Preprints.org

Article

Not peer-reviewed version

Self-Organized Feature Engineering Model for Accurate Speech Disorder Classification

Zeynep Hilal Kilimci , Burak Celik , Ayhan Küçükmanisa *

Posted Date: 26 October 2026

Keywords: acoustic signal analysis; speech disorder classification; self-organized feature engineering; interpretable classification

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Self-Organized Feature Engineering Model for Accurate Speech Disorder Classification

Zeynep Hilal Kilimci [1], Burak Celik [2] and Ayhan Küçükmanisa [2,*]

[1] Department of Information Systems Engineering, Faculty of Technology, Kocaeli University, Kocaeli, Türkiye
[2] Department of Electronics and Communication Engineering, Faculty of Engineering, Kocaeli University, 41001 Kocaeli, Türkiye
* Correspondence: ayhan.kucukmanisa@kocaeli.edu.tr

Abstract

Dysarthria, spasmodic dysphonia and stuttering are still diagnosed mainly through perceptual assessment, which is subjective and difficult to scale. This study presents a lightweight and explainable feature engineering model that distinguishes these three disorders from normal speech. Its core is the expansion–contraction pattern (ExCoPat), a lattice-based textural descriptor whose traversal path is determined by the data of each signal block. ExCoPat and twenty statistical measures are computed from the raw signal and from eleven low-pass sub-bands of a db4 multilevel discrete wavelet transform, giving 10,176 features per clip. Iterative neighborhood component analysis and iterative ReliefF select the informative features, k-nearest neighbors and support vector machine classifiers generate the predictions, and iterative majority voting with greedy selection provides the final decision. On a curated dataset of 5,244 three-second clips, the model achieved 98.95% accuracy, 98.92% unweighted average recall and 98.94% F1-score under 10-fold cross-validation. Under the same protocol, an artificial neural network with three hidden layers trained on cepstral statistics and a shallow three-layer convolutional neural network trained on log-mel spectrograms reached only 93.97% and 89.84% accuracy, respectively. Removing the ExCoPat features reduced accuracy by 11.27 percentage points, confirming their central role. Feature extraction required 433 ms per clip on a CPU, and the extractor has no trainable parameters. These results suggest that a transparent handcrafted pipeline can serve as a practical tool for multi-disorder speech screening.

Keywords: acoustic signal analysis; speech disorder classification; self-organized feature engineering; interpretable classification

1. Introduction

Speech is one of the most expressive channels of human communication, and its disruption has a direct impact on quality of life, social participation, and clinical outcome. Motor speech disorders constitute a heterogeneous group of such conditions. Dysarthria, which arises from neuromuscular impairment of the speech mechanism, affects a large share of patients with neurological disease such as Parkinson's disease and stroke, and its acoustic correlates have been studied extensively with both handcrafted features and deep learning [1]. Spasmodic dysphonia (SD), a focal laryngeal dystonia, produces strained-strangled or breathy voice breaks and remains notoriously difficult to diagnose; a large consensus study has shown that even expert clinicians rely on a limited set of perceptual attributes and disagree on many cases [2]. Voice analysis has consequently been proposed as an objective complement to perceptual evaluation in SD, both for diagnosis and for monitoring botulinum toxin response [3], although treatment remains symptomatic [4]. Stuttering, a fluency disorder characterized by repetitions, prolongations, and blocks, affects about 1% of adults and has motivated a growing body of automatic detection research [5]. Despite their clinical distinctness, these disorders share a common practical bottleneck: diagnosis depends on perceptual judgment by experienced speech-language pathologists, which is time-consuming, subjective, and not easily scalable.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

Against this background, automatic classification of pathological speech from acoustic signal analysis has attracted increasing attention. Early studies demonstrated that classical pattern-recognition pipelines built on handcrafted acoustic features can separate disordered from healthy voices: Schlotthauer et al. applied nonlinear and spectral descriptors to discriminate SD from muscle tension dysphonia [6], and subsequent work on SD confirmed that feature extraction combined with conventional classifiers yields diagnostically useful performance [7]. Machine-learning assessment of SD based on acoustic and perceptual parameters has recently been extended toward clinically oriented outcome measures [8], and multi-class frameworks have begun to separate more than one disorder from the same acoustic recording [9]. Comparable feature-based modeling has reached competitive diagnostic accuracy for Parkinson's disease from voice recordings using gradient boosting over interpretable measures, reinforcing the value of transparent acoustic descriptors [10]. Studies that explicitly combine handcrafted features with systematic classifier comparison have likewise reported strong results for stuttered-speech pause classification [11] and for multi-label discrimination of co-occurring voice disorders [12]. Neural approaches entered the field along the same path: wavelet-transform features coupled with artificial neural networks (ANNs) were applied to voice-disorder identification as early as 2011 [13], and ANN-based screening of Parkinson's disease from dysphonia measures has since been reported repeatedly [14]. More recently, the field has shifted decisively toward deep learning, with convolutional neural networks (CNNs) in particular dominating pathological speech analysis. Residual neural networks have been shown to quantify dysarthria severity from short-duration speech segments [15], and FluentNet combined CNN and recurrent components for end-to-end detection of stuttered disfluencies [16]. CNN-based architectures have since been applied to real-time dysfluency classification [17], to stuttering detection with data augmentation and class-balanced losses [18], and to end-to-end representation learning for dysarthria in Parkinson's disease [19]. In the voice-pathology domain, CNNs have been used for pathological voice detection from time-frequency representations [20], for voice-disorder classification with one-dimensional CNNs operating on acoustic features [21], for pathological voice detection based on phase reconstitution [22], for multi-scale CNN recognition of pathological voices [23], and for CNN-based automatic classification of stuttering repetitions and prolongations [24]. These models report excellent benchmark accuracy, but they typically require large corpora and substantial computational resources, they embed their decisions in millions of opaque parameters, and their learned representations are difficult to map onto the acoustic dimensions that clinicians reason about. Interpretability, where addressed at all, is usually added post hoc to an otherwise black-box model [25].

This trend exposes a clear gap. In clinical speech pathology, models must often run on modest hardware, must be validated on the small and imbalanced datasets typical of hospital recordings, and must present evidence that a clinician can inspect; heavy deep networks with automatically learned representations satisfy none of these constraints well. To the best of our knowledge, no study has simultaneously distinguished dysarthria, spasmodic dysphonia, and stuttering from normal speech within a single lightweight framework built exclusively on handcrafted acoustic features while remaining fully interpretable. In this work, we address this gap by developing a handcrafted-feature-based, explainable, and lightweight machine-learning model that classifies these four speech categories directly from the speech signal. The proposed approach offers transparent decision cues rooted in well-established acoustic descriptors, a small computational footprint suitable for clinical deployment, and competitive discrimination performance, providing a practical and clinically credible alternative to deep-learning pipelines for multi-disorder speech screening.

1.1. Literature Gaps

We have presented our literature review in the previous section. Based on this literature, the identified literature gaps are:

- The dominant paradigm in pathological speech analysis relies on deep learning (DL) architectures such as CNN, ResNet, LSTM, TDNN and their hybrid variants. While these models have

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

delivered strong benchmark performance, they have high computational complexity, depend on large annotated corpora and GPU infrastructure, and embed their decisions in millions of opaque parameters, which restricts their deployment in hospital and primary care settings.

- Due to the widespread adoption of DL-based pipelines, feature engineering research for speech disorders has largely stagnated. Existing handcrafted approaches mostly rely on static and conventional descriptors such as cepstral coefficients, perturbation measures and large generic acoustic feature sets. Self-organized textural pattern extractors, whose descriptor is selected according to the signal block being analysed, remain absent from the pathological speech literature.
- Most published models focus on a single disorder, a severity scale within one disorder, or a binary healthy-versus-pathological decision. To the best of our knowledge, no study has distinguished dysarthria, spasmodic dysphonia and stuttering from normal speech within a single lightweight framework, although these disorders share the same clinical bottleneck of subjective perceptual assessment.
- Interpretability, where it is addressed at all, is usually added post hoc to an otherwise black-box model. Models whose individual decision cues can be traced back to the signal component, feature type and encoding function that produced them, and whose selected feature subsets are shown to be stable, are rarely reported.
- Feature selection in speech disorder pipelines is commonly performed with a single selector and a fixed, manually chosen number of features. Iterative selectors that determine the optimal feature count automatically, such as iterative neighborhood component analysis (INCA) and iterative ReliefF (IReliefF), together with post-processing fusion strategies such as iterative majority voting (IMV) and greedy outcome selection, have been underexplored in this domain.
- Although many studies describe their models as efficient, lightweight claims are seldom supported by measured evidence such as per-stage runtime, inference latency, model footprint or empirical scaling with signal length.

1.2. Motivation

In this research, our primary motivation is to propose an accurate, explainable and lightweight speech disorder classification model with linear-time feature extraction and GPU-free inference. To this end, we have designed an expansion–contraction lattice pattern, termed ExCoPat, as the core textural feature extraction function of the proposed framework. ExCoPat maps each signal block of length 28 onto a nine-layer lattice with 1-2-4-1-2-4-8-4-2 nodes. In the expansion layers, each node branches into two child nodes and the number of candidate nodes grows, whereas in the contraction layers several nodes converge into fewer nodes. This alternating structure allows the pattern to examine local signal variations at different widths within the same block. In every layer the node to be visited is selected from the data of the current block, so the traversal path is not fixed a priori but is determined by the signal itself. This self-organized path is encoded with three kernel functions into binary descriptors and histogram-based features.

To capture information at multiple levels of signal detail, ExCoPat is applied both to the raw speech signal and to the low-pass sub-bands produced by multilevel discrete wavelet transform (MDWT) decomposition, and statistical features are extracted from the signals and from the generated textural features. In this way, a multilevel hybrid feature set combining textural (nonlinear) and statistical descriptors is obtained. The most informative features are selected through two iterative selectors, INCA and IReliefF, which determine the size of the selected feature subset automatically. Classification is performed with k-nearest neighbors (kNN) and support vector machine (SVM) classifiers, and the final decision is obtained through IMV and greedy outcome selection. Because every selected feature can be traced back to its input band, feature type, kernel and statistic, the framework provides transparent decision cues that clinicians can inspect, and its computational cost is quantified through runtime, latency, footprint and scaling analyses.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

1.3. Innovations and Contributions

Novelties:

- An expansion–contraction lattice pattern is proposed in this research, and this pattern is named ExCoPat (Expansion–Contraction Pattern). ExCoPat is a self-organized textural feature extractor in which the traversal path through a 28-node, nine-layer lattice with alternating expansion and contraction layers is determined per block by data-dependent node selection. This structure distinguishes ExCoPat from the static and conventional descriptors used in the existing pathological speech literature.
- A self-organized hybrid feature engineering model is proposed that integrates MDWT-based multilevel decomposition, ExCoPat-based textural feature extraction and statistical feature generation, two iterative feature selectors (INCA and IReliefF), two shallow classifiers (kNN and SVM), and IMV with greedy outcome selection.
- An explainability and efficiency analysis framework is introduced as an integral part of the model. It reports the provenance of the selected features by input band, feature type, kernel and statistic, class-wise feature profiles, the stability of the selected features across repeated runs, group ablation, and measured runtime, latency, footprint and scaling behaviour.

Contributions:

- There is a large and growing body of DL-based speech disorder classification models in the literature, whereas feature engineering approaches remain limited and self-organized pattern extractors are almost entirely absent. By proposing ExCoPat, we have provided a concrete and reproducible example of a self-organized lattice pattern that achieves discriminative power without any deep learning component.
- To the best of our knowledge, this is the first study that distinguishes dysarthria, spasmodic dysphonia and stuttering from normal speech within a single lightweight framework built exclusively on handcrafted features. By introducing a fully CPU-based pipeline with linear-time feature extraction and GPU-free inference, we have contributed a practically deployable solution for clinical environments lacking GPU infrastructure.
- By tracing every decision cue back to the signal component and encoding function that produced it, we have contributed a fully interpretable alternative to black-box DL models, providing clinicians with inspectable evidence to support multi-disorder speech screening.
- By benchmarking the proposed model against an artificial neural network and a shallow convolutional neural network trained on the same clips under the identical 10-fold protocol, we have shown that the ExCoPat-based handcrafted features outperform compact learned representations by 4.98 to 9.11 percentage points in accuracy.

2. Dataset

The dataset used in this study was curated from publicly available speech recordings on YouTube. Recordings were selected to represent four speech categories: dysarthria, normal speech, spasmodic dysphonia, and stuttering. For each category, videos containing clearly audible speech from individuals with the corresponding condition were identified, and only the audio tracks were retained for analysis. Recordings with excessive background music, overlapping speakers, or severe noise were excluded.

All audio files were converted to a uniform format before segmentation. Each recording was converted to a single (mono) channel and resampled to 44.1 kHz. The signals were then divided into three-second clips, following common practice in sound-based classification research. Segments shorter than three seconds but longer than one second were retained as single clips, whereas shorter fragments were discarded. In total, 5,244 labelled clips with a total duration of 261.55 min were obtained. The class distribution of the curated dataset is given in Table 1.

