Acoustic Signal Analysis and Feature-Based Classification of BOAS

For the Health and Welfare of Brachycephalic Dogs

Master's thesis in Biomedical Engineering

JENNIE BERNDTSON

Department of Physics

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Acoustic Signal Analysis and Feature-Based Classification of BOAS

For the Health and Welfare of Brachycephalic Dogs

JENNIE BERNDTSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Division of Material Physics
Chalmers University of Technology
Gothenburg, Sweden 2025

Acoustic Signal Analysis and Feature-Based Classification of BOAS
For the Health and Welfare of Brachycephalic Dogs
JENNIE BERNDTSON

Supervisor & Examiner: Magnus Karlsteen, Department of Physics

Master's Thesis 2025
Department of Physics
Division of Material Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A close-up of a French bulldog, CC0 1.0 [1].

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Acoustic Signal Analysis and Feature-Based Classification of BOAS
For the Health and Welfare of Brachycephalic Dogs
JENNIE BERNDTSON
Department of Physics
Chalmers University of Technology

Abstract

This thesis examines a feature-based approach for classifying Brachycephalic Obstructive Airway Syndrome (BOAS) in dogs using acoustic signal analysis and machine learning. Audio recordings of dogs breathing, collected both before and after physical exercise, were preprocessed through normalization, filtering, and data augmentation techniques to enhance signal quality. Features were extracted using the openSMILE toolkit and refined through statistical tests, notably the Mann-Whitney U-test, to identify those most indicative of BOAS severity. Two modeling strategies were employed: separate classifiers for pre- and post-exercise recordings and a hybrid model that incorporates both. The hybrid model, trained using decision tree-based methods including Random Forest and XGBoost, demonstrated superior performance, achieving an AUC of 1.0 and an average prediction confidence of 88.5% when evaluated on an unseen dataset of five dogs. Although more data is needed to ensure the model's reliability and generalization to unseen data, these findings highlight the potential of a feature-based tool as a practical and accessible option for BOAS classification, thereby improving the health and welfare of brachycephalic dogs.

Acknowledgments

First and foremost, I would like to thank Johan Thorell, a Licensed Veterinarian at Hallands Djursjukhus Slöinge, for allowing me to participate in his functional grading tests with four French Bulldogs. I also appreciate Gunilla Mattsson, Clinic Manager, and Henrik Hedberg, Licensed Veterinarian at Viskadalens Djurklinik Evidensia, for allowing me to participate in their grading test of a pug. Also, a big thanks to the dog owners for allowing me to collect the data. This project would not have been the same without your willingness to contribute to this research.

Thank you to my supervisor and examiner, Magnus Karlsteen, for all the support, helpful advice, and encouragement throughout the project. Also, thanks to Tim Pagrell for being such a great sounding board while working on a similar thesis.

Special thanks to Isabella Sykkö for her support when starting the project and her earlier work in gathering data, which provided a great starting point for this project. Finally, thanks to Maria Dimopoulou at SLU for your important work in improving the health of brachycephalic dogs, and for your help in gathering the dogs and data that made this project possible.

Jennie Berndtson, Gothenburg, June 2025

Definitions & Acronyms

BOAS	Brachycephalic Obstructive Airway Syndrome
Brachycephalic	Means shortened head, used to describe dog breeds with a flat face
RFG-Scheme	Respiratory Function Grading Scheme
BOAS-negative	When the result from the RFG-Scheme is 0 or 1
BOAS-positive	When the result from the RFG-Scheme is 2 or 3
ROC	Receiver Operating Characteristic
AUC	Area Under the ROC
FFT	Fast Fourier Transform
RMS	Root Mean Square
openSMILE	A toolkit used for extracting audio features
XGBoost	Extreme Gradient Boosting

Contents

List of Figures

List of Tables

1

Introduction

Brachycephalic obstructive airway syndrome (BOAS) is a chronic and lifelong pathological condition that significantly impairs breathing and reduces the quality of life in many popular dog breeds [2]. BOAS primarily affects brachycephalic breeds, those with characteristically shortened skulls, such as Bulldogs, French Bulldogs, Pugs, and Boston Terriers [3], [4]. These breeds are increasingly popular among dog owners, particularly in Western countries.

Several reports have documented a marked rise in the popularity of brachycephalic breeds over recent years. For example, a UK pet insurance company reported that registrations of French Bulldogs alone have increased by more than 500% over the past decade [5]. As the demand for these breeds increases, so does the number of affected dogs, resulting in a corresponding rise in cases of BOAS and associated health complications. This highlights the urgent need for accessible and reliable methods to assess the severity of this condition.

To address this, the University of Cambridge has developed a Respiratory Function Grading Scheme (RFG-Scheme) in which a dog undergoes a comprehensive veterinary examination [6]. The evaluation includes an auditory assessment of the dog's breathing before and after a three-minute trotting exercise. Based on the findings, the dog is assigned a severity grade ranging from 0 to 3. This grading system has proven effective in identifying the severity of BOAS and guiding treatment decisions.

However, when this project was proposed approximately five years ago, the Cambridge Functional Grading Test faced a significant accessibility barrier. At that time, only veterinarians who had completed a certification course at the University of Cambridge were authorized to conduct the test. This requirement meant that very few professionals in each country were qualified to perform the assessment, thereby limiting its practical use on a broader scale. Also, as of early 2025, the Swedish Kennel Club requires all brachycephalic dogs to pass this test to be eligible for breeding, further increasing the demand for accessible testing solutions [7]. This presented a clear gap in diagnosing BOAS.

This project aims to address this gap by developing a classification system for BOAS-related breathing sounds in dogs, modeled on the principles of the Cambridge grading scale but designed to operate independently of a certified veterinarian. By increasing accessibility to BOAS assessment, this approach could contribute to earlier diagnosis and, ultimately, improved welfare for brachycephalic dogs.

The expected outcome of this project is a feature set that effectively represents the relevant characteristics of the breathing data, enabling a model to classify dogs as BOAS-negative or BOAS-positive with high probability and reliability.

1.1 Background

In some moderately affected dogs, clinical signs of BOAS may not be apparent while the dog is at rest. To more accurately assess the condition, the functional grading test is used [6]. This test provides a standardized evaluation of the dog's respiratory function under light physical stress.

The procedure begins with the veterinarian listening to the dog's breathing using a stethoscope while the dog is at rest. The dog is then assigned to perform a light exercise, typically a trotting task intended to cover 400 meters in three minutes, which corresponds to a light run. Immediately following the exercise, the veterinarian listens again to the dog's breathing. Additionally, the veterinarian observes the dog's nostrils for signs of stenosis (narrowing). However, the appearance of the nostrils is not included in the final grading. According to Maria Dimopoulou, a doctoral candidate and clinical veterinarian at the Department of Clinical Sciences at the Swedish University of Agricultural Sciences (SLU), the nostrils may appear narrower when a dog is stressed, such as during a clinical evaluation, making the nostrils an unreliable indicator in this context.

As mentioned, the dog's condition is then graded from 0-3 in the RFG-Scheme, and the Department of Veterinary Medicine at Cambridge University defines the BOAS grades as [6]:

- Grade 0 - BOAS free; annual health check is suggested if the dog is under 2 years old.
- Grade 1 - clinically unaffected but with mild respiratory signs, an annual health check is suggested if the dog is under 3 years old.
- Grade 2 - BOAS affected with moderate respiratory signs. The dog has a clinically relevant disease and requires management, including weight loss and/or surgical intervention.
- Grade 3 - BOAS affected with severe respiratory signs. The dog should undergo a thorough veterinary examination, including possible surgical intervention.

Where grades 0 and 1 are considered BOAS-negative, and grades 2 and 3 are BOAS-positive.

Dogs showing symptoms should undergo a comprehensive veterinary evaluation, which may include surgical intervention [8]. However, diagnosing and grading BOAS can be challenging due to limited access to specialized veterinarians and the subjective nature of assessments, which depend heavily on the veterinarian's expertise and experience. Therefore, a classification model is needed to provide veterinarians with valuable support in their decision-making process.

1.2 Aim

This project aims to collect audio data from brachycephalic dogs and apply signal processing techniques to extract features that strongly correlate with the breathing characteristics of phone recordings from BOAS-negative and BOAS-positive dogs. These features will then be used to train a classification model to determine whether a dog is BOAS-negative or BOAS-positive.

1.3 Related Work

There have been two previous master's theses focused on classifying BOAS using machine learning [9], [10]. Both projects aimed to develop models based on spectrogram representations of audio signals using a shared dataset recorded with a dictaphone and a digital stethoscope. However, neither study applied extensive signal processing to enhance the signal characteristics, and they trained separate models for pre- and post-exercise recordings. Both reported poor performance on the pre-exercise data. To address this limitation, this thesis examines whether a hybrid model can outperform separate models, particularly for the challenging pre-exercise data. Both also reported problems with overfitting, which is common when having small datasets. Therefore, decision tree classifiers are used as they reduce the risk of overfitting [11].

Building on these theses, researchers from Chalmers and SLU investigated the use of signal analysis for BOAS severity assessment [12]. Using digital stethoscope recordings, they extracted seven features from frequency-transformed audio segments and evaluated them with ANOVA and ROC curves. Their work emphasized feature-based analysis rather than model complexity. This thesis builds on that direction by focusing on signal preprocessing and feature extraction to identify parameters that better capture the distinctions between BOAS-negative and BOAS-positive dogs. Unlike the earlier theses, spectrograms will not be used.

More recently, Isabella Sykkö at Chalmers introduced the use of smartphones for audio recording, replacing the earlier hardware tools. The main part of the dataset used in this thesis was collected during her collaboration with Dimopoulou at SLU. Sykkö applied basic feature extraction and simple models, such as logistic regression, and developed a prototype smartphone app for practical use. However, neither her features nor the app are included in this work.

Another related scientific article used the `openSMILE` toolkit for audio feature extraction [13]. OpenSMILE will also be the primary tool for feature extraction in this thesis.

Finally, a closely related work is Tim Pagrell's master's thesis, where he aims to develop a similar model [14]. However, in addition to a feature-based approach, he also investigates the use of spectrograms to represent the data, a method similar to that employed in previous master's thesis works. He is also investigating the most appropriate machine learning model for the intended task.

1.4 Limitations

This project does not aim to compare different machine learning models in depth. However, it will include the use of two decision tree-based classifiers and a brief discussion of other models that may be suitable for this type of data.

The dataset used in this thesis will not include audio recordings from earlier theses, as those were not recorded using smartphones. Instead, the primary dataset will consist of recordings collected during Sykkö's work, supplemented by a small number of new recordings gathered during this project.

The prototype app developed in earlier work will not be used or further developed as part of this thesis.

1.5 Research Questions

List of questions that the thesis will answer:

- What signal processing methods enhance the breathing characteristics of a phone recording of a BOAS-negative and BOAS-positive dog?
- Which features show the strongest correlation with BOAS-negative and BOAS-positive dog samples?
- Is it more suitable to use separate models for pre- and post-exercise recordings for the intended task, or a hybrid model that incorporates both?
- Do the audio recordings differ between phones?

2

Theory

This section provides the theoretical background for the concepts and methods employed throughout the thesis. It covers signal processing tools, the feature extraction method, statistical tests, and relevant information on the machine learning techniques applied in the study.

2.1 Signal Processing

This section provides information on normalization methods, filters, and data augmentation.

2.1.1 Normalization

Normalization of audio refers to applying a constant audio gain to an audio recording by altering the signal amplitude for all values in the signal [15]. This can be performed through Peak normalization or Loudness normalization.

Peak normalization involves scaling the signal based on its loudest point (peak) [15]. A standard method, which many people associate with normalization, involves adjusting the audio so that its maximum absolute value equals 1. Peak normalization does not account for the loudness of the signal, which varies with frequency and duration.

Loudness normalization adjusts the average loudness of the signal to a target level by adjusting the gain [15]. One approach to estimating the average loudness of the signal is to determine its average power, such as the root mean square (RMS) amplitude. Using the RMS amplitude, the signal can be scaled to a desired level.

2.1.2 Data Augmentation

To address variability in loudness caused by differences in measurement distance and the recording properties of various phones, this project also applies data augmentation. Data augmentation is a widely used technique in machine learning, particularly when working with small datasets [16]. It involves a range of methods to expand the dataset and enhance the diversity of the data, which can help reduce overfitting and improve the model's performance. This project uses data augmentation to simulate the variability introduced by recording with different phone devices. Thus, it

serves more as an alternative to normalization than a strategy for improving model performance.

2.1.3 Filtering

The filters used to preprocess the audio signals are frequency-selective, specifically high-pass and low-pass filters. A high-pass filter enhances high frequencies by filtering out lower frequencies, while a low-pass filter enhances low frequencies by filtering out higher ones [17]. In this project, Butterworth low-pass and high-pass filters are used. The Butterworth filter is well-suited for this purpose because it has no ripple in either the passband or the stopband, which is a desirable property for frequency-selective filtering [17], [18]. It, therefore, preserves the desired parts of the signal without distortion and cleanly removes the undesired frequencies.

