POLITECNICO DI TORINO

Master's Degree
in Data Science and Engineering

Master's Degree Thesis

Machine Learning in Renal Failure using voice as biomarker

Supervisor
Prof. Antonio Servetti

Candidate
Valerio Mastrianni

Academic Year 2023-2024

Summary

This thesis presents a novel approach to renal failure detection, proposing the use of voice analysis as a non-invasive biomarker. Renal failure, also known as kidney failure, is a condition affecting around 10% of the global adult population, occurs when the kidneys are unable to efficiently filter waste from the bloodstream, leading to fluid and toxic accumulation and other severe health complications. Current diagnostic methods rely on clinical assessments and laboratory tests, which are often time-consuming, resource-intensive and stressful for the patient. This research explores an alternative, automated detection method by focusing on changes in vocal characteristics, hypothesizing that fluid retention can influence the voice in measurable ways.

The primary goal of the study is to develop a machine learning model capable of detecting changes in patients' voices that correspond to renal failure, particularly in those undergoing dialysis. Dialysis is the process of removing the toxin which the body is not bale to expel by removing the fluid accumulated in the body. By analyzing voice recordings from patients before and after dialysis, the study identifies patterns that correlate vocal changes with fluid removal during the treatment process. The dataset used includes voice recordings as in figure from 86 patients, collected over a period of 90 days. The analysis examines how different machine learning models—such as Support Vector Machines (SVM), Random Forest (RF), and Gradient Boosting (GB)—perform in classifying these changes.

The study reveals several findings:

- **Support Vector Machines Outperform Other Models:** Among the different models tested, SVM demonstrated the highest accuracy in detecting voice changes related to renal failure. This is particularly true for anuric patients, who experience more significant fluid retention.
- **Vocal Characteristics are Linked to Fluid Retention:** The results show a significant correlation between fluid accumulation and changes in voice, supporting the hypothesis that voice can be a reliable biomarker for renal health monitoring. The presence of vocal changes, such as differences in pitch and tone, can signal fluid retention levels in patients.
- **Challenges in Model Generalization:** While the models performed well in detecting voice changes for individual patients, generalizing these results across a broader population remains challenging. This suggests that future research should focus on refining the models to improve their applicability to diverse patient groups.

This research introduces the potential for voice analysis to be used as a cost-effective, non-invasive tool in the clinical detection of renal failure. The use of voice as a biomarker not only reduces the reliance on resource-intensive tests but also provides an accessible means of monitoring patients over time.The study also highlights the challenges in developing models that can generalize well across a broader population. To address this,

further research is needed to improve the robustness and accuracy of the models, particularly when applied to a more diverse set of patients.

This thesis establishes a foundation for using vocal characteristics as a biomarker for renal failure, offering an alternative to traditional diagnostic methods. Using machine learning techniques, this approach could lead to the development of more efficient, patient-friendly tools for monitoring renal failure.

Acknowledgements

This Thesis has been possible thanks with the support of the whole Noah Labs as a company. A special thanks to Stefan and Mikołaj for the great moment spent together, the work and the great amount of fun. A big thanks also to Professor Servetti for the dedication and support during these months. Vorrei anche ringraziare la mia famiglia e Vittoria, senza di loro nulla di tutto ciò sarebbe stato possibile. Un grande ringraziamento anche a tutte le persone incontrate durante il percorso che consapevolmente o meno mi hanno reso una persona, si spera migliore.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Renal failure (also known as kidney failure) is a common medical condition, affecting about 10% of the global adult population. Renal failure is characterized by an individual's decreased ability to filter waste products from the blood, resulting in a build-up of toxins and an inability to effectively manage bodily functions. If not addressed timely can lead to the risk of life-threatening complications, including cardiovascular disease and severe infections. The World Health Organization (WHO) predicted renal failure to be a significant contributor to global disability and mortality by 2030 [2]. Rapidly identify and asses renal failure play a crucial role in an effective treatment. Patient monitoring of renal function are essential for informed treatment choices and evaluating treatment progress.

Even today, the diagnosis of renal failure relies heavily on clinical assessment and laboratory tests. The report by Hong et al. [3] highlights that standardised scales in research and clinical settings rely on objective measures such as glomerular filtration rate (GFR) and creatinine levels to improve the diagnosis and monitoring of renal function. While these measures help to minimise bias, there remains the potential for variability in the interpretation of results during clinical assessment, leading to inconsistencies in diagnosis. This variability has implications for treatment decisions. In addition, the lack of resources and well-trained professionals poses a significant challenge to the effective diagnosis and monitoring of patients with renal failure. There is currently a need for more robust, affordable and automated tools for the clinical detection of renal failure. Recent research suggests significant potential in leveraging advanced technologies, specifically non-invasive biomarkers and imaging techniques for automatically detecting renal failure. Non-invasive options stand out as favorable for incorporation into an automated system, given their cost-effectiveness, ease of use, and reduced patient discomfort. Clinicians frequently rely on various diagnostic indicators such as changes in urine output, electrolyte imbalances, and imaging studies showing kidney damage as indicative signs of renal failure.

This thesis aims to investigate how innovative technologies can be utilized in a reliable manner for the detection of renal failure leveraging on voice as biomarker.

Pathological changes in patients with chronic kidney disease (CKD) were examined and analyzed. CKD is a progressive condition that leads to the gradual loss of kidney function, resulting in the accumulation of waste products and fluid imbalances. The diminished kidney function results in characteristic clinical symptoms such as hypertension, edema, and anemia, while non-clinical symptoms encompass fatigue and cognitive impairments. Pathological analysis of kidney function can provide valuable insights into the presence of renal failure, even when patients may not exhibit overt symptoms. However, accurately characterizing changes and finding insights from biomarkers and imaging studies can be challenging, as it typically requires precise and consistent measurements.

By exploiting different machine learning and deep learning methods, this research aims to:

- Compare the performances of different classification methods for detecting renal failure
- Investigate how the fluid accumulation can lead to a change in voice and be used as biomarker to detect the decompensation.

1.1 State of the Art

Chronic kidney disease (CKD) has emerged as a significant health concern, with recent research efforts focusing on the potential for automated detection through speech analysis. In the study by Mun et al. [4], glottal features were analyzed in speakers with CKD from the database [5], highlighting their potential for automated CKD detection. The study identified notable differences in glottal source features between CKD patients and non-CKD controls, with CKD-affected speech often exhibiting breathy characteristics. The researchers utilized a combination of voice quality, glottal, and spectral features in classification experiments, demonstrating the efficacy of these features in distinguishing CKD presence. Specifically, the study achieved impressive classification results, with a combined feature set producing an F1-score of 88%. However, it is essential to note that speech analysis studies for renal failure are significantly limited compared to other investigated diseases, indicating a need for further research in this area.

1.2 Outline

Chapter 2, Background, defines speech production and its related features. Also, it gives an overview of feature extraction and classification methods used in the study.

Chapter 3, Datasets, Protocols and Evaluation Metrics, describes the dataset provided for the analysis and defines the experimental protocols and metrics used in the experiments.

Chapter 4, presents the Handcrafted features methodology and gives the relative results with comments.

Chapter 5, reports comments and considerations on the results obtained.

Chapter 2

Background

2.1 Speech production

The speech signal is often represented using a source-filter model, modeled as a two-stage process. The first stage models the sound source originating at the glottis as a time-varying signal $e(t)$, typically a periodic pulse train with pulse spacing τ_p. The second stage acts as a filter that amplifies and attenuates the signal with a continuous impulse response, peaking at chosen resonance frequencies called formants. This filter represents the vocal tract system $v(t)$. The resulting speech signal $s(t)$ is obtained by the convolution of $e(t)$ and $v(t)$ in the time domain:

$$s(t) = e(t) * v(t).$$

In the frequency domain, this involves multiplying the Fourier transform (FT) of the excitation signal by the FT of the vocal tract:

$$S(j\omega) = E(j\omega) \cdot V(j\omega).$$

The resulting waveform is periodic with a period of τ_p and features a line spectrum with a frequency of $\frac{1}{\tau_p}$, with an envelope determined by the vocal tract's frequency response.

The sound source (the glottis): The source of voiced speech sounds emanates from the vibration of the vocal folds within the glottis. When air is forced from the lungs through a closed glottis, the vocal folds vibrate, creating the primary sound source for most speech sounds. However, not all speech sounds are generated this way. Voiceless sounds originate higher in the vocal tract. For example, the voiceless labiodental fricative [f] is produced by air passing through the constriction between the lower lip and upper teeth, with minimal filtering since there's little structure in front to alter the sound.

The filter (the vocal tract): The glottal source wave is filtered within the vocal tract as it progresses towards the external environment. Several anatomical structures play a role in this filtering process, including the epiglottis, pharynx, velum, various parts of the tongue (blade, tip, body, and root), the alveolar ridge, hard palate, teeth, lips, and the nasal cavity. Each component serves to modify the sound originating from the glottal source wave.

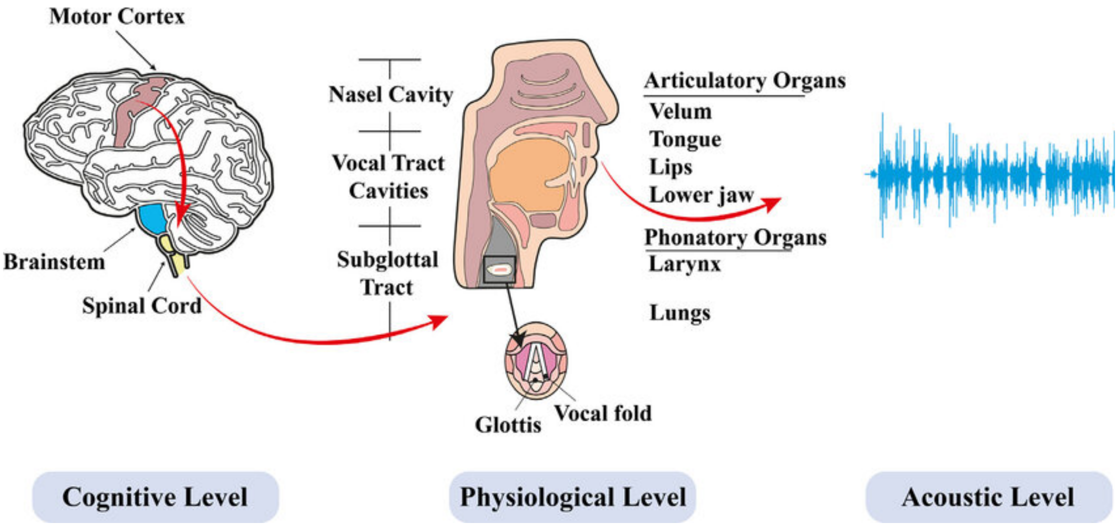

Figure 2.1. Speech production entails a three-level process: cognitive planning level, physiological level (muscular actions), and acoustic level (sound generation) [1]

2.2 Prosodic and Acoustic Features

Speech features can be divided into four main groups: source, spectral, prosodic, and formant features.

Source-related features: These convey information about the glottis during natural voice production, either parameterizing the glottal flow or vocal fold movements. Research has shown that depression affects source measures, primarily focusing on voice quality attributes like jitter, shimmer, and harmonic-to-noise ratio (HNR) [6, 7]. Depressed speech often exhibits breathy and tense voice qualities, indicating a decline in laryngeal coordination [6].

Spectral features: These characterize the speech spectrum, representing the frequency distribution of the speech signal at a specific moment. Common spectral features include Power Spectral Density (PSD) and Mel Frequency Cepstral Coefficients (MFCCs) [8, 9]. Spectral features capture various characteristics such as intensity decay, prosodic irregularities, and articulatory errors. However, their comprehensiveness can pose challenges for classification or prediction systems. Studies have noted shifts in spectral energy, particularly in relation to depression severity [7].

Prosodic features: These represent long-term variations in rhythm, stress, and intonation. Key examples include speaking rate, pitch (fundamental frequency, F0), and loudness [10, 11]. Depression can affect prosodic patterns, often resulting in reduced speaking intensity, narrower pitch range, slower speech rate, diminished intonation, and lack of linguistic stress [10, 11].

Formant features: Formants are dominant components in the speech spectrum, providing information on the resonance properties of the vocal tract. Studies have identified significant differences in formant locations in individuals with depression [7]. For example, changes in the second formant (F2) can indicate slower tongue movements. Formant features are crucial for developing systems to classify depressive speech, with some classifiers achieving high accuracy using these features [6, 11].