As shown in Table 1, the dataset is nearly balanced. The dysarthria class is the largest, with 1,429 clips (27.25%), and the stuttering class is the smallest, with 1,232 clips (23.49%). The ratio

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

between the largest and smallest classes is approximately 1.16. Therefore, no synthetic balancing method such as SMOTE or ADASYN was applied, and the natural acoustic characteristics of the recordings were preserved.

The proportional distribution of the clips across the four classes is further illustrated in Figure 1. As can be seen from the figure, the bar lengths are close to each other, and no class is markedly under-represented. This visual balance confirms that the classification results are not dominated by a single majority class. The steps followed to collect and prepare the clips are summarized in Figure 2.

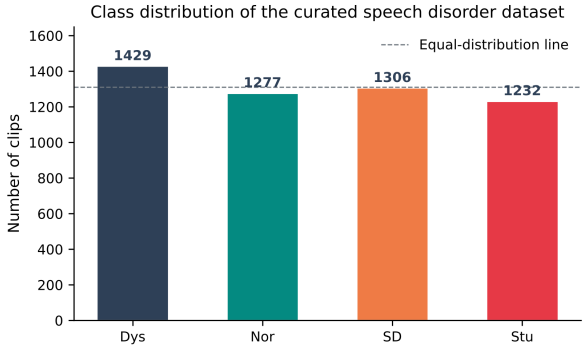


Figure 1. Class distribution of the curated speech disorder dataset.

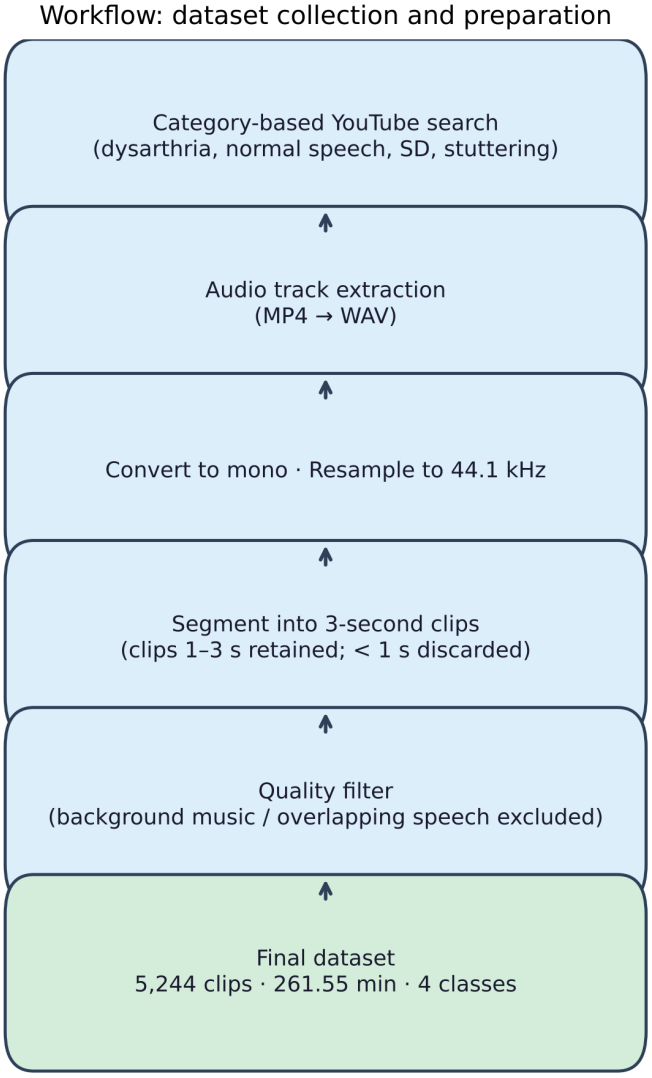


Figure 2. Workflow for collecting and preparing the speech disorder dataset.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

Table 1. Attributes of the curated speech disorder dataset.

Class	Number of clips	Percentage (%)	Time (total / average)
Dysarthria	1,429	27.25	71.38 min / 3.00 s
Normal speech	1,277	24.35	63.79 min / 3.00 s
Spasmodic dysphonia	1,306	24.90	65.15 min / 2.99 s
Stuttering	1,232	23.49	61.22 min / 2.98 s
Total	5,244	100	261.55 min / 2.99 s

3. The Proposed Model

The proposed model is a self-organized hybrid feature engineering framework that classifies speech recordings into four categories: dysarthria, normal speech, spasmodic dysphonia, and stuttering. As shown in Figure 3, the framework consists of four phases. In the first phase, each three-second clip is decomposed by an 11-level multilevel discrete wavelet transform (MDWT) using the db4 wavelet (Figure 4). The raw signal and the 11 low-pass sub-bands are fed in parallel to the proposed ExCoPat extractor and to statistical feature extractors, which yields a multilevel hybrid feature vector containing textural and statistical features. In the second phase, the most informative features are selected by INCA and IReliefF, which determine the size of the selected subset automatically. In the third phase, the selected feature sets are classified with kNN and SVM classifiers. In the final phase, the classifier outputs are fused by IMV, and the best outcome is selected by a greedy algorithm. The ExCoPat pattern is described in Section 3.1, and the complete feature engineering model is detailed in Section 3.2. The artificial neural network (ANN) and shallow convolutional neural network (CNN) baselines used for comparison are described in Section 3.3.

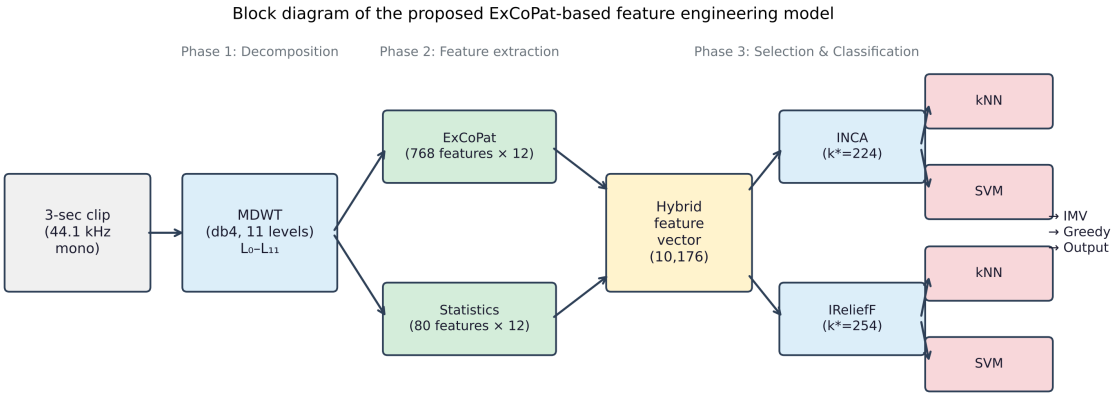


Figure 3. Block diagram of the proposed ExCoPat-based feature engineering model.

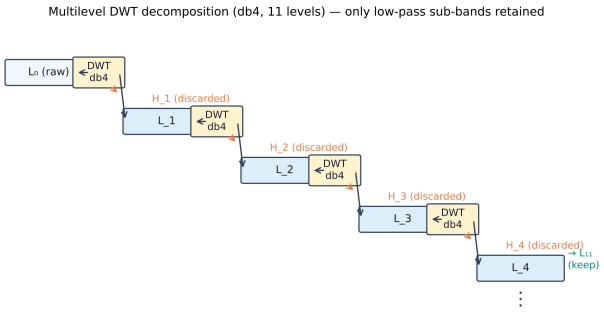


Figure 4. Multilevel discrete wavelet decomposition used to generate the low-pass sub-bands.

Figures 2–4 summarize the data preparation process and the structure of the proposed model. Figure 2 outlines the sequence of steps from the category-based search of YouTube recordings to the final corpus of 5,244 three-second clips. Figure 3 presents the four phases of the proposed model.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

Here, the raw clip follows a separate path to the feature extractors, and the notation box defines the intermediate outputs (L, f, s, c and v). Figure 4 details the decomposition step, in which only the low-pass sub-bands (L_1–L_{11}) are forwarded to feature generation, while the high-pass detail sub-bands (H) are discarded.

3.1. The Proposed ExCoPat Pattern

ExCoPat is a lattice-based textural feature extractor. Its lattice contains 28 nodes arranged in nine layers with 1-2-4-1-2-4-8-4-2 nodes, as illustrated in Figure 5. In the expansion layers, each node branches into two child nodes, and in the contraction layers several nodes converge into fewer nodes. The signal is divided into overlapping blocks of length 28, and the samples of each block are mapped onto the lattice nodes:

$$bl_i(j) = x(i+j-1), \quad i \in \{1,2,\ldots,N-27\}, \quad j \in \{1,2,\ldots,28\} \tag{1}$$

where x denotes the input signal of length N and bl_i is the i-th block. For each block, the absolute deviation of every sample from the global mean of the signal (μ_x) is computed:

$$d_i(j) = |bl_i(j) - \mu_x|, \quad \mu_x = \frac{1}{N}\sum_{n=1}^{N} x(n) \tag{2}$$

The traversal starts from the entry node and moves along the directed edges of the lattice in Figure 5. At each step, the next node is chosen only among the successors of the current node, and the successor with the minimum deviation is selected:

$$w(1) = 1, \quad w(l) = \arg\min_{j \in \mathcal{C}(w(l-1))} d_i(j), \quad l \in \{2,3,\ldots,9\} \tag{3}$$

where $\mathcal{C}(u)$ denotes the set of nodes connected to node u by an outgoing edge:

$$\begin{aligned}
&\mathcal{C}(1) = \{2,3\}, \quad \mathcal{C}(2) = \{4,5\}, \quad \mathcal{C}(3) = \{6,7\}, \quad \mathcal{C}(4) = \cdots = \mathcal{C}(7) = \{8\},\\
&\mathcal{C}(8) = \{9,10\}, \quad \mathcal{C}(9) = \{11,12\}, \quad \mathcal{C}(10) = \{13,14\}, \quad \mathcal{C}(11) = \{15,16\},\\
&\mathcal{C}(12) = \{17,18\}, \quad \mathcal{C}(13) = \{19,20\}, \quad \mathcal{C}(14) = \{21,22\}, \quad \mathcal{C}(15) = \mathcal{C}(16) = \{23\},\\
&\mathcal{C}(17) = \mathcal{C}(18) = \{24\}, \quad \mathcal{C}(19) = \mathcal{C}(20) = \{25\}, \quad \mathcal{C}(21) = \mathcal{C}(22) = \{26\},\\
&\mathcal{C}(23) = \mathcal{C}(24) = \{27\}, \quad \mathcal{C}(25) = \mathcal{C}(26) = \{28\}
\end{aligned} \tag{4}$$

Hence, the traversal never leaves the lattice edges. In the expansion layers, the current node has two successors and the choice between them depends on the block values, whereas in the contraction layers the successor is unique and the transition is fixed by the lattice topology. A worked example of this procedure is given in Figure 6.

Consecutive nodes along the path are compared with three kernel functions (signum, lower ternary and upper ternary), producing three eight-bit-binary words per block. These words are converted into three map signals, and a 256-bin histogram is extracted from each map, giving 768 textural features per input.

Formally, the directed difference between two consecutive path nodes is defined as:

$$\Delta_i(t) = bl_i(w(t)) - bl_i(w(t+1)), \quad t \in \{1,2,\ldots,8\} \tag{5}$$

The three kernel functions convert each difference into a binary value:

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

$$\beta_{1i}(t) = 1 \text{ if } \Delta_i(t) \geq 0; \quad 0 \text{ otherwise} \tag{6}$$

$$\beta_{2i}(t) = 1 \text{ if } \Delta_i(t) < -\tau; \quad 0 \text{ otherwise} \tag{7}$$

$$\beta_{3i}(t) = 1 \text{ if } \Delta_i(t) > \tau; \quad 0 \text{ otherwise} \tag{8}$$

where β_1, β_2 and β_3 correspond to the signum, lower ternary and upper ternary kernels, respectively, and the ternary threshold τ is set to half of the standard deviation (σ_x) of the input signal:

$$\tau = \sigma_x / 2 \tag{9}$$

The eight bits of each kernel are weighted by powers of two to obtain the map value of the block:

$$m_k(i) = \sum_{t=1}^{8} \beta_{ki}(t) \cdot 2^{8-t}, \quad k \in \{1, 2, 3\} \tag{10}$$

Each map signal is then summarized by counting the occurrences of its 256 possible codes:

$$h_k(v) = \sum_{i=1}^{N-27} \mathbf{1}[m_k(i) = v], \quad v \in \{0, 1, \ldots, 255\} \tag{11}$$

where $\mathbf{1}[\cdot]$ is the indicator function. Finally, the three histograms are concatenated into the textural feature vector of the input:

$$\mathbf{T} = [h_1 \| h_2 \| h_3] \in \mathbb{R}^{768} \tag{12}$$

where $\|$ denotes concatenation.

