy is clearly the most important feature in the model and zero-crossing rate is the least important. However, all of the features are statistically significant, as by using Equation (2.11), the likelihood ratio test value for comparing the full model and the model without the least important feature zero-crossing rate is $G = -2(-8535)$. P-value for this can be calculated as $P(\chi^2(1) > G)$ and the value is negligible. The degrees of freedom $d = 1$ in this test is the difference in number of features of the compared models [12].

The fact that each of the features has significant importance is rather expected. From a statistical point of view, the number of training data instances is very large compared to the number of features, and thus the model is not in danger of having too many features for too little amount of training data. From a signal processing point of view, the importance of all features can be explained by considering that all of the features describe a different characteristic of the signal, as discussed in Section 3.3. RMS energy describes the amplitude of the waveform, MSC compares the similarity of signals between different microphones and SSC and ZCR are related to the frequency spectrum of the signal. Wind noise affects all of the different characteristics and thus for a successful model it is important to have information about all of them. The smallest importance of ZCR can also be explained with similar arguments, as it describes the spectral characteristics along with SSC, but it is a less powerful explanator than SSC [6]. The substantial importance of

RMS energy is likely slightly biased in this measurement setup, as the target signal was played with a constant volume and thus an increase in volume of the recording is very likely caused by wind noise. In realistic audio recording situations the volume of the target signal is not constant and thus it can be expected that the classification performance of the RMS energy would be slightly lower.

All of the features used were relatively well known and commonly used features used in wind noise applications [6, 7] and thus it was rather expected that they would be successful in predicting the presence of the wind noise. The good performance in the classification can also be interpreted as meaning that the features and the labels used describe the same phenomenon of the data [17]. As the labels were created by using a method motivated by the measurement equipment, the successful classification indicates that the absolute wind noise approximation method suggested in Section 3.4 was reasonable and valid and gives at least a similar level of information about wind noise as four commonly used signal features.

The classifiers presented performed reasonably well with this measurement setup in question. However it is necessary to note that the microphones and their implementation methods used in the recording device of this work can be different to microphones of other devices, which can alter the recorded sound substantially. Thus it is necessary to perform similar measurements and analysis with other devices, such as commercial mobile phones, too, if the performance in wind detection is wanted to be generalized. With this kind of work performed it would then be possible to use this kind of classifiers for frame-by-frame wind detection in different kinds of applications.

5.2 Wind noise in multiple microphones

Having multiple microphones available on different sides of the recording device is a good asset in studying the sequential effects of wind noise in different sides of the device. The purpose is to find mainly qualitative information about the time dependence and evolution of wind noise and see if any patterns is possible to be found.

5.2.1 Describing the model

The time dependence of the appearance of wind noise in different microphones is investigated using a Hidden Markov model. The scenario to be modelled is one with wind entering the system from a relatively constant direction and the observations X_i for the model are the noise differences, calculated with Equation (3.16), for each of the three microphones that are considered. With this setup the hidden states Y_i will be all the combinations of individual microphones being affected by wind noise or not. The microphones considered are from different parts of the device and both in front of and behind the approaching wind as described in Figure 5.3. That is in order to investigate the vortices all over the device and to see if there are situations where only a part of the microphones are affected by wind. The different hidden states of the model are collected in Table 5.5.