2.2 Feature Extraction Using OpenSMILE

OpenSMILE is a toolkit for audio feature extraction designed for applications in speech, music, and general sound recognition [19], [20]. It processes raw audio signals and extracts a wide range of features. This project uses the ComParE 2016 feature set from the openSMILE library. This set extracts 6 373 features derived from both the time and frequency domains, including, for example, signal energy, loudness, Mel-frequency cepstral coefficients (MFCCs), pitch, and voice quality [20]. A comprehensive description of all features is available in the openSMILE documentation [20]. OpenSMILE provides an extensive, high-dimensional feature set. Therefore, feature reduction is necessary to identify the most relevant features for this type of audio [21]. Feature reduction is performed using a statistical method described in the following section.

2.3 Statistical Tests

Statistical tests can be used to evaluate the extracted features and determine whether variables are correlated. This section provides an overview of the main statistical tests applied in this thesis.

2.3.1 Pearson and Spearman

The Pearson product-moment correlation coefficient measures the strength and direction of the linear relationship between two variables [22]. The correlation coefficient ranges between -1 and 1, where values closer to 1 indicate a strong positive linear relationship, values closer to -1 indicate a strong negative linear relationship, and values near 0 suggest little to no linear relationship. The Pearson correlation coefficient, therefore, describes how closely the data points align with an imaginary straight line. The closer the points are to this line, the closer the coefficient is to ± 1, and the stronger the linear correlation between the two variables.

The Spearman's rank-order correlation coefficient is a non-parametric counterpart to the Pearson product-moment correlation coefficient [23]. Like Pearson's, it measures the strength and direction of the relationship between two variables, but instead of focusing on linear relationships, it captures monotonic relationships. A monotonic relationship means that as the value of one variable increases or decreases, the value of the other tends to do the same, though not necessarily at a constant rate. The resulting coefficient, like Pearson's, ranges between -1 and 1 and is interpreted in the same way; values closer to ± 1 indicate a stronger monotonic relationship, while values closer to 0 indicate little to no monotonic relationship.

2.3.2 Mann-Whitney U-test

The Mann–Whitney U-test is a non-parametric statistical test used to compare two independent samples or groups [24], [25]. It is beneficial for assessing differences between groups when the data are continuous but not normally distributed [24]. Often described as the non-parametric equivalent of the t-test, it does not assume normality and instead relies on the ranks of the data rather than their raw values. All observations from both groups are combined and ranked, and the test assesses whether the sum of ranks differs significantly between the two groups.

The null hypothesis of the Mann–Whitney U-test states that the two populations are equal, meaning there is no difference in the distribution of ranks between the samples. To evaluate whether this hypothesis can be rejected, the p-value is calculated. If the p-value exceeds the significance level (typically set at 0.05), we do not reject the null hypothesis [25]. This analysis aims to identify features that differ significantly between the two groups: BOAS-negative and BOAS-positive. Therefore, we focus on features with p-values less than 0.05.

2.4 Machine Learning

This section includes information about the machine learning models and evaluation metrics used for the classification.

2.4.1 Models

A Random Forest classifier consists of multiple individual decision trees, as illustrated in Figure 2.1 [11]. Each decision tree is trained on a random subset of the dataset and predicts the class label, in this case, either Class 0 (BOAS-negative) or Class 1 (BOAS-positive). The final classification is obtained by taking the majority vote of the predictions from each decision tree. Additionally, the Random Forest classifier provides a probability estimate for each class, calculated as the proportion of trees that predicted that class out of the total number of trees. This probability reflects the model's confidence in its prediction.

For example, suppose decision tree 1 predicts Class 0, while decision trees 2 and 3 predict Class 1. Since the majority of the trees (2 out of 3) predict Class 1, the final predicted class will be Class 1. The prediction probability for Class 1, in this case,

is 2/3. The Random Forest classifier, therefore, improves predictive accuracy and reduces the risk of overfitting [11].

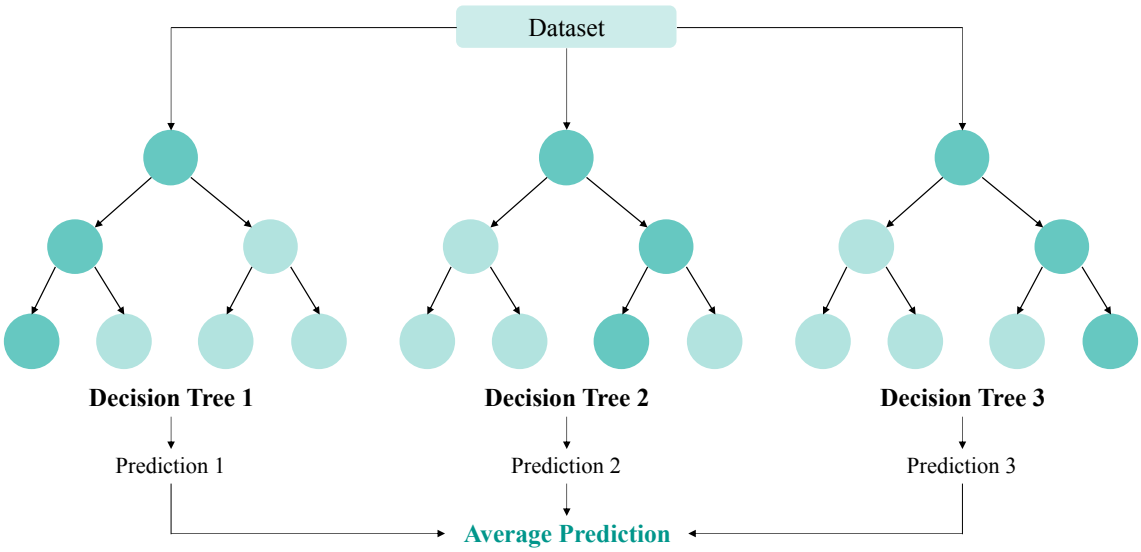


Figure 2.1: Architecture of a Random Forest Model.

Another model tested for classification is Extreme Gradient Boosting (XGBoost). XGBoost is a learning algorithm based on gradient boosting [26]. Unlike Random Forest, which builds trees independently in parallel, gradient boosting allows XGBoost to build decision trees sequentially, where each new tree tries to correct the errors made by the previous ones [27]. This gradient-boosting approach allows XGBoost to achieve high predictive performance.

2.4.2 Evaluation Metrics

The Receiver Operating Characteristics (ROC) plot is a popular measure used for evaluating the performance of a classifier, especially in medical classification problems [28], [29]. The ROC curve is a graphical plot with the False Positive Rate (FPR) on the x-axis and the True Positive Rate (TPR) on the y-axis, as shown in Figure 2.2. The FPR represents the proportion of negative observations that are incorrectly classified as positive [29]. Similarly, the TPR represents the proportion of positive observations that are correctly classified.

Given the ROC curve, the area under the curve (AUC) can be derived, as shown by the blue area in Figure 2.2. AUC is a useful tool for differentiating between classifiers, as it summarizes each classifier's performance into a single measure [30]. An AUC of approximately 0.5 indicates that the model has no class separation capacity, while a value of 1.0 means that the model has perfectly differentiated between the classes, with no false positives or false negatives [30], [31]. If an AUC of 1.0 cannot be achieved, an AUC above 0.8 is often considered acceptable [32].

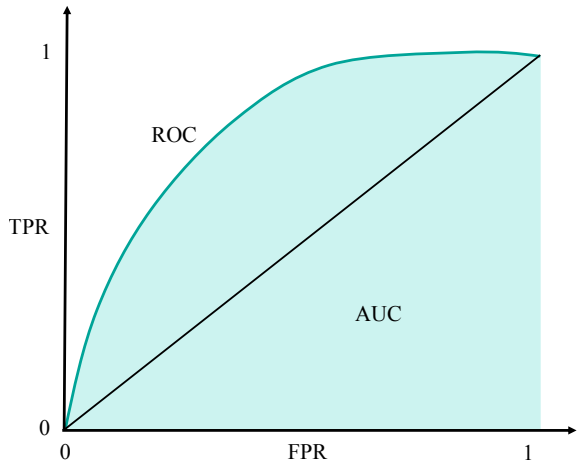


Figure 2.2: The ROC curve (blue line) shows the trade-off between TPR and FPR. The shaded area represents the AUC.

Accuracy is used to evaluate the classifier when using cross-validation. Classifier accuracy is determined by:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}.$$

Prediction probability represents how confident the Random Forest classifier is in its decision, based on the averaged outputs from all decision trees. This probability is used as the evaluation metric when classifying the new dataset, providing not only a predicted class but also a measure of certainty behind that prediction.

2.4.3 Cross-validation

Cross-validation is a technique used in machine learning to obtain a more reliable estimate of a model's performance. In traditional approaches, the dataset is randomly divided into separate training and test subsets. In k-fold cross-validation, the dataset is split into k equal-sized folds [33]. The model is trained k times. For training, it uses $k-1$ folds, with the remaining fold used for testing. This process ensures that every data point is used for training and evaluation. An illustration of this procedure is shown in Figure 2.3.

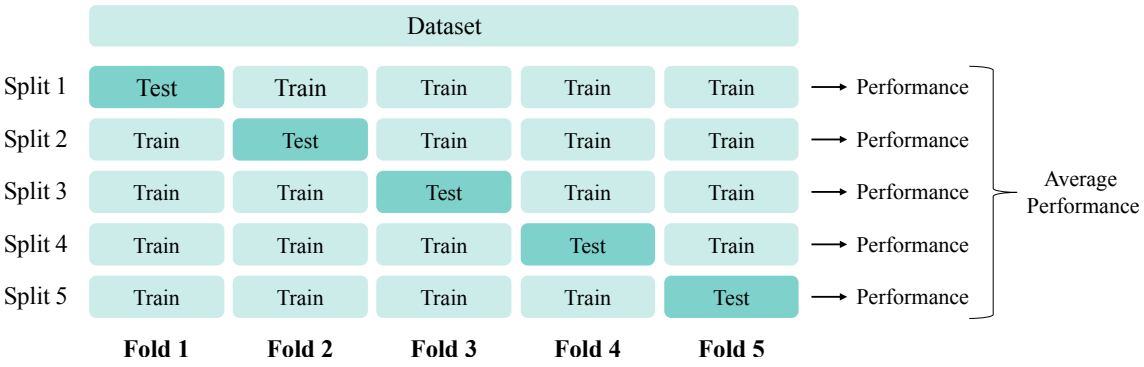


Figure 2.3: Illustration of k-fold cross-validation.

In this project, Stratified k-fold cross-validation is used, as the dataset is imbalanced, with more BOAS-negative dogs than BOAS-positive dogs. Stratification ensures that each fold maintains the same class distribution as the whole dataset, which helps produce more consistent and representative performance estimates, particularly for classification tasks involving class imbalance [34].

3

Methods

This thesis aimed to develop a model capable of classifying whether a dog is BOAS-negative or BOAS-positive based on two audio recordings of its breathing. The problem was addressed using the pipeline illustrated in Figure 3.1 to identify features that effectively capture the differences between BOAS-negative and BOAS-positive audio signals. The methodology consisted of five main stages: data collection, preprocessing, feature extraction, feature reduction using statistical methods, and classification-based evaluation.

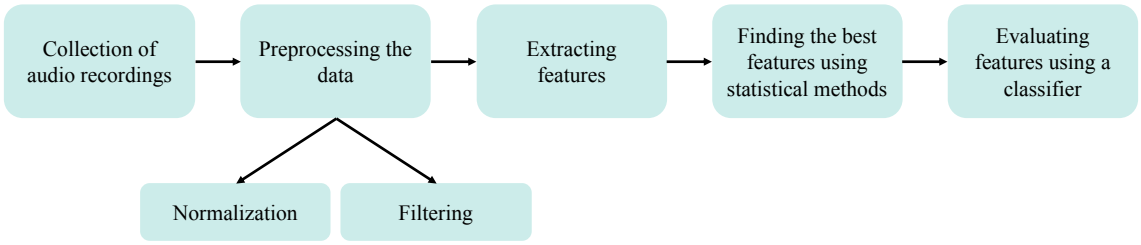


Figure 3.1: Pipeline of the methodology.

The first step involved collecting audio recordings. Although a set of recordings from previous studies was available, additional data were needed to test the model's performance in a real-world setting and to enhance the robustness of the evaluation. Therefore, new recordings were collected from a group of dogs to supplement the existing dataset.

In the preprocessing step, raw audio recordings are refined through normalization and filtering techniques to enhance the signal quality and ensure consistency. Next, in the feature extraction stage, a broad set of acoustic features was extracted from the preprocessed signals. To identify the most informative features, statistical methods were then applied, narrowing down the feature set to those most relevant for distinguishing between BOAS-negative and BOAS-positive cases. Finally, these selected features were evaluated using a classifier, providing insight into the quality and discriminative ability of the features.

