Towards Automated, Robust, and Reproducible Voice Pathology Detection: A Machine Learning Approach

Jan Vrba[1,3*], Jakub Steinbach[1,3], Tomáš Jirsa[1], Laura Verde[2], Roberta De Fazio[2], Noriyasu Homma[3], Yuwen Zeng[3], Kei Ichiji[3], Lukáš Hájek[1], Zuzana Sedláková[1], Jan Mareš[1]

[1*]Department of Mathematics, Informatics, and Cybernetics, University of Chemistry and Technology, Prague, Technická 5, Prague - Dejvice, 166 28, Czech Republic.

[2]Department of Mathematics and Physics, University of Campania "Luigi Vanvitelli", Viale Abramo Lincoln 5, Caserta, 81100, Italy.

[3]Department of Radiological Imaging and Informatics, Tohoku University Graduate School of Medicine, 2-1-1 Katahira, Aoba-ku, Sendai, 980-8577, Japan.

*Corresponding author(s). E-mail(s): jan.vrba@vscht.cz;
Contributing authors: jakub.steinbach@vscht.cz; tomas.jirsa@vscht.cz; laura.verde@unicampania.it; roberta.defazio@unicampania.it; homma@tohoku.ac.jp; yuwen@tohoku.ac.jp; ichiji@tohoku.ac.jp; lukas.hajek@vscht.cz; zuzana2.sedlakova@vscht.cz; jan.mares@vscht.cz;

Abstract

Voice pathology is a recurrent issue affecting a substantial portion of the population. Machine learning (ML) models and their training on various databases, can enhance and support the diagnosis.

In this study, we propose a robust set of features derived from a thorough research of contemporary practices in voice pathology detection. The feature set is based on the combination of acoustic handcrafted features. Additionally, we introduce pitch difference as a novel feature. We combine this feature set, containing data from the publicly available Saarbrücken Voice Database (SVD), with preprocessing using the K-Means Synthetic Minority Over-Sampling Technique algorithm to address class imbalance.

Moreover, we applied multiple ML models as binary classifiers. We utilized support vector machine, k-nearest neighbors, naive Bayes, decision tree, random forest and AdaBoost classifiers. To determine the best classification approach, we performed grid search on feasible hyperparameters of respective classifiers and subsections of features.

Our approach has achieved the state-of-the-art performance, measured by unweighted average recall in voice pathology detection on SVD database. We intentionally omit accuracy as it is highly biased metric in case of unbalanced data compared to aforementioned metrics. The results are further enhanced by eliminating the potential overestimation of the results with repeated stratified cross-validation. This advancement demonstrates significant potential for the clinical deployment of ML methods, offering a valuable tool for an objective examination of voice pathologies. To support our claims, we provide a publicly available GitHub repository with DOI 10.5281/zenodo.13771573. Finally, we provide REFORMS checklist.

Keywords: voice pathology detection, machine learning, acoustic features, synthetic minority over-sampling technique

1 Introduction

In the era of Big Data, AI and digital Twin, the support of technologies for monitoring of people's health, detection of specific diseases, optimizing the effectiveness of therapy, and defining personalized and precision medicine is becoming increasingly widespread. Pathologies such as voice disorders can benefit from effective tools such as these technologies for their early detection.

Voice disorders are characterized by the alteration of voice quality due to functional or morphological changes. Although these disorders are widespread, they are often underestimated, which can delay proper healing. Voice disorder diagnosis consists of several medical examinations, including acoustic analysis. This involves the processing and analysis of various acoustic features of the voice signal, which can reveal changes or alterations in voice quality caused by specific diseases [1]. In this context, AI can be a valid and powerful tool. However, utilization of these solutions imposes many additional tasks, ranging from data collection and preprocessing to ground truth labeling and correct algorithm selection, as well as correct experimental setup.

Moreover, the choice of appropriate acoustic features is fundamental. The use of acoustic features to characterize pathological voice quality has been investigated in a variety of contexts and for a variety of purposes [2, 3]. These features can provide a quantitative method of assessing voice characteristics that are otherwise difficult to measure. However, there is no standardized set of acoustic measures, making the selection of appropriate acoustic metrics and their interpretation an ongoing challenge.

In this paper on voice pathology detection, we highlight the importance of training classifiers separately for male and female patients. As we do not investigate the relationship between feature causality or correlation with pathology and we work with a large number of features, we employ ML algorithms as suitable classifiers. Our feature selection process revealed that the optimal feature sets may differ between sexes and classifiers. To identify robust feature sets, we trained multiple classifiers with various hyperparameters for different combinations of feature subsets (see Table 2). By estimating the mean MCC, we were able to identify the most promising combinations of feature sets and classifiers. Subsequently, we conducted repeated stratified 10-fold cross-validation on top performing combinations of classifiers and feature subsets to estimate the average performance metrics and their corresponding standard deviations. Contrary to many studies, our approach encompasses all possible pathologies rather than focusing on a selected few, ensuring that our models reflect the diversity of pathologies present in the population.

The document is organized as follows. In Section 2, we explore the current state-of-the-art practices in the field of pathology detection from voice recordings. In Section 3, we describe the dataset that was used for the development of our feature set. In Section 4, we describe the first preprocessing step to obtain the feature matrix. In Section 5, we explain the augmentation technique used to address the class imbalance in the used dataset. In Section 6, we describe the algorithms used as classifiers for healthy and pathological patients. In Section 7, we introduce the metrics that we used to determine the effectiveness of the developed model, with a proper explanation of the chosen metrics. Lastly, in the Section 8, we interpret the results and compare them with the state-of-the-art models and summarize our findings in the Section 9.

2 Related Works

We used the Scopus database to conduct a research of current works in the field of voice pathology detection and classification from voice recordings. Specifically, we used two search strings:

- ("machine learning" OR "deep learning") AND TITLE ("voice pathology"),
- ("machine learning" OR "deep learning") AND TITLE ("voice disorder").

The search resulted in a wide variety of articles. There was an observable trend in the data used, The majority of works utilized data from the SVD [4], MEEI database [5], AVPD [6], VOICED [7, 8], and FEMH data collected for the 2018 FEMH Voice Data Challenge [9]. A few works used their own datasets to train the classifier. We decided to concentrate primarily on studies using the SVD (see Section 3 for more information about the data) due to two main reasons. First, to the authors' best knowledge, SVD and VOICED are the only publicly available databases from the mentioned, disqualifying the remaining databases from being used as they do not allow to reproduce any experiments conducted on this data. Additionally, the preference for the SVD comes from its size, being larger than VOICED, and from the range of pathologies it contains. As we aim to develop a model for voice pathology detection, the data should also represent rare diseases along with the common ones. Note, that the SVD is unique in that it includes multiple recordings from the same subjects, which could result in data leakage if not properly managed.

In [10], authors of the study test various acoustic features in combination with XGBoost, IsolationForest, and DenseNet models to determine the pathologic samples, reaching an F1 score of 73.3% and UAR we computed as 73.3%. While, in [11], features derived from self-supervised learning models, Data2Vec and Wav2Vec, alongside MFCC are explored. The reliability of these features to evaluate voice quality is tested using SVM and DNN, achieving an accuracy of 77.83% and UAR we computed as 77.86%.

A different approach is taken in [12], where the authors transform sound wave data into spectrograms and treat classification as an image recognition problem using a CNN, achieving 73.93% accuracy and UAR we computed as 70.68%.

In [13], the authors use MFCC and GFCC with a classifier based on a NN, achieving 81.84% accuracy. Unlike other studies that use the /a:/ sound for feature extraction, they employ whole sentences. Unfortunately, we were not able to compute UAR, due to the erroneously reported results. Another study [14] tests a DNN with convolutional layers, using the voice signal as a feature set for the convolutional layer, achieving 68.08% accuracy and UAR we computed as 72.32%. In [15], the authors combine CNN-based feature extractors with various classifiers, such as SVM and DNNs, achieving an UAR score of 84.97% on the entire SVD dataset.

Finally, in [16], the authors test a combination of SVM and CNN-based feature extractors on spectrograms of sound and EGG signals. They integrate these with traditional acoustic features like MFCC, LPC, and f_0, reaching an accuracy of 90.10% and UAR we computed as 88.75%. Similarly, they reach an accuracy of 87.41% by extracting the mentioned features and using SVM for classification. However these results were not obtained using cross-validation or train-test split.

Additionally, there were several works that took a subset of the SVD based on selected pathologies, among the researched works.

In [17], the authors extract various acoustic features from the time domain, such as f_0, jitter, shimmer, and HNR, along with age and sex information. They train ML models, including boosted trees, SVM, DT, NB, and KNN, achieving the highest accuracy of 84.5% with a NB model on an imbalanced dataset and UAR we computed as 84.55%. In another study [18], they expand their work by incorporating MFCC and their derivatives, and test additional classifiers such as the logistic model tree and instance-based learning algorithms, reaching 85.77% accuracy with a SVM model and UAR we computed as 85.76%. However, they utilized only subset of SVD.

In [19], the authors extract features based on MFCC and self-supervised algorithms from samples of healthy individuals and patients with hyperfunctional dysphonia and vocal fold paresis. Using SVM, they achieve 75.65% and 74.50% accuracy for male and female patients, respectively. Compared to that, [20] focuses on developing classifiers to distinguish between healthy subjects and patients suffering from spasmodic dysphonia and laryngeal nerve paralysis. Their exploration of multi-modal classification methods results in an accuracy of 68.11%.

Further expanding the scope, [21] investigates multi-modal classification for patients with various conditions, including dysphonia, laryngitis, Reinke's edema, vox senilis, and central laryngeal motion disorder. By employing SVM and NN, they achieve an average accuracy of 88.46%.

Similarly, [22] explores multimodal classification for conditions such as nodules, polyps, edema, and paralysis, as well as binary classification between healthy and unhealthy individuals. They utilize a SMOTE-based method for balancing datasets and reach a maximum F1-score of 90% with a CNN-based model for binary classification and we computed UAR as 90%. However, they do not describe how they handle duplicities in SVD dataset and very likely introduce the data leakage due their methodology.

In another study [23], the combination between MFCC and log-mel-frequency spectral coefficients with deep models, using recordings from the SVD and their own database is proposed. This approach results in an accuracy of up to 81.6%. Meanwhile, [24] focuses on patients with dysphonia, chronic laryngitis, and vocal cord paralysis, employing features extracted by a VGGish model in combination with a LSTM network. They achieve an F1-score of up to 80% in distinguishing between healthy individuals and those with paralysis.

In [25], the author tests quantitative voice parameters combined with a MLP model to classify healthy individuals and those with hyperfunctional dysphonia, laryngitis, and recurrent laryngeal nerve paralysis, achieving 87.5% accuracy.

Additionally, [26] selects 280 samples for pathology detection, extracting MFCC features and using them in a DT model, which results in an accuracy of 67.9%.

Expanding the feature set, [27] uses MFCC, first and second derivatives of MFCC, LPC, and constant-Q cepstral coefficients with a Bi-LSTM to classify eight selected pathologies, namely dysody, dysphonia, functional dysphonia, hyperfunctional dysphonia, hypofunctional dysphonia, spasmodic dysphonia, vocal cyst polyp, and healthy individuals, achieving an accuracy of 92.7%, which is the only metrics they provide. Finally, [28] examines the impact of MFCC extraction window length on detecting pathologies in patients with dysphonia and reflux laryngitis. Using SVM, they achieve up to 75.1% accuracy and UAR we computed as 75%.

2.1 Remarks on Deviations from Consensus-Based Reporting Standards in Machine Learning Research

During reviewing the works that we introduced in this section, we noticed that none of the researched work handled the potential data leakage introduced by the existence of multiple recordings produced by the same patient. Moreover, some works might have introduced data leakage by scaling data and oversampling before splitting into training and testing subsets. Then, we observed that most works did not report proper performance metrics that reflect the imbalanced dataset. Also, none of the works shared working code to reproduce the results and most works did not include enough information about the data selection, preprocessing methods, models and their parameters, hyperparameters, and architectures, or the validation process.

As mentioned in the previous text, some works took a subset of the available data which, according to us, severely limits the applicability in clinical practice.

Due to these facts, unfortunately, we suspect that most of the reported performances might be overoptimistic.

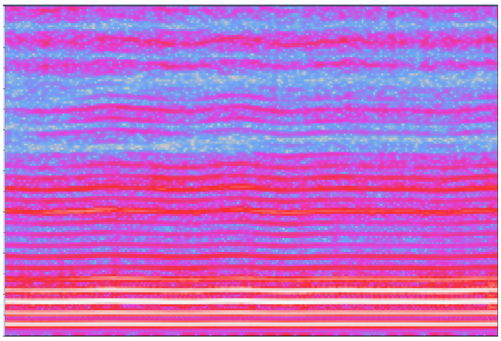


Fig. 1: Healthy female subject ID 1

3 Data

Voice pathology detection studies often utilize the MEEI [5], VOICED [7, 8], FEMH Voice Data Challenge 2018 [9], and AVPD [6] datasets. However, the MEEI dataset is no longer available, and the FEMH dataset is not publicly available. We do not consider the AVPD dataset feasible for our study, as it includes only five types of pathologies: vocal fold cysts, nodules, paralysis, polyps, and sulcus. Thus, it does not reflect the number of various pathologies presented in the general population. The VOICED database comprises 20 different pathologies, but it contains a limited sample size of 208 recordings.

We used the SVD [4] for our work, as was already mentioned in Section 2. The SVD was developed by the Phonetics group at the Department of Language Science and Technology, Saarland University and is obtainable at https://stimmdb.coli.uni-saarland.de/help_en.php4.

The dataset contains data from 1853 patients and includes different types of voice recordings, such as pronunciations of various vowels in low, neutral, and high pitches, as well as tones alternating between high and low pitches. Additionally, the dataset features recordings of patients saying "Guten Morgen, wie geht es Ihnen?" (Good morning, how are you?). Alongside the voice recordings, the database also offers EGG signals for download.

The database includes not only recordings but also an overview of patient information specific to each recording. To be precise, the database contains the recording's unique identifier, patient's unique identifier, sex, age at the time of recording, recording date, type of recording, list of pathologies (*pathology* column), list of diagnoses, and any comments. Note that the overview has to be copied manually, therefore, we include it in a file named `svd_information.csv` in the provided repository. The full list of files downloaded for this work is included in the repository along with sha256 checksum to ensure reproducibility.

