University of Rouen Normandié
University of Paris-Saclay

Improving beehive state prediction from audio using relevant audio timbre features and deep neural networks

Realized by: Khellouf Leila
Internship tutor: Dominique Fourer
IBISC Lab (EA 4526)
University of Evry-Val-d'Essonne / Paris-Saclay, France,
Email: lkhellouf90@yahoo.com — dominique.fourer@univ-evry.fr

Internship tutor: Dominique Cassou-Ribehart, Jean-Paul Gavini
Starling Partners
Paris, France
Email: dcr@starling.partners — jpg@starling.partners

Final Year Project

Master Sciences et ingénierie des données, Septembre 2020

1 Acknowledgment

I would like to express my special thanks of gratitude to my teacher **Dominique Fourer** who gave me the golden opportunity to do this wonderful project on the topic *Improving beehive state prediction from audio using relevant timbre features and deep neural networks* , which also helped me in doing a lot of Research even in this special situation of COVID 19 and i came to know about so many new things I am really thankful to them.

My sincere thanks also goes to my internship tutors from Staling Partners Company Dominique Cassou-Ribehart and Jean-Paul Gavini for offering me the summer inter-ship opportunities in their team and my gartitude goes also to all my teachers at the university of ROUEN who have contributed to the knowledge and training I have acquired.

Secondly, I would like to thank my family, my husband Aziz and my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Finally, I dedicate this project TO MY ANGEL my lovely Son ADAM and my beautiful gift in this world.

Contents

2 General introduction

At a high level, any machine learning problem can be divided into three types of tasks: data tasks (data collection, data cleaning, and feature formation), training(building machine learning models using data features) and evaluation (assessing the model). Feature, defined as individual measurable properties or characteristics of a phenomenon being observed, are very useful because they help a machine understand the data and classify it into categories or predict a value.

However , Dataset preprocessing, feature extraction and engineering are steps we take to extract information from the underlying data, information that in a machine learning context should be useful for predicting the class of a sample or the value of some target variable. In audio analysis this process is largely based on finding components of an audio signal that can help us distinguish it from other signals.

In this project, we aim to explore the potential of the different methods of the feature extraction and machine learning to the problem of beehive sound recognition. where the major contribution of this project is the comparison of the Audio Timbre ToolBox algorithm with other feature extraction methods.

This project is organized as follow. In the first chapter we will define the general context of the project, Then the second chapter we will propose a supervised methodology for automatically predicting the health state of a honey bee colony from instantaneous field sound recording and we will introduce the feature extraction methods, we comparatively assess a new proposed technique with several state-of-the-art methods with a consideration for the prediction accuracy and for the number of computed features. Finally, our results are discussed with eventual future works.

3 Chapter 1: General context of the project

This first chapter gives a general vision of the conduct of the project. First, we will introduce the state of the art of different research and work on the health of the hive bee. Then, we will discuss the context of the project and the problem. Finally, we will present the Dataset used in this project.

3.1 State of the art

Honey bee is one of the important pollinator insects for flowers, fruits and vegetables. Since decades, several regions of the world are suffering of a decrease of the bees population and of their economic activity based on the managed honey bee colonies. A large number of factors such as pests and diseases are being investigated [1, 2], but require accurate monitoring techniques to assist beekeepers. and breeding, the environment, including weather, agricultural practices and the use of pesticides and the availability and quality of food sources. Bees produce specific sounds when exposed to stressors such as failing queens, predatory mites, and airborne toxicants, however experienced beekeepers are not always able to explain the exact causes of the sound changes without a hive inspection. Unfortunately, hive inspections disrupt the life cycle of bee colonies and put additional stress factors to the bees [3].

In order to develop systems for automatically discriminating different states of a beehive, several works propose to analyze the audio signature of beehive through a machine learning approach [4]. In [5, 6], the authors propose a method which combine the short-time Fourier transform of the analyzed audio with convolutional neural networks to discriminate bee sounds from the chirping of crickets and ambient noise. This approach outperforms classical machine learning methods such as k-nearest neighbors, support vector machines or random forests for classifying audio samples recorded by microphones deployed above landing pads of Langstroth beehives. Hence, the detection of the bee queen presence appears to be the most important task for smart beekeeping and isaddressed [7] with a complete beehive machine-learning-based audio monitoring system.

More recently, in [3, 8] the authors use their experience in audio signal processing and investigate the use of mel-frequency cepstral coefficients (MFCC), and spectral analysis of the sinusoidal components (or modes) as input features of a supervised classification method to automatically predict the absence or the presence of the queen in a beehive from the analyzed audio signal. The present paper pursue these works and propose to improve the bee queen presence prediction accuracy of the existing approaches with a specific attention to the number of the computed feature coefficients which should be as low as possible when an embedded system is used in a real-world application scenario.

In [6] the authors designed several convolutional neural networks and compared their performance with four standard machine learning methods. In [9], the authors explore the use of CNNs to the problem of beehive sound identification and highlight the long-term aspects of such sounds. They stress the need for long-term contextual representations for modeling such data. Also, in [10], the authors present a method to extract long-term features from spectrograms for the task of sound scene classification.

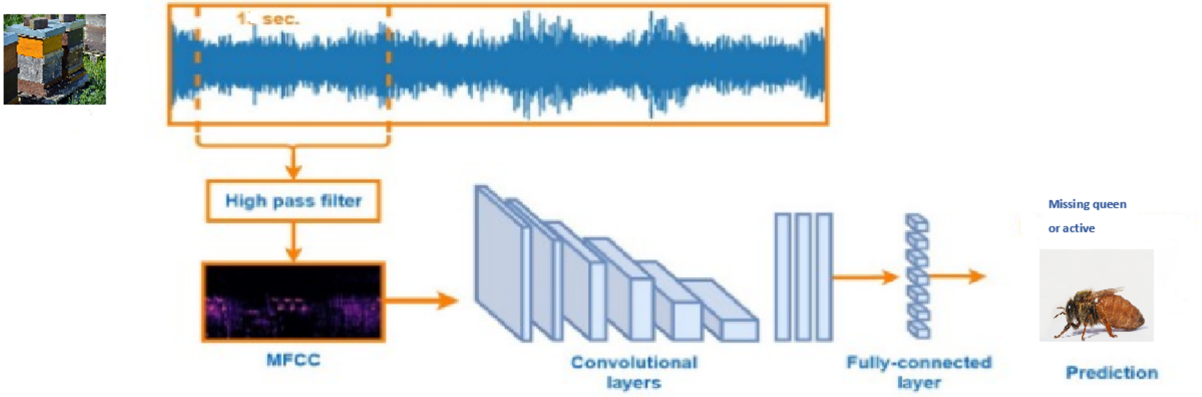
3.2 Problem Formulation

A related problem focus on natural audio recordings extracted from a unique beehive. the goal of this inter ship is to detect the state of the health of the beehive using audio signal as input data.

Hence, the audio signal can convey accurate information about the health of a beehive through a limited amount of data and it is rich in physical characteristics, such as energy, fundamental frequency, and formant, as well as in perceptual characteristics, such as pitch, timbre and rhythm.

Therefore, learning feature representation of audio signals is one of the key interests for audio classification.

In this report, we firstly investigate the most relevant audio features which enable to efficiently monitor a beehive from noisy field recordings and we focused on audio timbre descriptors which offers a large set of audio features which shown its efficiency for different classification tasks such as music instrument recognition from field recordings [11]. Then, we built machine learning models to classify, describe, and generate audio typically concerns modeling tasks. Finally, we will develop and assessment of one or several new techniques using natural recordings provided by **Starling Partners company** The figure bellow show the different steps for the audio classification .



The solution summary — audio data preprocessing and neural networks model

3.3 Data Exploratory

The dataset used in this project was developed in the context of the work in [8] that focuses on the automatic recognition of beehive sounds where the problem was posed as the classification of the sound segments in two classes: *Bee* and *noBee*.

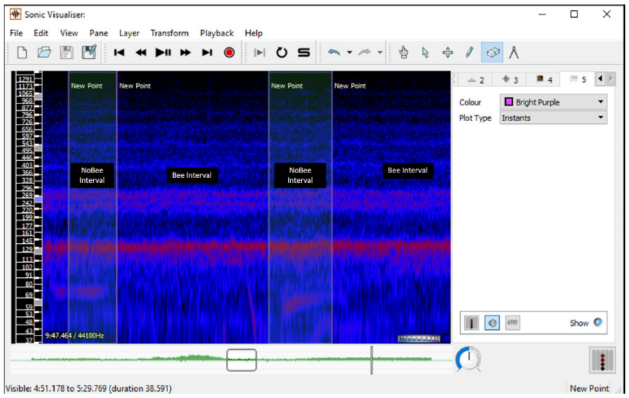
The annotated dataset was developed based on a selected set of recordings acquired in the context of two different projects: the Open Source Beehive (OSBH) project and the NU-Hive project . Both projects main goal is to develop a beehive monitoring system capable of identifying and predict certain events and states of the hive that are of interest to the beekeeper[12].