2.3 Short-time feature representations

2.3.1 Acustic feature extraction

The extraction of audio features is of great importance for the comprehension and analysis of the characteristics of audio signals across a wide range of fields, including speech recognition, music analysis, environmental sound classification, and medical voice analysis. This research makes use of audio features from the ComParE 2016 set [12], implemented via openSMILE [13]. The concise set of low-level descriptors (105 in total) was selected to enhance the model's explanatory capacity, despite the capability of extracting over 6,000 features using the same package. Features were extracted using the default parameters with a window size of 60 ms and an overlap of 10 ms. For audio features that yield more than one value per sample, the results were averaged. Furthermore, the original nomenclature was revised to enhance the transparency and comprehensibility of the methodology. The principal operations in this phase include feature extraction using openSMILE, temporal averaging to mitigate temporal variations, and standardization to optimize the performance of the model.

2.3.2 Timing features engineering

In the analysis of audio signals, it is of particular importance to consider the influence of pathological changes in the vocal tract on the ability to read or speak. In order to gain a comprehensive understanding of this influence, it is essential to utilize every segment of the audio, including respiration and pauses, which are inherent to audio recordings conducted for aforementioned tasks.

The segmentation of audio is a complex process that requires not only the application of appropriate signal processing techniques but also the implementation of machine learning methodologies to overcome the limitations of rule-based algorithms and assess the results on audio recorded in various environments and with diverse hardware.

In their study, Hlavnička et al. [14] present a complex solution for dividing the recordings into four distinct segments: voiced, unvoiced, respiration, and pauses. This solution incorporates a variety of techniques, including calculation of power, ZCR, or autocorrelation, unsupervised clustering methods, and a set of rules when applying the algorithm. Additionally, the authors present a set of features related to phonation, articulation, respiration, and timing. The efficacy of the proposed method was evaluated on a cohort of Parkinson's disease patients and a control group. In this work, similar algorithm and feature calculation methods were re-implemented and then applied to the investigated dataset.

2.3.3 Spectrogram generation

A spectrogram is a visual representation of the spectrum of frequencies as they vary with time. It is generated from audio signals utilizing the Python library librosa [15]. This process involves converting the audio waveform into a time-frequency domain using the Short-Time Fourier Transform (STFT).

Mathematically, the STFT is defined as:

$$X(t,\omega) = \int_{-\infty}^{\infty} x(\tau)w(\tau - t)e^{-j\omega\tau}d\tau \tag{2.1}$$

where $x(\tau)$ is the input signal, $w(\tau - t)$ is a window function, t represents time, and ω represents frequency.

The results of the vowel trimming algorithm were truncated to the first second of the audio, which aligns with the recommended signal length for analysis [16, 17]. This approach ensured that each audio recording was represented as an array of the same shape. As previously stated in research, the first second of a vowel articulation recording is sufficient to detect pathological changes in voice production.

Additionally, it was decided to utilize a spectrogram normalization technique called Per-Channel Energy Normalization (PCEN) [18]. This approach allows for the independent adjustment of the energy levels of each frequency channel, rendering it effective in a variety of acoustic environments, such as the dataset tested where patients performed recordings in their homes with a considerable amount of background noise. PCEN enhances robustness against noise and variability in recording conditions, which is crucial for reliable model performance in real-world medical applications.

After applying PCEN, log-mel spectrograms were employed, which are the preferred audio frequency representation for DL approaches. Log-mel spectrograms apply a logarithmic transformation to the mel spectrogram, compressing the dynamic range of the audio signal. This compression helps to reduce the effect of very high amplitude values, making the features more manageable and less sensitive to variations in loudness. The logarithmic scale better represents how humans perceive sound. The human auditory perception process is logarithmic in nature, meaning that we are more sensitive to changes in quieter sounds than in louder sounds. By using log-mel spectrograms, the input features are aligned more closely with human hearing, which often leads to improved performance in tasks that involve human-related audio, such as speech or medical voice analysis. In many real-world audio tasks, the relevant features are often found in the mid to low amplitude ranges. The logarithmic transformation is more effective than a linear scale in highlighting these features, thereby improving the ability of the model to learn and discriminate between different classes. This is particularly important in medical applications where subtle differences in audio signals can be critical [19, 20].

A log-mel spectrogram is derived by first computing the Mel-frequency cepstral coefficients (MFCCs) and then applying a logarithmic transformation. Mathematically, the mel scale can be approximated as:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.2}$$

where f is the frequency in Hz, and m is the mel frequency.

The log-mel spectrogram is computed by:

$$\text{Log-Mel}(t, m) = \log \left(\sum_k |X(t, \omega_k)|^2 H_m(\omega_k) \right) \tag{2.3}$$

where $X(t, \omega_k)$ is the STFT of the signal, and $H_m(\omega_k)$ is the mel filterbank.

Log-mel spectrograms are preferred for this type of task because they closely mimic the human ear's perception of sound, providing a more relevant feature space for machine learning models dealing with audio data. They emphasize perceptually important features and compress dynamic range, making it easier for models to learn and generalize from the data.

2.4 Classification methods

Classification is the process of identifying, understanding, and organizing objects and concepts into predefined groups. Machine learning classification algorithms utilize training data to estimate and generate the probability or likelihood that incoming data will belong to one of the established categories or classes. Essentially, a classifier is a model that, based on input training information, assigns new observations to specified classes or clusters.

Selecting suitable classification methods posed a significant challenge. After thorough deliberation, the final decision favored a combination of fundamental techniques commonly used for classification tasks. These techniques include Random Forest (RF), Support Vector Machine (SVM), and Gradient Boosting (GB).

2.4.1 Support Vector Machine

The aim of a Support Vector Machine (SVM) is to identify the optimal hyper-plane in N-dimensional space (where N is the number of features) that correctly classifies the data. Among the possible hyper-planes, the optimal one maximizes the margin—the distance between the nearest data points of each class—which helps minimize classification errors. Generally, a larger margin corresponds to a lower generalization error for the classifier [21].

SVMs are capable of handling both linearly and non-linearly separable data by utilizing the kernel trick. This technique enables the algorithm to implicitly transform the input features into a higher-dimensional space where the data can be linearly separated [22]. As a result, SVMs can effectively tackle complex classification problems that do not have a linear decision boundary in the original feature space.

Support Vector Classification (SVC) is a classification method based on SVMs [23].

The following parameters are crucial for the algorithm and will be fine-tuned to achieve the best F1 score:

- **Kernel**: This parameter determines how the input data is transformed into the hyperplane defined by the kernel's mathematical function. The kernels tested in our case include linear, polynomial, Radial Basis Function (RBF), and sigmoid [23].

- **Regularization Parameter (C)**: This parameter controls the 'smoothness' of the margins, allowing the SVM to tolerate a certain degree of classification error. A high value of C results in a harder model (less tolerant to misclassifications), while a low value of C results in a softer model (more tolerant to misclassifications) [21].
- **Gamma** (γ): This kernel coefficient applies to the sigmoid, RBF, and polynomial kernels. It governs how much influence a single training point has on the surrounding region. Lower gamma values indicate a broader similarity radius, grouping more points together. In contrast, higher gamma values require points to be in very close proximity to each other to be classified within the same category [23].

2.4.2 Random Forest

Random Forest (RF) is an ensemble learning method based on decision trees that combines multiple trees to make predictions [24]. To build each tree within the forest, a random subset of the original training data is chosen using bootstrap sampling, resulting in each tree being trained on a slightly varied dataset, thereby infusing diversity into the forest.

At each node of a decision tree, a random subset of features is considered to determine the best split [24]. This approach ensures that each tree only assesses a subset of features, reducing the risk of any single feature dominating the decision-making process. The tree is constructed by iteratively dividing the data based on different features and thresholds to minimize impurities in the resulting subsets.

Once all the trees are constructed, predictions are made by aggregating the outputs of individual trees through a voting mechanism. In classification tasks, the class that garners the most votes becomes the predicted class [24].

The following parameters are essential for the algorithm and will be fine-tuned to achieve the best F1 score:

- **Number of estimators**: This represents the number of decision trees in the random forest. Typically, the greater the number of estimators, the better the performance of the random forest up to a certain threshold. Nonetheless, it's important to note that an excessive number of estimators can lead to computational complexity and prolonged training times [24].
- **Maximum depth**: A decision tree expands by separating data recursively based on characteristics and thresholds until a stopping criterion, which may be the maximum depth, is satisfied. The trees can recognize more intricate patterns in the data when the maximum depth is greater, although overfitting is also possible. To prevent overfitting and let the trees identify significant correlations in the data, it is critical to tweak this parameter properly.
- **Minimum samples split**: This dictates the minimum number of samples needed to split an internal node within a decision tree [24]. When the number of samples at a node falls below the specified minimum, that node is designated as a leaf node, and any additional splitting is halted. This parameter serves the purpose of regulating the depth of the tree, ensuring that it does not excessively divide regions with inadequate data.

- **Minimum samples leaf**: This establishes a minimum requirement for the number of samples that must be present in a leaf node. Should a split operation lead to a leaf node with fewer samples than the specified minimum, that split is abstained from. Like the minimum samples split parameter, this setting is essential for managing the size and depth of the decision tree and serves as a safeguard against overfitting.

Tuning these parameters is crucial to optimize the Random Forest model's performance and enhance its ability to generalize well on new, unseen data [24].

2.4.3 Gradient Boosting

Gradient Boosting (GB) harnesses the predictive power of decision trees through a sequential approach. Unlike Random Forest, which constructs trees independently, Gradient Boosting builds a sequence of trees, starting with a simple model—typically a shallow tree—and iteratively improving upon it [25]. Each subsequent tree in the ensemble aims to correct the errors of its predecessors [26].

During the construction of each tree, Gradient Boosting assigns higher weights to data points that were previously misclassified or had larger prediction errors . This adaptive weighting ensures that subsequent trees focus more on challenging instances in the data, gradually improving overall predictive accuracy [27].

Variety among the trees is promoted by allowing each tree to consider only a subset of features at each node when determining the optimal split, akin to Random Forest [24]. This strategy mitigates the risk of any single aspect dominating the decision-making process.

Once all trees are constructed and trained, predictions are made by combining the outputs of individual trees. In classification tasks, the final prediction is determined by the class that receives the most weighted votes across all trees, resulting in robust and accurate predictions [25].

Similar to Random Forest, Gradient Boosting shares parameters such as Number of Estimators, Maximum Depth, Minimum Sample Split, and Minimum Samples Leaf. However, it introduces a unique parameter called the "learning rate," which controls the contribution of each tree to the ensemble. This learning rate governs how quickly the model learns from errors during the boosting process, influencing the overall convergence and final model performance.

2.5 Summary

This chapter delves into the essential stages of speech analysis management, emphasizing the generation of voice sounds, feature extraction techniques, and classification methodologies. Feature extraction is a a crucial procedure that converts raw data into meaningful representations, facilitating the analysis of speech signals. Additionally, we explore various classification techniques such as Support Vector Machines, Random Forests and Gradient Boosting. Each of these approaches provides distinct methods for categorizing and analyzing speech signals, each with its own set of hyperparameters, strengths, and limitations.

Chapter 3

Datasets, protocols and evaluation metrics

In this chapter an overview of available datasets and their use is presented, followed by a description of the dataset that the rest of the chapters focus on.

3.1 Dataset

The dataset related to renal failure is part of a series of experiments conducted by Charité Hospital in Berlin, included in the Telemed5000 collection [28]. For simplicity, it will be referred to as Telemed5000 in the following pages.
It includes audio recordings from 91 patients undergoing dialysis over a span of 90 days as part of the study as in table 3.1. The patients, with a mean age of 60 ± 14 years, consisted of 29 women and 62 men, all of German nationality. Each patient contributed voice samples both before and after their dialysis sessions. However, the frequency of dialysis sessions per patient varied, averaging 3 sessions per week as in figure 3.2. All patients utilized the same mobile device for recording.

The audio recordings were sampled at 16 kHz. Data collection involved capturing stable, sustained vowel articulations (/a/, /i/, and /u/) averaging 5.6 ± 1.8 seconds in length per original audio recording, an example of vowel at 3.3. The dataset facilitates a comprehensive exploration of voice changes associated with physiological shifts due to dialysis. Table **??** presents the distribution of recordings, excluding audio files affected by distortions.

3.2 Organising dataset

To achieve the scientific objective, it was determined that a meticulously curated dataset was essential. The initial approach for the task aimed to maximize or minimize fluid accumulation. To accomplish this, the dialysis events for each patient were retrieved.

