Archives of Acoustics Online First August 25, 2025 https://doi.org/10.24425/aoa.2025.154823

Research Paper

The Impact of Training Strategies on Overfitting in Vowel Classification Using PS-HFCC Parametrization for Automatic Speech Recognition

Stanislaw GMYREK*, Urszula LIBAL, Robert HOSSA

Faculty of Electronics, Photonics and Microsystem, Department of Acoustics, Multimedia and Signal Processing
Wroclaw University of Science and Technology
Wrocław, Poland

*Corresponding Author e-mail: stanislaw.gmyrek@pwr.edu.pl

Received February 19, 2025; accepted July 28, 2025;
published online August 25, 2025.

This paper investigates the overfitting problem in vowel classification task for automatic speech recognition (ASR). It utilizes a pitch synchronized human factor cepstral coefficients (PS-HFCC) as the parametrization method, which outperforms traditional methods like HFCC and mel-frequency cepstral coefficients (MFCC) in frame-level classification accuracy. While deep learning models are prevalent in contemporary ASR systems, they often lack explainability, a characteristic of classical classifiers. Therefore, this study examines overfitting phenomenon using a range of classifiers with well-understood properties. Specifically, it analyzes the impact of different training strategies on classifier performance, comparing the susceptibility to overfitting of several widely used classifiers, including the Gaussian mixture model (GMM), a standard approach in speech recognition. The analysis of training strategies considers various data splitting methods: random, speaker-based, and cluster-based. Our analysis of training strategies highlights the crucial role of data splitting methods: while random splitting is commonly used, it can lead to inflated accuracy due to overfitting. We demonstrate that speaker-independent splitting, where the classifier is trained on one set of speakers and tested on a separate, unseen set, is essential for robust evaluation and for accurately assessing generalization to new speakers. Potentially, the resulting insights may inform the future development and training of more reliable ASR systems.

Keywords: automatic speech recognition; vowel classification; classifier training strategy; pitch synchronized human factor cepstral coefficient; overfitting; robust parametrization; speaker grouping.

1. Introduction

The objective of speech recognition is to leverage machines, computers, and appropriate software to process speech signal and extract useful information for humans. This information can include the semantic content of speech, and the considered systems are referred to as automatic speech recognition (ASR) (Kundegorski *et al.*, 2014; Uma Maheswari *et al.*, 2020; Cherifi, Guerti, 2021), automatic voice recognition (AVR) systems for voice or speaker recognition (Maciejko, 2015), and automatic emotion recognition systems (AER) for emotional state recognition (Nedeljković *et al.*, 2020; Piątek, Kłaczyński, 2021; Stefanowska, Zieliński, 2024). Speech recognition has been a highly researched topic in recent years and continues to develop intensively. It is inherently interdisciplinary, encompassing a multitude of fields, including acoustics, digital signal processing, mathematical statistics, machine learning, artificial intelligence, linguistics, semantics, and psychology (particularly the study of emotions). In order to optimize the efficacy of an ASR system, it is essential to consider a variety of factors influencing speech, as well as the operational conditions under which it is processed, during the system design phase. It is therefore necessary to distinguish between three main categories of ASR systems: speaker dependent (SD), designed for a single speaker, speaker independent (SI) dedicated to working with multiple speakers, and speaker adaptive (SA), in which parameters can be adjusted to fit the active speaker. In order to optimize the ASR performance, it is also necessary to take into account other specific operational conditions. These include the need to work with continuous speech, single and isolated commands, and conversions to text discussions involving multiple speakers (Makowski, 2011).

Feature extraction is fundamental component of both traditional and modern ASR system architectures. Classical approaches, such as hidden Markov models with Gaussian mixture models (HMM-GMM) and hybrid HMM-deep neural network (HMM-DNN) rely on signal parameterization based on well-established methods such as spectral, cepstral, or time-frequency transformation methods. While end-to-end (E2E) architectures integrate, or partially integrate, feature extraction within DNNs, they still often utilize internal representations, such as acoustic feature vectors (in encoders) or spectrograms and mel-spectrograms (in convolutional neural networks – CNNs). Signal parameterization, therefore, remains a crucial step impacting the accuracy and efficiency of ASR systems. The ongoing search for robust parameterization methods is warranted to mitigate the negative influence of various factors related to the variability of acoustic speech features, which can detrimentally affect ASR performance. Additionally, improved signal representation can reduce the complexity requirements for recurrent neural networks (RNNs), CNNs in deep models, and E2E systems, addressing a key challenge in ASR system design: the reduction of computational complexity.

The Polish language contains six basic vowel sounds, which can be classified as either oral or nasal vowels. In this study, we focus only on the classification of six oral vowels in Polish speech: A /a/, E /ε/, I /i/, O /ɔ/, U or Ó /u/, Y /ɨ/, without the two nasal vowels: Ą /ɔ/ and Ę /ε/. The main cause of overfitting in vowel classification is the use of a small, homogeneous training dataset. When training data includes recordings from only a few speakers or environments, the model might overfit to these specific conditions and fail to generalize across different voices or settings. The other reasons for overfitting are overly complex models (e.g., DNN and E2E with many layers). In vowel classification, this can lead to overfitting, as the model may capture intricate details of the training data that do not generalize well.

2. Theoretical background of speech production

The Fant source-filter model assumes that the speech signal $s(n)$ can be described by the following relationship:

$$s(n) = u(n) \star v(n) \star r(n), \tag{1}$$

where $\star$ is the convolution operator, n is the time index, and the component $u(n)$ denotes the excitation signal, $v(n)$ is the vocal tract, and $r(n)$ describes emission of the signal through the speaker's mouth (Rabiner, Schafer, 2010). For voiced speech, the excitation signal assumes a periodic form, a noisy character in voiceless speech, or a mixed model to describe plosive phonemes (Quatieri, 2001). When the air from the lungs vibrates the vocal cords, the excitation takes the form:

$$u(n) = g(n) * p(n) = \sum_{k=0}^{+\infty} g(nT_s - kT_0), \tag{2}$$

where $g(n)$ is the shape of a single excitation pulse, $p(n)$ is a pulse train with a repetition period equal to the fundamental period T_0, which is related to the periodic opening and closing of the vocal cords, and T_s is the sampling interval.

3. Feature extraction

In general, speech is characterized by both high variability and randomness, therefore, its time signature is not an adequate representation of it. One of the key elements in the signal processing scheme of ASR systems is therefore the preprocessing and feature extraction stage. The main goal of the parameterization of the speech signal here is to represent the signal using a possibly small set of parameters that effectively extract its distinctive features relevant for further processing and analysis. The literature in this area is very extensive. In general, speech parameterization methods can be divided into two categories: solutions based on signal filtering, i.e., using linear predictive coding (LPC) analysis, and methods based on time-frequency transformations, usually short-time Fourier transform (STFT), and cepstral analysis (psychoacoustic model) of the signal. The latter are considered classic solutions.

The cepstral parameterization process results in a vector of cepstral coefficients, expressed as

$$c(t,m) = \sum_{j=1}^{J} Y_l(t,j)\cos\left(m\left(j-\frac{1}{2}\right)\frac{\pi}{J}\right); \quad m = 1,...,M, \tag{3}$$

where t is the index of the signal frame, m is the index of the cepstral coefficient, M is the number of coefficients, j is the index of the mel scale bin, J is the number of mel bands, and Y_l is the logarithm of the amplitude spectrum in the mel-frequency scale obtained at the output of a bank of perceptual filters.