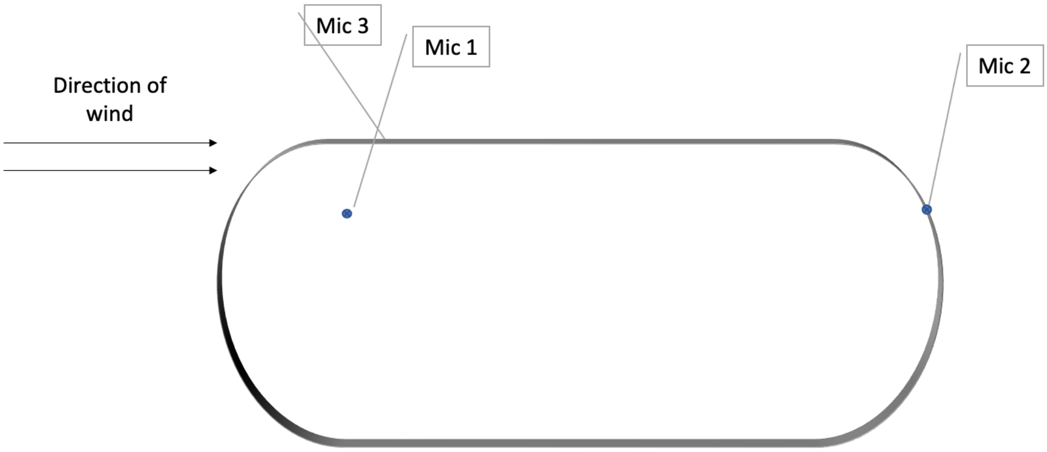


Figure 5.3. *Illustration of the modelled case for wind in different microphones*

State	Windy mics
W_0	None
W_1	$\{1\}$
W_2	$\{2\}$
W_3	$\{3\}$
W_4	$\{1,2\}$
W_5	$\{1,3\}$
W_6	$\{2,3\}$
W_7	$\{1,2,3\}$

Table 5.5. *The hidden states of the created HMM*

The purpose of utilizing of the Hidden Markov model in this approach is to verify the hypothesis of time dependency occurring in the vortices producing the wind noise reaching each microphone. Other hypothesis to consider is to check if there are situations where the device offers protection from wind to some of the microphones. These hypotheses can be investigated using the prior probabilities τ and the transition matrix A of the trained model, as τ contains information about the total occurrences of each state in the training data and A describes if some transitions are more probable than others [17]. Given that the direction of the wind is constant, the most probable state transitions can give insight in how the wind reaches each microphone, if some time dependence is occurring.

Also, from the probabilities of each state it is possible to see, if some microphone is less affected by the wind than the others.

The model was trained using a sequence of recording with relatively constant wind direction. Data was taken from three microphones around the recording device, as described in Figure 5.3, and data is divided to frames as described in Section 4.3. Noise differences between shielded and unshielded devices for each microphone used are calculated with Equation (3.16). The occurrence of wind for each frame is decided from the noise differences with Equation (4.1) and the sequence of hidden states Y for the training data is then labeled by combining the information from individual microphones to form states as described in Table 5.5. The observations X consist of noise differences for all of the microphones and thus $X_i = \{N_{\text{tot},i1}, N_{\text{tot},i2}, N_{\text{tot},i3}\}$, where the indices $i1$, $i2$ and $i3$ refer to the ith data instance and the number of microphone used. The duration of the recording instance used is 10 minutes and the average direction of wind during the recording is $-45°$ relative to microphone 1.

As the hidden states are known in the training data, parameters of the Hidden Markov model are calculated using the supervised learning method described in Section 2.2 with Equations (2.39), (2.40) and (2.41). The prior probabilities are described in 5.6 and the emissions for each state and microphone are collected in Table 5.7. The transition matrix is described in Figure 5.5.

State	W_0	W_1	W_2	W_3	W_4	W_5	W_6	W_7
τ_j	0.019	0.065	0.018	0.002	0.220	0.039	0.011	0.627

Table 5.6. *Prior probabilities of the HMM*

State	$[m_1, \Sigma_1]$	$[m_2, \Sigma_2]$	$[m_3, \Sigma_3]$
W_0	$[4.147, 23.366]$	$[-3.733, 44.253]$	$[-8.997, 53.781]$
W_1	$[54.211, 2161.958]$	$[-4.146, 64.076]$	$[-3.969, 62.707]$
W_2	$[3.686, 29.708]$	$[55.602, 1498.894]$	$[-8.022, 73.647]$
W_3	$[4.534, 21.387]$	$[-2.321, 44.455]$	$[31.397, 457.392]$
W_4	$[75.794, 2595.715]$	$[78.957, 2456.081]$	$[-3.089, 66.981]$
W_5	$98.618, 3873.436]$	$[-2.366, 80.998]$	$[33.698, 520.167]$
W_6	$[4.262, 23.45]$	$[71.136, 1747.739]$	$[45.774, 1106.274]$
W_7	$[124.53, 4113.57]$	$[106.785, 2892.036]$	$[66.526, 1848.988]$

Table 5.7. *Emissions for each HMM state*

The model was verified by solving the HMM decoding problem of the model for the given training sequence of observations. The sequence $\hat{Y}$ calculated with the Viterbi algorithm implementation of the Python library `hmmlearn` was compared to the manually labelled training states Y and the accuracy score was 0.89. This confirms that the model parameters predict the characteristics of the observations well.