The outcome variable is derived from the file containing patient information. We create a new variable called *pathology_binary* that takes the value of 0 if there are no values in the pathology column and the value of 1 if one or more pathologies are listed in the column.

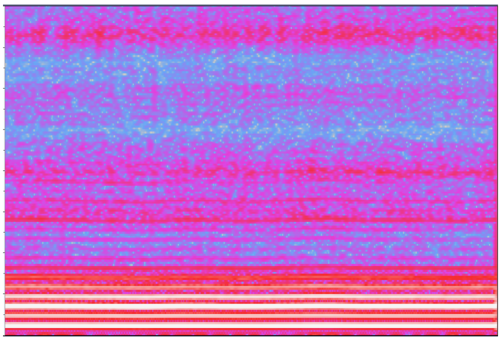


Fig. 2: Female subject ID 1, suffering with functional dysphonia

For our research, we used the recordings containing the sound /a:/ in neutral pitch for the pathology detection model due to its frequent use in previous studies and clinical protocols [29, 30]. Figure 1 and Figure 2 presents examples of healthy and pathological spectrograms, illustrating the /a:/ sound at a neutral pitch from a healthy female subject (patient ID 1) and the same subject with functional dysphonia.

Before the feature extraction, we had to consider some limitations to ensure robust and unbiased results in voice pathology detection. The database contains samples from underage patients. Due to developmental changes at a young age, we decided to exclude any recordings of patients younger than 18 years old. Research [31, 32] has shown that during development, the characteristics of young voices, such as the fundamental frequency, are distinct from those of fully developed voices. These developmental changes could cause problems for the classifier and reduce its ability to accurately detect pathology. By excluding these age groups, we aimed to maintain a more consistent and reliable dataset for our analysis.

Another significant issue is the presence of multiple recordings for some patients. For example, the patient with the talker ID 2027 has 24 recordings of the /a:/ sound in a neutral pitch. This repetition poses a risk of data leakage, potentially skewing the results if not properly addressed.

To mitigate this risk, we implemented a reproducible selection process for our dataset. For patients with multiple recordings of the same type (either all healthy or all pathological), we retained only the oldest recorded sample (based on the recording date and session unique identifier). For patients with both healthy and pathological recordings, we selected the oldest sample of each type, resulting in a maximum of two recordings per patient — one healthy and one pathological. We believe this approach minimizes the likelihood of the classifier learning patient identities, as the patient's state remains independent of their identity.

In addition to this selection process, we excluded specific recordings labeled *1573-a-n.wav* and *87-a-n.wav*. The former contained two distinct sound recordings, while the latter was corrupted by an artifact likely caused by hardware or software errors.

Furthermore, to enhance the quality of the recordings used for feature extraction, we trimmed all selected recordings to eliminate any potential silent parts. Utilizing the *librosa* library [33], we split the recording into overlapping windows and identified and removed quiet sections that were 15 dB quieter than the maximum root mean square value of the amplitude in the analyzed window, and finally composed the trimmed recording.

The preprocessing step resulted in 1658 edited recordings containing 66 pathologies. The pathologies excluded based on the age and duplicity of recordings are *Carcinoma in situ*, *Dysplastische Dysphonie* (Dysplastic Dysphonia), *Juvenile Dysphonie* (Juvenile Dysphonia), *Psychogene Aphonie* (Psychogenic Aphonia), and *Psychogene Mikrophonie* (Psychogenic Microphonia) from our data for feature extraction. The age distribution of data split by sex and pathology are described in Table 1 as well as the Figure 3 and Figure 4. The complete list of recordings excluded from the experiment, along with the reason for their exclusion, are included in the repository in the file `misc/list_of_excluded_files.csv`.

Table 1: Age distribution among the sex and voice state in the used data

	Female		Male	
	Healthy	Pathological	Healthy	Pathological
Mean	25.39	48.26	31.41	52.12
Standard deviation	11.19	15.35	11.55	15.21
Minimum	18.00	18.00	18.00	18.00
25% percentile	20.00	36.00	22.00	40.00
50% percentile	21.00	49.00	28.00	55.00
75% percentile	24.25	60.00	38.00	63.00
Maximum	84.00	94.00	69.00	89.00
Count	408	555	252	443

The whole preprocessing workflow is illustrated in Figure 5.

4 Feature Extraction

Due to the nature of the data, our first step after trimming the quiet parts of the recordings was extracting information in the form of various acoustic features. Acoustic features can be categorized into several groups based on the type of transformation of the sound signal: time-domain, spectral, and cepstral features. We depended on various Python libraries for feature extraction, namely *parselmouth* for features related to f_0 and formants, *spkit* for Shannon entropy, *torchaudio* for LFCC, *librosa* for the remaining acoustic features, and scikit-learn and numpy for calculating statistical values from the extracted features or from the signal. Table 2 describes which library was used to extract each feature.

4.1 Time-Domain Features

The majority of the time-domain features stem from the measurement of regularity of the sound signal. The idea behind the measurement of (ir)regularities in a voice recording is that human voice is composed of periodic and random components and

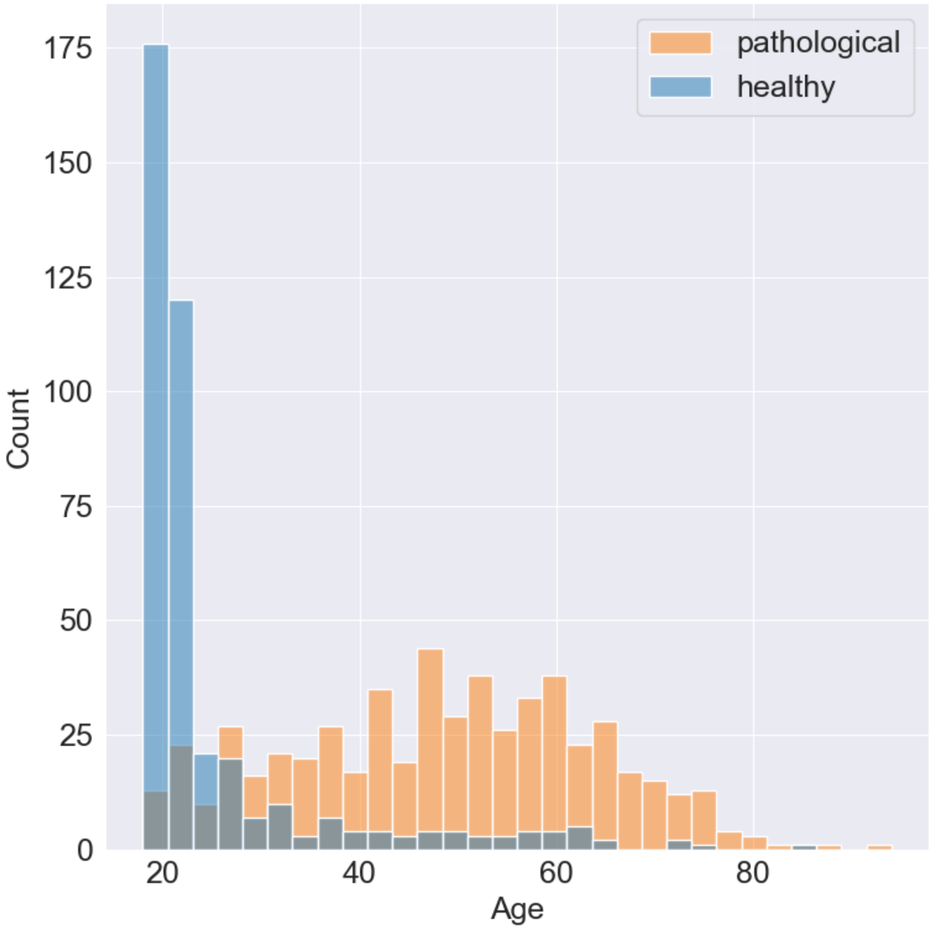


Fig. 3: Age distribution of healthy and pathological female subjects

there is a difference between the amount of irregularity between the healthy and pathological voices.

4.1.1 Fundamental frequency

The periodic part is characterized by f_0. Voice pathologies usually increase the ratio between the random and periodic components, also altering f_0. Thus, pathologies might be observable from the change of these parameters.

The main disadvantage of several time-domain measures is the necessity to estimate f_0. In heavily altered voices, caused by some pathologies, it might be impossible to estimate f_0. There are also multiple algorithms for f_0 estimation and generally, each may yield different results.

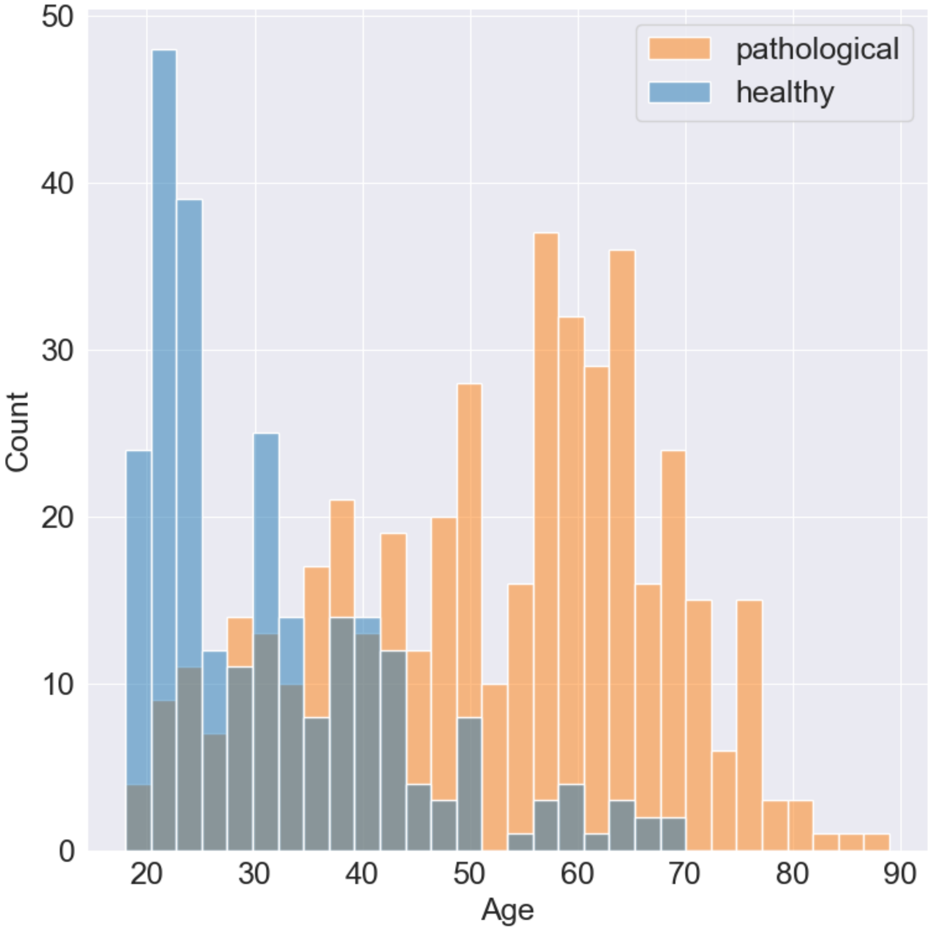
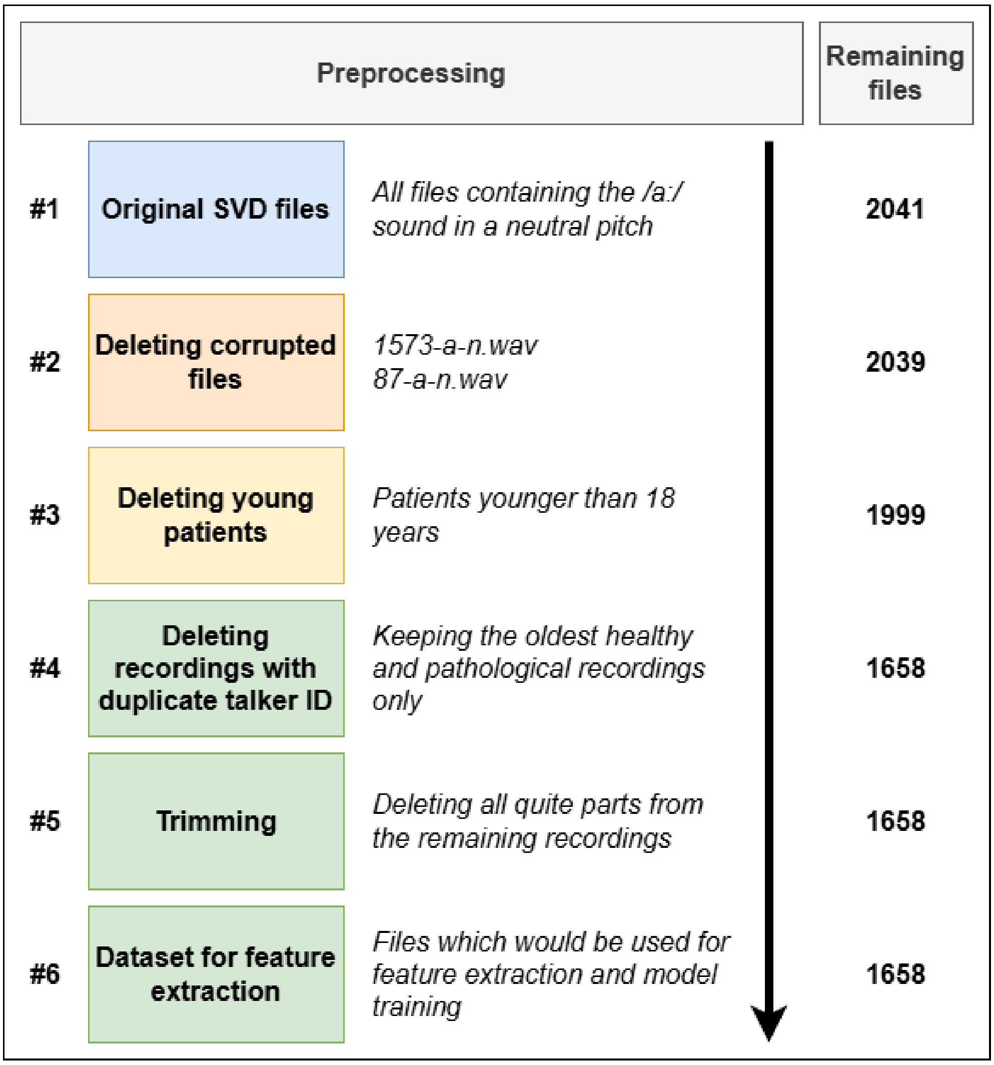