Various cepstral parameterization solutions differ mainly in the way the perceptual filter bank coefficients are determined. A commonly used feature extraction method in speech recognition are mel-frequency cepstral coefficients (MFCC), introduced by Davis and Mermelstein (1980). MFCCs are popular preprocessing method, not only in speech recognition (Upadhyaya *et al.*, 2015), but also in multiple

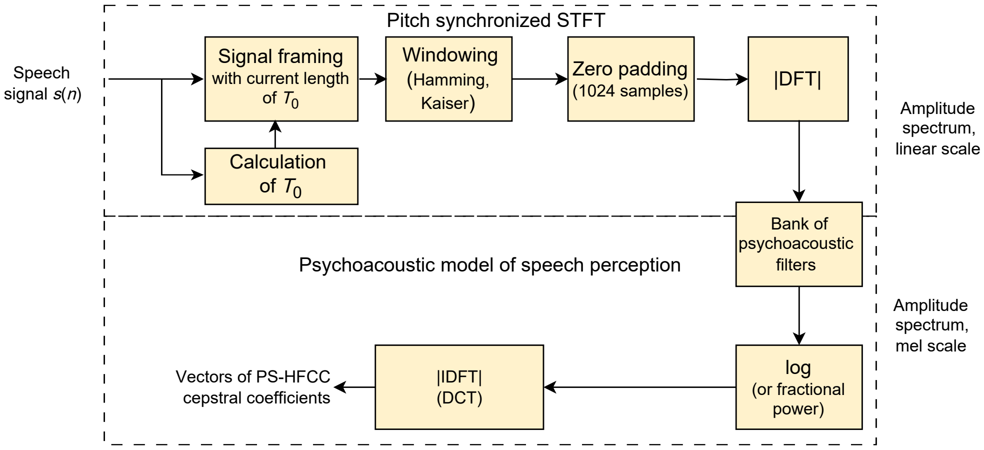

Pitch synchronized STFT
Speech signal $s(n)$
Signal framing with current length of T_0
Windowing (Hamming, Kaiser)
Zero padding (1024 samples)
|DFT|
Calculation of T_0
Amplitude spectrum, linear scale
Bank of psychoacoustic filters
Psychoacoustic model of speech perception
Amplitude spectrum, mel scale
log (or fractional power)
|IDFT| (DCT)
Vectors of PS-HFCC cepstral coefficients

Fig. 1. PS-HFCC cepstral parameterization scheme.

other applications, such as honeybee sound analysis (Libal, Biernacki, 2024a; 2024b; 2024c). However, a more robust cepstral representation is obtained by human factor cepstral coefficient (HFCC) parameterization (Skowronski, Harris, 2004). Studies show that HFCCs perform better under noisy signal conditions and lead to improved classification results, i.e., lower errors during the single-frame recognition stage of the signal (Skowronski, Harris, 2004).

In order to make the classical HFCC parametrization method robust against the negative effects of excitation periodicity in voiced speech phonemes, the pitch synchronized human factor cepstral coefficient (PS-HFCC) parametrization is employed. This approach utilizes variable-length signal frame processing, as depricted in Fig. 1.

The result of the frequency analysis of a signal frame $s_w(n)$ containing a voiced fragment of speech can be described by the following relation (Gmyrek *et al.*, 2023):

$$S_w(\omega) = (S(\omega) \cdot G(\omega, T_0)) \star W(\omega), \tag{4}$$

where $S_w(\omega)$ is the frame spectrum of the speech signal, $S(\omega)$ is the desired form of the spectrum with clearly visible formants, and $W(\omega)$ is responsible for the impact of the windowing operation needed to extract individual frames from the recorded signal. The negative impact of the fundamental frequency f_0 on HFCC coefficients was studied in detail in (Gmyrek *et al.*, 2023; 2024).

The PS-HFCC method makes it possible to compensate for the undesired effect of $G(\omega, T_0)$, which results in occurrence of amplitude spectrum ripples at multiples of the fundamental frequency f_0. In this case, the modification of the method consists of estimating the current value of the fundamental frequency f_0 and synchronizing the signal frame length with the fundamental period T_0. Details of this solution are described in (Gmyrek, Hossa, 2025a; 2025b). By applying the PS-HFCC method, the values of the variance estimators of the cepstral coefficients decrease and, consequently, a higher concentration of areas with data representing the cepstral parameters of the elementary frames is observed. At the same time, this results in narrower multivariate probability density distributions of the data, which in turn translates into better classification results, i.e., a decrease in recognition errors at the level of individual signal frames levels.

The preprocessing of the Polish vowel recordings using PS-HFCC parametrization was performed using custom-prepared MATLAB scripts.

4. Database

The authors developed a proprietary speech corpus, comprising recordings from 37 adult male speakers, collected from various regions of Poland. For each speaker, 150 Polish words were recorded, with speech fragments containing vowels (six classes) from 43 words subsequently employed in the experiment. The sampling rate of the signals was 12 kHz. The original database was characterized by a low noise level with a signal-to-noise ratio (SNR) of 35 dB. The experiments, presented in this work, were conducted on both the original dataset and its noisy versions, with SNRs of 20 dB, 10 dB, and 5 dB (representing progressively higher noise level). This database is highly representative – it captures both inter-speaker and intra-speaker variability, as well as contextual and phonetic diversity. The preparation of this dataset was exceptionally labor-intensive and costly, involving semi-automatic signal segmentation and detailed phonetic labeling. All recordings were manually segmented and labeled, with six phonemes (‘a’, ‘e’, ‘i’, ‘o’, ‘u’, and ‘y’) chosen as the phonetic units for labeling process. The frame length was set to 30 ms, with a 10 ms shift. The YIN estimator (de Cheveigné, Kawahara, 2002) and its statistically improved version PYIN (Mauch, Dixon, 2014) were employed to estimate the current value of the fundamental period T_0. After signal preprocessing, we obtained $N = 14$ cepstral coefficients for each frame. In our database, the numbers of frames for individual phonemes were: 12 208 for 'i', 9288 for 'y', 29 778 for 'e', 35 406 for 'a', 23 628 for 'o', and 8082 for 'u'. The total number of frames that contain vowel sounds was 118 390.

The authors acknowledge the limitations associated with analyzing only male speech recordings, so a comparable female speech corpus is currently under development. However, it is not expected to play a central role in the current analysis. The primary acoustic distinction between male and female voices lies in the higher fundamental frequency (f_0) typically observed in female speakers. In pitch-synchronous analysis, this leads to a greater number of pitch cycles within a single frame, which may facilitate improved signal averaging and potentially enhance recognition accuracy. Nonetheless, this factor has a limited impact on overfitting, which is primarily influenced by the chosen training and test data partitioning strategy.

5. Training and test sets

All classification methods based on the maximum likelihood lead to fit models to the training dataset. The real challenge is to construct the training and testing sets in a way that prevents overfitting. To trace the influence of the training process of the classifier, we conducted a series of tests using a database of vowel sounds from Polish speech. These tests involved dividing the data into training and test sets in various ways, with an effort to maintain the following proportions: two-thirds of data for the training set, and the remaining one-third for the test set. The training and testing sets were always kept disjoint.

For the purposes of this research, we employed three methods of splitting the data into training and testing sets: random split, speaker split, and cluster split, each of which is described further.

5.1. Method 1: random split

The first method for splitting the data into training and test sets is a random split, with exact proportions of two-thirds for the training set and one-third for the test set. Each recording frame was randomly assigned to one of these sets. The training and test sets remained disjoint. The random split was performed 10 times, and all classifiers were trained 10 times on the obtained sets to average the results.

5.2. Method 2: speaker split

The second method relies on dividing the speakers into two groups: one for training the classifiers and the second for testing them. The database contains recordings from 37 speakers. While the division into training and test sets was not exactly two-thirds to one-third, but quite close with 24 speakers in the training set and 13 speakers in the test set. The random choice of speakers for the two sets was also repeated 10 times, leading to 10 separate experiments to average the results. In this split, the training set was disjoint from the test set, and additionally, no recordings from the same speaker appeared in both sets at the same time.

