signals

MDPI

Article

Machine Learning with Evolutionary Parameter Tuning for Singing Registers Classification

Tales Boratto [1], Gabriel de Oliveira Costa [2], Alexsandro Meireles [3], Anna Klara Sá Teles Rocha Alves [4], Camila M. Saporetti [5], Matteo Bodini [6,*], Alexandre Cury [7] and Leonardo Goliatt [7]

1 Graduate Program in Computational Modeling, Federal University of Juiz de Fora, Juiz de Fora 36036-900, MG, Brazil; tales.boratto@estudante.ufjf.br
2 Department of Mechatronics Engineering, Federal Institute of Southeast Minas Gerais, Juiz de Fora 36080-001, MG, Brazil; gabrieloliveira@grupocsc.com.br
3 Department of Languages and Literature, Federal University of Espírito Santo, Vitória 29075-910, ES, Brazil; meirelesalex@gmail.com
4 Graduate Program in Nursing, Federal University of Juiz de Fora, Juiz de Fora 36036-900, MG, Brazil; alves.anna@estudante.ufjf.br
5 Department of Computational Modeling, Polytechnic Institute, Rio de Janeiro State University, Nova Friburgo 22000-900, RJ, Brazil; camila.saporetti@iprj.uerj.br
6 Dipartimento di Economia, Management e Metodi Quantitativi, Università degli Studi di Milano, Via Conservatorio 7, 20122 Milano, Italy
7 Department of Computational and Applied Mechanics, Federal University of Juiz de Fora, Juiz de Fora 36036-900, MG, Brazil; alexandre.cury@ufjf.br (A.C.); leonardo.goliatt@ufjf.br (L.G.)
* Correspondence: matteo.bodini@unimi.it

Abstract: Behind human voice production, a complex biological mechanism generates and modulates sound. Recent research has explored machine-learning (ML) techniques to analyze singing-voice characteristics. However, the classification efficiency reported in such research works suggests the possibility of improvement. In addition, there is also scope for further improvement through the application of still under-utilized optimization techniques. Thus, the present article proposes a novel approach that leverages the Differential Evolution (DE) algorithm to optimize hyperparameters within three selected ML models, with the aim of classifying singing-voice registers i.e., chest, mixed, and head registers). To develop the present study, a dataset of 350 audio files encompassing the three aforementioned registers was constructed. Then, the TSFEL Python library was employed to extract 14 pieces of temporal information from the audio signals for subsequent classification by the employed ML models. The obtained findings demonstrated that the Extreme Gradient Boosting model, optimized with DE, achieved an average classification accuracy of 97.60%, thus indicating the efficacy of the proposed approach for singing-voice register classification.

Keywords: singing registers; machine learning; differential evolution; classification; optimization

Academic Editors: Alexander Kocian and Constantine Kotropoulos

Received: 31 December 2024
Revised: 4 February 2025
Accepted: 18 February 2025
Published: 21 February 2025

Citation: Boratto, T.; Costa, G.d.O.; Meireles, A.; Alves, A.K.S.T.R.; Saporetti, C.M.; Bodini, M.; Cury, A.; Goliatt, L. Machine Learning with Evolutionary Parameter Tuning for Singing Registers Classification. *Signals* **2025**, *6*, 9. https://doi.org/10.3390/signals6010009

1. Introduction

Behind the apparent simplicity of the sound of the human voice lies an intricate interplay of resonances that both mold and define the unique quality of each voice. According to [1], the human phonation system is made up of three parts: (I) the respiratory system, (II) the pair of vocal folds, and (III) a system of cavities, i.e., the vocal tract. Since the respiratory system performs the function of a compressor by compressing the air stored in the lungs, only the last two mentioned parts contribute directly to the formation of the

vocal timbre. Indeed, the pair of vocal folds are responsible for generating sound through vibrations that interrupt the airflow from the lungs in a sequence of air pulses.

The vocal tract includes the oral and nasal cavities, the pharynx, and the larynx, which contains the vocal folds. In particular, the larynx is divided into glottis, supraglottis, and subglottis. Moreover, the vocal tract plays the role of a resonance chamber or filter that molds the sound generated by the vocal folds. Figure 1 illustrates the latter voice production process in detail. The air coming from the lungs provides the energy needed for the vocal folds to vibrate. This periodic vibration generates a spectrum of fundamental and harmonic frequencies. The vocal tract then filters the excitation signal produced by the vocal folds. The interaction between the spectrum of the sound source and the transfer function of the vocal tract produces the spectrum of the radiated sound, which is the sound actually perceived by the listener (speech signal). The relative amplitudes of the different frequencies in the radiated spectrum determine the acoustic characteristics of the sound, such as timbre and intensity.

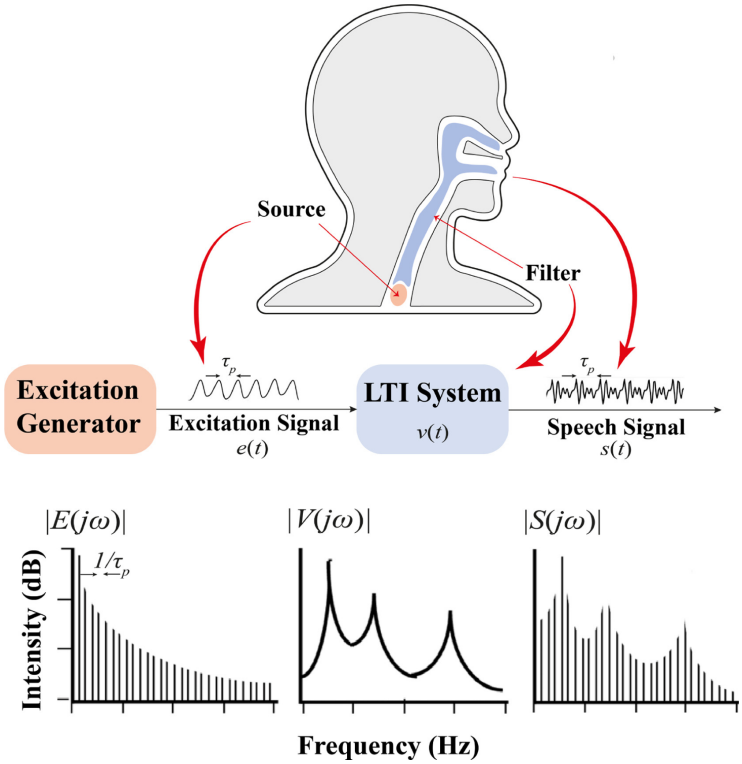


Figure 1. The source-filter model of speech production consists of the glottis as the excitation source and the vocal tract (nasal and oral cavities) as the filter. Furthermore, the temporal and spectral representations of the source, vocal tract, and resulting speech signal are represented. Reprinted from Almaghrabi et al. [2], Copyright (2023), with permission from Elsevier.

Relying on the above-described process, it is possible to adjust the formation of vocal acoustics. Indeed, the act of altering the shape of the vocal tract, known as articulation, makes it possible to control the formants of speech since the movement of the articulators can modify the acoustic resonances of the vocal tract (with articulators, it is referred to the structures used to organize the shape of the vocal tract in different ways, namely: pharynx, velum, tongue parts, and lips). Similarly, specific kind of adjustments can be made by activating the laryngeal muscles, which, in a way, can control the geometry and mechanical properties of the vocal folds, thus making it possible to control or modify the voice itself [1,3].

Vocal folds, in turn, exhibit a non-linear, anisotropic shape and viscoelastic behavior. In particular, the stress-strain curve shown in Figure 2 illustrates the non-linear mechanical behavior typical of the anteroposterior direction of vocal folds.

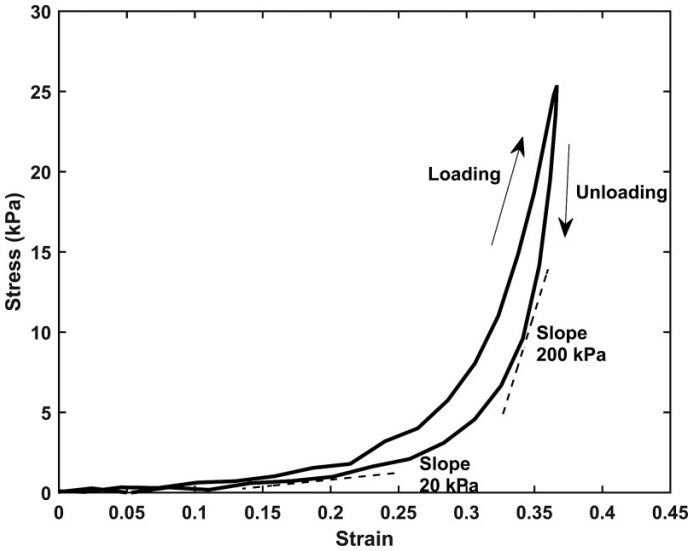
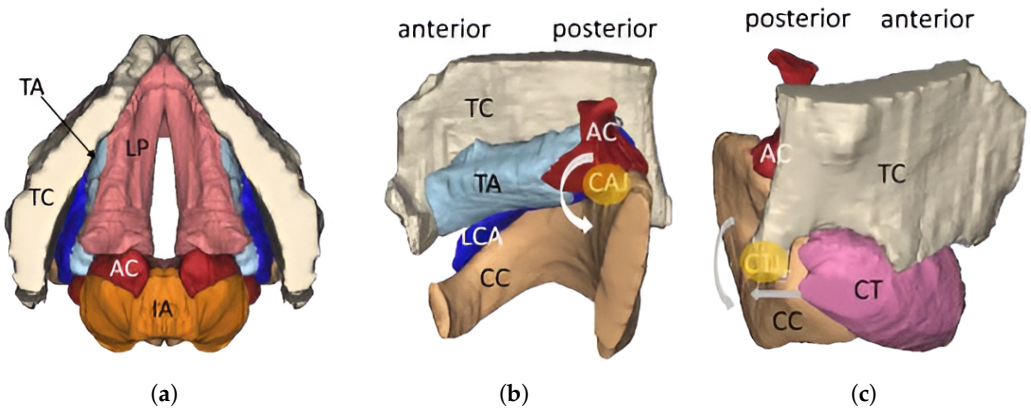


Figure 2. The figure reports a typical tensile stress-strain curve for the vocal fold along the anterior-posterior direction, measured during loading and unloading at a frequency of 1 Hz. The slope of the tangent line (represented by dashed lines) to the stress-strain curve indicates the tangent stiffness. Due to the viscous nature of the vocal folds, the stress is generally higher during loading than unloading. This curve was derived by averaging data over 30 cycles following a 10-cycle preconditioning. Reprinted with permission from Zhang [3]. Copyright 2016, Acoustical Society of America.