Fig. 4: Age distribution of healthy and pathological male subjects

We used the *parselmouth* library, a Python implementation of the Praat program [39], to determine f_0, therefore, we utilized the algorithm described in [40] to extract f_0 for N segments of the samples, denoted as $\mathbf{f}_0$. Given that the estimation of f_0 is calculated for individual segments of each recording, we took the mean (Equation 1) and sample standard deviation (Equation 2) of f_0 as features for each sample.

$$\overline{f}_0 = \frac{1}{N}\sum_{i=0}^{N-1} f_{0,i} \tag{1}$$

$$\sigma_{\mathbf{f}_0} = \sqrt{\frac{\sum_{i=0}^{N-1}(f_{0,i} - \overline{f}_0)^2}{N-1}} \tag{2}$$



	Preprocessing		Remaining files
#1	Original SVD files	*All files containing the /a:/ sound in a neutral pitch*	2041
#2	Deleting corrupted files	*1573-a-n.wav* *87-a-n.wav*	2039
#3	Deleting young patients	*Patients younger than 18 years*	1999
#4	Deleting recordings with duplicate talker ID	*Keeping the oldest healthy and pathological recordings only*	1658
#5	Trimming	*Deleting all quite parts from the remaining recordings*	1658
#6	Dataset for feature extraction	*Files which would be used for feature extraction and model training*	1658

Fig. 5: Preprocessing before feature extraction

4.1.2 Pitch Difference

We determine the pitch difference as a difference between the maximum and minimum value of the extracted f_0 (Equation 3). Our reasoning is that a healthy voice should maintain a stable frequency for the full duration of the recording compared to the pathological voice which may fluctuate.

$$\Delta f_0 = \frac{\max(\mathbf{f}_0) - \min(\mathbf{f}_0)}{\min(\mathbf{f}_0)} \tag{3}$$

4.1.3 Jitter

Jitter, also called fundamental frequency perturbation, is a measure of irregularities in f_0. Jitter is a well-known metric, often used in acoustic analysis for different types of signals. In voice pathology, higher values of jitter usually indicate pathology.

There are several variations of the measure, in our work, we used the local jitter, also known as *jitta* (Equation 4) [39], calculated as the mean difference between periods of consecutive glottal cycles T_i across N glottal cycles divided by the mean glottal cycle period of the signal $\overline{T}$ (Equation 5).

$$\text{jitta} = \frac{\sum_{i=1}^{N-1} |T_i - T_{i-1}|}{(N-1) \cdot \overline{T}} \tag{4}$$

$$\overline{T} = \sum_{i=0}^{N-1} \frac{T_i}{N} \tag{5}$$

4.1.4 Shimmer

Shimmer, also known as amplitude perturbation, measures irregularities in the amplitude of the sound wave and high values of shimmer also indicate potential voice pathology.

There are several variations of shimmer similar to the jitter and we opted for the local shimmer, also called *shim* (Equation 6) [39]. *Shim* is calculated as the mean difference between amplitudes of consecutive glottal cycles A_i across N glottal cycles divided by the mean glottal cycle amplitude of the signal $\overline{A}$ (Equation 7).

$$\text{shim} = \frac{\sum_{i=1}^{N-1} |A_i - A_{i-1}|}{(N-1) \cdot \overline{A}} \tag{6}$$

$$\overline{A} = \sum_{i=0}^{N-1} \frac{A_i}{N} \tag{7}$$

4.1.5 Harmonics-to-Noise Ratio

HNR measures the ratio between the energies of the periodic and aperiodic signal components (Equation 8) [39, 40].

$$\text{HNR} = 10 \log_{10} \frac{\mathcal{X}_{i_{max}}}{1 - \mathcal{X}_{i_{max}}} \tag{8}$$

We use the recommended cross-correlation implementation of the harmonicity algorithm where the energy of the harmonic component in a windowed signal $\mathcal{X}_{k_{max}}$ is calculated as the value of a cross-correlation of the signal amplitude s with the same signal shifted by the length k_{max} where the cross-correlation yields the highest result (Equation 9). Note that the energy is normalized by the cross-correlation of the signal with itself without any shift.

$$\mathcal{X}_{i,k} = \sum_{j=m}^{m+n-1} s_j s_{j+k} \tag{9}$$

In Equation 9, i represents the window index, k represents the index of a value in the i-th window, m represents the index of an element with respect to the current window, and n represents the number of samples in a window. The length of the window is chosen with respect to the expected f_0.

4.1.6 Zero-Crossing Rate

ZCR measures the rate at which a sound signal s changes signs in a signal split to a predetermined number of windows. This feature is not dependent on the estimation of f_0. The calculation for the i-th window is described in Equation 10, where k represents the index of a value in the i-th window, m represents the index of an element with respect to the current window, and n represents the number of samples in a window, set to default value of 2048. Note that $\nVdash$ is described in Equation 11, and $\mathrm{sgn}(x)$ is described in Equation 12.

$$\mathrm{ZCR}_i = \frac{1}{n-1} \sum_{j=m}^{m+n-1} \nVdash(s_j, s_{j+1}) \tag{10}$$

$$\nVdash(s_j, s_{j+1}) = \begin{cases} 1 & \mathrm{sgn}(s_j)\,\mathrm{sgn}(s_{j+1}) = -1 \\ 0 & \mathrm{sgn}(s_j)\,\mathrm{sgn}(s_{j+1}) = +1 \end{cases} \tag{11}$$

$$\mathrm{sgn}(x) = \begin{cases} -1 & x < 0 \\ +1 & x \geq 0 \end{cases} \tag{12}$$

4.1.7 Shannon Entropy

Shannon entropy measures the average level of uncertainty inferent to the possible outcomes of a random variable. Generally, it is defined as in Equation 13 for an independent variable X defined by a set of values $\mathcal{X}$ with a probability of occurrence p.

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{13}$$

In our case, the probabilities p are taken from the histogram of the signal amplitude. The number of bins n_{bins} is determined by the Freedman–Diaconis rule (Equation 14), where N is the length of the signal and Q_3, Q_1 are amplitude values representing the third and first quartiles, respectively.

$$n_{bins} = 2 \cdot \frac{Q_3 - Q_1}{\sqrt[3]{N}} \tag{14}$$

This feature is also independent of f_0.

4.1.8 Skewness

Skewness measures the concentration of values on either side of a data distribution. Based on the results of the study [41], we expected that the skewness value might be influenced by a pathology, and thus, we included this feature in our experiments.

We calculate the skewness according to the definition in Equation 15.

$$\text{skew} = \frac{m_3}{m_2^{3/2}} \tag{15}$$

The individual cumulants m_i can be determined by Equation 16.

$$m_i = \frac{1}{N}\sum_{j=1}^{N}(s_j - \overline{s})^i \tag{16}$$

4.2 Spectral Features

We use spectral features to measure characteristics and differences in the spectra of the sound signal. The most frequent transformation for obtaining spectral information is the Fourier transform. To gain more information about the spectral variations of the signal in time, the signal is split into short windows containing tens to hundreds of milliseconds of the original signal with varying overlaps between these windows. Then, the STFT algorithm is applied to each window to gain the spectrum of each window. Finally, the spectra are concatenated and plotted in the form of a spectrogram - a heatmap showing time on the x-axis, frequency on the y-axis, and power of the spectrum as a color. In regard to the voice pathology detection, the expectations are that due to the difference in the ratio of (dis)harmonic components, the spectral features should be able to distinguish between pathological and healthy voices.

4.2.1 Spectral Flatness

Spectral flatness is used to quantify how much tone-like a signal is, as opposed to being noise-like [42]. It is calculated as a ratio of the geometric mean of the power spectrum S to its arithmetic mean measured across all bands [43] (see Equation 17), where N denotes the total number of bands of the power spectrum.

$$\text{flatness} = \frac{\sqrt[N]{\prod_{n=1}^{N} S(n)}}{\frac{1}{N}\sum_{n=1}^{N} S(n)} = \frac{\exp\left(\frac{1}{N}\sum_{n=1}^{N} \ln S(n)\right)}{\frac{1}{N}\sum_{n=1}^{N} S(n)} \tag{17}$$

In this form, spectral flatness reaches values from 0, representing a pure tone, to 1, representing a white noise with equal energy distribution across all bands. We use the representation from [42, 43], implemented in the *librosa* library [33] for the purpose of our work. Due to practical implementation, the zero values of the power spectrum are replaced by the value of 10^{-10}.

4.2.2 Spectral Roll-Off

Spectral roll-off shows the frequency band such that at least a predetermined part of the spectral energy is contained in this and previous bins. We used the threshold value of 85% during our experiments.

4.2.3 Mean Spectral Centroid

Spectral centroid is the simple spectral shape feature, that is used in some works on voice pathology detection. Assuming the normalized spectrum $\tilde{S}_i$ in the i-th time frame and the set of frequency bins K given as

$$\tilde{S}_i(f) = \frac{|S_i(f)|}{\sum_{j \in K} |S_i(j)|} \ , \tag{18}$$

we can compute the spectral centroid C_i as

$$C_i = \sum_{f \in K} f \tilde{S}_i(f) \ . \tag{19}$$

To get the single-value feature, and avoid the problem with different lengths of recording, we used the mean value of spectral centroid as a feature.

4.2.4 Spectral Contrast

Spectral contrast measures the difference between the mean energy of the top and bottom quantiles of selected energy bands. We calculate the spectral contrast with the *librosa* library, which implements it as described in Algorithm 1.

Algorithm 1 Spectral contrast

```
create a spectrogram using STFT with Hann window of length 2024
divide the frequency domain into 7 bands octave-scale sub-bands, 0 - 200 Hz, 200 - 400 Hz, 400 - 800 Hz, 800 - 1600 Hz, 1600 - 3200 Hz, 3200 - 6400 Hz, 6400 - 11025 Hz (recordings are resampled to 22050 Hz)
for each band $k$ and time window $i$ of length 2048 do
    order the values for each frequency band in each time window in ascending order
    get the $\text{peak}_{k,i}$ value as a mean value from first $n$ values, where $n$ represents the number of values in the top 2%
    get the $\text{valley}_{k,i}$ value as a mean value from last $n$ values, where $n$ represents the number of values in the bottom 2%
    compute spectral contrast of $k$th band and $i$th time windows as $\text{SC}_{i,k} = \text{peak}_{i,k} - \text{valley}_{i,k}$
end for
Calculate the average spectral contrast for each band $\text{SC}_{mean,k}$ (Equation 20), where $N$ represents the total number of time windows.
```

$$\mathrm{SC}_{mean,k} = \frac{1}{N}\sum_{i=0}^{N-1}\mathbf{SC}_{i,k} \tag{20}$$

4.2.5 Formants

Formants characterize peaks in frequency spectra of a human voice. In linguistics, they are often used to differentiate between different vowels. The most prominent are the first two formants, based on which it is usually possible to distinguish between the vowels [44].

We extracted the frequencies of the first three formants using the *parselmouth* library.

4.3 Cepstral Features

While time-domain and spectral features are common in many signal processing fields, the cepstral features are more prominent in the voice analysis. Cepstral features are derived from a cepstrum, which describes the spectral information the same way spectrum describes the time-domain information of the signal.

4.3.1 Mel-Frequency Cepstral Coefficients

Cepstral coefficients are used in voice and musical analysis to highlight differences in the power between individual frequencies of a sound signal.

The construction of a cepstrogram involves several steps. We explain the algorithms implemented in the *librosa* library we used for the feature extraction.

First, we take a power spectrogram of the recording, using the STFT algorithm defined in Equation 21. $X(k, f)$ represents the transformation as a function of time frame k and frequency band f. The algorithm utilizes a window w_{n-k}, in our case, we use a Hann window centered around time frame k with the size $N = 2048$ samples and overlap of $N/4$ or 75%, which are default values of the algorithm.

$$X(k, f) = \sum_{n=-\infty}^{+\infty} x_n \cdot w_{n-k} \cdot \exp\left(-2\pi j f n\right) \tag{21}$$

Next, we transform X into a power spectrogram $P[dB]$ as in Equation 22. In the equation, we describe the transformation of each value in the time frame k and frequency band f.

$$P[\mathrm{dB}]_{k,f} = 10\log_{10}(|X(k, f)|^2) \tag{22}$$

Next, the power spectrogram $P[\mathrm{dB}]$ is transformed to a mel spectrogram $P[\mathrm{mel}]$ using a mel filterbank H. The mel filterbank represents a series of triangular filters with their centers evenly distributed on the mel scale. We decided to choose the default number of filters for the filterbank $M = 128$. The centers would be evenly distributed between $f_{\text{mel, min}} = 0$ and $f_{\text{mel, max}} = 3176$, which represents the maximum frequency of 11025 Hz, half of the sampling frequency. The maximum frequency is chosen with respect to the Nyquist theorem. The formula for transforming hertz frequency to mel scale is in Equation 23 and it can be transformed back by Equation 24.

$$f_{\text{mel}} = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{23}$$

$$f = 700\left(10^{\left(\frac{f_{\text{mel}}}{2595}\right)} - 1\right) \tag{24}$$

The filterbank $H_{m,f}$ for the m-th filter and a frequency band f is then calculated by Equation 25. Note that f_m here represents the frequency center of the m-th filter in Hz and f represents a frequency band in the time frame.

$$H_{m,f} = \begin{cases} 0 & f < f_{m-1} \\ \dfrac{f - f_{m-1}}{f_m - f_{m-1}} & f_{m-1} \leq f < f_m \\ \dfrac{f_{m+1} - f}{f_{m+1} - f_m} & f_m \leq f < f_{m+1} \\ 0 & f > f_{m+1} \end{cases} \tag{25}$$

Now, we can calculate the m-th band of the mel spectrogram $P[\text{mel}]_m$ by multiplying each time frame k of the spectrogram $P[dB]_k$ by the amplitude for each frequency band f of the respective filter bank $H_{m,f}$ as in Equation 26. The resulting mel spectrogram has dimensions $[M, N]$, where M is the number of filters in the mel filterbank and N is the number of frames of the STFT.

$$P[\text{mel}]_m = \sum_{k=0}^{N-1} X_k H_{m,f} \tag{26}$$

Finally, the cepstrogram or the MFCC are calculated by transforming the mel spectrogram with DCT. The l-th band of the cepstrogram C is calculated by Equation 27.

$$C_l = 2 \sum_{m=0}^{M} P_{\text{mel}}[m] \cos\left(\frac{\pi l(2m+1)}{2M}\right) \tag{27}$$

The result is a cepstrogram with dimensions $[L, N]$, where L represents the number of taken MFCC. Since the coefficients are different across the time windows, we took the mean value of each band as well as the variance of the values in each band as our features.