The Open Source Beehives OSBH Project is a network of citizen scientists tracking bee decline. they used sensor enhanced beehives and data science to study honeybee colonies throughout the world. All of their technology and methods, from the hive and sensor kit designs to the data. The recordings OSBH project, were acquired through a citizen science initiative which asked people from the general public to record the sound from their beehives together with the registering of the hive state at the moment [12].

The NU-Hive project [12] is a comprehensive effort of data acquisition, concerning not only sound, but a vast amount of variables that will allow the study of bees behaviors and other unknown aspects. The selected recordings are taken from 2 hives and labeled regarding two states: queen bee is present, and queen bee not present. Contrary to the OSBH project recordings, the recordings from the NU-Hive project are from a much more controlled and homogeneous environment. Here the occurring external sounds are mainly traffic, car honks and birds.

3.3.1 Annotated dataset

For each selected recording, time segments are labeled as Bee or noBee depending on the perceived source of the sound signal being from bees or external to the hive [12]. As shown in Figure , the annotation procedure consists in hearing the selected recordings and marking the beginning and the end of every sound that could not be recognized as a beehive sound.



Colour: Bright Purple
Plot Type: Instants

Visible: 4:51.178 to 5:29.769 (duration 38.591)

Figure 1: Example of the annotated procedure for one audio file

The whole annotated dataset consists of 78 recordings of varying lengths which make up for a total duration of approximately 12 hours of which 25% is annotated as noBee events [12] . About 60% of the recordings are from the NU-Hive dataset and represent 2 hives, the remaining are recordings from the OSBH dataset and 6 different hives. The recorded hives are from 3 main locations: North America, Australia and Europe [12] .

In our case, the dataset used contains only six hives , 2 hives from OSBH project and 4 hives from NU-Hive project. All the audio signal was decomposed into one second. The table 1 show the decomposition of the data by hive (e.g. CF003 is made of 3700 s of audio signal labeled as active− bee)

State\ Hive	Label	CF001	CF003	$CJ001$	$GH001$	Hive1	Hive3
Active	bee	0	3700	0	1401	2687	656
	nobee	0	500	0	1724	901	530
Missing Queen	bee	16	0	802	0	1476	6557
	nobee	0	0	698	0	903	2265

Table 1: Content of the investigated database expressed in seconds

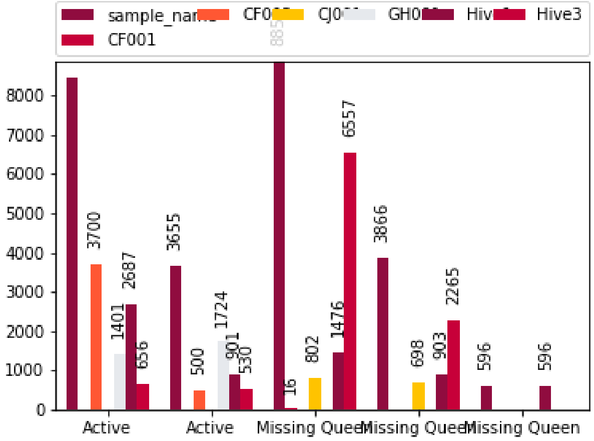


Figure 2: data representation

4 Chapter 2: Audio signal processing rationale

This chapter somewhat technical, so before we dive in, we define a few key terms pertaining to digital signal processing and audio analysis.

4.1 Audio signal processing

The audio signal is usually represented as time series, where the y-axis measurement is the amplitude of the waveform. the amplitude is usually measured as a function of the change in pressure around the microphone or receiver device that originally picked up the audio. Unless there is metadata associated with the audio samples, these time series signals will often be the only input data for fitting a model.

In signal processing, **sampling** is the reduction of a continuous signal into a series of discrete values. **The sampling frequency** or **rate** is the number of samples taken over some fixed amount of time. A high sampling frequency results in less information loss but higher computational expense, and low sampling frequencies have higher information loss but are fast and cheap to compute.

4.2 Amplitude

The amplitude [1] of a sound wave is a measure of its change over a period (usually of time). Another common definition of amplitude are:

- The maximum extent of a vibration or displacement of a sinusoidal (!) oscillation, measured from the position of equilibrium. Amplitude is the maximum absolute value of a periodically varying quantity.
- The maximum difference of an alternating electrical current or potential from the average value.

The term "**amplitude**", is used to refer to the magnitude of an oscillation, so the amplitude of the sinusoid "$y = A \cdot sin(\omega \cdot t)$" , is $|A|$, where $|A|$ is the absolute value of A.

4.3 Frequency

Amount of times the period occurs and it is computed by measuring the periodicity of the time-domain waveform in Hz or cycles per second. It may also be estimated from the signal spectrum as the frequency of the first harmonic or as the spacing between harmonics of the periodic signal. Humans can hear frequencies between 20 Hz and 20,000 Hz (20 KHz).[13]

[1] "http://www.sengpielaudio.com/calculator-amplitude.htm

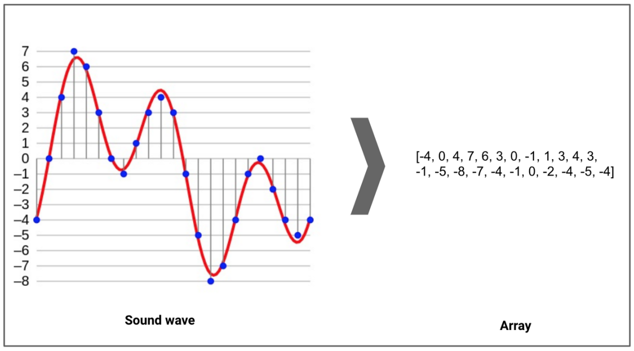


Figure 3: A sound wave, in red, represented digitally, in blue (after sampling and 4-bit quantisation), with the resulting array shown on the right. Original Aquegg Wikimedia Commons

4.4 Fourier Transform

The Fourier Transform [2]decomposes a function of time (signal) into constituent frequencies. In the same way a musical chord can be expressed by the volumes and frequencies of its constituent notes, a Fourier Transform of a function displays the amplitude (amount) of each frequency present in the underlying function (signal).

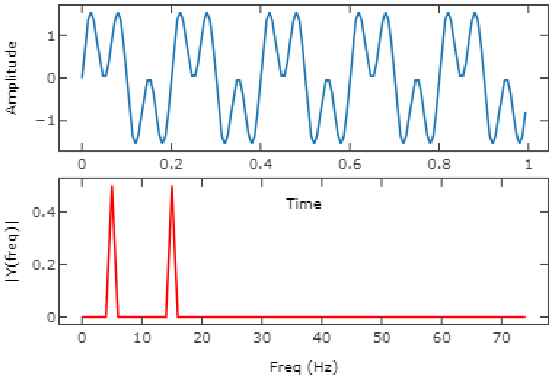


Figure 4: Top: a digital signal; Bottom: the Fourier Transform of the signal

There are variants of the Fourier Transform including the Short-time fourier transform, which is implemented in the Librosa library and involves splitting an audio signal into frames and then taking the Fourier Transform of each frame. In audio processing generally, the Fourier is an elegant and useful way to decompose an audio signal into its constituent frequencies.

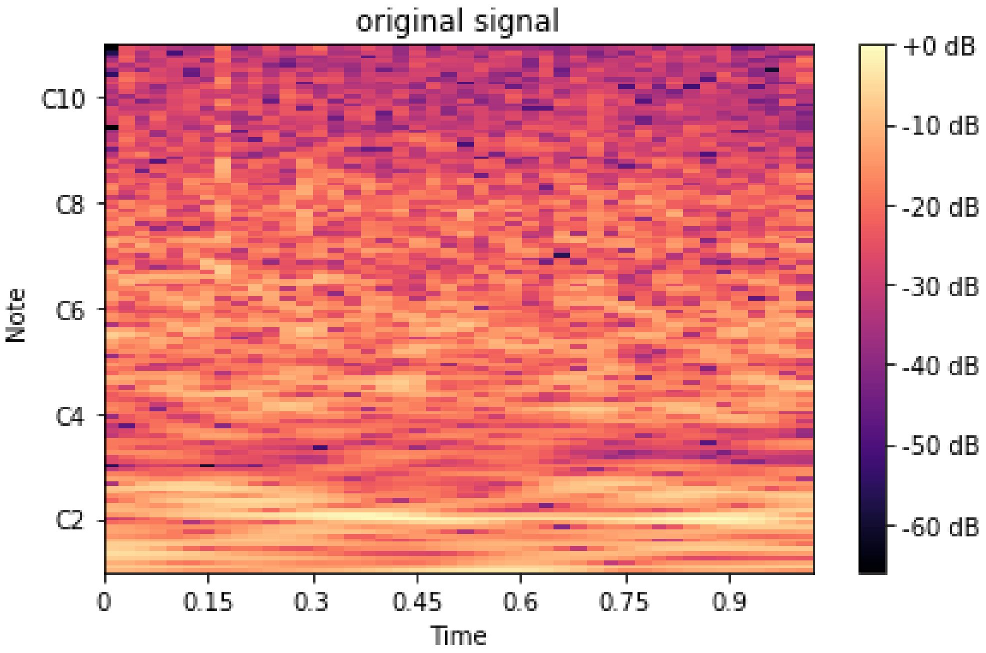


Figure 5: Mel-frequency spetrogram of an audio sample in Bee-hive dataset

4.5 Spectrogram

A spectrogram [3] is a visual representation of the spectrum of frequencies of a signal as it varies with time. When applied to an audio signal, spectrograms are sometimes called sonographs, voiceprints, or voicegrams. When the data is represented in a 3D plot, they may be called **waterfalls**. In 2-dimensional arrays, the first axis is frequency while the second axis is time. The spectrogram is by definition the squared modulus of the STFT, ie: $|STFT(x)(t,\omega)|^2$

4.6 Cepstrum

Cepstrum name was derived from spectrum by reversing the first four latters of the spectrum. The cepstrum is the transformer of the Fourier transform of the registry with unwrapped phase Fourier transform. [13]

The cepstrum is the result of following sequence of mathematical operations:

- transformation of a signal from time domain to frequency domain.
- log of the spectral amplitudes.
- transformation to quefrency domain, where the final independent variable, the quefrency, has actually a time scale.