The above-reported behavior provides a means for regulating the stiffness and tension of the vocal folds by lengthening or shortening them, which plays an important role in controlling the fundamental frequency (f_0) or pitch of voice production. In particular, the two laryngeal muscles involved in regulating the length of the vocal folds are the cricothyroid muscle (CT) and the thyroarytenoid muscle (TA), whose activation also alters the shape of the medial surface of the vocal folds and the geometry of the glottal canal [3]. In addition, Meireles and Mixdorff [4] stated that the vocal folds are adducted at their posterior portion by the inter-arytenoid muscle (IA), abducted at their posterior portion by the posterior cricoarytenoid muscle (PCA), and adducted at their posterior portion by the lateral cricoarytenoid muscle (LCA), which also controls f_0, intensity, and register. Figure 3 shows all the latter muscles involved in voice production.

Regarding the signal-based representation of speech, Kent and Read [5] (p. 9) stated that "the acoustic signal of speech is the physical event transmitted in telecommunications or recorded on magnetic tape, laser disc, or other mediums". Such a signal contains both the linguistic message of the conveyed speech and all the information needed to represent the human voice properly. Within the latter context, the source-filter theory developed by Fant [6] is the usual choice when analyzing speech acoustics. Indeed, according to Fant's theory [6], a simplified version of the vocal tract, i.e., the tube model, can represent the acoustics of speech. In particular, the vibrating vocal folds act as the signal source of sound, and the articulators of the vocal tracts, acting as filters, change the sound resonances. Moreover, in the context of speech, each vowel has a unique filter that modifies the source sound. Such filters act as frequency selectors, producing different resonance frequencies known as formants in speech technology literature. Finally, the same theoretical framework can be applied to analyze the singing voice [7]. Indeed, in the present article, Fant's theory

was leveraged to analyze the singing voice since singing and speaking can be explained by the same underlying acoustic theory.



(**a**) (**b**) (**c**)

Figure 3. Illustration of the muscles involved in voice production and control: (**a**) superior view of the vocal folds, including the cartilaginous framework and laryngeal muscles; (**b**) medial view of the cricoarytenoid joint, which is formed between the arytenoid and cricoid cartilages; (**c**) posterolateral view of the cricothyroid joint, formed by the thyroid and cricoid cartilages. The arrows in (**b**,**c**) show the possible directions of movement of the arytenoid and cricoid cartilages due to the activation of the LCA and CT muscles, respectively. Reprinted with permission from Zhang [3]. Copyright 2016, Acoustical Society of America.

Vocal resonance is a fundamental acoustic phenomenon in singing, influenced both by anatomical characteristics of the vocal tract and by the technique employed by the singer. Recognizing different types of vocal registers can provide valuable information for applications in areas such as vocal training, diagnosis of vocal conditions, and musicological analysis. However, due to the complexities involved, such as individual variations between singers, identifying and classifying resonance patterns presents significant challenges.

With the modernization of technology and advances in computing, it has become possible to process large amounts of data and identify complex patterns with high precision. Thus, in addition to techniques that can capture temporal dependencies in time series data, such as autoregressive models [8,9], machine-learning methods have been gaining prominence because they make it possible to find certain patterns in signals that carry information that would be difficult to perceive beyond machines. As a result, the study of singing records has become relevant in many application contexts, for example, in vocal pedagogy [10], vocal health [11], and music technology [12]. In addition, it is also possible to use the sound of the voice to help in the medical diagnosis of certain diseases, such as COVID-19 [13], Alzheimer [14], and autism [15].

Thus, still in the context of ML algorithms leveraged to detect specific nuances in singing voices, Xu et al. [10] evaluated the openSMILE feature set with Signal Feature space Support Vector Machine (SF-SVM), End-to-End Deep Learning (E2EDL), and deep embedding with Support Vector Machine (SVM) to classify audio files according to seven paralinguistic singing characteristics commonly used in vocal pedagogy. The authors built a singing-voice quality and technique database consisting of processed audio clips downloaded from YouTube, annotated with their respective labels to achieve the latter aim. The experimental results showed that it was possible to reach an Unweighted Average Recall (UAR) between 40% and 47%, relying on the deep embedding with the SVM algorithm and the Residual neural network (ResNet) architecture as the feature extractor. Wang et al. [11] collected a database consisting of 2004 audio files from professional opera singers to provide singers early guidance improved by ML models, thereby averting probable phonomicrosurgery and/or premature end to their

professional careers. To achieve the latter goal, the Authors employed the Random Forest (RF) model to construct an efficient Fach classifier (the German Fach system categorizes singers, especially opera performers, based on their vocal range, weight, and timbre. While it is employed globally, it is particularly prevalent in the context of Europe, in particular within German-speaking regions and repertory opera houses [16]). Thus, it was possible to achieve an accuracy of almost 80% in most of the examined voice types. Sha et al. [12] used the SVM model, tuned with the radial basis function (RBF) kernel, to classify Chinese popular music based on six timbre qualities of the singing voice. The authors could achieve an accuracy of 79.84% in the timbre classification task, considering the usage of five-fold stratified cross-validation as a performance assessment strategy.

Although research exploring ML techniques to detect specific nuances in singing voices is found in the literature, this approach remains relatively under-researched. In addition, the reported classification performance and computational efficiency of the previously reported research suggest potential improvement. Complementarily, to the best of our knowledge, existing studies have not evaluated the efficacy of hybrid models optimized by metaheuristics for the considered task, as has been done in investigations with similar contexts [17]. These gaps in the available literature motivate our investigation into the utilization of hybrid models optimized by metaheuristics for singing-voice analysis.

The performance of ML models depends heavily on the choice of hyperparameters, once it influences the model's ability to learn patterns in the data [18–20]. In addition, tuning a model for optimal performance is a challenging and time-intensive process, especially when many hyperparameters need adjustment, as this necessitates exploring a vast number of possible combinations [21]. Furthermore, manually adjusting these hyperparameters or through exhaustive search can be inefficient and not guarantee the best configuration. Thus, bio-inspired optimization algorithms, such as the Differential Evolution (DE), have stood out for handling this task well in different scenarios. Thus, such an approach can offer superior performance in identifying subtle characteristics within singing voices compared to existing methods [20].

Furthermore, this work also sought to optimize the computational efficiency of the proposed model to ensure its practical applicability in real-world scenarios. Thus, we chose to use low-level audio descriptors and, consequently, low computational cost, combined with traditional machine-learning models since Deep Learning (DL) models can have a significantly higher computational cost [22].

Therefore, the main objective of this work is to evaluate the SVM, Multilayer Perceptron (MLP), and eXtreme Gradient Boosting (XGB) models combined with the DE algorithm for optimizing hyperparameters in the context of classifying three distinct singing registers based solely on their sounds. To achieve this purpose, the first planned step was to build a database made up of 350 6-s audio files containing the sounds of three different singing registers performed by an experienced singer, i.e., chest voice, mixed voice, and head voice (for additional details, refer to Miller [23]). The latter three registers correspond to the voice resonances associated with low frequencies (chest voice), middle frequencies (middle voice), and high frequencies (head voice). Then, the Python Time Series Feature Extraction Library (TSFEL) [24] was leveraged to extract 14 features in the time domain from each of the considered signals. After constructing the corresponding feature matrix, the selected three ML models were optimized by the DE algorithm, and evaluated using five classification performance metrics.

Relying on the above-presented experimental strategy, the main contributions of the present article are summarized here below as follows:

- Building a database consisting of audio files collected from three singing registers in a controlled scenario;
- Proposing the usage of hybrid ML models, optimized by metaheuristics, and low-level audio descriptors for singing-voice classification;
- Assessing three different hybrid ML methods, i.e., SVM, MLP, and XGB, for classifying different singing registers via their collected sounds;
- Showing that optimizing the hyperparameters of ML models using DE algorithms can be a proper alternative for adjusting the internal parameters of the leveraged ML classifiers.

The structure of the forthcoming part of the article is organized as follows: The next Section 2 presents the issue of categorizing singing registers, the process of audio recording, the features retrieved from the audio signals, the employed experimental data, and the ML models' implementation; The obtained results and their respective analysis, together with an evaluation of the ML models that includes a parametric analysis, feature importance analysis, and a discussion of the models' advantages and disadvantages, are presented in Section 3; Finally, Section 4 reports the concluding discussion and findings for the carried study.

2. Materials and Methods

This section outlines the foundational framework for the model development presented in the current study. First, the data acquisition process is described in detail, including database construction and feature extraction methods for audio signals (Sections 2.1 and 2.2). The ML models employed in the present research—Multilayer Perceptron Neural Network (Section 2.4), Support Vector Classifier (Section 2.3), and Extreme Gradient Boosting (Section 2.5)—are then introduced, with a focus on their mathematical formulation and parameter determination. A comprehensive explanation of the DE algorithm (Section 2.6) is provided, highlighting its critical role in optimizing hyperparameters for the employed ML models. Additionally, the SHapley Additive exPlanations (SHAP) method (Section 2.7) is discussed to evaluate the interpretability of model predictions. Finally, the computational framework is presented (Section 2.8), which develops the search for optimal internal parameters of the ML models as an optimization problem.