In contrast to random split, the speaker split method for the training of the classifiers should prevent overfitting to some degree and provide more realistic (lower) performance results.

5.3. Method 3: cluster split

The third method involves K-means clustering. Three clusters were separated from the data by the K-means algorithm with $K = 3$. Each recording frame was allocated to one of the three clusters. The clusters produced by the K-means algorithm are groups of data points that share similar features and are spatially close to one another in the feature space. In K-means clustering, each cluster is defined by a centroid, which represents the center of the cluster and is calculated as the average of all data points within that cluster. The algorithm iteratively adjusts the positions of the centroids to minimize the sum of squared distances between the data points and their respective centroids, ensuring data points are grouped in a way that reduces intra-cluster variance.

To maintain the two-thirds to one-third ratio, in each experiment one cluster served as the test set, while the remaining two clusters were used as the training set. The clusters are formed by grouping similar feature vectors based on Euclidean distance. We expect that this data-splitting method will produce the worst performance for the classifiers.

6. Vowel classification

6.1. Classifiers

For the classification of vowels and the study of overfitting, we performed a series of classifications using the following classifiers: Gaussian mixture model (GMM), K-nearest neighbors (KNN), random forest (RF), support vector machines (SVM), and multi-layer perceptron (MLP). The classification experiments were performed in the Python programming language with the scikit-learn library.

In the context of speech recognition, the GMM (Reynolds, 2009; McLachlan, Peel, 2000) is a popular statistical approach used to model the distribution of acoustic features in speech. A GMM is a mixture of M multivariate, normal distributions, which together describe the distribution of input data, such as acoustic feature vectors (e.g., MFCC, HFCC or PS-HFCC) extracted from speech signals. GMM is particularly useful in acoustic classification, as it allows for modeling the variability in speech across different speakers, acoustic conditions, and over time. The diagonal covariance matrices Σ_{ic} were determined based on the expectation-maximization (EM) algorithm:

$$p_c(o) = \sum_{i=1}^{M} w_{ic} \mathcal{N}(o, m_{ic}, \Sigma_{ic}), \tag{5}$$

where w_{ic} denotes the weights and m_{ic} denotes the means for the mixture of i-th component and c-th phoneme (class). The EM algorithm was described in detail in (Dempster *et al.*, 1977).

The KNN (Cover, Hart, 1967; Bishop, 2006) is a simple, instance-based algorithm that classifies data points based on the majority class of their nearest neighbors. It uses distance metrics (such like Euclidean distance) to find the closest K-neighbors and assigns the most common class among them. To investigate the overfitting, tests were conducted for $K = 1, 21$, and 51 neighbors. KNN can overfit when K is too small (e.g., $K = 1$), as it becomes sensitive to noise and outliers in the training data. This leads to a model that performs well on the training data but poorly on unseen data.

MLP, also called deep feedforward network (Rumelhart *et al.*, 1986; Goodfellow *et al.*, 2016), is a type of neural network with multiple layers of neurons. It uses backpropagation to learn the weights, making it capable to learn complex patterns and nonlinear decision boundaries. MLPs are widely used for various classification tasks. MLP can overfit when the network is too deep (i.e., there are too many layers or neurons) or when training is conducted for too many epochs. Overfitting occurs when the network becomes too specialized to the training data, capturing noise and irrelevant patterns, leading to poor performance on new data. This is particularly likely if regularization techniques such as dropout or L2 regularization are not used. For our experiments, we utilized an MLP with one hidden layer consisting of 100 neurons and reclective linear unit (ReLu) activation function. The MLP was trained for 200 epochs.

RF (Breiman, 2001) is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy. Each tree is trained on a random subset of the data, and the final prediction is based on a majority vote from all trees. This approach reduces variance and improves accuracy compared to a single decision tree. Although RFs are less prone to overfitting compared to individual decision trees, they can still overfit if the trees are grown too deep or if the number of trees is too large. Overfitting can occur if the model becomes overly complex and captures noise in the data.

SVM (Bishop, 2006; Cortes, Vapnik, 1995) is a powerful classification algorithm that finds the hyperplane that best separates the data into different classes. It maximizes the margin between the closest data points (support vectors) from different classes. For non-linear data, SVM can use kernel functions (e.g., radial basis functions (RBFs) or polynomial) to map the data into higher-dimensional spaces where it can be linearly separated. In this study, we used RBFs as kernels. SVM can overfit when a very complex kernel (e.g., a high-degree polynomial) is used, or when the regularization parameter is set too high, causing the model to fit the training data too closely, including noise, at the expense of generalization.

6.2. Classification error analysis

The classification results were analyzed by using three error measures: accuracy, frame error rate (FER), and the confusion matrix.

Accuracy is the most general method for comparing different classification methods. It is defined as the fraction of correct predictions N_{correct} out of all predictions N:

$$\text{Acc} = \frac{N_{\text{correct}}}{N} \cdot 100\,\%. \tag{6}$$

The higher the accuracy, the better the classification quality. On the other hand, the accuracy measure does not distinguish between the accuracy across individual classes, which can sometimes be crucial when analyzing classifiers. Other measures, such as the FER and the confusion matrix, are employed to address this limitation.

A confusion matrix is a performance evaluation tool commonly used in machine learning to evaluate the accuracy of classification models. It provides a summary of prediction results for a classification problem by comparing predicted labels with the actual labels for each class. Due to varying number of frames for each class, we present confusion matrices containing percentage results instead of numbers.

FER measure is traditionally used to assess the quality of speech recognition at the individual frame level and is defined for a class c as

$$\text{FER}(c) = \frac{N_{\text{err}}(c)}{N(c)} \cdot 100\,\%, \tag{7}$$

where $N(c)$ is the total number of frames undergoing recognition and $N_{\text{err}}(c)$ is the number of unrecognized frames from class c. FER can also be calculated directly from the confusion matrix by taking the ratio of the sum of all values in a row of the confusion matrix, excluding the diagonal value, to the sum of all values in that row.

6.3. Classifier overfitting

The aim of this paper is to investigate the overfitting of vowel classification for Polish speech. Classifier overfitting (Hastie *et al.*, 2001; Kuhn, Johnson, 2013; Ng, 2004) occurs when a model learns to perform exceedingly well on the training data to the extent that it begins to memorize irrelevant details, noise, or peculiarities that are specific to that dataset. As a result, while the model achieves high accuracy on the training set, its performance deteriorates on unseen test data. In essence, the model fails to generalize well, being overly tailored to the particular examples it has encountered, rather than capturing the underlying patterns that could apply to new, unseen data.

Table 1. Accuracy [%] for random split (method 1).

Classifier	35 dB	20 dB	10 dB	5 dB
GMM	85.92	85.89	85.56	84.20
1NN	99.52	99.05	98.89	98.02
21NN	94.36	92.70	92.19	90.75
51NN	91.82	90.64	90.32	89.12
MLP	92.76	92.00	90.52	89.16
RF	92.59	90.94	90.10	88.71
SVM	94.03	92.93	91.90	90.53

Table 2. Accuracy [%] for speaker split (method 2).

Classifier	35 dB	20 dB	10 dB	5 dB
GMM	81.11	80.73	81.42	80.04
1NN	77.52	76.97	76.39	75.15
21NN	81.19	81.54	81.25	80.62
51NN	82.47	82.65	82.38	81.58
MLP	79.84	79.54	79.63	79.58
RF	82.84	82.97	82.84	82.12
SVM	82.72	83.24	83.02	82.44

Table 3. Accuracy [%] for cluster split (method 3).