Moreover, we also estimated the first and second derivatives from the extracted MFCC using the Savitzky-Golay filter. Similar to the coefficients, we took the mean values and variance of the derivatives as our features.

4.3.2 Linear-Frequency Cepstral Coefficients

LFCC are constructed similarly to the MFCC with one distinction, the triangular filters applied to the frequency spectrum are spaced equidistantly in the standard frequency scale.

4.4 Feature Dataset Setup

In addition to the aforementioned features, we also included information about the age of the speakers, as age has an effect on the overall healthiness of the voice and because this data is practically always known.

During the extraction of features, we noticed that there were distinct differences in some feature values between male and female patients. This is likely caused by the higher pitch of female voice compared to male, altering features such as f_0 (see Figure 6). Since several features are dependent on f_0, even if indirectly, and the patient's biological sex should be always known during the examination, we decided to treat the data as two separate datasets and train two separate classifiers, one for each sex.

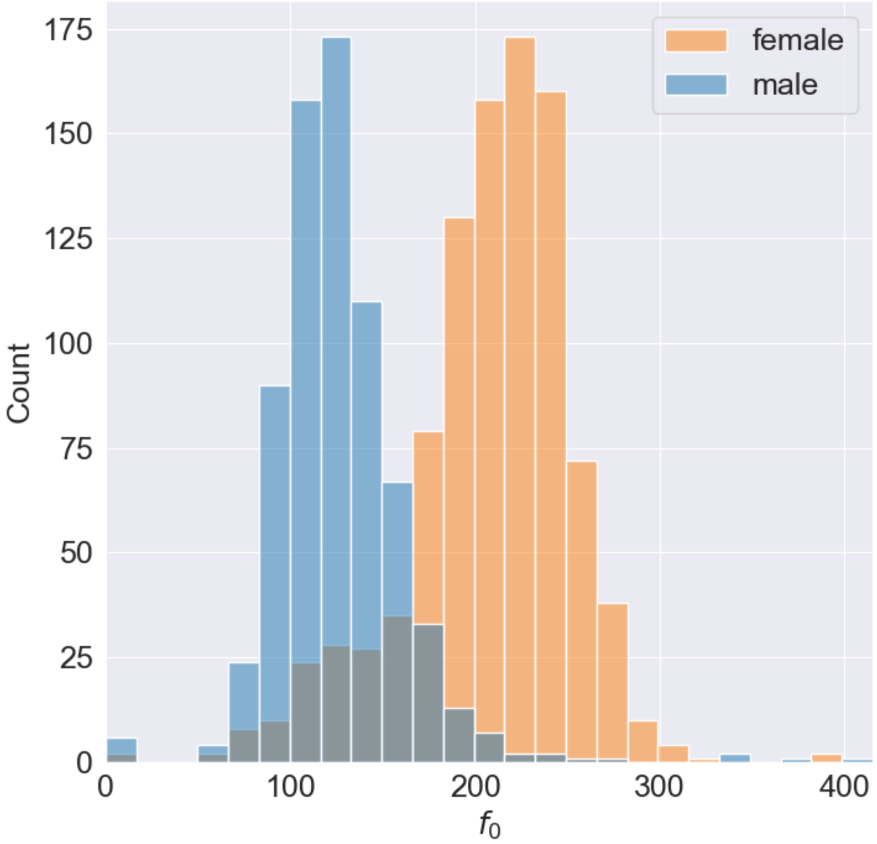


Fig. 6: Distribution of the estimated mean f_0 values among male and female patients

Moreover, for eight recordings, specifically *1338-a_n.wav, 1407-a_n.wav, 1716-a_n.wav, 2235-a_n.wav, 492-a_n.wav, 719-a_n.wav, 720-a_n.wav*, and *915-a_n.wav*, the algorithm failed to extract information about f_0 and therefore, the feature values for μ_{f_0}, Δf_0, σ_{f_0}, jitta, and shim were set to NaN. This was likely caused by the dominance of the disharmonic part caused by serious cases of voice pathology. We decided to replace the NaN values with zero and add a new binary feature that would track the occurrence of NaN values in each row. As this inability to detect the fundamental frequency is an important information about the recording, we consider introducing this feature to be a legitimate approach. Note, that these eight recordings represent 2 female and 6 male subjects in our dataset, and all of them suffer from some form of voice pathology according to the database records.

We tested different configurations of features to see their influence on the classification performance. While always keeping the age, mean f_0, HNR, jitter, and shimmer in each feature configuration, we added all possible combinations of the remaining used features. Note that for MFCC, and their derivatives, the combinations were 13 and 20 coefficients. Only mean values of the coefficients and their derivations were present in all configurations, contrary to variances of MFCC and their derivations, which were considered optional features. The possible configurations of optional features are listed in Table 2. In total, we generated 8192 different datasets for both male and female samples with different configurations of the features. Note, that exhaustive feature selection would lead to evaluation of more than 10^{49} datasets.

See Figure 7 for the workflow of the feature extraction process.

5 Data Augmentation

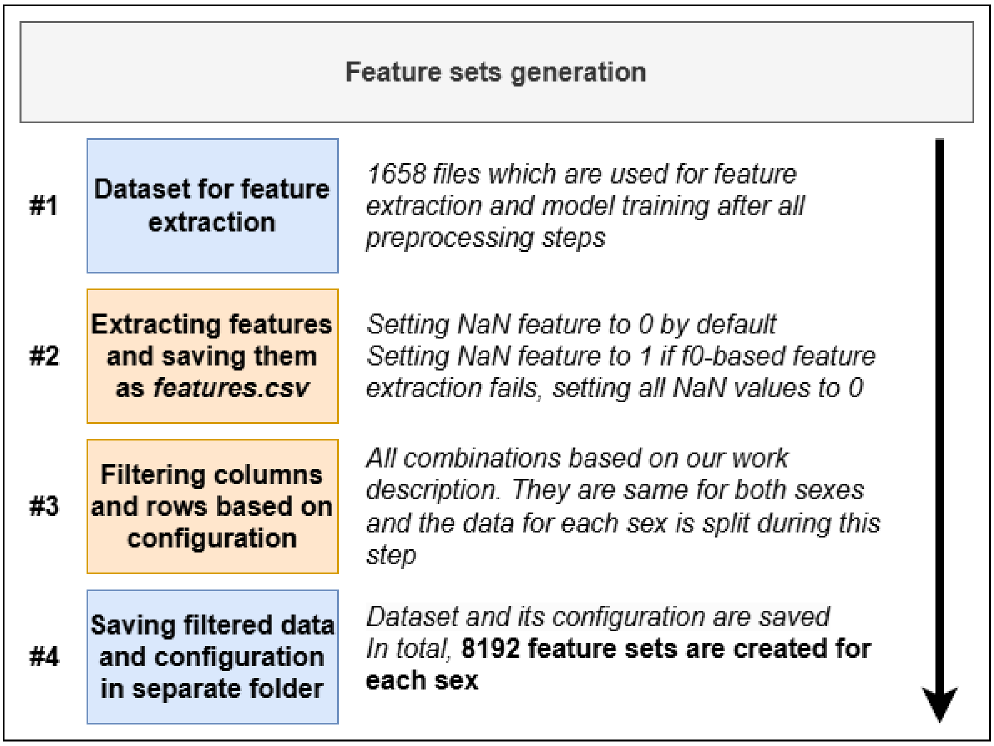
In order to address the issue of class imbalance in utilized dataset, that leads to classifiers with high recall and low specificity, we employed the k-means SMOTE algorithm [45]. This technique was specifically applied to the training dataset to enhance the model's ability to learn from minority class instances and thus improve its predictive performance.

K-means SMOTE is an advanced oversampling method that combines the clustering capabilities of k-means [46] with the synthetic data generation process of SMOTE [47]. The algorithm operates in the following manner:

1. Apply k-means to identify clusters that are dominated by the minority class.
2. Estimate the sparsity of the minority class in these clusters.
3. For each of those clusters, generate the number of samples with respect to the cluster sparsity.
4. Add the newly generated data to the training dataset.

The method has a risk of failing if the locations of the clusters are initialized close to outliers, therefore, we try initializing the method repeatedly, up to a maximum of ten times. If the method fails after ten repetitions, the algorithm is interrupted and SMOTE [47] is used instead.

By applying the SMOTE-based algorithm only to the training data, we aimed to prevent data leakage and ensure that the model's performance evaluation on the test


Feature sets generation
#1
Dataset for feature extraction
#2
Extracting features and saving them as *features.csv*
#3
Filtering columns and rows based on configuration
#4
Saving filtered data and configuration in separate folder

1658 files which are used for feature extraction and model training after all preprocessing steps

Setting NaN feature to 0 by default
Setting NaN feature to 1 if f0-based feature extraction fails, setting all NaN values to 0

All combinations based on our work description. They are same for both sexes and the data for each sex is split during this step

Dataset and its configuration are saved
In total, **8192 feature sets are created for each sex**

Fig. 7: Feature extraction after preprocessing

data remains unbiased. This approach allowed us to mitigate the adverse effects of class imbalance, resulting in a more robust and reliable predictive model.

6 Machine Learning Models

In our research, we selected SVM, KNN, DT, RF, AdaBoost and NB algorithms for binary classification of healthy versus pathological patients. Our choice is based on prior research, where these algorithms were successfully used for binary classification tasks. Additionally, different studies extracted various features from recordings, motivating us to evaluate different feature combinations and test the performance of the classifiers on such combinations. Given that we are dealing with relatively high-dimensional datasets with a limited number of samples, deep learning models would be challenging to apply effectively. Therefore, we believe that using traditional ML models is a suitable approach and aligns with findings from existing research.

All evaluated ML models were implemented via the *scikit-learn v1.5.1* library [38]. The naming of hyperparameters in the tables in the following text corresponds to the naming of function parameters in *scikit-learn*.

6.1 Support Vector Machine

For SVM, we performed a grid search across the kernel functions, the regularization parameter (C), and the gamma parameter. Specifically, we tested the radial basis function (Equation 28) and polynomial kernels (Equation 29) of different degrees due to their occurrences in previous works.

$$\text{rbf} = \exp\left(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2\right) \tag{28}$$

$$\text{poly} = (\gamma\langle\mathbf{x}, \mathbf{x}'\rangle)^{\text{degree}} \tag{29}$$

In both kernel functions, γ serves as a tunable hyperparameter. Note, that the expression $\gamma =$ ”auto” stands for a

$$\gamma = \frac{1}{n} \tag{30}$$

where n represents the number of features. The combination of the hyperparameters is described in Table 3.

6.2 k-Nearest Neighbors

The KNN model has three tunable hyperparameters, namely the number of neighbors, the order of the Minkowski metric, and the weights of the neighbors. We tested only the first and second order p of this metric, which also represent the Manhattan and Euclidean distances. Additionally, we tested: 1) uniform weights, where all neighbors are given equal weight, and 2) distance-based weights, where the weight of each neighbor is inversely proportional to its distance from the classified sample.

$$d_p = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{\frac{1}{p}} \tag{31}$$

The combination of the hyperparameters is described in Table 4.

6.3 Gaussian Naive Bayes

The way the NB model is implemented, there is only one hyperparameter representing a portion of the maximum feature variance added to the variance of the remaining features to improve the stability of the calculation. Since minor changes would lead to no difference and major changes would worsen the performance significantly, we decided not to tune the hyperparameter and use the default value 10^{-9}.

6.4 Decision Tree

For the DT classifier [48], which is relatively fast in terms of computational time, we performed a grid search across various hyperparameters. To evaluate the quality of splits, we experimented with different criteria, including Gini impurity, log loss, and entropy. Regarding the strategy to choose the node for a split, we tested the "best" strategy, which chooses the best split, and the "random" strategy. The minimum number of samples required to split a node was evaluated across values ranging from 2 to 10. For the number of features considered when looking for the best split, we evaluated both the square root and the binary logarithm of the total number of features. The summarized specification of hyperparameters that were used for a grid search is in Table 5.

6.5 Random Forest

During preliminary testing of RF on several datasets, we concluded that the combination of the Gini criterion and the number of features to consider when looking for the best split equal to the square root of the total number of features would lead to better performance and allow us to decrease the dimensionality of the grid search. We utilized the bootstrapping technique for building the trees, keeping its default settings in *scikit-learn*.

For the grid search, we considered two hyperparameters, namely the minimum number of samples in a node before split and number of trees in the forest.

The values of hyperparameters tested in all possible combinations are in Table 6.

6.6 AdaBoost

We used DTs as weak learners for training the AdaBoost classifier [49]. Hyperparameters utilized in the grid search procedure were the learning rate and the number of estimators. See Table 7 for specific values of those hyperparameters.

7 Validation of Results

It is crucial to establish robust performance metrics for our model to accurately assess its capability in distinguishing between healthy and pathological patients. Equally important is to ensure that the reported validation results are robust, minimizing the influence of randomness to guarantee the reliability and consistency of our conclusions, as well as the possibility to independently reproduce our findings.

7.1 Used Metrics

The imbalance between healthy and pathological samples in the dataset (408 healthy females, 545 females with pathologies, 252 healthy males, and 443 males with pathologies) introduces a bias into commonly used metrics such as accuracy, F1 score, precision, and negative predictive value [50]. Especially accuracy can dangerously show overoptimistic results and provide misleading information [51]. To address this issue and provide metrics that reflect class imbalance, we evaluate the performance of the ML algorithms using sensitivity (Equation 32), specificity (Equation 33), unweighted average recall (Equation 35), geometric mean (Equation 34), and bookmaker informedness (Equation 36). As all these metrics provide information about the successful classification, we also evaluate the MCC (Equation 37), which, although not entirely unbiased, has a smaller bias compared to accuracy and also takes into account the errors [50].

In our research, true positive (TP) predictions are correctly predicted positive (pathological) samples, and true negative (TN) predictions are correctly predicted negative (healthy) samples. False positive (FP) predictions mark positive predictions of negative samples, and false negative (FN) predictions mark negative predictions of positive samples. An optimal classification algorithm should reach zero FP and FN predictions.