2.1. Data Acquisition

The audio acquisition process was conducted by an experienced singer with mastery of the three types of singing register selected for the performance: chest voice (Register 1), mixed voice (Register 2), and head voice (Register 3). The choice of the latter three types of register is justified by the fact that, in general, there is an audible difference between them, as well as by the fact that they are less uncomfortable to perform since the amount of execution carried out for each type can become tiring [25].

For recording a clear execution of the singing registers, the vowel "a" was defined to be sung for a period of 6 s and in the following register order: Register 2, Register 3, and Register 1. For the sound capture, it was used a Behringer Xenyx Q1202USB sound mixer—additional information available on [26]—configured with the gain knob set to zero and an Audio Technica AT2020 cardioid microphone—additional information available on [27]— positioned approximately 10 cm from the singer's face. In addition, the REAPER software was used as a digital audio workstation—version 6.73, developed by Cockos Incorporated (Rosendale, NY, USA), available at: https://www.reaper.fm (accessed on 17 February 2025).

2.2. Dataset

The number of samples and the features to be used in this work were previously defined during the audio signal-capture planning stage, based on the process of acquiring the sound of drum cymbals carried out by Boratto et al. [17]. In this paper, the authors collected a total of 276 audios from 4 cymbals and extracted temporal attributes from the signals, using them as input variables for the ML model used. The high classification performance achieved (around 97%) shows that both the process to capture the audio signals and the time domain features used effectively solved the problem. In this sense, it was decided to collect a minimum of 100 samples for each type of vocal register. This number was established to mitigate limitations in the process, such as variations in the recording conditions (environment, equipment, etc.) and the number of experienced singers available. In addition, it was expected that a greater number of repetitions would increase the variability of the audio since the vocal musculature would be progressively demanded, making it more difficult to maintain the execution of the vocal registers.

Consequently, acquiring audio samples resulted in constructing a database of 350 audio signals lasting 6 s, rendered at 44,100 Hz in .wav audio format. The division of audio files for each voice register consisted of 110 files for Register 2 (mixed voice), 136 for Register 3 (head voice), and 104 for Register 1 (chest voice). Moreover, in the present study, ML models' classification performance was assessed in the case where input features were low-level audio descriptors, i.e., temporal information retrieved from each signal. In particular, a total amount of 14 features, reported in Table 1, was extracted with the aid of the Python TSFEL (Time Series Feature Extraction Library) library [24].

Table 1. Temporal features description computed with the Python TSFEL library, employed in the present study to retrieve information from the collected audio signals. The theoretical description of each reported feature may be retrieved from Bhattacharyya et al. [28].

	Feature Descriptions		**Feature Descriptions**
X_1	Area under the signal waveform.	X_8	No. negative signal turning points.
X_2	Signal autocorrelation.	X_9	No. peaks from a defined signal neighborhood.
X_3	Time axis centroid.	X_{10}	No. positive signal turning points.
X_4	Mean absolute signal differences.	X_{11}	Signal traveled distance.
X_5	Mean of the signal differences.	X_{12}	Signal Slope.
X_6	Median absolute differences of the signal.	X_{13}	Sum of absolute signal differences.
X_7	Median of differences of the signal.	X_{14}	Signal Zero-crossing rate.

2.3. Support Vector Classifier

Support Vector Classifiers (SVC) are powerful classification models usually leveraged in the context of ML, initially introduced by Vapnik [29]. SVC can effectively learn non-linear decision boundaries in high-dimensional feature spaces by finding a hyperplane that maximizes the margin between the data points of different classes [30]. The latter capability is achieved by identifying a subset of training data points, called support vectors, that define the optimal hyperplane.

Formally, given a training set $T = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i$ is a data point and y_i its corresponding class label, an SVC solves the optimization problem showed by Equation (1):

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi, \quad C > 0 \tag{1}$$

$$\text{subject to} \quad y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i > 0 \tag{2}$$

where $\phi(\mathbf{x}_i)$ maps the input vector $\mathbf{x}_i$ to a higher-dimensional space, C is a regularization parameter controlling the trade-off between maximizing the margin and minimizing training errors, and ξ_i represents the slack variables allowing for misclassified points. Due to the potentially high dimensionality of $\mathbf{w}$, the problem is typically solved in its dual form according to the following Equation (3):

$$\min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^T\mathbf{Q}\mathbf{u} - \mathbf{1}^T\mathbf{u} \tag{3}$$

$$\text{subject to} \quad \mathbf{y}^T\mathbf{u} = 0, \quad 0 \leq u_i \leq C \tag{4}$$

where $\mathbf{1}$ is a vector of ones, $\mathbf{Q}$ is a positive semi-definite kernel matrix defined as $Q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and $K(\mathbf{x}_i, \mathbf{x}_j)$ represents a kernel function that computes the similarity between data points. Standard kernel functions include linear, RBF, and sigmoid (Table 2). The relationship between the weight vector $\mathbf{w}$ and the dual solution $\mathbf{u}$ is given by Equation (5):

$$\mathbf{w} = \sum_{i=1}^{N} y_i u_i \phi(\mathbf{x}_i) \tag{5}$$

Table 2. Kernel functions, where γ represents the kernel parameter. It must be noted that the choice of kernel function usually impacts the decision boundary learned by the SVM in a significant way.

Kernel	**Mathematical Formulation $K(\mathbf{x_i}, \mathbf{x_j})$**
Linear	$\mathbf{x}_i^T\mathbf{x}_j$
Sigmoid	$\tanh(\mathbf{x}_i^T\mathbf{x}_j + 1)$
RBF	$\exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$

2.4. Multilayer Perceptron Neural Network

The MLP is a ubiquitous feed-forward artificial neural network architecture extensively used in supervised learning tasks. In particular, the MLP adopts a layered structure consisting of input, hidden, and output layers. The interconnection between neurons within the latter layers is established through several weights [31]. First, the input data are received by neurons in the input layer. Then, hidden layer neurons process their inputs or outputs from the prior layer using non-linear activation functions, selected from the linear function ($\varphi(x) = x$), the sigmoid logistic function ($\varphi(x) = 1/(1 + e^{-x})$), hyperbolic tangent ($\varphi(x) = tanh(x)$), or rectified linear unit ($\varphi(x) = \max(0, x)$) [32].

The backpropagation algorithm, the core of the learning process, computes the gradient of the loss function in the output layer [33]. In particular, the latter algorithm facilitates iterative adjustments to network weights, enabling the model to minimize the loss function progressively. Moreover, it must be noted that for a dataset with dimensions $m \times n$, the input layer represents input features as a feature vector $\{x_i | x_1, \ldots, x_n\}$. Then, each hidden layer neuron performs a linear combination of the previous layer's outputs $[x_1, x_2, \ldots, x_n]^T$ with corresponding weights $[w_1, w_2, \ldots, w_n]^T$. The latter can be mathematically expressed as $w_1x_1 + w_2x_2 + \ldots + w_nx_n$.

The network's signals are then passed through the hidden-to-output activation function, and the outputs are calculated to the output layer, which generates predictions. Figure 4 illustrates the above-introduced architecture, for example, of an MLP composed of two hidden layers [34].

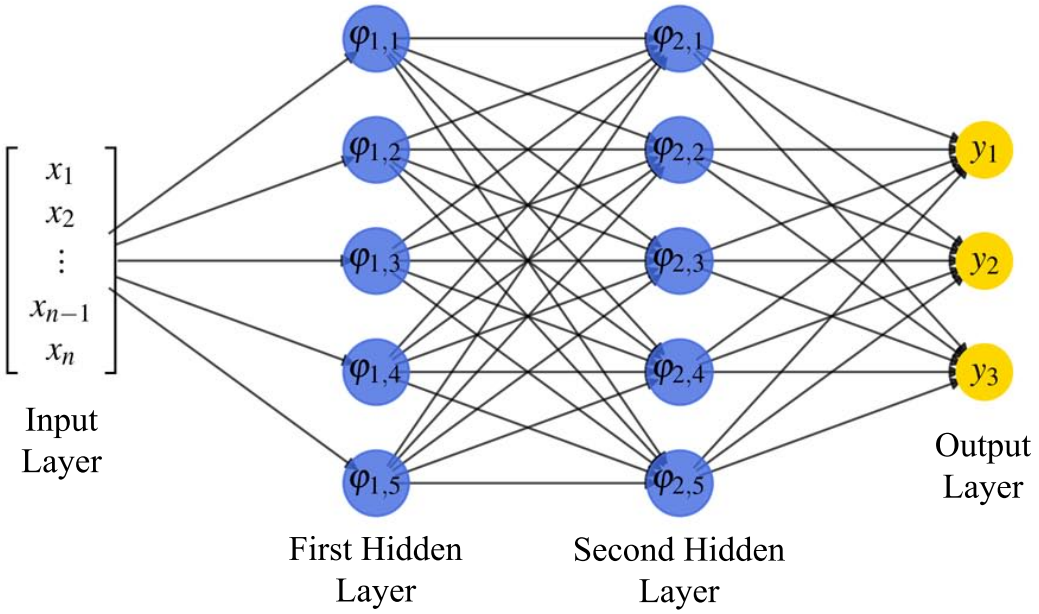


Figure 4. Illustration of the architecture of an MLP with two hidden layers composed of 5 neurons. The activation function is represented by φ. The diagram shows the flow of information from the input layer through the first and second hidden layers to the output layer, highlighting the fully connected structure of the network.

2.5. Extreme Gradient Boosting

The XGB model is a high-performance implementation of gradient boosting for supervised learning tasks [35]. In particular, the latter leverages an ensemble of decision trees, where each employed tree progressively refines the predictions of the prior trees by focusing on the errors of previous models [36].