Classifier	35 dB	20 dB	10 dB	5 dB
GMM	64.05	71.28	65.27	64.73
1NN	60.22	58.20	57.62	54.90
21NN	57.81	55.98	53.66	52.52
51NN	54.75	54.51	50.35	50.37
MLP	77.01	76.76	74.41	76.55
RF	56.50	56.41	56.04	57.53
SVM	81.44	78.05	76.77	73.82

7. Results

7.1. Comparison of accuracy with split methods

In this section, we analyze the performances of various classifiers using different train-test split methods: random, speaker and cluster. In this experiment, we used seven classifiers with different overfitting tendencies: GMM, KNN with $K = 1$, 21, and 51, a fully connected neural network of 100 perceptrons (MLP), RF with trees of 15 branches in depth, and SVM with RBF kernels. Results for our custom dataset with a signal-to-noise ratio (SNR) of 35 dB (indicating a low level of noise), as well as artificially noised versions of the dataset (with SNRs of 20 dB, 10 dB, and 5 dB, representing very high levels of noise) are presented in Fig. 2 and Tables 1, 2, and 3. The most important factor in overfitting analysis is the comparison of the first two methods, i.e., random split versus speaker split. The greater the accuracy between these two splits (assuming the random split yields higher accuracy compared to the speaker split), the stronger the overfitting effect. At higher noise levels (i.e., lower SNRs), all classifiers exhibit a small drop in accuracy, at most by just a few percentage points. Nonetheless, the PS-HFCC parametrization generally demonstrates robustness against significant noise in the recordings. As depicted in Fig. 2, classification results across various SNRs (original 35 dB, 20 dB, 10 dB, and 5 dB) remain consistent regardless of the train-test split method. This consistency underscores the noise resilience of PS-HFCC parametrization.

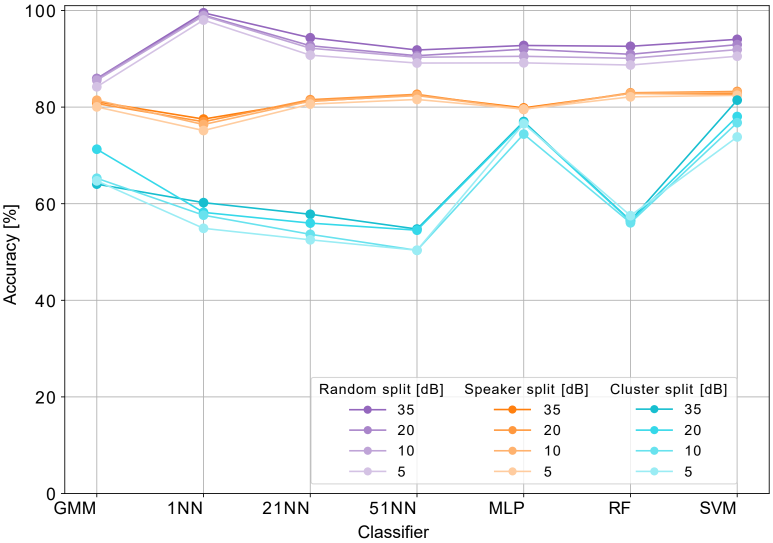


Fig. 2. Classification accuracy for different train-test splits and SNRs.

As expected, the KNN classifier with one neighbor ($K = 1$) exhibited overfitting, serving as a 'litmus test' for this behavior. The difference in accuracy between the random and speaker splits for 1NN was very high, at 22 %. This indicates that the 1NN algorithm fits the training data. Considering that the signal was divided into 30 ms frames with 10 ms shifts, it is highly probable that a neighboring frame, with 20 ms overlap, could be chosen as the nearest neighbor. The difference in accuracy between the random and speaker splits for 21NN dropped to 13.17 %, indicating better generalization of results due to the majority voting among the 21 neighbors. For 51NN, the difference further decreased to 9.35 %. Obviously, the more neighbors vote for the predicted class label, the better the generalization. At the same time, there are disproportions in the number of frames for individual classes (since speech naturally contains more instances of some vowels, e.g., 'a' and 'e'), which influences the final classification result.

Very similar results to 51NN were obtained for the RF classification, for which the difference in accuracy between the random and speaker splits was equal to 9.75 %. The RF model used in this experiment was an ensemble of 100 trees. These trees were allowed to grow to the maximum depth of 15. The RFs constructed by shallower trees led to relatively lower classification accuracy, especially for the speaker split. Using deeper trees can lead to much stronger overfitting effect, which is observed here, but at the same level to that of relatively well generalized 51NN classifier.

To a certain extent, comparable behavior was noticed for the MLP and SVM methods. The difference in accuracy between the random and speaker splits was 12.94 % and 11.31 %, respectively, representing average results of overfitting compared to the other classifiers. Interestingly, the performance of the two classifiers on the speaker and cluster splits was very close, with only negligible small drop in accuracy from 79.84 % to 77.01 % for MLP, and from 82.72 % to 81.44 % for SVM – see Tables 2 and 3. This indicates that both methods handle clustered data (from K-means partitioning) significantly well, due to their nonlinear mappings. However, despite this strength, the overfitting effect remains evident in the case of the random split, for both MLP and SVM.

Among the many parameters by which speakers can be divided into groups, their personal vocal characteristics, including vocal tract parameters, are especially noteworthy. One such parameter is the fundamental frequency f_0, which can be taken as an indicator of vocal tract size, as it is closely related to the length of the speaker's vocal cords (Makowski, 2011). The lengths of the oral and pharyngeal parts of the vocal tract can be taken as a basis for grouping, as they directly affect the positioning of formants on the frequency axis (Naito *et al.*, 2002), as well as influence the parametrization coefficients. These coefficients aim to maximize the distance between the multidimensional probability distributions of the feature vectors in terms of the chosen distance measure. Partitioning can also be performed hierarchically, using multiple factor to distinguish speakers from one another. For example, clustering based on gender and speaking speed has been proposed in (Hazen, 2000). One of the more recent algorithms proposed in the literature is an approach based on adapting the weights of universal background model (UBM) proposed in (Hossa, Makowski, 2016). However, in the current study, clustering was performed numerically using the K-means method. As expected, the classification quality in this case was the lowest among all classifiers used – see Table 3.

7.2. FER analysis

Analysis of the FER for the original custom dataset with an SNR of 35 dB, as presented in the Fig. 3, leads