Since the final evaluation depended on the classification performance, it was essential to carry out all preceding steps, including preprocessing, to assess the impact of each methodological choice, such as different normalization and filtering strategies, on the overall performance.

The final code is available on GitHub, including all functions and files mentioned in the following sections.

3.1 Audio Recording

The first dataset consisted of 85 audio recordings collected from 31 dogs representing various brachycephalic breeds. Table 3.1 shows the distribution of the dogs along with their corresponding BOAS grades. Several dogs had multiple recordings, including both pre- and post-exercise samples. The recordings were captured using three different Android phones: OnePlus A5000, Samsung A5, and Samsung S10. All dogs included in the dataset were over one year of age and free from respiratory diseases other than BOAS.

Table 3.1: Distribution of dogs by BOAS grade in the first dataset.

BOAS Functional Grading	Number of dogs
Grade 0	6
Grade 1	14
Grade 2	10
Grade 3	1
Total	31

To improve the model's performance, new data were collected to form a smaller supplementary dataset. A total of five dogs were recorded, comprising one Pug and four French Bulldogs. The Pug was recorded at Viskadalens Djurklinik Evidensia, while the four Bulldogs were recorded at Hallands Djursjukhus Slöinge. Recordings were taken both before and after the 3-minute exercise test. Airway evaluations were conducted by authorized veterinarians using the validated RFG scheme. The audio was recorded simultaneously using two different mobile phones, a Samsung Galaxy Note 20 Ultra and a OnePlus A5000. Before all recordings, the dog owners signed a written informed consent allowing us to process the data.

The file names of each recording in the dataset, along with its BOAS grade and a definition of whether it was a pre- or post-exercise recording, were saved in a Metadata CSV file to easily extract the desired files using the Python library `Pandas` [35].

3.2 Data Preprocessing

To begin processing the audio recordings, they were first imported into Python using the `Soundfile` module, which reads audio files and returns both the audio data and its corresponding sample rate [36]. An if-statement was added to ensure that if the audio is recorded in a two-channel configuration, the code processes only one channel. Based on the sample rate information, a time axis was then constructed, allowing the audio recordings to be represented as time series data for subsequent preprocessing steps.

The new data that was recorded using the Samsung was initially stored in the M4A format, whereas the new OnePlus and earlier recordings were in WAV format. The M4A files were therefore converted to WAV format to ensure consistency across the dataset before further processing.

3.2.1 Normalization

To account for variabilities between audio recordings, such as differences in the distance between the microphone and the dog during recording, normalization was applied to the signals. Two normalization approaches were tested and evaluated to determine the most suitable method for this data.

The first normalization approach was peak normalization. When performing this, the signals were scaled based on their maximum amplitude so that the waveform was within the range of -1 to 1. The audio is loaded, and Peak normalized using the function `read_and_normalize_file`.

Another normalization method was Loudness normalization, in this case, RMS normalization. In this approach, each signal was scaled by a factor calculated as the signal's RMS value plus 0.01 to avoid scaling the signal by zero depending on the RMS value. The audio is loaded, and RMS normalized using the function `load_and_RMS_normalize`.

Another approach tested for addressing the variability between audio recordings was data augmentation, which is described in Section 3.2.3.

The model outputs were analyzed after each technique was applied to evaluate the performance of the normalization methods. Plots were generated to investigate further the differences between peak normalization and loudness (RMS) normalization, showing both the original signal and its corresponding normalized signal. These plots were created for two representative cases: one dog with BOAS grade 0 and very quiet breathing and another with BOAS grade 2 exhibiting noticeably loud breathing in the audio recordings. This analysis was performed because it is essential that the model can accurately differentiate between the breathing sounds of BOAS-negative and BOAS-positive dogs.

3.2.2 Filtering

The audio signals were filtered using high-pass and low-pass Butterworth filters to reduce background noise and other disturbances introduced by phone recordings. The high-pass filter was applied to remove low-frequency components, such as wind noise, that are unlikely to contribute useful information for classification. Similarly, the low-pass filter was used to eliminate high-frequency components, such as background sounds, which were irrelevant for distinguishing breathing patterns.

To determine appropriate cutoff frequencies for the filters, several representative recordings were cropped to isolate specific sounds. This was performed in file `cropped_data.py`. These included breaths recorded before and after exercise, non-breathing noises such as wind noise from the dog's exhalation into the micro-

phone, and background sounds like a door opening. Each sound segment was analyzed in the frequency domain using the Fast Fourier Transform (FFT) in the file `cropped_data_fft_plot.py`. This analysis helped identify the frequency ranges where important breathing-related sounds occur and distinguish them from irrelevant or disruptive frequency components that should be filtered out.

To determine an appropriate cutoff frequency for the high-pass filter, the signals were initially high-pass filtered using a range of fixed cutoff frequencies between 80 Hz and 1000 Hz.

To further improve it, I implemented a dynamic filtering approach that consists of a function determining an estimated cutoff for each particular signal. The function is named `estimate_cutoff_frequency` and analyzes each signal's frequency spectrum by first converting the time-domain signal using the FFT. The function then calculates the cumulative energy distribution of the frequency spectrum and determines the frequency at which a specified percentage of the total signal energy is exceeded. This determined frequency is then used as the cutoff for the high-pass filter. Therefore, if the threshold is set to 5%, the function selects a cutoff frequency such that the lowest-frequency components, which contain the first 5% of the signal's total energy, are removed. This approach allows the filtering to adapt to the characteristics of each recording. The threshold were set to 5%, 10%, 15% and 20% to determine an appropriate cutoff frequency threshold for the high-pass filter.

To determine an appropriate cutoff frequency for the low-pass filter, the signals were processed using a range of fixed cutoff frequencies between 14000 Hz and 5000 Hz. The methodology was also adapted for low-pass filtering using the same dynamic filtering function developed for the high-pass filter. In this case, the cumulative energy distribution of the frequency spectrum was used to identify the frequency above which a specified percentage of the total signal energy was exceeded. Thresholds of 90% and 95% were tested, resulting in cutoff frequencies that preserved the lower-frequency components carrying 90% and 95% of the signal's total energy, respectively, while removing the remaining high-frequency content. This allowed the low-pass filtering to adapt to the characteristics of each signal.

The signals were filtered using a digital low-pass and high-pass Butterworth filter from the `SciPy` library [37]. The functions used for high-pass and low-pass filtering are called `butter_highpass_filter` and `butter_lowpass_filter`, respectively.

3.2.3 Data Augmentation

Data augmentation was another approach used to address the variability between audio recordings rather than normalization. Data augmentation is performed not only for this purpose but also to improve the overall performance of the models. Therefore, it is described as a separate part of the preprocessing methodology rather than a part of normalization.

The data augmentation approach was used to randomly amplify or reduce the amplitude of the entire signal by a specified percentage. The augmentation was performed in the file `AUG_random_amplitude_up_or_down.py`. First, the signals were

randomly amplified or reduced by 10%. Then, they were randomly amplified by 15% and reduced by 5%. After performing this, along with the other desired preprocessing methodologies, the signals were saved, and features were extracted. These features were then saved as a DataFrame and concatenated on top of a similar feature set in file `Concat_CSV_files.py`, but that set did not have the random amplification as part of the preprocessing. This resulted in a dataset twice as large, augmented to account for the variability in amplitude between recordings taken with different phones.

3.3 Feature Extraction

Since the recordings are time series signals, feature extraction is necessary to capture and describe relevant patterns within the recordings in a form suitable for machine learning. After all signals were preprocessed using various signal processing methods, features were extracted using the `openSMILE` toolkit [38]. The extracted features were organized into DataFrames using Pandas and saved as CSV files for further analysis using the function `create_csv_with_OpenSMILE_features`.

3.4 Statistical Methods

After feature extraction, feature reduction was performed to identify the most relevant features. This step is important in machine learning, as it helps retain only the most informative features, thereby improving prediction accuracy and generalization to unseen data [21].

Using the feature DataFrame obtained in the previous section, three different statistical tests were applied to analyze the relationships between features and to select those that best distinguish between BOAS-negative and BOAS-positive dogs.

Spearman and Pearson correlation tests were applied to examine the relationships between individual features and the target label. Both tests output a correlation coefficient for each feature, and features with an absolute correlation coefficient greater than 0.6 were selected for further analysis. Both tests were applied in the file `Spearman_and_Pearson.py`.

In addition, the Mann–Whitney U-test was used to identify features that best differentiate between BOAS-negative and BOAS-positive dogs. Features with p-values less than 0.05 were considered statistically significant, as they indicate a difference between BOAS-negative and BOAS-positive. However, since this often resulted in a large number of features, only the top 100 features with the smallest p-values were retained. The implementation included a check to raise an error if fewer than 100 features met the threshold. In that case, all features with p-values less than 0.05 were selected for further analysis. The function `Mann_Whitney_U_test_csv` extracts these features and creates a CSV file.

3.5 Machine Learning

Once the feature set had been created, the final step was to train a machine learning model to evaluate its ability to classify the data correctly and test the various preprocessing settings. This was achieved using two different model setups, as described in the following section.

3.5.1 Model Setup

As mentioned in the introduction, the classifier was trained using two different setups, as shown in Figure 3.2. Setup 1 involves training two separate models: one using the pre-exercise recordings and the other using the post-exercise recordings. Setup 2 is a hybrid approach, trained on a combined feature set that includes both pre- and post-exercise recordings. When Setup 1 was performed, the dataset was split into two separate feature DataFrames: one for the pre-exercise recordings and one for the post-exercise recordings. Each of these datasets was processed independently following the methodology described in Figure 3.1. The file `Extract_Features_Setup_1.py` was developed to process all audio recordings by performing preprocessing based on user-defined parameters, extracting openSMILE features, reducing them using the Mann–Whitney U test, and generating the final feature set.

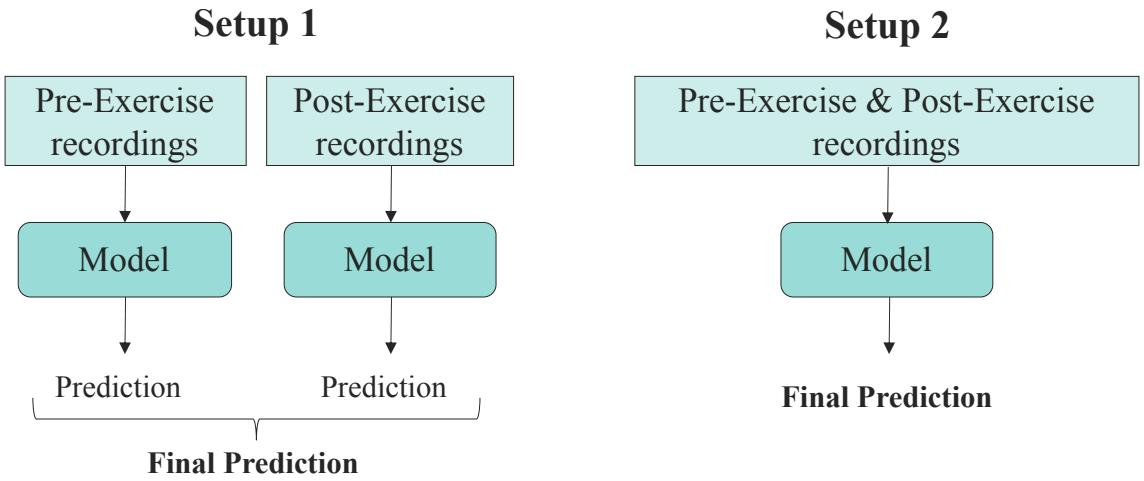


Figure 3.2: Processes when performing model Setup 1 and Setup 2.

In Setup 2, a hybrid model was constructed using a combined feature set. To do this, the original dataset was filtered to include only those dogs for which both pre- and post-exercise recordings were available. Since three dogs were missing one of the two recordings, there were only 28 dogs included when performing Setup 2. The same preprocessing and feature extraction methods were then applied to both subsets. After extracting features using openSMILE, the feature names in each DataFrame were renamed to indicate the recording context: pre-exercise features were renamed with the suffix `_before_et`, and post-exercise features with `_after_et`. The two feature DataFrames were then concatenated side by side using `Pandas`, resulting in a

single feature vector that included information from both recordings for each dog, in file `OpenSMILE_feature_extraction_setup_2.py`. This combined DataFrame was subsequently subjected to the statistical test feature selection, allowing the model to identify relevant features from both pre- and post-exercise recordings.

3.5.2 Training and Evaluating the Classifier

Before the new dataset was introduced, the initial work was done using the first dataset. After preprocessing the data, features were extracted and reduced using the Mann-Whitney U-test. The resulting feature set was then saved in a CSV file for use in training and evaluation.

The primary classifier used in this study was the Random Forest classifier. To prepare the data for training, the column representing each dog's BOAS grade (class 0 or class 1) was separated from the rest of the dataset and used as the model's label. All remaining columns were treated as input features, consisting of numerical features derived from previous steps.