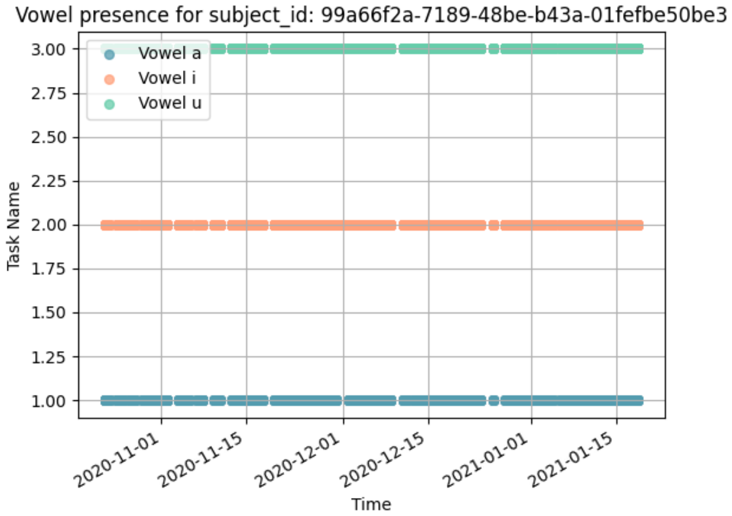

Figure 3.1. Presence of recording over time for a given patient.

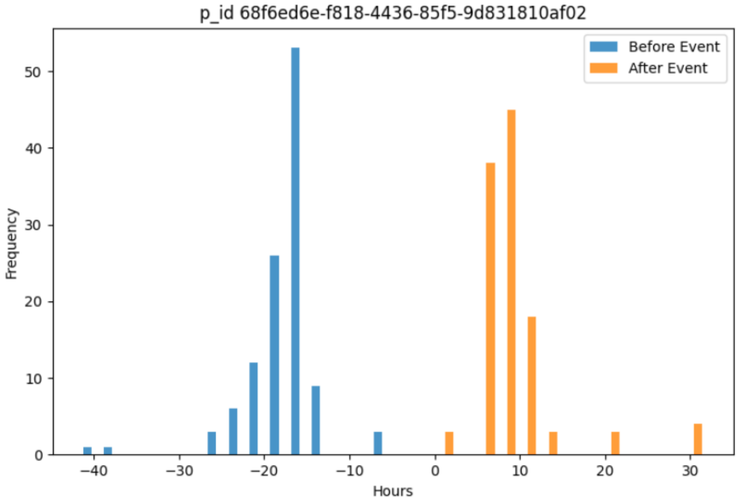

Figure 3.2. Distance in time from dialysis event for a given patient.

Subsequently, the nearest recordings in time before and after each event were selected. The state before the event was labeled as ’1’ (wet state), while the state after was labeled as ’0’ (dry state).

In order to maximize even more the fluid accumulation a tolerance criteria was applied in terms of hour for the before and after, respectively 18 and 12 hours, as showed in figure

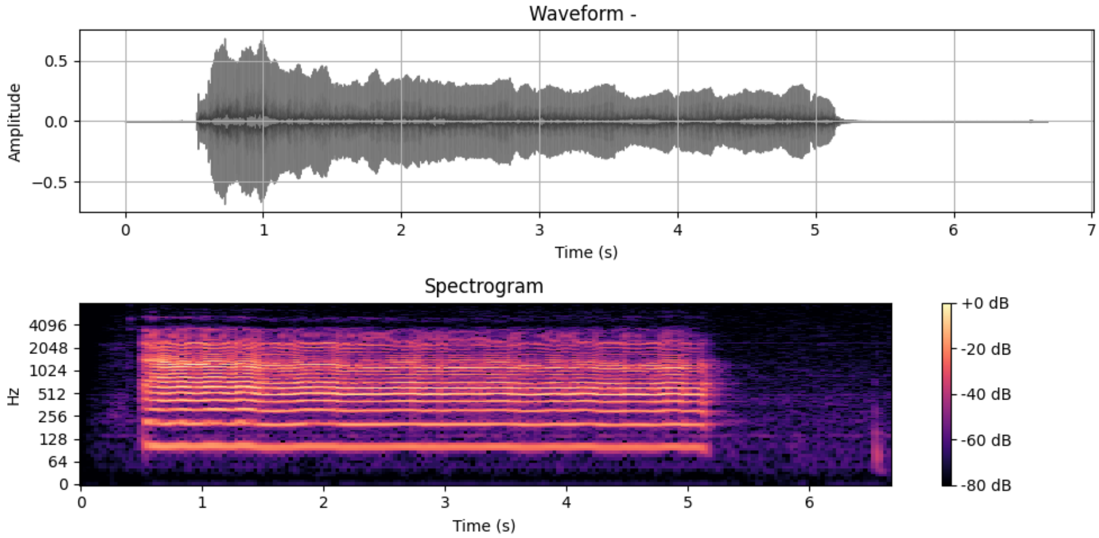

Figure 3.3. Example of voice waveform (top) and log spectrogram (bottom) present in the dataset.

Table 3.1. **Dataset description**

	n=91 (100%)	**Mean±SD**	**Task**		
			vowel /a/	vowel /i/	vowel /u/
			7518	7513	7508
Female	29 (32%)		2376	2375	2374
Age		59±13			
Male	62 (68%)		4961	4957	4953
Age		60±14			

Notes:
n - Number of patients,
SD - Standard Deviation,
The % are rounded.

3.4. This was done since many recordings where to far away in time to be considered. These thresholds were also specifically chosen because of the prolonged inactivity typically occurring the night before dialysis, requiring a longer pre-dialysis window compared to the post-dialysis window, as shown in figure.

Another precaution that has been implemented is to include only pairs of dialysis events where both the before and after events are present, ensuring a balanced dataset as shown in figure 3.5, in this case the recording between Friday and Saturday has been discarded since there is no recording in that time tolerance for the before.

Since voice changes through time the recording should not be considered as independent. In order to keep the temporal component into account each, recording per vowel task, was concatenated with the following one temporally. If between the recordings a dialysis event was present the label applied would be "1", "0" otherwise. The intuition is that between dialysis event the change in voice is greater than when there is no dialysis event. The visual representation of the process is at figure 3.6. The strength of this approach is that the model is able to compare each state t with the following $t+1$ making a comparison between the two. It's important to notice that the concatenation is just one of the possible operation that can be performed between the recording features.

In the dataset construction and in the following analysis some attention has been devoted to the difference between type of patients. In particular since voice change accordingly to the amount of fluid retrained by the body the differentiation between patient category is fundamental. In particular we can divide patients in tree main categories:

- Anuric: patients who are able to expel naturally less than 100ml
- Oliguric: patients who are able to expel naturally between 100ml and 500ml
- Normuric: patients who are able to expel naturally more than 500ml

This distinction is crucial since we expect better results for those patient who are more unable to expel fluid naturally and will benefit more from the dialysis resulting in a bigger delta in fluid accumulation.

3.3 Performance metrics

Accuracy counts the number of times a model correctly predicts over the entire dataset. However, this measure is only reliable if the dataset is class-balanced, meaning each class contains an equal number of samples. Therefore, for binary classification tasks like depression detection, other metrics such as the F1 score, precision (P), and recall (R) are often used.

These metrics can be calculated using frame-level results, including true positives (TP), false positives (FP), and false negatives (FN). Precision measures the proportion of predicted positive samples that are actually true positives. Recall is calculated as the ratio of true positives (TP) to the sum of true positives and false negatives (FN). The F1 score, which is the harmonic mean of recall and precision, provides a single metric that balances both precision and recall.

These metrics are computed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

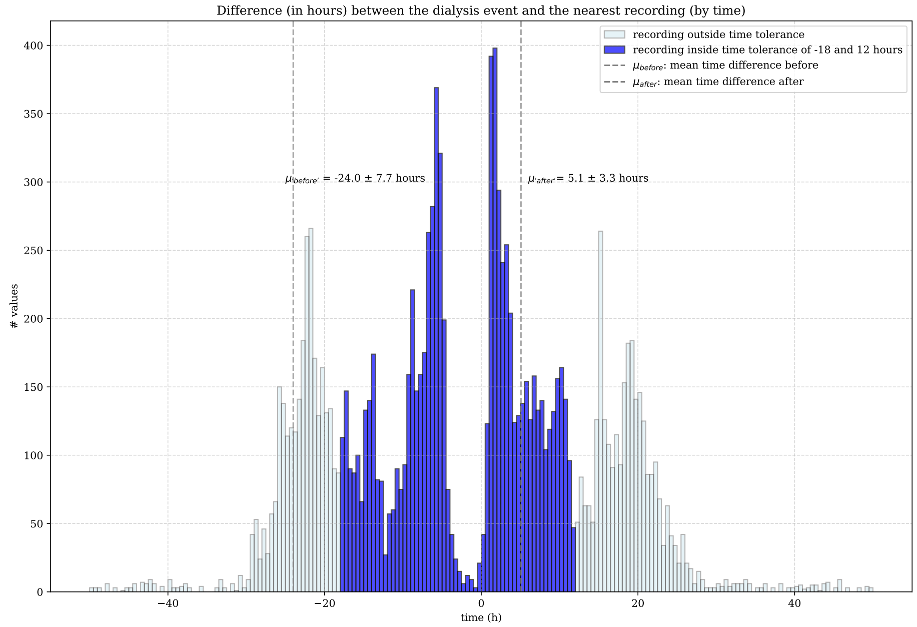

Figure 3.4. Difference between the dialysis event and the nearest recording in hours.

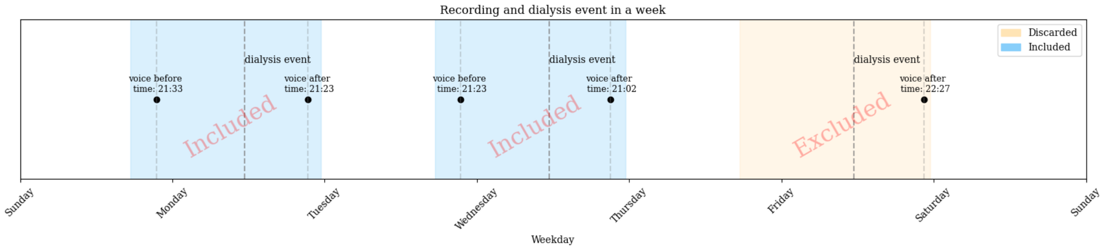

Figure 3.5. Difference between the dialysis event and the nearest recording in hours.

$$\text{Precision (P)} = \frac{TP}{TP + FP} \tag{3.2}$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} \tag{3.3}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

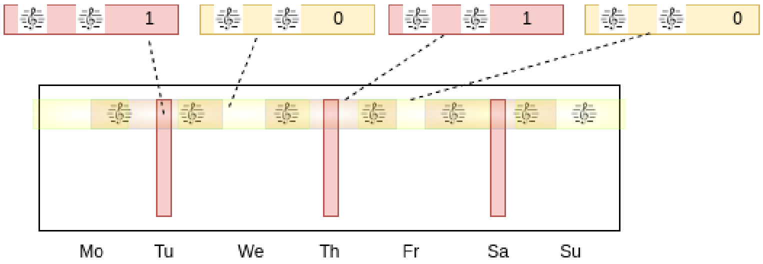

Figure 3.6. Visual representation of the pairwise approach

3.4 Summary

This chapter introduces the telemed5000 dataset used in the study. It is composed by a total of 91 patients, 29 female and 62 man, each monitored for 90 days. All the patients are German speaker. The main strategy to deal with the dataset are presented alongside with the underlying hypothesis of a correlation between a fluid build-up and a chage in voice. The evaluation metrics are also presented to asses the performance of the model.

Chapter 4

Handcrafted feature study

A fundamental step for audio analysis is to transform the audio recordings into a set of features. The features are extracted from speech by a feature extractor. They are aggregated to obtain a fixed length representation of the the speaker and use them as input for the classifier.

4.1 Feature Extraction

To achieve uniform power distribution among all voice samples during analysis, loudness normalization following the ITU-R BS.1770 standard was implemented using the `pyloudnorm` library [29, 30]. Research has shown that this method produces favorable results in speech prediction and recognition tasks [31]. The procedure involves assessing the loudness level of each audio sample and adjusting it to a uniform target loudness level of -23 LUFS. Following this, peak normalization is applied to the audio tracks to ensure that the peak is at the highest or lowest permissible level without causing distortion. This involves determining the maximum absolute value of the audio signal and then scaling all samples in the signal by this value. The scaling ensures that the peak amplitude ranges between 1.0 and -1.0 [29].