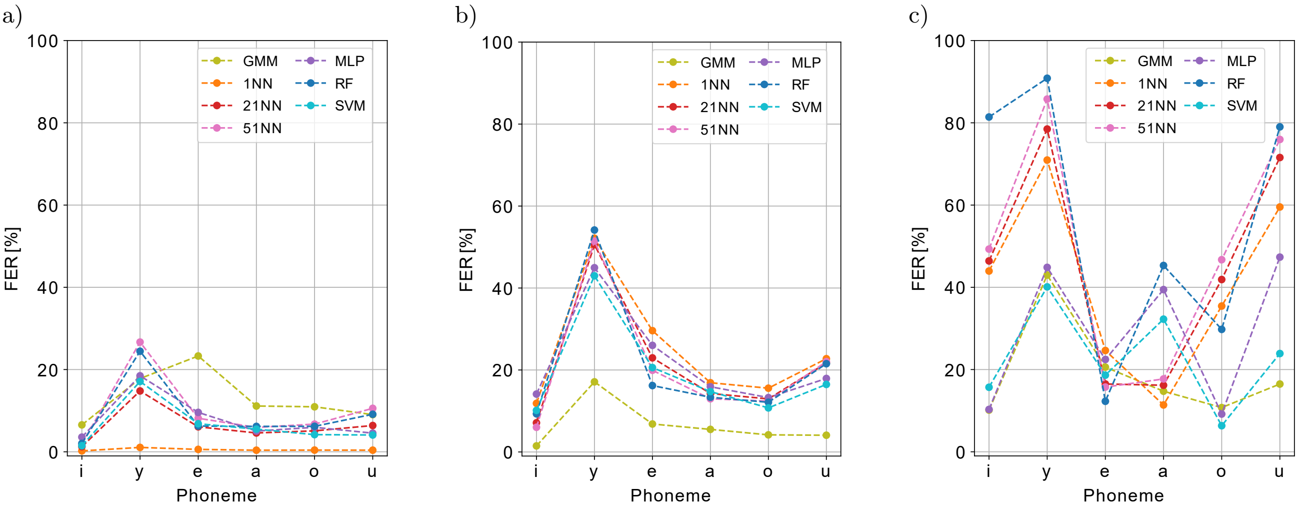
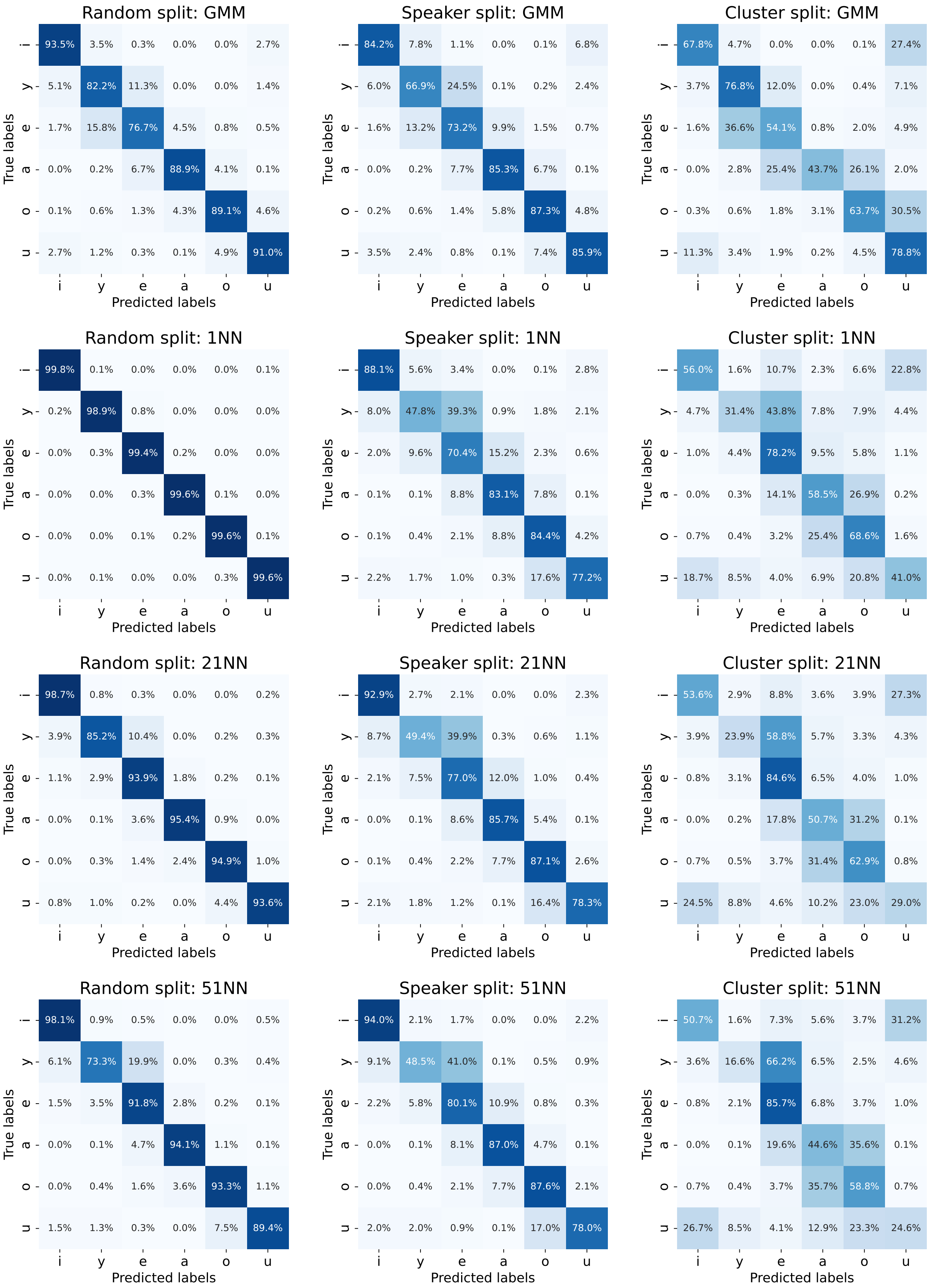


Fig. 3. FER for SNR = 35 dB across three data split approaches used during classifiers training: a) random split; b) speaker split; c) cluster split.



Random split: GMM

	i	y	e	a	o	u
i	93.5%	3.5%	0.3%	0.0%	0.0%	2.7%
y	5.1%	82.2%	11.3%	0.0%	0.0%	1.4%
e	1.7%	15.8%	76.7%	4.5%	0.8%	0.5%
a	0.0%	0.2%	6.7%	88.9%	4.1%	0.1%
o	0.1%	0.6%	1.3%	4.3%	89.1%	4.6%
u	2.7%	1.2%	0.3%	0.1%	4.9%	91.0%

Speaker split: GMM

	i	y	e	a	o	u
i	84.2%	7.8%	1.1%	0.0%	0.1%	6.8%
y	6.0%	66.9%	24.5%	0.1%	0.2%	2.4%
e	1.6%	13.2%	73.2%	9.9%	1.5%	0.7%
a	0.0%	0.2%	7.7%	85.3%	6.7%	0.1%
o	0.2%	0.6%	1.4%	5.8%	87.3%	4.8%
u	3.5%	2.4%	0.8%	0.1%	7.4%	85.9%

Cluster split: GMM

	i	y	e	a	o	u
i	67.8%	4.7%	0.0%	0.0%	0.1%	27.4%
y	3.7%	76.8%	12.0%	0.0%	0.4%	7.1%
e	1.6%	36.6%	54.1%	0.8%	2.0%	4.9%
a	0.0%	2.8%	25.4%	43.7%	26.1%	2.0%
o	0.3%	0.6%	1.8%	3.1%	63.7%	30.5%
u	11.3%	3.4%	1.9%	0.2%	4.5%	78.8%

Random split: 1NN

	i	y	e	a	o	u
i	99.8%	0.1%	0.0%	0.0%	0.0%	0.1%
y	0.2%	98.9%	0.8%	0.0%	0.0%	0.0%
e	0.0%	0.3%	99.4%	0.2%	0.0%	0.0%
a	0.0%	0.0%	0.3%	99.6%	0.1%	0.0%
o	0.0%	0.0%	0.1%	0.2%	99.6%	0.1%
u	0.0%	0.1%	0.0%	0.0%	0.3%	99.6%

Speaker split: 1NN

	i	y	e	a	o	u
i	88.1%	5.6%	3.4%	0.0%	0.1%	2.8%
y	8.0%	47.8%	39.3%	0.9%	1.8%	2.1%
e	2.0%	9.6%	70.4%	15.2%	2.3%	0.6%
a	0.1%	0.1%	8.8%	83.1%	7.8%	0.1%
o	0.1%	0.4%	2.1%	8.8%	84.4%	4.2%
u	2.2%	1.7%	1.0%	0.3%	17.6%	77.2%