Formally, consider a dataset with m features and n samples denoted as follows: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where $\mathbf{x}_i \in \mathbb{R}^m$ represents the feature vector for the i-th sample and $y_i \in \mathbb{R}$ represents the corresponding label. The ensemble learning approach in XGB aims to minimize a selected loss function $L(\phi)$ defined over the entire dataset. In particular, the core principle of XGB lies in iteratively building an ensemble model $\hat{y}_i = \phi_M(\mathbf{x}_i)$ through a sequence of decision trees $f_k(\mathbf{x}_i)$, where M represents the number of trees in the ensemble. Each of the latter trees learns from the residual errors of the previous trees, aiming to improve the overall predictive performance. Mathematically, the latter additive strategy can be expressed according to the following Equation (6):

$$\phi_M(\mathbf{x}_i) = \phi_{M-1}(\mathbf{x}_i) + \eta f_n(\mathbf{x}_i) \tag{6}$$

where η represents the learning rate that controls the impact of each new tree on the ensemble. It must be noted that the decision trees themselves are usually constrained to a maximum depth of m_{depth} to prevent the occurrence of overfitting.

The loss function $L(\phi)$ typically includes the following two key components:

1. Data Loss: This term measures the discrepancy between the predicted outputs, $\hat{y}_i$, and the actual labels, y_i. A common choice for the latter one is the absolute loss function, defined as $l(y_i, \hat{y}_i) = |\hat{y}_i - y_i|$.
2. Model Complexity Penalty: This term, often referred to as regularization, discourages overly complex models that might lead to overfitting. In the context of XGB, an L_2 regularization term, expressed as $\lambda||\mathbf{w}||^2$, is frequently employed, where λ controls the strength of the penalty and $\mathbf{w}$ represents the weights associated with each leaf node in the employed decision trees.

Through the minimization of the loss $L(\phi)$ with the iterative construction of the ensemble, XGB achieves enhanced prediction accuracy within supervised learning tasks.

2.6. Differential Evolution

DE is a population-based stochastic search method for global optimization problems, which mimics the concept of natural evolution [37]. In particular, a population of candidate solutions evolves iteratively to improve their fitness according to a chosen objective function. The current work relied on such an approach to optimize the hyperparameters for SVC, MLP, and XGB ML classifiers. The DE algorithm operates in six steps:

1. Population initialization: An initial population of NP candidate solutions is created, which is denoted by $\{\boldsymbol{\theta}_i^G \mid i = 1, 2, \ldots, NP\}$ for the generation G. Each solution is a vector representing a point in the search space.
2. Evaluation: Each candidate solution fitness is evaluated based on the objective function (e.g., the F1-Score).
3. Selection: Three mutually distinct solutions, r_1, r_2, and r_3, are randomly selected from the population as parents to generate a new descendant solution.
4. Mutation: A mutation operation is applied to each parameter within the descendant vector. Such mutation introduces a small perturbation by adding or subtracting a user-defined scaling factor F, multiplied by the difference between two parent solutions, namely $\boldsymbol{\theta}_{r_2}^G$ and $\boldsymbol{\theta}_{r_3}^G$, finally generating a mutated vector $\mathbf{v}_i$ as follows:

$$\mathbf{v}_i^{G+1} = \boldsymbol{\theta}_{r_1}^G + F(\boldsymbol{\theta}_{r_2}^G - \boldsymbol{\theta}_{r_3}^G) \tag{7}$$

5. Crossover: A crossover operation combines information from the parents and the mutated vector to create a trial solution $\boldsymbol{\mu}_{ji}$. Each parameter in the latter trial solution is inherited from either the parent vector or the mutated vector with a probability determined by a user-defined crossover rate CR. The trial solution $\boldsymbol{\mu}_{ji}$ is determined for the next generation $G + 1$ as follows:

$$\boldsymbol{\mu}_{ji}^{G+1} = \begin{cases} \mathbf{v}_{ji}^{G+1}, & \text{if } rand_b(j) \leq CR \text{ or } j = rnbr(i), \\ \boldsymbol{\theta}_{ji}^G, & \text{if } rand_b(j) > CR \text{ or } j \neq rnbr(i), \end{cases} \tag{8}$$

   where $j = 1, 2, \ldots, D$ represents the number of dimensions, $rand_b(j)$ is a uniform random number within the interval $[0, 1]$, D is the dimensionality of the search space, and $rnbr(i)$ is a randomly selected index which ensures at least one parameter comes from the mutated vector.
6. Selection: The offspring, i.e., the trial solution, is compared with its parent. If the latter offspring exhibits better fitness, it replaces the parent in the next generation. Otherwise, the parent is retained.
7. Termination: The algorithm iterates through the above-reported phases until a stopping criterion is satisfied, such as attaining a predetermined number of generations or fitness levels.

2.7. SHapley Additive exPlanations

SHAP is a remarkable method used to interpret the output of ML models [38]. The latter is based on Shapley values, borrowed from cooperative game theory, which allocates credit for an ML model prediction to each feature or group of feature values. Such an approach ensures that the contribution of each feature is fairly distributed, thus providing a clear and consistent explanation of the considered ML model behavior.

SHAP operates by decomposing ML model output into the contributions of each feature. In particular, the contribution of each feature to the model prediction is represented by a SHAP value. Such values aim to explain the importance of each feature in determining the model output. Mathematically, the SHAP value ϕ_i for a feature i is defined as:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \quad (9)$$

where N is the set of all features, S is a subset of N that does not include feature i, $|S|$ is the number of features in subset S, $f(S)$ is the model prediction using the features in subset S.

The above-reported formula ensures that the contribution of each feature is fairly distributed based on its impact on the model prediction. Indeed, the factorial terms $|S|!$ and $(|N| - |S| - 1)!$ account for the different permutations of features, ensuring that each feature contribution is weighted appropriately. As a result, by breaking down a model output into the contributions of individual features, SHAP provides a clear and intuitive explanation of how the model makes its predictions.

It is worth noting that one of the key strengths of SHAP is its neutrality towards models. The latter means that SHAP values can be computed for any ML model, regardless of its complexity or type. Indeed, this model-agnostic nature allows SHAP to generate consistent explanations across different models, making it a versatile tool for interpreting complex model behaviors. As a final result, SHAP values can be leveraged to convey the significance of each feature to a human user. Finally, this method can be especially beneficial in fields like finance, healthcare, and any area where comprehending a model's decision-making process is essential.

2.8. Computational Framework

The proposed approach basically consists of three steps: (i) data collection, (ii) model development, and (iii) performance evaluation and further analysis. Figure 5 illustrates these steps. The data collection stage begins by reading the audio files. These are then submitted to the feature extraction process, which, in this case, was carried out by the TSFEL library. At the end of these steps, the database containing the attributes of all the signals is built. After that, the model development stage begins. This part was set up to run 50 times independently to obtain more statistically robust results. Initially, the previously built database is randomly divided into a training set (70%) and a test set (30%). From the training set, the DE algorithm was used in conjunction with the cross-validation method to optimize the parameters of the classification model used (SVC, MLP, XGB). The best set of hyperparameters is then selected to adjust the model, which is then evaluated from the perspective of the test set. Finally, once the classes have been predicted, it is possible to evaluate the classification performance and carry out both parametric and feature importance analysis.

In the present study, DE was employed to optimize the hyperparameter of the selected ML models. In particular, the DE configuration involved four key parameters, as detailed in Table 3. Such parameters guided the search process to identify the optimal configurations for the chosen ML models.

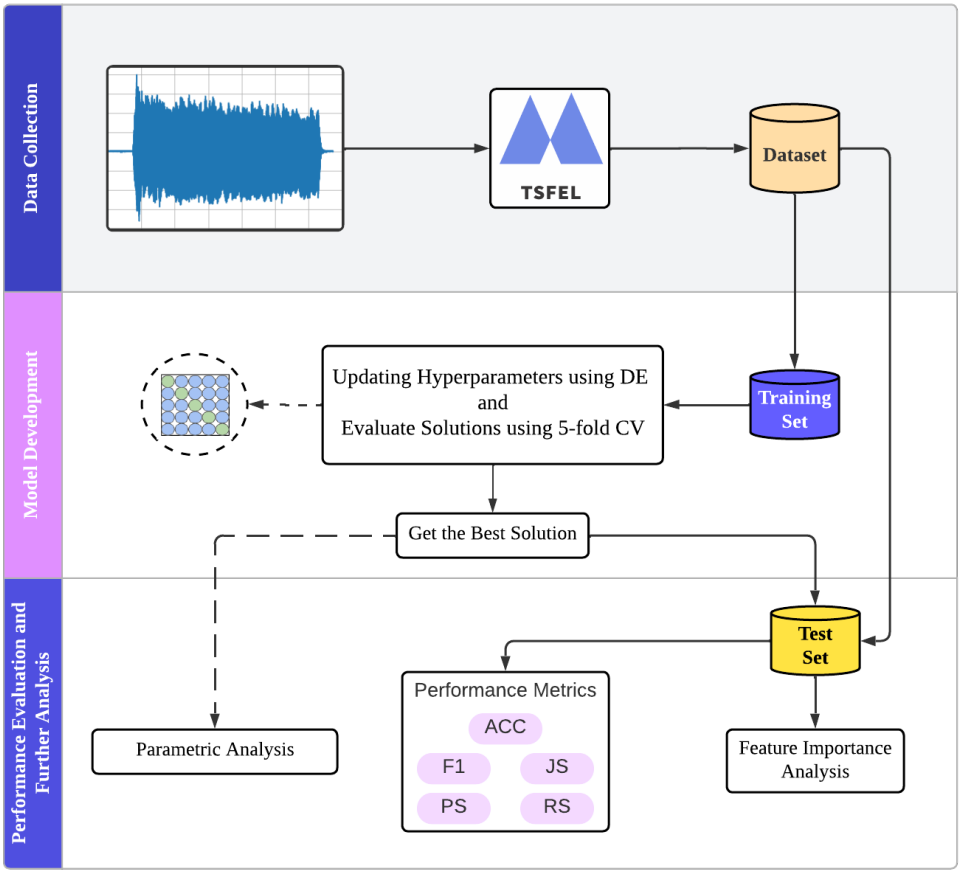


Figure 5. Flowchart of the methodological process used.

