IDIAP COMMUNICATION REPORT

ON LEARNING TO CLASSIFY MEERKAT CALLS

Imen Ben Mahmoud

Idiap-Com-01-2024

MAY 2024

Centre du Parc, Centre du Parc, Rue Marconi 19, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

On Learning to Classify Meerkat Calls

Thesis performed in the Speech and Audio Processing Research Group at the Idiap Research Institute in the context of the UniDistance Artificial Intelligence Program to obtain the degree of Master of Science in Artificial Intelligence

Student :	Imen BEN MAHMOUD
Student number :	15-817-554
Project supervisor :	Dr.Mathew Magimai Doss
Company supervisor :	Alexandre Nanchen

Master's Thesis presented on

July 11, 2023

Martigny, Idiap Research Institute

Acknowledgements

First and foremost, I would like to express my gratitude to my project supervisor, Dr. Mathew Magimai Doss, for enabling me to embark on this exciting project. His guidance, support, and expertise have been invaluable in ensuring the successful completion of this thesis. I am deeply grateful for his trust in me, as well as for the countless ideas and teachings he has generously shared over the course of the past year and a half.

I would also like to extend my appreciation to Alexandre Nanchen, my company supervisor, for their valuable insights and thoughtful suggestions. Their input have significantly enriched the quality of this work.

Furthermore, I would like to acknowledge the collaboration and support of my colleague and office roommate, Eklavya Sarkar. Their constructive criticism, insightful discussions, and willingness to share their knowledge have greatly contributed to the progress and success of this research project. Also, thank you to Dr. Marta Manser and Isabel Driscoll , who graciously shared their data with us. I am thankful for their willingness to collaborate.

Lastly, my appreciation goes to all the individuals who have provided their support in various ways throughout my master's journey. To my classmates, friends, family, and colleagues, thank you all for your support!

Lausanne, February 15, 2024 I. BM

Abstract

This thesis focuses on the classification of meerkat vocalizations using machine learning techniques. Meerkats have a complex social structure and a diverse communication system. They communicate through vocalizations, also called calls, which serve multifaceted purposes and can be classified into distinct categories. Even though several studies have been published analyzing their intriguing acoustic signals, the task of classification of the calls has been barely explored so far. It is a task that is still manually performed by human experts, hence the need for a computational method.

This research explores feature extraction methods and classification algorithms to effectively categorize meerkat call types utilizing three distinct datasets. The experimental results demonstrate the effectiveness of the support vector machine algorithm applied to convolution neural network crafted features, achieving the best performance across the three datasets.

Furthermore, the thesis explores the generalization capability of a trained end-to-end convolution neural network, emphasizing the importance of re-adapting the filter stage of the convolutional neural network to the new dataset for improved classification performance.

Keywords: Bioacoustics, CNN, meerkats, calls classification.

Contents

List of Figures

List of Tables

Acronyms

ANN Artificial Neural Network

CNN Convolutional Neural Network

DFT Discrete Fourier Transform

DL Deep Learning

eGeMAPS Emotion and Gender-Related Acoustic Features

HCTSA Highly Comparative Time-Series Analysis

HMM Hidden Markov Model

LLD Low-Lever Descriptors

MFCC Mel Frequency Cepstral Coefficients

ML Machine Learning

MLP Multilayer Perceptron

RBF Radial Basis Function

ReLU Rectified Linear Unit

RF Random Forest

RNN Recurrent Neural Network

RvNN Recursive Neural Network

SSL Self Supervised Learning

STFT Short Time Fourier Transform

SVM Support Vector Machine

t-SNE t-distributed Stochastic Neighbor Embedding

UAR Unweighted Average Recall

UMAP Uniform Manifold Approximation and Projection

1 Introduction

In recent decades, extensive research on human speech has been conducted to drive advancements in various tasks, such as automatic speech recognition, speech synthesis, and speaker recognition. This exploration has also sparked curiosity about non-human speech. Researchers are showing a growing interest in elucidating the mechanisms underlying the acoustic signals emitted by animals. They seek to understand how these sounds are generated and the underlying reasons behind their production. Bioacoustics, the study of sound produced by living organisms, has become interdisciplinary, and it does not only combine acoustics and biology anymore but also ecology, zoology, and other expertise. By understanding the language of animal sounds, researchers have gained valuable insights into animal behavior and evolution. They could use them to assess an ecosystem's health. Moreover, breakthroughs in the study of animal language could help us understand our human language since we hypothesize that human and animal signals share many commonalities. One animal that has garnered considerable attention is the meerkat, alternatively called *Suricata suricatta*.

Meerkats are small-bodied carnivores belonging to the mongoose family. They spend most of their active time during the day foraging, generally with other group members. However, they rarely share prey with other adult individuals. They are fascinating subjects to study thanks to the following:

- **Complex social structure**: They live in cooperative groups and work together for everyday tasks such as foraging, babysitting, or sentinel duty.
- **Adaptability**: They have been adapted to live in harsh environments and can cope with high temperatures, water missing, and resource availability.
- **Complex communication system**: They have a diversified vocal repertoire that enables them to communicate various messages within their group.

This last point is what interest us the most. These organisms exhibit a diverse repertoire of vocalizations that serve varied functions. Given our project's focus, we will explore prior related work in the following section.

1.1 Prior work

These last twenty years have seen an improvement in the knowledge of the communication system of meerkats. Most of the research was done at the Kalahari Meerkat Project based at the Kuruman River Reserve in the southern part of the Kalahari in South Africa [Clutton-Brock et al., 1999]. This research concentrated on decoding the context of a call. For example, in [Manser et al., 2002], it is shown that the meerkat alarm call simultaneously encodes information about both predator type and the signaler's perception of urgency. Alternatively, in [Gall and Manser, 2018], the paper demonstrates that the spatial structure of foraging meerkat groups is influenced by social factors such as affiliation and aggression among group members, as well as predation risk and foraging success, indicating that meerkats adjust their location within the group based on both the current social dynamics and the physical environment. These studies concentrated on understanding these small animals' vocal, social, and communicative complexity.

Unfortunately, a surprisingly limited amount of research has been published on the application of machine learning or deep learning to address what has yet to be discovered concerning these animals. Regarding the classification problem of these calls, in [De Luca et al., 2022], machine learning methods are used and compared for the vocal identification of meerkats. Indeed, in this case, the author classified individuals not calls. In [Thomas et al., 2022], the authors employed a dataset containing annotated vocalizations of meerkats to showcase the effectiveness of a particular method to underline patterns and categorize these sounds into distinct groups. The method uses neighborhood-based dimensionality reduction of spectrograms to produce a latent space representation of calls and evaluate how accurately it represents the recognized taxonomy of different call types. The Uniform Manifold Approximation and Projection (UMAP) approach of [Sainburg et al., 2020] used in the paper encodes the vocalizations as row-wise concatenated spectrograms and then applies the unsupervised dimensionality reduction.

Besides these last papers mentioned, the literature does not mention other studies where a set of features could help differentiate between the calls or an algorithm giving accurate classification results.

Considering this information, we will rely on transfer learning to achieve our objectives. Transfer learning is an approach in Machine Learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. In this case, segmenting and classifying speech is a problem prevalent in the speech processing domain. We are already comfortable processing human speech. However, animal vocalization is different and less known to us. There is still a lack of prior knowledge lying in an animal sound, and one should be careful about the method applied to analyze the audio. Most of the

time, these methods have been developed to analyze human sound and frequencies. They should be adapted if we are considering other sounds. However, if the transfer learning is successful, if we achieve satisfying results, the similarities shared by the human speech and animal sound, meerkats, in this case, should be carefully considered in future research.

1.2 Motivation

The analysis and classification of animal vocalizations is a laborious task that heavily depends on the expertise of human listeners. Even among these experts, there is often a lack of consensus in the annotations. Therefore, there is a pressing need for computational methods to address the classification challenge more objectively and quantifiable. While methods exist for analyzing vocalizations of certain animal species, the development of a reliable method tailored explicitly for meerkats calls is currently lacking. This highlights the need to address the unique characteristics and complexities of meerkats' vocalizations. We would like to provide a novel approach to enhance our comprehension of meerkat communication patterns and advance our understanding of animal acoustic communication systems. This project seeks to contribute to the broader field of animal vocalization analysis.

Thus, for this task, this thesis takes some distance from current state-of-the-art systems in speech processing and develops call classification systems. To do so, we use findings in ML, showing that relevant features and classifiers can be learned directly from raw signal [Palaz et al., 2019]. Nevertheless, we also explore the transferability of feature extraction and classification methods between human speech and bioacoustics.

1.3 Objectives

The goals of this thesis:

- Explore different ML approaches to categorize call-types of meerkats.
- Investigate the feature extraction methods utilized.
- Assess the generalization of one of the methods to another dataset.

To summarize, our objective for this thesis is to explore feature extraction methods and classification methods for meerkats call-type classification. We will use three different datasets for this exploration and compare the results. The approach for the classification is either machine learning or deep learning oriented. Since we are provided with three different sets of inputs, we would like to assess the generalization of the deep learning approach between the datasets.

1.4 Collaboration

In the pursuit of advancing the field of meerkat call analysis, this research project was conducted in collaboration with Martha Manser, Full Professor in Animal Behaviour at the Department of Evolutionary Biology and Environmental Studies, University of Zurich. This partnership emerged within the scope of the Swiss National Science Foundation's (SNSF) National Centre of Competence in Research (NCCR) Evolving Language, a nationwide project that brings together experts from diverse disciplines, including humanities, linguistics, and social sciences, to deepen our understanding of the evolution of language. The NCCR aims to investigate the nature of language, including its development in our species, as well as the process of transmitting new linguistic variations to future generations.

This work was shaped through meetings and discussions to identify the specific needs related to the study of meerkats. Additionally, data sharing played a critical role in this collaboration, allowing us to access and utilize valuable data for meerkat analysis.

1.5 Outline

Chapter 2, *Background*, defines speech and, more specifically, meerkat calls. It gives an overview of features extraction and classification method in bioacoustics. It also describes which methods will be used during the project.

Chapter 3, *Datasets*, describes the datasets provided during the thesis.

Chapter 4, *In-Dataset Study*, will better explain the methodology of the experiments and gives the results of the features and methods selected.

Chapter 5, *Data Visualization*, will concentrate on the feature sets selected and their visualization in a low dimension feature space.

Chapter 6, *Cross-Dataset Generalization*, will evaluate the performance of a model trained on one dataset and applied to another.

2 Background

This chapter is primarily theoretical and will introduce human and non-human speech, explicitly focusing on meerkat vocalizations. It will present classification methods and feature extraction approaches applied to bioacoustics. Additionally, we will discuss the methods utilized during the project.

Bioacoustics is defined as the study of sounds produced by animals. However, it is essential to note that in this thesis, we expressly exclude human speech when referring to bioacoustics. The same applies when mentioning animal sounds.

2.1 Voice

2.1.1 Human speech

Speech is defined as the most common form of communication by humans. Human sound is produced through a complex process but can be simplified. As air is exhaled from the lungs, it passes through the vocal cords. It creates vibration, hence sound. This vibration generates a fundamental frequency ($F0$), the rate at which the vocal cords open and close. The sound waves then travel to the vocal tract, where its shape and configuration, including the position of the tongue, lips, jaw, and other parameters, modify the sound. These adjustments enhance or attenuate specific frequencies, giving rise to different sounds or phonemes. It is to be noted that the rate of the periodic vibration of vocal cords differs from one person to another. For some sounds, there is no periodic vibration.

A phoneme is the smallest sound unit that distinguishes the meaning of words. Its number differs from one language to another, grouped as vowels and consonants. These units are combined to create meaningful words and phrases to create a human language and express a limitless amount of information. In the study of non-human animal sounds, we mainly talk about vocalizations: any sound or utterance produced by an organism using its vocal

apparatus, typically involving the vocal cords or other sound-producing structures. In the case of humans, vocalizations include spoken language, singing, shouting, laughter, crying, and various vocal expressions used to convey thoughts, emotions, and intentions or to engage in social interactions.

In the animal world, vocalizations play a significant role in communication across the environment and are essential for survival. Like human speech, animal vocalizations can provide listeners with information about objects and events in the environment, therefore information that is both semantic and emotional Seyfarth and Cheney [2003]. The following section will dive further into that notion.

2.1.2 Non-human animal vocalization

Animal vocalizations are complex. Characterizing these vocalizations requires effort and a deep understanding of the species vocal as the analysis method is often species-specific. They can carry important information and are subject to evolutionary pressures, resulting in the development of unique vocal repertoires and communication systems in different species. Some evident animal vocalizations that we may hear in our everyday life are, for example:

Insect chirping : Although we associate the sounds of cicadas callings and crickets chirping with the beginning of summer, these sounds are most of the time mechanisms of defense, made in response to disturbance or even mating calls. The patterns and frequencies of chirps can vary between species and carry specific information. Insect chirping, known as stridulation, involves rubbing one body part against another. Different body parts are involved for other species, from wings to the abdomen. These chirpings serve various purposes, including warning signals, territorial defense, and mating calls to attract females.

Birds sounds : Another animal that we hear communicating everyday are birds. The sounds produced can be categorized into two different classes: songs and calls. Their songs are a remarkable aspect of avian behavior and communication. They are long and complex vocalizations produced during the breeding season. The smallest unit in this sound is an element or a note. A collection of them creates a syllable; a series of syllables is a phrase. In turn, songs are organized into several phrases. This structure is demonstrated in Fig. 2.1.

Many bird species have a repertoire of songs, and young birds acquire songs from their parents and other adult birds in their environment. These songs play a crucial role in species recognition by researchers. Males often use their songs to establish and defend their territories and attract females. Females evaluate the quality of a potential mate based on the quality and complexity of his song.

Bird calls differ from songs in the function, used to express alarm or distress, indicate the

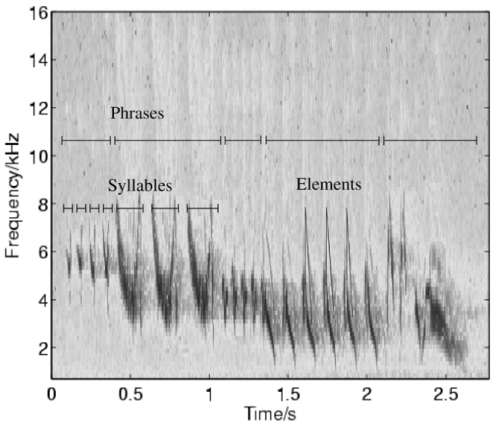


Figure 2.1: Hierarchical levels of song of the Common Chaffinch [Somervuo et al., 2006]

presence of predators, and signal territorial boundaries. However, they are generally shorter and more straightforward than songs, consisting of brief, distinct notes or phrases. Moreover, can be distinguished from songs by a specific pattern or rhythm. Each bird specie has its repertoire of calls that are unique to that specie. These calls help individuals recognize and communicate with members of their specie while distinguishing them from other bird species in the environment.

Mammalian calls : Roars, growls, barks, howls, and whistles are many examples of mammal calls. The use of the call depends on the situation of the mammal. Some mammals, like dolphins and whales, are known for their complex vocal repertoires and the ability to learn new vocalizations. In other cases, mammalian calls may be instinctual and not subject to learning. They can convey emotions such as fear, distress, happiness, etc. As for their interaction with humans, some of them developed specific vocalizations to communicate with them, such as dogs and cats. These domesticated animals can bark, meow, and purr, which humans have learned to interpret this behavior.

The common point between these vocalizations, independent of the species, is their role in the social dynamics, survival strategies, and expression of emotions and intentions. As exploring the mechanisms and purposes of communication between individuals has captivated scientists for many years, this pursuit is crucial for understanding the dynamics of animal societies and the evolutionary foundations of human language, our vocal communication system. Researchers have employed three primary approaches to enhance our understanding of animal vocal communication, and these approaches have significantly contributed to advancing the knowledge in this area [Garcia and Favaro, 2017] :

- First of all, it is crucial to understand the context of the vocalization and to describe it. Classification methods have been used to classify calls into vocal repertoires. However,

the limit to applying a human-based evaluation of vocal repertoire exists. The acoustic structure of vocalization is shaped by factors, mainly environmental and type of interaction. As these factors influence the vocal repertoire, they can also create differences in repertoires between different species. One important step in understanding vocal behavior is to investigate the acoustic structure of vocalizations and how external factors shape it.

- The second approach is to investigate the function of these animal vocalizations, to understand the behavioral context in which signals are used. Particularly the relationship between the acoustic structure of signals and their purpose.

- As the last approach to carrying out studies on animal vocal communication, one needs to understand and look for the production mechanisms involved. Despite understanding the sound-producing organs, in-depth studies regarding the anatomical and physiological determinants of vocal features is necessary.

The following section will focus on meerkats and introduce these small and endearing creatures.

2.1.3 Meerkat vocalization

Since the project focused primarily on meerkat vocalizations, it is essential to discuss the communication behavior theory of this species. Meerkats, scientifically known as Suricata suricatta, are small, highly social animals predominantly found in arid regions of South Africa [Ross-Gillespie and Griffin, 2007]. They are mammals with an average length of approximately 35 cm, excluding their tails, Fig. 2.2. There is a complex social structure within meerkat groups, featuring a dominant breeding pair and cooperative behaviors such as group defense, foraging, and babysitting. Meerkats dig safe places called bolt-holes throughout their foraging areas, which serve as hiding places during emergencies. Communication among meerkats occurs through various vocalizations, including barks, chirps, trills, and growls. These vocalizations are essential in coordinating group activities, warning of potential dangers, and maintaining social cohesion. Researchers have identified and classified around 30 vocalization types in meerkats [Townsend et al., 2014a]. Notably, there are several important categories of meerkat calls:

- **Alarm Calls**: Emitted when a potential predator is encountered, they can vary depending on the type of predator encountered, aerial or ground threat.

- **Contact Calls**: Meerkats use contact calls to maintain group cohesion and communicate with individuals dispersed over a larger area. These calls help clan members stay in contact while foraging or exploring. Contact calls are short, high-pitched chirps or trills that carry well in open habitats.

Figure 2.2: A meerkat gang is enjoying the sunlight. [Takahashi, 2018]

- **Begging Calls**: Pups produce distinct begging calls to solicit food from adult group members. These calls are often characterized by high-pitched, repetitive sounds and play an important role in the provisioning and care of the young.
- **Dominance Calls**: Individuals employ dominance calls during conflicts or to assert social hierarchy within the group. These calls are typically low-pitched growls or harsh barks and help to establish and maintain the dominant relationships among individuals.

They can produce additional vocalizations to express excitement, playfulness, or warnings to group members. They have gained popularity due to their charming appearance and cooperative behaviors. Their social dynamics and fascinating adaptations make them an interesting subject of study for researchers and wildlife enthusiasts alike. The next chapter will provide more information about the calls that will be classified (Chapter 3).

Since the thesis focuses on the classification of meerkat vocalizations, the following section will discuss the methodologies currently employed for feature extraction and classification of animal vocalizations.

2.2 Features extraction and classification methods in the literature

Classifying animal sounds can be challenging due to the many different types of sounds that span orders of magnitude along the dimensions of time, frequency, and amplitude, whether by the same specie or from one animal to another.

The categorization of acoustic repertoires allows for various applications, including the recognition of species, age, gender, and individual characteristics. As technological advancements have progressed, classification methods have also advanced. In the past, detection and classi-

fication tasks were performed by experienced bio-acousticians who listened to the sounds and visually reviewed spectrographic displays (e.g., [Gannon and Lawlor, 1989] and [Stafford et al., 1999]). Indeed, spectrograms, visual representations of sound frequency over time, were visually reviewed. They would look for specific frequency ranges, temporal patterns, harmonics, or other distinguishing elements that could indicate the presence of a certain animal. However, it was time-consuming and limited by the availability of skilled experts. The goal was back then to automate this process of detecting the sound of interest and classifying it. These techniques involve three mains steps :

1. Detection of potential sounds of interest
2. Extraction of relevant acoustic characteristics (or features) from these sounds
3. Classification of these sounds into the categories of interest.

As for detecting an animal sound in audio, one of the first and most used methods relied on energy or amplitude thresholding. It consists of measuring an incoming signal's amplitude or energy and determining if it exceeds a certain user-defined threshold. As in the example in [Ou et al., 2012] where boing whale sounds are detected by using a waveform envelope detector. Suppose a portion of an envelope is above a threshold for over half a second. In that case, it is identified as boing sounds.

Once the voice activity detection method is applied, the classification of the sound into the decided categories relies on the measured features of the sound. The feature extraction step refers to selecting, transforming, or representing certain characteristics or patterns of the data relevant to the task. These features serve as informative data representations and can be extracted from waveforms, spectrograms, and others. Based on these attributes, classification algorithms categorize data into predefined classes.

2.2.1 Features

In the classification process, whether it's an image or a waveform, the algorithm searches for specific characteristics or properties that can aid in distinguishing between classes. This is also the objective of the feature extraction phase. It involves transforming raw data into a suitable representation.

The feature selection step holds significant importance as it can determine the success or failure of the classification method. With the availability of modern software tools, an unlimited number of features can be easily measured, facilitating differentiation between various types of sounds. Here are several types of features that can be extracted from a signal and their applications:

Temporal features: Temporal features are also called time domain features. Simple to

extract, they include time-varying behavior of the signal such as zero-crossing rate, energy envelope, temporal centroid, etc. These features are appropriate if the temporal dynamics are crucial for distinguishing between classes. However, it is also interesting to incorporate additional features, such as spectral features, to better accomplish the task. As we can see in [Zamanian and Pourghassem, 2017], where cicadas species are identified using spectral and temporal features, or in [Kottege et al., 2012], where extracting spectral and temporal features from empirical fish sounds, including the tilapia specie in Australia enables the authors to classify this last specie from the other fishes.

Spectral features: The spectral features, called frequency-based features, are obtained by converting the time-based signal into the frequency domain using the Fourier Transform. Some features are fundamental frequency, frequency components, a sound's starting and ending frequencies, minimum and maximum frequencies present, peak energy frequency, bandwidth, spectral centroid, etc. These features are values extracted from spectrograms. To construct a spectrum, the magnitudes of Short Time Fourier Transform (STFT) frames of audio, often around 10 ms duration per frame, are plotted. The spectrum obtained through STFT represents the energy distribution over a linear range of frequencies. Examples in the literature are [Wu et al., 2022] or [Kershenbaum et al., 2016]

Cepstral Features: To extract cepstral features, one needs to be in the cepstral domain. This refers to a mathematical transformation from the time domain: The computation of the Fourier Transform, the application of the logarithm, and the subsequent application of the Discrete Cosine Transform. One of the main reasons to do cepstral analysis is that the logarithmic compression applied in the cepstral domain mimics the logarithmic perception of loudness by the human auditory system. Therefore this domain aligns better with the perception of sound by the human ear. This transformation not only aligns with human perception but also reduces the dimensionality of the spectrum. In this domain, Mel Frequency Cepstral Coefficients (MFCC) is the type of features commonly extracted and used. To obtain the coefficients, the frequency spectrum is passed through a series of triangular filters spaced on the Mel scale before applying the logarithm. It resembles the human auditory system better than the linearly-spaced frequency bands of the normal cepstrum. The coefficient is then extracted after applying the discrete cosine transform. Typically, the lower-order coefficients carry most of the energy and represent the overall spectral shape. In contrast, the higher-order coefficients capture fine spectral details. The number of coefficients to retain is often application-dependent and can vary. MFCC are widely used in speech and audio signals processing tasks such as speech recognition, speaker identification, music genre classification, and sound event detection. These features are commonly used in human speech. However, it has also been applied to many different animal species. For example, in [Clemins et al., 2005] to automatically classify African elephant vocalizations or in [Ramirez et al., 2018] for an automatic bird species recognition system.

Once an appropriate feature extraction method that suits the task and data has been selected, the subsequent step entails selecting the suitable classification method.

2.2.2 Classification methods

The choice of a classifier depends on the specific problem, the nature of the data, the interpretability requirements, and the performance goals. There are different types of classifiers. They can be probabilistic-based or deterministic-based, non-linear or linear, and model-based or instance-based classifiers. Let us explore the ones that have been given sufficient results applied to bio-acoustics.
Hidden Markov Model (HMM) is a statistical model widely used in various fields, including speech recognition. A system is modeled by hidden states that generate observable outputs. In classification, an HMM for each class is created by defining: the hidden states, observable outputs, transition probabilities, and emission probabilities. They have been used to classify the sounds produced by birds [Adi et al., 2010], dolphins [Datta and Sturtivant, 2002], and even elephants [Clemins et al., 2005].
Classification tree analysis, a non-parametric statistical technique, partitions data into nodes groups through binary splits based on specific variables. These splits are guided by primary splitting rules, aiming to create increasingly homogeneous nodes. The growth of the tree continues until perfectly homogeneous nodes are achieved. Subsequently, the tree is pruned by eliminating nodes and evaluating the error rates of resulting smaller trees. The optimal tree, offering the highest predictive accuracy, is selected based on this. RF is a collection of many (hundreds or thousands) individual classification trees, which are grown without pruning. Each tree is trained on a random subset of the features. Each tree independently generates its output during prediction, and the final prediction is determined by majority voting or averaging. It is a commonly-used machine learning algorithm. The algorithm can be used in speech processing for various tasks, including speech recognition, speaker identification, emotion detection, phoneme classification, or language identification. In bioacoustics, some examples are in [Armitage and Ober, 2010] for bat echolocation signals detection or in [Ross and Allen, 2014] for detecting passerine flight calls at night.
Another method used in a considerate way to solve classification tasks, and that should be mentioned, is SVM. It is one of the most popular supervised learning algorithms used for classification. The goal of the algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category. This best decision boundary is called a hyperplane. This hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points from each class. The larger the margin, the better the generalization ability of the model. In [Raju et al., 2012], it was used to classify animal sounds, in [Fagerlund, 2007] for bird species recognition, and in [Zeppelzauer et al., 2015] to recognize elephant's rumbles from background noise.

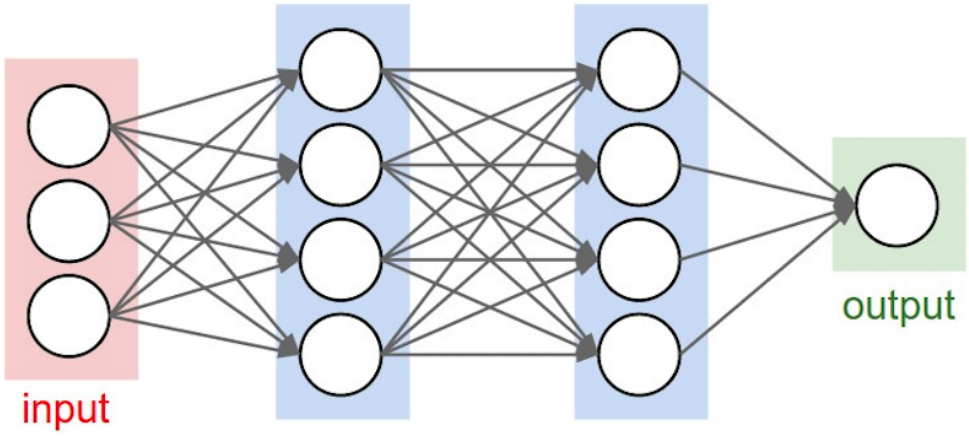


Figure 2.3: Diagram of the structure of an artificial neural network[1].

Bioacoustics has greatly advanced through the application of computational analysis methods, including signal processing, data mining, and machine learning. One significant breakthrough in the field of machine learning is Deep Learning (DL), which has revolutionized various computational disciplines. With the advent of digital recording devices, vast amounts of audio data can now be captured and processed, creating opportunities for DL algorithms to analyze and extract meaningful information from bioacoustic signals. These techniques enable researchers to address complex tasks that were previously considered difficult or impossible to tackle. From the detection of Hainan gibbon calls [Dufourq et al., 2021] to the automatic classification of frogs by processing their calls [Colonna et al., 2016], it is safe to say that the integration of DL into computational bioacoustics represents a significant leap forward, empowering researchers to explore new frontiers. In a few words, DL is a sub-field of machine learning that focuses on training deep neural networks with multiple layers to extract high-level representations and learn complex patterns from data. Artificial Neural Network (ANN) serve as the building blocks of these deep neural networks. It mimics the way nerve cells work in the human brain. All ANNs are composed of units called neurons and connections among them. They typically consist of three or more neuron layers: one input layer, one output layer, and one or more hidden layers. See Fig. 2.3 for representation. The input layer consists of several neurons corresponding to the dimension of the input of the neural network. The output layer consists of several neurons representing the number of classes. The selection of the number of hidden layers and neurons per layer in the neural network is typically determined through empirical exploration by the researcher. Each connection between neurons in the network is assigned a weight value, which undergoes iterative adjustments during training.

Depending of the architecture, there are multiple types of DL neural network : Recurrent Neural Network (RNN) [Hewamalage et al., 2021], CNN [Krizhevsky et al., 2017], Recursive Neural Network (RvNN) [Socher et al., 2013] and etc.

[1] https://www.eponamind.com/fr/neural-networks-deep-learning/

With an understanding of the available techniques in the literature, we will highlight the specific approach adopted in this project. This includes the methods selected for extracting relevant features from the data and the classification algorithms employed for the subsequent analysis.

2.3 Features extraction approach

After investigating the existing literature, a definitive choice has been made regarding selecting features to extract and the classification algorithms to employ. It should be noted that specific methodologies have not been previously employed in the analysis of bioacoustic data. This brings us to the notion of transfer learning, where the potential application of these methods in animal vocalization is explored. The objective is to evaluate whether the shared characteristics between human speech and animal vocalization are sufficient for achieving comparable performance in analyzing animal sounds as they do in human speech.

2.3.1 eGeMAPS

These features are extracted using the software OpenSmile [Eyben et al., 2010]. Emotion and Gender-Related Acoustic Features (eGeMAPS) [Eyben et al., 2016] is a set of acoustic features commonly used in the field of speech analysis and emotion recognition [Bao et al., 2023].

These features capture various acoustic characteristics of speech signals relevant to emotion and gender-related information. Until now, it does not seem that their efficiency has been tested on bioacoustic data. The features set include 88 different features. The features contain the following set of Low-Lever Descriptors (LLD) :

- Frequency-related parameters (8)
- Energy/Amplitude related parameters (3)
- Spectral (balance) parameters (14)

The arithmetic mean, and coefficient of variation are applied as functionals to those 25 LLD, yielding 50 parameters. Eight functionals are applied to a parameter of amplitude and frequency-related parameters, and the mean of four other parameters is calculated. That gives a total of 70 parameters. Six temporal features are then included :

- Rate of loudness peaks.
- Mean length and standard deviation of voiced regions.
- Mean length and standard deviation of unvoiced regions.
- Number of continuous voices regions per second.

With other functionals on other parameters, we arrive at 88 features.

2.3.2 ComParE

These features also have been extracted using OpenSmile software. It is the official baseline feature set of the INTERSPEECH Computational Paralinguistics Challenge [Schuller et al., 2016]. The features set include 6373 different parameters. The type of LLD used are :

- Energy-related LLDs (4)
- Spectral LLDs (54)
- Voicing related LLDs (6)

Multiple functionals are then applied to reach the final number of features.

2.3.3 catch22

Highly Comparative Time-Series Analysis (HCTSA) [Fulcher et al., 2013] are comprehensive features designed for analyzing and characterizing time series data. It extracts thousands of features of the group, such as auto-regressive modeling, frequency domain, wavelet-based, and many more. This extraction is computationally intensive and involves assessing numerous similar features. In [Phaniraj et al., 2022], by the use of some features from the set, the authors were able to correctly identify the source individual with high precision of marmoset calls. However, how to decide which features to select from this set? In [Lubba et al., 2019], a method to infer a small set of time series features is introduced. Only the 22 features that exhibit strong classification performance across a given collection of time-series problems and are minimally redundant are kept. pycatch22 - CAnonical Time-series CHaracteristics in python [Lubba et al., 2019] is used to extract this features. The mean and standard deviation of the features are added, and 24 features in total will be calculated.

2.3.4 CNN crafted features-embeddings

One of the methods of classification that will be tested is feeding the waveform of the signal to a CNN. More clarification about the model will be given in the next section. Since It is the knowledge that the early layer of a CNN captures low-level features such as edges and corners, while deeper layers learn more complex features and semantic information, we decide to extract these features. Indeed, the output of the second to last layer of this CNN is taken as a feature set after feeding it the waveform. The final feature set for every file is 80 features.

2.4 Classification methods approach

The selection process for determining the appropriate classification methods was a challenging task. After careful consideration, the final decision was reached, favoring a combination of baseline techniques commonly employed for classification tasks, namely RF and SVM, along with the inclusion of CNN.

2.4.1 Support Vector Machines

As already mentioned, SVM is a supervised ML algorithm used for classification and regression tasks. It works by finding an optimal hyperplane that maximally separates the data points of different classes in a high-dimensional feature space.

SVM can handle linearly separable data as well as non-linearly separable data by using a technique called the kernel trick. The kernel trick allows the algorithm to implicitly map the input features into a higher-dimensional space, where the data becomes linearly separable. This enables SVMs to solve complex classification problems that may not have a linear decision boundary in the original feature space. The following parameters are essential for the algorithm and will be fined tuned to search for the best accuracy :

- Kernel: The function of this parameter is to take the data as input and transform it into the hyperplane determined by the mathematical function of the kernel. The kernels tested in our case are linear, polynomial, Radial Basis Function (RBF), and sigmoid. Each kernel has its own set of parameters that need to be tuned.
- Regularization parameter (C): C is a crucial parameter that controls the trade-off between achieving a large margin and minimizing the training errors. If C is small, the penalty for misclassified points is low, so a decision boundary with a large margin is chosen at the expense of a greater number of misclassifications. Conversely, if C is large, the algorithm tries to minimize the number of misclassified.
- Gamma (γ): This kernel coefficient is used for sigmoid, RBF, and polynomial. It controls the distance of the influence of a single training point. Low gamma values ($0.008 - 0.01$) indicate a large similarity radius, resulting in more points being grouped. For high gamma values ($3.0 - 11.0$), the points must be very close to each other to be considered in the same class.

2.4.2 Random Forest

RF is based on the concept of decision trees and combines multiple decision trees to make predictions. A random subset of the original training data is selected for each tree in the forest through bootstrap sampling. This means that each tree is trained on a slightly different set of data, which introduces diversity into the forest. At each decision tree node, a random

subset of features is considered for determining the best split. This means that each tree only considers a subset of features, reducing the likelihood of individual features dominating the decision-making process. The tree is constructed by repeatedly splitting the data based on different features and thresholds to minimize impurities in the resulting subsets. Once all the trees are built, predictions are made by combining the outputs of individual trees through a voting mechanism. In classification tasks, the class that receives the most votes becomes the predicted class. The following parameters are important for the algorithm and will be fined tuned to search for the best accuracy :

- Number of estimators: The number of decision trees in the random forest. Increasing the number of estimators typically improves the performance of the random forest up to a certain point. However, having too many estimators can increase computational complexity and training time.
- Maximum depth: A decision tree grows by recursively splitting the data based on features and thresholds until a stopping criterion, which can be the maximum depth, is met. A more considerable maximum depth allows the trees to capture more complex patterns in the data, but it can also lead to overfitting. It is essential to tune this parameter carefully to avoid overfitting while allowing the trees to capture meaningful relationships in the data.
- Minimum samples split: The minimum samples split parameter sets the minimum number of samples required to split an internal node further in the decision tree. If the number of samples at a node is less than the specified minimum, the node becomes a leaf node, and further splitting is stopped. This parameter helps control the tree's depth and prevents it from splitting too much into regions with insufficient data.
- Minimum samples leaf: The minimum samples leaf parameter specifies the minimum number of samples required to be present in a leaf node. If a split would result in a leaf node with fewer samples than the specified minimum, that split is not performed. Similar to the minimum samples split parameter, this parameter helps control the size and depth of the decision tree and prevent overfitting.

These parameters are crucial for tuning the Random Forest model for optimal performance and generalization.

2.4.3 Convolution Neural Network

Regarding using deep learning for bioacoustics, CNN is now the dominant architecture [Stowell, 2021]. Researchers either use off-the-shelf CNN architecture: ImageNet in [Lasseck, 2018], AlexNet in [Guyot et al., 2021], and SincNet in [Bravo et al., 2021]. Others use self-designed architecture [Wang et al., 2021], [Li et al., 2020] and [Zualkernan et al., 2020]. In our case, we chose to design and test our CNN architecture.

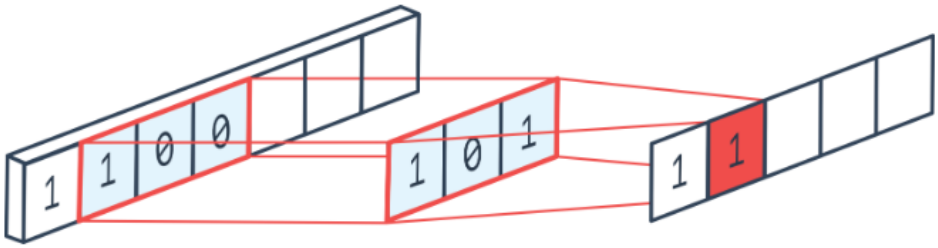


Figure 2.4: Example of a convolution operation [2]

As mentioned before, a CNN is a neural network that uses a convolution layer to filter and break down the input signal. The convolutional layer is the key component of a CNN. It consists of a set of learnable filters (also called kernels) that slide over the input, performing convolution operations. In Fig. 2.4, the operation with a 1D input is represented, with the middle list being a filter. It involves element-wise multiplication of the filter with a local region of the input, followed by summation, resulting in a single value known as the convolutional feature or activation. Multiple convolutional features are generated by sliding the filters across the entire input, creating feature maps that capture different aspects of the input.

Picking the right values for the hyperparameters of a CNN is crucial for achieving good performance. Following are some important hyperparameters :

- Learning Rate: The learning rate determines the step size at each iteration during training. It controls how much the model's parameters are updated. A learning rate that is too high can cause the model to converge quickly but may result in overshooting the optimal solution. On the other hand, a learning rate that is too low may cause slow convergence or the model to get stuck in suboptimal solutions.
- Number of Layers: A CNN's depth or the number of layers is a vital hyperparameter. Deeper networks can capture more complex patterns and features. However, they may require more computational resources and be prone to overfitting if the dataset is small. Finding the right balance between model complexity and available resources is important.
- Pooling Strategy: Pooling layers reduce the spatial dimensions of the input, helping to extract invariant features and reduce computational complexity.
- Activation Functions: Activation functions introduce non-linearities in the CNN, enabling it to model complex relationships. ReLU is widely used due to its simplicity and effectiveness in preventing vanishing gradients. We decide on that function.
- Batch Size and Number of Epochs: The batch size determines the number of samples processed before updating the model's parameters. A smaller batch size introduces

[2] https://ai.stackexchange.com/questions/28767/what-does-channel-mean-in-the-case-of-an-1d-convolution

more noise but may help the model converge faster. The number of epochs determines the number of times the entire dataset is passed through during training.

- Number of Filters and Filter Size: The number of filters in each convolutional layer and their size determine the receptive field of the network and the level of abstraction it can capture. It is this hyperparameter optimization that we will concentrate on the most. It is challenging since they will not always have the same size as a waveform.

To summarize the CNN, an activation function is applied element-wise to introduce non-linearity into the network after each convolution operation. Commonly used activation functions include Rectified Linear Unit (ReLU). Another important layer is the pooling layer, used to downsample the spatial dimensions of the feature maps while retaining the essential information. Following the convolutional and pooling layers, fully connected layers are employed. These layers are similar to those in traditional neural networks. They are responsible for capturing high-level representations and making predictions. Each neuron in a fully connected layer is connected to all neurons in the previous layer, allowing the network to learn complex relationships between features. The output of the last fully connected layer is the prediction of our model.

In Section 4.2.2, we will provide comprehensive information regarding the architecture selected for our model and the specific values assigned to the hyperparameters.

2.5 Summary

In this chapter, we explore the crucial steps involved in bioacoustic data analysis, focusing on feature extraction and classification methods. In order to distinguish between different sound categories, feature extraction plays a significant role in transforming raw data into informative representations. We discussed various types of features, including temporal, spectral, and cepstral features, highlighting their applications in different bioacoustic studies. Furthermore, we cited a few classification methods such as Hidden Markov Models, Random Forests, Support Vector Machines, and Deep Learning. These methods are commonly employed in bioacoustics. They offer diverse approaches to categorizing and analyzing bioacoustic signals, each with its strengths and limitations. Additionally, we introduced specific feature extraction approaches for the thesis, including eGeMAPS, ComParE, catch22, and CNN-crafted features, along with the selected classification algorithms: CNN, RF, and SVM.

3 Datasets

For this study, we were provided with three distinct datasets comprising meerkat vocalizations. Although all three datasets consist of the same animal sounds, they possess different labels, were collected from different locations, and represent diverse animal situations.

3.1 Set A

The first dataset shared with us was by Dr.Marta Manser, Professor in Animal Behaviour, Department of Evolutionary Biology and Environmental Studies, University of Zurich, Switzerland. It consists of 9 different folders of 9 types of meerkats calls. We define these nine types as our classes. There is a total of 90 audio files for a total duration of 1605.4s.

The audio duration ranges from a minimum of 204.3ms to a maximum of 79171.3ms. A file was not a single vocalization but rather, most of the time, a repetition of a call. For the study, every file was manually segmented using Koe [Fukuzawa et al., 2020], a web-based software to classify acoustic units and analyze sequence structure in animal vocalizations. Indeed, the silences were taken out, and only the single calls were kept. An example of how this pre-processing method is done is presented in Fig. 3.1.

In that figure, we went from an audio file containing three vocalizations separated by silence to three segments. This segmentation was manually performed. In the end, we found a total of 1795 signal files with a sampling rate of 44100Hz, belonging to 9 different categories of calls. In this case, to be noted that during the segmentation part, if a call were of duration inferior to 100ms, samples of the waveform were repeated until reaching 100ms.

The nine different labels of the calls are :

- **aggr-aggression :** These sounds are made during territorial disputes or conflicts within the group.
- **sen-sentinel :** A sentinel is an individual responsible for keeping watch for potential

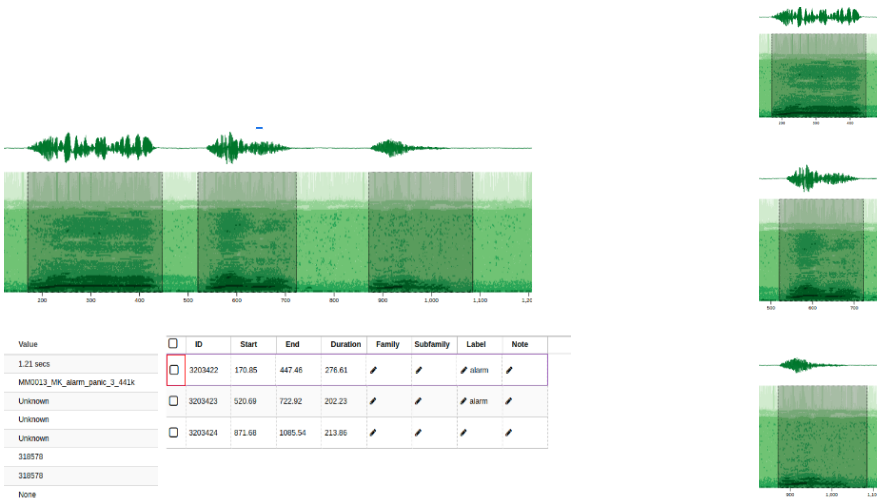


Figure 3.1: Example of use of Koe software for pre-processing of audio.

predators. In contrast, others forage or engage in other activities.

- ⋆ **al-alarm :** A call to alert members of their group of potential danger or threat.
- ⋆ **ch-chatter :** Observed when the individuals are excited and playful.
- ⋆ **gr-grooming :** Sounds are made when one cleans another individual fur.
- ⋆ **cc-close call :** Calls are emitted when close to each other.
- ⋆ **sub-submission :** Emitted when a submission behavior is displayed.
- ⋆ **ld-lead :** Used to guide other members of the group during foraging or movement.
- ⋆ **su-sunning :** Meerkats engage in sunning behavior, basking in the sun to warm themselves up. During this behavior, they communicate their relaxed state.