5.2.2 Discussion

The distribution of states in the model for the given dataset is described in Figure 5.4. As can be seen from the distribution, only a few states were frequent with some states being close to negligible with occurrence. This is a thing that needs to be taken into consideration when interpreting the transition matrix, described in Figure5.5. The modelled situation had constant wind present with relatively few windless moments, and it can also be seen in the state distribution. The number of frames in the totally windless state was very small and on the other hand the state with wind in all of the microphones was the most frequent by far. The latter observation also indicates that in this kind of wind situation, wind noise occurs in microphones all over the device. This implies that the device body does not protect from wind in most cases, which indicates that the vortices created on the device will move along the device body all over it.

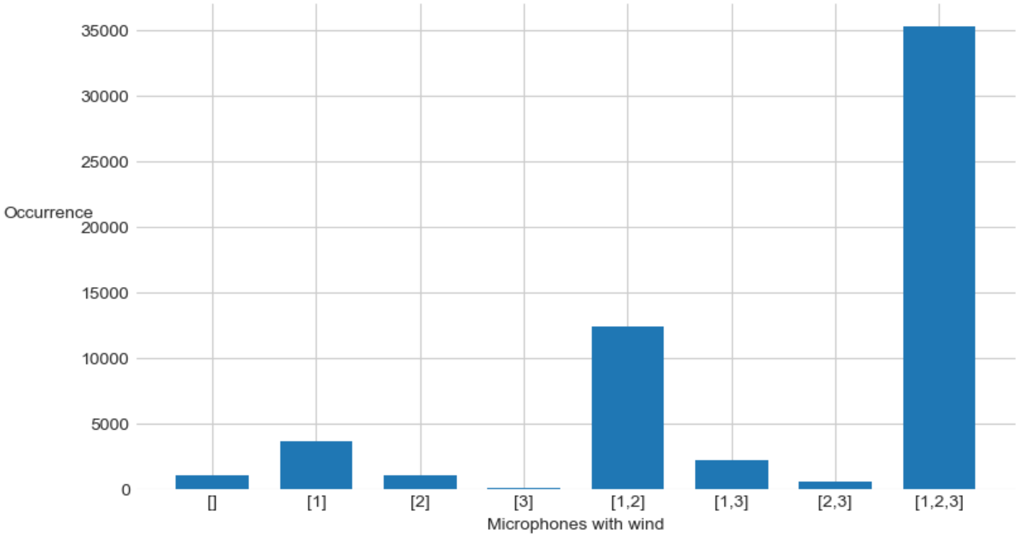
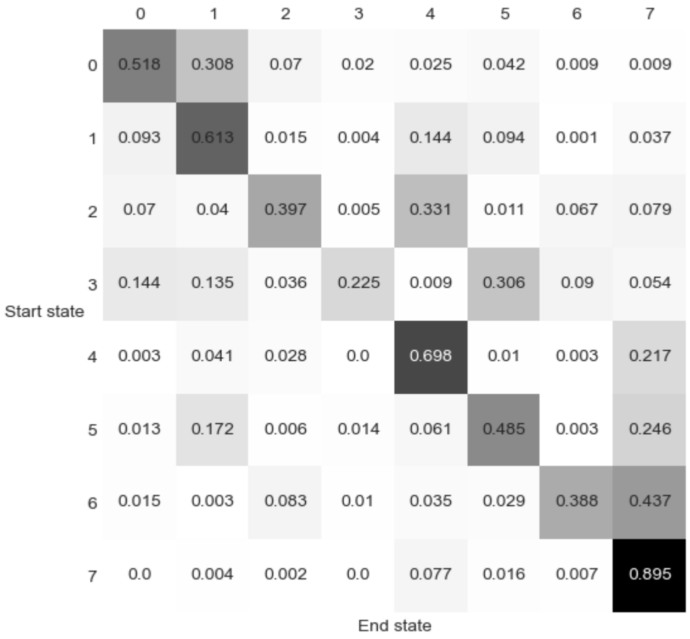


Figure 5.4. *The distribution of states in the trained Hidden Markov Model*

As the state with wind in microphones 1 and 2 is so frequent, it indicates that despite it not being very common, there still are instances, where the wind is not producing noise on the opposite side to the arriving wind. As seen in the transition matrix, $A_{44} = 0.698$ and thus the system has a relatively high probability to stay on this state. It indicates that the state W_4 is not only a transition state to the state with all microphones having noise W_7, but that there are longer sequences in the data, where one of the microphones is non-windy. The importance of the state W_4 as a transition state is also evident, as it is a common route to transition to state W_7 and also state W_1 has a high probability to be followed by W_4. Transition A_{74} is also the most probable one to transition out from the state W_7 and thus the microphone behind the device is the most likely to become windless.