Sensitivity, also known as recall or true positive rate, is calculated as a ratio between the TP and all positive samples. It shows the ability of a classifier to classify positive samples.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{32}$$

Specificity, also known as selectivity or true negative rate, is calculated as a ratio between the TN and all negative samples. It demonstrates the ability of a classifier to classify negative samples. In the case of the dataset we utilized in this work, specificity is an important metric as it clearly shows the ability of our model to learn to predict the minority class.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{33}$$

The GM is defined as the geometric mean of sensitivity and specificity (Equation 34)

$$\text{GM} = \sqrt{\text{Sensitivity} \cdot \text{Specificity}} \tag{34}$$

UAR is defined as an arithmetic average of sensitivity and specificity (Equation 35).

$$\text{UAR} = \frac{\text{Sensitivity} + \text{Specificity}}{2} \tag{35}$$

The BM is defined by Equation 36.

$$\text{BM} = \text{Sensitivity} + \text{Specificity} - 1 \tag{36}$$

Note that BM, UAR, and GM give the sensitivity and specificity the same weight, regardless of the distribution of positive and negative samples in the dataset. Unlike to UAR, BM and GM penalize the performance on the minority class more, resulting in 0 values for 0 sensitivity (or specificity).

MCC (Equation 37) summarizes the results using all available information about classification results in the form of TP, TN, FP, and FN.

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{\begin{aligned}&(\text{TP} + \text{FP}) \cdot (\text{TP} + \text{FN}) \cdot \\ &\cdot (\text{TN} + \text{FP}) \cdot (\text{TN} + \text{FN})\end{aligned}}} \tag{37}$$

7.2 Validation Approach

Due to the limited size of the dataset and its imbalance, we employed a stratified 10-fold cross-validation method during the grid search, which should lead to less biased results of classifier performances [52]. The dataset was split into ten folds, each preserving the distribution of healthy and pathological samples. For each iteration, one fold was used as the validation set, while the remaining nine folds were used for training. The training data was augmented using the k-means SMOTE-based algorithm (see Section 5). Then, each feature in the training dataset was scaled to an interval $[0, 1]$ using the min-max scaler. Parameters obtained for training dataset scaling were used to scale the validation dataset, which effectively prevents the data leakage.

Algorithm 2 Results validation

```
for each classifier type do
    load results of all classifiers
    sort results according to MCC score
    take 1000 results with highest MCC score
    for for each classifier with corresponding dataset and hyperparameters from previous step do
        i = 0
        while i < 100 do
            split the dataset randomly to 10 stratified folds
            for each fold do
                use this fold as a validation set and oversample rest of folds with k-means based SMOTE algorithm
                find a scaling parameters to scale each feature in training dataset to interval $[0, 1]$
                scale features in validation dataset using the scaling parameters from previous step
                fit the classifier
                compute performance metrics
            end for
            i = i + 1
        end while
        compute average performance metrics from all repetitions of stratified cross validation
    end for
    select the classifier with corresponding dataset and hyperparameters that has the highest average MCC as the best performing classifier
end for
```

This process yielded ten values for each performance metric. We then computed the mean value of each metric across the folds, following standard cross-validation practice.

The workflow of the data pipeline, from the augmentation to the calculation of the results from the cross-validation, is illustrated in Figure 8.

After completing the grid search, we identified the 1000 best performing classifier configurations for each classifier type based on MCC. To account for the variance in MCC introduced by the cross-validation splits, we performed repeated stratified 10-fold cross-validation with 100 repetitions for these configurations. This allowed us to estimate the average metrics for each classifier model and their corresponding standard deviations more reliably. The validation of the best results is shown in Algorithm 2.

To find the best performing dataset for each type of classifier, we first aggregated the results to gain the average performance as well as the standard deviation for each dataset and classifier combination (averaging out different configurations for the classifier). Next, we sorted the results based on MCC and selected the best performing dataset for each classifier based on the maximum of MCC.

7.3 Ensuring Reproducibility

To maintain reproducibility of our findings, we implemented multiple mechanisms to ensure that all computations provide the same results and that everybody can verify that our results were produced by the code we supply. The main problem we had to mitigate was that the data we used were not released under an explicit license. Therefore we could not share the original SVD data and we had to implement methods, based on using SHA256 checksum, to validate input data and intermediate results. This also caused more complicated setup of the working environment.

Some algorithms used in the experiment were based on randomness, usually derived from the initial random values of trainable parameters. To mitigate these problems, we explicitly initialized PRNGs with a seed (value 42 is used). Some differences between operating systems and versions of utilized software also hindered our efforts to provide a fully reproducible code. An example of such differences is the order of files listed in a directory. We tried to treat these problems, however, some edge cases may remain unnoticed and can emerge. We therefore report exact versions of all software and computing hardware and strict adherence to our setup is highly recommended.

Most computations were done in floating-point arithmetic, which introduced rounding errors. Moreover, to increase computational speed, software such as compilers does not always fully adhere to exact specifications and reference implementations. This leads to different results of the same computations made on different hardware or software and these errors usually multiply making differences in results larger. Mitigating these errors is extremely impracticable, neighboring with impossibility. To ensure a way for other researchers to reproduce our results, we rounded some intermediate results where these errors happened, as well as increased floating point precision to minimize these errors.

8 Results & Discussion

All calculations were implemented in Python 3.12 [53]. The code is available at https://github.com/aailab-uct/

Automated-Robust-and-Reproducible-Voice-Pathology-Detection. We used classifiers implemented in the *scikit-learn v1.5.1* [38]. The ML pipeline and k-means SMOTE-based algorithm are implemented via the *imbalance-learn v0.12.3* library [54]. The computations were done on a desktop computer with an AMD Ryzen 9 5900X 12 cores CPU running at 3701 MHz with 128 GB RAM, Samsung SSD 870 QVO 2TB hard drive, and GNU/Linux OS Ubuntu 22.04 LTS. The libraries used for feature extraction, data augmentation, and model training are listed in the *requirements.txt* file included in the aforementioned repository.

As we mentioned and explained in Section 7, we decided against presenting the accuracy score as the data used for training and evaluation of the dataset was moderately imbalanced. Instead, we provide an alternative in the form of MCC, UAR, GM, and BM. However, if needed, accuracy can still be found in raw results reported in the repository.

In Table 8, you can see the results for the best performing classifiers for each sex and classifier type, the configurations of the datasets that we used to reach the best results for the classifier are in Table 9 for females and Table 10 for males. Note that we did not optimize any parameters of the NB classifier, therefore, the standard deviation is kept to zero. Moreover, the results were obtained via 100 repeats of stratified 10-fold cross-validation.

In Table 11 and Table 12, you can see the best performing dataset for each sex and classifier type, demonstrating the prominence of individual features. We select the best performing dataset using average MCC, which is, for a given dataset, computed across all hyperparameter settings for each classifier.

Based on the MCC score, the best performing male classifier is SVM, reaching the MCC of 0.6668, the second being an AdaBoost model with the MCC of 0.6290. For the female dataset, the best performing model is AdaBoost with the MCC of 0.7099, while the second best is SVM with the MCC of 0.7050. It is worth mentioning that the RF model performed in classification of female dataset just slightly worse than the SVM model with the MCC score of 0.7046. Based on the results, the female classifiers perform better in general, which might be partially affected by the higher sample size and better ratio between the pathological and healthy samples. On the contrary, the male best performing classifier also performed best in other metrics balancing the sensitivity and specificity, i.e. GM, BM, and UAR, while the top three female classifiers outperformed the top male classifier in MCC.

Feature-wise, several trends are appearing in the results. Regarding the best classifiers, our proposed feature, the pitch difference, appears in the top performing classifiers, being more prominent in the female datasets. Specifically, it is utilized by the best performing male SVM and female AdaBoost and SVM classifiers. Regarding the cepstral features, it is inconclusive whether datasets with 20 MFCC outperform the datasets with 13 MFCC as both are prominent among the top performing classifiers. From the spectral features, both spectral flatness and spectral contrast look promising as they are present in the top performing classifiers for both sexes. On the other hand, the formant values were included only in the top performing NB classifiers so they appear not to contribute to successful classification. Additionally, ZCR appears to be ineffective as it is only present in the best performing datasets for the KNN and NB models which do not perform well in voice pathology detection.

Regarding the top performing datasets, the aforementioned spectral features were confirmed to be valuable as they also appear in the best performing datasets for each classifier type. The results also support the inconclusiveness of the number of MFCC. The variances of the MFCC and their derivatives do not seem to be useful in general as they appear only in a male AdaBoost classifier. On the other hand, ZCR and formants appear to be somewhat effective as they appear in several best male datasets for different classifiers.

It is important to note that the comparison of dataset results across all hyperparameter configurations was conducted without repeated cross-validation, therefore, the performance should not be compared between different classifiers and to the best performing datasets as the results might be influenced by the split to individual folds.

In addition to that the results we managed to reach with our feature extraction method are on par with or surpass the reported performance in Section 2, we also believe they are, in fact, valid, important, and fully reproducible. Especially regarding reproducibility, to our best knowledge, this is the first paper on voice pathology detection combining SVD and ML methods, that is fully reproducible and conforms to the REFORMS practices [55]. We provide the filled REFORMS checklist in Appendix A.

First, the majority of the reported works excluded data based on individual pathologies. While this may improve the classifier's performance and even allow multimodal classification of individual pathologies, we strongly believe this approach actually limits the classifier's applicability. In clinical practice, these classifiers could not be utilized unless the excluded pathologies were also excluded from the possible diagnoses for the examined patients, which may prove impractical for clinical use.

Second, none of the studies we investigated reported handling the potential data leakage stemming from training classifiers on datasets containing multiple recordings from the same patients. As it is common knowledge that ML models are able to learn patterns of individual patients, given enough input, we assume all results of the presented works might be overestimated due to this error. Therefore, we applied a method to exclude data with duplicity based on patient identity. Our proposed approach does not discriminate against either healthy or any pathologies and does not lead to overoptimistic results.

All features, except the two indicated in this paragraph, are obtained from voice recordings and are widely used in the models for voice pathology detection (see subsection Section 2). We consider the "AGE" feature legitimate, as other acoustic features depend on the age of the speaker. I.e. there are changes in speaking fundamental frequency with aging [56]. The feature, introduced as "NaN", reflects the fact that it was not possible to estimate the fundamental frequency for the patient (as described in Subsection 4.4). As the fundamental frequency is considered one of the dominant features in voice pathology diagnosis, we regard this "NaN" feature as a legitimate approach. Note, that in total, there are 2 females and 6 males in our dataset whose fundamental frequency value reaches NaN, and all of them suffer from a voice disorder.

However, we are aware of several limitations our work is subject to. First, our model was tested using SVD only. The used database does not fully reflect the general population, especially in the proportion of healthy/pathological voices. However, at this time, there is no other suitable database that would reflect the general population better. Despite the justification as the only viable source of data, we cannot extrapolate its performance outside of this dataset. As the data was recorded in a controlled environment, we can assume our classifiers might not be able to perform as well with datasets that are recorded during different conditions. Moreover, we limit our research only to individuals who are 18 years old and older. Another noteworthy limitation was the available computational capacity, which led to careful decision of the classification algorithms, hyperparameter space and selected features we drew on throughout our work.

9 Conclusion

In this work, we introduced several important topics related to voice pathology detection. We summarized the current state-of-the-art achievements of voice pathology detection and classification using the SVD in combination with various feature extraction techniques and ML classifiers. We explained the problematics of potential overoptimistic performance reports from voice pathology classification and the problems with reproducibility. We also addressed the potential problems with data leakage introduced by the SVD if multiple recordings from a single patient are not addressed properly.

Moreover, we proposed a combination of data augmentation based on k-means SMOTE and feature extraction with the addition of a pitch difference as a new feature. We also established a way of validating a classifier performance in the field of voice pathology detection which would possibly prevent the reporting of overestimated results.

While our sex-based models reached a MCC of 0.7099 for females (AdaBoost) and 0.6668 for males (SVM), we also examined and experimentally evaluated the contribution of various feature subsets, that are commonly used in voice pathology detection studies using ML classifiers.

During our research, we identified, as many researchers before us, the lack of data for voice pathology detection as a significant limitation in current research and a major obstacle to the use of ML-based models in clinical settings. Without more comprehensive databases that include a broad range of pathologies and are recorded with at least the same sampling rate as the SVD, it is difficult to establish the external validity of these models.

Acknowledgements. T.J. and J.S. were supported by their specific university grant IGA A1_FCHI_2024_002.

Declarations

Data availability

"Saarbruecken Voice Database" is available at: https://stimmdb.coli.uni-saarland.de/help_en.php4. The repository does not provide versioning of the dataset.

Code availability

We provide a publicly available GitHub repository https://github.com/aailab-uct/Automated-Robust-and-Reproducible-Voice-Pathology-Detection with DOI 10.5281/zenodo.13771573.

Author contribution

Conceptualization: JV, JS, LV Methodology: JV, JS, LV, NH Software: JV, JS, JT, LV, RDF Validation: JT, YZ, KI, LH, ZS Investigation: JV, JS Writing: all authors Supervision: JV, NH, JM Funding acquisition: JS, TJ, JM

Ethics approval

Not applicable.

Competing interests

None to declare.

Appendix A Reforms checklist

Module 1: Study design

1a. State the population or distribution about which the scientific claim is made

The population for our scientific claim consists of German females aged 18 to 94 and German males aged 18 to 89. These individuals recorded their voices pronouncing the /a:/ vowel at a normal pitch at the Institute für Phonetik, Universität des Saarlandes (see Table 1, Figure 3, and Figure 4).

1b. Describe the motivation for choosing this population or distribution (1a).

In our research, we examine the feasibility of ML methods for voice pathology detection in adult patients. The SVD is the only available suitable dataset, having relatively large number of samples, while containing wide number of diseases, which are also presented in general population (see Section 2 for reasoning and Section 3 for description of the data).

1c. Describe the motivation for the use of ML methods in the study.

We aim to build models exploiting the large number of feature for automatic voice pathology detection system that maximize the Matthews correlation coefficient.

In our research we do not investigate the relationship between feature causality or correlation with the pathology itself (see Section 1).

Module 2: Computational reproducibility

2a. Describe the dataset used for training and evaluating the model and provide a link or DOI to uniquely identify the dataset.