To evaluate the model's performance, the dataset was split into five folds for cross-validation. The Random Forest model was trained and validated across these five splits, in file `Train_RF_using_Cross_Validation.py`. For each fold, accuracy and the AUC were recorded. The final performance was reported as the mean and standard deviation of these metrics. The Random Forest classifier and the Stratified k-folk cross-validation was applied using the `scikit-learn` library [39].

Once the new dataset had been collected, the best-performing methodology identified during the initial experiments was further tested and refined. The Random Forest classifier was still used, along with stratified 5-fold cross-validation, to account for the imbalanced dataset. The model was trained on the first dataset and then evaluated on the newly collected dataset to investigate how the model generalized to new data, in the file `Classify_new_files.py`. Besides accuracy and AUC, the performance was assessed by examining which dogs were correctly classified and the associated prediction probabilities for each dog. Based on the best-performing results, the same feature set was used to train an XGBoost model to evaluate whether the prediction probabilities could be improved, in the file `XGBoost.py`. The model was applied using the `XGBoost` Python package [40].

This methodology was initially applied to model Setup 1, where the pre- and post-exercise data were evaluated individually. The methodology was then extended to model Setup 2, which incorporated a hybrid solution that combined both pre- and post-exercise data for evaluation. Performing Setup 2, the tests were conducted separately with the audio recordings on the OnePlus A5000 and the Samsung Galaxy Note 20 Ultra separately to ensure that the methodologies worked for both devices.

4

Results and Analysis

This chapter presents the project's most relevant results, focusing on signal processing techniques, feature extraction, statistical methods, and classification approaches.

The Appendix provides additional detailed classification results for the two model setups, focusing on how different preprocessing strategies affect model performance. Tables A.1 and B.1 display classification results obtained using the Random Forest Classifier and Stratified 5-fold cross-validation for the first dataset, with different preprocessing settings applied to model Setups 1 and 2, respectively. The following tables in Appendices A and B present classification results obtained by training the classifier on the first dataset and using the new dataset as input.

Appendix A presents results using model Setup 1, where the classification is divided between pre- and post-exercise data. Tables are presented in pairs, where each preprocessing condition is evaluated separately on pre- and post-exercise recordings. For instance, Tables A.2 and A.3 show classification results without preprocessing for pre- and post-exercise data, respectively. Similarly, Tables A.5 and A.6 present results obtained by applying a high-pass filter with a 5% cutoff to the pre- and post-exercise recordings. The individual classification probabilities for each recording type are further merged in Tables A.4 and A.7 to provide an average performance estimate across the pre- and post-exercise recordings.

Appendix B presents the classification results for Model Setup 2, initially using OnePlus recordings, followed by Samsung recordings. It begins with baseline results using raw data (Table B.2) and progressively adds complexity through RMS normalization (B.3) and high-pass filtering with an estimated cutoff frequency of varying thresholds 5–20% in Tables B.4 to B.7. Further refinement is introduced in Tables B.8 to B.12, which incorporate data augmentation, including both symmetric ($\pm$10%) and asymmetric (+15%, -5%) amplitude adjustments, alongside filtering and normalization, to investigate their cumulative impact on model generalization. Tables B.13 to B.16 present results obtained by applying selected preprocessing strategies to Samsung recordings, comparing the differences between phone recordings. The final Tables (B.17 to B.26) enable phone-recording comparisons (OnePlus vs. Samsung) under matched preprocessing conditions, offering insights into how preprocessing settings transfer across different phones.

4.1 Audio Recording

Table 4.1 presents the distribution of BOAS grades based on the RFG results for the five dogs included in the new dataset. Four dogs were classified as BOAS-negative, while one was classified as BOAS-positive.

Table 4.1: Distribution of dogs by BOAS grade in the new dataset.

BOAS Functional Grading	Number of dogs
Grade 0	1
Grade 1	3
Grade 2	1
Grade 3	0
Total	5

4.2 Signal Processing

This section presents key results from the signal processing, including normalization, filtering, and data augmentation.

4.2.1 Normalization

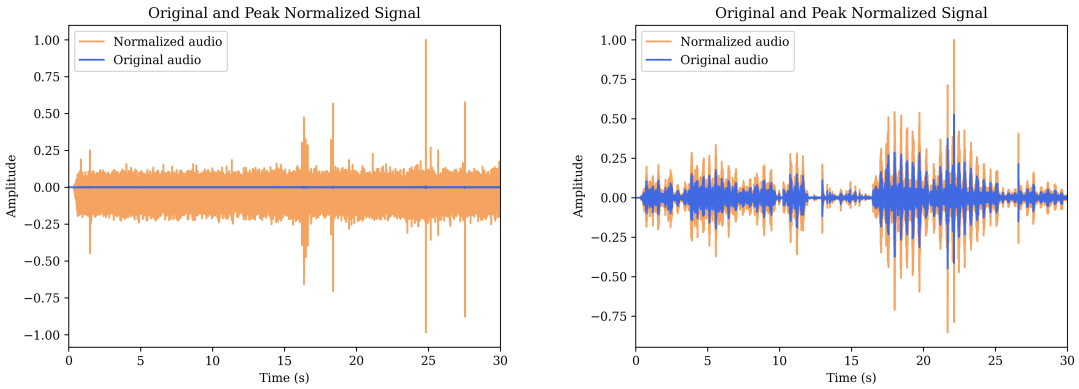
Table 4.2 includes a subset of the results in Table A. These results show that peak normalization performs poorly for the pre-exercise model. To understand why, we examine the plots showing the original and normalized signals of a BOAS-negative dog with quiet breathing and a BOAS-positive dog with loud breathing.

Table 4.2: Results of training a Random Forest classifier using cross-validation, applied to pre- and post-exercise data under three preprocessing settings.

Preprocessing settings	Pre-exercise		Post-exercise	
	Mean accuracy:	Mean AUC:	Mean accuracy:	Mean AUC:
Original	0.867 ± 0.083	0.961 ± 0.054	0.925 ± 0.061	1.000 ± 0.000
RMS normalization	0.867 ± 0.083	0.967 ± 0.044	0.925 ± 0.061	0.987 ± 0.027
Peak normalization	0.844 ± 0.133	0.886 ± 0.115	0.925 ± 0.061	0.987 ± 0.026

Figure 4.1 shows the results of Peak normalization applied to these two recordings. Peak normalization did not perform well on recordings with relatively quiet breathing, as it uniformly amplified the amplitude of all samples, making the quiet breaths appear as loud as the originally louder ones. For instance, in Figure 4.1a, the normalized quiet breath fluctuates around ±0.25, and similarly, in Figure 4.1b, the normalized loud breath fluctuates around a similar amplitude (±0.5). This diminishes the distinction between quiet and loud breathing. Therefore, it is reasonable that Peak normalization seems to work for the post-exercise data but not for the

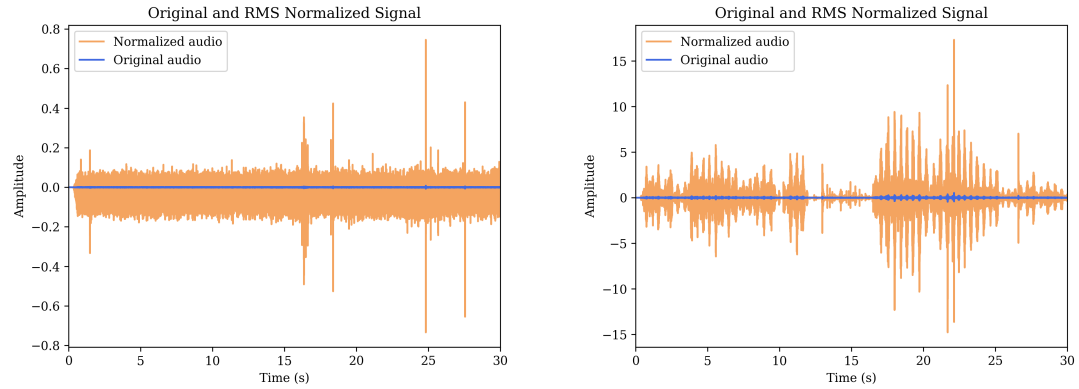
pre-exercise recordings since all signals in the post-exercise are not as quiet as they can be in the pre-exercise recordings.



(a) **(b)**

Figure 4.1: Time-domain plots of audio recordings of a dog's breathing, showing the original (blue) and Peak normalized (orange) signals. Plot (a) shows quiet breathing, while plot (b) shows loud breathing.

RMS normalization was applied instead to achieve more consistent results. Figure 4.2 shows the corresponding RMS-normalized recordings. This method scales the amplitude based on the overall energy of each recording, providing a more consistent loudness across samples. In Figure 4.2a, the quiet breathing remains around ± 0.25, but in Figure 4.2b, the loud breathing shows much larger fluctuations (around ± 10), preserving the difference in loudness between quiet and loud breathing more effectively.



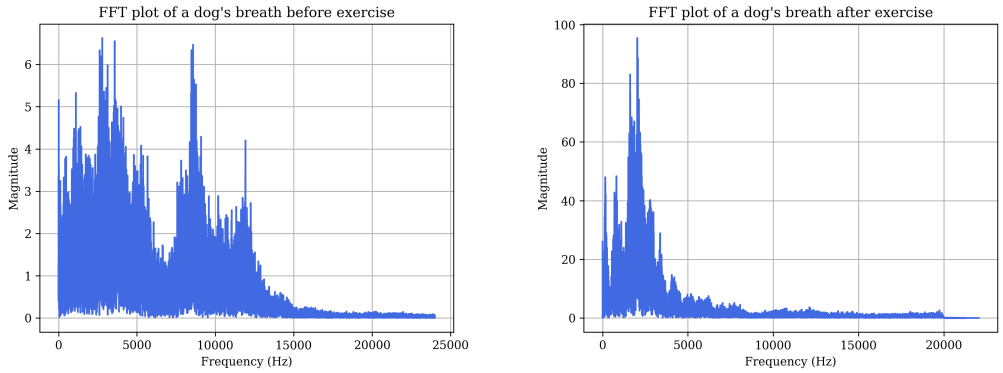
(a) **(b)**

Figure 4.2: Time-domain plots of audio recordings of a dog's breathing, showing the original (blue) and RMS normalized (orange) signals. Plot (a) shows quiet breathing, while plot (b) shows loud breathing.

Comparing Table B.2, B.3 verifies that RMS normalized data performs better than using original data without preprocessing.

4.2.2 Filtering

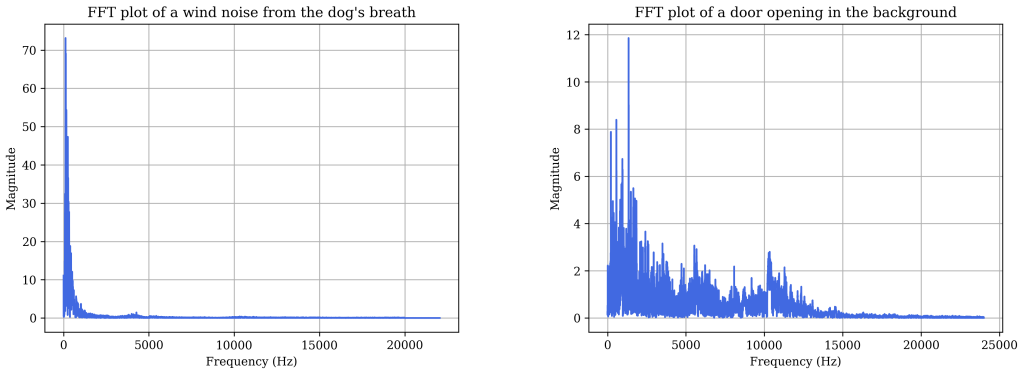
The Fourier transform of a single dog's breath before exercise is shown in Figure 4.3a, while the transform of a breath after exercise is shown in Figure 4.3b. The figures show that the breath when the dog is resting ranges over a broader range of frequencies (0 Hz to 14 000 Hz) compared to a breath after exercise, which ranges from 0 Hz to 5 000 Hz. This indicates which cutoff frequency one should use when low-pass filtering the signals. In this case, the pre-exercise data should have a cutoff frequency above 10 000 Hz, while the post-exercise data should have a cutoff frequency around 5 000 Hz. On the other hand, when examining the low-pass filtering results in Appendix A, it is not apparent which cutoff frequency is most appropriate.



(a) **(b)**

Figure 4.3: Frequency-domain plot of a single breath recorded (a) before exercise and (b) after exercise.

An example of noise that appears in several recordings is the scraping noise of the dog's breath into the microphone. Figure 4.4a shows the Fourier transform of the wind noise. The noise has a very low frequency; the peak is at 95 Hz, and the high-pass filter removes it.