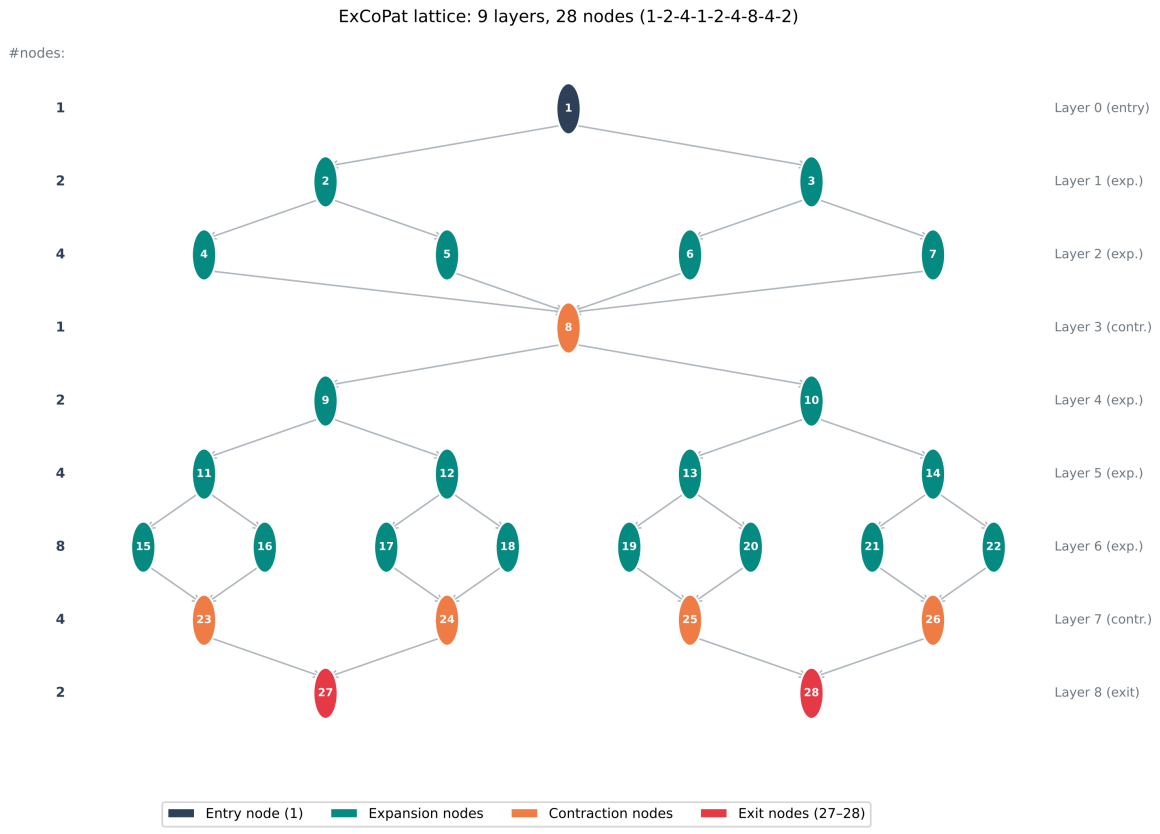


Figure 5. Graphical definition of the proposed ExCoPat extractor.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

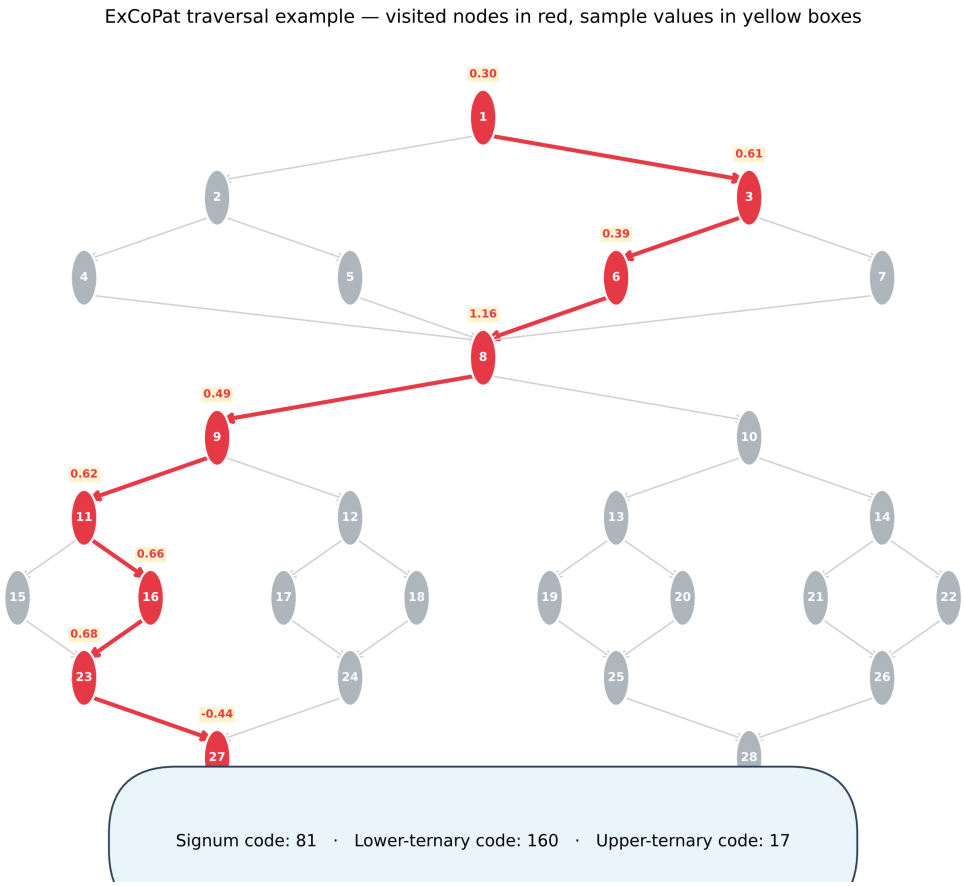


Figure 6. Illustrative example of the self-organized traversal in ExCoPat for a single 28-sample block. Starting from the entry node, the successor closest to the global signal mean is chosen at each step. The resulting route is drawn with red arrows, and the visited sample values are shown in yellow boxes.

The lattice topology that governs the ExCoPat traversal is illustrated in Figure 5. Each node is mapped to one sample position within a length-28 signal block, and the directed edges constrain how a traversal may progress from the entry node (index 1) to the exit layer (indices 27 and 28). The topology consists of two expansion–contraction stages. In the first stage, the entry node branches into two nodes (indices 2 and 3) and then into four nodes (indices 4–7), which converge into a single node (index 8). In the second stage, this node branches into two (indices 9 and 10), four (indices 11–14) and eight nodes (indices 15–22). These nodes then converge pairwise into four nodes (indices 23–26) and finally into the two exit nodes (indices 27 and 28). The node actually visited in each layer is not predetermined but is selected by the minimum-distance rule given in Equation (3). As a result, different blocks follow different routes through the same lattice. Each completed traversal visits nine nodes and generates eight directed comparisons, which are encoded by the three kernel functions and mapped onto 256-bin histograms.

3.1.1. Data-Adaptive Traversal

In ExCoPat, the descriptor of a block is not determined by the lattice topology alone. At each step, the node to be visited is selected by a data-dependent operator applied to the current block, so the same lattice can produce different traversal paths for different blocks. The expansion layers offer two candidate successors at every branching, and the contraction layers merge the branches back into fewer nodes, which allows the path to adapt to local signal variations of different widths within a single block. Formally, ExCoPat is a data-adaptive lattice-based binary pattern. The expansion–contraction terminology only describes the shape of the lattice and has no further meaning.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

3.1.2. Choice of the Block Length

The value 28 is not a free hyper-parameter but follows directly from the lattice geometry, since the nine layers contain $1+2+4+1+2+4+8+4+2=28$ nodes. Each traversal visits one node per layer and therefore yields eight directed comparisons, that is, exactly one eight-bit word per kernel and a 256-bin histogram. This matches the resolution of conventional one-dimensional local binary descriptors while keeping the histogram dense enough to be reliably estimated from a three-second segment. At 44.1 kHz, a 28-sample block covers only 0.63 ms of the raw signal. However, ExCoPat is also applied to the 11 low-pass sub-bands of the db4 MDWT. Because each decomposition level halves the sampling rate, the effective temporal support of the same 28-sample block roughly doubles at every level. As summarized in Table 2, the 12 inputs therefore cover analysis windows from 0.63 ms up to 1.58 s.

Table 2. Effective temporal support of a 28-sample ExCoPat block for each input (3-s clip, 44.1 kHz, db4 MDWT).

Input	Signal length (samples)	Effective support
Raw signal	132,300	0.63 ms
Level 1	66,153	1.41 ms
Level 2	33,080	2.95 ms
Level 3	16,543	6.03 ms
Level 4	8,275	12.20 ms
Level 5	4,141	24.54 ms
Level 6	2,074	49.21 ms
Level 7	1,040	98.55 ms
Level 8	523	197.23 ms
Level 9	265	394.60 ms
Level 10	136	789.34 ms
Level 11	71	1,578.82 ms

The extractor is therefore intrinsically multi-scale. The short-support inputs (raw signal to level 4) cover windows on the order of a single glottal cycle and capture cycle-to-cycle irregularities of phonation, such as those observed in dysarthric and dysphonic voices. The intermediate-support inputs (levels 5–8) span several pitch periods up to phoneme-length segments, where short voice breaks and strained phonation become visible. The long-support inputs (levels 9–11) cover syllable sequences, where repetitions, prolongations and blocks in stuttering, as well as the slowed articulation rate in dysarthria, unfold. In other words, temporal resolution is provided by the wavelet decomposition rather than by the block length, and a single fixed block length is sufficient precisely because it is applied across 12 scales.

3.2. The Proposed ExCoPat-Based Feature Engineering Model

This section formalizes the four phases of the proposed model shown in Figure 3.

3.2.1. Multilevel Wavelet Decomposition

As depicted in Figure 4, the raw clip is taken as the zeroth input, and each decomposition level splits the low-pass output of the previous level into a low-pass (L) and a high-pass (H) sub-band with the db4 wavelet:

$$L_0 = x \tag{13}$$

$$[L_q, H_q] = \mathrm{DWT}_{\mathrm{db4}}(L_{q-1}), \quad q \in \{1, 2, \ldots, 11\} \tag{14}$$

Only the low-pass sub-bands are retained, which gives the following input set for feature generation:

$$\mathcal{I} = \{L_0, L_1, \ldots, L_{11}\} \tag{15}$$

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

3.2.2. Statistical Feature Extraction

Let $\varphi(\cdot)$ denote an operator that returns 20 statistical measures of a signal: Tsallis, Rényi, Shannon, norm, log-energy, SURE, wavelet and threshold entropies, maximum, minimum, variance, mode, mean, kurtosis, sum of squared errors, range, energy, Katz and Higuchi fractal dimensions, and skewness. For each input L_q, direct statistical features are computed from the signal and from its first-order difference, $\nabla L_q(n) = L_q(n+1) - L_q(n)$:

$$S_q^{\text{dir}} = [\varphi(L_q) \| \varphi(\nabla L_q)] \in \mathbb{R}^{40} \tag{16}$$

In addition, the same operator is applied to the textural feature vector $\mathbf{T}_q$ and to the concatenated map signals $\mathbf{M}_q = [m_1 \| m_2 \| m_3]$ of that input, yielding the statistics of the textural features:

$$S_q^{\text{tex}} = [\varphi(\mathbf{T}_q) \| \varphi(\mathbf{M}_q)] \in \mathbb{R}^{40} \tag{17}$$

3.2.3. Hybrid Feature Vector

The textural and statistical features of each input are merged, and the 12 resulting vectors are concatenated into the final hybrid feature vector of a clip:

$$\mathbf{f}_q = [\mathbf{T}_q \| S_q^{\text{dir}} \| S_q^{\text{tex}}] \in \mathbb{R}^{848}, \quad q \in \{0, 1, \ldots, 11\} \tag{18}$$

$$\mathbf{F} = [\mathbf{f}_0 \| \mathbf{f}_1 \| \cdots \| \mathbf{f}_{11}] \in \mathbb{R}^{10,176} \tag{19}$$

Repeating this process for all M clips produces the feature matrix $\mathbf{X} \in \mathbb{R}^{M \times 10,176}$ with the label vector $\mathbf{y}$.