We utilize publicly available dataset "Saarbruecken Voice Database" available at: https://stimmdb.coli.uni-saarland.de/help_en.php4. There is no unique identifier and no version control of the database, and we do not have a license to share the dataset, therefore, we provide list of files along with sha256 checksum to ensure reproducibility (see Section 3).

2b. Provide details about the code used to train and evaluate the model and produce the results reported in the paper along with link or DOI to uniquely identify the version of the code used.

The code used to produce all results, along with all supplemental material and information to reproduce our results, is stored in publicly available GitHub repository https://github.com/aailab-uct/Automated-Robust-and-Reproducible-Voice-Pathology-Detection with following DOI: 10.5281/zenodo.13771573.

2c. Describe the computing infrastructure used.

All of the code produces same results regardless of architecture and operating system of the computer. The results were obtained with Ryzen 9 5900X 12 cores CPU running at 3701 MHz with 128 GB RAM, Samsung SSD 870 QVO 2TB hard drive and with GNU/Linux OS Ubuntu 22.04 LTS (see Section 8).

2d. Provide a README file which contains instructions for generating the results using the provided dataset and code.

See the GitHub repository.

2e. Provide a reproduction script to produce all results reported in the paper.

See the GitHub repository.

Module 3: Data quality

3a. Describe source(s) of data, separately for the training and evaluation datasets (if applicable), along with the time when the dataset(s) are collected, the source and process of ground-truth annotations, and other data documentation.

All data are from Saarbruecken Voice Database as described in 2a. As this database is relatively small, we utilize only stratified 10-fold cross-validation without test or evaluation dataset. We downloaded data from SVD on July 12, 2022. The recordings used in our study were recorded between November 20, 1997 and June 16, 2004. The ground truth annotations were obtained by evaluation of stroboscopical recording by database authors [4]. All information related to SVD can be found at https://stimmdb.coli.uni-saarland.de/help_en.php4.

3b. State the distribution or set from which the dataset is sampled (i.e., the sampling frame).

As we only adapt the database, there is no information on the methodology of selection for the recording.

3c. Justify why the dataset is useful for the modeling task at hand.

We believe SVD dataset is relevant as it contains various pathologies (71 different pathologies) that are also present in general population. In our study, after removing underaged patients and duplicities, we worked with 66 various pathologies (see Section 3).

3d. State the outcome variable of the model, along with descriptive statistics (split by class for a categorical outcome variable) and its definition.

The outcome variable in this study is the health status of patients, classified into two categories: 'Healthy' and 'Pathological'. This binary classification is based on information provided by the database, specifically the information about pathologies.

The outcome variable is derived from the table containing patient information. We create a new variable called pathology binary, which takes the value of 0 if there are no values in the pathology column in the svd_information.csv file, and the value of 1 if one or more pathologies are listed in the column (see Section 3). The distribution of healthy/pathological recordings of female and male subjects is provided in the Table 1.

3e. State the sample size and outcome frequencies.

In our study, we utilized 1658 recordings out of 2041 total. In total, 998 are pathological and 660 are healthy. More detailed distribution is explained in Table 1, Figure 3, Figure 4, and Figure 5 (see Section 3).

3f. State the percentage of missing data, split by class for a categorical outcome variable.

There was no missing data. However, 1 file (1573-a-n.wav) contained recording of time between two sessions and one was corrupted (87-a-n.wav) (see Section 3). These recordings were excluded from our research.

3g. Justify why the distribution or set from which the dataset is drawn (3b.) is representative of the one about which the scientific claim is being made (1a.).

The used database does not fully reflect the general population, in the sense of proportion of healthy/pathological voices. However, at this time, there is no other suitable database that would reflect the general population better (see Section 9).

Module 4: Data preprocessing

4a. Describe whether any samples are excluded with a rationale for why they are excluded.

From the dataset, we remove 2 corrupted recordings, 40 under age recordings and 341 recordings of patients with multiple recording sessions (except first healthy and first pathological recordings, see Figure 5). See Section 3 for rationale and GitHub repository for list of excluded files (misc/list_of_excluded_files.csv).

4b. Describe how impossible or corrupt samples are dealt with.

When the extraction of f_0 was impossible, the feature values for $\overline{f}_0$, σ_{f_0}, Δf_0, jitta, and shim were set to 0 and the binary NaN feature was set to 1 (see Section 4).

4c. Describe all transformations of the dataset from its raw form (3a.) to the form used in the model, for instance, treatment of missing data and normalization—preferably through a flow chart.

See Figure 5. More details are in sections 3, 4, and 5.

Module 5: Modeling

5a. Describe, in detail, all models trained.

We utilize multiple ML algorithms for classification (see Section 6) and k-means SMOTE algorithm for dataset augmentation (see Section 5).

5b. Justify the choice of model types implemented.

All ML algorithms are suitable for multi-dimensional data, that we are dealing with (see Section 6).

5c. Describe the method for evaluating the model(s) reported in the paper, including details of train-test splits or cross-validation folds.

Information about stratified 10-fold cross-validation and repeated stratified 10-fold cross-validation for the best models is described in the section Section 7.

5d. Describe the method for selecting the model(s) reported in the paper.

We performed repeated 10-fold cross-validation to estimate the average value of MCC and its corresponding standard deviation. We select the best model according to this MCC. See Section 7.

5e. For the model(s) reported in the paper, specify details about the hyperparameter tuning.

Hyperparameter tuning was approached via the grid search method. The range of hyperparameters was decided after preliminary experiments. See Section 6 for more details and tables 3 – 7 for possible hyperparameter values.

5f. Justify that model comparisons are against appropriate baselines.

Our results are comparable to results in Section 2. Regarding reproducibility, we believe to be the first paper combining SVD and ML methods while adhering to the REFORMS checklist. Our research distinguishes from the referred works by not eliminating data based on pathologies by addressing potential data leakage through duplicities, oversampling on full dataset, and improper data scaling. See more explanation in Section 8.

Module 6: Data leakage

6a. Justify that pre-processing (Module 4) and modeling (Module 5) steps only use information from the training dataset (and not the test dataset).

For patients with multiple recordings of the same type (either all healthy or all pathological), we retained only the oldest recorded sample. For patients with both healthy and pathological recordings, we selected the oldest sample of each type, resulting in a maximum of two recordings per patient — one healthy and one pathological. We believe this approach minimizes the likelihood of the classifier learning patient identities, as the patient's state remains independent of their identity. See Section 3.

By applying oversampling algorithm only to the training folds, we aimed to prevent data leakage and ensure that the model's performance evaluation on the test fold remains unbiased. This approach allowed us to mitigate the adverse effects of class imbalance, resulting in a more robust and reliable predictive model. See Sections 5 and 7.

The whole process, from preprocessing, to validation, is described by Figures 5, 7, 8, and 2.

6b. Describe methods used to address dependencies or duplicates between the training and test datasets (e.g. different samples from the same patients are kept in the same dataset partition).

See Section 3 where we describe more precisely how we select only valid recordings (at most one healthy and one pathological) for each subject in reproducible way. To avoid the data leakage, we applied k-means SMOTE-based algorithm on training data and identify the parameters of scaling transformation using the training data only (see Sections 5, 7, Figures 5, 8, and 2).

6c. Justify that each feature or input used in the model is legitimate for the task at hand and does not lead to leakage.

All features,except two, are obtained from voice recordings and are widely used in the models for voice pathology detection (see Section 2). We consider the "AGE" feature legitimate, as other acoustic features depends on the AGE. I.e. there are changes in speaking fundamental frequency with aging [56].

The feature, that we introduced as "NaN" reflects the fact, that it was not possible to estimate the fundamental frequency for the patient. As the fundamental frequency is considered as one of the dominant features in voice pathology diagnosis, we consider introducing this "NaN" feature legitimate approach. Note, that in total, there are 2 females and 6 males in our dataset, that has NaN value of fundamental frequency, all of them are suffering the voice disorder (see Section 4).

Module 7: Metrics and uncertainty

7a. State all metrics used to assess and compare model performance (e.g., accuracy, AUROC etc.). Justify that the metric used to select the final model is suitable for the task.

The choice of metrics, with the respect to the class imbalance in the data, is written in the Section 7. The claim regarding the best model is based on the Matthews correlation coefficient metric, that is suitable for imbalanced datasets and reflect both successes and errors in the classification.

7b. State uncertainty estimates (e.g., confidence intervals, standard deviations), and give details of how these are calculated.

For each of metrics specified in Section 7, we provide also the respective standard deviations that were obtained during the cross-validation procedure which is specified in this section.

7c. Justify the choice of statistical tests (if used) and a check for the assumptions of the statistical test.

We do not use statistical tests in this study.

Module 8: Generalizability and limitations

8a. Describe evidence of external validity.

As we consider SVD database for the only feasible database for our research, it is hard to describe evidence of external validity. See Section 8.

8b. Describe contexts in which the authors do not expect the study's findings to hold.

First, our model was tested using SVD only. The used database does not fully reflect the general population, in the sense of proportion of healthy/pathological voices. However, at this time, there is no other suitable database that would reflect the general population better. Despite the justification as an only viable source of data, we cannot extrapolate its performance outside of this dataset. Moreover, we limit our research only to individuals that are 18 years old and older. As the data was recorded in a controlled environment, we can assume our models classifiers might not be able to perform as well with datasets that are recorded during different conditions. Another noteworthy limitation was the available computational capacity which led to careful decision of the classification algorithms and hyperparameter space we drew from during our work. See Section 8.

References

[1] Silva, W.J., Lopes, L., Galdino, M.K.C., Almeida, A.A.: Voice acoustic parameters as predictors of depression. Journal of Voice (2021)

[2] Borsky, M., Mehta, D.D., Van Stan, J.H., Gudnason, J.: Modal and nonmodal voice quality classification using acoustic and electroglottographic features. IEEE/ACM transactions on audio, speech, and language processing **25**(12), 2281–2291 (2017)

[3] Lopes, L., Vieira, V., Behlau, M.: Performance of different acoustic measures to discriminate individuals with and without voice disorders. Journal of Voice **36**(4), 487–498 (2022)

[4] Koreman, J., Pützer, M.: A german database of patterns of pathological vocal fold vibration. (1997). http://web.archive.org/web/20061002035011/https://www.coli.uni-saarland.de/publikationen/softcopies/Putzer:1997:GDP.pdf

[5] Eye, M., Infirmary, E.: Voice disorders database, version. 1.03 (cd-rom). Lincoln Park, NJ: Kay Elemetrics Corporation (1994)

[6] Mesallam, T.A., Farahat, M., Malki, K.H., Alsulaiman, M., Ali, Z., Al-nasheri, A., Muhammad, G.: Development of the arabic voice pathology database and its evaluation by using speech features and machine learning algorithms. Journal of Healthcare Engineering **2017**(1), 8783751 (2017) https://doi.org/10.1155/2017/8783751 https://onlinelibrary.wiley.com/doi/pdf/10.1155/2017/8783751

[7] Cesari, U., De Pietro, G., Marciano, E., Niri, C., Sannino, G., Verde, L.: A new database of healthy and pathological voices. Computers & Electrical Engineering **68**, 310–321 (2018) https://doi.org/10.1016/j.compeleceng.2018.04.008

[8] Verde, L., Sannino, G.: VOICED Database. PhysioNet (2022). https://doi.org/10.13026/C25Q2N . https://physionet.org/content/voiced/1.0.0/

[9] 2018 IEEE International Conference on Big Data. https://cci.drexel.edu/bigdata/bigdata2018/index.html Accessed 2024-07-18

[10] Harar, P., Galaz, Z., Alonso-Hernandez, J.B., Mekyska, J., Burget, R., Smekal, Z.: Towards robust voice pathology detection: Investigation of supervised deep learning, gradient boosting, and anomaly detection approaches across four databases. Neural Computing and Applications **32**, 15747–15757 (2020)

[11] Gupta, R., Madill, C., Gunjawate, D.R., Nguyen, D.D., Jin, C.T.: Addressing data scarcity in voice disorder detection with self-supervised models. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 11866–11870 (2024). https://doi.org/10.1109/ICASSP48485.2024.10446075 . IEEE

[12] Verde, L., Brancati, N., De Pietro, G., Frucci, M., Sannino, G.: A deep learning approach for voice disorder detection for smart connected living environments. ACM Transactions on Internet Technology (TOIT) **22**(1), 1–16 (2021) https://doi.org/10.1145/3433993

[13] Kotarba, K., Kotarba, M.: Voice pathology assessment using x-vectors approach. Vibrations in Physical Systems **32**(1) (2021) https://doi.org/10.21008/j.0860-6897.2021.1.08

[14] Harar, P., Alonso-Hernandezy, J.B., Mekyska, J., Galaz, Z., Burget, R., Smekal, Z.: Voice pathology detection using deep learning: a preliminary study. In: 2017 International Conference and Workshop on Bioinspired Intelligence (IWOBI), pp. 1–4 (2017). https://doi.org/10.1109/IWOBI.2017.7985525 . IEEE

[15] Park, D., Yu, Y., Katabi, D., Kim, H.K.: Adversarial continual learning to transfer self-supervised speech representations for voice pathology detection. IEEE Signal Processing Letters (2023) https://doi.org/10.1109/LSP.2023.3298532

[16] Omeroglu, A.N., Mohammed, H.M., Oral, E.A.: Multi-modal voice pathology detection architecture based on deep and handcrafted feature fusion. Engineering Science and Technology, an International Journal **36**, 101148 (2022) https://doi.org/10.1016/j.jestch.2022.101148

[17] Verde, L., De Pietro, G., Alrashoud, M., Ghoneim, A., Al-Mutib, K.N., Sannino, G.: Leveraging artificial intelligence to improve voice disorder identification through the use of a reliable mobile app. IEEE Access **7**, 124048–124054 (2019)

https://doi.org/10.1109/ACCESS.2019.2938265

[18] Verde, L., De Pietro, G., Sannino, G.: Voice disorder identification by using machine learning techniques. IEEE Access **6**, 16246–16255 (2018) https://doi.org/10.1109/ACCESS.2018.2816338

[19] Tirronen, S., Kadiri, S.R., Alku, P.: Hierarchical multi-class classification of voice disorders using self-supervised models and glottal features. IEEE Open Journal of Signal Processing **4**, 80–88 (2023)

[20] Yagnavajjula, M.K., Mittapalle, K.R., Alku, P., Mitra, P., *et al.*: Automatic classification of neurological voice disorders using wavelet scattering features. Speech Communication **157**, 103040 (2024)