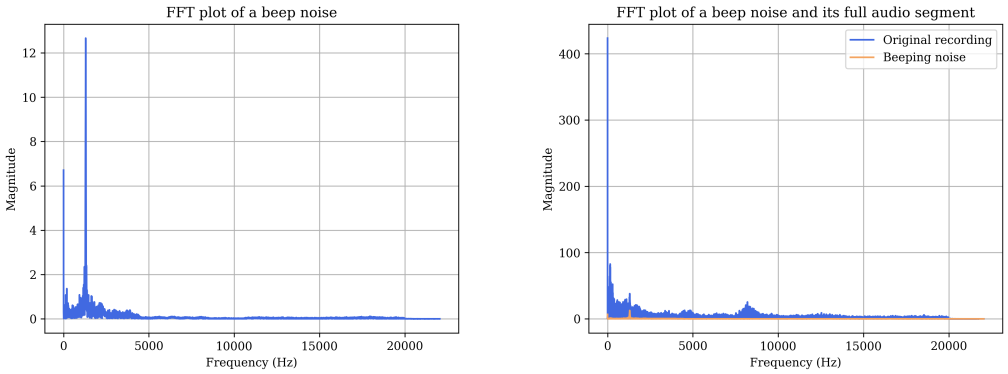


(a) **(b)**

Figure 4.4: Frequency-domain plot of a recorded (a) wind noise and (b) a door opening in the background.

A more particular background noise is the sound of a door opening, which is shown in the frequency domain in Figure 4.4b. Unlike wind noise, this sound spans a wide range of frequencies, making it more challenging to remove completely through filtering. However, despite this noise in the recording of dog number 5 from the new dataset, the model can still correctly classify the BOAS grade. This suggests that the model is robust to certain types of background noise that cannot be fully filtered out.

Another sound appearing in several recordings is a beeping noise produced when the person conducting the recording starts or stops a digital stethoscope. This beep has a distinct peak at 1 300 Hz, as shown in Figure 4.5a. Therefore, applying a high-pass filter to remove it would risk eliminating important lower-frequency information. However, Figure 4.5b shows that the beep's frequency content relative to the entire recording shows that its contribution is minimal compared to the overall frequency distribution. Therefore, I chose not to filter out the beeps, as that might risk removing other informative lower frequencies. Additionally, I reviewed all recordings and confirmed that the beeping occurs in both BOAS-negative and BOAS-positive cases, ensuring it does not bias the classification results.



(a) **(b)**

Figure 4.5: Frequency-domain plot of a (a) single beep tone and (b) the original recording where the beep occurs.

By observing the results of the high-pass and low-pass filtered signals in Table A.1, it is evident that a single, fixed cutoff frequency is unsuitable for all signals. For the pre-exercise data, the model performed better when a high-pass filter with a cutoff frequency of 80 Hz was applied. In contrast, the model showed similar performance across the different filter configurations for the post-exercise data, with none outperforming the original data without preprocessing. The pre-exercise data yielded better performance in the low-pass filtering tests with a low-pass filter set at 13 000 Hz. For the post-exercise data, the classification results were comparable to those of the original data when using cutoff frequencies of 8 000 Hz and 10 000 Hz. These findings suggest the need for a more dynamic approach, where the cutoff frequencies for both filters are determined for each individual signal.

From the results in Table B.4 to B.7, it can be concluded that thresholds of 10% or 15% for estimating the high-pass filter cutoff frequency yield the best results. Further analysis of Table B.11 and B.12 indicates that a threshold of 15% performs best overall for estimating the cutoff frequency.

Although the prediction probabilities vary slightly across different preprocessing settings using the OnePlus recordings, the overall model performance is good, as it correctly predicts the BOAS grade for all dogs in nearly every setting. However, the results for the Samsung phone are less promising (see Tables B.13 to B.16). A frequency-domain analysis of simultaneous phone recordings, shown in Figure 4.6, reveals significant differences in audio preprocessing. The Samsung recording appears to have been low-pass filtered with a cutoff frequency of 20 000 Hz, likely to prevent aliasing by the Nyquist theorem, as both phones sample at 48 000 Hz. In contrast, the OnePlus recording appears to have undergone smoothing around 13 000 Hz, possibly due to an older microphone with reduced high-frequency sensitivity. This observation clarified that a low-pass filter would be necessary, even though the initial results in Table A.1 did not suggest its need. Additionally, normalization became more critical, as the overall loudness in the Samsung recording is higher than in the OnePlus recording, as shown in Figure 4.6.

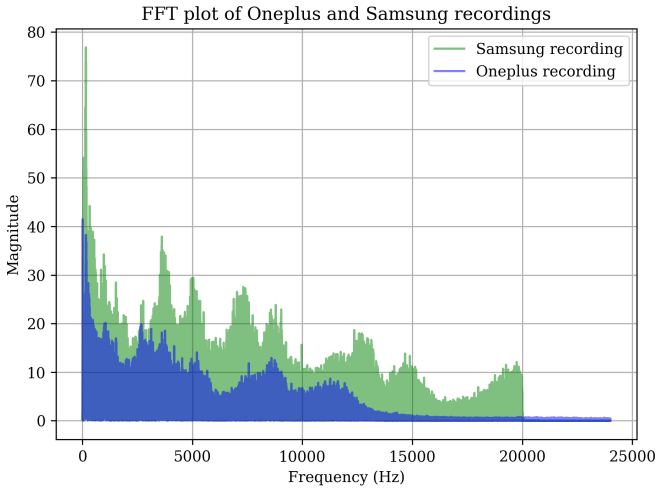


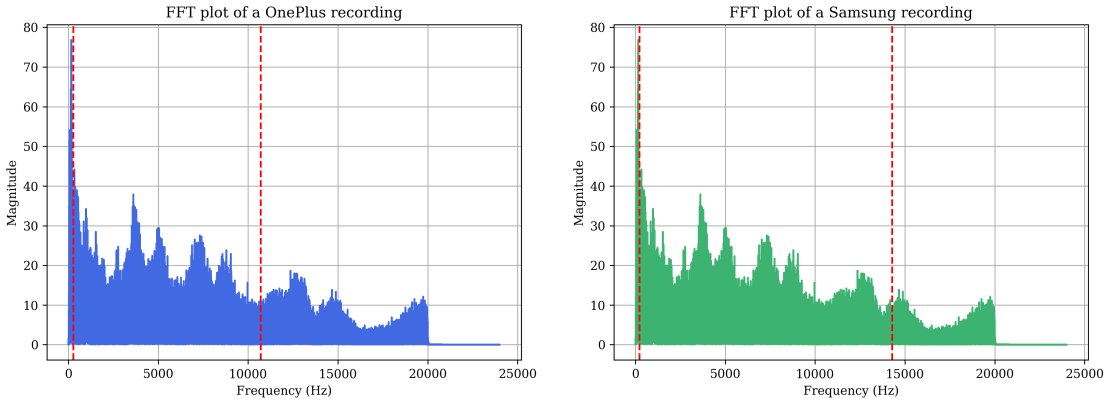
Figure 4.6: Frequency-domain plot of the same sound recorded during the same time but with two different phones.

As mentioned, Table A.1 shows that a single, fixed cutoff frequency is unsuitable for all signals. It is unclear whether a low-pass filter was necessary, as the results were almost identical to those from the non-normalized data. However, based on the observations regarding the differences in preprocessing between the two phones, low-pass filtering was applied to achieve the best results for preprocessing, including high-pass filtering, normalization, and data augmentation, as investigated on the OnePlus recordings, to ensure that the model also predicted correctly for the Samsung phone.

All proceeding results are high-pass filtered with a threshold of 15%. Results from low-pass filtering with an estimated cutoff threshold of 95% performed better for

both phones (Table B.17 and B.18) than low-pass filtering with an estimated cutoff threshold of 90% (Table B.19 and B.20). Performing RMS normalization on the low-pass filtered data with an estimated cutoff threshold of 95% (Table B.21 and B.22) produced worse results, even when combined with data augmentation.

Figure 4.7 shows the same sound recorded with the OnePlus and the Samsung in the frequency domain with corresponding estimated cutoff frequencies. The red dotted line at the lower frequency end represents the cutoff frequency for the high-pass filter, and the other red dotted line represents the estimated cutoff frequency for the low-pass filter.



(a) **(b)**

Figure 4.7: Frequency-domain plot of a simultaneous recording using (a) OnePlus and (b) Samsung, with the estimated cutoff frequencies for both the high-pass and low-pass filters marked by red dotted lines.

4.2.3 Data Augmentation

An important observation regarding data augmentation through amplitude amplification or reduction is that it tends to be canceled out when being combined with Peak normalization. However, this was not the case when using RMS normalization.

Results related to data augmentation are shown in Table B.9 to B.12 for the OnePlus recordings, Table B.15 and B.16 for the Samsung recordings, and Table B.23 to B.26 for the final comparison between the OnePlus and Samsung recordings.

Comparing Table B.10 and B.12, where the signals underwent the same preprocessing but differed in amplitude adjustment, the average prediction probability for the correct class is higher when using 15% amplification and 5% reduction, compared to using 10% for both amplification and reduction.

To compare RMS normalization with data augmentation, refer to Tables B.21 and B.22 versus Tables B.25 and B.26. The results show that the model performs better with data augmentation alone than with RMS normalization for the OnePlus phone. A combination of data augmentation and RMS normalization yields the best results for the Samsung phone, see Table B.24.

4.3 Feature Extraction

Among the hundred features extracted by the Mann-Whitney U-test, the feature types listed in Table 4.3 were the openSMILE features that appeared in most subsets. Therefore, these are the features that best describe the differences in characteristics between BOAS-negative and BOAS-positive dog recordings:

Table 4.3: Distribution of openSMILE features among 100 Mann-Whitney features for non-preprocessed data.

Feature type	Number of features
Auditory Spectra	60
MFCC	19
RMS Energy	13
FFT Magnitude	3
Jitter	2
Shimmer	1
HNR	2
Total	100

The Auditory spectra, according to the openSMILE documentation, is used to describe psychoacoustic sharpness, which, according to a study, is a way to explore the psychology behind how people perceive sound, often about loudness [20], [41]. This can be studied by looking at the frequency content of a specific sound [41]. One way to do this is by observing the frequency spectrum, a graph showing the amplitude in decibels of the different frequencies in the sound.

Mel-frequency Cepstral Coefficients (MFCCs) capture the shape of a signal's power spectrum in a way that aligns with human auditory perception [42]. Central to this process is the Mel scale, a perceptual scale of pitch where equal steps correspond to equal perceived differences in pitch [43]. Because human hearing is more sensitive to changes in lower frequencies than higher ones, the Mel scale emphasizes finer resolution at the lower end of the frequency spectrum. To derive MFCCs, the signal is first transformed into the frequency domain using the Discrete Fourier Transform (DFT), after which the Mel scale is applied to approximate how humans perceive sound [42].

RMS Energy or Root Mean Square Energy represents the average loudness of a signal. The openSMILE documentation does not provide a detailed explanation of this feature, but several related features appear to be derived from RMS energy. Based on their feature names, these features likely correspond to different percentiles of the signal.

Magnitude of the Fast Fourier Transform (FFT) represents the energy distribution across frequencies and is derived by calculating the magnitude of the signal's frequency domain, which is determined using the FFT in this case.

Jitter and Shimmer are voice-quality parameters. Jitter refers to the small, rapid variations in a sound wave's frequency pitch from one cycle to the next [44]. Shimmer measures the rapid variations in the amplitude (loudness) of the sound wave across successive cycles. Figure 4.8 represents Jitter and Shimmer.

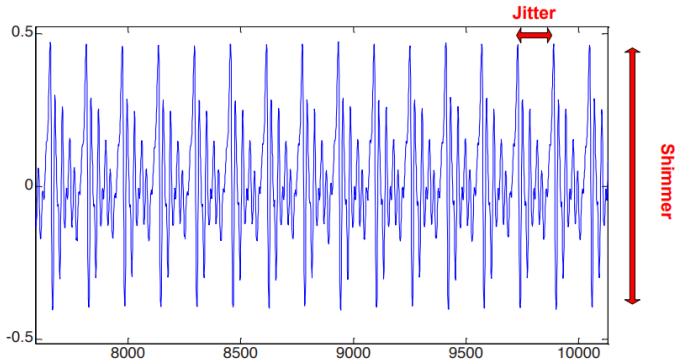


Figure 4.8: Representation of Jitter and Shimmer perturbation measures in a speech signal [44] CC BY-NC-ND 3.0.

Harmonics-to-Noise Ratio (HNR) is a measure of the ratio between the periodic (harmonic) and aperiodic (noise) components in a speech signal, expressed in decibels [45]. Higher HNR values indicate a cleaner, more periodic sound, while lower values suggest increased noise. The HNR is a logarithmic measure of the signal's energy ratio, which the following formula can derive,

$$HNR = 10 \times \log_{10} \frac{\int_w |H(w)|^2}{\int_w |N(w)|^2}$$

where $X(\omega)$ corresponds to the speech signal in the frequency domain, $H(\omega)$ to the harmonic component and $N(\omega)$ to the noise component.