Table 3. The employed DE algorithm configuration. Each reported parameter is followed by its respective description and selected parameter value.

Parameter	Description	Value
NP	Population Size	20
G_{max}	Maximum Generations	30
CR	Amplification Factor	0.9
F	Mutation Rate	0.8
$f(\boldsymbol{\theta})$	Objective Function	F1-Score

The stopping criteria for DE were twofold:

1. Maximum generations: The search terminated after reaching a user-specified maximum number of generations, i.e., G_{max}. In the present study, G_{max} was set to 30 generations.
2. Convergence threshold: The search halted when the standard deviation of the Euclidean distance between individuals within the population fell below a predefined threshold, i.e., 10^{-4}. Such criterion ensured convergence towards a stable solution.

In the present research, each candidate solution within the DE population, denoted by $\boldsymbol{\theta}$, encodes the hyperparameters of a specific employed ML model (SVM, MLP, or XGB). The specific encoding scheme for each ML model is presented in Table 4.

Table 4. The below table reports the encoding of candidate solutions for hyperparameters where, for each ML model parameter, it is respectively reported the search range or set.

Model	θ	Description	Range/Set
SVC	θ_1	Kernel, ϕ	0: linear, 1: rbf, 2: sigmoid
	θ_2	Bandwidth parameter, γ	$[10^{-5}, 1000]$
	θ_3	Regularization parameter, C	[1, 10,000]
MLP	θ_1	L_2 regularization term, (α)	[0.0001, 1]
	θ_2	Number of hidden layers (NL)	[1, 3]
	θ_3	No. neurons in the layers (NN)	[1, 50]
	θ_4	Activation function (φ)	0: Identity, 1: Logistic, 2: Tanh, and 3: ReLU
	θ_5	Solver	0: Aadam, 1: lbfgs, 2: sgd
XGB	θ_1	L1 regularization on weights (α)	[0, 1]
	θ_2	L2 regularization on weights (λ)	[0, 1]
	θ_3	Learning rate	$[10^{-6}, 1]$
	θ_4	No. estimators	[10, 300]

Moreover, the classification performance of the ML models with hyperparameters optimized by DE was assessed relying on the following performance metrics: Accuracy (Acc), Precision Score (PS), Recall Score (RS), F1-Score (F1), and Jaccard Score (JS), reported as follows:

- Accuracy: it is determined relying on Equation (10), and it represents the percentage of accurately classified samples compared to the total amount of samples, as follows

$$Acc = \frac{1}{N}\sum_{k=1}^{N} I(f(\mathbf{x}_k) = c_k) \tag{10}$$

  where $f(\mathbf{x}_k)$ denotes the predicted class label of a test sample , c_k represents the true class label of this sample, and $I(true) = 1$ and $I(false) = 0$ are indicator functions.
- Precision Score: It is the proportion of accurately predicted positive instances to all positive expected cases. Equation (11) shows its mathematical formulation as follows:

$$PS = \frac{TP_k}{PP_k} \tag{11}$$

  where TP_k and PP_k are the number of true positives and predicted positives for the class c_k.
- Recall Score: In mathematical terms, it may be expressed as the ratio of properly predicted positive instances to the total number of true positive cases, as reported in the following Equation (12)

$$RS = \frac{TP_k}{TP_k + FN_k} \tag{12}$$

  where FN_k represents the number of false negatives for the class c_k.
- F1-Score: It is the harmonic mean of Precision and Recall, thus emphasizing both high precision and recall. Its mathematical formulation is given by the next Equation (13)

$$F1 = \frac{2TP_k}{2TP_k + FP_k + FN_k} \tag{13}$$

  where FP_k is the total number of false positives for the class c_k.
- Jaccard Score: It is computed as the ratio of the intersection size between the predicted and actual label sets to the size of their union. Such metric is calculated according to the following Equation (14)

$$JS = \frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|} \tag{14}$$

where the actual label set is represented by y_i while $\hat{y}_i$ stands as the predicted label set.

3. Computational Experiments

The proposed method achieves consistent performance across all the employed evaluation metrics, as shown in Table 5. Notably, the XGB model exhibits superior classification performance compared to SVC and MLP models. The latter is evident from the higher average accuracy and lower standard deviation values obtained by XGB.

Table 5. Average performance metrics and their respective standard deviation. Standard deviation is computed using a five-fold stratified cross-validation set as a performance assessment strategy.

Estimator	Acc	F1	PS	RS	JS
SVC	0.872 (0.081)	0.866 (0.092)	0.881 (0.080)	0.872 (0.081)	0.794 (0.112)
MLP	0.912 (0.095)	0.902 (0.121)	0.911 (0.119)	0.912 (0.095)	0.853 (0.137)
XGB	**0.976** (0.012)	**0.976** (0.012)	**0.978** (0.012)	**0.976** (0.012)	**0.955** (0.023)

To further confirm the above-reported findings, a comparison between the SVC, MLP, and XGB models' performance across all metrics was carried out via the Analysis of Variance (ANOVA) test. Results of ANOVA were reported in Table 6, and it must be noted that statistically significant differences were observed between XGB and both SVC and MLP models, with p-values less than 0.05 for all metrics. Conversely, the ANOVA test between SVC and MLP yielded p-values greater than 0.05 for all metrics, indicating no statistically significant difference in classification performance. Such results confirmed the superiority of the proposed classification method, employing XGB for achieving statistically significant improvements if compared to the other baseline models, i.e., SVC and MLP.

Table 6. The table contains the obtained p-values from the ANOVA test performed by combining the models two by two. Values below 0.05 indicate a statistically significant difference with 95% confidence.

Models in Comparison	Performance Metrics				
	Acc	F1	JS	PS	RS
SVC × MLP	0.011175	0.072524	0.008140	0.091105	0.011175
MLP × XGB	7.76×10^{-6}	4.80×10^{-5}	1.02×10^{-6}	1.37×10^{-4}	7.76×10^{-6}
SVC × XGB	7.59×10^{-16}	1.80×10^{-14}	8.69×10^{-19}	3.75×10^{-14}	7.59×10^{-16}

To complement the performance analysis, Figure 6 reports the average confusion matrices for all models. As anticipated, due to their inherent similarities, the highest classification errors occurred when distinguishing between mixed and chest singing registers. The SVC model had the highest misclassification rate, with 23% of mixed voice records incorrectly classified as chest voice. Its second-highest error occurred in the opposite direction, with 17% of chest voice records classified as mixed voice. The MLP model showed a similar confusion rate between these categories of singing registers (11%). In contrast, the XGB model confusion matrix revealed an opposite trend to SVC, with the highest confusion rate (3.7%) occurring during the chest-to-mixed misclassification, followed by mixed-to-chest (2.4%).

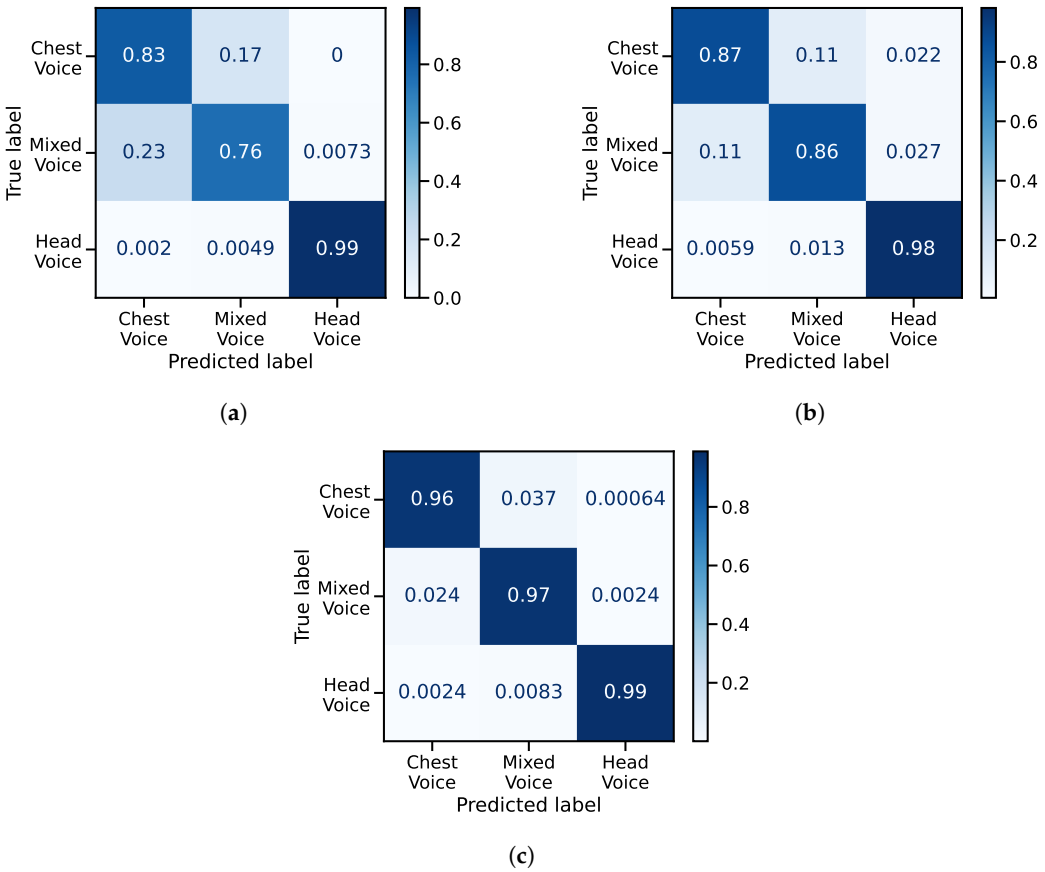


Figure 6. The figure reports the average confusion matrices for each evaluated ML model: (**a**) SVC, (**b**) MLP, and (**c**) XGB. The reported classes, respectively, refer to chest, mixed, and head voices.