[2] https://en.wikipedia.org/wiki/Fourier_transform
[3] https://en.wikipedia.org/wiki/Spectrogram

5 Chapter 3: Proposed approach

This chapter will deal in detail with the different proposed approach. In order to predict the heath state of a beehive, we use the supervised learning approach illustrated in Fig. 6 which consists in (step-1) extracting features from the analyzed audio signals and then (step-2) computing labels using a classification method trained on annotated audio examples.

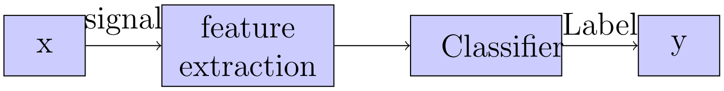


Figure 6: Overview of the proposed approach.

5.1 Feature extraction

Every audio signal consists of many features. However, we must extract the characteristics that are relevant to the problem we are trying to solve. The process of extracting features to use them for analysis is called feature extraction. Let us study a few of the features in detail

5.1.1 Mel-frequency cepstral coefficients

Mel-frequency cepstral coefficients (MFCC) is a classical representation based on the non linear mel frequency scale which finds a large number of audio applications such as speech recognition and music information retrieval [14].
MFCCs were introduced by Davis and Mermelstein in the 1980's, and have been state-of-the-art ever since. Prior to the introduction of MFCCs, Linear Prediction Coefficients (LPCs) and Linear Prediction Cepstral Coefficients (LPCCs) and were the main feature type for automatic speech recognition (ASR), especially with HMM classifiers.
They are introduced in several studies for beehive audio analysis in [15] for which they provide promising results.

The various steps for implementing MFCC are explained in subsequent paragraphs.

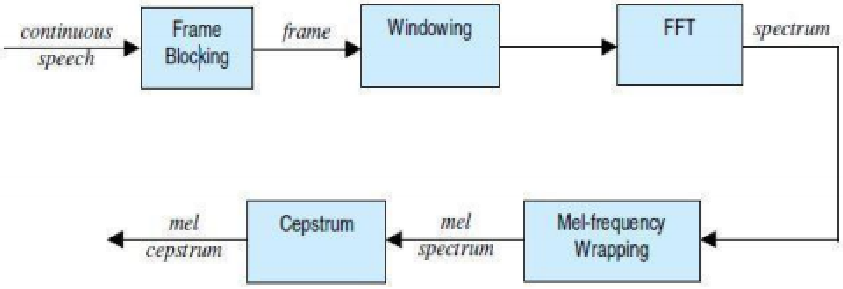


Figure 7: MFCC Process

5.1.1.1 Frame Bolcking: In this step, the continuous speech signal is divided into frames of N samples. The next frame starts after M samples. The value of M is less than N. The first frame consists of the first N samples. The adjacent frames are separated by M samples and there is a overlapping of N - M samples. Usual values of N and M are 256 and 128 respectively. Hence, a speech signal is processed in frames which are overlapping with each other. From each frame, a feature vector is computed [16].

5.1.1.2 Windowing: Windowing reduces the effect of the spectral artifacts (spectral leakage/smearing) that arise from discontinuities at the frame endpoints. The signal discontinuities are minimized by using the window to taper the signal to zero at the beginning and end of each frame. Let $w(n)$ represents the window function, then the result of windowing is $y(n)$ represented by $y(n) = x(n)w(n), 0 \leq n \leq N - 1$. Hammming window is usually used [16].

5.1.1.3 Short-Time Fast Fourier Transform: Each frame of N samples is transformed from the time domain into the frequency domain using Fast Fourier Transform (FFT) [16]. The FFT is used for fast implementation of the Discrete Fourier Transform (DFT) [16]. For the set of N samples it is given by Equation bellow:

$$X_k = \sum_{n=0}^{N-1} e^{-j2\pi kn/N}, k = 0, 1, 2, .., N - 1 \tag{1}$$

In general $X_k's$ are complex numbers and their absolute values are taken into account.

5.1.1.4 Mel-Frequency Warping: Mel is an abbreviation of the word melody. It is a unit of pitch. It is defined to be equal to one thousandth of the pitch (π) of a simple tone with frequency of 1000 Hz and amplitude of 40 dB above the auditory threshold. The mel-frequency is given by Equation bellow:

$$Mel(f) = 2595log(1 + f/700) \tag{2}$$

The filter bank has a triangular band pass frequency response. This filter bank when applied in the frequency domain means applying the triangular-shape windows to the spectrum. At low frequencies there is linear spacing between filters but at high frequencies they are spaced logarithmically [16]

5.1.1.5 Log Compression and Discrete Cosine Transform: The log Mel spectrum is converted back to time domain. The result is called the Mel Frequency Cepstral Coefficients (MFCC). A good estimation of the spectral properties of the signal for a given frame is obtained from this type of representation of the speech spectrum.The Mel spectrum coefficients are real numbers. They are converted to time domain using Discrete Cosine Transform (DCT) [16].

In this study, we chose $K_m = 20$ as proposed in [8] and we used the python implementation provided in librosa [17].

5.1.2 The Short-Time Fourier Transform (STFT)

Short-time Fourier transform (STFT) is a sequence of Fourier transforms of a windowed signal. STFT provides the time-localized frequency information for situations in which frequency components of a signal vary over time, whereas the standard Fourier transform provides the frequency information averaged over the entire signal time interval [18].

Given a signal x, we define its STFT $F_x^h(t,\omega)$ at any time t (expressed in seconds) and frequency ω(expressed in red.s^{-1}), using a differentiable analysis window h, as: [19]

$$F_x^h(t,\omega) = \int_{\mathbb{R}} x(u)h(t-u)^* e^{-j\omega u} du \tag{3}$$

$$= e^{-j\omega t} \int_{\mathbb{R}} x(t+u)h(-u)^* e^{-j\omega u} du \tag{4}$$

with $j^2 = -1$, and z^* being the complex conjugate of z. Thus, as Time Frequency representation is provided by the spectrogram that can be computed as $|F_x^h(t,\omega)|^2$.

Figure Fig. 8

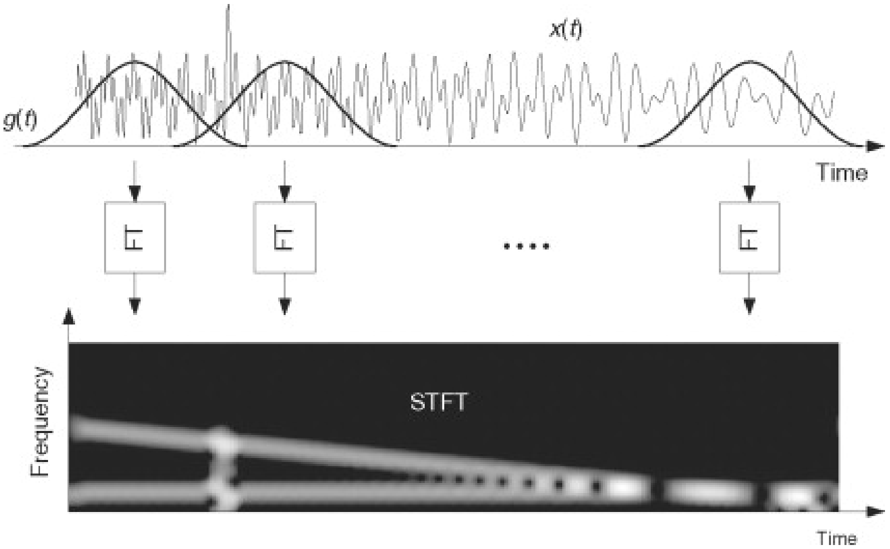


Figure 8: Short-time Fourier transform

5.1.3 Audio timbre features

Peeters *et al.* proposed in [20] a large set of audio features which convey information about the perceived timbre of a given sound (e.g. brightness that is commonly associated to the spectral centroid [21]). These audio features can be organized as follows (cf. table 2):

1. Temporal features describe the temporal evolution of a signal
2. Harmonic features are deduced from detected the estimated fundamental frequency (F_0) and use the prior waveform model of quasi-harmonic sounds. Moreover, the tonal part of sounds can be isolated from signal mixture and be described (e.g. noisiness, inharmonicity, etc.).
3. Spectral features use the time-frequency representation of the signal (i.e. STFT) without prior waveform model (e.g. spectral centroid, spectral decrease, etc.)
4. Perceptual features use auditory-filtered bandwidth versions of the signal and aim at approximating the human perception of sounds using Equivalent Rectangular Bandwidth (ERB) scale [22] and gammatone filter-bank [23] (e.g. loudness, ERB spectral centroid, etc.).