3.2.4. Feature Selection

Two iterative selectors, INCA and IReliefF, are used. For INCA, NCA is run $R = 5$ times on stratified subsamples containing 90% of the clips $(\mathbf{X}_r, \mathbf{y}_r)$, and the resulting weights are averaged to obtain a stable ranking. For IReliefF, the weights are computed with ReliefF on the full data:

$$\omega_{\text{INCA}} = \frac{1}{R} \sum_{r=1}^{R} \text{NCA}(\mathbf{X}_r, \mathbf{y}_r), \qquad \omega_{\text{IRF}} = \text{ReliefF}(\mathbf{X}, \mathbf{y}) \tag{20}$$

The features are sorted in descending order of weight, giving the ranking vector ρ_s for selector $s \in \{\text{INCA}, \text{IRF}\}$. For each candidate size k, the top-k features are evaluated with a 1-NN classifier (L_1 distance) under 10-fold cross-validation, and the misclassification rate is computed:

$$\varepsilon_s(k) = \frac{1}{M} \sum_{n=1}^{M} \mathbf{1}[\hat{y}_{s,k}^n \neq y_n], \quad k \in \{10, 11, \ldots, 256\} \tag{21}$$

where $\hat{y}_{s,k}^n$ is the cross-validated prediction of the n-th clip. The subset size with the minimum error is selected automatically, and the corresponding columns form the selected feature set:

$$k_s^* = \arg\min_k \varepsilon_s(k), \qquad S_s = \mathbf{X}(:, \rho_s(1 : k_s^*)) \tag{22}$$

3.2.5. Classification

Each selected feature set is classified with kNN ($k = 1$, L_1 distance, equal weights) and SVM (cubic kernel) under 10-fold cross-validation, which yields four classifier-wise predictions:

$$\begin{aligned} c_1 &= \text{kNN}(S_{\text{INCA}}, \mathbf{y}), \quad c_2 = \text{SVM}(S_{\text{INCA}}, \mathbf{y}), \\ c_3 &= \text{kNN}(S_{\text{IRF}}, \mathbf{y}), \quad c_4 = \text{SVM}(S_{\text{IRF}}, \mathbf{y}) \end{aligned} \tag{23}$$

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

3.2.6. Information Fusion and Final Decision

The classifier-wise predictions are ranked by accuracy, and π denotes the resulting ordering. IMV then combines the top-t predictions with the mode function, starting from the three best predictions:

$$v_{t-2}(n) = \text{mode}\{c_{\pi(1)}(n), c_{\pi(2)}(n), \ldots, c_{\pi(t)}(n)\}, \quad t \in \{3,4\} \tag{24}$$

This gives two voted predictions (v_1 and v_2). Finally, the greedy step evaluates the six candidate outcomes and selects the one with the highest accuracy as the final prediction:

$$\hat{\mathbf{y}} = \arg\max_{o \in \mathcal{O}} \text{Acc}(o, \mathbf{y}), \quad \mathcal{O} = \{c_1, c_2, c_3, c_4, v_1, v_2\} \tag{25}$$

3.3. Baseline Models

To position the proposed handcrafted pipeline against learned representations, two neural network baselines were trained on the same 5,244 clips: an ANN operating on a compact cepstral representation and a shallow CNN operating on log-mel spectrograms. Neither baseline uses ExCoPat, the MDWT sub-bands or the iterative feature selectors, so the comparison isolates the contribution of the proposed feature engineering. Both baselines share the front end described in Section 3.3.1, and their architectures are summarized in Table 3.

3.3.1. Input Representations

Each clip was loaded as a mono signal at 44.1 kHz and zero-padded to three seconds (132,300 samples) where necessary. A 128-band mel spectrogram was computed with a 2,048-point fast Fourier transform (46.4 ms Hann window) and a hop length of 512 samples (11.6 ms), and the power values were converted to decibels relative to the maximum of the clip. This yields a log-mel matrix of 128×259 (mel bands $\times$ frames), which is the input of the shallow CNN. For the ANN, 40 mel-frequency cepstral coefficients (MFCCs) were obtained from the log-mel matrix by the discrete cosine transform, together with their first-order temporal derivatives (delta coefficients). The mean and standard deviation of each coefficient over the 259 frames were concatenated into a 160-dimensional vector consisting of 40 MFCC means, 40 MFCC standard deviations, 40 delta means and 40 delta standard deviations.

3.3.2. Artificial Neural Network

The ANN is a fully connected feed-forward network with three hidden layers of 168, 64 and 32 neurons, followed by an output layer with four neurons, one per class. Each hidden layer applies an affine transformation followed by the rectified linear unit (ReLU) and dropout:

$$\mathbf{h}^{(0)} = \mathbf{z}, \quad \mathbf{h}^{(l)} = \delta(\text{ReLU}(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)})), \quad l \in \{1,2,3\} \tag{26}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weight matrix and bias vector of the l-th layer, $\text{ReLU}(a) = \max(0, a)$, and $\delta(\cdot)$ denotes dropout with a rate of 0.3, which is active only during training. The weight matrices have sizes 168×160, 64×168 and 32×64. The output layer maps the 32-dimensional representation to four class scores, and the predicted class is the one with the largest softmax probability:

$$\hat{y} = \arg\max_{c} \text{softmax}(\mathbf{W}^{(4)}\mathbf{h}^{(3)} + \mathbf{b}^{(4)})_c \tag{27}$$

The network has 40,076 trainable parameters, of which 27,048 belong to the first hidden layer (Table 3). The input vectors were standardized to zero mean and unit variance with statistics computed on the training portion of each fold only.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

3.3.3. Shallow Convolutional Neural Network

The shallow CNN consists of three convolutional blocks, a global average pooling (GAP) layer and a fully connected output layer. Each block applies a 3×3 convolution with stride 1 and zero padding of one pixel, batch normalization (BN), ReLU activation and 2×2 max pooling:

$$\mathbf{A}^{(b)} = \mathrm{MaxPool}_{2\times 2}(\mathrm{ReLU}(\mathrm{BN}(\mathbf{K}^{(b)} * \mathbf{A}^{(b-1)}))), \quad b \in \{1,2,3\} \tag{28}$$

where $\mathbf{A}^{(0)}$ is the single-channel log-mel input of size 128×259, $\mathbf{K}^{(b)}$ is the bank of convolution kernels of block b and $*$ denotes two-dimensional convolution. The three blocks contain 16, 32 and 64 filters, respectively. Because each pooling operation halves both dimensions, the feature maps shrink from 128×259 to 64×129, 32×64 and 16×32. GAP then reduces each of the 64 maps of the last block to a single value, and a dropout layer (rate 0.3) followed by a fully connected layer produces the four class scores:

$$\hat{y} = \arg\max_{c} \mathrm{softmax}(\mathbf{W}_o\, \delta(\mathrm{GAP}(\mathbf{A}^{(3)})) + \mathbf{b}_o)_c \tag{29}$$

Using GAP instead of flattening keeps the classification head small, so the network has only 23,780 trainable parameters, 18,496 of which belong to the third convolutional layer (Table 3). The log-mel inputs were standardized with the global mean and standard deviation of the training portion of each fold.

3.3.4. Training Protocol

Both baselines were implemented in PyTorch and trained with the cross-entropy loss and the AdamW optimizer (initial learning rate 10^{-3}, weight decay 10^{-4}). The learning rate was halved whenever the validation loss did not improve for six epochs (ANN) or three epochs (CNN). In each of the 10 cross-validation folds, 10% of the training portion was held out in a stratified manner as a validation set for early stopping, and the weights with the lowest validation loss were restored before testing. The ANN was trained with a batch size of 64 for at most 200 epochs and an early-stopping patience of 20 epochs, and the shallow CNN with a batch size of 32 for at most 60 epochs and a patience of 10 epochs. The baselines were evaluated with stratified 10-fold cross-validation (random seed 42) and with the same metrics as the proposed model (Section 4), and no information from the test fold was used for normalization, early stopping or model selection.

Table 3. Architectures of the baseline models.

Model	Layer	Configuration	Output shape	Parameters
ANN	Input	MFCC statistics (z-scored)	160	0
	Hidden 1	FC + ReLU + dropout (0.3)	168	27,048
	Hidden 2	FC + ReLU + dropout (0.3)	64	10,816
	Hidden 3	FC + ReLU + dropout (0.3)	32	2,080
	Output	FC + softmax	4	132
	Total			**40,076**
Shallow CNN	Input	Log-mel spectrogram (z-scored)	$1 \times 128 \times 259$	0
	Conv block 1	$16 \times (3 \times 3)$ conv + BN + ReLU + 2×2 max pool	$16 \times 64 \times 129$	192
	Conv block 2	$32 \times (3 \times 3)$ conv + BN + ReLU + 2×2 max pool	$32 \times 32 \times 64$	4,704
	Conv block 3	$64 \times (3 \times 3)$ conv + BN + ReLU + 2×2 max pool	$64 \times 16 \times 32$	18,624
	GAP	Global average pooling	64	0
	Output	Dropout (0.3) + FC + softmax	4	260
	Total			**23,780**

FC: fully connected layer; BN: batch normalization; GAP: global average pooling. Parameter counts of the convolutional blocks include the BN parameters.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

4. Experimental Results

The proposed model was implemented in Python 3.13 with NumPy, PyWavelets, scikit-learn and PyTorch, and all experiments were run on a workstation with a 48-core x86-64 processor and an NVIDIA RTX PRO 6000 GPU. The GPU was used only to accelerate the NCA and ReliefF weighting and the iterative subset search during training, whereas feature extraction and classification were executed on the CPU. The ANN and shallow CNN baselines were trained on the GPU, and their inference latency was measured on the CPU. All results were obtained with stratified 10-fold cross-validation on the 5,244 clips described in Section 2.

Performance was measured with accuracy (Acc), unweighted average recall (UAR), unweighted average precision (UAP), F1-score and geometric mean (GM). For a confusion matrix with C classes, where TP_c, FP_c, FN_c and TN_c denote the true positives, false positives, false negatives and true negatives of class c, these metrics are computed as:

$$\mathrm{Acc} = \frac{1}{M}\sum_{c=1}^{C}\mathrm{TP}_c \tag{30}$$

$$\mathrm{Rec}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}, \qquad \mathrm{UAR} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{Rec}_c \tag{31}$$

$$\mathrm{Pre}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c}, \qquad \mathrm{UAP} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{Pre}_c \tag{32}$$

$$\mathrm{F1} = \frac{2\cdot\mathrm{UAP}\cdot\mathrm{UAR}}{\mathrm{UAP}+\mathrm{UAR}}, \qquad \mathrm{GM} = \left(\prod_{c=1}^{C}\mathrm{Rec}_c\right)^{1/C} \tag{33}$$

In the class-wise analysis, specificity is additionally reported as $\mathrm{Spec} = \mathrm{TN}_c/(\mathrm{TN}_c + \mathrm{FP}_c)$. Because UAR, UAP and GM give equal weight to each class, they complement accuracy in the presence of the small imbalance reported in Table 1.

4.1. Feature Selection Results

The iterative search of both selectors is shown in Figure 7. INCA reached its minimum misclassification rate of 1.05% with 224 features, and its error fell below 5% with only 25 features. IReliefF converged more slowly and attained its minimum of 4.96% at 254 features, close to the upper limit of the search range. This difference indicates that the NCA weights concentrate the discriminative information into a much smaller subset than the ReliefF weights.

The stability of the INCA ranking is summarized in Figure 8. All five NCA runs converged within about 10 epochs, the weight rankings of different runs were highly consistent (mean Spearman correlation of 0.95), and the selected top-224 subsets overlapped with a mean Jaccard index of 0.67. In addition, 78.1% of the finally selected features were chosen in at least four of the five runs and 62.9% in all of them, which shows that the averaged ranking is not driven by a particular subsample.

The weights assigned to all 10,176 features are plotted in Figure 9. INCA produces a sparse profile in which a limited number of features from the raw signal and the lower sub-bands receive high weights, whereas IReliefF distributes moderate weights over most inputs. The selected features of both methods are concentrated in the raw signal and the first eight sub-bands, and the deepest sub-bands (L_9–L_{11}) contribute very little.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

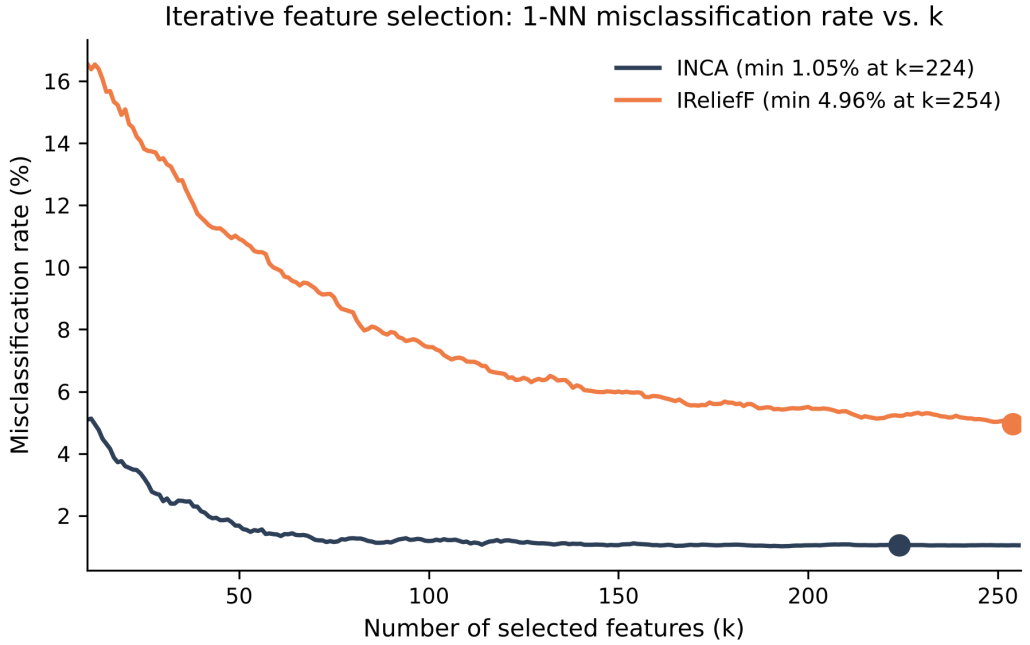


Figure 7. Misclassification rate of the 1-NN evaluator versus the number of selected features for INCA and IReliefF. The circles mark the automatically selected subset sizes.