Finally, all the employed ML models achieved the highest classification accuracy for the head voice registers. The second-highest correct classification rate was associated with the chest voice for SVC and MLP models and mixed voice for XGB. Notably, the XGB model achieved superior performance with over 96% accuracy in correctly classifying all three singing registers, highlighting its effectiveness in singing register recognition.

3.1. Parametric Analysis

The distributions of the identified parameters were analyzed to assess the parameter variations across the models obtained from 50 independent DE runs.

First, the SVC model exhibited a concentrated distribution for the parameter C between 2844 and 6841, with a median value around 4659—Figure 7a. Similarly, the parameter γ displayed a majority distribution within the range of 300 and 699, with a median of approximately 492—Figure 7b. Notably, the kernel parameter remained consistently selected as the linear type across all 50 runs, as shown in Figure 7c.

For the MLP model, the values chosen for the number of neurons (NN) were mostly distributed between 56 and 80, with a median value close to 72—Figure 8a. The regularization term (α), on the other hand, was mostly defined within the $[0.37, 0.73]$ interval, around the median 0.57 —Figure 8b. The number of layers (NL) was set as 1 in 42 runs and set as 2 in 8 runs—Figure 8c. Concerning the activation function, the hyperbolic tangent activation function was selected more often (30 times) than the logistic function (20 times)—Figure 8d. Finally, the *lbfgs* was unanimously chosen as the solver parameter—Figure 8e.

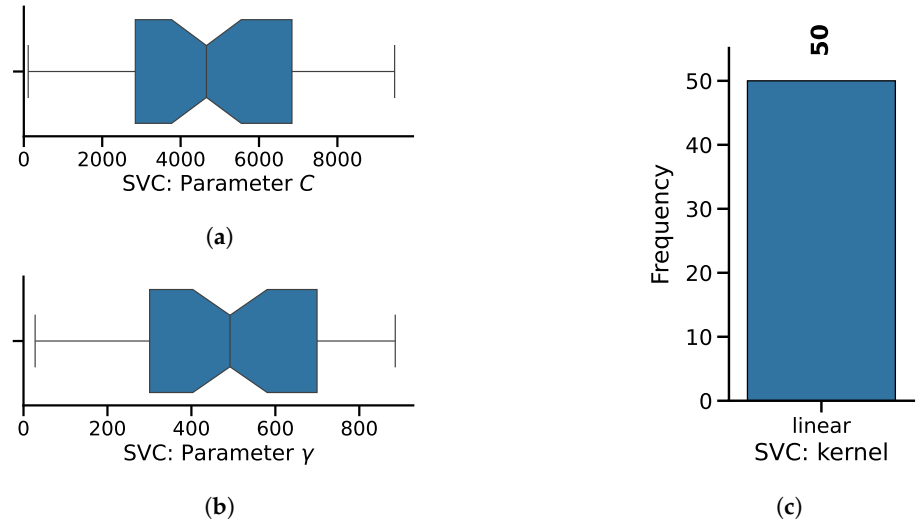


Figure 7. Distribution of internal parameters for the SVC model: (**a**) parameter C, (**b**) parameter γ, and (**c**) kernel type.

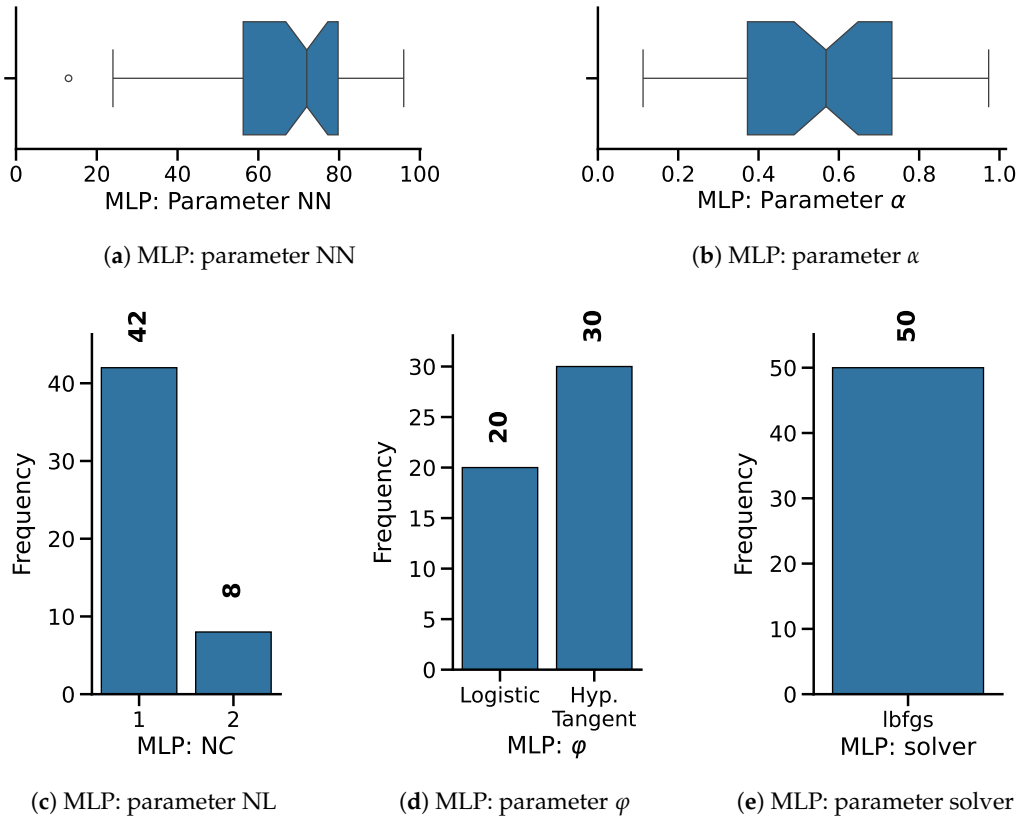


Figure 8. Distribution of internal parameters for the MLP model: (**a**) number of neurons, (**b**) parameter α, (**c**) number of layers, (**d**) parameter ϕ, and (**e**) solver type.

Concerning the XGB model parameters, the parameter α had the choice concentrated in the range 0.14 and 0.56, whose central value of the Distribution was 0.25—Figure 9a. The λ parameter was mostly distributed between 0.27 and 0.75, with a median value of 0.50 —Figure 9b. The learning rate parameter had the highest distribution density in the interval $[0.42, 0.84]$, centered on 0.61—Figure 9c. Finally, the number of estimators showed a distribution centered on 149, with 95 and 224 first and third-quartile values, respectively—Figure 9d.

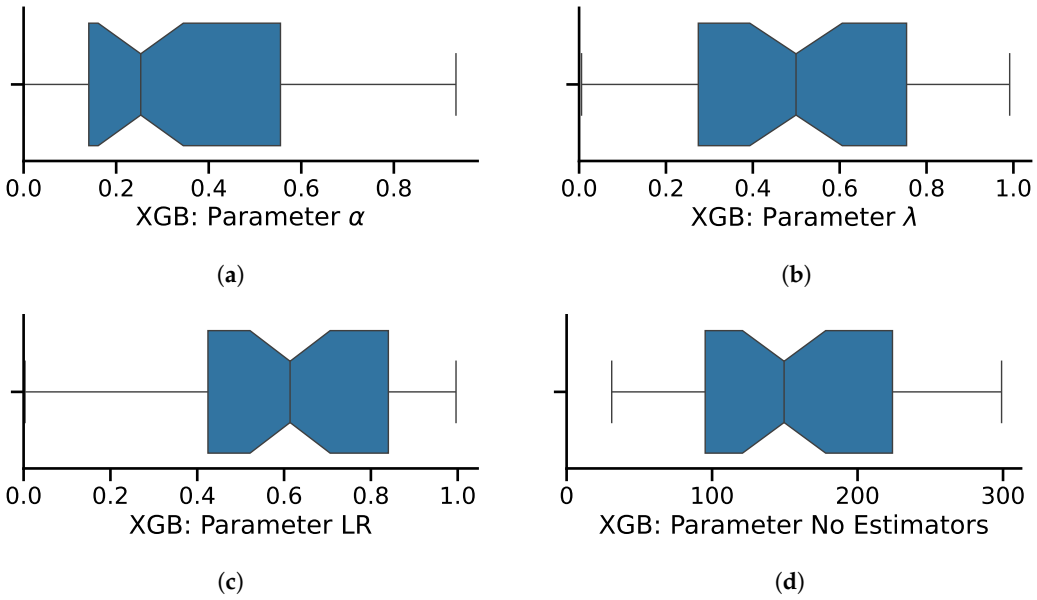


Figure 9. Distribution of internal parameters for the XGB model: (**a**) parameter α, (**b**) parameter λ, (**c**) learning rate, and (**d**) number of estimators.

3.2. Feature Importance Analysis

Relying on the SHAP method, presented in Section 2.7, feature importance scores were calculated for each ML model during the classification analysis on the employed dataset. The average importance scores were summarized in Figure 10. Such a figure presents the variable importance graphs for the MLP, SVC, and XGB ML models, all applied to classify voice registers. The latter graphics emphasize the most influential variables, consistent with the fundamental behavior. Indeed, X_{14} (the signal zero-crossing rate) was the variable that significantly impacted the classification for all methods. For SVC and XGB, X_{11} (the signal traveled distance) and X_9 (no. peaks from a defined signal neighborhood) came in the second and third positions. Finally, the other features had lower values for all the employed ML methods.