Cluster split: 1NN

	i	y	e	a	o	u
i	56.0%	1.6%	10.7%	2.3%	6.6%	22.8%
y	4.7%	31.4%	43.8%	7.8%	7.9%	4.4%
e	1.0%	4.4%	78.2%	9.5%	5.8%	1.1%
a	0.0%	0.3%	14.1%	58.5%	26.9%	0.2%
o	0.7%	0.4%	3.2%	25.4%	68.6%	1.6%
u	18.7%	8.5%	4.0%	6.9%	20.8%	41.0%

Random split: 21NN

	i	y	e	a	o	u
i	98.7%	0.8%	0.3%	0.0%	0.0%	0.2%
y	3.9%	85.2%	10.4%	0.0%	0.2%	0.3%
e	1.1%	2.9%	93.9%	1.8%	0.2%	0.1%
a	0.0%	0.1%	3.6%	95.4%	0.9%	0.0%
o	0.0%	0.3%	1.4%	2.4%	94.9%	1.0%
u	0.8%	1.0%	0.2%	0.0%	4.4%	93.6%

Speaker split: 21NN

	i	y	e	a	o	u
i	92.9%	2.7%	2.1%	0.0%	0.0%	2.3%
y	8.7%	49.4%	39.9%	0.3%	0.6%	1.1%
e	2.1%	7.5%	77.0%	12.0%	1.0%	0.4%
a	0.0%	0.1%	8.6%	85.7%	5.4%	0.1%
o	0.1%	0.4%	2.2%	7.7%	87.1%	2.6%
u	2.1%	1.8%	1.2%	0.1%	16.4%	78.3%

Cluster split: 21NN

	i	y	e	a	o	u
i	53.6%	2.9%	8.8%	3.6%	3.9%	27.3%
y	3.9%	23.9%	58.8%	5.7%	3.3%	4.3%
e	0.8%	3.1%	84.6%	6.5%	4.0%	1.0%
a	0.0%	0.2%	17.8%	50.7%	31.2%	0.1%
o	0.7%	0.5%	3.7%	31.4%	62.9%	0.8%
u	24.5%	8.8%	4.6%	10.2%	23.0%	29.0%

Random split: 51NN

	i	y	e	a	o	u
i	98.1%	0.9%	0.5%	0.0%	0.0%	0.5%
y	6.1%	73.3%	19.9%	0.0%	0.3%	0.4%
e	1.5%	3.5%	91.8%	2.8%	0.2%	0.1%
a	0.0%	0.1%	4.7%	94.1%	1.1%	0.1%
o	0.0%	0.4%	1.6%	3.6%	93.3%	1.1%
u	1.5%	1.3%	0.3%	0.0%	7.5%	89.4%

Speaker split: 51NN

	i	y	e	a	o	u
i	94.0%	2.1%	1.7%	0.0%	0.0%	2.2%
y	9.1%	48.5%	41.0%	0.1%	0.5%	0.9%
e	2.2%	5.8%	80.1%	10.9%	0.8%	0.3%
a	0.0%	0.1%	8.1%	87.0%	4.7%	0.1%
o	0.0%	0.4%	2.1%	7.7%	87.6%	2.1%
u	2.0%	2.0%	0.9%	0.1%	17.0%	78.0%

Cluster split: 51NN

	i	y	e	a	o	u
i	50.7%	1.6%	7.3%	5.6%	3.7%	31.2%
y	3.6%	16.6%	66.2%	6.5%	2.5%	4.6%
e	0.8%	2.1%	85.7%	6.8%	3.7%	1.0%
a	0.0%	0.1%	19.6%	44.6%	35.6%	0.1%
o	0.7%	0.4%	3.7%	35.7%	58.8%	0.7%
u	26.7%	8.5%	4.1%	12.9%	23.3%	24.6%

Fig. 4. Confusion matrices for the classifiers: GMM and KNN with $K = 1$, 21, and 51. Classification was performed using three data split methods between training and testing sets. The results were obtained on the original custom Polish vowels dataset with an SNR of 35 dB.

to the following general conclusions. For the random split of data between training and testing sets (Fig. 3a), the 1NN classifier performed worse than all others, demonstrating a strong overfitting. For the speaker split (Fig. 3b), the only well and consistently performing classifier was GMM. Under the cluster split condition (Fig. 3c), the comparable results were obtained for GMM, MLP, and SVM methods, with a slight advantage of GMM.

The most problematic phoneme in terms of recognition accuracy was the vowel ‘y’, which exhibited the highest error values for all three splitting methods. The class ‘y’ was most frequently misclassified as ‘i’. However, for the GMM with a random split, the highest FER was observed for the vowel ‘e’, which was again confused with ‘y’ sounds.

7.3. Local error analysis

We analyzed the confusion matrices (Figs. 4 and 5) for all tested classifiers to determine local errors, i.e., the confusion between true and predicted vowels. The most problematic vowel pair is ‘y’ and ‘e’. This is due to the close proximity of their formant frequencies