[21] Junior, S.B., Guido, R.C., Aguiar, G.J., Santana, E.J., Junior, M.L.P., Patil, H.A.: Multiple voice disorders in the same individual: investigating handcrafted features, multi-label classification algorithms, and base-learners. Speech Communication **152**, 102952 (2023)

[22] Fan, Z., Wu, Y., Zhou, C., Zhang, X., Tao, Z.: Class-imbalanced voice pathology detection and classification using fuzzy cluster oversampling method. Applied Sciences **11**(8), 3450 (2021)

[23] Ding, H., Gu, Z., Dai, P., Zhou, Z., Wang, L., Wu, X.: Deep connected attention (dca) resnet for robust voice pathology detection and classification. Biomedical Signal Processing and Control **70**, 102973 (2021)

[24] Guedes, V., Teixeira, F., Oliveira, A., Fernandes, J., Silva, L., Junior, A., Teixeira, J.P.: Transfer learning with audioset to voice pathologies identification in continuous speech. Procedia Computer Science **164**, 662–669 (2019)

[25] Hemmerling, D.: Voice pathology distinction using autoassociative neural networks. In: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 1844–1847 (2017). IEEE

[26] AL-Dhief, F.T., Latiff, N.M.A., Malik, N.N.N.A., Baki, M.M., Sabri, N., Albadr, M.A.A., Sazihan, N.F.S.M.: Voice pathology detection using decision tree classifier. In: 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), pp. 36–41 (2023). IEEE

[27] AnilKumar, V., Reddy, R.V.S.: Classification of voice pathology using different features and bi-lstm. In: 2023 International Conference on Smart Systems for Applications in Electrical Sciences (ICSSES), pp. 1–4 (2023). IEEE

[28] Tirronen, S., Kadiri, S.R., Alku, P.: The effect of the mfcc frame length in automatic voice pathology detection. Journal of Voice (2022)

[29] Dibazar, A.A., Berger, T.W., Narayanan, S.S.: Pathological voice assessment. In: 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1669–1673 (2006). IEEE

[30] Ricci Maccarini, A., Lucchini, E.: La valutazione soggettiva ed oggettiva della disfonia: il protocollo sifel. In: Presented at the Relazione Ufficiale al XXXVI Congresso Nazionale della Società Italiana di Foniatria e Logopedia (2002)

[31] Berger, T., Peschel, T., Vogel, M., Pietzner, D., Poulain, T., Jurkutat, A., Meuret, S., Engel, C., Kiess, W., Fuchs, M.: Speaking voice in children and adolescents: Normative data and associations with bmi, tanner stage, and singing activity. Journal of Voice **33**(4), 580–2158030 (2019)

[32] Hollien, H.: On pubescent voice change in males. Journal of Voice **26**(2), 29–40 (2012)

[33] McFee, B., McVicar, M., Faronbi, D., Roman, I., Gover, M., Balke, S., Seyfarth, S., Malek, A., Raffel, C., Lostanlen, V., Niekirk, B., Lee, D., Cwitkowitz, F., Zalkow, F., Nieto, O., Ellis, D., Mason, J., Lee, K., Steers, B., Halvachs, E., Thomé, C., Robert-Stöter, F., Bittner, R., Wei, Z., Weiss, A., Battenberg, E., Choi, K., Yamamoto, R., Carr, C., Metsai, A., Sullivan, S., Friesch, P., Krishnakumar, A., Hidaka, S., Kowalik, S., Keller, F., Mazur, D., Chabot-Leclerc, A., Hawthorne, C., Ramaprasad, C., Keum, M., Gomez, J., Monroe, W., Morozov, V.A., Eliasi, K., nullmightybofo, Biberstein, P., Sergin, N.D., Hennequin, R., Naktinis, R., beantowel, Kim, T., Åsen, J.P., Lim, J., Malins, A., Hereñú, D., Struijk, S., Nickel, L., Wu, J., Wang, Z., Gates, T., Vollrath, M., Sarroff, A., Xiao-Ming, Porter, A., Kranzler, S., Voodoohop, Gangi, M.D., Jinoz, H., Guerrero, C., Mazhar, A., toddrme2178, Baratz, Z., Kostin, A., Zhuang, X., Lo, C.T., Campr, P., Semeniuc, E., Biswal, M., Moura, S., Brossier, P., Lee, H., Pimenta, W.: librosa/librosa: 0.10.2.post1. Zenodo (2024). https://doi.org/10.5281/zenodo.11192913 . https://doi.org/10.5281/zenodo.11192913

[34] Jadoul, Y., Thompson, B., Boer, B.: Introducing Parselmouth: A Python interface to Praat. Journal of Phonetics **71**, 1–15 (2018)

[35] Harris, C.R., Millman, K.J., Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M.H., Brett, M., Haldane, A., Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E.: Array programming with NumPy. Nature **585**, 357–362 (2020)

[36] Bajaj, N.: Nikeshbajaj/spkit: 0.0.9.4. Zenodo (2022). https://doi.org/10.5281/zenodo.4710694 . https://doi.org/10.5281/zenodo.4710694

[37] Hwang, J., Hira, M., Chen, C., Zhang, X., Ni, Z., Sun, G., Ma, P., Huang, R., Pratap, V., Zhang, Y., Kumar, A., Yu, C.-Y., Zhu, C., Liu, C., Kahn,

J., Ravanelli, M., Sun, P., Watanabe, S., Shi, Y., Tao, Y., Scheibler, R., Cornell, S., Kim, S., Petridis, S.: TorchAudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for PyTorch (2023)

[38] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[39] Boersma, P., Weenink, D.: Praat: doing Phonetics by Computer — fon.hum.uva.nl. https://www.fon.hum.uva.nl/praat/. [Accessed 03-07-2024] (2024)

[40] Boersma, P., *et al.*: Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In: Proceedings of the Institute of Phonetic Sciences, vol. 17, pp. 97–110 (1993). Amsterdam. https://api.semanticscholar.org/CorpusID:2373348

[41] Barreira, R.R., Ling, L.L.: Kullback–leibler divergence and sample skewness for pathological voice quality assessment. Biomedical Signal Processing and Control **57**, 101697 (2020)

[42] Dubnov, S.: Generalization of spectral flatness measure for non-gaussian linear processes. IEEE Signal Processing Letters **11**(8), 698–701 (2004)

[43] Jayant, N.S., Noll, P.: Digital Coding of Waveforms, Principles and Applications to Speech and Video, p. 688. Prentice-Hall, Englewood Cliffs NJ, USA (1984). N. S. Jayant: Bell Laboratories; ISBN 0-13-211913-7

[44] Kent, R.D., Vorperian, H.K.: Static measurements of vowel formant frequencies and bandwidths: A review. Journal of Communication Disorders **74**, 74–97 (2018) https://doi.org/10.1016/j.jcomdis.2018.05.004

[45] Douzas, G., Bacao, F., Last, F.: Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. Information Sciences **465**, 1–20 (2018) https://doi.org/10.1016/j.ins.2018.06.056

[46] Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)

[47] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)

[48] Breiman, L.: Classification and Regression Trees. Routledge, ??? (2017)

[49] Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning

and an application to boosting. Journal of computer and system sciences **55**(1), 119–139 (1997)

[50] Luque, A., Carrasco, A., Martín, A., Las Heras, A.: The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition **91**, 216–231 (2019)

[51] Chicco, D., Jurman, G.: The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC genomics **21**, 1–13 (2020)

[52] Kohavi, R., *et al.*: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai, vol. 14, pp. 1137–1145 (1995). Montreal, Canada

[53] Van Rossum, G., Drake, F.L.: Python 3 Reference Manual. CreateSpace, Scotts Valley, CA (2009)

[54] Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research **18**(17), 1–5 (2017)

[55] Kapoor, S., Cantrell, E.M., Peng, K., Pham, T.H., Bail, C.A., Gundersen, O.E., Hofman, J.M., Hullman, J., Lones, M.A., Malik, M.M., *et al.*: Reforms: Consensus-based recommendations for machine-learning-based science. Science Advances **10**(18), 3452 (2024)

[56] Nishio, M., Niimi, S.: Changes in speaking fundamental frequency characteristics with aging. Folia phoniatrica et logopaedica **60**(3), 120–127 (2008)

Table 2: Extracted features, their notation, their use in datasets, and library used for extraction

Feature	Symbol	Configuration for dataset creation	Python library
mean f_0 across all window of the signal	$\overline{f_0}$	used in all datasets	parselmouth [34]
harmonic-to-noise ratio	$\overline{HNR}$	used in all datasets	parselmouth [34]
jitter	$jitta$	used in all datasets	parselmouth [34]
shimmer	$shim$	used in all datasets	parselmouth [34]
occurrence of NaN values in f_0-related features	NaN	used in all datasets	–
age	age	used in all datasets	–
standard deviation of f_0	$\sigma_{\mathbf{f}_0}$	used / not used	parselmouth [34], numpy [35]
difference between the highest and lowest f_0	Δf_0	used / not used	parselmouth [34], numpy [35]
Shannon entropy	H	used / not used	spkit [36]
mean values of the first 20 LFCC	$\overline{\mathbf{LFCC}}$	used / not used	torchaudio [37]
mean values of the first three formants	$\bar{\mathbf{f}}$	used / not used	parselmouth [34]
skewness of the sound spectra	$skew$	used / not used	scikit-learn [38]
mean spectral centroid	$\overline{S}$	used / not used	librosa [33], numpy [35]
mean spectral contrast	$\overline{\mathbf{SC}}$	used / not used	librosa [33], numpy [35]
mean spectral flatness	$\overline{SF}$	used / not used	librosa [33], numpy [35]
mean spectral roll-off	$\overline{RO}$	used / not used	librosa [33], numpy [35]
mean zero-crossing rate	$\overline{ZCR}$	used / not used	librosa [33], numpy [35]
mean values of the selected MFCC	$\overline{\mathbf{MFCC}}$	used 13 / used 20	librosa [33], numpy [35]
mean first derivative of the selected MFCC	$\overline{\Delta\mathbf{MFCC}}$		librosa [33], numpy [35]
mean second derivative of the selected MFCC	$\overline{\Delta^2\mathbf{MFCC}}$		librosa [33], numpy [35]
variance of selected MFCC	$\sigma^{\mathbf{2}}_{\mathbf{MFCC}}$	used / not used	librosa [33], numpy [35]
variance of first derivative of the selected MFCC	$\sigma^{\mathbf{2}}_{\mathbf{\Delta MFCC}}$		librosa [33], numpy [35]
variance of second derivative of the selected MFCC	$\sigma^{\mathbf{2}}_{\mathbf{\Delta^2 MFCC}}$		librosa [33], numpy [35]

Table 3: SVM hyperparameters and their values tested for optimization

Hyperparameter	Tested values
kernel	"rbf", "poly"
degree ("poly" only)	2, 3, 4, 5, 6
gamma	0.5, 0.1, 0.05, 0.01, 0.005, 0.001, "auto"
C	0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000, 3000, 5000, 7000, 10000, 12000

Table 4: KNN hyperparameters and their values tested for optimization

Hyperparameter	Tested values
n_neighbors	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23
p	1, 2
weights	"uniform", "distance"

Table 5: DT hyperparameters and their values tested for optimization

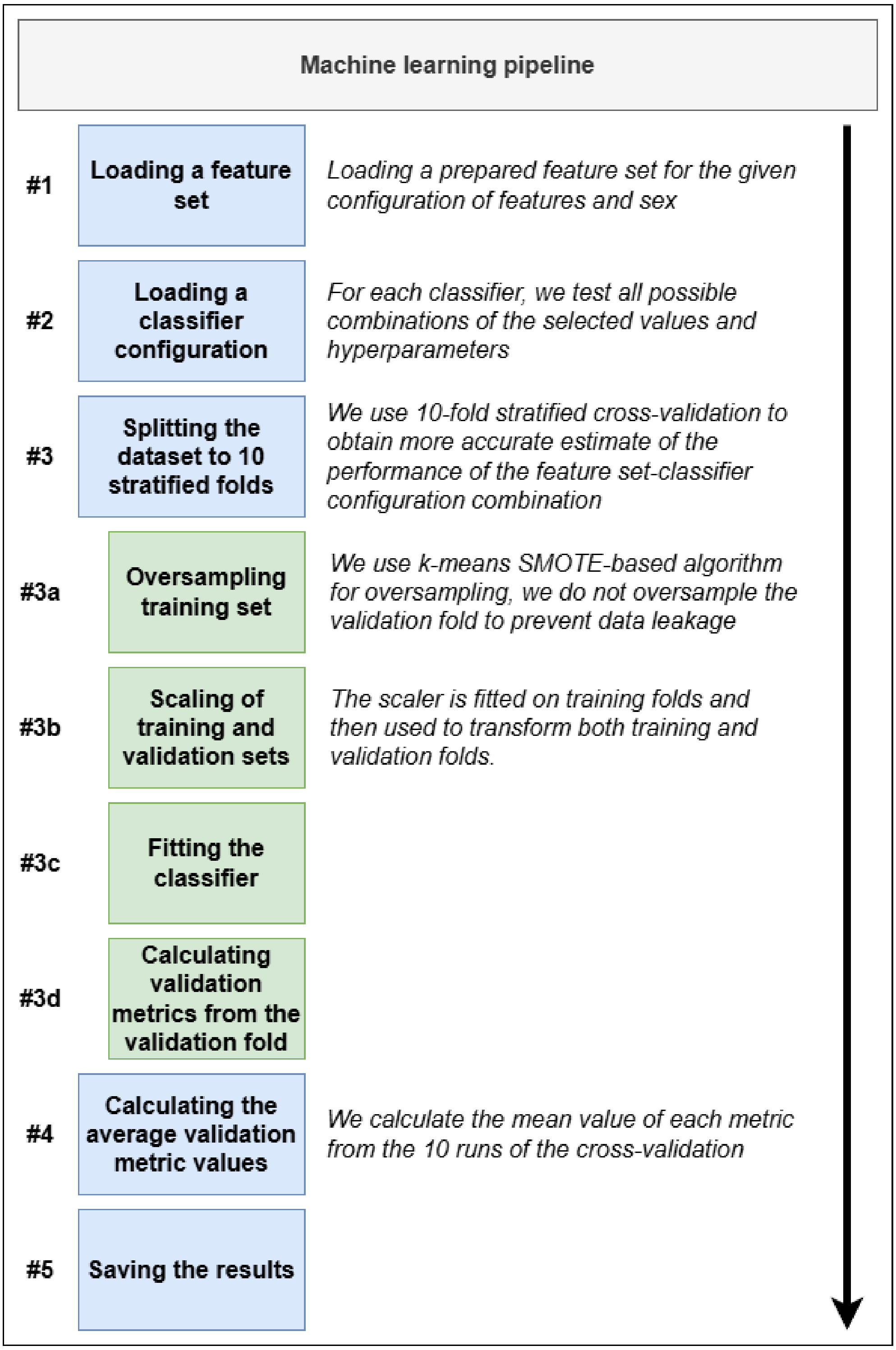
Hyperparameter	Tested values
criterion	"gini", "entropy", "log_loss"
splitter	"best", "random"
min_samples_split	2, 3, 4, 5, 6, 7, 8, 9, 10
max_features	"sqrt", "log_loss"