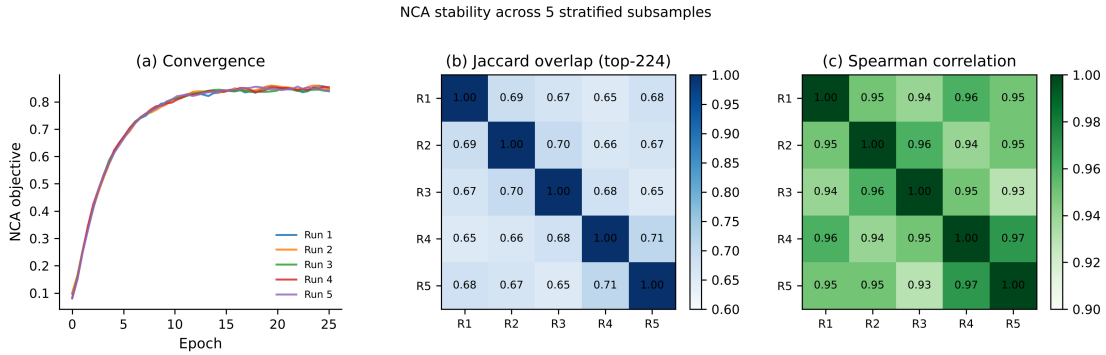


Figure 8. Stability of NCA across five stratified subsamples: convergence of the objective (left), Jaccard overlap of the top-224 subsets (middle) and Spearman correlation of the feature weights (right).

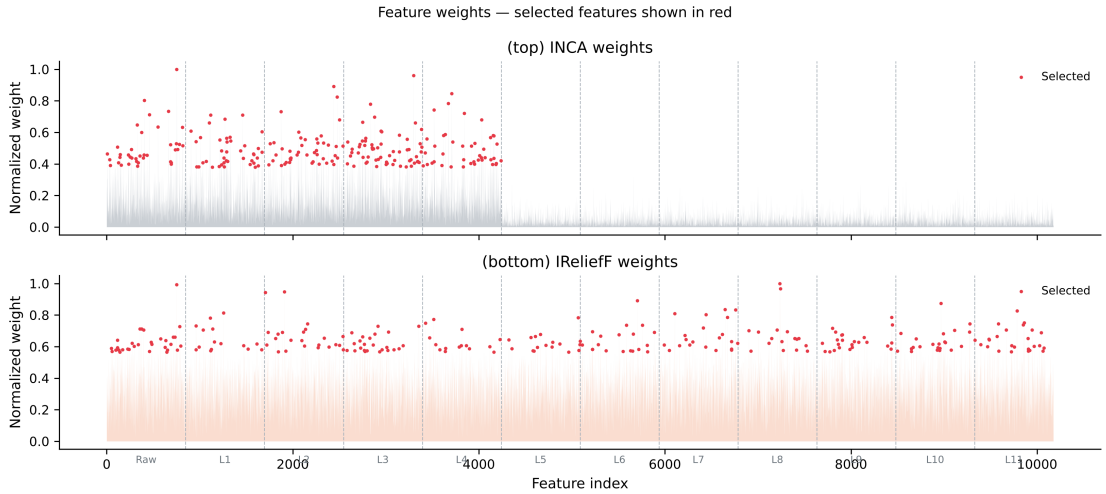


Figure 9. Feature weights produced by INCA (top) and IReliefF (bottom) across the 12 inputs. Selected features are shown in red.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

4.2. Classifier-Wise Results

The results of the four selector–classifier combinations are listed in Table 4. The INCA + kNN combination yielded the best result with 98.95% accuracy, followed by INCA + SVM with 97.77%. Both IReliefF-based outcomes remained around 95%, so the choice of selector affected performance more than the choice of classifier.

Table 4. Classifier-wise results. The best result is shown in bold.

Selector	k	Classifier	Acc (%)	UAR (%)	UAP (%)	F1 (%)	GM (%)
INCA	224	kNN	**98.95**	**98.92**	**98.96**	**98.94**	**98.92**
INCA	224	SVM	97.77	97.71	97.76	97.74	97.71
IReliefF	254	kNN	95.04	94.92	95.04	94.98	94.89
IReliefF	254	SVM	95.44	95.35	95.39	95.37	95.34

The confusion matrices of these outcomes are given in Figure 10. The IReliefF-based models mainly confused stuttering with spasmodic dysphonia (67 clips with kNN and 40 clips with SVM), whereas this confusion decreased to 13 clips with INCA + kNN.

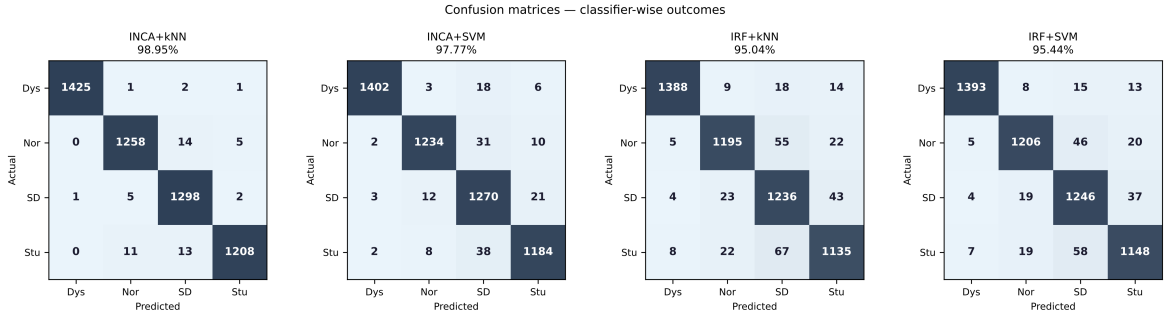


Figure 10. Confusion matrices of the four classifier-wise outcomes.

4.3. Voted and Final Results

The outcomes generated by IMV are presented in Table 5. Voting the three best outcomes produced 98.47% accuracy, and adding the fourth outcome reduced it to 98.11%. Neither voted result exceeded INCA + kNN, because the additional members were considerably less accurate and introduced more errors than they corrected. Accordingly, the greedy step selected INCA + kNN as the final outcome, as shown in Figure 11.

Table 5. Voted results produced by IMV.

Voted outcome	Members	Acc (%)	UAR (%)	UAP (%)	F1 (%)	GM (%)
Top-3	INCA + kNN, INCA + SVM, IReliefF + SVM	98.47	98.43	98.47	98.45	98.42
Top-4	All four outcomes	98.11	98.04	98.12	98.08	98.03

The final model achieved 98.95% accuracy, 98.92% UAR, 98.96% UAP, 98.94% F1-score and 98.92% GM. Its confusion matrix is shown in Figure 12. Only 55 of the 5,244 clips were misclassified. The most frequent errors were normal speech predicted as spasmodic dysphonia (14 clips) and stuttering predicted as spasmodic dysphonia (13 clips), while dysarthria was almost never confused with the other classes.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

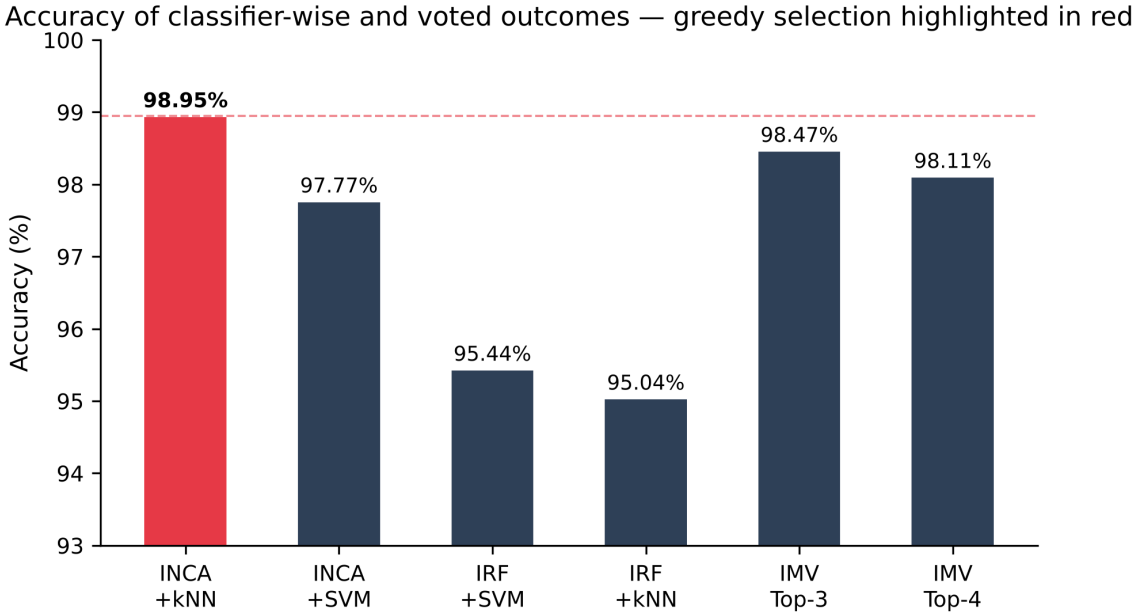


Figure 11. Accuracy of the classifier-wise and voted outcomes. The outcome selected by the greedy algorithm is highlighted.

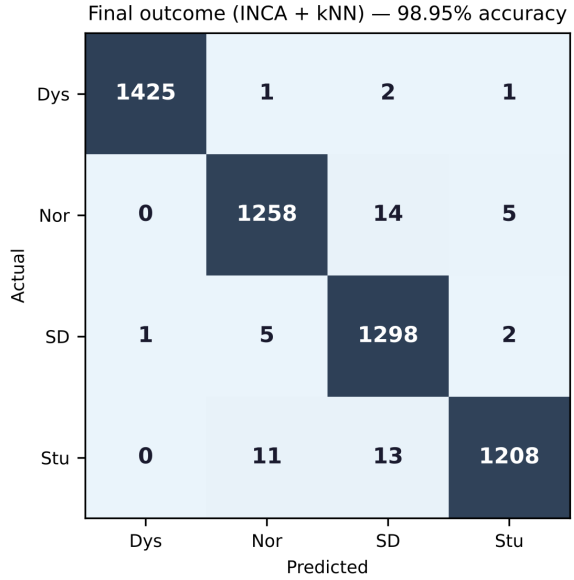


Figure 12. Confusion matrix of the final outcome (INCA + kNN).

The class-wise results are given in Table 6 and Figure 13. All classes exceeded 97.8% in every metric. Dysarthria obtained the highest F1-score (99.51%). Spasmodic dysphonia had the lowest precision (97.81%) because it absorbed most of the errors of the other classes, and stuttering had the lowest recall (98.05%), which is consistent with the confusion pattern in Figure 12.

Table 6. Class-wise results of the final outcome.

Class	Precision (%)	Recall (%)	Specificity (%)	F1 (%)
Dysarthria	99.30	99.72	99.74	99.51
Normal speech	99.53	98.51	99.85	99.02
Spasmodic dysphonia	97.81	99.39	99.26	98.59
Stuttering	99.18	98.05	99.75	98.61

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

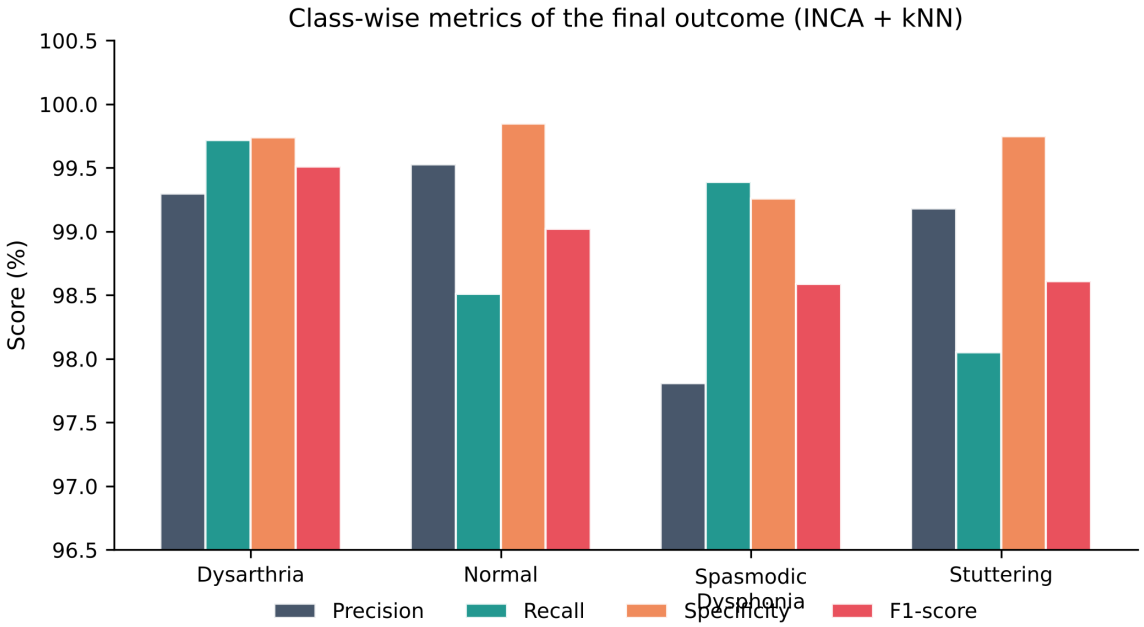


Figure 13. Class-wise precision, recall, specificity and F1-score of the final outcome.