We implemented the proposed set of features in python according to the existing matlab timbre toolbox described in detail in [20].

5.2 Classification

Machine learning involves predicting and classifying data and to do so we employ various machine learning algorithms according to the dataset. This project investigates three distinct supervised classification methods which are respectively called (M1) support vector machines (SVM), (M2) convolutional neural network (CNN) and (M3) densely connected neural network (DNN).

5.2.1 Support Vector Machines

The (M1) model based on Support Vector Machines (SVM), let's understand the basics of how SVM works. SVM is a supervised Machine Learning algorithm that is used in many classifications and regression problems. It still presents as one of the most used robust prediction methods that can be applied to many use cases involving classifications.

Support vector machine works by finding an optimal separation line called a 'hyperplane' to accurately separate 2 or more different classes in a classification problem. The principle of SVM consists of a projection of the data into a feature space with a higher dimension before computing a linear hyper-plane separator trained from annotated examples. [25].

Support vectors, on the other hand, are the points that lie the closest to the hyper-plane between the 2 or more classes. These are the data points that are the most difficult to classify. In general, the larger the margin or distance between the support vectors, the easier it is for the algorithm to classify accurately. Hence, once the hyperplane is optimised, it is said to be the optimal separation or the maximum margin classifier. [25]

Due to their robustness, SVM are often used as a baseline classification method in comparison to more efficient techniques such as decision trees or artificial neural networks. However, it requires a suitable choice of the kernel function.

Acronym	Feature name	#
Att	Attack duration (see ADSR model[24])	1
AttSlp	Attack slope (ADSR)	1
Dec	Decay duration (ADSR)	1
DecSlp	Decay slope (ADSR)	1
Rel	Release duration (ADSR)	1
LAT	Log Attack Time	1
Tcent	Temporal centroid	1
Edur	Effective duration	1
FreqMod, AmpMod	Total energy modulation (frequency,amplitude)	2
RMSenv	RMS envelope	2
ACor	Signal Auto-Correlation function (12 first coef.)	24
ZCR	Zero-Crossing Rate	2
HCent	Harmonic spectral centroid	2
HSprd	Harmonic spectral spread	2
HSkew	Harmonic skewness	2
HKurt	Harmonic kurtosis	2
HSlp	Harmonic slope	2
HDec	Harmonic decrease	2
HRoff	Harmonic rolloff	2
HVar	Harmonic variation	2
HErg, HNErg, HFErg,	Harmonic energy, noise energy and frame energy	6
HNois	Noisiness	2
HF0	Fundamental frequency F_0	2
HinH	Inharmonicity	2
HTris	Harmonic tristimulus	6
HodevR	Harmonic odd to even partials ratio	2
Hdev	Harmonic deviation	2
SCent, ECent	Spectral centroid of the magnitude and energy spectrum	4
SSprd, ESprd	Spectral spread of the magnitude and energy spectrum	4
SSkew, ESkew	Spectral skewness of the magnitude and energy spectrum	4
SKurt, EKurt	Spectral kurtosis of the magnitude and energy spectrum	4
SSlp, ESlp	Spectral slope of the magnitude and energy spectrum	4
SDec, EDec	Spectral decrease of the magnitude and energy spectrum	4
SRoff, ERoff	Spectral rolloff of the magnitude and energy spectrum	4
SVar, EVar	Spectral variation of the magnitude and energy spectrum	4
SFErg, EFErg	Spectral frame energy of the magnitude and energy spectrum	4
Sflat, ESflat	Spectral flatness of the magnitude and energy spectrum	4
Scre, EScre	Spectral crest of the magnitude and energy spectrum	4
ErbCent, ErbGCent	ERB scale magnitude spectrogram / gammatone centroid	4
ErbSprd, ErbGSprd	ERB scale magnitude spectrogram / gammatone spread	4
ErbSkew, ErbGSkew	ERB scale magnitude spectrogram / gammatone skewness	4
ErbKurt, ErbGKurt	ERB scale magnitude spectrogram / gammatone kurtosis	4
ErbSlp, ErbGSlp	ERB scale magnitude spectrogram / gammatone slope	4
ErbDec, ErbGDec	ERB scale magnitude spectrogram / gammatone decrease	4
ErbRoff, ErbGRoff	ERB scale magnitude spectrogram / gammatone rolloff	4
ErbVar, ErbGVar	ERB scale magnitude spectrogram / gammatone variation	4
ErbFErg, ErbGFErg	ERB scale magnitude spectrogram / gammatone frame energy	4
ErbSflat, ErbGSflat	ERB scale magnitude spectrogram / gammatone flatness	4
ErbScre, ErbGScre	ERB scale magnitude spectrogram / gammatone crest	4
Total		164

Table 2: Acronym, name and number of the used timbre descriptors.

In this project, we use a Gaussian radial basis function $f(x, \hat{x}) = \exp(-\gamma||x - \hat{x}||^2)$ where γ is defined as one divided by the number of input features. In our study, we used the python implementation provided by Scikit-learn[4].

[4] https://scikit-learn.org

5.2.2 Artificial neural network

Neural network is a type of artificial intelligence that attempts to imitate the way a human brain works, it have been shown to be very promising systems in many forecasting applications and business classification applications due to their ability to learn from the data [26]. Neural networks are particularly effective for predicting events when the networks have a large dataset.

ANNs are composed of multiple nodes. The neurons are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its activation or node value. Each link is associated with weight. ANNs are capable of learning, which takes place by altering weight values.

There are several kinds of artificial neural networks. These type of networks are implemented based on the mathematical operations and a set of parameters required to determine the output.

- **The multilayer perceptrons**: is a structure composed by several hidden layers of neurons where the output of a neuron of a layer becomes the input of a neuron of the next layer. Moreover, the output of a neuron can also be the input of a neuron of the same layer or of neuron of previous layers (this is the case for recurrent neural networks).On last layer, called output layer, we may apply a different activation function as for the hidden layers depending on the type of problems we have at hand : *regression or classification* [27]

- **Feed forward Neural Network Artificial Neuron** : This neural network is one of the simplest form of ANN, where the data or the input travels in one direction. The data passes through the input nodes and exit on the output nodes. This neural network may or may not have the hidden layers. In simple words, it has a front propagated wave and no back propagation by using a classifying activation function usually [26].

- **Convolutional Neural Network**: Convolutional neural networks are similar to feed forward neural networks , where the neurons have learn-able weights and biases. Its application have been in signal and image processing which takes over OpenCV in field of computer vision [26].

5.2.3 Convolutional neural networks

A convolutional neural network is typically composed of a collection of layers which will be sorted by their functionalities. CNN was introduced by LeCun et al[28] , they design and established the framework of CNN by developing seven learned layers including four convolutional and pooling layers followed by three fully-connected layers artificial neural network known as LeNet-5. LeNet-5 was accustomed to classify written digits and trained with the algorithmic backpropagation program [29] that made it attainable to acknowledge patterns directly from raw pixels, and therefore, eliminating a separate feature extraction mechanism. But, because of the lack of sufficient training information and computing power, this design didn't perform well under challenging issues.