	0	1	2	3	4	5	6	7
0	0.518	0.308	0.07	0.02	0.025	0.042	0.009	0.009
1	0.093	0.613	0.015	0.004	0.144	0.094	0.001	0.037
2	0.07	0.04	0.397	0.005	0.331	0.011	0.067	0.079
3	0.144	0.135	0.036	0.225	0.009	0.306	0.09	0.054
4	0.003	0.041	0.028	0.0	0.698	0.01	0.003	0.217
5	0.013	0.172	0.006	0.014	0.061	0.485	0.003	0.246
6	0.015	0.003	0.083	0.01	0.035	0.029	0.388	0.437
7	0.0	0.004	0.002	0.0	0.077	0.016	0.007	0.895

Figure 5.5. *The transition matrix of the trained Hidden Markov Model*

The most likely state to follow the windless state is W_1, which is expected as microphone 1 is the closest microphone to the wind. Considering all of the states that include a windy microphone 1, it can be seen from the state distribution that they are the four most frequent states. This implies that it is very likely that the microphone closest to the wind is affected by wind noise. The modelled data had very few instances without wind and thus not a lot of information can be acquired regarding wind entering or leaving the system and the states related to that process.

It is possible that more information about the system going from windless to having wind would have been achieved with different choice of training data. However the used sequence was the best one available from the set of measurements of this work, as the constant direction of wind was deemed more important for this consideration. More in-depth consideration would require more measurement recordings with constant wind direction but varying speed. It is also possible that using a shorter FFT step length in the data would have given more precise information. That is because with higher wind speeds the wind vortices could have travelled so fast that some of the states would not be caught with the used sample rate of data instances.

Altogether, the system had some occurrences for every state transition and it generally shows that the behaviour of wind is relatively chaotic around a body of mass such as the recording device here. It can however be stated, that in this kind of constant wind situation, a wind reduction system of a recorder cannot rely on wind being less present in some part of the device. There are some instances where the device body protects the microphone from the wind but in most situations wind occurs in every microphone. Also some time-dependencies were possible to be found, as the most common microphone to encounter wind first is the one closest to the wind. Also it is likely for the wind-induced vortices to travel along the surface of the device along with the wind direction.

6. CONCLUSION

This work aimed to investigate the behaviour and occurrence of wind noise in recorded audio signals by using the characteristics of wind noise signals. The analysis was performed by using multiple microphones in different parts of the recording device and by comparing the sound recorded simultaneously by two devices; one positioned in wind and one shielded from wind. This approach made it possible to get information about behaviour of wind noise in different parts of the device, related to the direction of arrival of the wind.

Detecting the occurrence of wind noise in recordings was performed using two classifier models, logistic regression and Gaussian mixture model. The explanatory data consisted of four signal features, that were zero-crossing rate, root mean square energy, sub-band spectral centroid and magnitude squared coherence between two microphones. The mathematical motivation for the models used was discussed and the characteristics of wind noise described by each used feature were detailed. Induction of wind noise in microphones was also considered. Values of each feature were compared between two recordings performed simultaneously in wind and shielded from it, and thus it was visualized, what kind of an effect wind noise has to the feature. Both classifiers performed well in detecting presence of wind noise and can thus be stated to be well suitable and useful in wind detection. The result can be generalized further by performing similar analysis with other types of recording devices.

The measurement equipment and the two comparable recording devices also gave motivation to a method of approximating the absolute amount of wind noise present in the recordings. The analysis was performed by comparing the level differences in frequency bands of shielded and unshielded recordings. Some calibration was applied to consider the sound level differences present between recording devices regardless of wind. This method was used to generate labels for the machine learning applications in this work and the good performance of the wind detectors indicate that this method too performed relatively well.