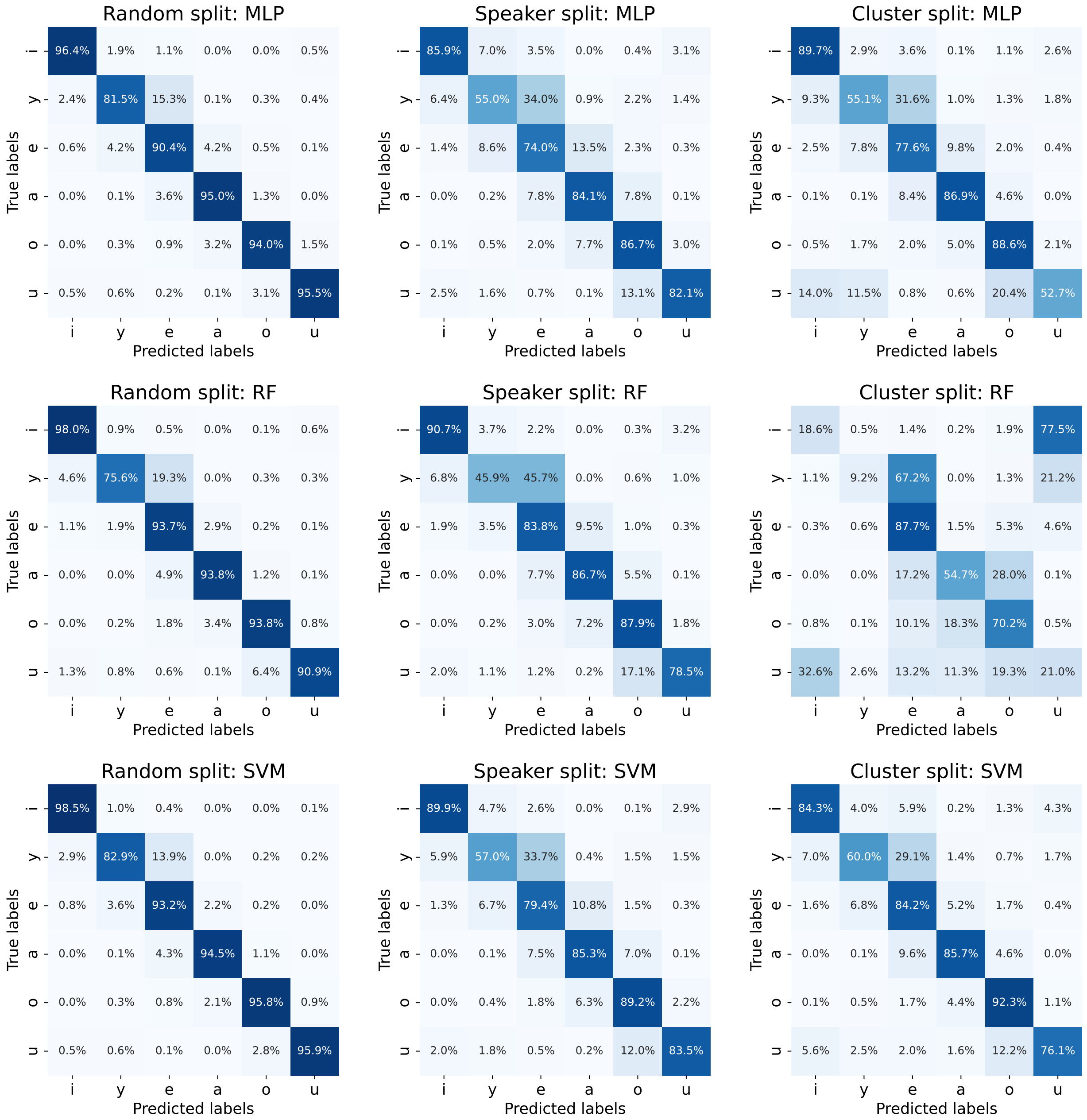


Fig. 5. Confusion matrices for the classifiers: MLP, RF, and SVM. Classification was performed using three methods of data splitting between training and testing sets. The results were obtained on the original custom Polish vowels dataset with an SNR of 35 dB.

(Makowski, 2011). This can be verified by analyzing widely available formant frequency tables for Polish vowels, for example, such as those in (Jassem, 1973). Consequently, these vowels are often confused.

The confusion between vowels ‘y’ and ‘e’ is the strongest in the speaker split for both training and testing sets. All classifiers for the speaker split predict frames from the vowel class ‘y’ as class ‘e’ quite often, with the frequency (estimated probabilities) ranging from 24.5 % for GMM to as high as 45.7 % for RF – see Figs. 4 and 5. The worst performance was obtained with RF for all three methods of data split.

The confusion between ‘o’ and ‘u’ in the speaker split is highest for the KNN and RF algorithms (around 17 %). The best result, with the lowest number of misclassified phonemes ‘o’ and ‘u’ in the speaker split, was achieved again by GMM (7.4 %). However, for the cluster split, GMM performed worst with 30.5 % of ‘o’ phonemes misclassified as ‘u’. This is due to the characteristic of the GMM training process, where a separate model is built for each class and some phonemes are not properly represented in the training clusters after K-means clustering of the data.

The confusion between ‘i’ and ‘u’ and ‘a’ and ‘o’ appears most prominently in the cluster split. For the first phoneme pair (‘i’ and ‘u’), the errors rate rage from 14.0 % for MLP to 77.5 % for RF. Moreover, under the speaker split all classifiers except GMM, achieved error rates of around 3 % or lower, while only GMM exceeded twice that level. Phoneme ‘u’ naturally occurs quite rarely in Polish speech compared to other phonemes. Confusion between ‘a’ and ‘o’ is mainly visible for the cluster split with quite high errors from around 25 % to 35 %, except MLP and SVM classifiers, which performed very well in that case. We guess well trained nonlinearity of decision function is responsible for those correct classifications.

The confusion between ‘e’ and ‘a’ is observed for all classifiers under the seakers split and error rate ranges from 9.5 % for RF to 15.2 % for 1NN. For the random split, these two phonemes were generally confused in 4 %–5 % of cases, with two exceptions for the GMM and 1NN algorithms. For GMM, we observe error rates of 4.5 % and 6.7 %, while for the 1NN – extremely low error rates of 0.2 % and 0.3 %. These low error rates for 1NN indicate, a strong overfitting tendency (only one neighbor decides on predicted class). In contrast, GMM demonstrates to well-generalizable performance.

8. Conclusions

Overfitting in classifiers remains a challenging phenomenon to quantify in a rigorous scientific manner, especially in real-world applications. However, it can have a detrimental effect, causing models to make inaccurate predictions, even when the tested data suggest otherwise.

The experiments conducted in this study involved a comparison of the learning performance of the following seven classifiers: GMM, 1NN, 21NN, 51NN, MLP, RF, and SVM. These classifiers were trained on three data setups, each applyinh different training strategies for splitting the data into training and testing sets:

- random split using all frames;
- speaker split, where speakers were grouped to avoid repetition;
- cluster split, based on the most distant feature vectors, selected according to a chosen metric.

Using an ASR system, outside the conditions it was trained on, can lead to a significant drop in performance. Speech signals are highly variable, influenced by factors such as speaker characteristics (e.g., gender, age, vocal tract anatomy), intra-speaker variability, linguistic diversity, as wells as regional, cultural, and contextual factors. Therefore, ASR systems must account for these variations to maintain accuracy across different environments and user populations.

The PS-STFT, a generalization of classical cepstral parameterization methods such as MFCC and HFCC, as well as other spectrogram-based approaches, enhances recognition performance in ASR systems. It achieves this by smoothing the amplitude spectra (and, consequently, spectrograms), and by reducing the variance of cepstral coefficient estimators. Our classification results across various SNRs (original 35 dB, 20 dB, 10 dB, and 5 dB) consistently demonstrate the high robustness of the PS-HFCC parametrization to noise in recordings, regardless of the train-test split method used.

The aim of the study was to compare different popular classifiers with – the default algorithm commonly used in speech recognition – the GMM algorithm. GMM is the most robust against overfitting among tested classifiers and well generalizes the data, even in the case of a random split between training and testing sets in the classifier learning process. This is a scenario in which we expected the highest overfitting effect. Considering that the signals were divided into 30 ms frames with 10 ms shifts, it is highly probable that neighboring frames with 20 ms overlap were chosen as nearest neighbors. For this reason, the KNN algorithm served as a reference point for overfit detection.

While random split is the default method used in most studies on classification tasks, speech recognition is a specialized task that often requires alternative splitting strategies, such as speaker grouping by gender or other individual characteristics. In scenarios like ours, where the classifier is trained on one set of speakers and tested on a different set (i.e., a speaker split), this approach is essential. Training an algorithm for all potential speakers is impossible due to the vast voice diversity, as encountered in real-world applications such as smartphone-based speech recognition.

For this reason, popular big-data systems incorporated in everyday software utilize users voice samples to retrain the systems and improve their performance. However, collecting additional samples from new speakers is not always feasible. Therefore, it is important to use methods that are as resistant to overfitting as possible.

Beyond simply separating speakers for training and testing, advanced speaker grouping techniques (e.g., based on gender, dialect, or vocal characteristics) can further enhance the robustness of ASR systems. While our current study utilizes only male recordings, the observed benefits of speaker-independent train-test splitting and the robust performance of PS-HFCC parametrization are expected to extend to other demographic groups. This underscores at broader implication: speaker-independent evaluation, potentially incorporating detailed speaker grouping, is crucial for developing ASR models that truly generalize and avoid overfitting to specific speaker characteristics present in the training data, thereby ensuring reliable performance with unseen users across diverse populations.

Funding

The work was not supported by any external organizations or institutions. It is the result of scientific research conducted within the Department of Acoustics, Multimedia, and Signal Processing at Wrocław University of Science and Technology, Poland.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors' contribution

Stanisław Gmyrek was responsible for developing the MATLAB software, conceptualizing and implementing the signal processing algorithms described in the paper, and editing the corresponding sections of the manuscript. Urszula Libal was responsible for implementing the Python-based software, data classification, visualization, and results interpretation. Robert Hossa contributed to the conceptualization of the signal processing algorithms and to the overall writing and editing of the article. All authors reviewed and approved the final manuscript

Acknowledgments

The authors are grateful to Prof. Ryszard Makowski from the Wroclaw University of Science and Technology, Poland, for providing the Polish Speech Database and for many insightful discussions.