4.4 Statistical Tests

Applying the Spearman correlation test to the original combined openSMILE feature set in Setup 2, 29 out of 12 745 features had a Spearman correlation coefficient greater than 0.6. Using the Pearson correlation test, 13 features had a Pearson correlation coefficient greater than 0.6.

Applying the Mann–Whitney U-test to the same DataFrame, the test identified 1 179 features with a p-value smaller than 0.05. After selecting 100 of these features with the lowest p-value, counting the number of features from the pre-exercise data (those named `before_et`), there were 13 features and 87 post-exercise features (those named `after_et`). This clearly shows that the post-exercise recordings are more informative and correlate better with the characteristics of BOAS-negative and BOAS-positive dogs' recordings.

4.5 Classification

The best-performing classification results, using a preprocessing methods that was most effective for both OnePlus and Samsung recordings, are shown in Tables B.25 and B.26. This pipeline includes data augmentation by randomly increasing the amplitude by 15% and decreasing it by 5%, along with high-pass filtering (cutoff estimated at 15%) and low-pass filtering (cutoff estimated at 95%). For the Samsung recordings, the addition of RMS normalization further improved performance, as demonstrated in Table B.24.

For the **OnePlus** recordings, the optimal preprocessing setup includes:

- No normalization
- High-pass filtering with an estimated cutoff of 15%
- Low-pass filtering with an estimated cutoff of 95%
- Data augmentation (amplitude increased by 15% and decreased by 5%)

According to Table B.1, this setup yields a mean accuracy of 81.3% and a mean AUC of 1.0 when training the Random Forest classifier with the filtering methods alone. Adding data augmentation increases the mean accuracy to 100%, while the mean AUC remains at 1.0.

For the **Samsung** recordings, the optimal preprocessing setup includes:

- RMS normalization
- High-pass filtering with an estimated cutoff of 15%
- Low-pass filtering with an estimated cutoff of 95%
- Data augmentation (amplitude increased by 15% and decreased by 5%)

Using only the filtering methods and RMS normalization results in a mean accuracy of 85.3% and a mean AUC of 1.0. When data augmentation is added, mean accuracy rises to 96.4%, with the mean AUC still at 1.0.

Classification results using the XGBoost model on the same feature sets are presented in Tables 4.4 and 4.5.

Table 4.4: Classification results of the **OnePlus** recordings preprocessed using data augmentation, high-pass filtering, and low-pass filtering, and classified using XGBoost.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.978	0.022
2	0	0	0.864	0.136
3	1	1	0.037	0.963
4	0	0	0.813	0.187
5	0	0	0.751	0.249

The average probability of the correctly predicted classes across all dogs is 87.4%.

Table 4.5: Classification results of the **Samsung** recordings preprocessed using RMS normalization, data augmentation, high-pass filtering, and low-pass filtering, and classified using XGBoost.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.970	0.030
2	0	0	0.963	0.037
3	1	1	0.424	0.576
4	0	0	0.950	0.049
5	0	0	0.965	0.035

The average probability of the correctly predicted classes in this case is 88.5%.

5

Discussion

5.1 Answering the Research Questions

What signal processing methods enhance the breathing characteristics of a phone recording of a BOAS-negative and BOAS-positive dog?

The results clearly show that Peak normalization is not suitable for this type of data. Since the amplitude of the breathing signals correlates with whether the dog is BOAS-negative or BOAS-positive, normalizing using the peak distorts this important information. BOAS-negative dogs tend to breathe more quietly, and peak normalization removes this difference by making all signals equally loud.

RMS normalization, however, proved to be a significantly better method. It adjusts the signal while maintaining the relative loudness, which means that the natural variations in breathing volume between dogs are preserved.

The results also show that using fixed cutoff frequencies for all recordings when filtering does not yield optimal results. Each recording can have different frequency characteristics, and one filter setting does not suit all signals. That is why a dynamic filtering method, where the cutoff frequencies are determined individually for each signal, gave better results. This ensures that the filtering adapts to the signal rather than vice versa. Filtering with a high-pass filter, estimated to have a cutoff at a threshold of 15%, and low-pass filtering, estimated to have a cutoff at a threshold of 95%, performed best for both phones.

The resulting filtering methods and data augmentation techniques performed well across recordings from both phones. However, RMS normalization was less effective for the OnePlus recordings. Further investigation is needed to develop a preprocessing pipeline that consistently performs well across a broader range of phones.

Which features show the strongest correlation with BOAS-negative and BOAS-positive dog samples?

The openSMILE features most strongly correlated with BOAS-negative and BOAS-positive labels were frequency spectra, MFCCs, RMS energy, FFT magnitude, jitter, shimmer, and HNR. RMS energy, jitter, and shimmer are time-domain features, while the others are frequency-domain features. This shows that features from the frequency domain are essential, which also supports the idea of using filtering and

processing techniques that enhance frequency-related properties in the signal. Since most relevant features are derived from the frequency domain, it is crucial to focus on preserving and enhancing the signal's frequency characteristics through effective preprocessing.

Is it more suitable to use separate models for pre- and post-exercise recordings for the intended task, or a hybrid model that incorporates both?

When comparing the two model setups, the classification results indicate that Setup 2, the hybrid model trained on both pre- and post-exercise data, outperforms Setup 1, which employs two separate models. The hybrid model had higher accuracy, AUC, and prediction confidence overall. This was expected, considering that the feature selection process (using the Mann-Whitney test) yielded 87 features from post-exercise data and only 13 from pre-exercise recordings. That means post-exercise recordings contain more useful information related to the characteristics of a BOAS-negative and BOAS-positive dog's recording.

While using separate models (Setup 1) is still a valid alternative, it would require combining the predictions using some form of weighted average, most likely giving more weight to post-exercise data. However, doing so adds complexity and is not guaranteed to yield better results. Therefore, the hybrid model is the best option for this task.

Do the audio recordings differ between phones?

The audio recordings between different phone brands, in this case a OnePlus and a Samsung, differ significantly. As shown in the results, the frequency distribution differed between the two phones. This affects the preprocessing and the classification of these recordings. Although I found a preprocessing method that worked for both, it still varied significantly between them. In the future, it would be desirable to have a preprocessing setup that works for several other phones as well.

Another variable to consider is how the recording conditions impact the data. When collecting the first dataset, recordings were taken by Sykkö and Dimopoulou, who were well-informed and careful during the recording. However, the environment at the veterinary clinics where I performed the recordings was not ideal, and not everyone was aware that recordings were being made. This resulted in some recordings having more background noise and other sounds, which could affect quality.

On the other hand, it can be beneficial for the model to be trained on recordings from different conditions. In the future, this system may be used by regular people at home, and the recordings will also vary significantly. Training on noisy or varied data can make the model more robust, if there is sufficient data to train it on. However, this variability still needs to be accounted for during signal analysis to avoid classification issues.

5.2 Model Performance and Limitations

One clear observation that can be drawn from the classification results is that the model generally has the most difficulty classifying the BOAS-positive dog in the new dataset. This is reasonable since both datasets are imbalanced, with a greater number of BOAS-negative recordings than BOAS-positive recordings. Since the model is trained on this imbalanced dataset, it will naturally predict BOAS-negative dogs more easily than BOAS-positive dogs.

The statistical tests revealed that more features had a strong monotonic relationship with the BOAS label rather than a linear one. That is one reason I chose to use a Random Forest classifier rather than a linear model. Still, the difference was not huge, so a linear model could have been used too, but it likely would not have performed as well.

It is also important to note that the feature set was selected and optimized specifically for the Random Forest model. That means the same set might not work well for other models or neural networks. Although the final results appear promising, they do not guarantee that the model will perform well on all new data. The dataset is still small, and the evaluation was based on how well the model predicted the dogs in the new dataset. It is possible that some of the new recordings were very similar to older ones. However, even so, it demonstrates that a feature-based classification of BOAS is possible if the signal processing is designed to highlight the relevant characteristics in the recordings.

Better models may exist for this task, and it could be worth testing other machine learning methods or data augmentation techniques. Pagrell's thesis includes a more detailed investigation regarding models and data augmentation [14].

An important aspect regarding machine learning models when working with medical data is that the model should be interpretable [46]. One should avoid using "black box" models as there is no simple way to interpret the model's decisions. The Random Forest classifier is not the best alternative, as tracking its decisions throughout all its trees can be challenging. Therefore, it has been considered a black box. On the other hand, with the right tools, it is possible to make the model interpretable, thereby gaining insight into its internal decision-making process. The cited article presents tools for this, but this project has not investigated it thoroughly [46]. XGBoost is a better choice in this regard, as it is a more advanced model that can output feature importance, allowing users to see which features have been used most in each tree's decision [47]. However, using a less interpretable model for the feature set created in the project is not particularly risky, as I have complete insight into which features I input into the model, and I know that all of them represent properties of the signal's characteristics. No particular artifacts or background noises directly influence an extracted feature; the signal is normalized and filtered, and the features are determined based on the whole signal, not individual segments.

Generally, the classification results are promising. For the final classification results, the AUC remains 1.0, which is ideal when working with medical data. This means

that the model has perfectly differentiated between the two classes, with no false positives or false negatives, which is ideal.

5.3 Future Work

The most important thing going forward is to collect more data. A larger, more varied dataset would improve many of the challenges with classification and generalization. Not only would it diminish the imbalance in the dataset, but it would also improve differences between phones; for example, recordings from the OnePlus and Samsung phones were quite different, likely due to differences in microphone hardware and how the phones process sound.

Although I developed a signal processing pipeline that handled both devices, the optimal preprocessing methods varied slightly between them. This suggests that other phone brands may introduce similar issues. Due to these potential additional variabilities, the first step in future data collection should be to gather recordings from a broader range of phones. This would likely necessitate further investigation of the signal processing methodologies until there is sufficient data for the model to learn and generalize effectively across these differences.

In addition to expanding the dataset, future data collection could benefit from including other relevant parameters that may serve as useful features for classification. For example, since BOAS in dogs is related to their physique, including information such as the dog's weight and neck circumference could be valuable. Similarly, physiological measures like heart rate and oxygen saturation provide insight into the dog's respiratory condition and could enhance model performance. However, these types of data would only be feasible to collect in veterinary settings, as they require special instruments and would not be practical for use by dog owners at home.

Another future project is to continue developing the mobile app that Sykkö initiated. One way would be to create an external API that receives recordings from the app, performs the preprocessing, openSMILE feature extraction and classification, and then returns the prediction. In that way, the app could be used more easily in a real-life setting.

6

Conclusion

This project demonstrates the potential of feature-based classification for diagnosing BOAS. By applying appropriate signal processing methods and extracting features using openSMILE, a feature set can be constructed that captures the distinguishing characteristics of recordings from BOAS-negative and BOAS-positive dogs.

Preprocessing the audio using RMS normalization, combined with a dynamic filtering approach that adapts the cutoff frequencies to each recording, enhances the most relevant aspects of the signal. Additionally, a hybrid model that incorporates both pre- and post-exercise recordings yields a more informative feature set. Using the Mann-Whitney U-test for statistical analysis further supports the effectiveness of feature reduction, resulting in a smaller yet more meaningful set of features.

Despite variations in audio recordings between phones, the preprocessing pipeline successfully extracted key information from both OnePlus and Samsung recordings. The Random Forest classifier accurately identified all five test cases, and using the XGBoost model further improved prediction probability.

Although additional data is needed to ensure the model's reliability and generalization to unseen data, this approach shows promising outcomes for using feature-based classification of BOAS.

References

[1] "A close up of a dog's face with a blurry background. french bulldog smart look dog. - PICRYL - public domain media search engine public domain image." (), [Online]. Available: https://timelessmoon.getarchive.net/amp/media/french-bulldog-smart-look-dog-animals-885295 (visited on 04/28/2025).

[2] S. Mitze, V. R. Barrs, J. A. Beatty, S. Hobi, and P. M. Bęczkowski, "Brachycephalic obstructive airway syndrome: Much more than a surgical problem," *The Veterinary Quarterly*, vol. 42, no. 1, pp. 213–223, ISSN: 0165-2176. DOI: 10.1080/01652176.2022.2145621. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9673814/ (visited on 03/04/2025).

[3] Administrator. "About BOAS." (Feb. 16, 2016), [Online]. Available: https://www.vet.cam.ac.uk/boas/about-boas (visited on 03/04/2025).

[4] "Brachycephalic dogs: What we know about frenchies, pugs and bulldogs," Felcana. (), [Online]. Available: https://felcana.com/blogs/blog/brachycephalic-dogs (visited on 03/04/2025).

[5] "The rise and fall of popular dog breeds | everypaw." (), [Online]. Available: https://www.everypaw.com/all-things-pet/the-rise-and-fall-of-popular-dog-breeds (visited on 04/10/2025).

[6] Administrator. "Recognition & diagnosis." (Feb. 16, 2016), [Online]. Available: https://www.vet.cam.ac.uk/boas/about-boas/recognition-diagnosis (visited on 03/04/2025).