The behaviour of wind in different parts around the recording device was investigated using a Hidden Markov model. The model took into consideration wind with a constant direction of arrival and microphones in different parts of the device on different sides. The states used in the model were combinations of different microphones that had wind noise at each time step. The data that was modelled was relatively windy and that caused some states to be much more frequent than others. Still it was noticed that in windy conditions there is a probability that the recording device provides some wind protection for the microphone on the opposite side of the arriving wind. However it is not the most likely state and in most cases wind noise affects the microphones everywhere on the device. From the transition matrix of the Hidden Markov model it was also interpreted that it is likely for the wind-induced vortices to travel along the surface of the device aligned with the direction of wind. Modelling the process of wind arriving to a windless recording system or vice versa is possible using the same method, if large enough amounts of data with constant wind direction and varying wind speed are used.

REFERENCES

[1] Ventura, R., Mallet, V., Issarny, V., Raverdy, P.-G. and Rebhi, F. Evaluation and calibration of mobile phones for noise monitoring application. *The Journal of the Acoustical Society of America* 142.5 (2017), pp. 3084–3093.

[2] De Coensel, B., Botteldooren, D., De Muer, T., Berglund, B., Nilsson, M. E. and Lercher, P. A model for the perception of environmental sound based on notice-events. *The Journal of the Acoustical Society of America* 126.2 (2009), pp. 656–665.

[3] Zheng, Z. C. and Tan, B. K. Reynolds number effects on flow/acoustic mechanisms in spherical windscreens. *The Journal of the Acoustical Society of America* 113.1 (2003), pp. 161–166.

[4] Au, A., Blakeley, J. M., Dowell, R. C. and Rance, G. Wireless binaural hearing aid technology for telephone use and listening in wind noise. *International journal of audiology* 58.4 (2019), pp. 193–199.

[5] Grimm, S. and Freudenberger, J. Wind noise reduction for a closely spaced microphone array in a car environment. *EURASIP Journal on Audio, Speech, and Music Processing* 2018.1 (2018), pp. 1–9.

[6] Nelke, C. M. *Wind noise reduction: signal processing concepts*. Wissenschaftsverlag Mainz, 2016.

[7] Sapozhnykov, V. V. Sub-Band Detector for Wind-Induced Noise. *Journal of Signal Processing Systems* 91.5 (2019), pp. 399–409.

[8] Nelke, C. M. and Vary, P. Measurement, analysis and simulation of wind noise signals for mobile communication devices. *2014 14th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE. 2014, pp. 327–331.

[9] Virtanen, T., Plumbley, M. D. and Ellis, D. *Computational analysis of sound scenes and events*. Springer, 2018.

[10] Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[11] Mohri, M., Rostamizadeh, A. and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

[12] Hosmer Jr, D. W., Lemeshow, S. and Sturdivant, R. X. *Applied logistic regression*. Vol. 398. John Wiley & Sons, 2013.

[13] Edwards, A. W. F. *Likelihood*. CUP Archive, 1984.

[14] Osborne, J. W. *Best practices in logistic regression*. Sage Publications, 2014.

[15] Hilbe, J. M. *Logistic regression models*. CRC press, 2009.

[16] Maddala, G. S. and Lahiri, K. *Introduction to econometrics*. Vol. 2. Macmillan New York, 1992.

[17] Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[18] Bishop, C. M. *Pattern recognition and machine learning*. springer, 2006.

[19] Bonaccorso, G. *Machine learning algorithms*. Packt Publishing Ltd, 2017.

[20] Petersen, K. and Pedersen, M. The matrix cookbook, version 20121115. *Technical Univ. Denmark, Kongens Lyngby, Denmark, Tech. Rep* 3274 (2012).

[21] Powers, D. M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061* (2020).

[22] Olson, D. L. and Delen, D. *Advanced data mining techniques*. Springer Science & Business Media, 2008.

[23] Tharwat, A. Classification assessment methods. *Applied Computing and Informatics* (2020).

[24] Goutte, C. and Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. *European conference on information retrieval*. Springer. 2005, pp. 345–359.

[25] Boussemart, Y., Cummings, M. L., Fargeas, J. L. and Roy, N. Supervised vs. unsupervised learning for operator state modeling in unmanned vehicle settings. *Journal of Aerospace Computing, Information, and Communication* 8.3 (2011), pp. 71–85.