4.4. Explainability Analysis

Because every feature index can be traced back to its input, feature type, kernel and statistic, the selected subsets were examined according to their provenance (Figure 14). Of the 224 features selected by INCA, 175 (78.1%) were ExCoPat textural features, 27 were direct statistical features and 22 were statistics of the textural features. Most textural features came from the signum kernel (135 of 175). The raw signal provided 50 features, the sub-bands L_1–L_4 provided 125 features, L_5–L_8 provided 47 features, and L_9–L_{11} contributed only two. IReliefF selected relatively more statistical features (90 of 254) and relied more heavily on the raw signal (113 features).

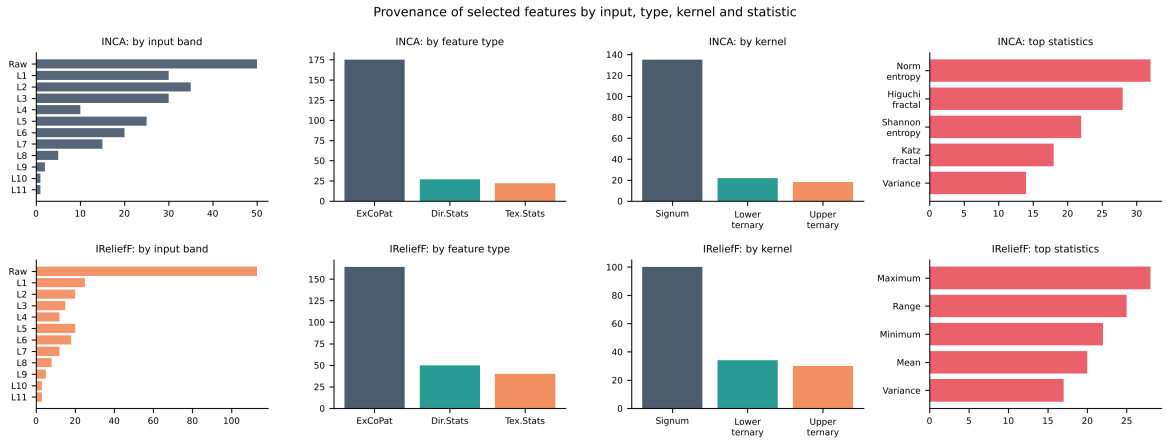


Figure 14. Provenance of the features selected by INCA (top) and IReliefF (bottom) by input, feature type, ExCoPat kernel and statistic.

The 20 highest-weighted features of each selector are shown in Figure 15. The INCA list is led by entropy and fractal measures computed on the intermediate sub-bands, such as the norm entropy of the first-order difference of L_7 and the Higuchi fractal dimension of the ExCoPat code maps of L_7 and L_8, followed by ExCoPat signum codes of the raw signal. In contrast, the IReliefF list is dominated by amplitude statistics (maximum, range and minimum) of the raw signal and the first three sub-bands.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

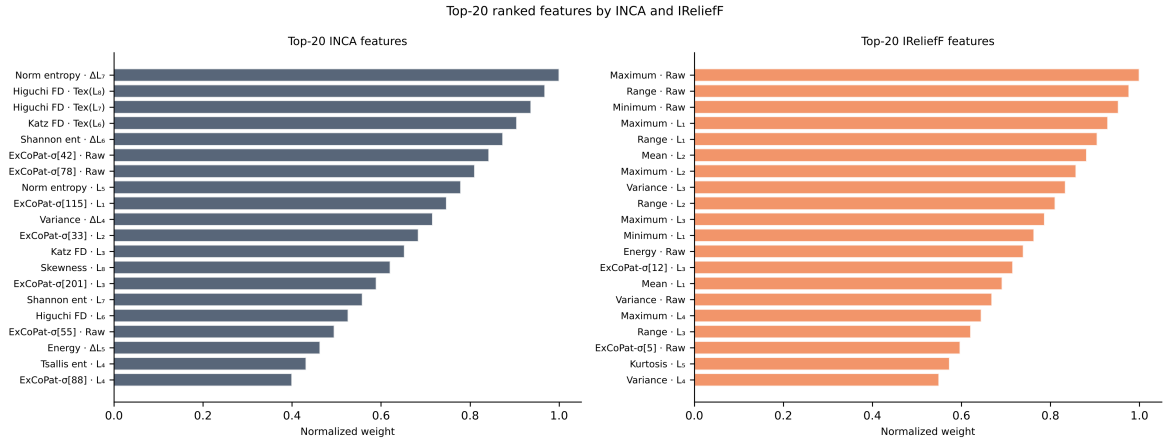
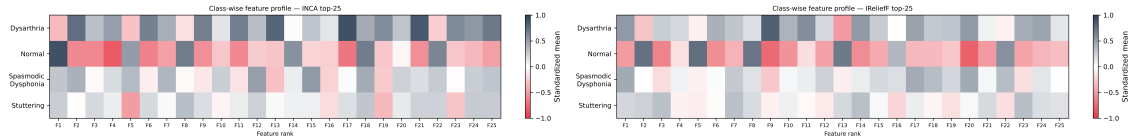


Figure 15. Top-20 features ranked by INCA (left) and IReliefF (right).

Figure 16 presents the class-wise mean values of the 25 highest-ranked features. Normal speech displays a clearly different profile, with below-average values for many of the features that are above average for dysarthria. Spasmodic dysphonia and stuttering show weaker but distinct deviations, which explains why these two classes account for most of the remaining confusions.



(**a**) INCA (**b**) IReliefF

Figure 16. Class-wise standardized mean values of the top-25 features selected by INCA and IReliefF. Red and blue indicate above- and below-average values, respectively.

The contribution of each feature group was measured by removing the group from the selected subset and repeating the evaluation (Figure 17). Removing the ExCoPat textural features caused the largest drop, 11.27 percentage points for INCA and 12.66 percentage points for IReliefF. The signum kernel was the most influential kernel (4.44 and 9.08 percentage points), and the raw signal was the most influential input (1.07 and 4.06 percentage points). Removing any single sub-band reduced accuracy by less than 0.65 percentage points, which shows that the information is distributed across scales rather than concentrated in one sub-band.

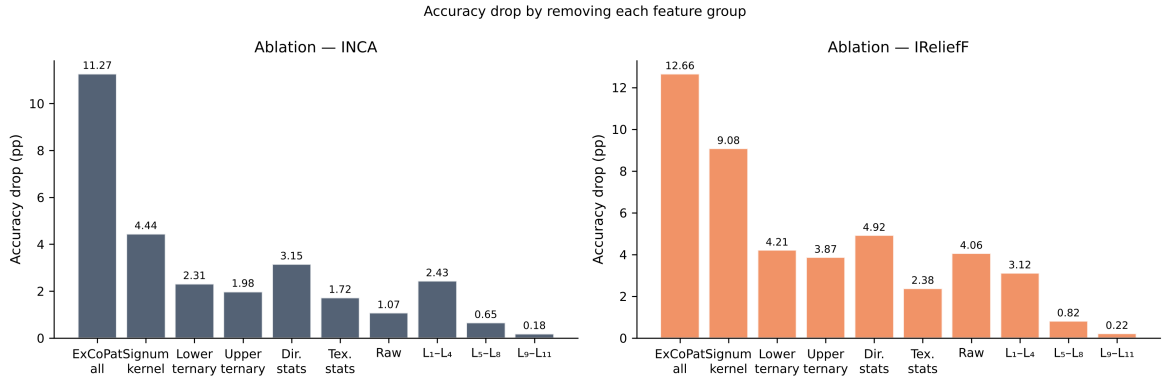


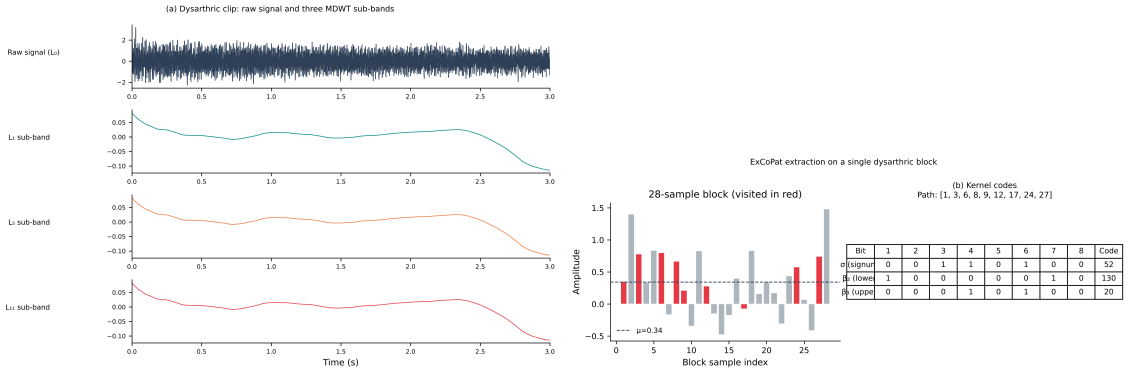
Figure 17. Accuracy drop obtained by removing each feature group from the subsets selected by INCA (left) and IReliefF (right).

Figure 18 illustrates the extraction stage on a real dysarthric clip. The MDWT sub-bands preserve the temporal envelope of the utterance while progressively reducing the signal length (Figure 18a),

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

and the ExCoPat traversal on a single 28-sample block follows the lattice edges and yields the signum, lower-ternary and upper-ternary codes 224, 14 and 32, respectively (Figure 18b).



(**a**) Sub-bands (**b**) ExCoPat traversal

Figure 18. Extraction stage on a real dysarthric clip, showing the raw signal and selected sub-bands alongside the ExCoPat traversal on a single 28-sample block.

4.5. Computational Cost

The runtime measurements are summarized in Table 7 and Figure 19. Extracting the 10,176 features of one three-second clip took 433.1 ms on the CPU, of which ExCoPat accounted for 200.9 ms, the statistical features for 230.1 ms and the MDWT for only 1.9 ms. Inference with INCA + kNN required 0.20 ms per clip. The resulting real-time factor of 0.14 means that a clip is processed about seven times faster than its duration. The scaling test confirmed that the extraction time grows linearly with the signal length ($R^2 = 0.998$), at approximately 41 ms per second of audio.

Table 7. Computational cost of the proposed model.

Item	Value
MDWT decomposition (per clip)	1.9 ms
ExCoPat, 12 inputs (per clip)	200.9 ms
Statistical features, 12 inputs (per clip)	230.1 ms
Total feature extraction (per clip)	433.1 ms
Inference, INCA + kNN (per clip)	0.20 ms
Real-time factor	0.14
Linear fit of extraction time vs. signal length (R^2)	0.998
Feature extraction, whole dataset	78.4 s
NCA weighting, five runs	242.1 s
Classifier training per fold (kNN / SVM)	0.005 s / 0.23 s
Trainable parameters of the feature extractor	0
Selected features (INCA / IReliefF)	224 / 254

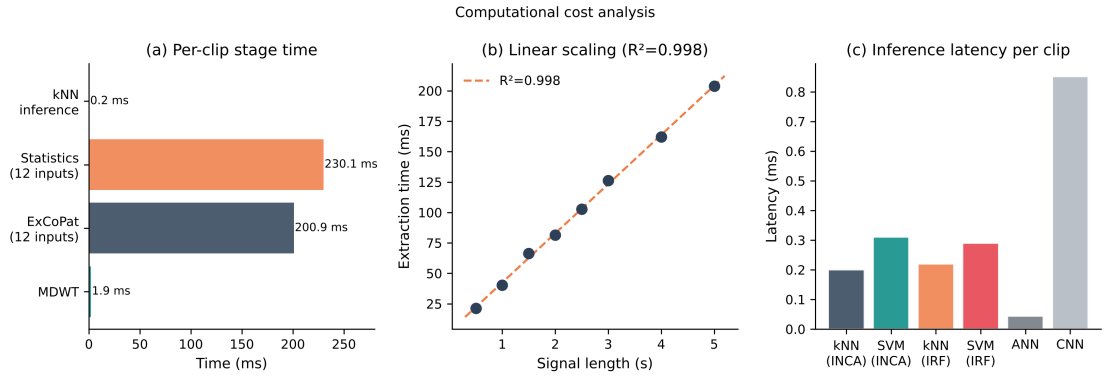


Figure 19. Per-clip processing time by stage (left), extraction time versus signal length (middle) and inference latency of the classifiers (right).

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

The one-time training cost of each stage is shown in Figure 20. The five NCA runs dominated the training time (242.1 s), followed by feature extraction for the entire dataset (78.4 s), whereas the iterative searches, IMV and classifier training each took less than 2.5 s.