A concern that needs to be addressed is the introduction of noise or unwanted silence in audio recordings. This occurs because, when someone records audio, there is always a period of silence after the recorder is turned on and before it is turned off. For the study, an automated tool was developed to properly trim audio where there is no vowel articulation. It has been shown that there are differences between speech and non-speech audio segments in terms of the following features: Zero Crossing Rate (ZCR) and Short-Time Energy (STE) values [32, 33]. The ZCR for speech segments is much lower, while the STE is higher. Conversely, for non-speech segments, the values are reversed [33].

The features are calculated as follows:

$$ZCR = \frac{1}{T-1}\sum_{n=1}^{T-1} |\,\mathrm{sgn}(x[n]) - \mathrm{sgn}(x[n-1])| \tag{4.1}$$

$$\operatorname{sgn}(x[n]) = \begin{cases} 1 & \text{if } x[n] \geq 0 \\ -1 & \text{if } x[n] < 0 \end{cases} \tag{4.2}$$

$$STE = \sum_{n=0}^{T-1} x[n]^2 \tag{4.3}$$

Where $x[n]$ indicates the discrete time signal, sgn represents the sign function, and T is the total number of samples in the frame.

The vowel articulation is a voiced sound, which means that it is a quasi-periodic signal. This is due to the glottal vibrations that are responsible for producing this sound [33]. Studies have shown that the length of the range of voiced sounds lasts between 20 and 30 milliseconds [32]. Therefore, the window length for the calculation of features was set to 25 milliseconds with an overlap of 12.5 milliseconds. This overlap is crucial for the correct reconstruction of the signal. Furthermore, the application of the window function, in this case, the Hamming window, makes the process of reconstruction almost perfect. This is known as the overlap-add process [34]. However, it is common practice in studies to set the threshold manually in order to distinguish between segments. An alternative approach is to apply Gaussian Mixture Models (GMM), which have been shown to be a commonly used method for speech segment classification and applied for this work [35].

Furthermore, the majority-voting technique is utilized to mitigate the risk of some labels being incorrectly identified. This algorithm is used to reclassify frames that were initially marked as unwanted artifacts, reassigning them to the vowel articulation category. This method operates under the assumption that during a sustained vowel articulation task, the audio will exhibit a pattern of artifacts, vowel articulation, followed by more artifacts [36]. Additionally, the minimum articulation interval is set to 1 second. Throughout the implementation of these methods, the original audio sampling rate was preserved to ensure that no valuable waveform information was lost.

4.2 Hyperparameter Tuning

To find the most efficient set of hyperparameter values for a specific model, grid search is a well-known hyperparameter optimization technique [37]. This process involves selecting the best set of hyperparameters before the training phase, significantly impacting the model's performance [38].

Each combination of values in the grid is used to train and evaluate the model using a predefined evaluation metric described in Section 3.4 [39]. This systematic evaluation process assesses the model's performance across all possible hyperparameter combinations.

The ultimate objective of grid search is to identify the most efficient set of hyperparameters that produces the best performance on the evaluation measure. This optimal combination is then selected as the most suitable set to evaluate the algorithm on a test set, which is separated out before the k-fold cross-validation, according to the protocol splitting for each dataset [40].

4.3 Hypothesis and results

The following is a brief explanation of the methodology and the results.

In the context of renal failure monitoring using machine learning, this study proposes the use of voice as a biomarker to detect dialysis events between two audio recordings. The rationale stems from the well-established relationship between changing vocal characteristics and fluid accumulation in patients suffering from kidney failure. Rather than using a traditional time series model, which can be cumbersome for this application, the approach is to compare each audio recording with its predecessor. This method aims to use machine learning techniques to detect subtle differences between successive recordings, thereby identifying potential dialysis events based on voice biomarkers.

The experiment involved 91 patients, each individually trained using grid search to optimise model parameters. Metrics were computed and aggregated across predefined categories within the dataset. Due to the recognised challenge of generalising models across patients, this approach was chosen. The study acknowledges the difficulty of achieving cross-patient generalisation. However, this aspect is left for future analysis beyond the scope of this research.

The following section describes the results of experiments on kidney failure datasets applying the traditional pipeline. For each dataset, first, the performance will be discussed, and the different classifiers for each feature set will be com- pared. Then, for the best-performing models, a qualitative analysis of the most representative features for renal failure detection will be presented.

4.3.1 Kidney failure: classification results

Table 4.2 presents the aggregated results of the classification using SVM. The results are grouped first by type of disease (anuric, normuric, oliguric) and then based on this difference by sex. The total number of patients involved in the aggregation of the results is smaller than the initial number, because for a few of them there was a high imbalance between the number of 0/1 labels, resulting in a high accuracy value, which would have affected the mean values after aggregation. The process of removing those patient has been done with a quantile analysis, removing those patient with a support less than 11 for the 0 class and less than 14 for the class 1 as in figure 4.2, this has been done to include in the computation of the results only those patients having a large enough number of samples in the test set.x

The same results for the accuracy and F1 values can be visualized in the box plots in figure 3.1 and 3.4 respectively.

The same tables and plots are shown for XGBoost classifier below.

As it's clearly visible from the tables above SVM performs better than XGBoost and RF across all the categories.

It's also worth notice how the initial hypothesis regarding patients across the different categories is reflected in the results showing better performances in both accuracy and F1 score when dealing with anuric patient rather than normuric and oliguric. This fact clearly shows the major impact that fluid accumulation brings to voice.

Despite this results provide high performances for some categories the standard deviation remains high indicating a poor capability of the model to generalize well for all the patients.

The comparison across model is shown in figure 4.1

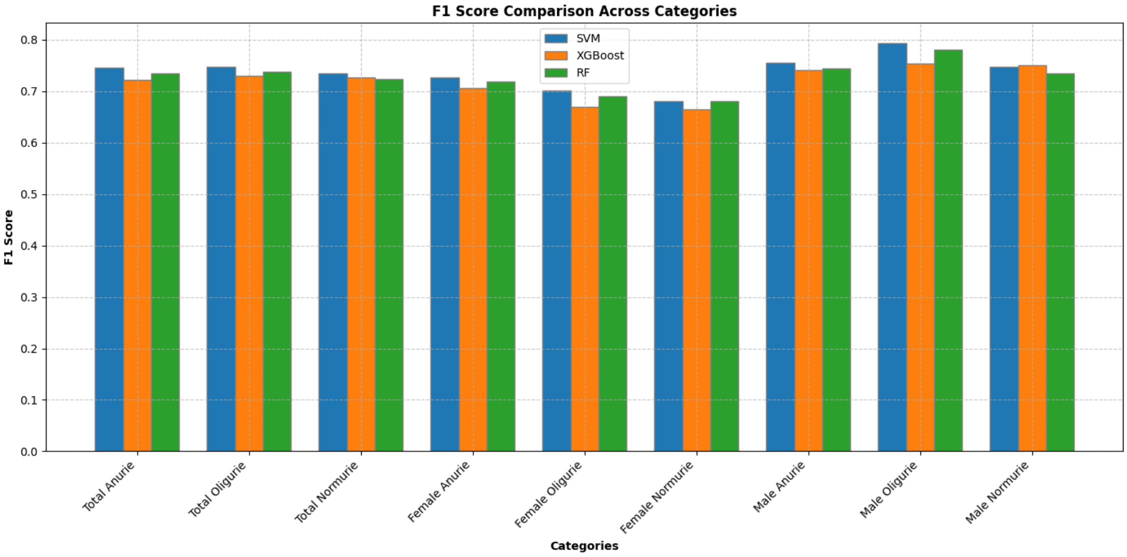

Figure 4.1. Performance across models

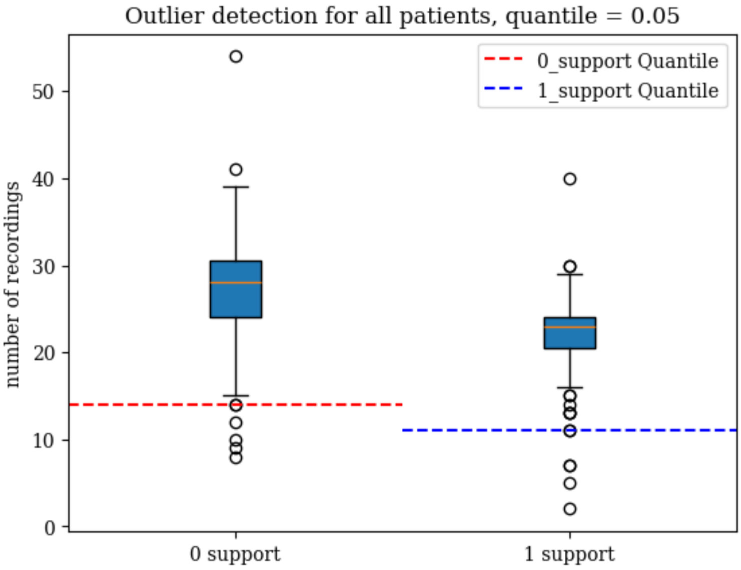

Figure 4.2. Outlier detection for all patients. patients having less than the quantile value for both classes are removed from the result aggregation.

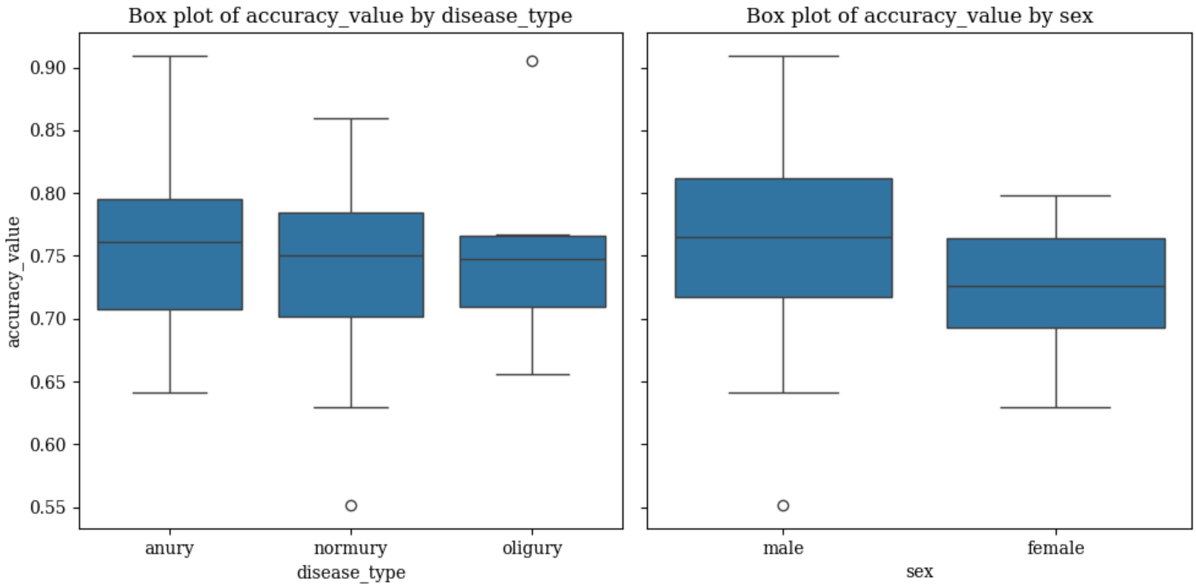

Figure 4.3. Accuracy values using SVM grouped by disease type on the left and by sex on the right