References

1. Bishop C.M. (2006), *Pattern Recognition and Machine Learning*, Springer, New York.
2. Breiman L. (2001), Random forests, *Machine Learning*, **45**(1): 5–32, https://doi.org/10.1023/A:1010933404324.
3. Cherifi E., Guerti M. (2021), Conditional random fields applied to Arabic orthographic-phonetic transcription, *Archives of Acoustics*, **46**(2): 237–247, https://doi.org/10.24425/aoa.2021.136574.
4. Cortes C., Vapnik V. (1995), Support-vector networks, *Machine Learning*, **20**(3): 273–297, https://doi.org/10.1007/BF00994018.
5. Cover T., Hart P. (1967), Nearest neighbor pattern classification, *IEEE Transactions on Information Theory*, **13**(1): 21–27, https://doi.org/10.1109/TIT.1967.1053964.
6. Davis S., Mermelstein P. (1980), Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **28**(4): 357–366, https://doi.org/10.1109/TASSP.1980.1163420.
7. de Cheveigné A., Kawahara H. (2002), YIN, a fundamental frequency estimator for speech and music, *The Journal of the Acoustical Society of America*, **111**(4): 1917–1930, https://doi.org/10.1121/1.1458024.
8. Dempster A.P., Laird N.M., Rubin D.B. (1977), Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1): 1–22, https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.
9. Gmyrek S., Hossa R. (2025a), Improving the vowel classification accuracy using varying signal frame length, *Vibrations in Physical Systems*, **36**(1): 2025114, https://doi.org/10.21008/j.0860-6897.2025.1.14.
10. Gmyrek S., Hossa R. (2025b), Robust speech parametrization based on pitch synchronized cepstral solutions, *International Journal of Electronics and Telecommunications*, **71**(3): 1–7, https://doi.org/10.24425/ijet.2025.153614.
11. Gmyrek S., Hossa R., Makowski R. (2023), Reducing the impact of fundamental frequency on the HFCC parameters of the speech signal, [in:] *2023 Signal Processing Symposium (SPSympo)*, pp. 49–52, https://doi.org/10.23919/SPSympo57300.2023.10302705.
12. Gmyrek S., Hossa R., Makowski R. (2024), Amplitude spectrum correction to improve speech signal classification quality, *International Journal of Electronics and Telecommunications*, **70**(3): 569–574, https://doi.org/10.24425/ijet.2024.149580.
13. Goodfellow I., Bengio Y., Courville A. (2016), *Deep Learning*, MIT Press.
14. Hastie T., Tibshirani R., Friedman J. (2001), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York.

15. Hazen T. (2000), A comparison of novel techniques for rapid speaker adaptation, *Speech Communication*, **31**(1): 15–33, https://doi.org/10.1016/S0167-6393(99)00059-X.

16. Hossa R., Makowski R. (2016), An effective speaker clustering method using UBM and ultra-short training utterances, *Archives of Acoustics*, **41**(1): 107–118, https://doi.org/10.1515/aoa-2016-0011.

17. Jassem W. (1973), *Fundamentals of Acoustic Phonetics* [in Polish: *Podstawy Fonetyki Akustycznej*], PWN.

18. Kuhn M., Johnson K. (2013), *Applied Predictive Modeling*, Springer, New York.

19. Kundegorski M., Jackson P.J.B., Ziółko B. (2014), Two-microphone dereverberation for automatic speech recognition of Polish, *Archives of Acoustics*, **39**(3): 411–420, https://doi.org/10.2478/aoa-2014-0045.

20. Libal U., Biernacki P. (2024a), Drone flight detection at an entrance to a beehive based on audio signals, *Archives of Acoustics*, **49**(3): 459–468, https://doi.org/10.24425/aoa.2024.148796.

21. Libal U., Biernacki P. (2024b), MFCC-based sound classification of honey bees, *International Journal of Electronics and Telecommunications*, **70**(4): 849–853, https://doi.org/10.24425/ijet.2024.152069.

22. Libal U., Biernacki P. (2024c), MFCC selection by LASSO for honey bee classification, *Applied Sciences*, **14**(2): 913, https://doi.org/10.3390/app14020913.

23. Maciejko W. (2015), The effect of voice over IP transmission degradations on MAP-EM-GMM speaker verification performance, *Archives of Acoustics*, **40**(3): 407–417, https://doi.org/10.1515/aoa-2015-0042.

24. Makowski R. (2011), *Automatic Speech Recognition – Selected Problems* [in Polish: *Automatyczne Rozpoznawanie Mowy – Wybrane Zagadnienia*], Oficyna Wydawnicza Politechniki Wrocławskiej.

25. Mauch M., Dixon S. (2014), Pyin: A fundamental frequency estimator using probabilistic threshold distributions, [in:] *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 659–663, https://doi.org/10.1109/ICASSP.2014.6853678.

26. McLachlan G., Peel D. (2000), *Finite Mixture Models*, Wiley-Interscience.

27. Naito M., Deng L., Sagisaka Y. (2002), Speaker clustering for speech recognition using vocal tract parameters, *Speech Communication*, **36**(3–4): 305–315, https://doi.org/10.1016/S0167-6393(00)00089-3.

28. Nedeljković Ž., Milošević M., Đurović Ž. (2020), Analysis of features and classifiers in emotion recognition systems: Case study of Slavic languages, *Archives of Acoustics*, **45**(1): 129–140, https://doi.org/10.24425/aoa.2020.132489.

29. Ng A.Y. (2004), Feature selection, L_1 vs. L_2 regularization, and rotational invariance [in:] *Proceedings of the Twenty-First International Conference on Machine Learning*, pp. 78–85, https://doi.org/10.1145/1015330.1015435.

30. Piątek Z., Kłaczyński M. (2021), Acoustic methods in identifying symptoms of emotional states, *Archives of Acoustics*, **46**(2): 259–269, https://doi.org/10.24425/aoa.2021.136580.

31. Quatieri T.F. (2001), *Discrete-Time Speech Signal Processing: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ.

32. Rabiner L., Schafer R. (2010), *Theory and Application of Digital Speech Processing*, Pearson.

33. Reynolds D.A. (2009), Gaussian mixture models, [in:] *Encyclopedia of Biometrics*, Li S.Z., Jain A. [Eds.], Springer, https://doi.org/10.1007/978-0-387-73003-5_196.

34. Rumelhart D.E., Hinton G.E., Williams R.J. (1986), Learning representations by back-propagating errors, *Nature*, **323**(6088): 533–536, https://doi.org/10.1038/323533a0.

35. Skowronski M.D., Harris J. (2004), Exploiting independent filter bandwidth of human factor cepstral coefficients in automatic speech recognition, *The Journal of the Acoustical Society of America*, **116**(3): 1774–1780, https://doi.org/10.1121/1.1777872.

36. Stefanowska A., Zieliński S.K. (2024), Speech emotion recognition using a multi-time-scale approach to feature aggregation and an ensemble of SVM classifiers, *Archives of Acoustics*, **49**(2): 153–168, https://doi.org/10.24425/aoa.2024.148784.

37. Uma Maheswari S., Shahina A., Rishickesh R., Nayeemulla Khan A. (2020), A study on the impact of Lombard effect on recognition of Hindi syllabic units using CNN based multimodal ASR systems, *Archives of Acoustics*, **45**(3): 419–431, https://doi.org/10.24425/aoa.2020.134058.

38. Upadhyaya P., Farooq O., Abidi M.R., Varshney P. (2015), Comparative study of visual feature for bimodal Hindi speech recognition, *Archives of Acoustics*, **40**(4): 609–619, https://doi.org/10.1515/aoa-2015-0061.