The zero-crossing rate (X_{14}) is closely tied to the periodicity and noise characteristics of the human voice, reflecting the distinction between voiced sounds (low X_{14}) and unvoiced sounds (high X_{14}) [39]. The distance a signal travels can affect the human voice by introducing attenuation, delays, and environmental noise, which alter its amplitude and clarity. The voice quality degrades over distance due to energy loss and interference, impacting communication effectiveness [40]. The number of peaks in a defined signal neighborhood reflects the harmonic structure and periodicity of the human voice, which are key aspects of the source-filter model of speech production. This feature helps distinguish voiced sounds (with regular peaks) from unvoiced sounds (with irregular or fewer peaks), aiding in pitch detection and speech analysis [41].

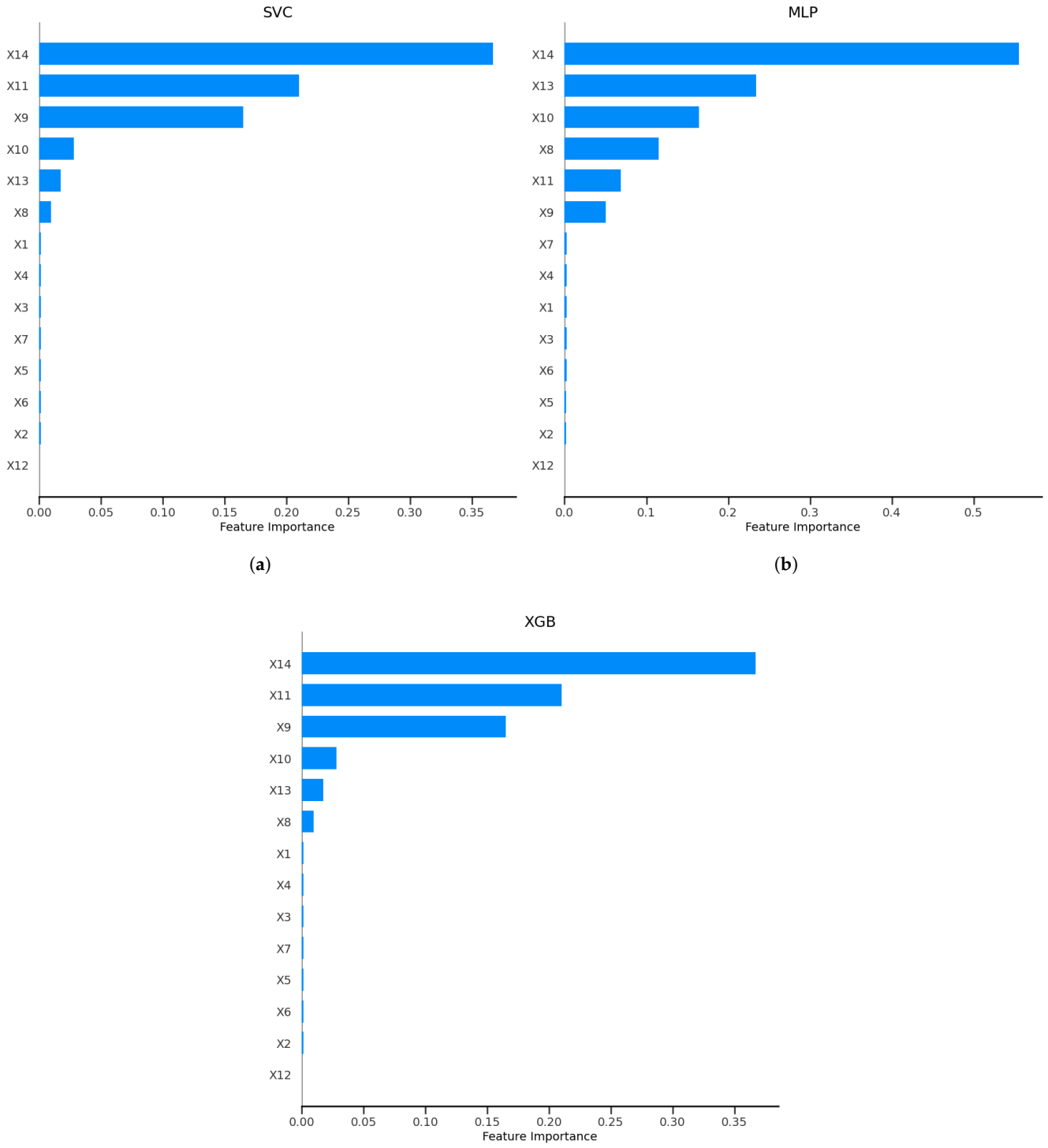


Figure 10. Feature importance (SHAP) scores for classification of the test datasets for each ML model: (**a**) SVC, (**b**) MLP, (**c**) XGB.

3.3. Model Strengths and Limitations

Although the data collection process followed standardized procedures, there are opportunities for improvement. One key limitation lies in the consistency of recording conditions, as the audio data were obtained using a single microphone in a single environment. Future research should explore datasets incorporating more diverse recording setups to enhance robustness. Additionally, although this research employed a substantial number of samples per singing register category, the dataset's scope was constrained by the limited number of singers. While

experienced singers were deliberately selected to ensure reliable execution of vocal registers—a necessary condition once some registers may require considerable expertise—increasing the number and diversity of singers, i.e., including a broader range of vocal characteristics, age groups, and expertise levels would enrich the dataset, improve model generalizability, and enable more comprehensive evaluations of ML and DL methods.

Expanding the analysis to include high-level audio descriptors, such as Mel-Frequency Cepstral Coefficients [42], is highly recommended. These descriptors are widely regarded in audio processing for their ability to simulate the human auditory capacity and could offer deeper insights into the acoustic properties of singing registers. Another promising direction is exploring advanced ML or DL models, particularly architectures like recurrent neural networks capable of directly processing raw audio signals. Such models could uncover latent features overlooked by traditional descriptors, thereby advancing the understanding of intricate, non-linear relationships within the data [43,44].

The encouraging performance of the models developed in the present work opens new avenues for practical applications. In particular, one promising direction could be represented by the development of real-time, online classification models for distinguishing between singing registers. Such models could be integrated into user-friendly tools for singing pedagogy. For instance, a smartphone application could leverage these algorithms to assist vocal coaches and students refine vocal techniques. By providing immediate feedback, the app could serve as an educational aid that promotes safer vocal practices, reducing the risk of phonatory injuries.

Further research should focus on creating new, high-quality datasets with flexible audio signal collection using accessible devices like smartphones to democratize participation and facilitate innovation. These datasets could support the integration of ML and DL models into low-cost embedded systems, enabling the development of affordable tools for a wider audience.

Finally, while the present study lays a strong foundation for singing register classification using computational techniques, future work should focus on enhancing feature extraction processes, diversifying datasets, and exploring state-of-the-art ML and DL models. Additionally, translating these advancements into practical, accessible tools would bridge the gap between research and real-world applications, ultimately benefiting educators, students, and practitioners in the field of vocal training.

4. Conclusions

The presented article introduced the problem of identifying different singing registers by leveraging a hybrid learning strategy that consisted of optimizing the hyperparameters of ML models through a DE algorithm. To accomplish the latter task, a database consisting of 350 audio files from three different singing registers was built. In order to retrieve valuable information from the latter signals, 14 temporal features were extracted from each recorded signal. Finally, three ML models were evaluated using such features as input data.

The top-performing DE-optimized XGB model achieved a high classification performance, with an average classification accuracy of approximately 97.60% and a low standard deviation of 1.2%. The latter results suggested that the extracted features effectively captured the relevant information for the singing register recognition task, thus enabling the XGB model to learn robust decision boundaries for accurate classification. In addition, an analysis of the confusion matrices for the three employed ML models revealed that most of the classification confusion occurred between the mixed and chest registers, as expected, due to the more remarkable similarity between them than with the head voice registers.

In the latter sense, the main contribution of the present research work consisted of proposing a hybrid ML tool for singing register classification with a flexible computational framework that allows for automated ML model parameters search. The investigated method combines evolutionary search on internal parameters with ML models, providing an accurate

alternative for singing register categorization tasks. The found significance lies in its ability to automatically adjust the ML model's internal parameters, effectively compensating for variations in the input data. Furthermore, the latter strategy helps optimize the user's effort in determining the optimal internal parameters for a classifier because manual tuning may be time-consuming and require particular ML understanding.

Author Contributions: Conceptualization: T.B. and G.d.O.C.; Methodology: L.G., T.B. and A.M.; Software: T.B. and C.M.S.; Validation: C.M.S. and M.B.; Formal Analysis: A.M., L.G., M.B. and A.C.; Investigation: G.d.O.C., A.C. and A.K.S.T.R.A.; Resources: A.M., A.K.S.T.R.A., M.B. and L.G.; Funding: M.B.; Data Curation: T.B. and G.d.O.C.; Writing—original draft: A.M., A.K.S.T.R.A., T.B., A.C. and G.d.O.C.; Writing—review and editing: L.G., M.B., C.M.S., A.C. and A.M.; Supervision: A.C. and L.G. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the support of the funding agencies CNPq (grants 401796/2021-3, 307688/2022-4, 303982/2022-5, 409433/2022-5, and 402533/2023-2), FAPES (grant 518/2021), FAPEMIG (grants APQ-04458-23 and APQ-00032-24) and CAPES (finance code 001) for their financial support.