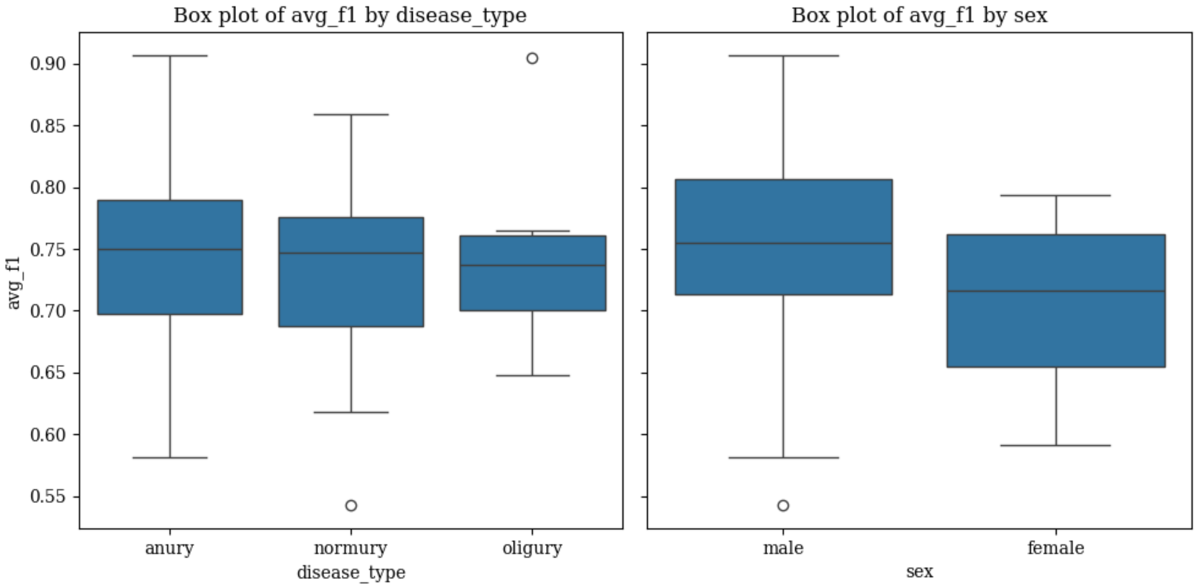

Figure 4.4. F1 values using SVM grouped by disease type on the left and by sex on the right

Table 4.1. **Analysis of Telemed5000 dataset with differential classifier.**

Renal Failure

	n = 76 (100%)	Mean ± SD		
			Accuracy	F1
Total Anurie	42 (54%)		0.758 ± 0.065	0.746 ± 0.074
Total Oligurie	6 (36%)		0.754 ± 0.085	0.747 ± 0.088
Total Normurie	28 (34%)		0.740 ± 0.069	0.734 ± 0.074
Female	23 (30%)			
Anurie	14 (18%)		0.741 ± 0.046	0.726 ± 0.063
Oligurie	3 (3%)		0.708 ± 0.056	0.702 ± 0.059
Normurie	6 (8%)		0.700 ± 0.053	0.681 ± 0.063
Age		59 ± 13		
Men	54 (70%)			
Anurie	28 (36%)		0.767 ± 0.072	0.756 ± 0.079
Oligurie	3 (3%)		0.800 ± 0.093	0.793 ± 0.098
Normurie	22 (28%)		0.751 ± 0.070	0.748 ± 0.071
Age		60 ± 14		

Notes:
n - Number of patients,
SD - Standard Deviation,
Training was conducted on a per-patient basis using **Support Vector Machines (SVM)** with grid search for hyperparameter tuning. For each patient, the model was evaluated 10 times, each with a different random state for the train/test split. The final results were stored and aggregated.
The % are rounded.

Table 4.2. **Analysis of Telemed5000 dataset with differential classifier.**

Renal Failure				
	n = 76 (100%)	Mean ± SD		
			Accuracy	F1
Total Anurie	42 (54%)		0.732 ± 0.048	0.722 ± 0.044
Total Oligurie	6 (36%)		0.731 ± 0.039	0.730 ± 0.068
Total Normurie	28 (34%)		0.740 ± 0.069	0.727 ± 0.049
Female	23 (30%)			
Anurie	14 (18%)		0.727 ± 0.036	0.706 ± 0.061
Oligurie	3 (3%)		0.701 ± 0.052	0.67 ± 0.050
Normurie	6 (8%)		0.700 ± 0.049	0.665 ± 0.043
Age		59 ± 13		
Men	54 (70%)			
Anurie	28 (36%)		0.734 ± 0.058	0.741 ± 0.053
Oligurie	3 (3%)		0.793 ± 0.074	0.754 ± 0.099
Normurie	22 (28%)		0.731 ± 0.072	0.751 ± 0.051
Age		60 ± 14		

Notes:
n - Number of patients,
SD - Standard Deviation,
Training was conducted on a per-patient basis using **XGBoost** with grid search for hyperparameter tuning. For each patient, the model was evaluated 10 times, each with a different random state for the train/test split. The final results were stored and aggregated.
The % are rounded.

Table 4.3. Grid search parameters and value for SVM, RF and GB

Model	paramters	Grid search values
SVM	C	[0.1, 1, 10, 100]
	γ	[0.001, 0.01, 0.1, 1]
	Kernel	Linear, RBF, polynomial, sigmoid
RF	Number of estimators	[10, 20, 30, 40, 50, 70, 80, 100, 150, 200]
	Maximum depth	[5, 7, 10, 20]
	Minimum samples split	[2, 3, 5, 7, 10, 15]
	Minimum samples leaf	[3, 4, 5]
GB	Number of estimators	[10, 20, 30, 40, 50, 70, 80, 100, 150, 200]
	Maximum depth	[5, 7, 10, 20]
	Minimum samples split	[2, 3, 5, 7, 10, 15]
	Minimum samples leaf	[3, 4, 5]
	Learning rate	[0.001, 0.01, 0.1]

Table 4.4. **Analysis of Telemed5000 dataset with Random Forest classifier.**

Renal Failure

	n = 76 (100%)	Mean ± SD		
			Accuracy	F1
Total Anurie	40 (52%)		0.748 ± 0.060	0.735 ± 0.072
Total Oligurie	7 (9%)		0.745 ± 0.078	0.738 ± 0.082
Total Normurie	30 (39%)		0.730 ± 0.065	0.723 ± 0.071
Female	22 (29%)			
Anurie	12 (16%)		0.730 ± 0.050	0.718 ± 0.061
Oligurie	4 (5%)		0.700 ± 0.052	0.690 ± 0.057
Normurie	6 (8%)		0.695 ± 0.051	0.680 ± 0.062
Age		58 ± 12		
Men	55 (71%)			
Anurie	28 (36%)		0.760 ± 0.070	0.745 ± 0.078
Oligurie	3 (4%)		0.790 ± 0.089	0.780 ± 0.093
Normurie	24 (31%)		0.740 ± 0.065	0.735 ± 0.068
Age		61 ± 13		

Notes:
n - Number of patients,
SD - Standard Deviation,
Training was conducted on a per-patient basis using **Random Forest (RF)** with grid search for hyperparameter tuning. For each patient, the model was evaluated 10 times, each with a different random state for the train/test split. The final results were stored and aggregated.
The % are rounded.

Chapter 5

Final Analysis

The best results were observed within the male population, prompting a detailed analysis of the relationship between the results and the fluid extracted during dialysis.

Figure 5.1 displays a scatter plot of the median F1 score weighted per class versus the median ultrafiltration extracted during the dialysis period per patient. These values were obtained using an SVM model, which was proven to be the best-performing model. The analysis was conducted using only patients with more than 50 recordings throughout their hospitalization period. The plot shows the labels provided by the hospital. It is evident from the definitions given that some patients who, theoretically, should not be able to eject a large amount of fluid are shown to have minimal fluid extraction, while those capable of ejecting fluid have a higher volume of fluid extracted. Despite the anuric and normuric categorization not being strictly defined in medical terms, it affects the research outcome. The expectation would be to have a higher F1 score among the anuric group compared to the normuric group. In this study, the mean F1 score for anuric patients is 74.45%, while for normuric patients, it is 73.79%.

We further examined how the results would vary by redefining the threshold for categorizing anuric and normuric patients at 2, 2.5, and 3 liters. The results and corresponding graphs are presented in Figures 5.2, 5.3, and 5.4 respectively.

The summary graph in Figure 5.5 illustrates that with the hospital's original categorization, there is minimal difference between anuric and normuric patients. However, when using a custom threshold of 2 liters, the difference becomes pronounced, highlighting the effect of fluid accumulation on the patient's body. As expected, raising the threshold increases the fluid retention values for both normuric and anuric groups, as more patients in both groups are classified as having higher fluid retention.

Despite some values supporting the underlying hypothesis, the linear correlation between the F1 score and the median ultrafiltration is only 0.3803. Specifically, the anuric patients located in the bottom right part of the plot present the most significant issue.

When increasing the threshold for the number of recordings required for the model, the scatter plots are shown in Figures 5.6, 5.7, 5.8, and 5.9.

In this scenario the correlation gets much stronger since the amount of recordings per patients requires is larger. Applying this threshold the correlation is above 0.67% showing a clear effect of the fluid accumulation on the model performance.

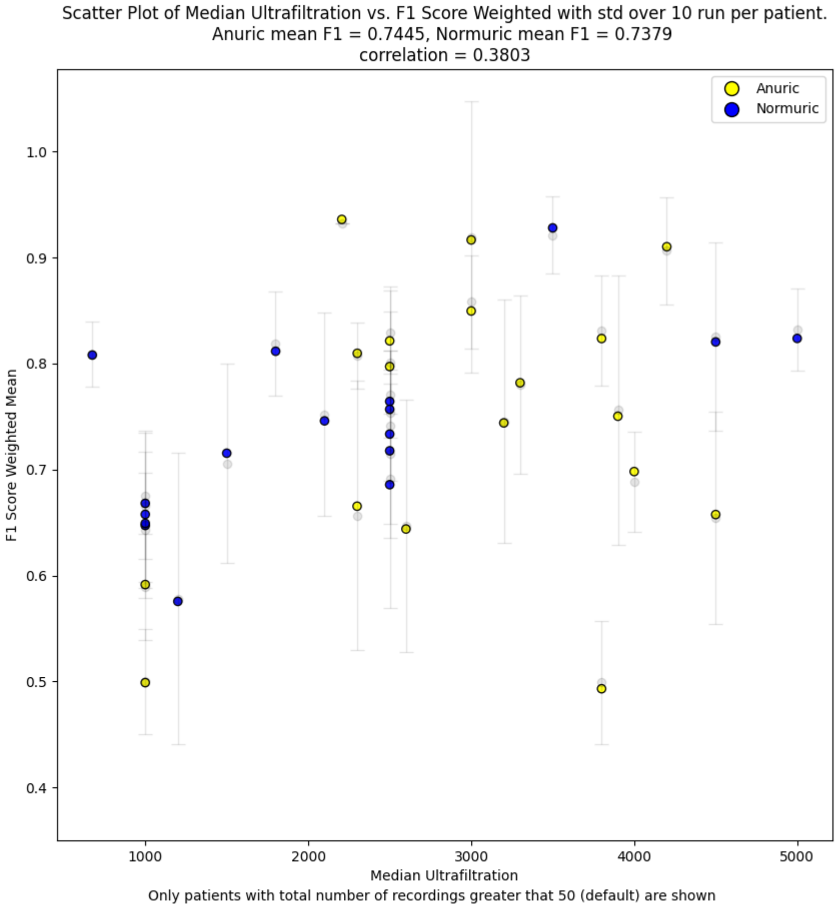

Figure 5.1. Scatter plot of weighted F1 score versus median ultrafiltration extraced per patient.

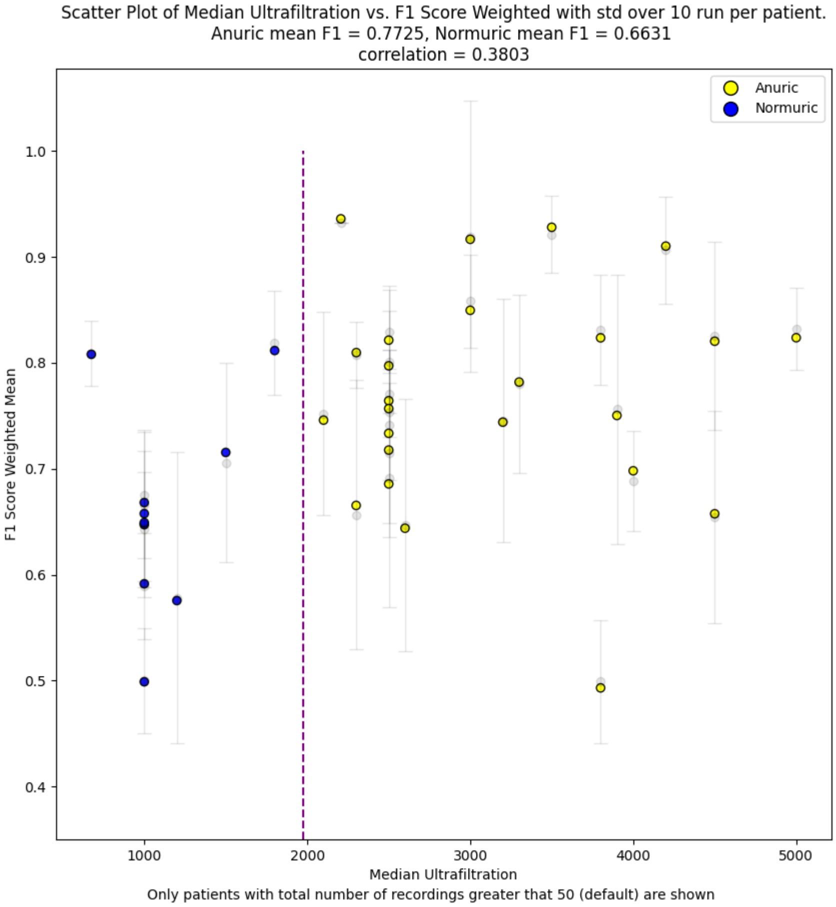

Figure 5.2. Scatter plot of weighted F1 score versus median ultrafiltration extraced per patient.