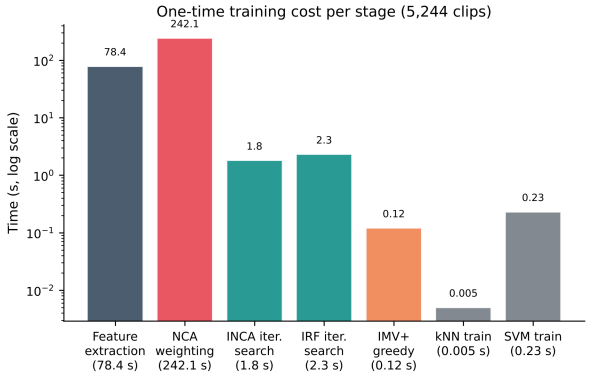


Figure 20. One-time training cost of each stage for the 5,244 clips (logarithmic scale).

4.6. Comparison with Neural Network Baselines

The results of the two baselines are compared with those of the proposed model in Table 8. Under the same 10-fold protocol, the ANN reached 93.97% accuracy, 93.84% UAR and 93.88% F1-score, and the shallow CNN reached 89.84% accuracy, 89.62% UAR and 89.69% F1-score. The proposed model therefore outperformed the ANN by 4.98 and the shallow CNN by 9.11 percentage points in accuracy, and gaps of similar size were observed for every metric. The fold-wise accuracies were $93.97 \pm 1.45\%$ for the ANN (range 92.19–96.76%) and $89.84 \pm 1.57\%$ for the shallow CNN (range 87.79–91.81%), so the differences between the models are considerably larger than the variation between folds.

Table 8. Comparison of the proposed model with the neural network baselines (10-fold CV). The best result is shown in bold.

Model	Input	Params	Acc (%)	UAR (%)	UAP (%)	F1 (%)	GM (%)
Proposed (ExCoPat + INCA + kNN)	224 selected hybrid features	0*	**98.95**	**98.92**	**98.96**	**98.94**	**98.92**
ANN (168-64-32)	160 MFCC statistics	40,076	93.97	93.84	93.92	93.88	93.80
Shallow CNN (3 conv blocks)	128×259 log-mel	23,780	89.84	89.62	89.75	89.69	89.49

Params: trainable parameters. *The ExCoPat extractor has no trainable parameters, and the kNN classifier stores the training samples instead of learning weights.

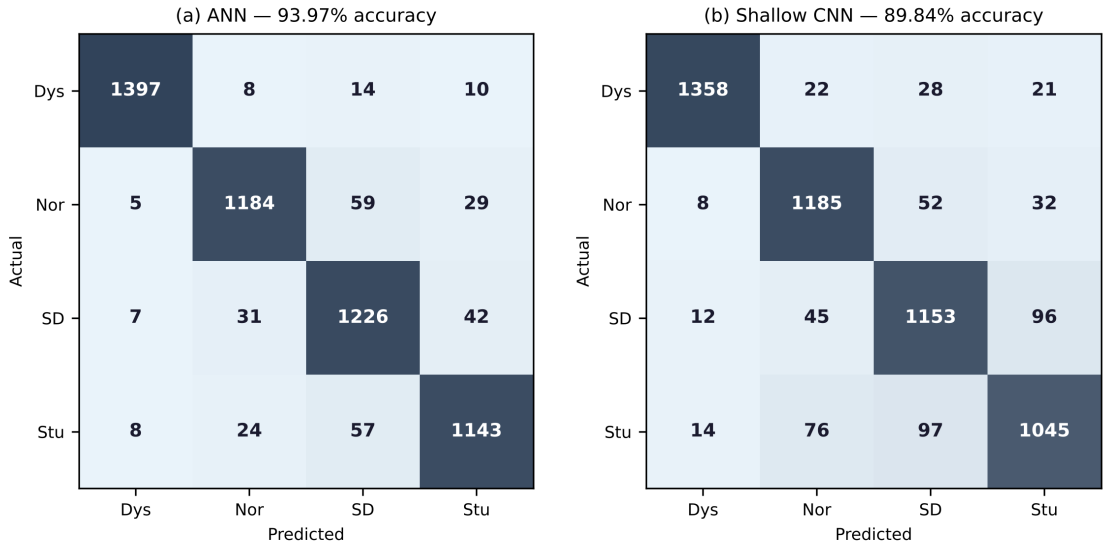
The class-wise recall and F1-scores of the three models are listed in Table 9, and the confusion matrices of the baselines are shown in Figure 21. The ANN misclassified 316 clips and the shallow CNN 533 clips, compared with 55 clips for the proposed model. Like the proposed model, both baselines recognized dysarthria best (F1-scores of 97.39% and 94.40%), and both had the most difficulty with spasmodic dysphonia and stuttering. For the ANN, the largest confusions were normal speech predicted as spasmodic dysphonia (59 clips) and stuttering predicted as spasmodic dysphonia (57 clips). The shallow CNN showed the same pattern more strongly: stuttering was predicted as spasmodic dysphonia in 97 clips and as normal speech in 76 clips, and its recall for stuttering fell to 82.39%. Hence, the class pairs that remain difficult for the proposed model are also the most difficult for the baselines, but the ExCoPat-based features reduce these confusions substantially; for example, only 13 stuttering clips were predicted as spasmodic dysphonia by the proposed model, compared with 57 by the ANN and 97 by the shallow CNN.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

Table 9. Class-wise recall and F1-score of the proposed model and the baselines.

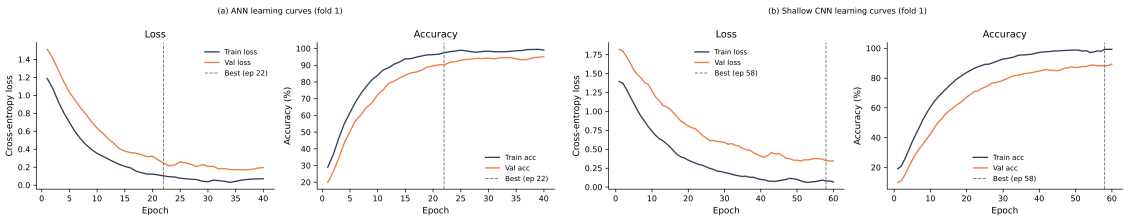
Class	Proposed Rec (%)	Proposed F1 (%)	ANN Rec (%)	ANN F1 (%)	CNN Rec (%)	CNN F1 (%)
Dysarthria	99.72	99.51	97.76	97.39	95.03	94.40
Normal speech	98.51	99.02	92.72	93.71	92.80	91.75
Spasmodic dysphonia	99.39	98.59	93.87	92.18	88.28	86.76
Stuttering	98.05	98.61	90.99	92.19	82.39	85.65



(**a**) ANN (**b**) Shallow CNN

Figure 21. Confusion matrices of the ANN and shallow CNN baseline models under 10-fold cross-validation.

The learning curves of the first fold are shown in Figure 22. The ANN converged quickly: its validation accuracy exceeded 90% within ten epochs, and early stopping selected epochs between 12 and 37 across the folds. Its training accuracy, however, approached 100% while the validation accuracy levelled off at about 94–95%, indicating that the network fits the training clips more closely than the MFCC statistics allow it to generalize. The shallow CNN learned more slowly, and its validation curve fluctuated strongly during the first 35 epochs before stabilizing. Its best epoch fell between 56 and 60 in all folds, that is, at or near the 60-epoch limit, so a longer training schedule could raise its accuracy somewhat. Nevertheless, the validation accuracy of the first fold had already flattened at about 88–89%, far below that of the proposed model.



(**a**) ANN (**b**) Shallow CNN

Figure 22. Learning curves (loss and accuracy) of the first fold for the ANN and shallow CNN. The dashed lines mark the epoch with the lowest validation loss.

Both baselines are compact and fast at inference. The ANN has 40,076 and the shallow CNN 23,780 trainable parameters; training over all 10 folds took 24.2 s and 358.2 s on the GPU, and the CPU

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

inference time of the networks was 0.044 ms and 0.852 ms per clip, respectively. These times exclude the computation of the MFCC and log-mel inputs. The proposed model required 0.20 ms per clip for kNN inference, and its 433.1 ms feature extraction makes its total per-clip cost higher than that of the network forward passes (Table 7), although it still runs about seven times faster than real time. In exchange, it provides 4.98 to 9.11 percentage points higher accuracy, has no trainable parameters in the feature extractor, and its decisions can be traced back to specific inputs, kernels and statistics (Section 4), which neither baseline offers.

5. Discussion

The results show that a fully handcrafted pipeline can separate dysarthria, spasmodic dysphonia, stuttering and normal speech with high accuracy. Three observations explain this performance. First, the ablation analysis identifies ExCoPat as the main source of discriminative information, since its removal causes the largest accuracy loss for both selectors. Second, the selected features are concentrated in the raw signal and the sub-bands L_1–L_8, whose effective supports range from a single glottal cycle to phoneme-length segments (Table 2). This agrees with the motivation of the multi-scale design, because the irregular phonation of dysarthria and spasmodic dysphonia and the disfluencies of stuttering are expressed at these time scales. Third, INCA outperformed IReliefF because the NCA objective directly optimizes nearest-neighbor accuracy, which matches the 1-NN evaluator and the kNN classifier used in the final outcome.

The fusion stage did not improve on the best single outcome. This does not contradict the design of the model: the greedy step is included precisely to retain a single outcome when voting brings no benefit, and in this study it correctly kept INCA + kNN.

The comparison with the neural network baselines (Section 4) supports the motivation of this work. Although the ANN and the shallow CNN were trained on the same clips and evaluated with the same protocol and metrics, they remained 4.98 and 9.11 percentage points below the proposed model. Both baseline inputs describe the short-time spectral envelope, whereas ExCoPat encodes the local ordering of samples at 12 temporal scales, from a single glottal cycle to syllable sequences (Table 2). The considerably poorer separation of stuttering and spasmodic dysphonia by the baselines suggests that this multi-scale textural information is particularly useful for these two classes. The comparison should nevertheless be read with two caveats. First, the baselines were trained with a single configuration and were not extensively tuned. Second, the feature selection of the proposed model was performed on the whole dataset, whereas the baselines used only training-fold information for normalization and early stopping (Section 5). The size of the gap should therefore be re-examined under a strictly nested protocol.

5.1. Comparison with Previous Studies

A direct comparison is not possible because, to the best of our knowledge, no previous study has addressed the same four-class problem on a common dataset. Table 10 therefore summarizes representative studies on the same disorders, including two recent works [34,35], together with their datasets and validation protocols. The proposed model reached accuracy comparable to the strongest results in the table while solving a four-class problem with a parameter-free feature extractor and a CPU-based pipeline. The table also shows that the validation protocol strongly affects the reported performance. For example, Zhou et al. [34] used a patient-level hold-out split on clinical recordings and obtained 71.60% accuracy for the three-class discrimination of healthy voices, muscle tension dysphonia and spasmodic dysphonia, whereas most of the higher values were obtained with record-level cross-validation. The comparison should therefore be interpreted with caution.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

Table 10. Comparison with representative studies.

Study	Dataset	Task (classes)	Method	Validation protocol	Acc (%)
Cesarini et al. [9]	111 HC, 51 PD, 60 AdSD	HC / PD / AdSD (3)	openSMILE + GA + NB	10-fold CV	99.46
Gupta et al. [15]	UA-Speech	Dysarthria severity (4)	ResNet	90/10 train-test split	98.90
Disken [28]	Saarbruecken Voice Database (/a/)	Multi-label voice disorders	SincNet + gated residual CNN	10-fold stratified CV	99.00
Prabhu and Seliya [31]	SEP-28k	Stuttering types (5)	CNN	Random 80/10/10 split	95.00
Zhou et al. [34]	1,597 clinical /a/ samples	HC / MTD / SD (3)	Log-Mel + CNN ensemble	Patient-level hold-out (3:1:1)	71.60
Duan et al. [35]	THE-POSSD (4,290 samples, 68 subjects)	Control / Mild / Moderate / Severe (4)	M-DBResNet	Stratified 5-fold CV	96.21
Lillini et al. [36]	HeyJay! (36 speakers, 8,669 utterances)	Dysarthria severity: low / moderate / severe (3)	Frozen wav2vec 2.0 XLSR-53 + MLP	Speaker-independent 5-fold CV	64.60
Gupta et al. [37]	Saarbruecken Voice Database (9 diagnoses)	Normal + 3 disorder groups (4)	UniSpeech-SAT features + FF classifier	Patient-level 70/10/20 split	67.20
Wahab Sait et al. [38]	Saarbruecken Voice Database (2,043 recordings)	Healthy / pathological (2)	1D-CNN + ViT (Linformer) + CaiT	Subject-level 70/15/15 split	97.50
Alnamazi et al. [39]	Arabic clinical stuttering corpus (48 speakers, 7,200 clips)	Fluent / disfluent (2)	Fine-tuned Whisper-Large + linear head	Speaker-level 80/10/10 split	81.00
Proposed	5,244 YouTube clips	Dysarthria / normal / SD / stuttering (4)	MDWT + ExCoPat + INCA + kNN	10-fold CV (clip-level)	**98.95**

HC: healthy controls; PD: Parkinson's disease; AdSD: adductor spasmodic dysphonia; MTD: muscle tension dysphonia; SD: spasmodic dysphonia; GA: genetic algorithm; NB: naïve Bayes; CV: cross-validation.