Convolutional Neural Network (CNN) is a supervised method. It consists of an input and an output layer, as well as multiple hidden layers. These layers are generally divided into three types: CONV, POOL, and FC (short for fully-connected).[30]

- **Convolution Layer**: the convolution layer plays a significant role in how CNN operates. It forms the fundamental unit of a ConvNet wherever most of the computation is concerned. The layer's parameters focus around the use of learnable kernels. These kernels are typically tiny in spatial dimensionality, however, unfold on the whole dimension of the depth of the input. Once the information hits a convolution layer, the layer convolves every filter across the spatial dimensionality of the data to provide a 2D activation map. The output of neurons of which are connected to local regions of the input can be verified through the convolution layer through the calculation of the scalar product between their weights and also the area connected to the input volume.Neurons that consist of identical feature map shares the weight (parameter sharing) thereby reducing the complexness of the network by keeping the number of parameters low [30].
- **Pooling Layer** : CNN contains not solely convolution layers but also, conjointly some pooling layers. There may be a pooling layer instantly after a convolutional layer. It suggests that the outputs of the convolution layers are the inputs to the pooling layers of the network. Pooling operations cut the dimensions of the feature maps by the victimization of some functions to summarize subregions, like taking the common or the maximum value. Pooling layers aim to step by step cut the dimensionality of the data, and therefore additionally reduce the number of parameters and also the procedure complexness of the model and thus control the matter of overfitting. A number of the common pooling operations are max pooling, average pooling, stochastic pooling , spectral pooling , spatial pyramid pooling , L2-norm pooling , and multiscale orderless pooling [30].
- **Fully Connected Layer**: In a fully connected layer Neurons at some stage have connections to all the activations in the previous layer. Their activations will thus be computed with a matrix operation followed by a bias offset. A fully connected layer passes the two dimensional output to the output layer wherever we can utilize a softmax function or a sigmoid to predict the input class label. [30]

5.2.3.1 Proposed CNN architecture The (M2) model based on convolutional neural networks (CNN, or ConvNet) has shown its efficiency in audio and image classification problems. In our study, we use the deep CNN detailed in Table 3 inspired from [8] where the size of the input layer depends on the choice of the computed audio features (i.e MFCC or STFT). The input is then processed by 4 convolutional blocks (two layers with 16 filters of size 3×3 and two layers of 16 filters of size 3×1) including a batch normalization, max pooling layer of size 2×2 and a dropout of 25%. The output of the convolutional blocks is followed by 3 fully connected layers (256 units, 32 units and 1 unit). All the fully connected layers use a leaky rectifier activation function except for the output layer which uses the softmax function. We implemented this model in python using Keras[5].

#	Output size	Layer type	Filter size
1	$20 \times 44 \times 1$	Input	
2	$20 \times 44 \times 16$	Convolutional	3×3
3	$20 \times 44 \times 16$	BatchNormalization	-
4	$10 \times 22 \times 16$	Max pooling	2×2
5	$10 \times 22 \times 16$	Convolutional	3×3
6	$10 \times 22 \times 16$	BatchNormalization	-
7	$5 \times 11 \times 16$	Max pooling	2×2
8	$5 \times 11 \times 16$	Convolutional	3×1
9	$5 \times 11 \times 16$	BatchNormalization	-
10	$3 \times 6 \times 16$	Max pooling	2×2
11	$3 \times 6 \times 16$	Convolutional	3×1
12	$3 \times 6 \times 16$	BatchNormalization	-
13	$2 \times 3 \times 16$	Max pooling	2×2
14	256	Fully connected	
15	32	Fully connected	
16	2	Fully connected	
17	2	Soft max	
18	1	Output	

Table 3: Architecture of our proposed CNN.

5.2.3.2 Proposed DNN architecture The proposed Dense neural network (DNN) is described in table 4. Its input layer corresponds to the 164 timbre feature coefficients which are computed according to Table 2. Then, it is processed by 4 fully connected layers of 328 neurons and a output layer of 2 neurons. All the layers use leaky rectifier as activation function with an exception for the output layer which uses the softmax function.

[5] https://keras.io/

#	Output size	Layer type
1	164×1	Input
2	164	Fully connected
3	328	BatchNormalization
4	328	Fully connected
5	328	BatchNormalization
6	328	Fully connected
7	328	BatchNormalization
17	2	Soft max
18	1	Output

Table 4: Architecture of the proposed DNN for which the input layer corresponds to the 164 computed timbre features.

5.3 Feature Selection

Feature selection (also known as variable selection or attribute selection) , appears in different areas such as pattern recognition, machine learning , data mining and statistical analysis [31]. Essentially, it is the process of selecting the most important/relevant features of a dataset, and it not only reduces the dimensionality of the data facilitating their visualization and understanding; but also it commonly leads to more compact models with better generalization ability (Pal and Mitra 2004). All these characteristics make feature selection an interesting research area, wherein the last decades, numerous feature selection methods have been introduced.[32].

The importance of feature selection, can best be recognized when we dealing with a dataset that contains a vast number of features. This type of dataset is often referred to as a *high dimensional dataset*. Now, with this high dimensionality, comes a lot of problems such as - this high dimensionality will significantly increase the training time of machine learning model, it can make the model very complicated which in turn may lead to Overfitting [32].

Often, in a high dimensional feature set, there remain several features which are redundant meaning these features are nothing but extensions of the other essential features. These redundant features do not effectively contribute to the model training as well. So, clearly, there is a need to extract the most important and the most relevant features for a dataset in order to get the most effective predictive modeling performance [32].

5.3.1 The difference between feature selection and dimensionality reduction

Sometimes, feature selection is mistaken with dimensionality reduction. But they are different.

Feature selection is different from dimensionality reduction. Both methods tend to reduce the number of attributes in the dataset, but a dimensionality reduction method does so by creating new combinations of attributes (sometimes known as feature transformation), whereas feature selection methods include and exclude attributes present in the data without changing them.

Some examples of dimensionality reduction methods are **Principal Component Analysis, Linear Discriminant Analysis , Singular Value Decomposition, etc** [6]

5.3.2 Different typed of feature selection methods

The image below provides a summary of the hierarchy of feature selection techniques [32].

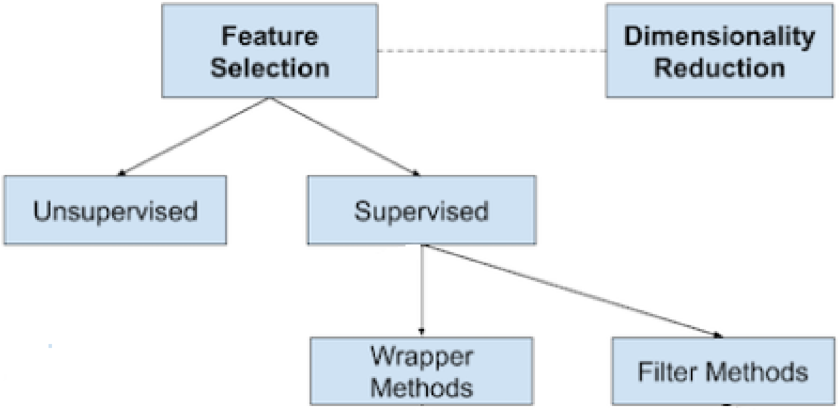


Figure 9: Overview of feature selection techniques. Image Source: machineLearningMastery.com

We will cover some different types of general feature selection methods like Filter methods, Wrapper methods, and Embedded methods.

5.3.3 Filter methods

The following image best describes filter-based feature selection methods :

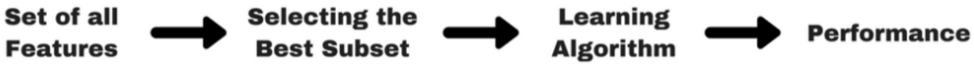


Figure 10: Filter based feature selection. Image Source: Analytics Vidhya

Filter feature selection methods use statistical techniques to evaluate the relationship between each input variable and the target variable, and these scores are used as the basis to choose (filter) those input variables that will be used in the model. Applied directly on

[6] https://towardsdatascience.com/feature-selection-and-dimensionality-reduction-f488d1a035de

the training data, the filter approach is based on feature ranking techniques that use an evaluation critirion and threshold to determine the feature relevance and decide whether to keep it or discard it. The feature relevance it determined by its capability to provide useful information about the different classes (Chandrashkar and Sahin, 2014). Filter algorithms are usually computationally less expensive than the other methods (Boln-Canedo et al.,2016). A common drawback for filter methods is that they are adequate only for aindependent features, otherwise, feature will be redundant (Guyon et al.,2008).

5.3.4 Wrapper methods

Like filter methods, The following image describes well the Wrapper methods based feature selection [32]:

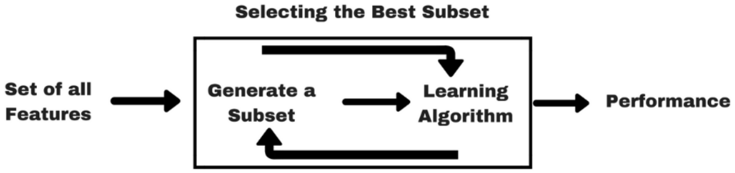


Figure 11: Wrapper based feature selection. Image Source: Analytics Vidhya

A wrapper method needs one machine learning algorithm and uses its performance as evaluation criteria. This method searches for a feature which is best-suited for the machine learning algorithm and aims to improve the mining performance .it is based on three components: a search strategy, a predictor, and an evaluation function (Liu and Motoda, 2007). **The search strategy** determines the subset of features to be evaluated. **The predictor** (considered as a black box) can be any classification method and its performance is used as the objective function **to evaluate** the subset of features defined by the search strategy so as to find the optimum subset that gives the best accuracy of it (Guyon et al., 2008). The wrapper approach outperform the filter approach bu it is more time consuming and requires more computational resources(Boln-Canedo et al.,2016).

Some typical examples of wrapper methods are forward feature selection, backward feature elimination, recursive feature elimination, etc.