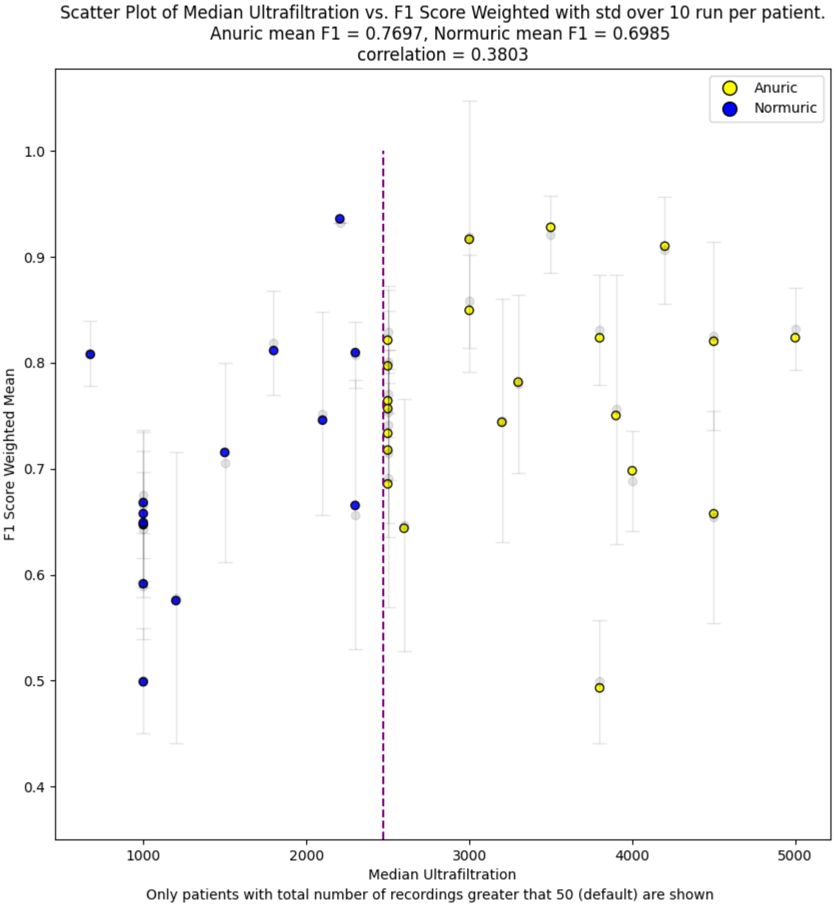

Figure 5.3. Scatter plot of weighted F1 score versus median ultrafiltration extraced per patient.

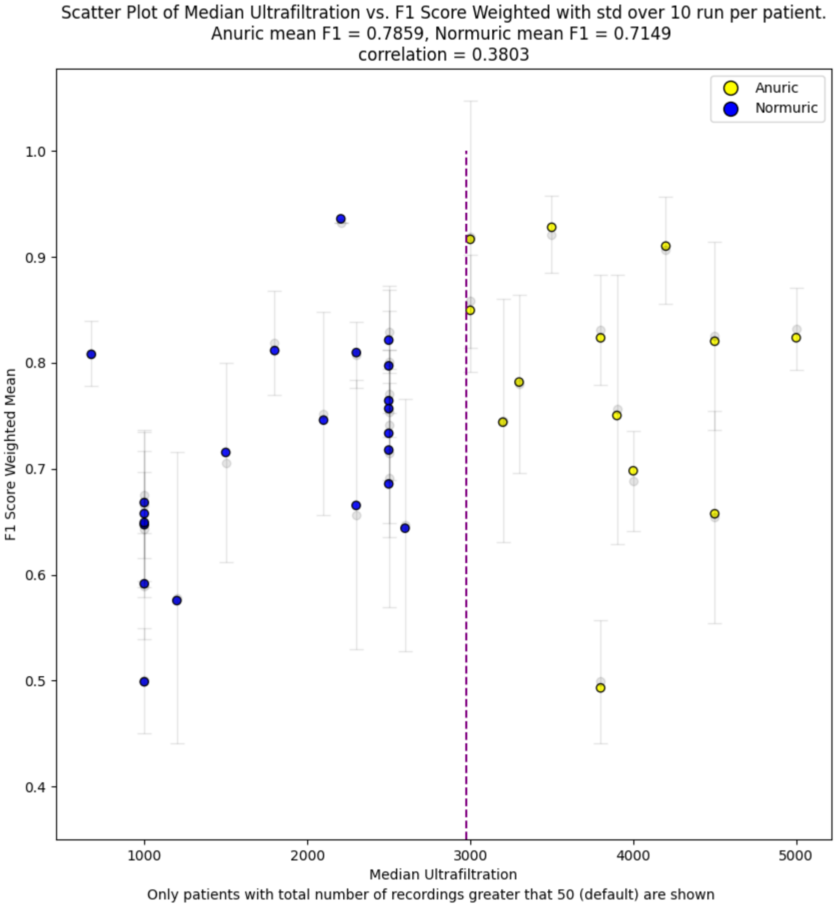

Figure 5.4. Scatter plot of weighted F1 score versus median ultrafiltration extraced per patient.

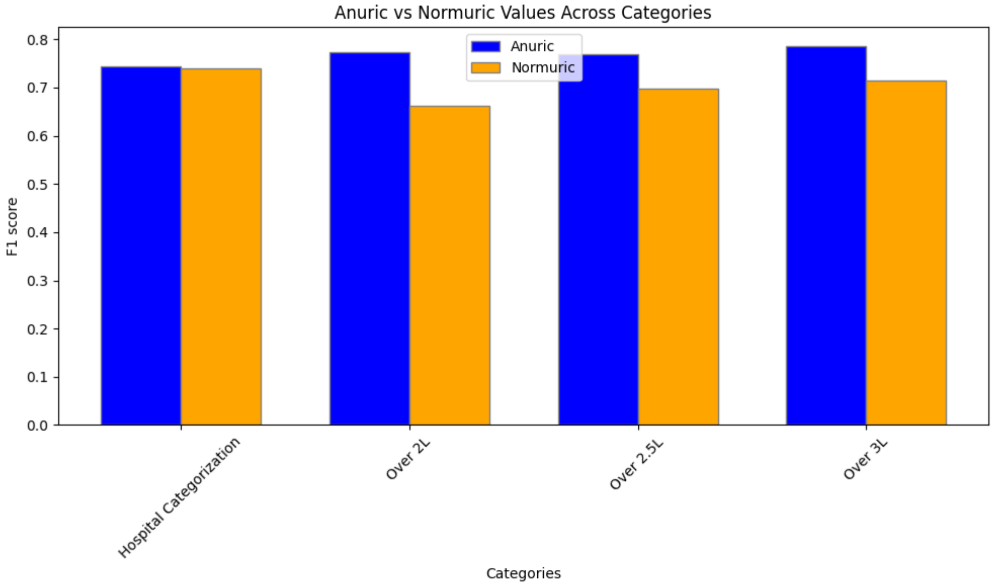

Figure 5.5. Scatter plot of weighted F1 score versus median ultrafiltration extraced per patient.

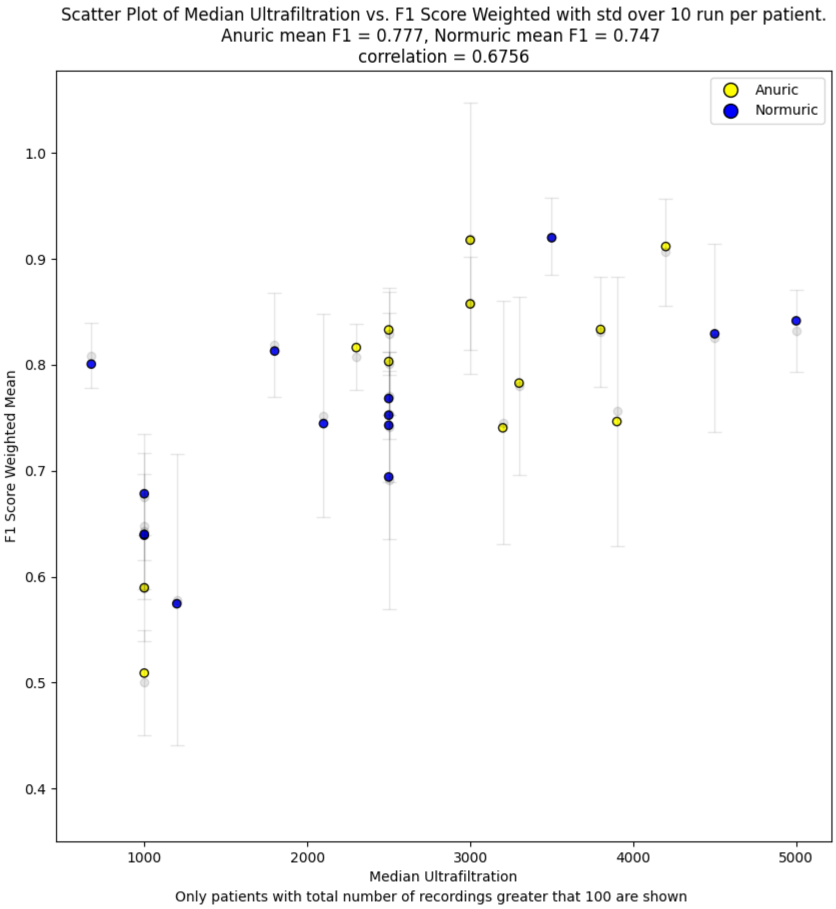

Figure 5.6. Scatter plot of weighted F1 score versus median ultrafiltration extraced per patient.

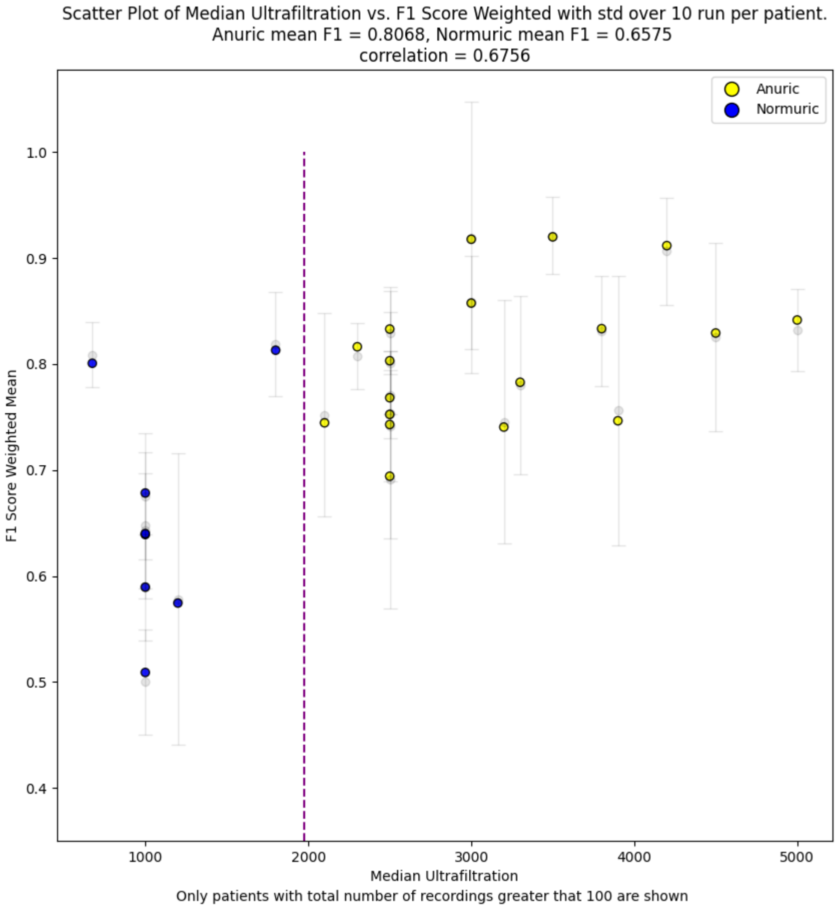

Figure 5.7. Scatter plot of weighted F1 score versus median ultrafiltration extraced per patient.