[26] Rabiner, L. and Juang, B. An introduction to hidden Markov models. *ieee assp magazine* 3.1 (1986), pp. 4–16.

[27] Giannakopoulos, T. and Pikrakis, A. *Introduction to Audio Analysis: a MATLAB® approach*. Academic Press, 2014.

[28] Brunton, S. L. and Kutz, J. N. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[29] Sharma, G., Umapathy, K. and Krishnan, S. Trends in audio signal feature extraction methods. *Applied Acoustics* 158 (2020), p. 107020.

[30] Zhao, S., Cheng, E., Qiu, X., Burnett, I. and Chia-chun Liu, J. Spatial decorrelation of wind noise with porous microphone windscreens. *The Journal of the Acoustical Society of America* 143.1 (2018), pp. 330–339.

[31] Walker, K. T. and Hedlin, M. A. A review of wind-noise reduction methodologies. *Infrasound monitoring for atmospheric studies* (2010), pp. 141–182.

[32] Raspet, R., Webster, J. and Dillion, K. Framework for wind noise studies. *The Journal of the Acoustical Society of America* 119.2 (2006), pp. 834–843.

[33] Strasberg, M. Dimensional analysis of windscreen noise. *The Journal of the Acoustical Society of America* 83.2 (1988), pp. 544–548.

[34] Speech and multimedia transmission quality (STQ); Part 1: Background noise simulation technique and background noise database. *ETSI EG 202 396-1* (2006).

[35] Jackson, I. R., Kendrick, P., Cox, T. J., Fazenda, B. M. and Li, F. F. Perception and automatic detection of wind-induced microphone noise. *The Journal of the Acoustical Society of America* 136.3 (2014), pp. 1176–1186.

[36] Mohd Hanifa, R., Isa, K., Mohamad, S., Mohd Shah, S., Soosay Nathan, S., Ramle, R. and Berahim, M. Voiced and unvoiced separation in malay speech using zero crossing rate and energy. eng. *Indonesian Journal of Electrical Engineering and Computer Science* 16.2 (2019), pp. 775–.

[37] Loy, G. *Musimathics: the mathematical foundations of music*. Vol. 1. MIT press, 2011.

[38] Nelke, C. M., Chatlani, N., Beaugeant, C. and Vary, P. Single microphone wind noise PSD estimation using signal centroids. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014, pp. 7063–7067.

[39] Chung, K. Comparisons of spectral characteristics of wind noise between omnidirectional and directional microphones. *The Journal of the Acoustical Society of America* 131.6 (2012), pp. 4508–4517.

[40] Welch, P. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics* 15.2 (1967), pp. 70–73.

[41] Stoica, P., Moses, R. L. et al. Spectral analysis of signals. (2005).

[42] Mirabilii, D. and Habets, E. A. Spatial Coherence-Aware Multi-Channel Wind Noise Reduction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 1974–1987.

[43] Corcos, G. The structure of the turbulent pressure field in boundary-layer flows. *Journal of Fluid Mechanics* 18.3 (1964), pp. 353–378.

[44] Mirabilii, D. and Habets, E. A. Simulating multi-channel wind noise based on the Corcos model. *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE. 2018, pp. 560–564.

[45] Harris, F., Chen, X. and Venosa, E. An efficient FFT based spectrum analyzer for arbitrary center frequencies and arbitrary resolutions analysis. *2011 IEEE 12th International Workshop on Signal Processing Advances in Wireless Communications*. IEEE. 2011, pp. 571–575.

[46] Iskra, D., Grosskopf, B., Marasek, K., Heuvel, H., Diehl, F. and Kiessling, A. Speecon-speech databases for consumer devices: Database specification and validation. (2002).

[47] Szöke, I., Skácel, M., Mošner, L., Paliesek, J. and Černock, J. Building and evaluation of a real room impulse response dataset. *IEEE Journal of Selected Topics in Signal Processing* 13.4 (2019), pp. 863–876.

[48] Hoogeboom, P. J. Off-line synchronization of measurements based on a common pseudorandom binary signal. *Behavior Research Methods, Instruments, & Computers* 35.3 (2003), pp. 384–390.

[49] Alpaydin, E. *Introduction to machine learning*. MIT press, 2020.