- **Forward Selection:** The procedure starts with an empty set of features [reduced set]. The best of the original features is determined and added to the reduced set. At each subsequent iteration, the best of the remaining original attributes is added to the set.
- **Backward Elimination:** The procedure starts with the full set of attributes. At each step, it removes the worst attribute remaining in the set.
- **Combination of forward selection and backward elimination:** The stepwise forward selection and backward elimination methods can be combined so that, at each step, the procedure selects the best attribute and removes the worst from among the remaining attributes.

- **Recursive Feature elimination:** Recursive feature elimination performs a greedy search to find the best performing feature subset. It iteratively creates models and determines the best or the worst performing feature at each iteration. It constructs the subsequent models with the left features until all the features are explored. It then ranks the features based on the order of their elimination. In the worst case, if a dataset contains N number of features RFE will do a greedy search for 2N combinations of features.

5.3.5 Embedded methods

Embedded methods [32] are iterative in a sense that takes care of each iteration of the model training process and carefully extract those features which contribute the most to the training for a particular iteration. Regularization methods are the most commonly used embedded methods which penalize a feature given a coefficient threshold [32]. This is why Regularization methods are also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm (such as a regression algorithm) that bias the model toward lower complexity (fewer coefficients).

Examples of regularization algorithms are the LASSO, Elastic Net, Ridge Regression, etc.

5.3.6 Principal Component Analysis PCA

PCA is fundamentally a simple dimensionality reduction technique that transforms the columns of a dataset into a new set features. It does this by finding a new set of directions (like X and Y axes) that explain the maximum variability in the data. This new system coordinate axes is called Principal Componenets (PCs). The projections of the original data on the new set of coordinate axes (PCs) serves as the new transformed dataset. The PCA is used for two reasons:

- **Dimensionality Reduction:** The information distributed across a large number of columns is transformed into principal components (PC) such that the first few PCs can explain a sizeable chunk of the total information (variance). These PCs can be used as explanatry variables in Machine Learning models.
- **Visualize Classes:** Visualising the separation of classes (or clusters) is hard for data with more than 3 dimensions (features). With the first two PCs itself, it's usually possible to see a clear separation.

5.3.7 Mutual information

Features selection algorithms aim at computing a subset of descriptors that conveys the maximal amount of information to model classes. From a statistical point of view, if we consider classes and feature descriptors as realizations of random variables C and F[11]. The mutual information (MI) is a measure of the amount of information that one random variable has about another variable [33] .This definition is useful within the context of feature selection because it gives a way to quantify the relevance of a feature subset with respect to the output vector [34]. The relevance can be measured with the mutual information defined by:

$$I(C,F) = \sum_{c}\sum_{f} P(c,f)\log\frac{P(c,f)}{P(c)P(f)} \tag{5}$$

where $P(c)$ denotes the probability of $C = c$ which can be estimated from the approximated probability density functions (pdf) using a computed histogram [11].

We can also say

$$I(C,F) = H(C) - H(C|F) \tag{6}$$

where, $H(C)$ is the marginal entropy, $H(C|F)$ is the conditional entropy, and $H(C,F)$ is the joint entropy of C and F. If $H(C)$ represents the measure of uncertainty about a random variable, then $H(C|F)$ measures what F does not say about C.This is the amount of uncertainty in C after knowing Y and this substantiates the intuitive meaning of mutual information as the amount of information that knowing either variable provides about the other [35].

In this study, we used the python implementation of Mutual information provided in scikit-lea [7].

[7] $https : //scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html$

6 Evaluation

6.1 Dataset

Our experiments use the publicly available dataset [8] which was investigated in [8] and was acquired from 6 distinct beehives in the context of two different projects: the Open Source Beehive (OSBH) project and the NU-Hive project [4]. The dataset contains about 96 hours of audio signals which were recorded inside the beehive using a MicroElectrical-Mechanical System (MEMS) microphone, we only used 30 files from 576 (Hive 1, Hive 3) of the NU Hive project (3H20), and 20 files of 5 min collected from OSBH project(1H40). The audio signals are labeled as “active” and “missing queen” which correspond to the distinct labels to predict. Moreover, each audio signal is segmented between the distinct labels “bee” and “no bee” to discriminate the sounds produced by the bees from ambient noises. A threshold of 0.5 disregards intervals shorter than half a second, thus defining the minimum duration of the no bee intervals.

6.2 setup

As a pre-processing step, the audio signals are segmented in one-second-long homogeneous time frames (with the same label) and are resampled at $F_s = 22.05$ kHz to obtain 17,295 distincts individual where 8,444 are labeled as “active” and 8,851 are labeled as “missing queen”. Two distinct experimental protocols are used to comparatively assess the different investigated methods.

- The first experiment uses a random split of the whole dataset and merges the data from the 6 beehives. This configuration uses 70% of the dataset as a training set and 30% as the test set.
- The second experiment uses a 4-fold-cross validation where each fold corresponds to a distinct beehive except for four hives which are merged into two distinct folds (CF001+CF003 and CJ001+GH001) because they only contain “active” (resp. “missing queen” individuals).

In each experiment, exactly the same split is used to assess each method to make the result comparable. For the training step, our computations based on the neural networks use the RMSprop optimizer with a batch size equal to 145 and a number of epochs of 50. Our hardware configuration is based on an Intel Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz CPU with 32GB or RAM and a NVIDIA GTX 1080 TI GPU.

6.3 Results

The results obtained for the two experiments expressed in terms of F-measure, Precision, Recall and Accuracy are presented in Tables 5 and 7. The detailed confusion matrices of the random split experiment is presented in Fig. 12.

[8] https://zenodo.org/record/1321278

The results shows that the MFCC + CNN approach obtains the best results for the random split experiment which obtain an excellent average F-measure equal to 0.99 that is higher than 0.90 obtained with the timbre features combined with DNN and than the F-measures obtained with the STFT+CNN method which obtain the poorest results.

Interestingly, the 4-fold-cross-validation experiment shown in the table 6, reveals a lack of generalization and a sensitivity problem of the MFCC-based trained models which obtain the poorest with an accuracy as low as 0.41 (MFCC+CNN). In this configuration, the STFT obtains the best results with an accuracy of 0.61.

In both experiment, the timbre features obtains the most balanced results with a significant lower number of parameters (164 real scalars) in comparison to MFCC (880) and STFT (22,528). This clearly shows an advantage of the timbre features to obtain good prediction using a very low number of computed features. The dataset[9] and python code [10] developed for this work are publicly available.

Method	Size	Label	F-measure	Prec.	Rec.	Acc.
MFCC+SVM	$\mathbb{R}^{20\times44}$	missing-queen	0.91	0.87	0.96	0.90
		active	0.90	0.85	0.90	
MFCC+CNN	$\mathbb{R}^{20\times44}$	missing-queen	0.99	1	0.99	0.99
		active	0.99	0.99	1	
STFT+CNN	$\mathbb{R}^{512\times43}$	missing-queen	0.30	0.97	0.18	0.58
		active	0.70	0.54	0.99	
Timbre feat.+SVM	$\mathbb{R}^{164}$	missing-queen	0.87	0.82	0.93	0.86
		active	0.85	0.92	0.79	
Timbre feat.+DNN	$\mathbb{R}^{164}$	missing-queen	0.90	0.87	0.94	0.90
		active	0.89	0.94	0.85	

Table 5: Comparative results obtained for the random split experiment.

Fold\ De	Training	Test
Fold 1	$CJ001 + GH001 + Hive1 + Hive3$	$CF001 + CF003$
Fold 2	$CF001 + CF003 + Hive1 + Hive3$	$CJ001 + GH001$
Fold 3	$CF001 + CF003 + CJ001 + GH001 + Hive3$	$Hive1$
Fold 4	$CF001 + CF003 + CJ001 + GH001 + Hive1$	$Hive3$

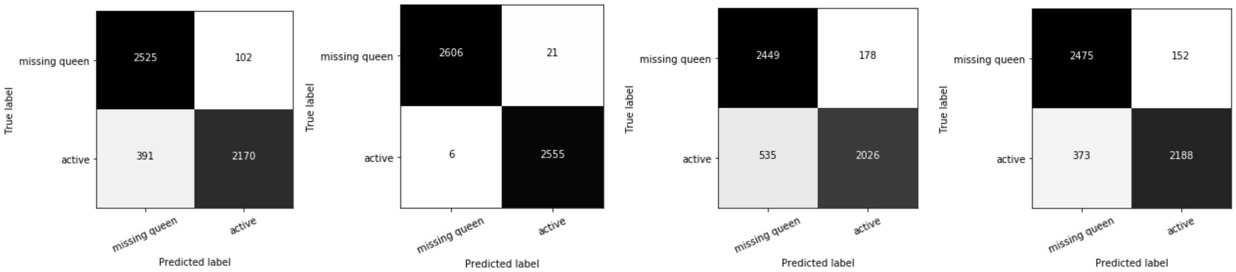
Table 6: Proposed manual partitioning of the data-set to apply a 4−fold−cross validation

[9] $https : //zenodo.org/record/2563940.XGVwpDP7SUk$

[10] $https : //github.com/khelloufleila/AUDIO-BASED-IDENTIFICATION-OF-BEEHIVE-STATES1$

Method	Size	Label	F-measure	Prec.	Rec.	Acc.
MFCC+SVM	$\mathbb{R}^{20\times44}$	missing-queen	0.12	0.09	0.45	0.09
		active	0.01	0.01	0.01	
MFCC+CNN	$\mathbb{R}^{20\times44}$	missing-queen	0.34	0.27	0.51	0.41
		active	0.38	0.417	0.57	
STFT+CNN	$\mathbb{R}^{512\times43}$	missing-queen	0.003	0.33	0.021	0.61
		active	0.76	0.87	0.86	
Timbre feat.+SVM	$\mathbb{R}^{164}$	missing-queen	0.25	0.25	0.39	0.31
		active	0.33	0.38	0.33	
Timbre feat.+DNN	$\mathbb{R}^{164}$	missing-queen	0.30	0.31	0.38	0.45
		active	0.54	0.6	0.7	

Table 7: Comparative results obtained for the 4-fold cross-validation experiment.