Data Availability Statement: The developed code and collected data presented in the study are openly available in the GitHub repository with url: https://github.com/LGoliatt/signals-3429481 (accessed on 17 February 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sataloff, R. *Vocal Health and Pedagogy: Science, Assessment, and Treatment*, 3rd ed.; Plural Publishing, Incorporated: San Diego, CA, USA, 2017.
2. Almaghrabi, S.A.; Clark, S.R.; Baumert, M. Bio-acoustic features of depression: A review. *Biomed. Signal Process. Control* **2023**, *85*, 105020. [CrossRef]
3. Zhang, Z. Mechanics of human voice production and control. *J. Acoust. Soc. Am.* **2016**, *140*, 2614–2635. [CrossRef] [PubMed]
4. Meireles, A.; Mixdorff, H. Voice Quality in Low and High Registers in Two Different Styles of Singing. In Proceedings of the 10th International Conference on Speech Prosody 2020, Tokyo, Japan, 25–28 May 2020.
5. Kent, R.; Read, C. *The Acoustic Analysis of Speech*; Singular Thomson Learning: London, UK, 1992.
6. Fant, G. *Acoustic Theory of Speech Production*; Mouton: Hague, The Netherlands, 1960.
7. Livingstone, S.R.; Peck, K.; Russo, F.A. Acoustic differences in the speaking and singing voice. *J. Acoust. Soc. Am.* **2013**, *133*, 3591. [CrossRef]
8. Ke, Z.; Yin, Y. Tail Risk Alert Based on Conditional Autoregressive VaR by Regression Quantiles and Machine Learning Algorithms. In Proceedings of the 2024 5th International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Wuhu, China, 8–10 November 2024.
9. Feng, J.; Wu, Y.; Sun, H.; Zhang, S.; Liu, D. Panther: Practical Secure Two-Party Neural Network Inference. *IEEE Trans. Inf. Forensics Secur.* **2025**, *20*, 1149–1162. [CrossRef]
10. Xu, Y.; Wang, W.; Cui, H.; Xu, M.; Li, M. Paralinguistic singing attribute recognition using supervised machine learning for describing the classical tenor solo singing voice in vocal pedagogy. *EURASIP J. Audio Speech Music Process.* **2022**, *2022*, 8. [CrossRef] [PubMed]
11. Wang, Z.; Müller, M.; Caffier, F.; Caffier, P.P. Harnessing Machine Learning in Vocal Arts Medicine: A Random Forest Application for "Fach" Classification in Opera. *Diagnostics* **2023**, *13*, 2870. [CrossRef]
12. Sha, C.Y.; Yang, Y.H.; Lin, Y.C.; Chen, H.H. Singing voice timbre classification of Chinese popular music. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 734–738. [CrossRef]
13. Han, J.; Montagna, M.; Grammenos, A.; Xia, T.; Bondareva, E.; Siegele-Brown, C.; Chauhan, J.; Dang, T.; Spathis, D.; Floto, R.A.; et al. Evaluating Listening Performance for COVID-19 Detection by Clinicians and Machine Learning: Comparative Study. *J. Med. Internet Res.* **2023**, *25*, e44804. [CrossRef]
14. Tripathi, T.; Kumar, R. Speech-based detection of multi-class Alzheimer's disease classification using machine learning. *Int. J. Data Sci. Anal.* **2023**, *18*, 83–96. [CrossRef]
15. Briend, F.; David, C.; Silleresi, S.; Malvy, J.; Ferré, S.; Latinus, M. Voice acoustics allow classifying autism spectrum disorder with high accuracy. *Transl. Psychiatry* **2023**, *13*, 250. [CrossRef]

16. Müller, M.; Wang, Z.; Caffier, F.; Caffier, P.P. New objective timbre parameters for classification of voice type and fach in professional opera singers. *Sci. Rep.* **2022**, *12*, 17921. [CrossRef]
17. Boratto, T.H.; Cury, A.A.; Goliatt, L. Machine learning-based classification of bronze alloy cymbals from microphone captured data enhanced with feature selection approaches. *Expert Syst. Appl.* **2023**, *215*, 119378. [CrossRef]
18. Martinho, A.D.; Ribeiro, C.B.M.; Gorodetskaya, Y.; Fonseca, T.L.; Goliatt, L. Extreme Learning Machine with Evolutionary Parameter Tuning Applied to Forecast the Daily Natural Flow at Cahora Bassa Dam, Mozambique. In Proceedings of the Bioinspired Optimization Methods and Their Applications, Brussels, Belgium, 19–20 November 2020; Filipič, B., Minisci, E., Vasile, M., Eds.; Springer: Cham, Switzerland, 2020; pp. 255–267.
19. Saporetti, C.M.; da Fonseca, L.G.; Pereira, E. A Lithology Identification Approach Based on Machine Learning With Evolutionary Parameter Tuning. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1819–1823. [CrossRef]
20. Boratto, T.H.A.; Saporetti, C.M.; Basilio, S.C.A.; Cury, A.A.; Goliatt, L. Data-driven cymbal bronze alloy identification via evolutionary machine learning with automatic feature selection. *J. Intell. Manuf.* **2022**, *35*, 257–273. [CrossRef]
21. Claesen, M.; Moor, B.D. Hyperparameter Search in Machine Learning. *arXiv* **2015**, arXiv:1502.02127. [CrossRef]
22. Ahmed, S.F.; Alam, M.S.B.; Hassan, M.; Rozbu, M.R.; Ishtiak, T.; Rafa, N.; Mofijur, M.; Shawkat Ali, A.; Gandomi, A.H. Deep learning modelling techniques: Current progress, applications, advantages, and challenges. *Artif. Intell. Rev.* **2023**, *56*, 13521–13617. [CrossRef]
23. Miller, R. *The Structure of Singing: System and Art Vocal Technique*; Schirmer: New York, NY, USA, 1996.
24. Barandas, M.; Folgado, D.; Fernandes, L.; Santos, S.; Abreu, M.; Bota, P.; Liu, H.; Schultz, T.; Gamboa, H. TSFEL: Time Series Feature Extraction Library. *SoftwareX* **2020**, *11*, 100456. [CrossRef]
25. Lee, Y.; Oya, M.; Kaburagi, T.; Hidaka, S.; Nakagawa, T. Differences Among Mixed, Chest, and Falsetto Registers: A Multiparametric Study. *J. Voice* **2023**, *37*, 298.e11–298.e29. [CrossRef]
26. Behringer | Product | Q1202USB—behringer.com. Available online: https://www.behringer.com/product.html?modelCode=0601-AGC (accessed on 9 January 2025).
27. AT2020—audio-technica.com. Available online: https://www.audio-technica.com/en-eu/at2020 (accessed on 9 January 2025).
28. Bhattacharyya, S.S.; Deprettere, E.F.; Leupers, R.; Takala, J. *Handbook of Signal Processing Systems*, 2nd ed.; Springer Publishing Company, Incorporated: Cham, Switzerland, 2013. [CrossRef]
29. Vapnik, V.N. *Statistical Learning Theory*; Wiley-Interscience: Hoboken, NJ, USA, 1998.
30. El Bouchefry, K.; de Souza, R.S. Chapter 12—Learning in Big Data: Introduction to Machine Learning. In *Knowledge Discovery in Big Data from Astronomy and Earth Observation*; Škoda, P., Adam, F., Eds.; Elsevier: Amsterdam, The Netherlands, 2020; pp. 225–249.
31. Gardner, M.; Dorling, S. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. *Atmos. Environ.* **1998**, *32*, 2627–2636. [CrossRef]
32. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
33. Wythoff, B.J. Backpropagation neural networks: A tutorial. *Chemom. Intell. Lab. Syst.* **1993**, *18*, 115–155. [CrossRef]
34. Haykin, S. *Neural Networks: A Comprehensive Foundation*, 3rd ed.; Prentice-Hall, Inc.: Wilmington, DE, USA, 2007.
35. Ibrahem Ahmed Osman, A.; Najah Ahmed, A.; Chow, M.; Feng Huang, Y.; El-Shafie, A. Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia. *Ain Shams Eng. J.* **2021**, *12*, 1545–1556. [CrossRef]
36. Chen, T.; He, T. Higgs Boson Discovery with Boosted Trees. In Proceedings of the 2014 International Conference on High-Energy Physics and Machine Learning HEPML'14, Montreal, QC, Canada, 8–13 December 2014; Volume 42, pp. 69–80. Available online: https://proceedings.mlr.press/v42/chen14.pdf (accessed on 17 February 2025).
37. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
38. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the 31st Conference on Neural Information Processing System, NIPS'17, Red Hook, NY, USA, 4–9 December 2017; pp. 4768–4777. Available online: https://dl.acm.org/doi/pdf/10.5555/3295222.3295230 (accessed on 17 February 2025).
39. Radhakrishnan, P.; Nampoori, V. Speech Analysis Using Modern Techniques of Nonlinear Dynamics. Ph.D. Thesis, Cochin University of Science and Technology, Kochi, India, 2009.
40. Härmä, A. Ambient human-to-human communication. In *Handbook of Ambient Intelligence and Smart Environments*; Springer: Cham, Switzerland, 2010; pp. 795–823.
41. Kunchur, M.N. The human auditory system and audio. *Appl. Acoust.* **2023**, *211*, 109507. [CrossRef]
42. Abdul, Z.K.; Al-Talabani, A.K. Mel Frequency Cepstral Coefficient and its Applications: A Review. *IEEE Access* **2022**, *10*, 122136–122158. [CrossRef]

43. Zhang, W.; Yang, G.; Lin, Y.; Ji, C.; Gupta, M.M. On Definition of Deep Learning. In Proceedings of the 2018 World Automation Congress (WAC), Stevenson, WA, USA, 3–6 June 2018; pp. 1–5. [CrossRef]
44. Bodini, M. A Review of Facial Landmark Extraction in 2D Images and Videos Using Deep Learning. *Big Data Cogn. Comput.* **2019**, *3*, 14. [CrossRef]