5.2. Limitations and Future Work

Several limitations should be noted. The recordings were collected from YouTube, and each clip was stored as a separate file without speaker or video identifiers, so clips from the same speaker may appear in both training and test folds. In addition, the feature selectors were applied to the whole dataset before cross-validation. Both factors can make the reported performance optimistic. Future work will therefore apply speaker-independent validation with selection performed inside each training fold and will test the model on an external clinical dataset. In addition, the ANN and shallow CNN baselines were trained with a single, untuned configuration, and the shallow CNN reached its lowest validation loss at or near the 60-epoch limit; deeper or pretrained networks and longer training schedules may perform better and will be included in future comparisons. The four-class setting could also be extended to other speech disorders and to severity grading.

6. Conclusions

This study introduced ExCoPat, a data-adaptive expansion–contraction lattice pattern, and built a self-organized feature engineering model around it for the classification of dysarthria, spasmodic dysphonia, stuttering and normal speech. On a curated dataset of 5,244 clips, the model achieved 98.95% accuracy with all class-wise metrics above 97.8%, outperforming an ANN trained on MFCC statistics (93.97%) and a shallow CNN trained on log-mel spectrograms (89.84%) under the same protocol. Its decisions can be traced to specific inputs, kernels and statistics, and its CPU-based feature extraction runs about seven times faster than real time without any trainable parameters. These properties make the proposed model a promising and transparent tool for multi-disorder speech screening, pending validation under speaker-independent conditions.

Author Contributions: Conceptualization, Z.H.K., A.K. and B.Ç.; methodology, Z.H.K., A.K. and B.Ç.; software, B.Ç.; validation, Z.H.K., A.K. and B.Ç.; formal analysis, Z.H.K., A.K. and B.Ç.; investigation, Z.H.K., A.K. and B.Ç.; resources, Z.H.K. and A.K.; data curation, B.Ç.; writing—original draft preparation, Z.H.K., A.K. and B.Ç.; writing—review and editing, Z.H.K., A.K. and B.Ç.; visualization, B.Ç.; supervision, Z.H.K. and A.K.; project administration, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by Kocaeli University Technopark.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

Data Availability Statement: We encourage all authors of articles published in MDPI journals to share their research data. In this section, please provide details regarding where data supporting reported results can be found.

Acknowledgments: During the preparation of this manuscript, the authors used Claude (Opus 5.5, Medium) for language revision. The authors reviewed and edited the output and take full responsibility for the content of this publication. The authors would also like to thank Erçin Dinçer for his valuable support in securing the funding that made this research possible.

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

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial neural network
BN	Batch normalization
CNN	Convolutional neural network
ExCoPat	Expansion–Contraction Pattern
GAP	Global average pooling
GM	Geometric mean
INCA	Iterative neighborhood component analysis
IMV	Iterative majority voting
IReliefF	Iterative ReliefF
kNN	k-nearest neighbors
MDWT	Multilevel discrete wavelet transform
MFCC	Mel-frequency cepstral coefficient
NCA	Neighborhood component analysis
SD	Spasmodic dysphonia
SVM	Support vector machine
UAP	Unweighted average precision
UAR	Unweighted average recall

References

1. Joshy, A.A.; Rajan, R. Automated Dysarthria Severity Classification: A Study on Acoustic Features and Deep Learning Techniques. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**. https://doi.org/10.1109/TNSRE.2022.3169814
2. Ludlow, C.L., et al. Consensus-Based Attributes for Identifying Patients With Spasmodic Dysphonia and Other Voice Disorders. *JAMA Otolaryngol. Head Neck Surg.* **2018**. https://doi.org/10.1001/jamaoto.2018.0644
3. Suppa, A., et al. Voice analysis in adductor spasmodic dysphonia: Objective diagnosis and response to botulinum toxin. *Parkinsonism Relat. Disord.* **2020**. https://doi.org/10.1016/j.parkreldis.2020.03.012
4. Yeung, W.; Richards, A.L.; Novakovic, D. Botulinum Neurotoxin Therapy in the Clinical Management of Laryngeal Dystonia. *Toxins* **2022**. https://doi.org/10.3390/toxins14120844
5. Sheikh, S.A.; Sahidullah, M.; Hirsch, F.; Ouni, S. Machine learning for stuttering identification: Review, challenges and future directions. *Neurocomputing* **2022**. https://doi.org/10.1016/j.neucom.2022.10.015
6. Schlotthauer, G.; Torres, M.E.; Jackson-Menaldi, M.C. A Pattern Recognition Approach to Spasmodic Dysphonia and Muscle Tension Dysphonia Automatic Classification. *J. Voice* **2010**. https://doi.org/10.1016/j.jvoice.2008.10.007
7. Snekhalatha, U.; Shamila Rachel, S.; Thulasi Rajalakshmi. Automated speech signal analysis based on feature extraction and classification of spasmodic dysphonia: a performance comparison of different classifiers. *Int. J. Speech Technol.* **2017**. https://doi.org/10.1007/s10772-017-9471-8
8. Calà, F., et al. Machine Learning Assessment of Spasmodic Dysphonia Based on Acoustical and Perceptual Parameters. *Bioengineering* **2023**. https://doi.org/10.3390/bioengineering10040426
9. Cesarini, V., et al. Voice Disorder Multi-Class Classification for the Distinction of Parkinson's Disease and Adductor Spasmodic Dysphonia. *Appl. Sci.* **2023**. https://doi.org/10.3390/app13158562
10. Karabayir, I.; Goldman, S.M.; Pappu, S.; Akbilgic, O. Gradient boosting for Parkinson's disease diagnosis from voice recordings. *BMC Med. Inform. Decis. Mak.* **2020**. https://doi.org/10.1186/s12911-020-01250-7

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

11. Afroz, F.; Koolagudi, S.G. Recognition and Classification of Pauses in Stuttered Speech Using Acoustic Features. *Proc. SPIN* **2019**. https://doi.org/10.1109/SPIN.2019.8711569
12. Barbon, S.; Guido, R.C.; Aguiar, G.J.; Santana, E.J.; Proença, M.L.; Patil, H.A. Multiple voice disorders in the same individual: Investigating handcrafted features, multi-label classification algorithms, and base-learners. *Speech Commun.* **2023**. https://doi.org/10.1016/j.specom.2023.102952
13. Carvalho, R.T.S.; Cavalcante, C.C.; Cortez, P.C. Wavelet transform and artificial neural networks applied to voice disorders identification. *Proc. NaBIC* **2011**. https://doi.org/10.1109/NABIC.2011.6089256
14. Ouhmida, A.; Raihani, A.; Cherradi, B.; Sara, S.; Khalili, T. Parkinson's Disease Identification using KNN and ANN Algorithms based on Voice Disorder. *Proc. IRASET* **2020**. https://doi.org/10.1109/IRASET48871.2020.9092228
15. Gupta, S., et al. Residual Neural Network precisely quantifies dysarthria severity-level based on short-duration speech segments. *Neural Netw.* **2021**. https://doi.org/10.1016/j.neunet.2021.02.008
16. Kourkounakis, T.; Hajavi, A.; Etemad, A. FluentNet: End-to-End Detection of Stuttered Speech Disfluencies With Deep Learning. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**. https://doi.org/10.1109/TASLP.2021.3110146
17. Jouaiti, M.; Dautenhahn, K. Dysfluency Classification in Stuttered Speech Using Deep Learning for Real-Time Applications. *Proc. ICASSP* **2022**. https://doi.org/10.1109/ICASSP43922.2022.9746638
18. Sheikh, S.A.; Sahidullah, M.; Hirsch, F.; Ouni, S. Advancing Stuttering Detection via Data Augmentation, Class-Balanced Loss and Multi-Contextual Deep Learning. *IEEE J. Biomed. Health Inform.* **2023**. https://doi.org/10.1109/JBHI.2023.3248281
19. Zhang, C.; Gong, C.; Sui, Y. Dysarthria Detection with Deep Representation Learning for Patients with Parkinson's Disease. *IEEE EMBC* **2024**. https://doi.org/10.1109/EMBC53108.2024.10781584
20. Wu, H.; Soraghan, J.J.; Lowit, A.; Di Caterina, G. Convolutional Neural Networks for Pathological Voice Detection. *IEEE EMBC* **2018**. https://doi.org/10.1109/EMBC.2018.8513222
21. Fujimura, S.; Kojima, T.; Okanoue, Y.; Shoji, K.; Inoue, M.; Omori, K.; Hori, R. Classification of Voice Disorders Using a One-Dimensional Convolutional Neural Network. *J. Voice* **2022**. https://doi.org/10.1016/j.jvoice.2020.02.009
22. Fu, D.; Zhang, X.-H.; Chen, D.; Hu, W. Pathological Voice Detection Based on Phase Reconstitution and Convolutional Neural Network. *J. Voice* **2022**. https://doi.org/10.1016/j.jvoice.2022.08.028
23. Yuan, D.Z.; Mingxuan, Y.; Yuyang, J.; Xiao-ping, P.; Xiao-jun, Z.; Tao, Z. Research on Pathological Voice Recognition using Multi-Scale Convolutional Neural Network. *Proc. ICSMD* **2023**. https://doi.org/10.1109/ICSMD60522.2023.10490471
24. Park, J.; Lee, C.G. CNN-based automatic classification of stuttering: Detection of repetitions and prolongations in stuttered speech. *Phonetics Speech Sci.* **2024**. https://doi.org/10.13064/ksss.2024.16.4.063
25. Xu, L.; Liss, J.; Berisha, V. Dysarthria detection based on a deep learning model with a clinically-interpretable layer. *JASA Express Lett.* **2023**. https://doi.org/10.1121/10.0016833
26. Varma, V.J., et al. Enhancing dysarthria severity classification: efficient audio based deep learning models. *Discov. Appl. Sci.* **2025**. https://doi.org/10.1007/s42452-025-07260-2
27. Du, Z.; Chen, H.; Ding, H.; Du, J.; Chen, Z. Hybrid Expert Knowledge and Self-Supervised Learning for Diagnostic Modeling of Adductor Spasmodic and Primary Myotonic Dysphonia. *Proc. Interspeech* **2025**. https://doi.org/10.21437/Interspeech.2025-1406
28. Disken, G. Multilabel voice disorder classification using raw waveforms. *Turk. J. Electr. Eng. Comput. Sci.* **2024**. https://doi.org/10.55730/1300-0632.4089
29. Kourkounakis, T.; Hajavi, A.; Etemad, A. Detecting Multiple Speech Disfluencies Using a Deep Residual Network with Bidirectional Long Short-Term Memory. *Proc. ICASSP* **2020**. https://doi.org/10.1109/ICASSP40776.2020.9053893
30. Al-Banna, A.-K.; Edirisinghe, E.; Fang, H. Stuttering Detection Using Atrous Convolutional Neural Networks. *Proc. ICICS* **2022**. https://doi.org/10.1109/ICICS55353.2022.9811183
31. Prabhu, Y.; Seliya, N. A CNN-Based Automated Stuttering Identification System. *Proc. ICMLA* **2022**. https://doi.org/10.1109/ICMLA55696.2022.00247
32. Filipowicz, P.; Kostek, B. Rediscovering Automatic Detection of Stuttering and Its Subclasses through Machine Learning—The Impact of Changing Deep Model Architecture and Amount of Data in the Training Set. *Appl. Sci.* **2023**. https://doi.org/10.3390/app13106192
33. Alhakbani, N.; Alnashwan, R.; Al-Nafjan, A.; Almudhi, A. Automated Stuttering Detection Using Deep Learning Techniques. *J. Clin. Med.* **2025**. https://doi.org/10.3390/jcm14103552

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 26 October 2026 https://doi.org/10.20944/preprints202610.2493.v1

34. Zhou, Z., et al. A deep learning approach for acoustic-based identification of muscle tension dysphonia and spasmodic dysphonia. *Front. Digit. Health* **2026**. https://doi.org/10.3389/fdgth.2026.1690534
35. Duan, S.; Cheng, Y.; Qin, Z.; Zhu, T.; Li, F.; Liang, Y.; Zhang, W., et al. Pathological speech classification with a dual-branch residual network considering the fluency features of speech expression. *Speech Commun.* **2026**, 103417. https://doi.org/10.1016/j.specom.2026.103417
36. Lillini, D.; Thebaud, T.; Migliorelli, L.; Dehak, N.; Squartini, S.; Moro Velazquez, L. Dysarthria Severity Classification on the HeyJay! Dataset: A Parameter-Efficient Approach Using Self-Supervised Speech Representations. *Proc. Odyssey* **2026**. https://doi.org/10.21437/Odyssey.2026-38
37. Gupta, R.; Madill, C.; Jin, C. Data-Driven Refinements for Voice Disorder Classification: Improving Accuracy and Generalisability. *Front. Digit. Health* **2026**. https://doi.org/10.3389/fdgth.2026.1800552
38. Wahab Sait, A.R.; Mohammed, H.A.J.; Bani Awad, T.A.M.; Bani Awad, A.M.A. A Novel Class-Attention Transformer-Driven Feature Fusion Technique-Based Speech Disorder Classification. *Front. Med.* **2026**. https://doi.org/10.3389/fmed.2026.1812143
39. Alnamazi, H.; Alamri, S.; Alrammah, H. Enhancing Arabic Speech Therapy with AI: Binary Fluent/Disfluent Classification of Arabic Stuttered Speech. *Electronics* **2026**. https://doi.org/10.3390/electronics15143008

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.