Table 6: RF hyperparameters and their values tested for optimization

Hyperparameter	Tested values
min_samples_split	2, 3, 4, 5, 6
n_estimators	50, 75, 100, 125, 150, 175

Table 7: AdaBoost hyperparameters and their values tested for optimization

Hyperparameter	Tested values
learning_rate	0.1, 1, 10
n_estimators	50, 100, 150, 200, 250, 300, 350, 400



Machine learning pipeline

#1 Loading a feature set — *Loading a prepared feature set for the given configuration of features and sex*

#2 Loading a classifier configuration — *For each classifier, we test all possible combinations of the selected values and hyperparameters*

#3 Splitting the dataset to 10 stratified folds — *We use 10-fold stratified cross-validation to obtain more accurate estimate of the performance of the feature set-classifier configuration combination*

#3a Oversampling training set — *We use k-means SMOTE-based algorithm for oversampling, we do not oversample the validation fold to prevent data leakage*

#3b Scaling of training and validation sets — *The scaler is fitted on training folds and then used to transform both training and validation folds.*

#3c Fitting the classifier

#3d Calculating validation metrics from the validation fold

#4 Calculating the average validation metric values — *We calculate the mean value of each metric from the 10 runs of the cross-validation*

#5 Saving the results

Fig. 8: Machine learning pipeline for oversampling, classifier fitting, and validation

Table 8: Best results reached for each classifier type

Sex	Classifier	MCC		SEN		SPE		GM		UAR		BM	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
F	SVM	0.7050	0.0692	0.8446	0.0464	**0.8642**	0.0537	**0.8535**	0.0350	**0.8544**	0.0349	**0.7087**	0.0697
	AdaBoost	**0.7099**	0.0688	0.9009	0.0388	0.8001	0.0596	0.8482	0.0364	0.8505	0.0352	0.7010	0.0703
	RF	0.7046	0.0682	**0.9016**	0.0387	0.7930	0.0608	0.8447	0.0365	0.8473	0.0350	0.6946	0.0700
	KNN	0.6066	0.0717	0.7337	0.0578	0.8774	0.0503	0.8013	0.0377	0.8055	0.0366	0.6110	0.0732
	DT	0.5796	0.0817	0.7962	0.0555	0.7850	0.0658	0.7893	0.0416	0.7906	0.0411	0.5812	0.0823
	NB	0.5000	0.0832	0.6833	0.0632	0.8196	0.0614	0.7469	0.0431	0.7515	0.0422	0.5029	0.0845
M	SVM	**0.6668**	0.0841	**0.8247**	0.0557	0.8574	0.0700	**0.8396**	0.0430	**0.8410**	0.0428	**0.6821**	0.0855
	AdaBoost	0.6290	0.0879	0.7834	0.0617	0.8644	0.0703	0.8215	0.0453	0.8239	0.0450	0.6478	0.0901
	RF	0.5826	0.0943	0.8237	0.0574	0.7638	0.0879	0.7911	0.0503	0.7938	0.0485	0.5875	0.0971
	KNN	0.5336	0.0840	0.6706	0.0731	**0.8802**	0.0625	0.7664	0.0467	0.7754	0.0441	0.5508	0.0882
	DT	0.4452	0.1126	0.7704	0.0669	0.6786	0.0985	0.7202	0.0599	0.7245	0.0573	0.4490	0.1146
	NB	0.4464	0.0938	0.6503	0.0698	0.8106	0.0808	0.7237	0.0487	0.7305	0.0485	0.4609	0.0970

Table 9: Configuration of the best model and dataset for each classifier type - females

Classifier	AdaBoost	SVM	RF	KNN	DT	NB
			’criterion’: ’gini’		’criterion’: ’gini’	
		’C’: 10	’max_features’: ’sqrt’	’n_neighbors’: 17	’max_features’: ’log2’	
	’learning_rate’: 0.1	’gamma’: 0.05	’min_samples_split’: 6	’p’: 2	’min_samples_split’: 10	
	’n_estimators’: 50	’kernel’: ’rbf’	’n_estimators’: 175	’weights’: ’uniform’	’splitter’: ’random’	’var_smoothing’: 1e-09
Dataset name	3166	3962	8188	5855	7355	8146
$\overline{f}_0$	Y	Y	Y	Y	Y	Y
$\overline{HNR}$	Y	Y	Y	Y	Y	Y
$jitta$	Y	Y	Y	Y	Y	Y
$shim$	Y	Y	Y	Y	Y	Y
NaN	Y	Y	Y	Y	Y	Y
age	Y	Y	Y	Y	Y	Y
σ_{f_0}	N	N	N	Y	N	N
Δf_0	Y	Y	N	N	N	N
H	Y	Y	Y	N	N	Y
$\overline{\mathbf{LFCC}}$	N	Y	N	N	Y	Y
$\overline{\mathbf{f}}$	N	N	N	N	N	Y
$skew$	N	N	Y	N	N	N
$\overline{S}$	N	N	N	N	N	N
$\overline{SC}$	Y	N	N	N	Y	N
$\overline{SF}$	Y	N	N	Y	Y	N
$\overline{RO}$	Y	Y	N	N	N	N
$\overline{ZCR}$	N	N	N	N	Y	N
$\overline{\mathbf{MFCC}}$	13	20	20	13	20	13
$\overline{\Delta\mathbf{MFCC}}$	13	20	20	13	20	13
$\overline{\Delta^2\mathbf{MFCC}}$	13	20	20	13	20	13
$\sigma^{\mathbf{2}}_{\mathbf{MFCC}}$	N	N	N	N	N	N
$\sigma^{\mathbf{2}}_{\mathbf{\Delta MFCC}}$	N	N	N	N	N	N
$\sigma^{\mathbf{2}}_{\mathbf{\Delta^2 MFCC}}$	N	N	N	N	N	N

Table 10: Configuration of the best model and dataset for each classifier type - men

Classifier	SVM 'C': 3000 'degree': 3 'gamma': 0.01 'kernel': 'poly'	AdaBoost 'learning_rate': 0.1 'n_estimators': 350	RF 'criterion': 'gini' 'max_features': 'sqrt' 'min_samples_split': 2 'n_estimators': 100	KNN 'n_neighbors': 15 'p': 2 'weights': 'distance'	NB 'var_smoothing': 1e-09	DT 'criterion': 'log_loss' 'max_features': 'sqrt' 'min_samples_split': 9 'splitter': 'best'
Dataset name	221	4458	5979	6111	7639	1150
$\overline{f}_0$	Y	Y	Y	Y	Y	Y
$\overline{HNR}$	Y	Y	Y	Y	Y	Y
$jitta$	Y	Y	Y	Y	Y	Y
$shim$	Y	Y	Y	Y	Y	Y
NaN	Y	Y	Y	Y	Y	Y
age	Y	Y	Y	Y	Y	Y
σ_{f_0}	Y	Y	Y	Y	N	Y
Δf_0	Y	N	N	N	N	Y
H	N	Y	N	N	N	Y
$\overline{\mathbf{LFCC}}$	N	Y	Y	N	N	N
$\overline{\mathbf{f}}$	N	N	N	N	Y	N
$skew$	Y	N	N	N	N	N
$\overline{S}$	Y	Y	N	N	N	N
$\overline{SC}$	Y	Y	N	N	Y	Y
$\overline{SF}$	Y	N	N	N	N	Y
$\overline{RO}$	N	Y	Y	N	N	Y
$\overline{ZCR}$	N	N	N	N	N	N
$\overline{\mathbf{MFCC}}$	13	20	13	13	13	20
$\overline{\Delta\mathbf{MFCC}}$	13	20	13	13	13	20
$\overline{\Delta^2\mathbf{MFCC}}$	13	20	13	13	13	20
$\sigma^2_{\mathbf{MFCC}}$	N	20	N	N	N	N
$\sigma^2_{\mathbf{\Delta MFCC}}$	N	20	N	N	N	N
$\sigma^2_{\mathbf{\Delta^2 MFCC}}$	N	20	N	N	N	N

Table 11: Best average dataset performance for each classifier - male patients

Classifier		SVM	KNN	NB	DT	RF	AdaBoost
Performance metrics							
MCC	μ	0.5719	0.4989	0.4596	0.4019	0.5941	0.5927
	σ	0.0579	0.0472	0.0000	0.0436	0.0148	0.0468
SEN	μ	0.7486	0.6540	0.6227	0.7373	0.8422	0.7709
	σ	0.1068	0.0338	0.0000	0.0222	0.0057	0.0692
SPE	μ	0.8323	0.8587	0.8489	0.6707	0.7540	0.8359
	σ	0.0748	0.0574	0.0000	0.0356	0.0144	0.0310
UAR	μ	0.7905	0.7563	0.7358	0.7040	0.7981	0.8034
	σ	0.0318	0.0243	0.0000	0.0222	0.0080	0.0208
GM	μ	0.7824	0.7460	0.7242	0.6995	0.7946	0.7993
	σ	0.0404	0.0233	0.0000	0.0231	0.0087	0.0223
BM	μ	0.5809	0.5127	0.4717	0.4080	0.5962	0.6068
	σ	0.0637	0.0486	0.0000	0.0444	0.0160	0.0417
Features used in datasets							
$\overline{f}_0$		Y	Y	Y	Y	Y	Y
$\overline{HNR}$		Y	Y	Y	Y	Y	Y
$jitta$		Y	Y	Y	Y	Y	Y
$shim$		Y	Y	Y	Y	Y	Y
NaN		Y	Y	Y	Y	Y	Y
age		Y	Y	Y	Y	Y	Y
σ_{f_0}		N	Y	Y	Y	Y	N
Δf_0		N	N	N	Y	N	N
H		Y	N	Y	N	N	N
$\overline{\mathbf{LFCC}}$		Y	N	N	N	Y	Y
$\overline{\mathbf{f}}$		Y	N	Y	Y	Y	N
$skew$		N	N	Y	N	Y	Y
$\overline{S}$		N	N	N	Y	N	N
$\overline{SC}$		Y	Y	Y	N	Y	Y
$\overline{SF}$		N	Y	N	Y	N	Y
$\overline{RO}$		N	N	N	Y	Y	N
$\overline{ZCR}$		N	Y	Y	N	Y	N
$\overline{\mathbf{MFCC}}$		20	13	13	13	20	20
$\overline{\Delta\mathbf{MFCC}}$		20	13	13	13	20	20
$\overline{\Delta^2\mathbf{MFCC}}$		20	13	13	13	20	20
$\sigma^{\mathbf{2}}_{\mathbf{MFCC}}$		N	N	N	N	N	20
$\sigma^{\mathbf{2}}_{\mathbf{\Delta MFCC}}$		N	N	N	N	N	20
$\sigma^{\mathbf{2}}_{\mathbf{\Delta^2 MFCC}}$		N	N	N	N	N	20

Table 12: Best average dataset performance for each classifier - female patients

Classifier		SVM	KNN	NB	DT	RF	AdaBoost
Performance metrics							
MCC	μ	0.5756	0.5588	0.5055	0.5380	0.7119	0.7031
	σ	0.1056	0.0600	0.0000	0.0297	0.0080	0.0123
SEN	μ	0.7236	0.7154	0.6828	0.7796	0.9010	0.8932
	σ	0.1639	0.0262	0.0000	0.0182	0.0054	0.0104
SPE	μ	0.8457	0.8470	0.8256	0.7593	0.8021	0.8025
	σ	0.0650	0.0451	0.0000	0.0248	0.0039	0.0042
UAR	μ	0.7846	0.7812	0.7542	0.7695	0.8516	0.8479
	σ	0.0594	0.0304	0.0000	0.0151	0.0038	0.0054
GM	μ	0.7631	0.7770	0.7496	0.7676	0.8492	0.8456
	σ	0.0933	0.0302	0.0000	0.0155	0.0038	0.0053
BM	μ	0.5693	0.5624	0.5084	0.5390	0.7031	0.6958
	σ	0.1188	0.0608	0.0000	0.0303	0.0076	0.0109
Features used in datasets							
$\overline{f_0}$		Y	Y	Y	Y	Y	Y
$\overline{HNR}$		Y	Y	Y	Y	Y	Y
$jitta$		Y	Y	Y	Y	Y	Y
$shim$		Y	Y	Y	Y	Y	Y
NaN		Y	Y	Y	Y	Y	Y
age		Y	Y	Y	Y	Y	Y
σ_{f_0}		N	Y	N	N	Y	Y
Δf_0		Y	N	N	N	Y	Y
$\overline{H}$		Y	N	Y	N	N	N
$\overline{\mathbf{LFCC}}$		Y	N	Y	N	N	N
$\overline{\mathbf{f}}$		N	N	Y	N	N	N
$skew$		N	N	N	N	N	N
$\overline{S}$		Y	N	N	N	N	N
$\overline{SC}$		N	N	N	Y	N	N
$\overline{SF}$		Y	Y	N	N	N	Y
$\overline{RO}$		N	N	N	Y	N	N
$\overline{ZCR}$		Y	N	N	N	N	N
$\overline{\mathbf{MFCC}}$		20	13	13	13	20	13
$\overline{\Delta\mathbf{MFCC}}$		20	13	13	13	20	13
$\overline{\Delta^2\mathbf{MFCC}}$		20	13	13	13	20	13
$\sigma^2_{\mathbf{MFCC}}$		N	N	N	N	N	N
$\sigma^2_{\Delta\mathbf{MFCC}}$		N	N	N	N	N	N
$\sigma^2_{\Delta^2\mathbf{MFCC}}$		N	N	N	N	N	N