(a) MFCC + SVM (b) MFCC + CNN (c) Timbre feat + SVM (d) Timbre feat. + DNN

Figure 12: Confusion matrices obtained respectively with the MFCC+SVM (a), MFCC+CNN (b), Timbre feat.+SVM (c) and Timbre feat.+DNN (d) for the random split experiment.

Feature selection is an unavoidable part for any classification algorithm. We use mutual information theory to select a subset of features from an original feature set based on feature-class information values. We evaluate our feature selection method using classification method.

From experimental analysis, the result of feature selection obtained in Fig. 13 is applied for each segment in Audio timbre features, and it is represented by vector of length $p = 164$ where each value corresponds to a descriptor. A features selection algorithm (mutual information) is first applied on all 164 features and then we rank the features in decreasing order of relevance shown in Table 8. For classification a Logistic Regression algorithm is used to calculated F1-score on the ordered vector.

We observe that for most datasets, the classification accuracy is much better for a subset of features compared to when using the full feature set. Due to significant reduction of the number of features, better computational efficiency is also achieved. As the inflection point shown in Fig. 13, with a minimum of features (i.e 82 features), we get a maximum score.

Acronym	Feature name	#	relevance
Dec	Decay duration (ADSR)	1	49
AttSlp	Attack slope (ADSR)	1	48
DecSlp	Decay slope (ADSR)	1	47
Tcent	Temporal centroid	1	46
Att	Attack duration (see ADSR model[24])	1	45
Rel	Release duration (ADSR)	1	44
LAT	Log Attack Time	1	43
Edur	Effective duration	1	42
FreqMod, AmpMod	Total energy modulation (frequency,amplitude)	2	41
HCent	Harmonic spectral centroid	2	40
RMSenv	RMS envelope	2	39
HNois	Noisiness	2	38
SSkew, ESkew	Spectral skewness of the magnitude and energy spectrum	4	37
SVar, EVar	Spectral variation of the magnitude and energy spectrum	4	36
Sflat, ESflat	Spectral flatness of the magnitude and energy spectrum	4	35
SSlp, ESlp	Spectral slope of the magnitude and energy spectrum	4	34
ErbSflat, ErbGSflat	ERB scale magnitude spectrogram / gammatone flatness	4	33
SRoff, ERoff	Spectral rolloff of the magnitude and energy spectrum	4	32
Scre, EScre	Spectral crest of the magnitude and energy spectrum	4	31
Hdev	Harmonic deviation	2	30
SDec, EDec	Spectral decrease of the magnitude and energy spectrum	4	29
SKurt, EKurt	Spectral kurtosis of the magnitude and energy spectrum	4	28
SFErg, EFErg	Spectral frame energy of the magnitude and energy spectrum	4	27
HinH	Inharmonicity	2	26
HF0	Fundamental frequency F_0	2	25
HVar	Harmonic variation	2	24
SSprd, ESprd	Spectral spread of the magnitude and energy spectrum	4	23
HSprd	Harmonic spectral spread	2	22
HKurt	Harmonic kurtosis	2	21
ErbKurt, ErbGKurt	ERB scale magnitude spectrogram / gammatone kurtosis	4	20
ErbFErg, ErbGFErg	ERB scale magnitude spectrogram / gammatone frame energy	4	19
HinH	Inharmonicity	2	18
ErbSlp, ErbGSlp	ERB scale magnitude spectrogram / gammatone slope	4	17
HDec	Harmonic decrease	2	16
HodevR	Harmonic odd to even partials ratio	2	15
HErg, HNErg, HFErg,	Harmonic energy, noise energy and frame energy	6	14
ErbScre, ErbGScre	ERB scale magnitude spectrogram / gammatone crest	4	13
HTris	Harmonic tristimulus	6	12
ErbRoff, ErbGRoff	ERB scale magnitude spectrogram / gammatone rolloff	4	11
SCent, ECent	Spectral centroid of the magnitude and energy spectrum	4	10
HSkew	Harmonic skewness	2	9
ErbSprd, ErbGSprd	ERB scale magnitude spectrogram / gammatone spread	4	8
HRoff	Harmonic rolloff	2	7
ErbDec, ErbGDec	ERB scale magnitude spectrogram / gammatone decrease	4	6
ErbVar, ErbGVar	ERB scale magnitude spectrogram / gammatone variation	4	5
ErbSkew, ErbGSkew	ERB scale magnitude spectrogram / gammatone skewness	4	4
ErbCent, ErbGCent	ERB scale magnitude spectrogram / gammatone centroid	4	3
ZCR	Zero-Crossing Rate	2	2
ACor	Signal Auto-Correlation function (12 first coef.)	24	1

Table 8: The relevance of the feature timbre toolbox.

In Figure . 14, though there is a certain degree of overlap, the points in MFCC Scatterplot belonging to the same category are distinctly clustered for each Hive and region bound of the 2 classes. This proves that the data captured in the first two PCs is not informative enough to discriminate the categories from each other.

In the other hand, the points in Audio Timbre Scatterplot belonging to the same category are distinctly clustered and region bound. This proves that the data captured in the first two

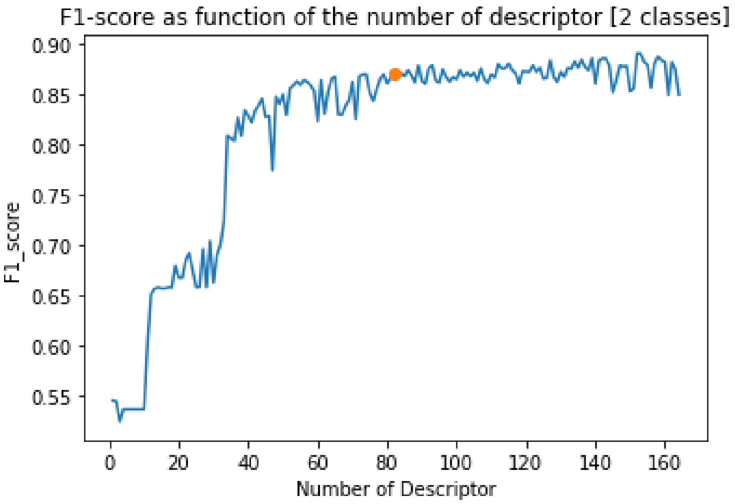
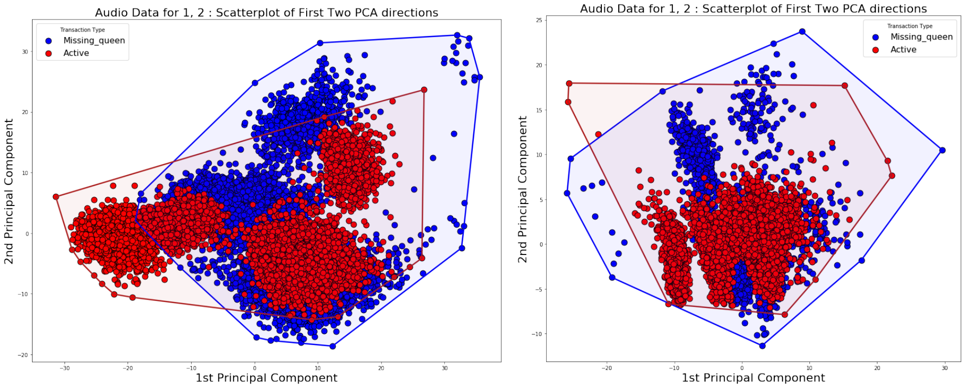


Figure 13: F1-score as function of the number of descriptor (timbre toolbox + LogisticRegression)

PCs is informative enough to discriminate the categories from each other. In other words, we now have evidence that the data is not completely random, but rather can be used to discriminate or explain the Y (the number a given row belongs to).



(a) Scatterplot of First Two PCA directions of MFCC

(b) Scatterplot of First Two PCA directions of Audio Timber Toolbox

Figure 14: Audio data for Missing queen and Active : Scatterplot of First Two PCA directions respectively for MFCC (a) and Audio Timber Toolbox (b).