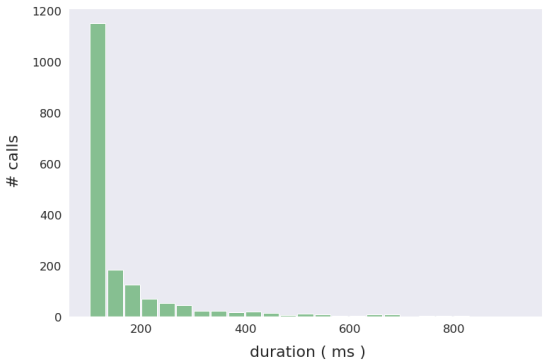
Our goal is to successfully classify these calls into their nine respective categories. The alarm calls are separated into subcategories: Aerial predator or terrestrial predator. But in our study, we will use it as only one category.

After exploration of these final segments, some statistics can be extracted, presented in Table 3.1 and Fig. 3.2 shows the distribution of vocalizations and the distribution of the duration of calls.

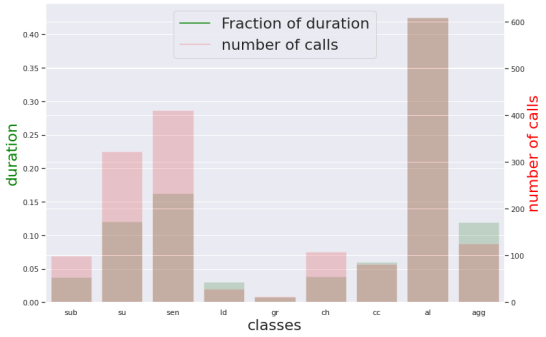
The maximum duration close to 1s concerns only one audio, accompanied by a few calls that exceed 600ms. However, as depicted in Fig. 3.2a, a significant proportion of the calls fall within the range of 100ms to 200ms. This observation is further supported by the median of 101.3ms, indicating that there is a likelihood that a substantial portion of the calls had durations less than 100ms. In Fig. 3.2b, the proportion of duration per each class is represented on one axis,

minimum duration	100ms
maximum duration	930ms
mean	161.4ms
median	101.3ms

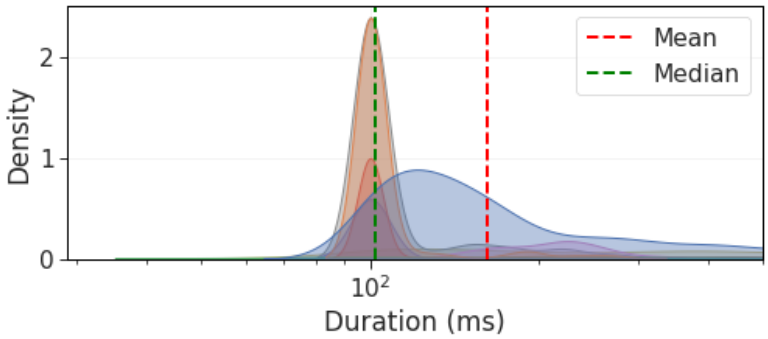
Table 3.1: Statistics on Set A



(a) Call count function of the duration in ms.



(b) distribution of the proportion of the total duration by category of call



(c) distribution of the duration of the calls

Figure 3.2: Study on the Set A of calls.

and the corresponding number of calls on the other axis. Notably, the alarm class exhibits the highest proportion, with more than 40 % of the total duration, encompassing more than 600 files. The grooming class represents the lowest proportion, constituting less than 5% of the duration and a very low number of files. When comparing the alarm class to the sentinel class, it can be inferred that the duration of sentinel calls is relatively shorter. With approximately 16% of the total duration distributed among 400 files, the sentinel class indicates a noticeable difference in the duration of individual calls. This difference in the duration of single calls is further depicted in Fig. 3.2c, which highlights that certain classes display a wider distribution with more occurrences of longer calls. In contrast, others exhibit a narrower distribution centered around the median duration. It is crucial to emphasize that these figures reveal a significant imbalance within the dataset, mirroring the real-world scenario.

3.2 Set B

We were given a second set of meerkat calls, previously utilized in the study referenced as [Thomas et al., 2022]. This dataset consists of 6,428 individual files, sampled at 48,000 Hz. The calls in this dataset are already categorized into seven distinct call types. Overall, the dataset spans a total duration of 954 seconds. It is to be noted that in this set, every file is already a single vocalization. Before analyzing the distribution of the calls, similar to the previous section, let us define the call types present. Four classes are already defined for Set A: aggression, alarm, close call, and lead. This dataset includes the following additional call types:

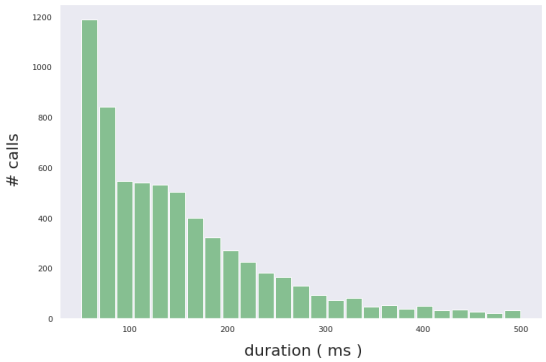
- **mo-move :** used to try and initiate departure from the current location of the group.
- **soc-social call :** produced in a case where several members are reunited, and no threat is around.
- **sn-short note:** These calls are used in different contexts while running or foraging; however, in sequences and very short.

Once again, some statistics are extracted, presented in Table 3.2, and Fig. 3.3 shows the distribution of vocalizations and of the duration of calls for this set.

minimum	50ms
maximum	500ms
mean	148.4ms
median	124ms

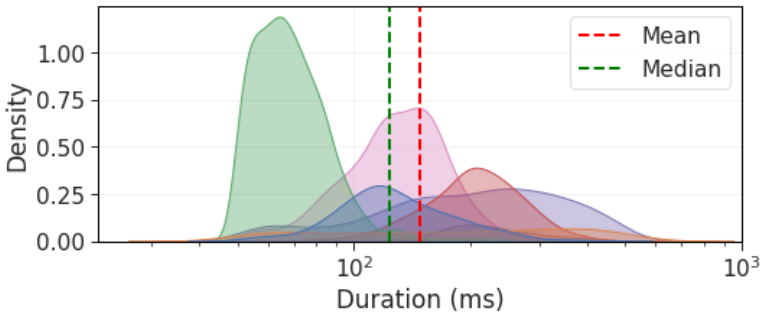
Table 3.2: Statistics on Set B

Fig. 3.3a demonstrates that the duration of calls in this dataset exhibits a broader distribution compared to Set A. A significant proportion of calls fall within the range of less than 200ms,



(a) Call count function of the duration inms

(b) distribution of the proportion of total duration by category of call.



(c) distribution of the duration of the calls

Figure 3.3: Study on the Set B of calls

minimum	30.5ms
maximum	724.4ms
mean	119.5ms
median	116.1ms

Table 3.3: Statistics on Set C

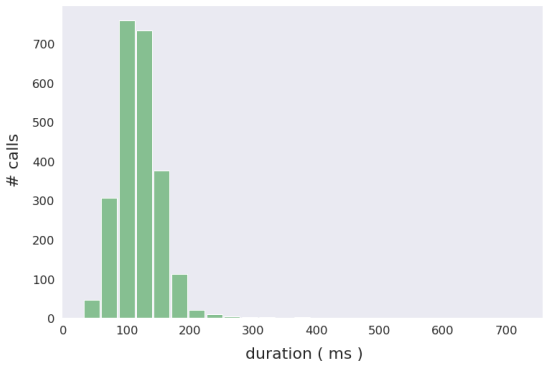
with a minimum duration of 50ms and a maximum duration of 500ms. However, this dataset shows a notable increase in the number of calls within the 200ms to 500ms duration window compared to Set A. Similarly, an imbalance is observed among the classes, with more calls in the short note, social, and close call categories. Conversely, the lead class is the least represented. Nonetheless, given this set's larger number of files, it still provides a sufficient quantity for training a model. Fig. 3.3c highlights that this dataset's distribution is more diverse than the first set of data. It reveals the presence of call duration occurring both before and after the mean and median values and some distribution intersecting with these values.

3.3 Set C

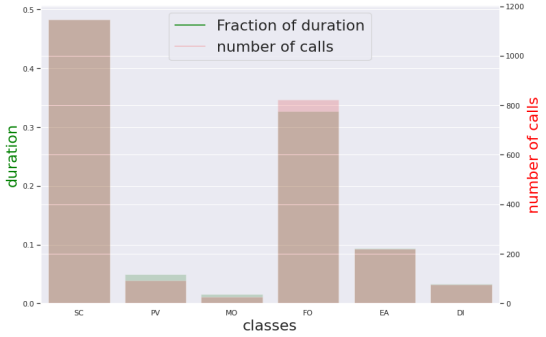
Driscoll Isabel, Ph.D. student in Communication and Cognition in Social Mammals, Department of Evolutionary Biology and Environmental Studies, University of Zurich, provided this set. The data collection occurred in 2022 at the Kalahari Meerkat Project in South Africa [Clutton-Brock et al., 1999]. During the acquisition, clan members were closely monitored, and their vocalizations were recorded. This particular dataset focuses on the behavioral context of close calls. Close calls are vocalizations emitted by meerkats when they are in proximity to each other. However, these calls can occur in various contexts. In this dataset, the close calls have been categorized into six different contexts:

- **SC-Scrabbling :** Walking and scratching the surface.
- **FO-Foraging :** Focused surface digging.
- **DI-Digging :** Digging using both paws.
- **EA-Eating :** Eating prey.
- **PV-Post-vigilance :** Given after a period of vigilance
- **MO-Moving :** Walking while not showing other call behavior.

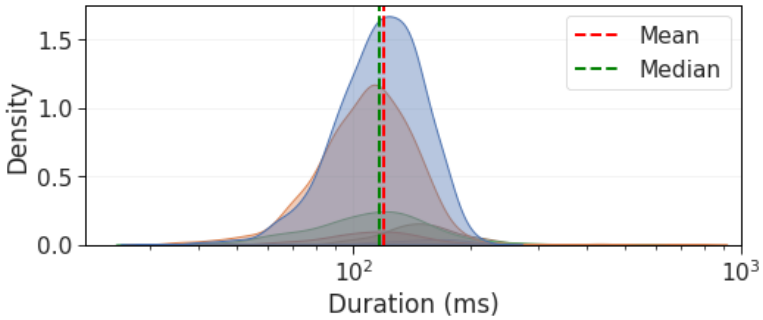
The total duration of this dataset is approximately 284 s for 2381 files. They are single vocalizations with a sampling rate of 48000 Hz. Once again, some statistics are extracted, presented in Table 3.3, and Fig. 3.4 shows the distribution of vocalizations and the distribution of the duration of calls for this set of calls.



(a) Call count function of the duration inms



(b) distribution of the proportion of total duration by category of call



(c) distribution of the duration of the calls

Figure 3.4: Study on the Set C of calls

Fig. 3.4a clearly illustrates that a substantial portion of the calls occurs between 30ms and 200ms, as evidenced by the median value slightly exceeding 100ms. Additionally, the significant class imbalance observed in Fig. 3.4c is worth noting. Specifically, the duration of the scrabbling class constitutes roughly half of the total duration across 1200 calls, accounting for more than half of the total number of calls. In contrast, the lowest class comprises only approximately 1% of the total duration and involves around 20 files. In the end, two classes, scrambling, and foraging, constitute more than 80% of the total duration and 90% of the total number of files.

3.4 Summary

This chapter introduces three distinct datasets comprising meerkat vocalizations used in the study. The first dataset, Set A, contains 9 types of meerkat calls. An imbalance exists among the classes, with alarm being the most prevalent. The second dataset, Set B, consists of audio files separated into 7 distinct call types. The dataset displays a wider distribution of call durations compared to Set A. An imbalance is observed among the classes, but not as significant as it is in Set A. The third dataset, Set C, focuses on close calls in different behavioral contexts. The audio files capture six different close calls contexts, such as foraging, digging, and eating. It is crucial to emphasize that Set C differs from Sets A and B. The primary objective of studying Set C is to investigate the possibility of achieving pure context distinction.

Overall, these datasets provide valuable resources for studying meerkat vocalizations and aim to classify calls into categories. However, they exhibit class distribution imbalances, mirroring real-world scenarios. These datasets will be used for training and evaluating machine learning models in subsequent chapters.

4 In-Dataset Study

This chapter aims to provide a comprehensive overview of this study's experimental setup and methodology. We will delve into the details of our experimental configuration and explain the process of tuning the models to determine optimal hyperparameter values. Additionally, we will define the performance metric used to compare the different experiments. Finally, we will present the results obtained from the experiments and engage in a comprehensive discussion to interpret and analyze the outcomes.

In this chapter, our objective is to evaluate the effectiveness of machine learning methods and feature extraction techniques that have shown promising results in classifying human speech. We aim to determine whether these approaches can yield favorable outcomes when applied to a previously unexplored task of classifying meerkat calls. By conducting this investigation, we aim to assess the transferability and generalizability of these methods to bioacoustics.

4.1 Methodology

During the study, two approaches are followed.

The first approach involves extracting features from the waveform data and utilizing either a RF or SVM algorithm. As depicted in Fig. 4.1, it follows a sequential process.

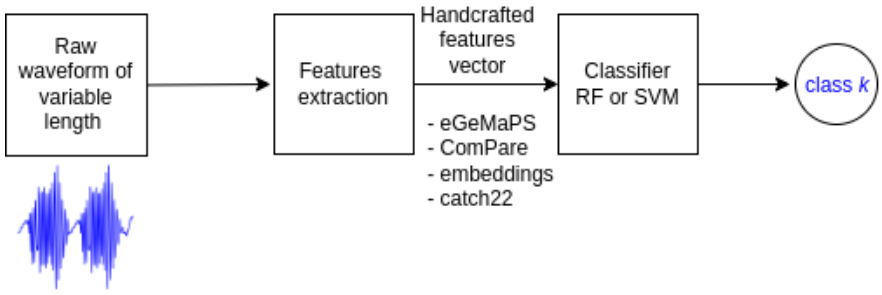


Figure 4.1: First set of experiments pipeline

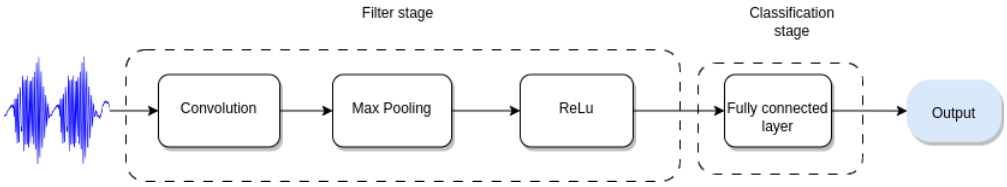


Figure 4.2: Overview of CNN architecture

The second approach entails directly feeding the waveform data into a neural network by employing an end-to-end CNN architecture. End-to-end is an approach where the model is trained to perform a task directly from raw input. In our case, the raw input is the raw waveform. This approach was originally proposed for automatic speech recognition in [Palaz et al., 2013] and developed further in [Palaz et al., 2019]. As illustrated in Fig. 4.2, the model consists of a feature stage that is repeated, followed by a classification stage. We have three convolution layers followed by a fully connected layer, also referred to as Multilayer Perceptron (MLP). Each convolution layer is composed of 3 operations: convolution, max-pooling, and a ReLU activation function. It is important to note that the features map obtained at the end of the filter stage is the feature vector used in Fig. 4.1.

However, we cannot employ exactly the same model and architecture defined in [Palaz et al., 2019]. Indeed, in his model, Palaz uses a fixed length input of 250ms. Compared to our variable size input, with a minimum of 100ms, we encounter *size too small* warnings. Hyperparameters such as kernel width and kernel shift need to be adapted to our study. Before extracting and analyzing the results of the two approaches, it is essential to optimize the algorithms by fine-tuning the parameters mentioned in the previous chapter. This optimization step allows us to search for the values of the hyperparameters that will yield the best performance in our classification task. By carefully adjusting these parameters, we aim to enhance the model's effectiveness and achieve optimal results.

4.2 Experimental set-up

Given that the two approaches diverge, we proceed to explicate each experiment individually, elucidating the manner in which distinct outcomes were attained. However, it is important to note that in both approaches, the audio file was subjected to resampling at a frequency of 16000 Hz. Furthermore, if the temporal extent of the audio file fell short of 100ms, it was systematically replicated until the wanted duration was achieved.

4.2.1 First approach: machine learning methods

In order to address potential issues that may arise when conducting a standard train/validation/test split for model training and testing, particularly when working with a relatively small dataset and imbalanced like ours, it becomes crucial to devise an appropriate train-test split

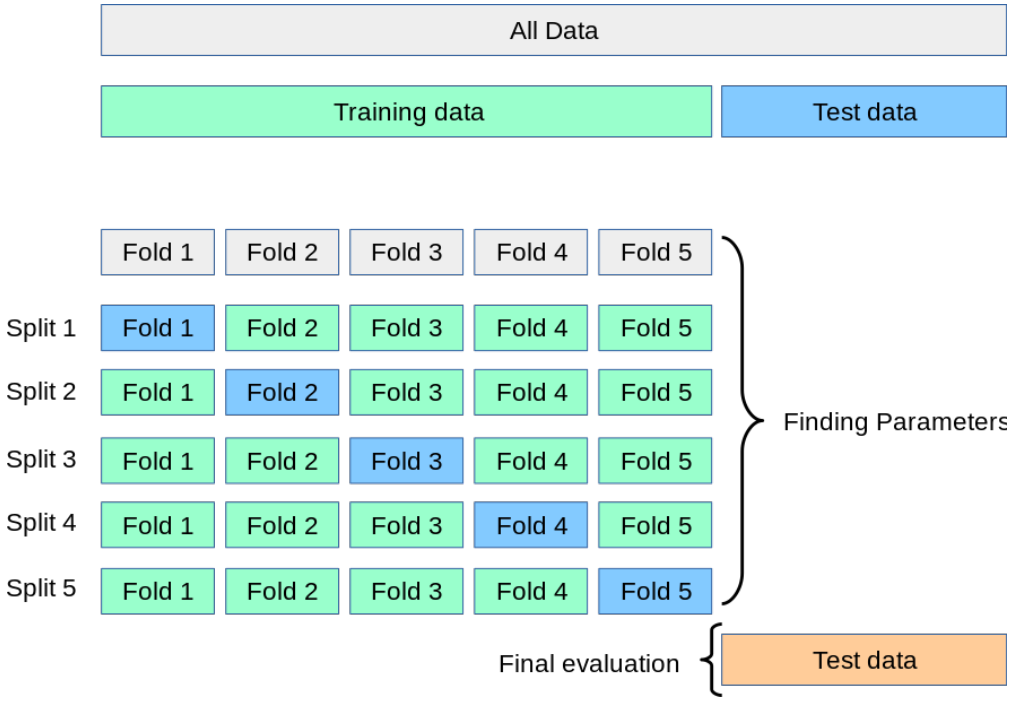


Figure 4.3: Scheme of how k-fold cross validation works [1]

procedure. One effective solution to this problem is the utilization of k-fold cross-validation.

Cross-validation involves dividing our dataset into random groups or folds, with one fold designated as the test set while the model is trained on the remaining folds. This process is repeated multiple times, with each fold being held out as the test set in turn. By doing so, we can obtain a more comprehensive evaluation of our model's performance.

For our specific task, we have opted to split the data into five folds, as illustrated in Fig. 4.3. This figure depicts the schematic representation of how k-fold cross-validation works. It showcases the iterative process of training and testing the model on different folds of the dataset.

In our project, we employ a combination of k-fold cross-validation and grid search for the RF and SVM. Grid search is a popular hyperparameter optimization technique utilized in machine learning to determine the most effective combination of hyperparameter values for a given model. These hyperparameters are set prior to the training process and can significantly impact the performance and generalization of the model.

The grid search technique involves defining a grid of hyperparameter values to explore. For each combination of values in the grid, the model is trained and evaluated using a predefined evaluation metric, as specified Section 4.2.3. This process systematically assesses the performance of the model across all possible hyperparameter combinations. The goal is to identify the combination that yields the highest performance on the evaluation metric, which

[1] https://scikit-learn.org/stable/modules/cross_validation.html

Algorithm	Parameters	Grid search values
RF	Number of estimators	[10, 20, 30, 40, 50, 70, 80, 100, 150, 200]
	Maximum depth	[5, 7, 10]
	Minimum samples split	[2, 3, 5, 7, 10]
	Minimum samples leaf	[1, 2, 3, 4]
SVM	C	[0.1, 1, 10, 100]
	γ	[0.001, 0.01, 0.1, 1]
	Kernel	Linear, RBF, polynomial, sigmoid

Table 4.1: Grid search parameters and value for SVM and RF.

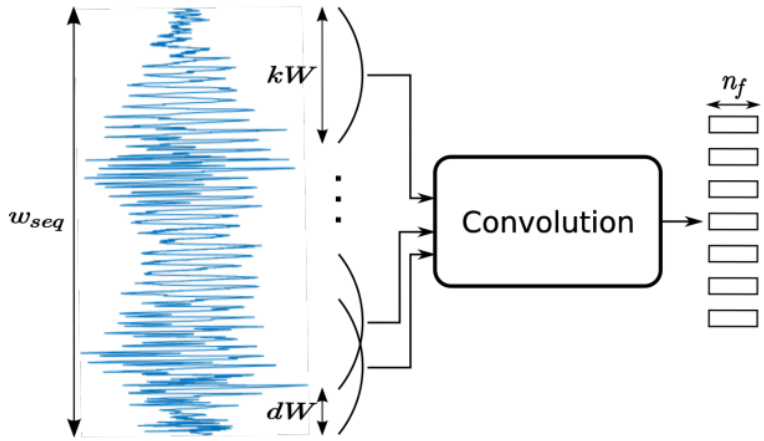


Figure 4.4: First convolution layer process [2]

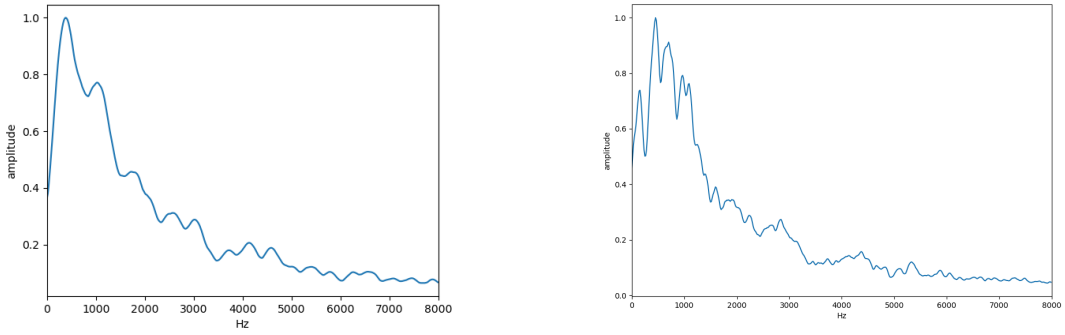
is then selected as the optimal set of hyperparameters. It is with that combination that we will evaluate the performance of the algorithm on a test set taken out before the k-fold following an 80 : 20 split.

4.2.2 End-to-End CNN Approach

As mentioned previously, we will not use exactly the same architecture as in [Palaz et al., 2019]. We will use Set A for the optimization of the model and generalize to the other sets.

The main parameter that will be adapted to our task is the number of filters, their width, and the filter shift. The first convolution layer plays a crucial role in the processing of the input data. In Fig. 4.4, we provide an illustration of the operation. For each convolution layer i, we define a kernel width kW_i, a kernel shift dW_i, and a number of filters n_{f_i}. In addition, there is the max-pooling size and the number of hidden units in the MLP.

[2] https://infoscience.epfl.ch/record/270134?ln=fr



(a) $kW_1 = 40$ (b) $kW_1 = 120$

Figure 4.5: Cumulative frequency responses of first layer filters, trained on the Set A

The fundamental frequency, also known as $F0$, refers to the lowest frequency component of a complex sound or waveform. By taking into account this definition, intuitively, we can suppose that the kernel size of the convolution layer should not be smaller than the $F0$ and should ideally be equal to or a multiple of the $F0$. However, meerkats produce noisy barks, and researchers have had difficulties until now seeing a clear fundamental frequency and harmonic structure [Townsend et al., 2014b]. If we want to apply that hypothesis to our problem, we will need to set an empirical $F0$ value. To do so, we observe several spectrograms of the different calls and suppose a value with the naked eye. We chose a value of 400 Hz. In a number of samples, this corresponds to 40 samples (16000 as sampling rate).

To test our choice of this value, we analyze what information is modeled by the first convolution layer for two different values of kernel width of that convolution layer. The method consists in computing the cumulative frequency response of the learned filters, similar to [Palaz et al., 2019]:

$$F_{cum} = \sum_{k=1}^{n_{f_1}} \mathscr{F}_k \tag{4.1}$$

Where $n_{f_1} = 40$ is the number of filters for the first convolution layer. $\mathscr{F}_\|$ is the magnitude spectrum of filter f_k computed with a 1024-point Discrete Fourier Transform (DFT). The cumulative frequency response indicates which frequency regions the filters focus on. In Fig. 4.5, we observe the cumulative frequency response of the filters of the first convolution layer of the CNN trained on the Set A, with a kernel size kW_1 of 40 and 120 samples. in Appendix chapter is presented the frequency response with Set B and Set C.

In both cases, we see that the filters give emphasis to the information lying between $0-2000$ Hz. Since there is no difference in the information that is learned between both cases, we set the kernel width of the first convolution layer at 40.

The remaining values were chosen through empirical selection while ensuring convenience for the shortest waveform. The filter stage architecture of the neural network is depicted in

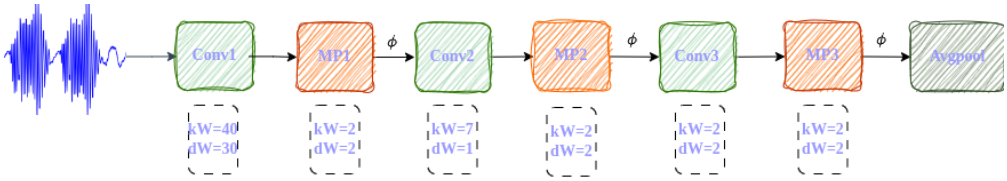


Figure 4.6: Architecture of the filter stage of the convolution neural network model

Fig. 4.6, where ϕ denotes the ReLU activation function, $Conv_i$ represents a convolution layer, and MP_i signifies a maximum pooling layer. To account for varying input sizes, an average pooling layer is added after the third convolution block, ensuring uniform feature map sizes before feeding them into the classification stage. Additional details about the model include the utilization of the Adam optimizer, cross-entropy as the loss function, and a learning rate of 0.003.

As previously mentioned, we employ a modified form of cross-validation called Stratified K-Folds cross-validator. This method constructs folds while preserving the percentage of samples for each class. By incorporating stratified sampling in cross-validation, we maintain the same class proportions in the training and test sets as in the original dataset. For instance, if the al class calls represent 40% of the total calls in the original dataset, the test set will also contain 40% of that class, and the same applies to the training set. In this case, a 5-fold approach is utilized, initially employing an 80 : 20 train-test split. Subsequently, a random 80 : 20 train-validation split is reapplied.

4.2.3 Performance Metric: UAR

By taking into account the imbalance of the data, using a simple accuracy as a performance metric does not seem relevant. After some research, we decided to use the UAR as a metric. It is defined as the average of recall obtained in each class. Recall is calculated as the ratio of true positives (TP) to the sum of true positives and false negatives (FN):

$$Recall = \frac{TP}{TP + FM} \tag{4.2}$$

The recall is also known as sensitivity or true positive rate. It provides insights into how well a model captures the positive instances, minimizing the number of false negatives. A higher recall value indicates a better ability to correctly identify positive instances, while a lower value indicates a higher number of missed positive instances. Table 4.2 is an example of a confusion matrix.

After calculating the recall of each class and averaging, $UAR = 0.75$ and $accuracy = 0.84$. Even though the accuracy value is better, the UAR is taking into account the low number of true positives observed for Class 1 and Class 4.

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Recall
Class 1	43	3	3	0	1	3	22	0.57
Class 2	0	108	3	2	7	8	1	0.84
Class 3	1	6	269	1	5	8	5	0.91
Class 4	0	1	4	16	11	0	1	0.48
Class 5	0	14	3	11	118	5	1	0.78
Class 6	0	6	5	0	1	359	0	0.97
Class 7	20	5	17	7	5	10	167	0.72

Table 4.2: Example of a confusion matrix with the Recall value as the last column

4.3 Results and Discussion

Now that we have optimized our models for optimal performance and determined all the parameter values, we can proceed with conducting the experiments on the test set. For each Set, we will obtain ten values of UAR.

In the case of the CNN approach, we sum the predictions of all the five folds and calculate the UAR with these values. The results from all the experiments are presented in Table 4.3.

	Set A		Set B		Set C	
	RF	SVM	RF	SVM	RF	SVM
eGeMAPS	0.55	0.61	0.66	0.66	0.24	0.34
ComParE	0.62	0.80	0.65	0.75	0.26	0.34
catch22	0.56	0.61	0.58	0.56	0.31	0.38
Embeddings	0.65	**0.84**	0.76	**0.84**	0.37	**0.58**
CNN average on folds	0.64		0.74		0.32	

Table 4.3: Values of UAR of the experiments for the three Sets

The confusion matrix of these results are presented in the Appendix chapter. We will start by analyzing the results Set by Set.

Set A In Set A, the Support Vector Machine classifier generally outperforms the Random Forest classifier for most feature sets. For the eGeMAPS feature set, SVM performs at 0.61, while RF obtains 0.55. Similarly, for the ComParE feature set, SVM achieves 0.80, whereas RF achieves 0.62. However, for the Embeddings feature set, SVM achieves an impressive 0.84, outperforming RF's 0.65. The CNN approach 0.64, performs competitively with RF but is outperformed by SVM.

Set B In Set B, the performance comparison between SVM and RF is mixed across feature sets. For the eGeMAPS feature set, both classifiers achieve similar performances. SVM performs better with ComParE, and it is RF turn to outperform catch22. For the Embeddings feature set, both SVM and RF achieve high performances, with SVM achieving 0.84 and RF achieving 0.76.

Similar to set A, it is SVM outperforms this time. The CNN approach performs better in Set B than in Set A.

Set C Set C presents lower overall performance compared to Sets A and B. For the eGeMAPS feature set, both SVM and RF classifiers yield relatively low performances, similar to ComParE and catch22. However, for the Embeddings feature set, SVM outperforms RF and the other feature set by achieving 0.58. The CNN approach also demonstrates a low performance among the evaluated classifiers in Set C. The results obtained from the analysis indicate that there may be challenges in separating the classes within the dataset. This implies that the acoustic features alone may not provide clear boundaries or distinctions between the different classes of calls. Despite introducing different behavioral contexts during the formation of the calls, it appears that these contexts did not significantly alter the acoustic characteristics of the calls. Furthermore, the results obtained with the Embeddings approach indicate that there may be separability among the classes. It is possible that the traditional methods used for classification, such as SVM and RF, may not be the most suitable or effective for capturing the unique characteristics of the classes in this particular dataset. Exploring alternative approaches or incorporating complementary data sources could be promising avenues to improve classification performance. Moreover, going deeper into the neural network architecture and exploring advanced architectures or optimization techniques can also contribute to extracting more relevant and discriminative features from the data.

The main distinction between Set A and Set B lies in the number of classes, the number of samples, and the class distribution within the datasets. As discussed in Chapter 3, Set B comprises a larger number of samples and exhibits better class balance than Set A. Consequently, our initial expectation was that Set B would yield superior performance. When considering the CNN approach, we observe that it performs better on Set B compared to Set A.

For the other approaches, the average UAR is 0.66 for Set A and 0.68 for Set B. This suggests that these approaches may exhibit similar performance levels on both sets, with a small difference observed. Overall, the performance discrepancy between Set A and Set B can be attributed to the variations in dataset characteristics, such as class distribution. In addition to the visible dataset characteristics, other underlying factors may influence the performance. Unseen factors, such as the quality of the audio data or the data collection process, could potentially contribute to the variations in performance. The quality of the audio data includes factors like background noise, recording conditions, or recording protocols. If Set B comprises higher-quality audio recordings compared to Set A, it could explain the improved performance observed in Set B.

The Embeddings feature set achieves higher UAR values when classified using both Random Forest and Support Vector Machine algorithms compared to the other feature sets, even when compared to the result obtained from CNN.

This superior performance of the feature set can be attributed to the representation learned by

the CNN model during its training process. As the ultimate layer of the CNN, the Embeddings capture and encode higher-level abstract features relevant to the classification task. This representation potentially contains richer and more discriminative information than the other handcrafted feature sets, which are manually engineered and may not capture all the relevant patterns in the data. This effectiveness highlights the potential of deep learning techniques in automatically extracting and representing informative features from the data, ultimately leading to improved classification accuracy.

Based on the provided results, it is observed that in most cases, the SVM algorithm achieves higher accuracy than the RF algorithm. The performance difference between SVM and RF can be attributed to several factors. Such as if the data is not linearly separable. SVM is particularly effective in that cases. It uses the kernel trick to transform the data into a higher-dimensional feature space where it becomes easier to separate different classes. Alternatively, if the model is too complex, RF can be prone to overfitting. In contrast, SVM promotes better generalization. Finally, the performance of SVM and RF can vary depending on the specific dataset and its characteristics. Different algorithms may excel in different scenarios based on the data's distribution, dimensionality, and inherent structure. Therefore, the relative performance of SVM and RF can differ from one dataset to another.

4.4 Summary

This chapter presents the experimental setup and methodology for evaluating the effectiveness of machine learning methods and feature extraction techniques in classifying meerkat calls. Two approaches were followed: the first involved extracting features and using RF or SVM algorithms, while the second employed an end-to-end CNN architecture.

A k-fold cross-validation technique in combination with grid search was used to optimize the RF and SVM models in the first approach. The hyperparameters were fine-tuned to achieve the highest performance, and the models were evaluated on a separate test set. In the second approach, a CNN architecture was used. Stratified K-Folds cross-validation was employed to maintain class proportions during training and testing. The performance metric used in this study was UAR due to the imbalanced nature of the sets. Out of all these results, the best performing model for the three sets is applying SVM to the CNN-crafted features. It seems that as the ultimate layer of the CNN, the Embeddings feature captures and encodes the most relevant features for this classification task.

5 Data Visualization

In the previous chapter, we examined the performance of different models. We explored various feature extraction methods to determine which yielded the highest performance metrics. However, a question arises: Could we have predicted the outcome of these experiments?

In the context of the CNN approach, it may not be possible. However, when determining which extracted features are most significant, dimensionality reduction techniques can provide valuable insights.

Dimensionality reduction techniques allow us to explore the underlying structure and patterns in the data. By reducing the dimensionality of the feature space, these techniques can potentially reveal essential characteristics that enable better class separation and clustering. Through the reduction, we can assess the effectiveness of different feature sets in capturing relevant information and facilitating distinct class boundaries.

During this chapter, we will apply two of these techniques to our feature sets and verify if what is observed is aligned with the results of Table 4.3.

5.1 Dimensionality Reduction Techniques

The two techniques chosen for dimensionality reduction are t-distributed Stochastic Neighbor Embedding (t-SNE) and UMAP.

5.1.1 t-Distributed Stochastic Neighbor Embedding

In t-SNE [Van der Maaten and Hinton, 2008], the dimension of the dataset is reduced while preserving the local structure and similarities between data points. Each high-dimensional data point is modeled as a probability distribution in the low-dimensional space. The first step of the method is to measure the similarity between pairs of data points using distance metric (e.g., Euclidean distance). The algorithm then converts the similarities into conditional

probabilities representing pairwise similarities between data points. A low-dimensional space, randomly initialized, is created. Data points are assigned initial coordinates in this space. These points are then adjusted to minimize the divergence between the probabilities.
The main parameter of this algorithm is perplexity, determining the radius of attention around data points. A lower perplexity emphasizes local structure and nearby points, while a higher perplexity considers global structure. The recommended perplexity range typically falls between 5 and 50.

5.1.2 Uniform Manifold Approximation and Projection

UMAP [McInnes et al., 2020] is composed of two main steps in this algorithm: graph construction in high dimensions and optimization to find a similar graph in lower dimensions. The algorithm leverages insights from algebraic topology and Riemannian geometry to do so. UMAP creates a weighted graph based on the data's proximity by extending a radius from each point and connecting those points when radius overlap. The edge's weight represents the likelihood that two points are connected.
The critical parameter in this algorithm is the number of nearest neighbors used to construct the high-dimensional graph. A low number focuses on the local structure, while a high number will look at the global structure.

5.2 Comparaison of techniques across features sets

For each set of data, we will systematically employ both techniques while varying the value of the respective parameter, namely perplexity for t-SNE and the number of neighbors for UMAP. We aim to discern any disparities in the separability among the classes with certain feature sets. If a separability exists between the classes, it should be noticed for all the values of perplexity and number of neighbors. Hence our desire to vary these parameters.

5.2.1 Set A

Initiating our analysis with the t-SNE projection:

- eGeMAPS in Fig. 5.1 exhibits sparsity among the points, particularly when utilizing perplexity values ranging from 5 to 20. However, for other perplexity values, the sparsity diminishes, but the presence of well-defined clusters is still not apparent.
- ComParE in Fig. 5.2 we observe abnormal curves without clearly discernible clusters independent of the value of the perplexity
- catch22 in Fig. 5.3 we identify sparse points when using perplexity values between 5 and 20. Subsequently, the distribution takes on a more curved shape, suggesting a different underlying structure.

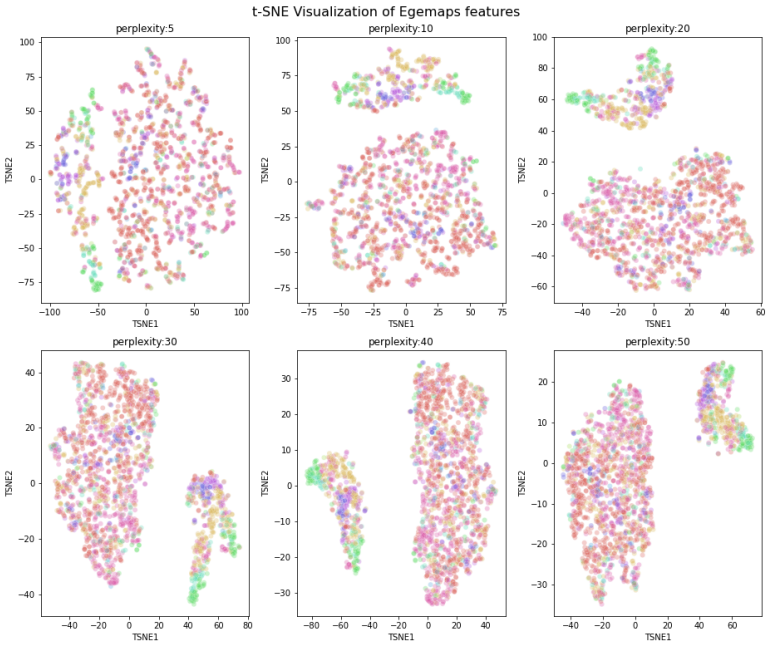


Figure 5.1: Set A - eGeMAPS features set via t-SNE projection

- Embeddings in Fig. 5.4 we notice a resemblance of more well-defined clusters, particularly in the last plot. The clusters become more discernible as the t-SNE perplexity increases but still present for every value of perplexity.

Now, let us examine the UMAP projection to ascertain if the Embeddings feature set continues to demonstrate pronounced cluster formations:

- eGeMAPS in Fig. 5.5 the classes consistently exhibit clustering patterns in two distinct regions of the graph. This suggests a certain level of separability within the data, but not in the desired number of categories.
- ComParE in Fig. 5.6 displays abnormal curves without clearly defined clusters, a result resembling the t-SNE projection.
- catch22 in Fig. 5.7 a curved form is present, although it appears to be less prominent compared to the ComParE feature set.
- Embeddings in Fig. 5.8 continues to exhibit well-defined clusters in the UMAP projection. The classes are more prominently and distinctly visible, reinforcing the notion of high separability within the data.

Regardless of whether the dimensionality reduction technique used was UMAP or t-SNE, it is noteworthy that only the Embeddings feature set demonstrated noticeable clusters among the

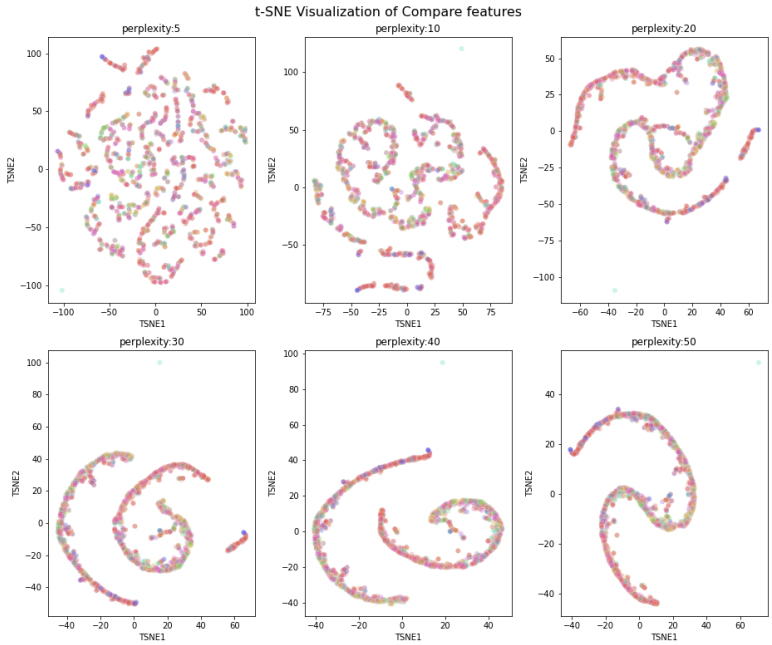


Figure 5.2: Set A - ComParE features set via t-SNE projection

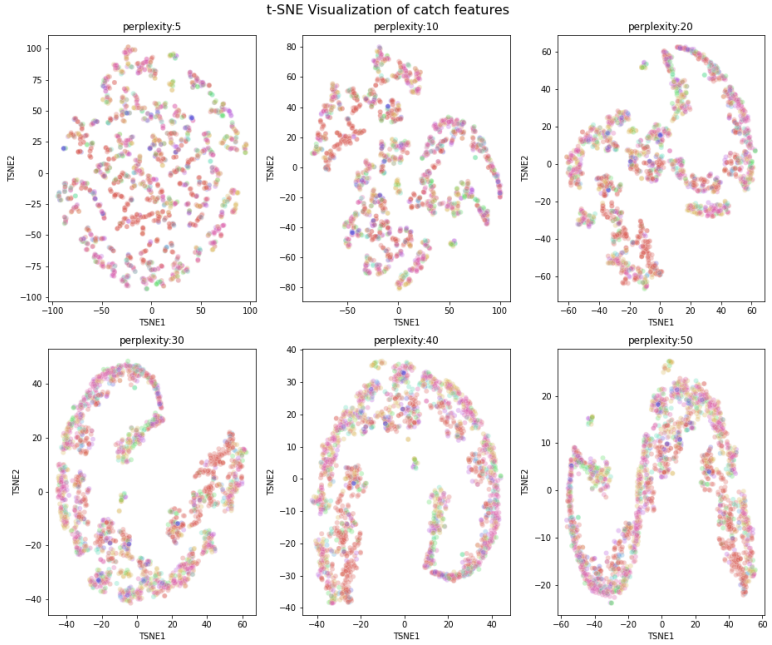


Figure 5.3: Set A - catch22 features set via t-SNE projection

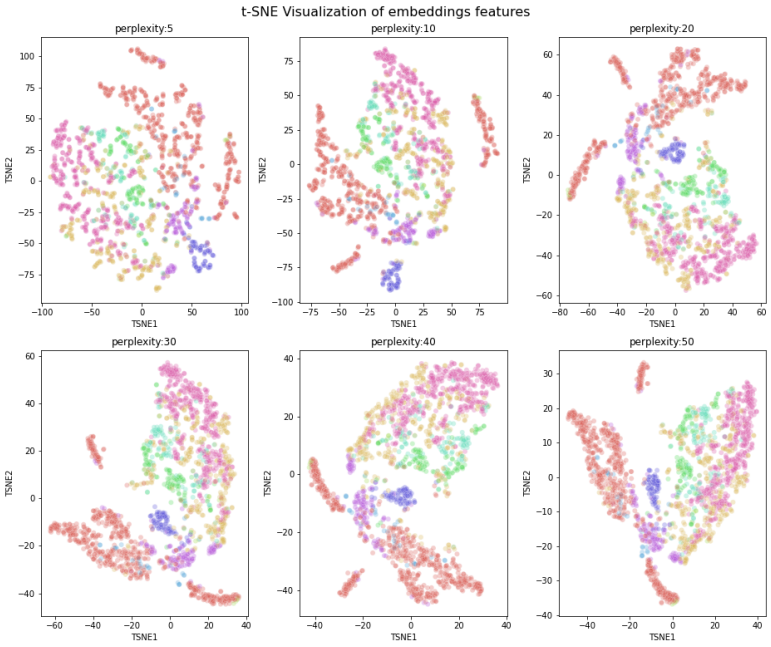


Figure 5.4: Set A - Embeddings features set via t-SNE projection

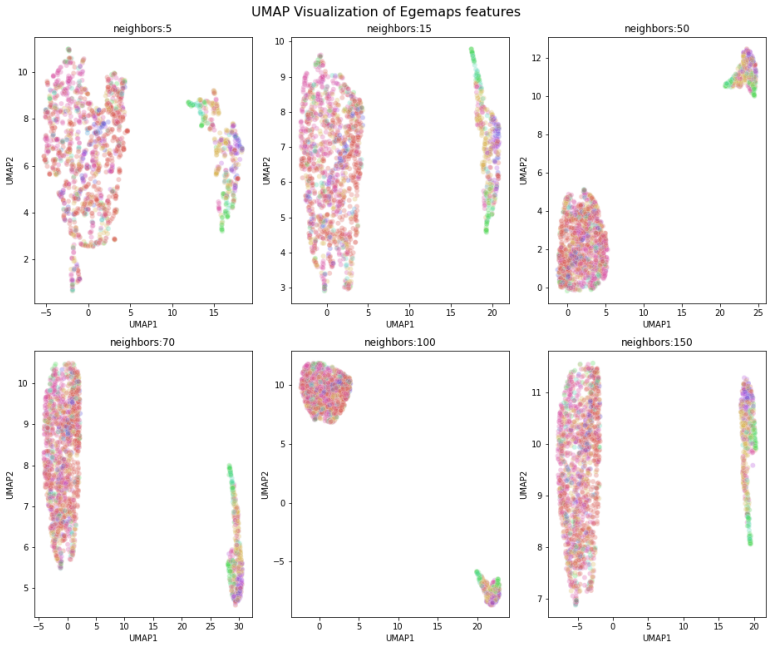


Figure 5.5: Set A - eGeMAPS features set via UMAP projection

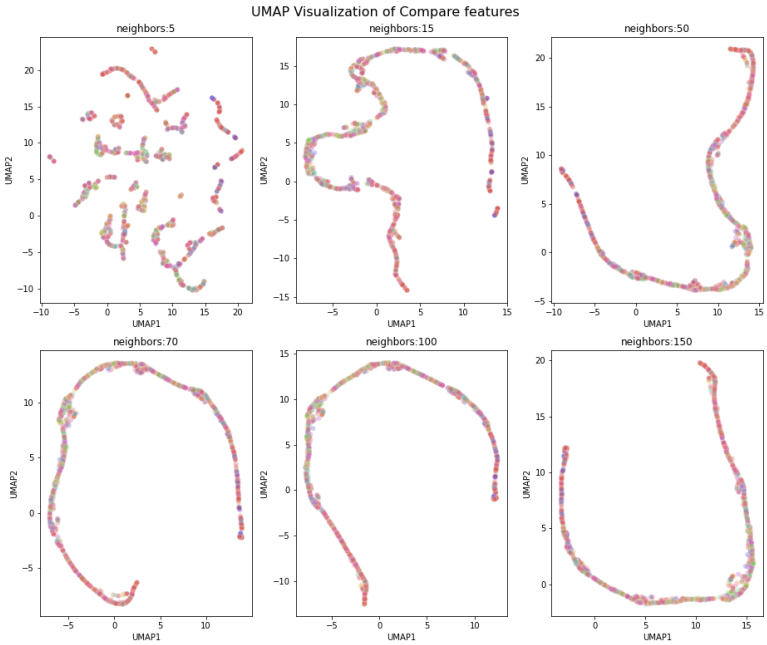


Figure 5.6: Set A - ComParE features set via UMAP projection

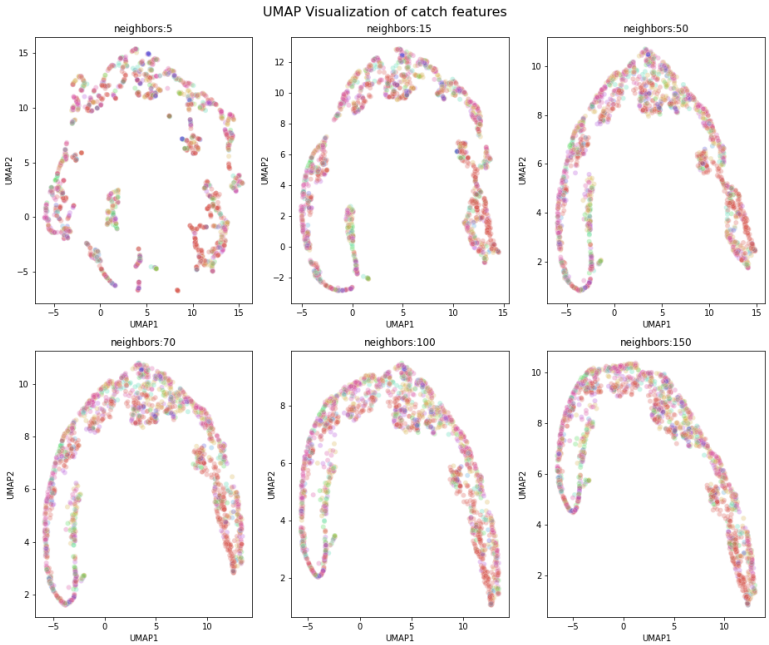


Figure 5.7: Set A - catch22 features set via UMAP projection

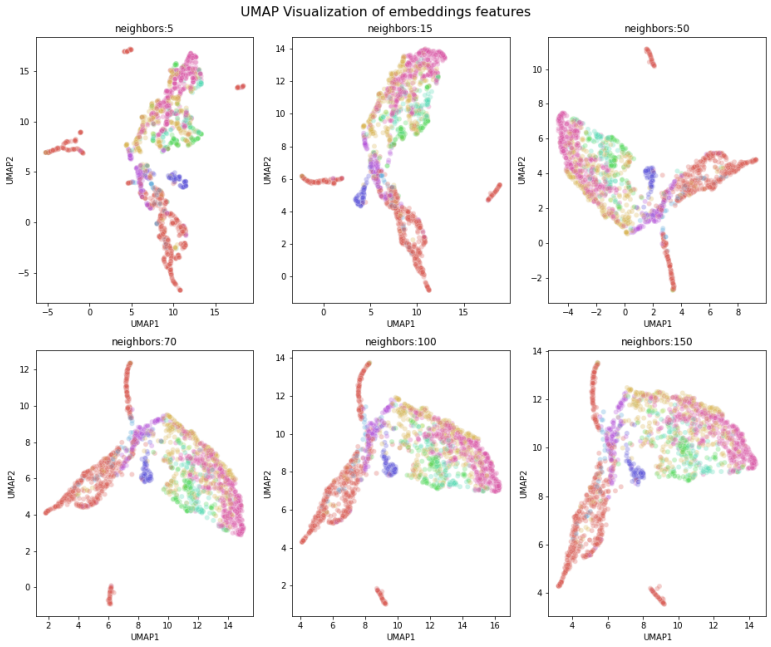


Figure 5.8: Set A - Embeddings features set via UMAP projection

different classes in both projections. This indicates that the Embeddings feature set possesses inherent characteristics that enable the formation of distinguishable clusters.

5.2.2 Set B

Now we proceed with testing the t-SNE projection using set B points:

- eGeMAPS in Fig. 5.9 shows some separation between certain classes. However, there is still a significant amount of overlap among the points. The distinct clusters are not as pronounced as desired.
- ComParE in Fig. 5.10 does not exhibit clear clusters. Instead, the projection displays curves without distinct separations among classes.
- catch22 in Fig. 5.11 there is a resemblance of clusters, similar to the eGeMAPS feature set. However, the points remain mixed, indicating that there is still a lack of clear separation among the classes.
- Embeddings in Fig. 5.12 showcase two classes (purple and green) that are distinctly separated from the others. While some intersection is observed among the remaining classes, this can be attributed to small imbalances in class distribution.

Now, we explore whether the Embeddings feature set continues to exhibit superior separability compared to other feature sets in the UMAP projection:

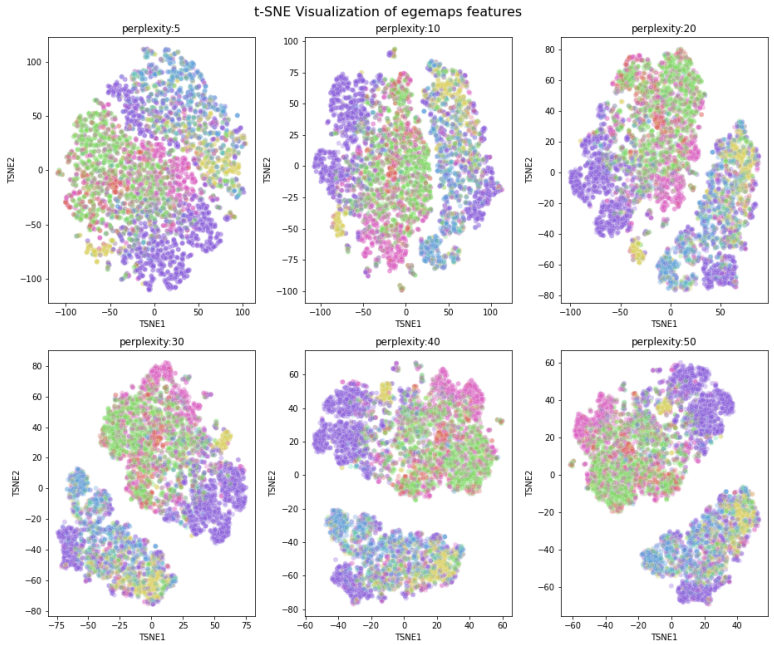


Figure 5.9: Set B - eGeMAPS features set via t-SNE projection

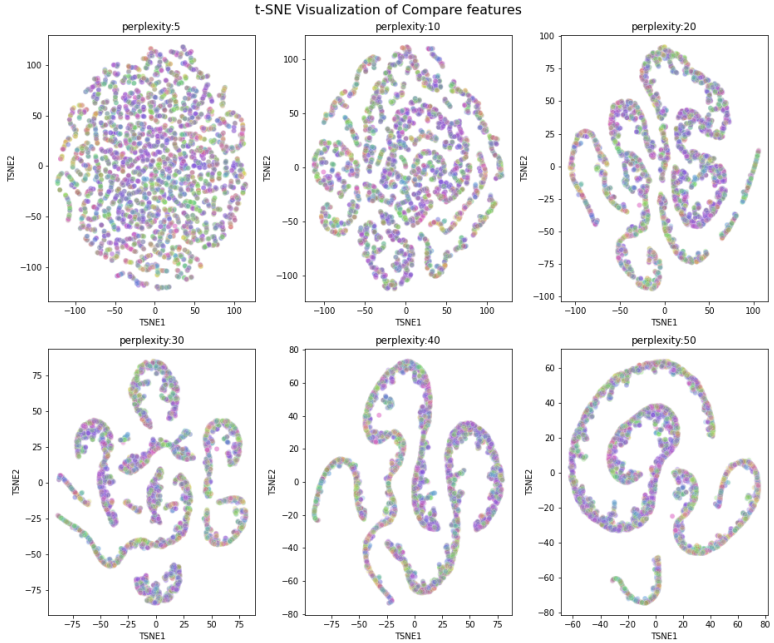


Figure 5.10: Set B - ComParE features set via t-SNE projection

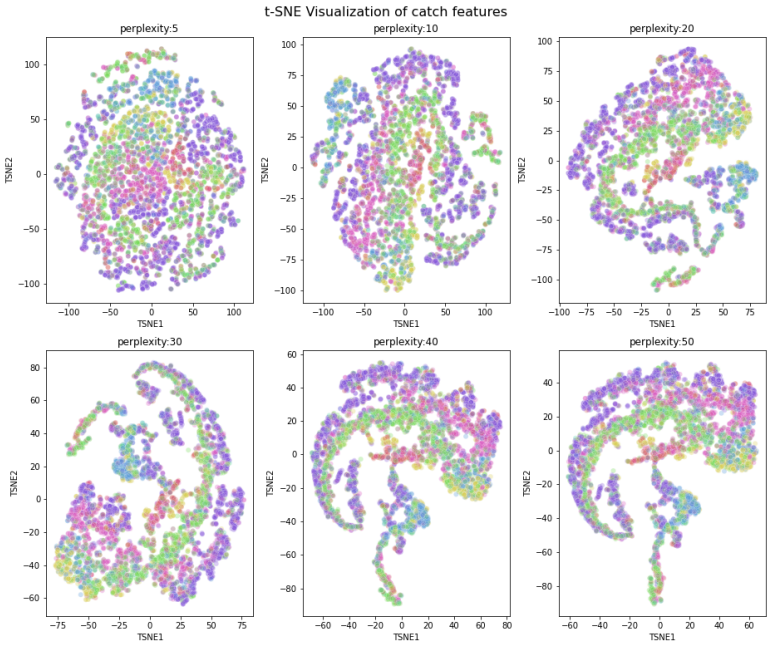


Figure 5.11: Set B - catch22 features set via t-SNE projection

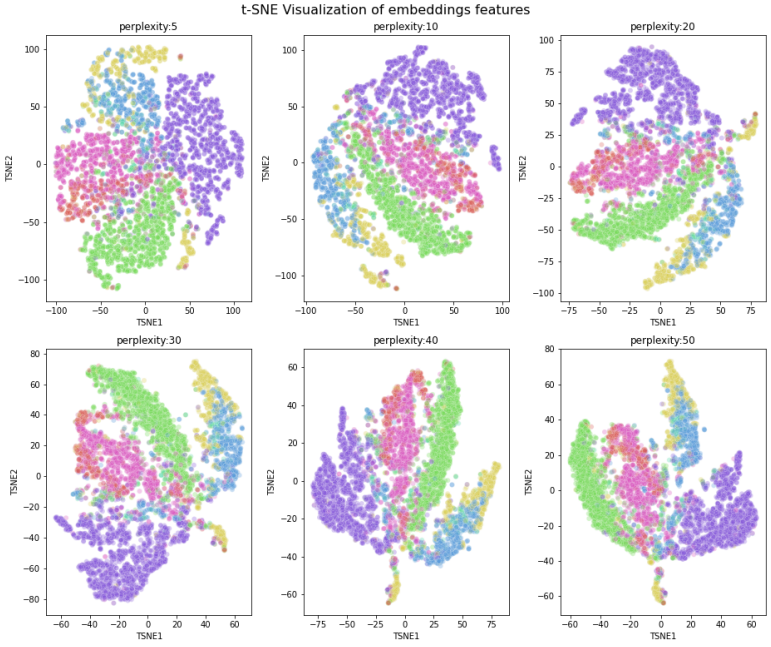


Figure 5.12: Set B - Embeddings features set via t-SNE projection

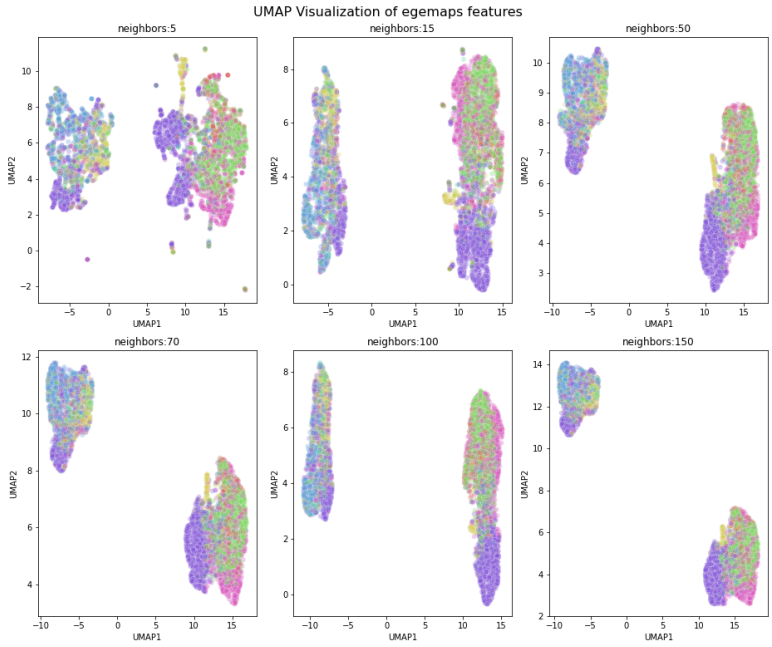


Figure 5.13: Set B - eGeMAPS features set via UMAP projection

- In Fig. 5.13 eGeMAPS, the classes are separated into two sides of the plot. However, there is still a mixing of points within the groups of classes, limiting the distinctiveness of the clusters.
- ComParE in Fig. 5.14 does not exhibit well-defined clusters.
- catch22 in Fig. 5.15there are curves representing some classes such as green, purple, and pink. However, the separability of these clusters is not as distinct as desired.
- Fig. 5.16 displays the early emergence of clusters from the beginning. As the number of neighbors increases, the clusters become more defined and separable.

Overall, the results suggest that the Embeddings feature set consistently exhibits more separability and more apparent cluster formations in both the t-SNE and UMAP projections. This reinforces the potential effectiveness of the Embeddings feature set for subsequent analysis and confirms the high performance of this set.

5.2.3 Set C

For this set, the Embeddings features set has a performance twice as high as the other sets. Even if it is still a low performance, we assess the separability and cluster formations of the different feature sets. In the t-SNE projection:

- In Fig. 5.17 due to class imbalance, only three classes are visible. However, there are no

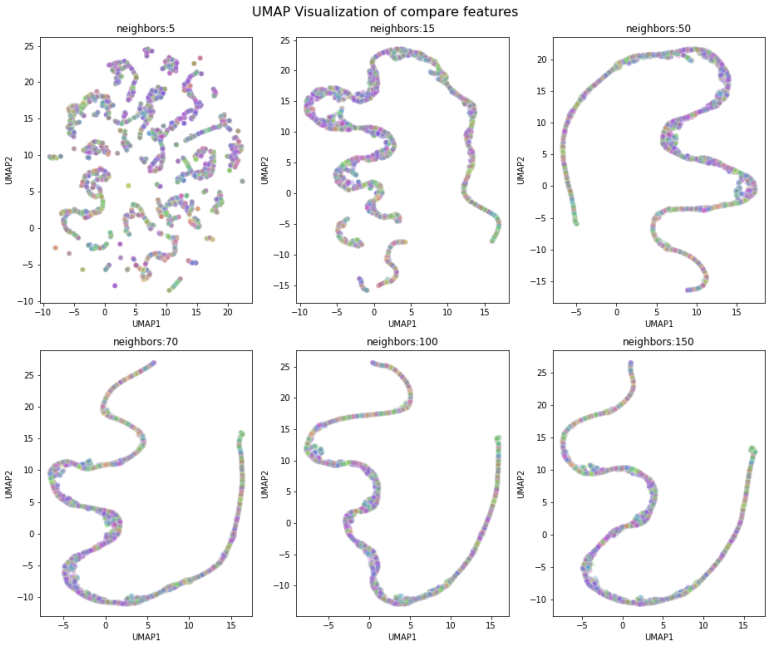


Figure 5.14: Set B - ComParE features set via UMAP projection

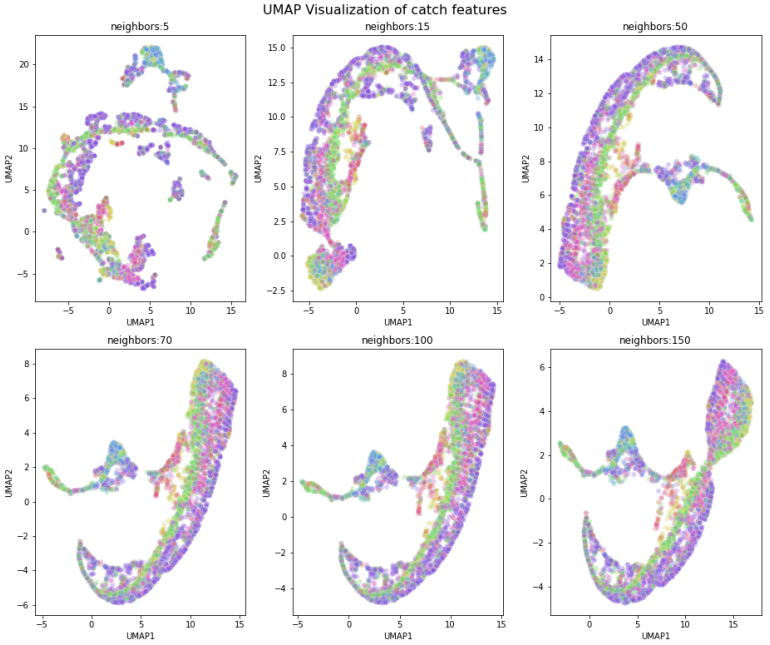


Figure 5.15: Set B - catch22 features set via UMAP projection

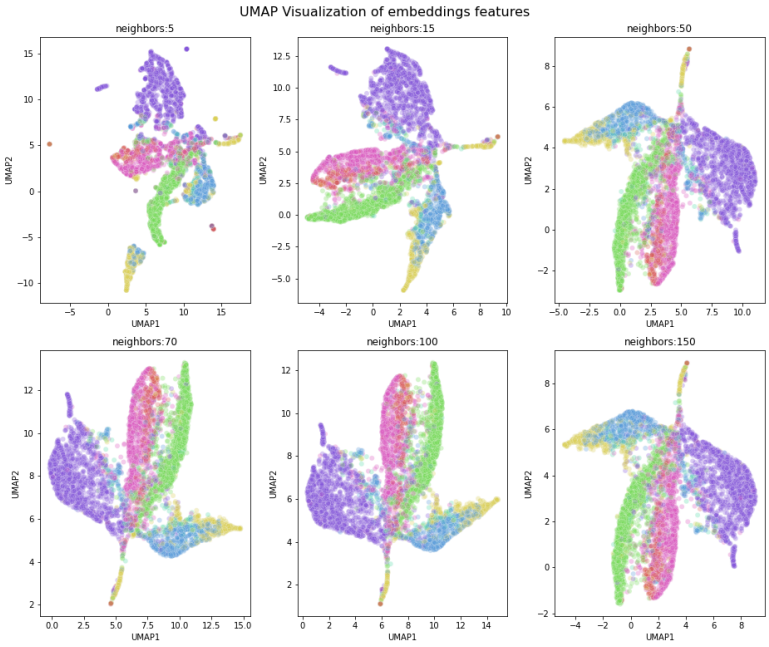


Figure 5.16: Set B - Embeddings features set via UMAP projection

distinct clusters observed in the projection.

- In Fig. 5.18 does not exhibit clear clusters but instead showcases curves without.
- In Fig. 5.19 only one class (pink) appears to be somewhat isolated, while the other points are considerably mixed without discernible clusters.
- In Fig. 5.20 the Embeddings feature set reveals the emergence of three distinct clusters formed by the most numerous classes. This indicates that the Embeddings feature set demonstrates better separability when splitting the space into three distinctive regions.

Now, we will examine the UMAP projection:

- In Fig. 5.21 the space is divided into two distinct parts.
- In Fig. 5.22 again shows the points restricted to a single line, lacking clear cluster formations.
- In Fig. 5.23, the classes are separated into two distinct parts, suggesting a certain level of separability within the data.
- In Fig. 5.24, the three majoritarian classes occupy different parts of the plot. While there are occasional points that mix, overall, the separability and distinctiveness of these classes are evident.

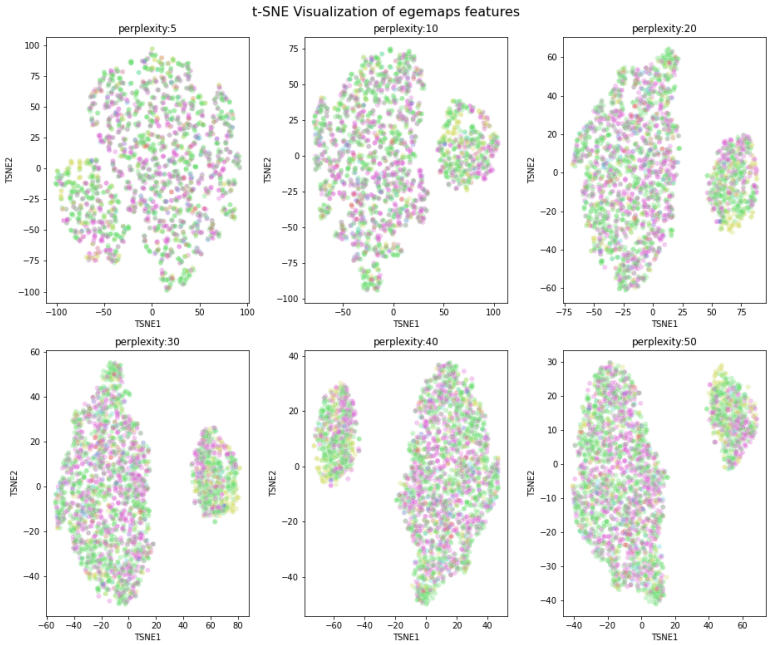


Figure 5.17: Set C - eGeMAPS features set via t-SNE projection

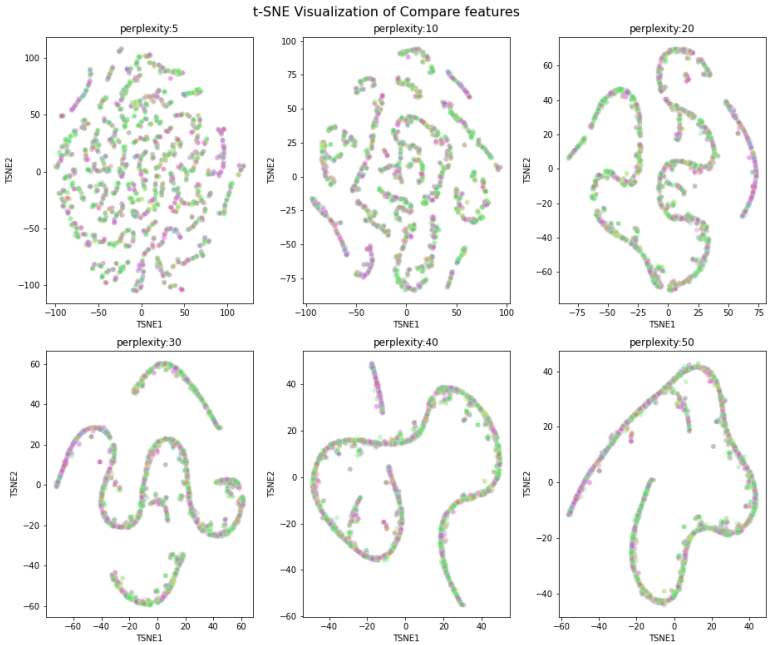


Figure 5.18: Set C - ComParE features set via t-SNE projection

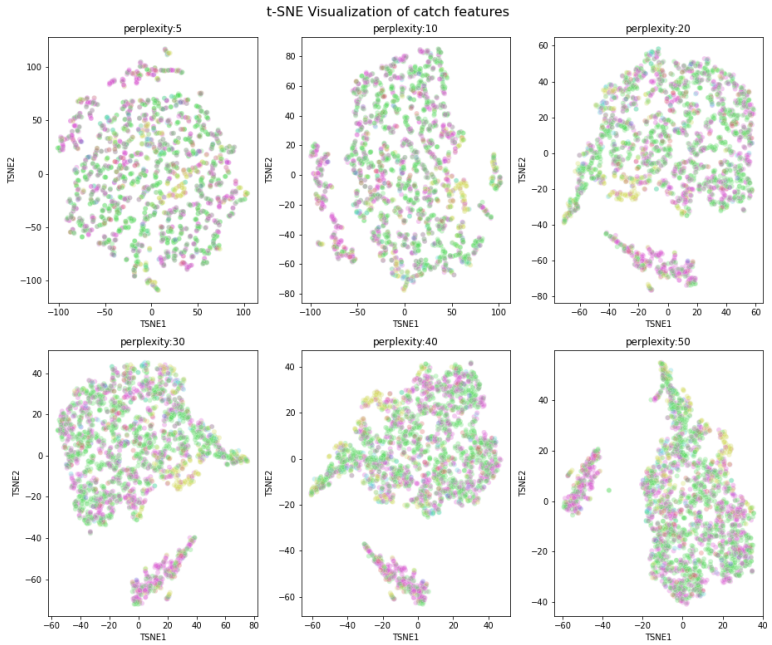


Figure 5.19: Set C - catch22 features set via t-SNE projection

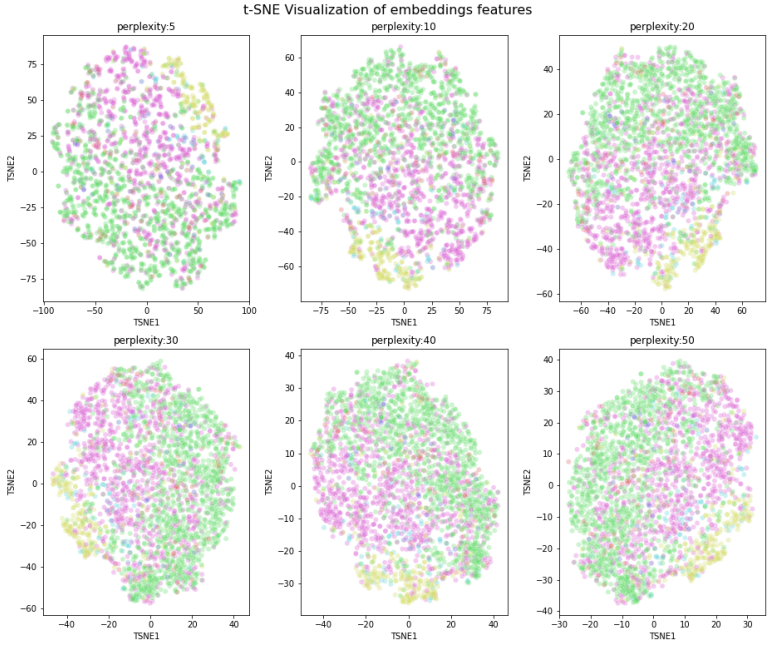


Figure 5.20: Set C - Embeddings features set via t-SNE projection

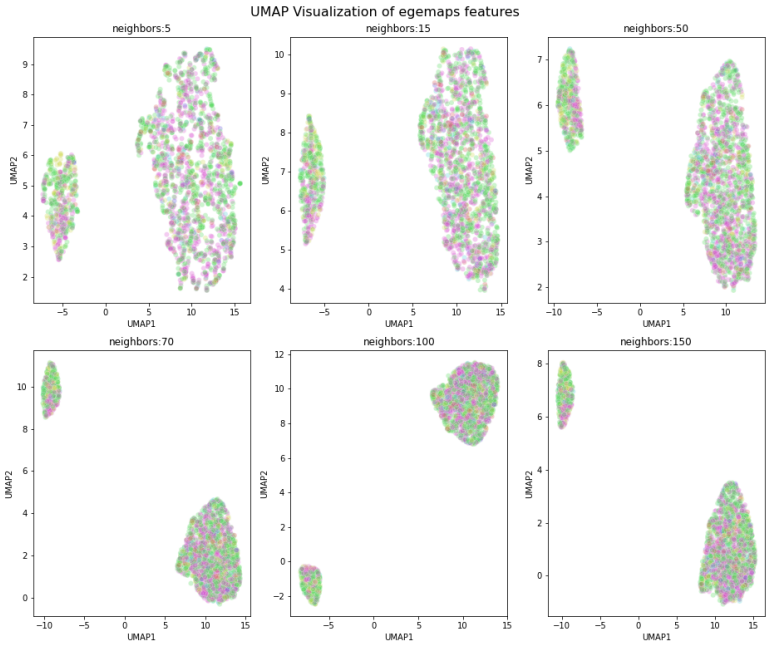


Figure 5.21: Set C - eGeMAPS features set via UMAP projection

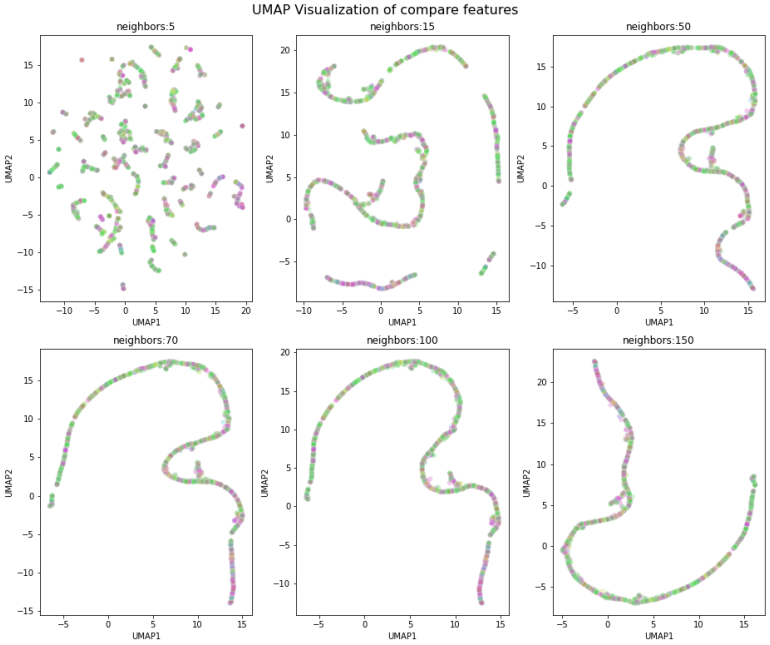


Figure 5.22: Set C - ComParE features set via UMAP projection

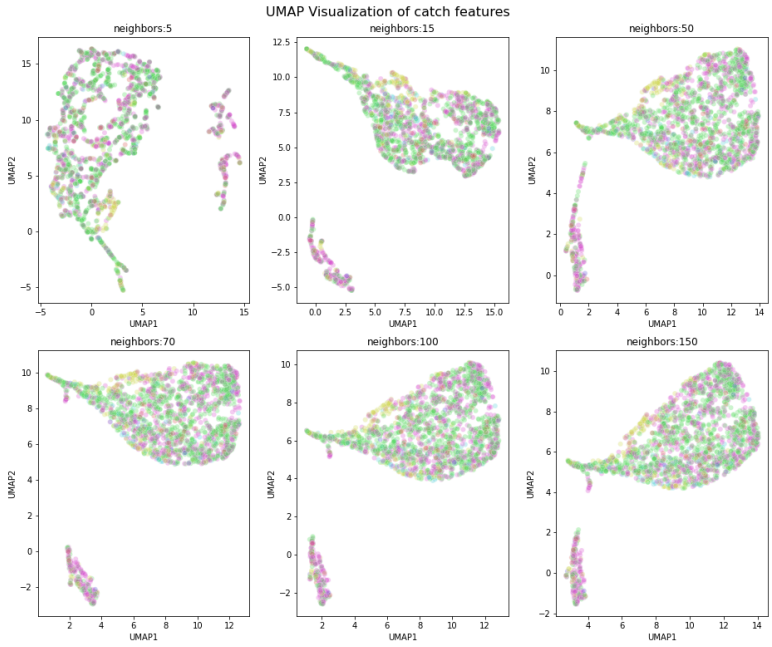


Figure 5.23: Set C - catch22 features set via UMAP projection

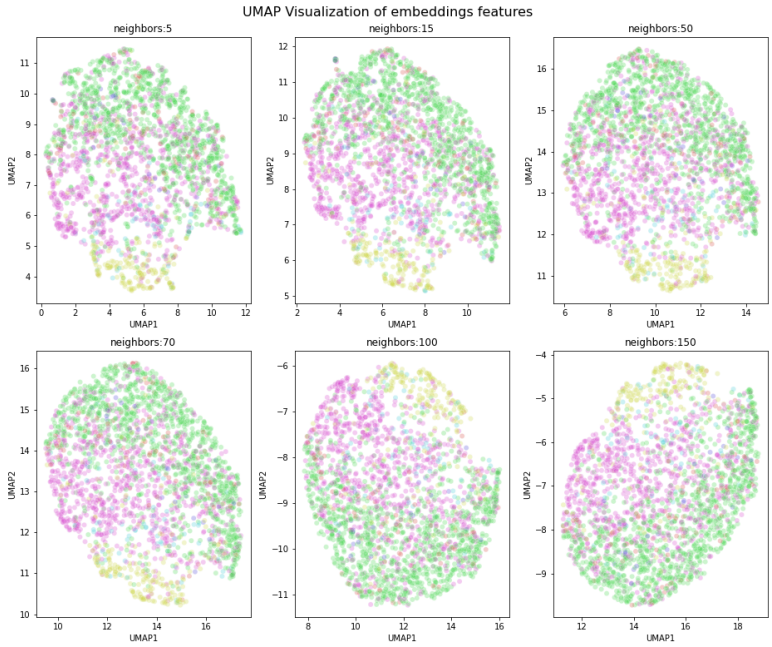


Figure 5.24: Set C - Embeddings features set via UMAP projection

Overall, the observations indicate that the Embeddings feature set consistently demonstrates better separability and cluster formations in the t-SNE and UMAP projections for the three sets. The ComParE feature set generally exhibited curves without well-defined clusters. While eGeMAPS and catch22 resembled clusters, the points were often mixed or lacked clear separations.

5.3 Summary

In this chapter on data visualization, two dimensionality reduction techniques, t-SNE and UMAP, were applied to the different feature sets to analyze their separability and cluster formations. The Embeddings feature set consistently exhibited better separability and more distinct cluster formations compared to other feature sets across all three sets. This suggests that the Embeddings feature set possesses inherent characteristics that enable the formation of distinguishable clusters and may be effective for subsequent analysis.

6 Cross-Dataset Generalization

In this chapter, we aim to evaluate the performance of a trained Convolutional Neural Network on Dataset *j*, which differs from the original training Dataset *i*. The objective is to assess the generalization capability of the trained model when applied to a new and potentially challenging dataset.

6.0.1 Approach Definition

Dataset *i* possesses distinct characteristics and challenges that set it apart from Dataset *j*. These include variations in distribution, class imbalances, or differences in data quality. In addition to evaluating the generalization capability of the trained CNN, it is important to consider whether the learned representations and features captured by the model on Dataset *i* are relevant and applicable to the new dataset.

CNN's ability to extract meaningful and discriminative features is crucial to its generalization performance. If the features learned by the model on Dataset *i* are highly specific to that dataset and do not capture the underlying patterns and characteristics that are relevant to Dataset *j*, the model's performance on the new dataset may be limited. More precisely, as illustrated in Fig. 6.1, the filter stage is trained on Set *i* of calls in these experiments. It is then used as a feature extractor to re-train the classification stage, which is also picked from the trained model. This feature extractor stage is either taken as it is and not re-trained (*Fixed*) or re-trained to fit Set *j* of calls. The UAR is calculated to classify Set *j*.

6.0.2 Results

In our results, we will depict every pair of set *i* and *j* :

- Fully trained on Set *i* re-trained on Set *j* with *fixed* filter stage.
- Fully trained on Set *i* re-trained on Set *j* with *training* filter stage.

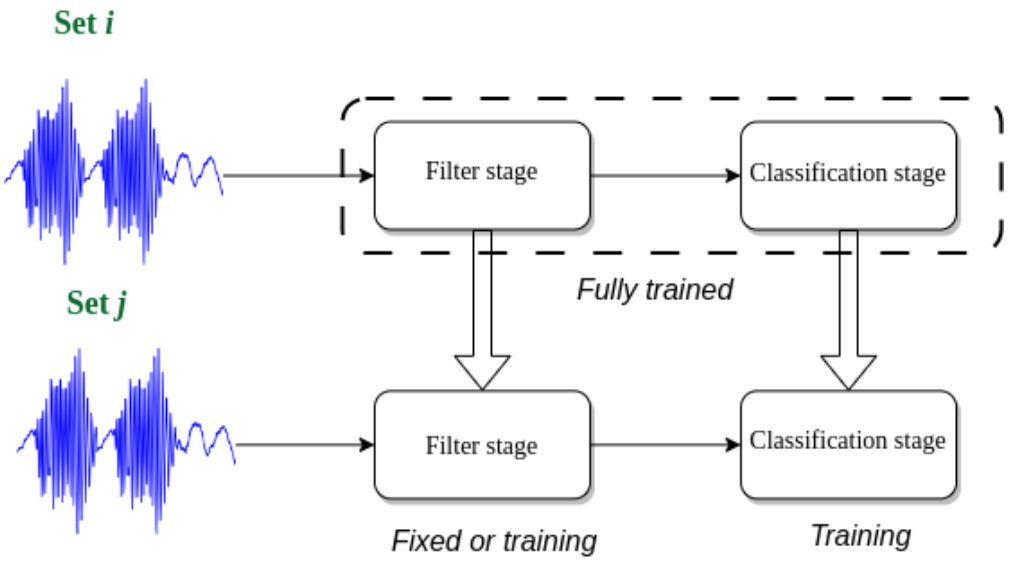


Figure 6.1: Illustration of the cross-dataset experiments

Fully trained on	Classification of	Filter stage *fixed* or *training*	UAR
Set B	Set A	Fixed	0.53 ± 0.04
Set B	Set A	Training	0.71 ± 0.07
Set A	Set B	Fixed	0.54 ± 0.03
Set A	Set B	Training	0.71 ± 0.03
Set A	Set C	Fixed	0.26 ± 0.03
Set A	Set C	Training	0.28 ± 0.01
Set B	Set C	Fixed	0.26 ± 0.06
Set B	Set C	Training	0.29 ± 0.02

Table 6.1: UAR values of the cross-dataset experiments

In our table of results in Table 6.1, the UAR is average on the five folds. The fully trained model chosen for every Set is the one that has the best performance out of the five folds.

The initial observation of the table is the low performance of the *fixed* filter stage in all situations. Other observations are :

- Fully trained on Set B and classification of Set A, $UAR = 0.71$ performs better than the CNN fully trained and classification of Set A, $UAR = 0.61$.
- Fully trained on Set A and classification of Set B, $UAR = 0.71$ performs lower than the CNN fully trained and classification on B, $UAR = 0.73$
- Fully trained Set A/B and classification of Set C performs lower than the CNN fully trained and classification on Set C.

From the initial observation that the *fixed* filter stage performs poorly, we suggest that the direct application of a pre-trained CNN model's feature extraction stage, without further adaptation, does not lead to the anticipated performance in terms of classification performance. However, when the filter stage of the CNN model is re-trained and adapted specifically to the target task, improved results are achieved compared to training the model from scratch. This remark is specifically pointing to the improvement of the classification of Set A.

This transfer of the filter stage from one model to train another set implies that the neural network layers are not initialized with random weights but rather provided with a more favorable starting point. By utilizing the pre-trained model, the network has already acquired knowledge of the relevant patterns to extract and facilitate the subsequent classification process.

These findings highlight the significance of transfer learning in enhancing the performance of deep neural networks. By leveraging pre-existing learned representations, particularly in the filter stage, the model can effectively leverage prior knowledge and build upon it to achieve improved classification accuracy.

Nevertheless, the performance with Set C remains unsatisfactory, indicating that the transferability of the learned features may not be effective in this particular case. A possible explanation could be the inherent dissimilarity in the nature of vocal calls between Set C and Sets A/B. The divergent characteristics and acoustic properties of the vocalizations in Set C may pose unique challenges. The poor performance on Set C suggests that the classification task for this Set requires a more specialized and tailored approach, potentially involving the development of specific feature extraction techniques or models that are specifically trained on Set C.

6.1 Summary

In this chapter, we evaluated the performance of a trained Convolutional Neural Network on a different set from the original training set. The goal was to assess the generalization capability of the trained model. It was also to determine if the learned representations and features were relevant and applicable to the new set.

We utilized the filter stage of the trained CNN as a feature extractor and re-trained the classification stage. We experimented with two scenarios: keeping the filter stage fixed and re-training the filter stage to fit the new set. We observed that the fixed filter stage performed consistently low in all situations. However, when we re-trained the filter stage specifically for the new set, we achieved improved results compared to training from scratch. Overall, our findings highlighted the significance of transfer learning in improving the performance of deep neural networks. By leveraging pre-existing learned representations, particularly in the filter stage, the model can build upon prior knowledge and achieve better classification accuracy.

7 Conclusion

This thesis explored feature extraction and classification methods for meerkat call-type classification, if there is a possibility to learn to classify meerkat calls. After providing an introduction to bioacoustics and the significance of understanding animal vocalizations. We introduced meerkats, with their complex social structure, adaptability, and diverse communication system. There is a limited application of machine learning and deep learning techniques in the study of meerkat vocalizations. Even though a number of studies have focused on vocal identification and pattern recognition, there is still much to be discovered in terms of accurately classifying different types of calls. It is a laborious task, manually done, which often lacks consensus among experts. Developing a reliable method tailored for this problematic is crucial to unravel their unique characteristics and complexities.

To conduct our experiments, we were provided with three sets of data. A feature set extracted from an end-to-end CNN approach applied to an SVM had the highest performance metric (UAR) for every set. That result shows that the CNN approach is the one that was able to encode the complex pattern of calls and their uniqueness to be able to differentiate between them. By using dimentionality reduction technique on the feature set, we were able to confirm the clustering tendency of this feature set.

The three sets of data differ. Both Set A and Set B contain approximately the same types of calls, with Set B being more class balanced and containing a greater number of audio files. Set C contains close-call subcategories. These categories are contextual behaviours observed by the researchers when recording the call. The results of classification of calls of Set C are unsatisfactory. It is necessary to investigate other features that may catch these conceptual behavior in more detail in order to gain a deeper understanding of these calls. An improved classification system needs to be developed that can better capture these close-call subcategories.

During this thesis, we also explored the generalization of the end-to-end CNN approach trained on one set and tested on another. This allowed us to increase the performance metric of Set A when its filter stage is replaced by the filter stage of the fully trained model on Set B. By

finding a satisfying method to classify meerkat calls, we are able to contribute to the broader field of animal vocalization analysis.

7.1 Future Directions

The next step in this study will be to expand the dataset by decreasing the imbalance of classes and adding a greater number of audio files. We should also investigate other feature extraction and classification methods to have a concrete results for Set C. For further exploration in the classification task, one can investigate Self Supervised Learning (SSL). In traditional supervised learning, models rely on labeled data, where each input is paired with a corresponding target or label. However, obtaining large-scale labeled datasets can be expensive and time-consuming. Thus the emergence of SSL techniques that offer a powerful alternative to these traditional learning methods by leveraging unlabeled data and designing pretext tasks. By doing that, it allows models to learn meaningful representations without relying on explicit human annotations. And there is a possibility to investigate the cross-transferability of these representations learned from human speech for analyzing bioacoustics.

Bibliography

Panu Somervuo, Aki Härmä, and Seppo Fagerlund. Parametric Representations of Bird Sounds for Automatic Species Recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14:2252–2263, December 2006. doi: 10.1109/TASL.2006.872624.

Daniel Y. Takahashi. Animal communication: Chit-chat in meerkats. *Current Biology*, 28(22): R1298–R1300, 2018. ISSN 0960-9822. doi: https://doi.org/10.1016/j.cub.2018.09.038. URL https://www.sciencedirect.com/science/article/pii/S0960982218312636.

Tim Clutton-Brock, Peter Brotherton, Rebecca Smith, G McIlrath, Ruth Kansky, David Gaynor, J.M. O'Riain, and J Skinner. Infanticide and expulsion of females in a cooperative mammal. *Proceedings. Biological sciences / The Royal Society*, 265:2291–5, 01 1999. doi: 10.1098/rspb.1998.0573.

Marta Manser, Robert Seyfarth, and Dorothy Cheney. Suricate alarm calls signal predator class and urgency. *Trends in Cognitive Sciences*, 6:55–57, 02 2002. doi: 10.1016/S1364-6613(00)01840-4.

Gabriella Gall and Marta Manser. Spatial structure of foraging meerkat groups is affected by both social and ecological factors. *Behavioral Ecology and Sociobiology*, 72, 04 2018. doi: 10.1007/s00265-018-2490-x.

Alessandro De Luca, Ariana Strandburg-Peshkin, Britta Walkenhorst, and Marta Manser. Comparison of machine learning methods for the vocal identification of meerkats (suricata suricatta), 07 2022.

Mara Thomas, Frants H. Jensen, Baptiste Averly, Vlad Demartsev, Marta B. Manser, Tim Sainburg, Marie A. Roch, and Ariana Strandburg-Peshkin. A practical guide for generating unsupervised, spectrogram-based latent space representations of animal vocalizations. *Journal of Animal Ecology*, 91(8):1567–1581, 2022. doi: https://doi.org/10.1111/1365-2656.13754. URL https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/1365-2656.13754.

Tim Sainburg, Marvin Thielk, and Timothy Q. Gentner. Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLOS Computational Biology*, 16(10):1–48, 10 2020. doi: 10.1371/journal.pcbi.1008228. URL https://doi.org/10.1371/journal.pcbi.1008228.

Dimitri Palaz, Mathew Magimai-Doss, and Ronan Collobert. End-to-end acoustic modeling using convolutional neural networks for hmm-based automatic speech recognition. *Speech Communication*, 108:15–32, 2019. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2019.01.004. URL https://www.sciencedirect.com/science/article/pii/S0167639316301625.

Robert M. Seyfarth and Dorothy L. Cheney. Meaning and emotion in animal vocalizations. *Annals of the New York Academy of Sciences*, 1000(1):32–55, 2003. doi: https://doi.org/10.1196/annals.1280.004. URL https://nyaspubs.onlinelibrary.wiley.com/doi/abs/10.1196/annals.1280.004.

Maxime Garcia and Livio Favaro. Animal vocal communication: function, structures, and production mechanisms. *Current Zoology*, 63(4):417–419, 05 2017. ISSN 1674-5507. doi: 10.1093/cz/zox040. URL https://doi.org/10.1093/cz/zox040.

Adin Ross-Gillespie and Ashleigh S Griffin. Meerkats. *Current Biology*, 17(12):R442–R443, 2007.

Simon W. Townsend, Benjamin D. Charlton, and Marta B. Manser. Acoustic cues to identity and predator context in meerkat barks. *Animal Behaviour*, 94:143–149, 2014a. ISSN 0003-3472. doi: https://doi.org/10.1016/j.anbehav.2014.05.021. URL https://www.sciencedirect.com/science/article/pii/S0003347214002413.

William L Gannon and Timothy E Lawlor. Variation of the chip vocalization of three species of townsend chipmunks (genus eutamias). *Journal of Mammalogy*, 70(4):740–753, 1989.

Kathleen M. Stafford, Sharon L. Nieukirk, and Christopher G. Fox. Low-frequency whale sounds recorded on hydrophones moored in the eastern tropical Pacific. *The Journal of the Acoustical Society of America*, 106(6):3687–3698, 12 1999. ISSN 0001-4966. doi: 10.1121/1.428220. URL https://doi.org/10.1121/1.428220.

Hui Ou, Whitlow W. L. Au, and Julie N. Oswald. A non-spectrogram-correlation method of automatically detecting minke whale boings. *The Journal of the Acoustical Society of America*, 132(4):EL317–EL322, 09 2012. ISSN 0001-4966. doi: 10.1121/1.4747816. URL https://doi.org/10.1121/1.4747816.

Hossein Zamanian and Hossein Pourghassem. Insect identification based on bioacoustic signal using spectral and temporal features. In *2017 Iranian Conference on Electrical Engineering (ICEE)*, pages 1785–1790, 2017. doi: 10.1109/IranianCEE.2017.7985340.

Navinda Kottege, Frederieke Kroon, Raja Jurdak, and Dean Jones. Classification of underwater broadband bio-acoustics using spectro-temporal features. In *Proceedings of the 7th International Conference on Underwater Networks and Systems*, WUWNet '12, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450317733. doi: 10.1145/2398936.2398961. URL https://doi.org/10.1145/2398936.2398961.

Xuan Wu, Silong Zhou, Mingwei Chen, Yihang Zhao, Yifei Wang, Xianmeng Zhao, Danyang Li, and Haibo Pu. Combined spectral and speech features for pig speech recognition. *PLOS*

ONE, 17(12):1–22, 12 2022. doi: 10.1371/journal.pone.0276778. URL https://doi.org/10.1371/journal.pone.0276778.

Arik Kershenbaum, Daniel T. Blumstein, Marie A. Roch, Çalar Akçay, Gregory Backus, Mark A. Bee, Kirsten Bohn, Yan Cao, Gerald Carter, Cristiane Cäsar, Michael Coen, Stacy L. DeRuiter, Laurance Doyle, Shimon Edelman, Ramon Ferrer-i Cancho, Todd M. Freeberg, Ellen C. Garland, Morgan Gustison, Heidi E. Harley, Chloé Huetz, Melissa Hughes, Julia Hyland Bruno, Amiyaal Ilany, Dezhe Z. Jin, Michael Johnson, Chenghui Ju, Jeremy Karnowski, Bernard Lohr, Marta B. Manser, Brenda McCowan, Eduardo Mercado III, Peter M. Narins, Alex Piel, Megan Rice, Roberta Salmi, Kazutoshi Sasahara, Laela Sayigh, Yu Shiu, Charles Taylor, Edgar E. Vallejo, Sara Waller, and Veronica Zamora-Gutierrez. Acoustic sequences in non-human animals: a tutorial review and prospectus. *Biological Reviews*, 91(1):13–52, 2016. doi: https://doi.org/10.1111/brv.12160. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/brv.12160.

Patrick J. Clemins, Michael T. Johnson, Kirsten M. Leong, and Anne Savage. Automatic classification and speaker identification of African elephant (Loxodonta africana) vocalizations. *The Journal of the Acoustical Society of America*, 117(2):956–963, 01 2005. ISSN 0001-4966. doi: 10.1121/1.1847850. URL https://doi.org/10.1121/1.1847850.

Angel David Pedroza Ramirez, Jose Ismael de la Rosa Vargas, Rogelio Rosas Valdez, and Aldonso Becerra. A comparative between mel frequency cepstral coefficients (mfcc) and inverse mel frequency cepstral coefficients (imfcc) features for an automatic bird species recognition system. In *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–4, 2018. doi: 10.1109/LA-CCI.2018.8625230.

Kuntoro Adi, Michael T Johnson, and Tomasz S Osiejuk. Acoustic censusing using automatic vocalization classification and identity recognition. *The Journal of the Acoustical Society of America*, 127(2):874–883, 2010.

S. Datta and C. Sturtivant. Dolphin whistle classification for determining group identities. *Signal Processing*, 82(2):251–258, 2002. ISSN 0165-1684. doi: https://doi.org/10.1016/S0165-1684(01)00184-0. URL https://www.sciencedirect.com/science/article/pii/S0165168401001840.

David W Armitage and Holly K Ober. A comparison of supervised learning techniques in the classification of bat echolocation calls. *Ecological Informatics*, 5(6):465–473, 2010.

Jesse C. Ross and Paul E. Allen. Random forest for improved analysis efficiency in passive acoustic monitoring. *Ecological Informatics*, 21:34–39, 2014. ISSN 1574-9541. doi: https://doi.org/10.1016/j.ecoinf.2013.12.002. URL https://www.sciencedirect.com/science/article/pii/S1574954113001234. Ecological Acoustics.

Nallagasu Raju, S. Mathini, Teena Priya, Preethi P Preethi, and M. Chandrasekar. Identifying the population of animals through pitch, formant, short time energy-a sound analysis. 03 2012. doi: 10.1109/ICCEET.2012.6203766.

Seppo Fagerlund. Bird species recognition using support vector machines. *EURASIP Journal on Advances in Signal Processing*, 2007:1–8, 2007.

Matthias Zeppelzauer, Sean Hensman, and Angela S Stoeger. Towards an automated acoustic detection system for free-ranging elephants. *Bioacoustics*, 24(1):13–29, 2015.

Emmanuel Dufourq, Ian Durbach, James P. Hansford, Amanda Hoepfner, Heidi Ma, Jessica V. Bryant, Christina S. Stender, Wenyong Li, Zhiwei Liu, Qing Chen, Zhaoli Zhou, and Samuel T. Turvey. Automated detection of hainan gibbon calls for passive acoustic monitoring. *Remote Sensing in Ecology and Conservation*, 7(3):475–487, 2021. doi: https://doi.org/10.1002/rse2.201. URL https://zslpublications.onlinelibrary.wiley.com/doi/abs/10.1002/rse2.201.

Juan Colonna, Tanel Peet, Carlos Abreu Ferreira, Alpio M. Jorge, Elsa Ferreira Gomes, and Joao Gama. Automatic classification of anuran sounds using convolutional neural networks. In *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering*, C3S2E '16, page 7378, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340755. doi: 10.1145/2948992.2949016. URL https://doi.org/10.1145/2948992.2949016.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Florian Eyben, Martin Wöllmer, and Björn Schuller. opensmile – the munich versatile and fast open-source audio feature extractor. pages 1459–1462, 01 2010. doi: 10.1145/1873951.1874246.

Florian Eyben, Klaus R. Scherer, Björn W. Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Y. Devillers, Julien Epps, Petri Laukka, Shrikanth S. Narayanan, and Khiet P. Truong. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7(2):190–202, 2016. doi: 10.1109/TAFFC.2015.2457417.

Ziqiang Bao, Shuai Zhao, Shuang Li, Guisong Jiang, Huazhi Sun, and Long Zhang. *Multi-dimensional Convolutional Neural Network for Speech Emotion Recognition*, pages 296–303. 01 2023. ISBN 978-981-19-3631-9. doi: 10.1007/978-981-19-3632-6_37.

Björn Schuller, Stefan Steidl, Anton Batliner, Julia Hirschberg, Judee Burgoon, Alice Baird, Aaron Elkins, Yue Zhang, Eduardo Coutinho, and Keelan Evanini. The interspeech 2016 computational paralinguistics challenge: Deception, sincerity and native language. pages 2001–2005, 09 2016. doi: 10.21437/Interspeech.2016-129.

Ben Fulcher, Max Little, and Nick Jones. Highly comparative time-series analysis: the empirical structure of time series and their methods. 2013. doi: https://doi.org/10.1098/rsif.2013.0048.

Nikhil Phaniraj, Kaja Wierucka, Yvonne Zürcher Zurcher, and Judith M. Burkart. Optimising source identification from marmoset vocalisations with hierarchical machine learning classifiers. *bioRxiv*, 2022. doi: 10.1101/2022.11.19.517179. URL https://www.biorxiv.org/content/early/2022/11/20/2022.11.19.517179.

Carl H. Lubba, Sarab S. Sethi, Philip Knaute, Simon R. Schultz, Ben D. Fulcher, and Nick S. Jones. catch22: Canonical time-series characteristics. 2019. doi: https://doi.org/10.1007/s10618-019-00647-x.

Dan Stowell. Computational bioacoustics with deep learning: a review and roadmap, 2021.

Mario Lasseck. Audio-based bird species identification with deep convolutional neural networks. 09 2018.

Patrice Guyot, Fanny Alix, Thomas Guerin, Elie Lambeaux, and Alexis Rotureau. Fish migration monitoring from audio detection with cnns. In *Proceedings of the 16th International Audio Mostly Conference*, AM '21, page 244247, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385695. doi: 10.1145/3478384.3478393. URL https://doi.org/10.1145/3478384.3478393.

Sanchez Bravo, Md Rahat Hossain, Nathan English, and Steven T. Moore. Bioacoustic classification of avian calls from raw sound waveforms with an open-source deep learning architecture. 2021. doi: doi.org/10.1038/s41598-021-95076-6.

Kui Wang, Pei Wu, Hongmei Cui, Chuanzhong Xuan, and Hue Su. Identification and classification for sheep foraging behavior based on acoustic signal and deep learning. *Computers and Electronics in Agriculture*, 187:106275, 2021. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2021.106275. URL https://www.sciencedirect.com/science/article/pii/S0168169921002921.

Pu Li, Xiaobai Liua, K. J. Palmer, Erica Fleishman, Douglas Gillespie, Eva-Marie Nosal, Yu Shiu, Holger Klinck, Danielle Cholewiak, Tyler Helble, and Marie A. Roch. Learning deep models from synthetic data for extracting dolphin whistle contours, 2020.

Imran Zualkernan, Jacky Judas, Taslim Mahbub, Azadan Bhagwagar, and Priyanka Chand. A tiny cnn architecture for identifying bat species from echolocation calls. In *2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G)*, pages 81–86, 2020. doi: 10.1109/AI4G50087.2020.9311084.

Yukio Fukuzawa, Wesley H. Webb, Matthew D.M. Pawley, Michelle M. Roper, Stephen Marsland, Dianne H. Brunton, and Andrew Gilman. Koe: Web-based software to classify acoustic units and analyse sequence structure in animal vocalizations. *Methods in Ecology and Evolution*, 11(3):431–441, 2020. doi: https://doi.org/10.1111/2041-210X.13336. URL https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13336.

Dimitri Palaz, Ronan Collobert, and Mathew Magimai. Doss. Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks, 2013.

Simon W. Townsend, Benjamin D. Charlton, and Marta B. Manser. Acoustic cues to identity and predator context in meerkat barks. *Animal Behaviour*, 94:143–149, 2014b. ISSN 0003-3472. doi: https://doi.org/10.1016/j.anbehav.2014.05.021. URL https://www.sciencedirect.com/science/article/pii/S0003347214002413.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

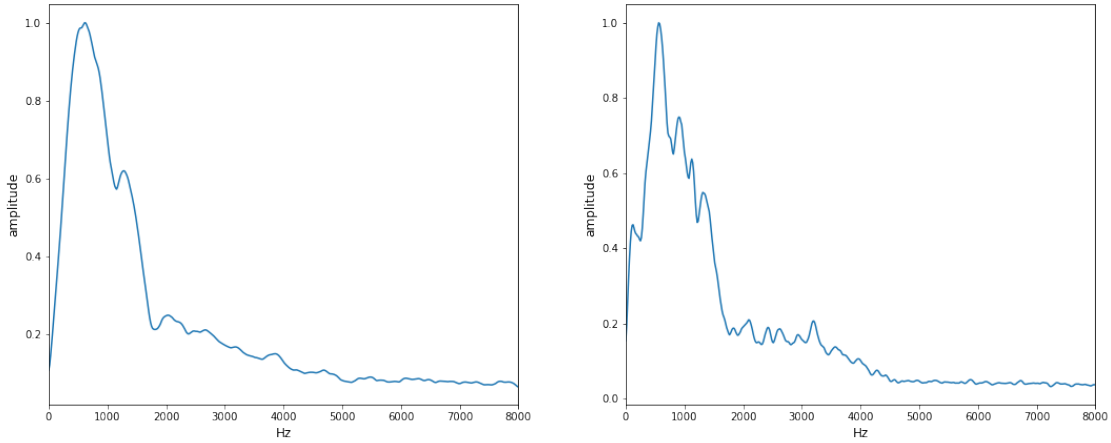
Appendix

Cumulative Frequency Response

This sections presents the cumulative frequency response of the filters of the first convolution layer of the CNN for Set A and Set B.

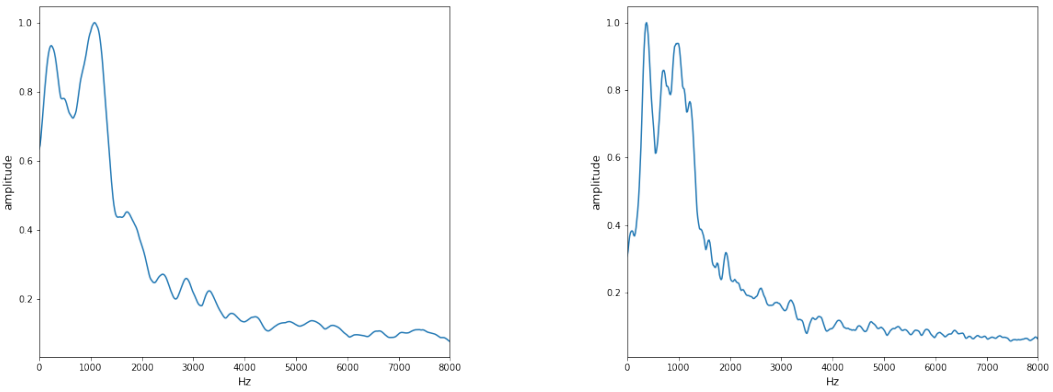
Confusion Matrix

This section presents the confusion matrix of every result of Table 4.3. Regarding the CNN classifier, the confusion matrix presented is the sum of the confusion matrix obtained from the five folds.



(a) $kW_1 = 40$ (b) $kW_1 = 120$

Figure 7.1: Cumulative frequency responses of first layer filters, trained on the Set B



(a) $kW_1 = 40$ (b) $kW_1 = 120$

Figure 7.2: Cumulative frequency responses of first layer filters, trained on the Set C

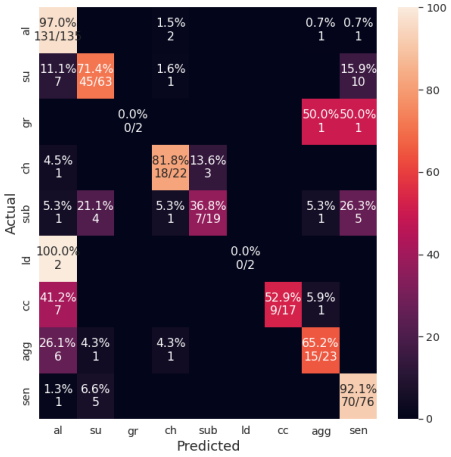


Figure 7.3: Confusion Matrix for RF Classifier Using eGeMAPS Feature Set of Set A.

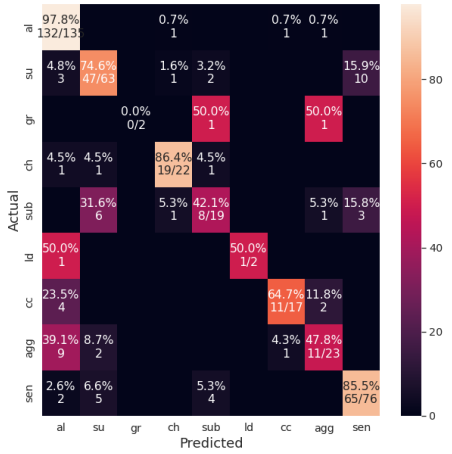


Figure 7.4: Confusion Matrix for SVM Classifier Using eGeMAPS Feature Set of Set A

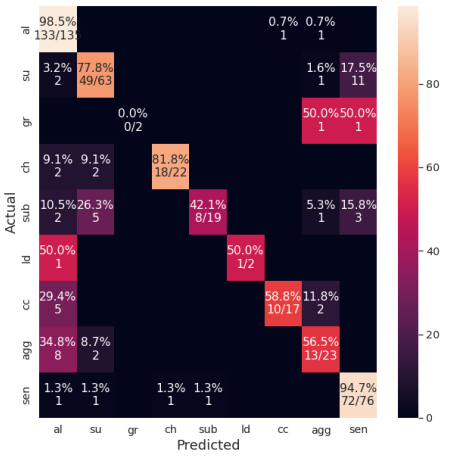


Figure 7.5: Confusion Matrix for RF Classifier Using ComParE Feature Set of Set A.

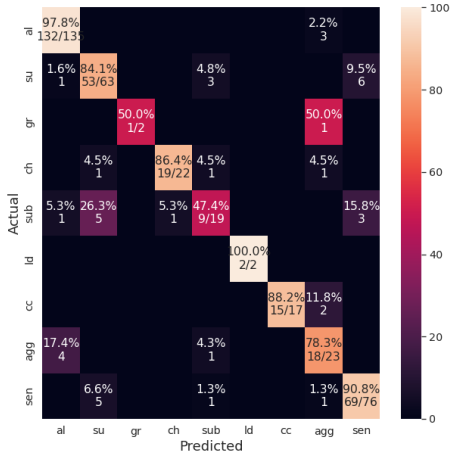


Figure 7.6: Confusion Matrix for SVM Classifier Using ComParE Feature Set of Set A.

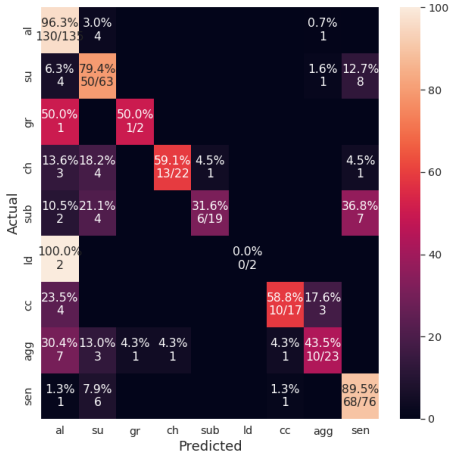


Figure 7.7: Confusion Matrix for RF Classifier Using catch22 Feature Set of Set A

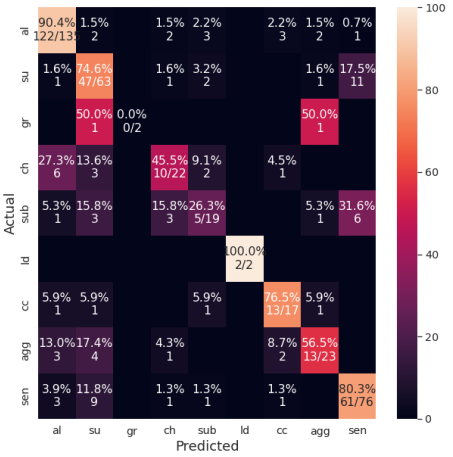


Figure 7.8: Confusion Matrix for SVM Classifier Using catch22 Feature Set of Set A.

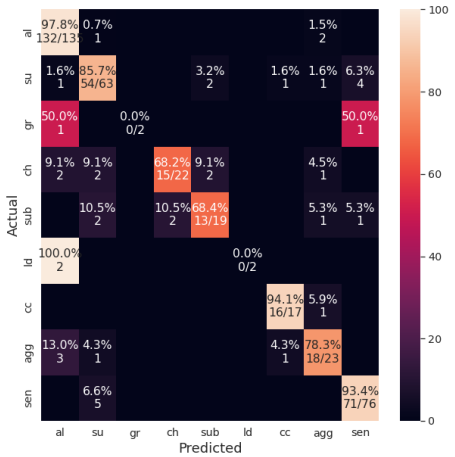


Figure 7.9: Confusion Matrix for RF Classifier Using Embeddings Feature Set of Set A.

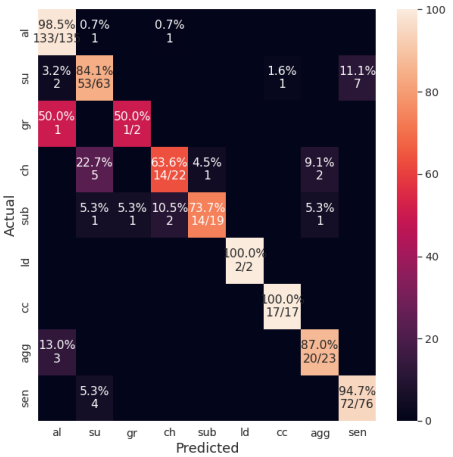


Figure 7.10: Confusion Matrix for SVM Classifier Using Embeddings Feature Set of Set A.

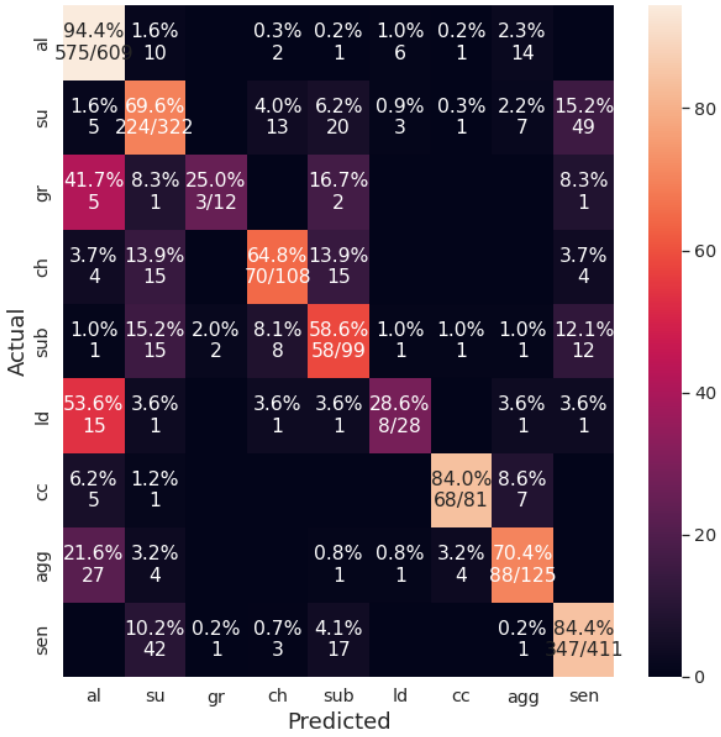


Figure 7.11: Summed Confusion Matrix of The Folds for CNN Classifier Using Set A.

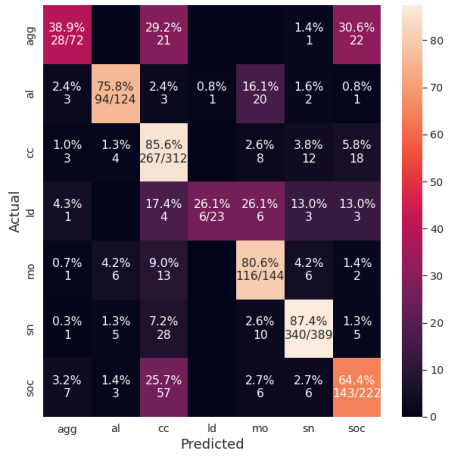


Figure 7.12: Confusion Matrix for RF Classifier Using eGeMAPS Feature Set of Set B.

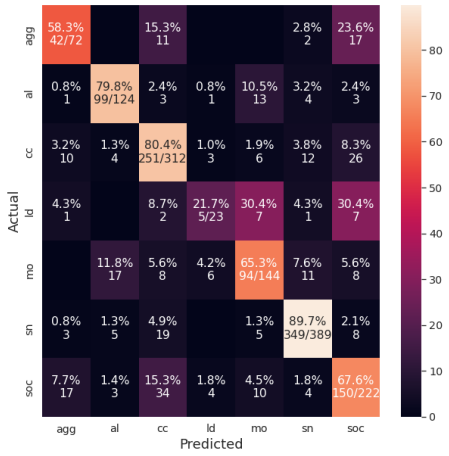


Figure 7.13: Confusion Matrix for SVM Classifier Using eGeMAPS Feature Set of Set B.

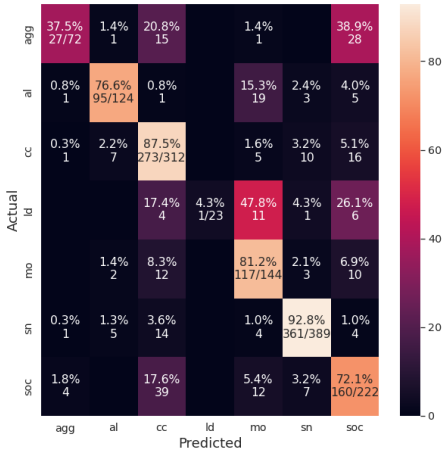


Figure 7.14: Confusion Matrix for RF Classifier Using ComParE Feature Set of Set B.

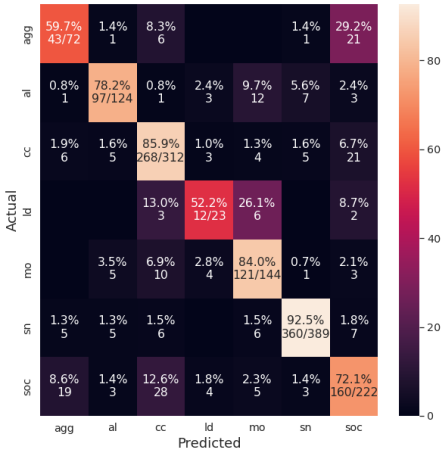


Figure 7.15: Confusion Matrix for SVM Classifier Using ComParE Feature Set of Set B.

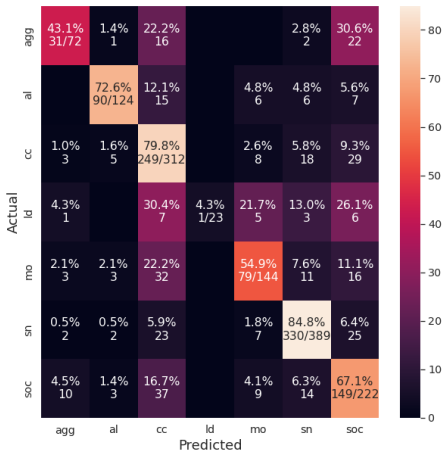


Figure 7.16: Confusion Matrix for RF Classifier Using catch22 Feature Set of Set B.

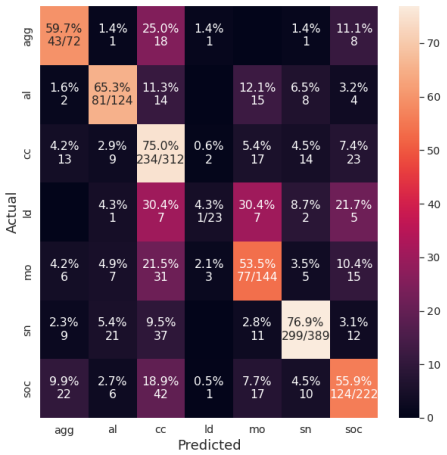


Figure 7.17: Confusion Matrix for SVM Classifier Using catch22 Feature Set of Set B.

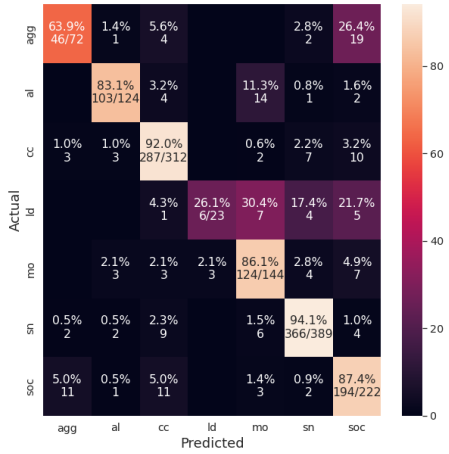


Figure 7.18: Confusion Matrix for RF Classifier Using Embeddings Feature Set of Set B.

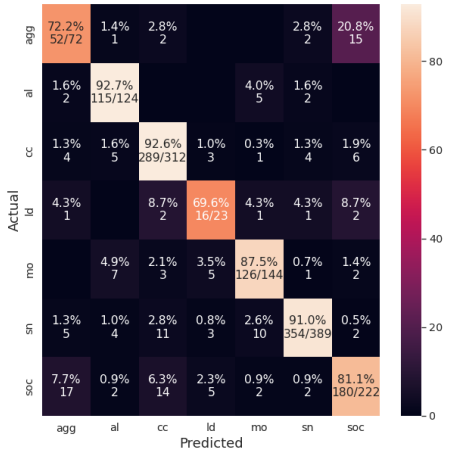


Figure 7.19: Confusion Matrix for SVM Classifier Using Embeddings Feature Set of Set B.

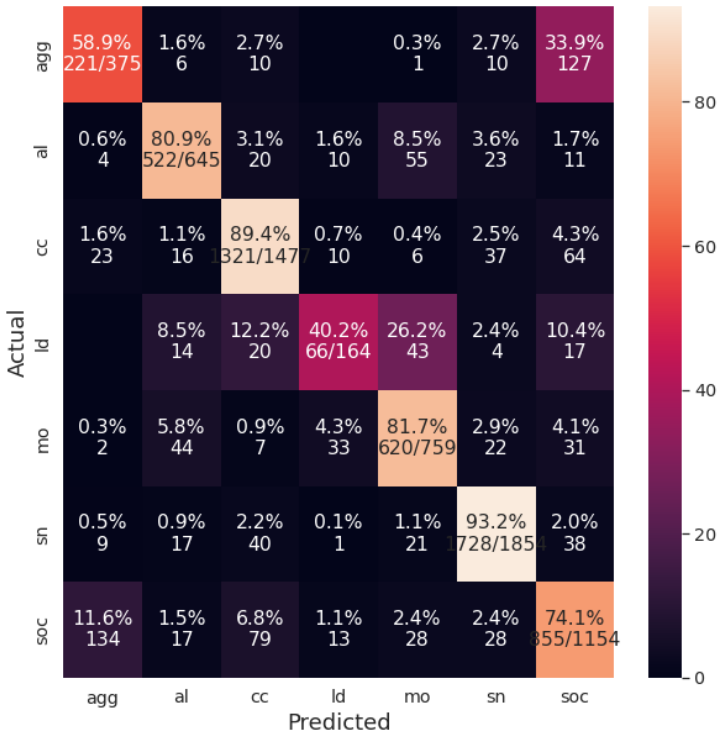


Figure 7.20: Summed Confusion Matrix of The Folds for CNN Classifier Using Set B.

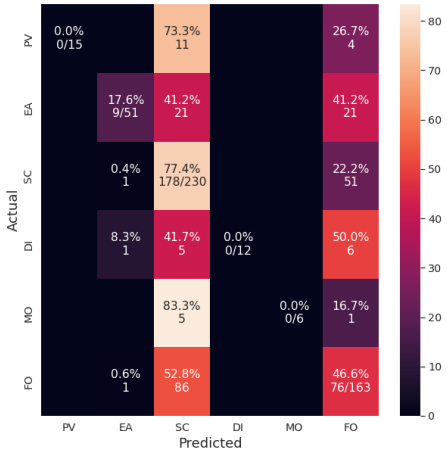


Figure 7.21: Confusion Matrix for RF Classifier Using eGeMAPS Feature Set of Set C.

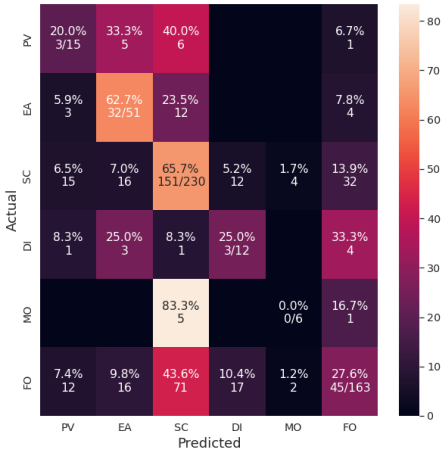


Figure 7.22: Confusion Matrix for SVM Classifier Using eGeMAPS Feature Set of Set C.

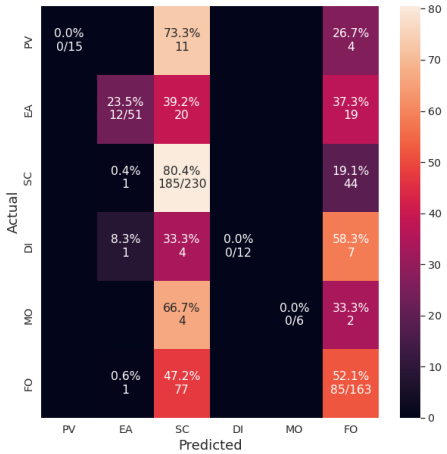


Figure 7.23: Confusion Matrix for RF Classifier Using ComParE Feature Set of Set C.

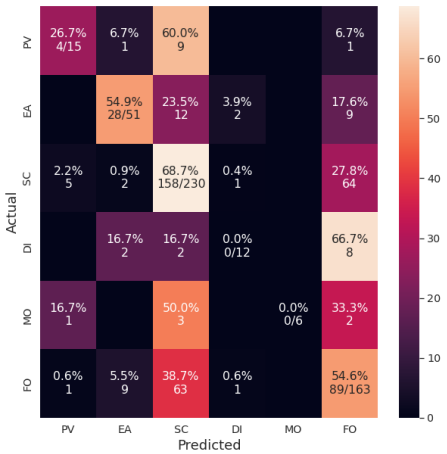


Figure 7.24: Confusion Matrix for SVM Classifier Using ComParE Feature Set of Set C.

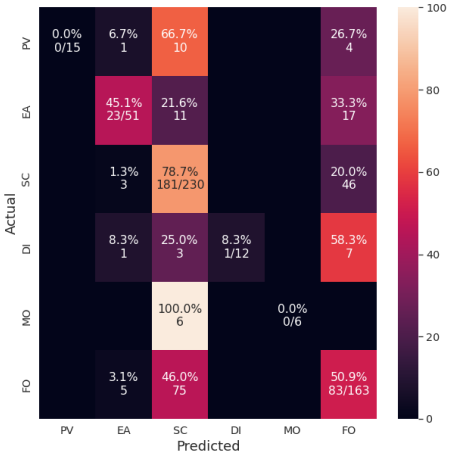


Figure 7.25: Confusion Matrix for RF Classifier Using catch22 Feature Set of Set C.

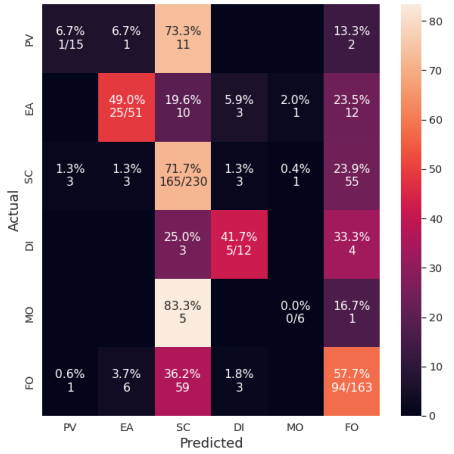


Figure 7.26: SVM classifier confusion matrix of catch22 feature set of Set C.

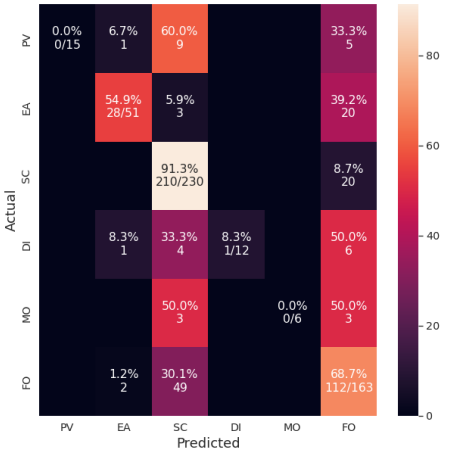


Figure 7.27: Confusion Matrix for RF Classifier Using Embeddings Feature Set of Set C.

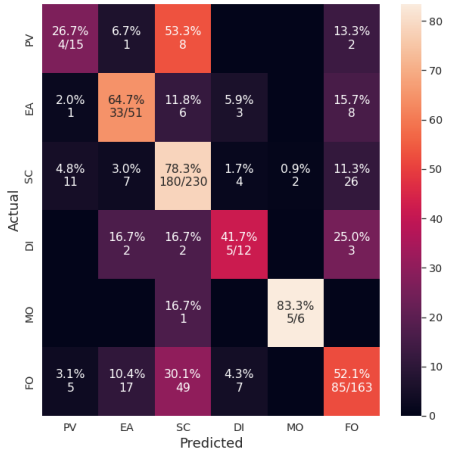


Figure 7.28: Confusion Matrix for SVM Classifier Using Embeddings Feature Set of Set C.

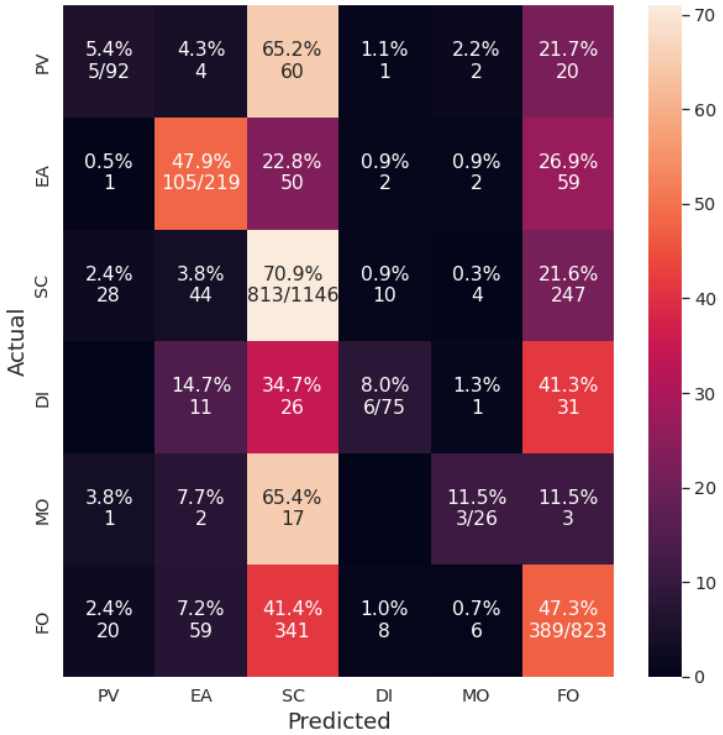


Figure 7.29: Summed Confusion Matrix of The Folds for CNN Classifier Using Set C.