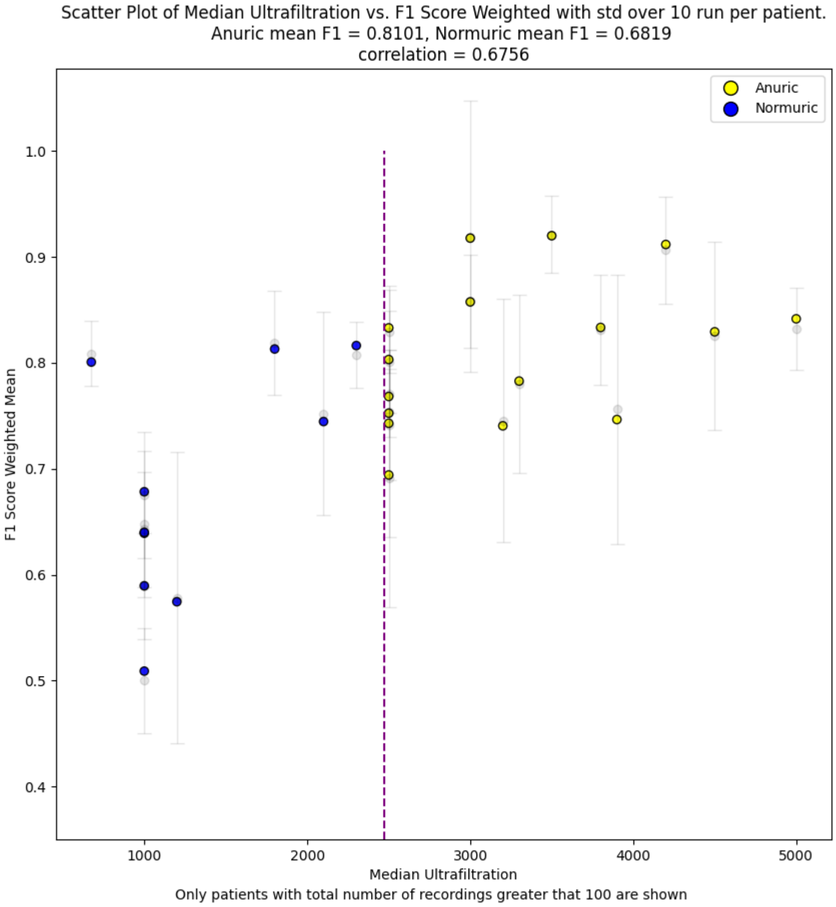

Figure 5.8. Scatter plot of weighted F1 score versus median ultrafiltration extraced per patient.

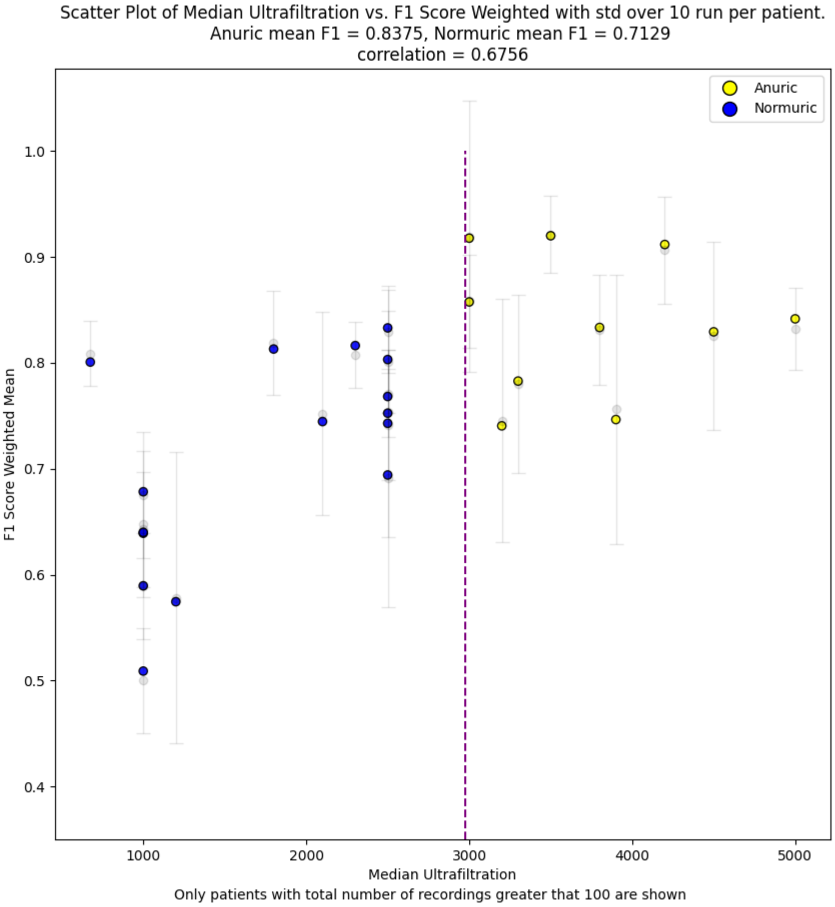

Figure 5.9. Scatter plot of weighted F1 score versus median ultrafiltration extraced per patient.

Chapter 6

Conclusions

The present research leverage the potential of speech analysis as a promising, non-invasive tool for detecting renal failure, particularly through the nuanced examination of vocal characteristics influenced by fluid accumulation. The observed distinctions between anuric and normuric patients highlight the profound impact of fluid retention and associated symptoms on speech patterns. These findings reinforce the hypothesis that vocal source and vocal tract features can serve as critical indicators of renal impairment.

The results of our experiments suggest that comprehensive spectral information—including fundamental frequency and other vocal tract-related features—is essential for accurate classification. This emphasis on retaining detailed acoustic information is crucial for improving detection accuracy, given the complex interplay between fluid retention, renal failure symptoms, and their effects on speech production.

Moreover, our analysis of mutual information values reveals that the influence of fluid retention on speech cannot be underestimated. The correlation between fluid accumulation and the model's ability to differentiate between "dry" and "wet" states underscores the significance of considering fluid-related factors in speech analysis for renal failure.

In summary, this research not only provides insights into the acoustic markers of renal failure but also lays the groundwork for further exploration in this field. Future studies might continue to refine feature extraction techniques and explore additional acoustic indicators to enhance the reliability and applicability of speech-based diagnostic tools for renal failure. Our findings advocate for the integration of speech analysis into clinical practice, offering a novel approach to early detection and monitoring of renal conditions.

Appendix A

Handcrafted features

Table A.1: Feature Descriptions

Feature Name	Description
meanF0_MEAN	Mean of the fundamental frequency (F0)
meanF0_STD	Standard deviation of the fundamental frequency (F0)
meanF0_MIN	Minimum value of the fundamental frequency (F0)
meanF0_MAX	Maximum value of the fundamental frequency (F0)
meanF0_RANGE	Range of the fundamental frequency (F0)
meanF0_KURTOSIS	Kurtosis of the fundamental frequency (F0)
meanF0_SKEWNESS	Skewness of the fundamental frequency (F0)
stdevF0_MEAN	Mean of the standard deviation of the fundamental frequency (F0)
stdevF0_STD	Standard deviation of the standard deviation of the fundamental frequency (F0)
stdevF0_MIN	Minimum value of the standard deviation of the fundamental frequency (F0)
stdevF0_MAX	Maximum value of the standard deviation of the fundamental frequency (F0)
stdevF0_RANGE	Range of the standard deviation of the fundamental frequency (F0)
stdevF0_KURTOSIS	Kurtosis of the standard deviation of the fundamental frequency (F0)
stdevF0_SKEWNESS	Skewness of the standard deviation of the fundamental frequency (F0)
hnr_MEAN	Mean of the harmonic-to-noise ratio (HNR)
hnr_STD	Standard deviation of the harmonic-to-noise ratio (HNR)

Feature Name	Description
hnr_MIN	Minimum value of the harmonic-to-noise ratio (HNR)
hnr_MAX	Maximum value of the harmonic-to-noise ratio (HNR)
hnr_RANGE	Range of the harmonic-to-noise ratio (HNR)
hnr_KURTOSIS	Kurtosis of the harmonic-to-noise ratio (HNR)
hnr_SKEWNESS	Skewness of the harmonic-to-noise ratio (HNR)
localJitter_MEAN	Mean of the local jitter
localJitter_STD	Standard deviation of the local jitter
localJitter_MIN	Minimum value of the local jitter
localJitter_MAX	Maximum value of the local jitter
localJitter_RANGE	Range of the local jitter
localJitter_KURTOSIS	Kurtosis of the local jitter
localJitter_SKEWNESS	Skewness of the local jitter
localabsoluteJitter_MEAN	Mean of the local absolute jitter
localabsoluteJitter_STD	Standard deviation of the local absolute jitter
localabsoluteJitter_MIN	Minimum value of the local absolute jitter
localabsoluteJitter_MAX	Maximum value of the local absolute jitter
localabsoluteJitter_RANGE	Range of the local absolute jitter
localabsoluteJitter_KURTOSIS	Kurtosis of the local absolute jitter
localabsoluteJitter_SKEWNESS	Skewness of the local absolute jitter
rapJitter_MEAN	Mean of the RAP jitter
rapJitter_STD	Standard deviation of the RAP jitter
rapJitter_MIN	Minimum value of the RAP jitter
rapJitter_MAX	Maximum value of the RAP jitter
rapJitter_RANGE	Range of the RAP jitter
rapJitter_KURTOSIS	Kurtosis of the RAP jitter
rapJitter_SKEWNESS	Skewness of the RAP jitter
ppq5Jitter_MEAN	Mean of the PPQ5 jitter
ppq5Jitter_STD	Standard deviation of the PPQ5 jitter
ppq5Jitter_MIN	Minimum value of the PPQ5 jitter
ppq5Jitter_MAX	Maximum value of the PPQ5 jitter
ppq5Jitter_RANGE	Range of the PPQ5 jitter
ppq5Jitter_KURTOSIS	Kurtosis of the PPQ5 jitter
ppq5Jitter_SKEWNESS	Skewness of the PPQ5 jitter
ddpJitter_MEAN	Mean of the DDP jitter
ddpJitter_STD	Standard deviation of the DDP jitter
ddpJitter_MIN	Minimum value of the DDP jitter
ddpJitter_MAX	Maximum value of the DDP jitter
ddpJitter_RANGE	Range of the DDP jitter
ddpJitter_KURTOSIS	Kurtosis of the DDP jitter
ddpJitter_SKEWNESS	Skewness of the DDP jitter
localShimmer_MEAN	Mean of the local shimmer
localShimmer_STD	Standard deviation of the local shimmer