7 Conclusion

We compared together several methods and we introduced a new one based on audio timbre features taken from the music information retrieval (MIR) literature for predicting the health state of a beehive. Our method shows an improvement of the model robustness when compared to several state-of-the-art methods. It allows to significantly reduce the number of required input size to 164 real scalars and paves the way of efficient embedded-system-based implementations. However, the cross-hive analysis results appear to be insufficient and will require a further investigation involving a larger annotated dataset. Future work will consist in a real-world integration of our approach using an embedded system and a consideration of more beehive state labels such as bees swarming or queen piping and quacking which are full of interest for beekeepers.

Abstract

In recent years with the decrease of the honey bees population, interest for smart beekeeping is growing. Recent works propose new robust methods and low-cost systems for remotely monitoring the health state of a beehive from field audio recording. To this end, the future methods should obtain the best accuracy and reduce the computation cost and the amount of transmitted data. Hence, this paper propose a comparative study of several supervised techniques applied on a publicly available dataset for predicting the presence or the absence of the queen. We also introduce a novel method based on audio timbre features which lead us to a significant reduction of the number of features coefficients and a significant improvement of the prediction results when compared to existing approaches based on mel frequency cepstral coefficients (MFCC) or short-time Fourier transform (STFT).

Key words: Beehive monitoring, audio timbre features, smart beekeeping, deep learning.

Résumé

Ces dernières années, avec la diminution de la population d'abeilles domestiques, l'intérêt pour l'apiculture intelligente est devenu croissant.

Des travaux récents proposent de nouvelles méthodes robustes et des systèmes peu coûteux pour surveiller l'état de santé d'une ruche à distance à partir d'enregistrements audio sur le terrain. À cette fin, les futures méthodes devront obtenir la meilleure précision possible et réduire le coût de calcul ainsi que la quantité de données transmises.

C'est pourquoi ce document propose une étude comparative de plusieurs techniques supervisées appliquées sur une dataset publique disponible afin de prédire la présence ou non de la reine.

Nous introduirons également une nouvelle méthode basée sur les caractéristiques du timbre audio qui nous permet de réduire considérablement le nombre de coefficients caractéristiques et d'améliorer significativement les résultats des prévisions par rapport aux approches existantes basées sur les coefficients cepstraux de fréquence de mel (MFCC) ou la transformée de Fourier à court terme (STFT).

Mots clé: Surveillance de la ruche, caractéristiques de timbre audio, apiculture intelligente, l'apprentissage en profondeur

References

[1] John Bryden, Richard J Gill, Robert AA Mitton, Nigel E Raine, and Vincent AA Jansen. Chronic sublethal stress causes bee colony failure. *Ecology letters*, 16(12):1463–1469, 2013.

[2] Ross D Booton, Yoh Iwasa, James AR Marshall, and Dylan Z Childs. Stress-mediated allee effects can cause the sudden collapse of honey bee colonies. *Journal of theoretical biology*, 420:213–219, 2017.

[3] Stefania Cecchi, Alessandro Terenzi, Simone Orcioni, and Francesco Piazza. Analysis of the sound emitted by honey bees in a beehive. In *Audio Engineering Society Convention 147*, 2019.

[4] S. Cecchi, A. Terenzi, S. Orcioni, P. Riolo, S. Ruschioni, and N. Isidoro. A preliminary study of sounds emitted by honey bees in a beehive. In *Audio Engineering Society Convention 144*, Milan, Italy, May 2018.

[5] Vladimir Alekseevich Kulyukin, Sarbajit Mukherjee, Yulia B Burkatovskaya, et al. Classification of audio samples by convolutional networks in audiobeehive monitoring. *Tomsk State University Journal of Control and Computer Science*, (45):68–75, 2018.

[6] Vladimir Kulyukin, Sarbajit Mukherjee, and Prakhar Amlathe. Toward Audio Beehive Monitoring: Deep Learning vs. Standard Machine Learning in Classifying Beehive Audio Samples. *Applied Sciences*, 8(9):1573, September 2018.

[7] Tymoteusz Cejrowski, Julian Szymański, Higinio Mora, and David Gil. Detection of the Bee Queen Presence Using Sound Analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10752 LNAI, pages 297–306, 2018.

[8] Inês Nolasco, Alessandro Terenzi, Stefania Cecchi, Simone Orcioni, Helen L Bear, and Emmanouil Benetos. Audio-based identification of beehive states. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8256–8260. IEEE, 2019.

[9] I. Nolasco and E. Benetos. To bee or not to bee: Investigating machine learning approaches for beehive sound recognition. *in 2018 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018.

[10] R. Serizel V. Bisot and S. Essid. Feature learning with matrix factorization applied to acoustic scene classification. *IEEE ACM Transactions on Audio, Speech and Language Processing,vol. 25, no.*, (6):1216–1229, 2017.

[11] D. Fourer, J-L. Rouas, P. Hanna, and M. Robine. Automatic timbre classification of ethnomusicological audio recordings. In *Proc. ISMIR*, Taipei, Taiwan, October 2014.

[12] I. Nolasco and E. Benetos. To bee or not to bee: an annotated dataset for beehive sound recognition. *Dataset documentation*, August 2018.

[13] P. Rao. *Speech, Audio, Image and Biomedical Signal Processing using Neural Networks, (Eds.)*. Springer-Verlag, 2007.

[14] Vibha Tiwari. Mfcc and its applications in speaker recognition. *International journal on emerging technologies*, 1(1):19–22, 2010.

[15] Antonio Robles-Guerrero, Tonatiuh Saucedo-Anaya, Efrén González-Ramírez, and José Ismael De la Rosa-Vargas. Analysis of a multiclass classification problem by lasso logistic regression and singular value decomposition to identify sound patterns in queenless bee colonies. *Computers and Electronics in Agriculture*, 159:69–74, 2019.

[16] Abhishek Thakur Rajeev Ranjan. Analysis of feature extraction techniques for speech recognition system. *International Journal of Innovative Technology and Exploring Engineering (IJITEE))*, 8, May. 2019.

[17] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25, 2015.

[18] Nasser Kehtarnavaz. *Digital Signal Processing System Design (Second Edition)*. ScienceDirect, 2008.

[19] Dominique Fourer Sarra Houidi and François Auger. On the use of concentrated time-frequency representations as input to a deep convolutional neural network: Application to non intrusive load monitoring. *entropy*, 2020.

[20] G. Peeters, B. Giordano, P. Susini, N. Misdariis, and S. McAdams. The timbre toolbox: Audio descriptors of musical signals. *Journal of Acoustic Society of America (JASA)*, 5(130):2902–2916, Nov. 2011.

[21] E. Schubert, J. Wolfe, and A. Tarnopolsky. Spectral centroid and timbre in complex, multiple instrumental textures. In *Proc. 8th Int. Conf. on Music Perception & Cognition (ICMPC)*, Evanston, Aug. 2004.

[22] B.C.J. Moore and B.R. Glasberg. Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. *Journal of the Acoustical Society of America*, 74:750–753, 1983.

[23] E.Ambikairajah, J. Epps, and L. Lin. Wideband speech and audio coding using gammatone filter banks. In *Proc. IEEE ICASSP'01*, pages 773–776, 2001.

[24] G. Torelli and G. Caironi. New polyphonic sound generator chip with integrated microprocessor-programmable adsr envelope shaper. *IEEE Trans. on Consumer Electronics*, CE-29(3):203–212, 1983.

[25] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[26] Emon Md Mohaiminul Islama. An overview of artificial neural network. *American Journal of Computer Sciences and Applications (ISSN:2575-775X)*, 2019.

[27] VALENTINA E. BALAS MARIUS-CONSTANTIN POPESCU and al. Multilayer perceptron and neural networks.

[28] et al. Y. LeCun. Handwritten digit recognition with a back-propagation network? in advances in neural information processing systems. pages 396–404, 1990.

[29] R. Hecht-Nielsen. Theory of the backpropagation neural network, in neural networks for perception. *ed: Elsevier*, pages 65–693, 1992.

[30] Nazib Ahmed Shadman Sakib and al. An overview of convolutional neural network: Its architecture and applications. 2018.

[31] Jason Brownlee. Data preparation for machine learning data cleaning, feature selection, and data transforms in python. (398 pages).

[32] Saúl Solorio-Fernández1 · J. Ariel Carrasco-Ochoa1 · José Fco. Martínez-Trinidad1. A review of unsupervised feature selection methods. *Springer Nature B.V*, 42, 2019.

[33] Thomas JA Cover TM. Elements of information theory. 2nd edn. wiley-interscience, new jersey. *Springer-Verlag London 2013*, 2006.

[34] Jorge R. Vergara • Pablo A. Estévez. A review of feature selection methods based on mutual information. *Springer-Verlag London 2013*, 2013.

[35] J. K. Kalitab N. Hoquea, D. K. Bhattacharyyaa. Mifs-nd: A mutual information-based feature selection method.