[7] S. Kennelklubben. "RFG-Scheme." (), [Online]. Available: https://www.skk.se/uppfodning/halsa/andning/rfg-scheme/ (visited on 05/09/2025).

[8] Administrator. "Management & treatment." (Feb. 16, 2016), [Online]. Available: https://www.vet.cam.ac.uk/boas/about-boas/management-treatment (visited on 05/09/2025).

[9] M. Mårtensson, "Brachycephalic obstructive airway syndrome (BOAS) classification in dogs based on respiratory noise analysis using machine learning," 2021. [Online]. Available: https://hdl.handle.net/20.500.12380/302233 (visited on 04/23/2025).

[10] H. Pettersson and O. Stensöta, "Data augmentation for audio based machine learning classifying brachycephalic obstructive airway syndrome (BOAS) in dogs," 2021. [Online]. Available: https://hdl.handle.net/20.500.12380/303984 (visited on 04/23/2025).

[11] "RandomForestClassifier," scikit-learn. (), [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (visited on 04/23/2025).

[12] M. Dimopoulou, H. Peterson, O. Stensöta, *et al.*, "Use of respiratory signal analysis to assess severity of brachycephalic obstructive airway syndrome (BOAS) in dogs," *The Veterinary Journal*, vol. 308, p. 106 261, Dec. 1, 2024, ISSN: 1090-0233. DOI: 10.1016/j.tvjl.2024.106261. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1090023324002004 (visited on 04/23/2025).

[13] A. Oren, J. D. Türkcü, S. Meller, *et al.*, "BrachySound: Machine learning based assessment of respiratory sounds in dogs," *Scientific Reports*, vol. 13, no. 1, p. 20 300, Nov. 20, 2023, Publisher: Nature Publishing Group, ISSN: 2045-2322. DOI: 10.1038/s41598-023-47308-0. [Online]. Available: https://www.nature.com/articles/s41598-023-47308-0 (visited on 04/23/2025).

[14] T. Pagrell, "Diagnosing Brachycephalic Obstructive Airway Syndrome in Dogs Using Computer Vision and Machine Learning," [Online]. Available: https://odr.chalmers.se/communities/82b3e123-24a1-47ec-8544-f8ee5b27ac29 (visited on 05/28/2025).

[15] P. priyanka. "Audio normalization," Medium. (Sep. 5, 2023), [Online]. Available: https://medium.com/@poudelnipriyanka/audio-normalization-9dbcedfefcc0 (visited on 03/20/2025).

[16] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, International Conference on Intelligent Engineering Approach(ICIEA-2022), vol. 3, no. 1, pp. 91–99, Jun. 1, 2022, ISSN: 2666-285X. DOI: 10.1016/j.gltp.2022.04.020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666285X22000565 (visited on 05/05/2025).

[17] S. R. Devasahayam, *Signals and Systems in Biomedical Engineering: Physiological Systems Modeling and Signal Processing*. Singapore: Springer Singapore, 2019, ISBN: 978-981-13-3530-3 978-981-13-3531-0. DOI: 10.1007/978-981-13-3531-0. [Online]. Available: http://link.springer.com/10.1007/978-981-13-3531-0 (visited on 05/05/2025).

[18] "Butterworth filter - an overview | ScienceDirect topics." (), [Online]. Available: https://www.sciencedirect.com/topics/engineering/butterworth-filter (visited on 05/05/2025).

[19] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: The munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM international conference on Multimedia*, ser. MM '10, New York, NY, USA: Association for Computing Machinery, 2010, pp. 1459–1462, ISBN: 978-1-60558-933-6. DOI: 10.1145/1873951.1874246. [Online]. Available: https://dl.acm.org/doi/10.1145/1873951.1874246 (visited on 05/05/2025).

[20] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in openSMILE, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM international conference on Multimedia*, ser. MM '13, New York, NY, USA: Association for Computing Machinery, 2013, pp. 835–838, ISBN: 978-1-4503-2404-5. DOI: 10.1145/2502081.2502224. [Online].

Available: https://dl.acm.org/doi/10.1145/2502081.2502224 (visited on 05/05/2025).

[21] "Feature reduction - an overview | ScienceDirect topics." (), [Online]. Available: https://www.sciencedirect.com/topics/computer-science/feature-reduction (visited on 04/30/2025).

[22] A. K. Kurtz and S. T. Mayo, "Pearson product moment coefficient of correlation," in *Statistical Methods in Education and Psychology*, A. K. Kurtz and S. T. Mayo, Eds., New York, NY: Springer, 1979, pp. 192–277, ISBN: 978-1-4612-6129-2. DOI: 10.1007/978-1-4612-6129-2_8. [Online]. Available: https://doi.org/10.1007/978-1-4612-6129-2_8 (visited on 05/02/2025).

[23] "Introduction to nonparametric methods | EBSCO research starters." (), [Online]. Available: https://www.ebsco.com/research-starters/business-and-management/introduction-nonparametric-methods (visited on 05/04/2025).

[24] "Mann-whitney u test: Assumptions and example," Informatics from Technology Networks. (), [Online]. Available: http://www.technologynetworks.com/informatics/articles/mann-whitney-u-test-assumptions-and-example-363425 (visited on 03/04/2025).

[25] J. L. Devore, *Probability and statistics for engineering and the sciences*, 8th ed. Boston, MA: Brooks/Cole, Cengage Learning, 2012, OCLC: 696106248, ISBN: 978-0-538-73352-6.

[26] "XGBoost documentation — xgboost 3.0.1 documentation." (), [Online]. Available: https://xgboost.readthedocs.io/en/release_3.0.0/ (visited on 05/08/2025).

[27] "Machine learning - a first course for engineers and scientists," sml-book-page. (), [Online]. Available: http://smlbook.org/ (visited on 05/08/2025).

[28] R. Kannan and V. Vasanthi, "Machine learning algorithms with ROC curve for predicting and diagnosing the heart disease," in *Soft Computing and Medical Bioinformatics*, N. B. Muppalaneni, M. Ma, and S. Gurumoorthy, Eds., Singapore: Springer, 2019, pp. 63–72, ISBN: 978-981-13-0059-2. DOI: 10.1007/978-981-13-0059-2_8. [Online]. Available: https://doi.org/10.1007/978-981-13-0059-2_8 (visited on 05/02/2025).

[29] "Introduction to the ROC (receiver operating characteristics) plot," Classifier evaluation with imbalanced datasets. (Jun. 9, 2015), [Online]. Available: https://classeval.wordpress.com/introduction/introduction-to-the-roc-receiver-operating-characteristics-plot/ (visited on 05/02/2025).

[30] C. Chan. "What is a ROC curve and how to interpret it," Displayr. (Jul. 5, 2018), [Online]. Available: https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/ (visited on 03/04/2025).

[31] S. Narkhede. "Understanding AUC - ROC curve," TDS Archive. (Jun. 15, 2021), [Online]. Available: https://medium.com/towards-data-science/understanding-auc-roc-curve-68b2303cc9c5 (visited on 03/11/2025).

[32] M. P. Muller, G. Tomlinson, T. J. Marrie, *et al.*, "Can routine laboratory tests discriminate between severe acute respiratory syndrome and other causes of community-acquired pneumonia?" *Clinical Infectious Diseases: An Offi-*

cial Publication of the Infectious Diseases Society of America, vol. 40, no. 8, pp. 1079–1086, Apr. 15, 2005, ISSN: 1537-6591. DOI: 10.1086/428577.

[33] T.-T. Wong and P.-Y. Yeh, "Reliable accuracy estimates from k-fold cross validation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1586–1594, Aug. 2020, ISSN: 1558-2191. DOI: 10.1109/TKDE.2019.2912815. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8698831 (visited on 05/02/2025).

[34] S. Prusty, S. Patnaik, and S. K. Dash, "SKCV: Stratified k-fold cross-validation on ML classifiers for predicting cervical cancer," *Frontiers in Nanotechnology*, vol. 4, Aug. 19, 2022, Publisher: Frontiers, ISSN: 2673-3013. DOI: 10.3389/fnano.2022.972421. [Online]. Available: https://www.frontiersin.orghttps://www.frontiersin.org/journals/nanotechnology/articles/10.3389/fnano.2022.972421/full (visited on 05/02/2025).

[35] "Pandas - python data analysis library." (), [Online]. Available: https://pandas.pydata.org/ (visited on 05/22/2025).

[36] "Python-soundfile — python-soundfile 0.13.1 documentation." (), [Online]. Available: https://python-soundfile.readthedocs.io/en/0.13.1/ (visited on 04/09/2025).

[37] "Butter — SciPy v1.15.3 manual." (), [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html (visited on 05/14/2025).

[38] "openSMILE: Audio fingerprinting and feature extraction," audEERING. (), [Online]. Available: https://www.audeering.com/research/opensmile/ (visited on 05/22/2025).

[39] "Scikit-learn: Machine learning in python — scikit-learn 1.6.1 documentation." (), [Online]. Available: https://scikit-learn.org/stable/ (visited on 05/29/2025).

[40] "Python package introduction — xgboost 3.0.2 documentation." (), [Online]. Available: https://xgboost.readthedocs.io/en/stable/python/python_intro.html (visited on 05/29/2025).

[41] "Psychoacoustics - an overview | ScienceDirect topics." (), [Online]. Available: https://www.sciencedirect.com/topics/earth-and-planetary-sciences/psychoacoustics (visited on 05/08/2025).

[42] uday. "MFCC technique for speech recognition," Analytics Vidhya. (Jun. 13, 2021), [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/mfcc-technique-for-speech-recognition/ (visited on 04/07/2025).

[43] L. Roberts. "Understanding the mel spectrogram," Medium. (Jan. 17, 2024), [Online]. Available: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53 (visited on 04/07/2025).

[44] J. P. Teixeira, C. Oliveira, and C. Lopes, "Vocal acoustic analysis – jitter, shimmer and HNR parameters," *Procedia Technology*, CENTERIS 2013 - Conference on ENTERprise Information Systems / ProjMAN 2013 - International Conference on Project MANagement/ HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies, vol. 9, pp. 1112–1122, Jan. 1, 2013, ISSN: 2212-0173. DOI: 10.1016/j.protcy.2013

.12.124. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212017313002788 (visited on 04/07/2025).

[45] J. Fernandes, F. Teixeira, V. Guedes, A. Junior, and J. P. Teixeira, "Harmonic to noise ratio measurement - selection of window and length," *Procedia Computer Science*, CENTERIS 2018 - International Conference on ENTERprise Information Systems / ProjMAN 2018 - International Conference on Project MANagement / HCist 2018 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2018, vol. 138, pp. 280–285, Jan. 1, 2018, ISSN: 1877-0509. DOI: 10.1016/j.procs.2018.10.040. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918316739 (visited on 04/07/2025).

[46] M. Aria, C. Cuccurullo, and A. Gnasso, "A comparison among interpretative proposals for random forests," *Machine Learning with Applications*, vol. 6, p. 100 094, Dec. 15, 2021, ISSN: 2666-8270. DOI: 10.1016/j.mlwa.2021.100094. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827021000475 (visited on 05/13/2025).

[47] S. Lundberg. "Interpretable Machine Learning with XGBoost," TDS Archive. (Feb. 2, 2025), [Online]. Available: https://medium.com/data-science/interpretable-machine-learning-with-xgboost-9ec80d148d27 (visited on 05/13/2025).

A

Results of Model Setup 1

Results from training two Random Forest classifiers using cross-validation with five folds. The mean accuracy and mean AUC from each fold's performance for various preprocessing settings are displayed in Table A.1.

Table A.1: Model performance for different preprocessing settings of Pre- and Post-exercise data.

Preprocessing settings	**Pre-exercise**		**Post-exercise**	
	Mean accuracy:	**Mean AUC:**	**Mean accuracy:**	**Mean AUC:**
Original	0.867 ± 0.083	0.961 ± 0.054	0.925 ± 0.061	1.000 ± 0.000
RMS normalization	0.867 ± 0.083	0.967 ± 0.044	0.925 ± 0.061	0.987 ± 0.027
Peak normalization	0.844 ± 0.133	0.886 ± 0.115	0.925 ± 0.061	0.987 ± 0.026
High-pass filtered cutoff: 80 Hz	0.889 ± 0.070	0.978 ± 0.044	0.925 ± 0.061	0.987 ± 0.027
High-pass filtered cutoff: 100 Hz	0.867 ± 0.083	0.975 ± 0.032	0.900 ± 0.050	0.987 ± 0.027
High-pass filtered cutoff: 150 Hz	0.889 ± 0.070	0.952 ± 0.043	0.925 ± 0.061	0.987 ± 0.027
High-pass filtered cutoff: 200 Hz	0.867 ± 0.044	0.930 ± 0.082	0.925 ± 0.061	0.973 ± 0.033
High-pass filtered cutoff: 500 Hz	0.844 ± 0.089	0.883 ± 0.123	0.925 ± 0.061	0.987 ± 0.027
High-pass filtered cutoff: 1000 Hz	0.822 ± 0.054	0.902 ± 0.104	0.900 ± 0.050	1.000 ± 0.000
Low-pass filtered cutoff: 14000 Hz	0.844 ± 0.054	0.978 ± 0.044	0.925 ± 0.061	0.973 ± 0.053
Low-pass filtered cutoff: 13000 Hz	0.867 ± 0.083	0.978 ± 0.027	0.925 ± 0.061	0.987 ± 0.027
Low-pass filtered cutoff: 12000 Hz	0.844 ± 0.054	0.941 ± 0.061	0.925 ± 0.061	0.987 ± 0.027
Low-pass filtered cutoff: 10000 Hz	0.844 ± 0.054	0.952 ± 0.066	0.925 ± 0.061	1.000 ± 0.000
Low-pass filtered cutoff: 8000 Hz	0.867 ± 0.083	0.947 ± 0.076	0.925 ± 0.061	1.000 ± 0.000
Low-pass filtered cutoff: 5000 Hz	0.867 ± 0.083	0.930 ± 0.065	0.850 ± 0.050	1.000 ± 0.000

Results from classifying the new dataset using a Random Forest classifier trained on the old dataset without preprocessing, recorded with the OnePlus.