Feature Name	**Description**
localShimmer_MIN	Minimum value of the local shimmer
localShimmer_MAX	Maximum value of the local shimmer
localShimmer_RANGE	Range of the local shimmer
localShimmer_KURTOSIS	Kurtosis of the local shimmer
localShimmer_SKEWNESS	Skewness of the local shimmer
localdbShimmer_MEAN	Mean of the local dB shimmer
localdbShimmer_STD	Standard deviation of the local dB shimmer
localdbShimmer_MIN	Minimum value of the local dB shimmer
localdbShimmer_MAX	Maximum value of the local dB shimmer
localdbShimmer_RANGE	Range of the local dB shimmer
localdbShimmer_KURTOSIS	Kurtosis of the local dB shimmer
localdbShimmer_SKEWNESS	Skewness of the local dB shimmer
apq3Shimmer_MEAN	Mean of the APQ3 shimmer
apq3Shimmer_STD	Standard deviation of the APQ3 shimmer
apq3Shimmer_MIN	Minimum value of the APQ3 shimmer
apq3Shimmer_MAX	Maximum value of the APQ3 shimmer
apq3Shimmer_RANGE	Range of the APQ3 shimmer
apq3Shimmer_KURTOSIS	Kurtosis of the APQ3 shimmer
apq3Shimmer_SKEWNESS	Skewness of the APQ3 shimmer
aqpq5Shimmer_MEAN	Mean of the APQ5 shimmer
aqpq5Shimmer_STD	Standard deviation of the APQ5 shimmer
aqpq5Shimmer_MIN	Minimum value of the APQ5 shimmer
aqpq5Shimmer_MAX	Maximum value of the APQ5 shimmer
aqpq5Shimmer_RANGE	Range of the APQ5 shimmer
aqpq5Shimmer_KURTOSIS	Kurtosis of the APQ5 shimmer
aqpq5Shimmer_SKEWNESS	Skewness of the APQ5 shimmer
apq11Shimmer_MEAN	Mean of the APQ11 shimmer
apq11Shimmer_STD	Standard deviation of the APQ11 shimmer
apq11Shimmer_MIN	Minimum value of the APQ11 shimmer
apq11Shimmer_MAX	Maximum value of the APQ11 shimmer
apq11Shimmer_RANGE	Range of the APQ11 shimmer
apq11Shimmer_KURTOSIS	Kurtosis of the APQ11 shimmer
apq11Shimmer_SKEWNESS	Skewness of the APQ11 shimmer
ddaShimmer_MEAN	Mean of the DDA shimmer
ddaShimmer_STD	Standard deviation of the DDA shimmer
ddaShimmer_MIN	Minimum value of the DDA shimmer
ddaShimmer_MAX	Maximum value of the DDA shimmer
ddaShimmer_RANGE	Range of the DDA shimmer
ddaShimmer_KURTOSIS	Kurtosis of the DDA shimmer
ddaShimmer_SKEWNESS	Skewness of the DDA shimmer
mfccs_1_MEAN	Mean of the 1st MFCC
mfccs_1_STD	Standard deviation of the 1st MFCC
mfccs_1_MIN	Minimum value of the 1st MFCC
mfccs_1_MAX	Maximum value of the 1st MFCC

Feature Name	**Description**
mfccs_1_RANGE	Range of the 1st MFCC
mfccs_1_KURTOSIS	Kurtosis of the 1st MFCC
mfccs_1_SKEWNESS	Skewness of the 1st MFCC
mfccs_2_MEAN	Mean of the 2nd MFCC
mfccs_2_STD	Standard deviation of the 2nd MFCC
mfccs_2_MIN	Minimum value of the 2nd MFCC
mfccs_2_MAX	Maximum value of the 2nd MFCC
mfccs_2_RANGE	Range of the 2nd MFCC
mfccs_2_KURTOSIS	Kurtosis of the 2nd MFCC
mfccs_2_SKEWNESS	Skewness of the 2nd MFCC
mfccs_3_MEAN	Mean of the 3rd MFCC
mfccs_3_STD	Standard deviation of the 3rd MFCC
mfccs_3_MIN	Minimum value of the 3rd MFCC
mfccs_3_MAX	Maximum value of the 3rd MFCC
mfccs_3_RANGE	Range of the 3rd MFCC
mfccs_3_KURTOSIS	Kurtosis of the 3rd MFCC
mfccs_3_SKEWNESS	Skewness of the 3rd MFCC
mfccs_4_MEAN	Mean of the 4th MFCC
mfccs_4_STD	Standard deviation of the 4th MFCC
mfccs_4_MIN	Minimum value of the 4th MFCC
mfccs_4_MAX	Maximum value of the 4th MFCC
mfccs_4_RANGE	Range of the 4th MFCC
mfccs_4_KURTOSIS	Kurtosis of the 4th MFCC
mfccs_4_SKEWNESS	Skewness of the 4th MFCC
mfccs_5_MEAN	Mean of the 5th MFCC
mfccs_5_STD	Standard deviation of the 5th MFCC
mfccs_5_MIN	Minimum value of the 5th MFCC
mfccs_5_MAX	Maximum value of the 5th MFCC
mfccs_5_RANGE	Range of the 5th MFCC
mfccs_5_KURTOSIS	Kurtosis of the 5th MFCC
mfccs_5_SKEWNESS	Skewness of the 5th MFCC
mfccs_6_MEAN	Mean of the 6th MFCC
mfccs_6_STD	Standard deviation of the 6th MFCC
mfccs_6_MIN	Minimum value of the 6th MFCC
mfccs_6_MAX	Maximum value of the 6th MFCC
mfccs_6_RANGE	Range of the 6th MFCC
mfccs_6_KURTOSIS	Kurtosis of the 6th MFCC
mfccs_6_SKEWNESS	Skewness of the 6th MFCC
mfccs_7_MEAN	Mean of the 7th MFCC
mfccs_7_STD	Standard deviation of the 7th MFCC
mfccs_7_MIN	Minimum value of the 7th MFCC
mfccs_7_MAX	Maximum value of the 7th MFCC
mfccs_7_RANGE	Range of the 7th MFCC
mfccs_7_KURTOSIS	Kurtosis of the 7th MFCC

Feature Name	Description
mfccs_7_SKEWNESS	Skewness of the 7th MFCC
mfccs_8_MEAN	Mean of the 8th MFCC
mfccs_8_STD	Standard deviation of the 8th MFCC
mfccs_8_MIN	Minimum value of the 8th MFCC
mfccs_8_MAX	Maximum value of the 8th MFCC
mfccs_8_RANGE	Range of the 8th MFCC
mfccs_8_KURTOSIS	Kurtosis of the 8th MFCC
mfccs_8_SKEWNESS	Skewness of the 8th MFCC
mfccs_9_MEAN	Mean of the 9th MFCC
mfccs_9_STD	Standard deviation of the 9th MFCC
mfccs_9_MIN	Minimum value of the 9th MFCC
mfccs_9_MAX	Maximum value of the 9th MFCC
mfccs_9_RANGE	Range of the 9th MFCC
mfccs_9_KURTOSIS	Kurtosis of the 9th MFCC
mfccs_9_SKEWNESS	Skewness of the 9th MFCC
mfccs_10_MEAN	Mean of the 10th MFCC
mfccs_10_STD	Standard deviation of the 10th MFCC
mfccs_10_MIN	Minimum value of the 10th MFCC
mfccs_10_MAX	Maximum value of the 10th MFCC
mfccs_10_RANGE	Range of the 10th MFCC
mfccs_10_KURTOSIS	Kurtosis of the 10th MFCC
mfccs_10_SKEWNESS	Skewness of the 10th MFCC
mfccs_11_MEAN	Mean of the 11th MFCC
mfccs_11_STD	Standard deviation of the 11th MFCC
mfccs_11_MIN	Minimum value of the 11th MFCC
mfccs_11_MAX	Maximum value of the 11th MFCC
mfccs_11_RANGE	Range of the 11th MFCC
mfccs_11_KURTOSIS	Kurtosis of the 11th MFCC
mfccs_11_SKEWNESS	Skewness of the 11th MFCC
mfccs_12_MEAN	Mean of the 12th MFCC
mfccs_12_STD	Standard deviation of the 12th MFCC
mfccs_12_MIN	Minimum value of the 12th MFCC
mfccs_12_MAX	Maximum value of the 12th MFCC
mfccs_12_RANGE	Range of the 12th MFCC
mfccs_12_KURTOSIS	Kurtosis of the 12th MFCC
mfccs_12_SKEWNESS	Skewness of the 12th MFCC
mfccs_13_MEAN	Mean of the 13th MFCC
mfccs_13_STD	Standard deviation of the 13th MFCC
mfccs_13_MIN	Minimum value of the 13th MFCC
mfccs_13_MAX	Maximum value of the 13th MFCC
mfccs_13_RANGE	Range of the 13th MFCC
mfccs_13_KURTOSIS	Kurtosis of the 13th MFCC
mfccs_13_SKEWNESS	Skewness of the 13th MFCC

Bibliography

[1] H. H. Speech production entails a three-level process: cognitive planning level, physiological articulation level, and acoustic level. https://www.researchgate.net/figure/Speech-production-entails-a-three-level-process-cognitive-planning-level-physiological_
fig1_371013471, 2017. Accessed: 2024-08-06.

[2] World Health Organization. World health organization predictions on renal failure. *WHO Global Health Estimates*, 2024. Retrieved from WHO website.

[3] X. Hong et al. Standardized scales in renal failure diagnosis. *Journal of Nephrology*, 30:123–134, 2023.

[4] J. Mun et al. Glottal features for automated detection of ckd. *Journal of Biomedical Engineering*, 45:567–578, 2023.

[5] Ckd speech database. Accessed from CKD Research Database, 2023.

[6] Y.-H. Lien, C.-H. Lee, S.-Y. Liao, T.-Y. Hsiao, and C.-H. Lin. Analysis of voice quality and its relationship to depression. *Journal of Voice*, 22(5):562–570, 2008.

[7] E. Mendoza, R. Esquivel, and A. Montoya. Voice and speech signal analysis for detecting depression: A review. *Biological Psychology*, 135:128–140, 2018.

[8] I. McCowan and M. P. Schuller. On the use of spectral features for speech analysis. *IEEE Transactions on Speech and Audio Processing*, 13(2):252–261, 2005.

[9] G. E. Hinton and R. R. Salakhutdinov. *Reducing the dimensionality of data with neural networks*, volume 313. Science, 2012.

[10] J. E. Cahn. *The effects of depression on prosody*, volume 87. Journal of the Acoustical Society of America, 1990.

[11] B. W. Schuller, S. Steidl, and A. Batliner. The interspeech 2013 computational paralinguistics challenge: Nativeness, cognitive load, and affect. In *Proceedings of INTERSPEECH*, pages 1–5, 2013.

[12] Björn Schuller, Stefan Steidl, Anton Batliner, and Felix Burkhardt. The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language. In *Interspeech*, 2016.

[13] Florian Eyben, Martin Wöllmer, Björn Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Devillers, Julien Epps, Petri Laukka, Shrikanth Narayanan, Khiet Truong, Saeid Vaseghi, Laurence Vidrascu, Wendi Lauw, and Maja Pantic. opensmile: the munich versatile and fast open-source audio feature extractor. *ACM Multimedia*, 1(1):145–149, 2010.

[14] Jan Hlavnička, Jirí Mekyska, Zdenek Galaz, Marcos Faundez-Zanuy, Zdenek Smékal, Ilona Eliasova, Ondrej Krejcar, and Irena Rektorova. Voice disorders: A comprehensive survey of the current techniques and methodologies. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(9):1325–1338, 2017.

[15] Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, Oriol Nieto, Thanos Giannakopoulos, Sean I. O'Keefe, Zhiyao Duan, et al. librosa: Audio and music signal analysis in Python. *Proceedings of the 14th Python in Science Conference*, 2015:18–25, 2015.

[16] R.C. Scherer. A model and scaling strategy for the perceptual dimensions of speech. *Language and Speech*, 38(1):5–21, 1995.

[17] K. Schraut and A. Smith. Vowel trimming algorithm for speech processing. *Speech Technology Journal*, 15(2):112–125, 2024.

[18] T. Wang, M. Kim, and A. Artemiou. Per-channel energy normalization: Why and how. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2396–2400, 2018.

[19] J. Doe and J. Smith. Log-mel spectrogram analysis in medical applications. *Medical Engineering Journal*, 12(4):321–335, 2023.

[20] J. Brown and M. White. Applications of log-mel spectrograms in biomedical signal processing. In *Biomedical Signal Processing Symposium*, pages 185–196, 2022.

[21] Vladimir Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[22] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[23] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2002.

[24] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[25] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[26] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[27] Gary James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning.* Springer, New York, 2013.

[28] Charité Universitätsmedizin Berlin. Telemed5000 project. `https://telemedizin.charite.de/en/research/telemed5000/`, 2024. Accessed: 2024-09-04.

[29] International Telecommunication Union. *Algorithms to measure audio programme loudness and true-peak audio level.* ITU-R BS.1770. ITU, 2012.

[30] Loudnorm Development Team. *pyloudnorm.* Software.

[31] M. R. Schroeder and B. S. Atal. Speech signal processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(1):1–12, 2011.

[32] J. O. Smith. *Spectral Audio Signal Processing.* Wiley, 1987.

[33] K. N. Stevens. *Acoustic Phonetics.* MIT Press, 2012.

[34] J. B. Allen. *The Vowel Space.* Journal of the Acoustical Society of America, 1977.

[35] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[36] V. Vijayakumar and D. Murphy. Robust speech recognition using majority voting. *IEEE Transactions on Speech and Audio Processing*, 16(3):582–591, 2008.

[37] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated Machine Learning: Methods, Systems, Challenges.* Springer, 2019.

[38] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[39] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

[40] H. Jin, L. Song, and X. Yao. Auto-sklearn: Efficient and robust automated machine learning. *Journal of Statistical Software*, 77(1):1–38, 2017.