Table A.2: Pre-exercise recordings without preprocessing

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.724	0.276
2	0	0	0.644	0.356
3	1	1	0.440	0.560
4	0	0	0.746	0.254
5	0	1	0.492	0.508

Table A.3: Post-exercise recordings without preprocessing

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	1	0.362	0.638
2	0	0	0.684	0.316
3	1	0	0.596	0.404
4	0	1	0.484	0.516
5	0	0	0.696	0.304

The mean of the predicted probability of the correct class for the pre-exercise and post-exercise models from Table A.2 and A.3 are shown in Table A.4.

Table A.4: The mean of the predicted probability of the correct class for the pre-exercise and post-exercise models.

Dog #	True class	Mean probability (Class 0)	Mean probability (Class 1)	Predicted class
1	0	0.543	0.457	0
2	0	0.664	0.336	0
3	1	0.518	0.482	0
4	0	0.615	0.385	0
5	0	0.594	0.406	0

Results from high-pass filtering the signals with the dynamic filtering approach, with a threshold of 5%.

Table A.5: High-pass filtered **pre-exercise** recordings with an estimated cutoff threshold of 5%.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.756	0.244
2	0	0	0.644	0.356
3	1	1	0.398	0.602
4	0	0	0.840	0.160
5	0	0	0.756	0.244

Table A.6: High-pass filtered **post-exercise** recordings with an estimated cutoff threshold of 5%.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.502	0.498
2	0	0	0.680	0.320
3	1	0	0.606	0.394
4	0	0	0.522	0.478
5	0	0	0.834	0.166

The mean of the predicted probability of the correct class for the pre-exercise and post-exercise models from Table A.5 and A.6 are shown in Table A.7.

Table A.7: The mean of the predicted probability of the correct class for the pre-exercise and post-exercise models.

Dog #	True class	Mean probability (Class 0)	Mean probability (Class 1)	Predicted class
1	0	0.629	0.371	0
2	0	0.662	0.338	0
3	1	0.502	0.498	0
4	0	0.681	0.319	0
5	0	0.795	0.205	0

B

Results of Model Setup 2

Results from training a Random Forest classifier using cross-validation with five folds. The mean accuracy and mean AUC from each fold's performance for various preprocessing settings are displayed in Table B.1.

Table B.1: Model performance for different preprocessing settings.

Preprocessing settings	**Mean accuracy:**	**Mean AUC:**
Original	0.960 ± 0.080	1.000 ± 0.000
RMS normalized	0.929 ± 0.098	1.000 ± 0.000
High-pass filtered estimated cutoff threshold 5%	0.920 ± 0.098	1.000 ± 0.000
High-pass filtered estimated cutoff threshold 10%	0.920 ± 0.098	1.000 ± 0.000
High-pass filtered estimated cutoff threshold 15%	0.920 ± 0.098	1.000 ± 0.000
High-pass filtered estimated cutoff threshold 20%	0.960 ± 0.080	1.000 ± 0.000
RMS normalized and high-pass filtered with threshold 15%	0.880 ± 0.160	0.933 ± 0.133
High-pass filtered with threshold 15% and low-pass filtered with threshold 95%	0.813 ± 0.128	1.000 ± 0.000
High-pass filtered with threshold 15% and low-pass filtered with threshold 90%	0.893 ± 0.088	0.975 ± 0.050
RMS normalized, high-pass filtered with threshold 15% and low-pass filtered with threshold 95%	0.853 ± 0.075	1.000 ± 0.000
Data augmented, RMS normalized, high-pass filtered with threshold 15% and low-pass filtered with threshold 95%	0.964 ± 0.073	1.000 ± 0.000
Data augmented, high-pass filtered with threshold 15% and low-pass filtered with threshold 95%	1.000 ± 0.000	1.000 ± 0.000

Original and RMS normalized data

Table B.2: Original data without preprocessing.

Dog #	**True class**	**Predicted class**	**Probability (Class 0)**	**Probability (Class 1)**
1	0	1	0.460	0.540
2	0	0	0.588	0.412
3	1	0	0.570	0.430
4	0	0	0.770	0.230
5	0	0	0.648	0.352

Table B.3: RMS normalized signals.

Dog #	**True class**	**Predicted class**	**Probability (Class 0)**	**Probability (Class 1)**
1	0	1	0.452	0.548
2	0	0	0.516	0.484
3	1	1	0.488	0.512
4	0	0	0.752	0.248
5	0	0	0.660	0.340

Dynamic high-pass filtering with different thresholds

Table B.4: High-pass filtered with an estimated cutoff threshold of 5%.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.594	0.406
2	0	0	0.570	0.430
3	1	1	0.496	0.504
4	0	0	0.886	0.114
5	0	0	0.890	0.110

Table B.5: High-pass filtered with an estimated cutoff threshold of 10%.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.552	0.448
2	0	0	0.620	0.380
3	1	1	0.404	0.596
4	0	0	0.810	0.190
5	0	0	0.820	0.180

Table B.6: High-pass filtered with an estimated cutoff threshold of 15%.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.564	0.436
2	0	0	0.682	0.318
3	1	1	0.414	0.586
4	0	0	0.786	0.214
5	0	0	0.818	0.182

Table B.7: High-pass filtered with an estimated cutoff threshold of 20%.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.534	0.466
2	0	0	0.690	0.310
3	1	0	0.544	0.456
4	0	0	0.822	0.178
5	0	0	0.812	0.188

High-pass filtering with a 15% threshold and RMS normalization

Table B.8: RMS normalized and high-pass filtered with an estimated cutoff threshold of 15%.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.580	0.420
2	0	0	0.856	0.144
3	1	1	0.480	0.520
4	0	0	0.730	0.270
5	0	0	0.890	0.110

High-pass filtering with a 15% threshold, RMS normalization and/or Data augmentation

Table B.9: Data augmentation by randomly amplifying and reducing the amplitude with 10% and high-pass filtered with an estimated cutoff threshold of 15%.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.526	0.474
2	0	0	0.672	0.328
3	1	1	0.338	0.662
4	0	0	0.816	0.184
5	0	0	0.834	0.166

Table B.10: Data augmentation by randomly amplifying and reducing the amplitude with 10% and high-pass filtered with an estimated cutoff threshold of 15% and RMS normalized.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.604	0.396
2	0	0	0.832	0.168
3	1	1	0.416	0.584
4	0	0	0.710	0.290
5	0	0	0.932	0.068

Combination of the best data augmentation, and high-pass filtering cutoff frequency thresholds.

Table B.11: Data augmentation by randomly amplifying the amplitude with 15% and reducing the amplitude with 5% and high-pass filtered with an estimated cutoff threshold of 10% and RMS normalized.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.542	0.458
2	0	0	0.910	0.090
3	1	0	0.604	0.396
4	0	0	0.914	0.086
5	0	0	0.818	0.182

Table B.12: Data augmentation by randomly amplifying the amplitude with 15% and reducing the amplitude with 5% and high-pass filtered with an estimated cutoff threshold of 15% and RMS normalized.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.638	0.362
2	0	0	0.824	0.176
3	1	1	0.402	0.598
4	0	0	0.776	0.224
5	0	0	0.878	0.122

Results from classifying the Samsung recordings

Table B.13: High-pass filtered with an estimated cutoff threshold of 15%

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	1	0.422	0.578
2	0	0	0.712	0.288
3	1	1	0.366	0.634
4	0	0	0.692	0.308
5	0	0	0.612	0.388

Table B.14: High-pass filtered with an estimated cutoff threshold of 15% and RMS normalized

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	1	0.254	0.746
2	0	0	0.558	0.442
3	1	1	0.284	0.716
4	0	0	0.666	0.334
5	0	0	0.784	0.216

Table B.15: Data augmentation by randomly amplifying and reducing the amplitude with 10% and high-pass filtered with an estimated cutoff threshold of 15% and RMS normalized

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	1	0.202	0.798
2	0	0	0.554	0.446
3	1	1	0.166	0.834
4	0	0	0.720	0.280
5	0	0	0.868	0.132

Table B.16: Data augmentation by randomly amplifying the amplitude with 15% and reducing the amplitude with 5% and high-pass filtered with an estimated cutoff threshold of 15% and RMS normalized

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	1	0.224	0.776
2	0	0	0.540	0.460
3	1	1	0.160	0.840
4	0	0	0.670	0.330
5	0	0	0.868	0.132

Preprocessing results when inputting OnePlus and Samsung recordings respectively

The mean accuracy and mean AUC from the 5 folds used for cross-validation on the training data is included in Table B.1 for each preprocessing setting used below.

Results from high-pass filtering with an estimated cutoff threshold of 15% and low-pass filtering with an estimated cutoff threshold of 95% are shown in Table B.17 and B.18.

Table B.17: Classification of the OnePlus recordings.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.708	0.292
2	0	0	0.682	0.318
3	1	1	0.234	0.766
4	0	0	0.714	0.286
5	0	0	0.700	0.300

Table B.18: Classification of the Samsung recordings.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.728	0.272
2	0	0	0.772	0.228
3	1	1	0.462	0.538
4	0	0	0.716	0.284
5	0	0	0.528	0.472

Results from high-pass filtering with an estimated cutoff threshold of 15% and low-pass filtering with an estimated cutoff threshold of 90% are shown in Table B.19 and B.20.

Table B.19: Classification of the OnePlus recordings.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.680	0.320
2	0	0	0.660	0.340
3	1	1	0.490	0.510
4	0	0	0.668	0.332
5	0	0	0.672	0.328

Table B.20: Classification of the Samsung recordings.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.690	0.310
2	0	0	0.828	0.172
3	1	1	0.366	0.634
4	0	0	0.772	0.228
5	0	0	0.542	0.458

Results from using RMS normalization, high-pass filtering with an estimated cutoff threshold of 15% and low-pass filtering with an estimated cutoff threshold of 95% are shown in Table B.21 and B.22.

Table B.21: Classification of the OnePlus recordings.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.820	0.180
2	0	0	0.884	0.116
3	1	1	0.336	0.664
4	0	1	0.466	0.534
5	0	0	0.864	0.136

Table B.22: Classification of the Samsung recordings.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.760	0.240
2	0	0	0.842	0.158
3	1	1	0.468	0.532
4	0	0	0.780	0.220
5	0	0	0.800	0.200

Results from using RMS normalization, high-pass filtering with an estimated cutoff threshold of 15%, low-pass filtering with an estimated cutoff threshold of 95%, and data augmentation by randomly amplifying the amplitude with 15% and reducing the amplitude with 5% are shown in Table B.23 and B.24.

Table B.23: Classification of the OnePlus recordings.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.782	0.218
2	0	0	0.870	0.130
3	1	1	0.254	0.746
4	0	1	0.430	0.570
5	0	0	0.846	0.154

Table B.24: Classification of the Samsung recordings.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.762	0.238
2	0	0	0.840	0.160
3	1	1	0.430	0.570
4	0	0	0.786	0.214
5	0	0	0.796	0.204

Results from high-pass filtering with an estimated cutoff threshold of 15%, low-pass filtering with an estimated cutoff threshold of 95%, and data augmentation by randomly amplifying the amplitude with 15% and reducing the amplitude with 5% are shown in Table B.25 and B.26.

Table B.25: Classification of the OnePlus recordings.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.808	0.192
2	0	0	0.756	0.244
3	1	1	0.162	0.838
4	0	0	0.664	0.336
5	0	0	0.732	0.268

Table B.26: Classification of the Samsung recordings.

Dog #	True class	Predicted class	Probability (Class 0)	Probability (Class 1)
1	0	0	0.732	0.268
2	0	0	0.792	0.208
3	1	1	0.342	0.658
4	0	0	0.772	0.228
5	0	0	0.528	0.472

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY