applied sciences

MDPI

*Article*

# 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

**Piotr Filipowicz and Bozena Kostek ***

Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdansk, Poland; s172158@student.pg.edu.pl
* Correspondence: bokostek@audioakustyka.org

**Featured Application: The present investigation shows a methodology that can support a speech therapist by automatically classifying various types of speech disorders.**

**Abstract:** This work deals with automatically detecting stuttering and its subclasses. An effective classification of stuttering along with its subclasses could find wide application in determining the severity of stuttering by speech therapists, preliminary patient diagnosis, and enabling communication with the previously mentioned voice assistants. The first part of this work provides an overview of examples of classical and deep learning methods used in automated stuttering classifications as well as databases and features used. Then, two classical algorithms (k-NN (k-nearest neighbor) and SVM (support vector machine) and several deep models (ConvLSTM; ResNetBiLstm; ResNet18; Wav2Vec2) are examined on the available stuttering dataset. The experiments investigate the influence of individual signal features such as Mel-Frequency Cepstral Coefficients (MFCCs), pitch-determining features in the signal, and various 2D speech representations on the classification results. The most successful algorithm, i.e., ResNet18, can classify speech disorders at the F1 measure of 0.93 for the general class. Additionally, deep learning shows superiority over a classical approach to stuttering disorder detection. However, due to insufficient data and the quality of the annotations, the results differ between stuttering subcategories. Observation of the impact of the number of dense layers, the amount of data in the training set, and the amount of data divided into the training and test sets on the effectiveness of stuttering event detection is provided for further use of this methodology.

**Keywords:** fluency disorder; stuttering; machine learning algorithms; ConvLSTM; ResNetBiLstm; ResNet18; Wav2Vec2; feature vector; 2D feature space representation



**Citation:** 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**, *13*, 6192. https://doi.org/10.3390/app13106192

Academic Editors: Alexander N. Pisarchik, Victor B. Kazantsev and Alexander E. Hramov

Received: 23 March 2023
Revised: 9 May 2023
Accepted: 16 May 2023
Published: 18 May 2023



## 1. Introduction

This work deals with automatic stuttering detection through machine learning. An algorithm capable of effectively classifying stuttering along with its subclasses could find wide application in determining the severity of stuttering by speech therapists, preliminary patient diagnosis, and enabling communication with the previously mentioned voice assistants. Despite its many potential applications, automatic stuttering detection employing deep learning appears sporadically in the topic of speech processing and machine learning [1–4]. At the same time, it should be noted that deep learning has dominated many areas related to speech recognition, such as speech recognition and speech synthesis [5–9].

The problem of speech impairment refers to difficulties in maintaining fluency when speaking. One category of speech disorder is stuttering, which is very complex due to its multiple subtypes in etiology [10]. It involves the involuntary addition of sounds and words and/or the inability to maintain speech fluency. It also refers to repetitions,

prolongations (difficulties with specific sounds or syllables as in part-word repetitions and prolongation), blocks while speaking words, word repetitions (or pauses in speech), and interjections (adding extra sounds, syllables, or words). It affects about 70 million people worldwide, roughly 1% of the population [11]. However, stuttering does not always affect people who have a congenital speech disorder. Even healthy people in situations of stress or nervousness can sometimes lose fluency and freedom of speech. Such a situation, however, does not imply a pathology. In addition, people who stutter suffer from different degrees of severity of the condition depending on the situation. It may arise during a conversation or while performing daily activities; it both may or may not be associated with social anxiety, which worsens this condition. People who stutter face difficulties in social and professional interactions, as well as an adverse public attitude toward stuttering [12]. Communicating with new technology through virtual voice assistants such as digital banking, flight customer services, and chatbots, and even Alexa, Google Assistant, and Siri can also be problematic [13].

Speech fluency disorders are not easily defined, as they are influenced by many factors, such as gender, age, accent, and the language spoken by the speaker. There are many classes of stuttering, each with its subclasses, which makes identifying all types of stuttering with a single model a difficult task. Even a specific kind of stuttering applied to a single word can be analyzed in different ways.

A common problem in training deep learning algorithms for speech disorder detection is the lack of sufficient training data. Many works rely on their own hand-recorded, transcribed, and labeled datasets, which are often relatively small due to the limited work put into creating them. The most widely used in the speech disorder field is the UCLASS (University College London Archive of Stuttered Speech) dataset [10]. A dataset released in 2021 by Apple, SEP28-k [14], contains approx. 28,000 short recordings containing speech disorders and may prove to be a breakthrough in this regard.

Methods for detecting stuttering use spectral features such as Mel-Frequency Cepstral Coefficients (MFCCs), Linear Prediction Cepstral Coefficients (LPCCs), pitch, zero-crossing coefficient, etc. Moreover, spectral analysis resulting in a 2D (two-dimensional) feature space is also employed. These features in the form of parameter vectors are analyzed by using statistical modeling methods such as hidden Markov models (HMM) [15,16], support vector machines (SVM) [17], neural networks [18,19], and rough sets [19]. Moreover, methods such as wavelet analysis [20] and dynamic time warping (DTW) [21] are also employed in stuttering analyses [22–24]. More recently, a 2D feature space was combined with deep learning algorithms [1–3]. An alternative strategy for detecting stuttering is to use designated signal features to create language models [25–27]. However, this is a computationally expensive and error-prone method.

The overall purpose of this study is to develop and test algorithms for stuttering detection in several contexts. However, specifically, the experimental setup aims to check the impact of applying different speech features, changing the deep model architecture, and changing the amount of data divided into the training and test sets. The paper starts with recalling the datasets and related works used in stuttering classification that constitute the background of this study. Then, the assumptions, along with a block diagram of the present investigation, are included. Several stages of the analysis performed follow this. Among the algorithms tested are two baseline, i.e., k-NN and SVM, as well as deep models, i.e., ConvLSTM, ResNetBiLstm, ResNet18, and Wav2Vec2. A transformer-based architecture, ResNet18 was determined to be the most successful algorithm, i.e., it can classify speech disorders at the F1 measure of 0.93 for the general class. Additionally, the deep learning model shows superiority over a classical approach to stuttering disorder detection. However, due to insufficient data and the low quality of the annotations, the results differ between stuttering subcategories. Observation of the impact of the number of dense layers, the amount of data in the training set, as well as the amount of data divided into training and test sets on the effectiveness of stuttering event detection is provided for further use of this methodology. Finally, a discussion and conclusion are presented.

The highlights of our work are as follows:

- We examine the impact of individual signal features such as Mel-Frequency Cepstral Coefficients (MFCCs), pitch-determining features in the signal, and various 2D speech representations on the recognition performance;
- We propose restructuring the network model so it achieves an F1 measure of 0.93 for the general class classification;
- We use the Wav2Vec2 transformer-based model that achieved better results for word and sound repetitions, which is promising for further research as there are issues with satisfactory results in those classes;
- We demonstrate that the results obtained by the best-performing algorithm are comparable to state of the art and it outperforms the literature-reported outcomes, both for general class classification and when detecting speech disfluency subclasses.

## 2. Background

### 2.1. Datasets Available

In this section, datasets containing speech disfluencies are recalled, as they constitute an essential part of the machine learning approach to the given problem. They may also be helpful for both researchers and therapists as examples of how to archive data when collecting samples of stuttered speech.

SEP-28K [14] is the most extensive collection of labeled stuttering samples published, containing multiple classes for both speech impairment and fluency in speech. It includes recordings from 385 interviews created through 8 series conducted with people with speech disorders. These episodes involve selected radio plays: He stutters; HVSA; I stutter, so what; My stuttering life; Stutter talk; Stuttering is cool; Women who stutter; Strong voices [14].

Between 40 and 250 short, 3 s segments were cut from each episode. Their final total number was 28,177. The editing of the recordings was conducted automatically using a speech detector. According to the authors, speech disorders most often occur just after, before, or during pauses in speech, and for this reason, the aforementioned several-second blocks are located around such pauses. In addition, to obtain the greatest variety in the labeled disorders, the recordings were cropped differently with respect to the localized breaks in speech. Each clip was annotated independently by three people. The evaluators received brief training in recognizing stuttering and its subcategories, but none professionally dealt with the disorders in question. This caused some errors in annotating due to the difficulty arising from labeling with audio only. Unlike other collections, the authors did not have access to any additional auxiliary information, such as physical or visual signs of stuttering.

A list of classes annotated as part of the collection is presented below. It is worth mentioning that each block can be assigned to multiple classes, both related to speech disorders and those not caused by disfluency [14]:

1. Block—a pause in a speech indicating taking a breath or stuttering;
2. Prolongation (prolongation)—prolongation of a single vowel, e.g., “M[mmm]ommy”;
3. Syllable repetition (sound repetition)—repetition of a single slab or sound, e.g., “I [pr-pr-pr-]prepared dinner”;
4. Word or phrase repetition (word/phrase repetition)—for example, “I made [made] dinner”;
5. Interjection—the insertion of a word or voice into an utterance, e.g., “um”, “uh”;
6. No dysfluencies.

The recordings also have non-speech disfluency labels to assist in speech processing:

1. Pause (natural pause)—a natural pause in speech not caused by stuttering;
2. Unintelligible passage (unintelligible)—the authors were unable to understand the speaker’s message;
3. Unsure—the denominator was uncertain into which class or classes the block should be classified;
4. No speech (no speech)—there is silence, or only noise occurring in the background;

5. Poor audio quality (poor audio quality);
6. Music—no speech, while background music can be heard.

As mentioned earlier, to minimize errors, each recording was labeled by three independent evaluators. The inter-rater reliability between the evaluators assigning grades was measured using Fleiss' kappa measure [28], which is calculated according to Equation (1) [14].

$$\kappa = \frac{P_o - P_e}{1 - P_e} \quad (1)$$

where:

- $P_o$—observed agreement between evaluators;
- $P_e$—expected agreement when data are randomly labeled.

Fleiss' kappa measure ranges from 0 to 1, with 1 indicating complete agreement and 0 indicating no compliance. The highest agreement was observed for the syllable repetition, interjection, sound repetition, and no disturbance classes, amounting to 0.62, 0.57, 0.4, and 0.39, respectively. The lowest score was obtained for interruptions and prolongations, at 0.25 and 0.11, respectively. This may be explained by the lack of additional modality beyond the audio signal when evaluating a speech sample.

Table 1 shows the distribution of classes in the SEP28-k dataset. It can be observed that the ratio of samples with any speech disorder to samples with fluent speech is about 44:56, so this is a balanced dataset, considering the general class. The percentage of the occurrence of other disfluency types, unfortunately, is not as good. Most of them amount to about 10% of the entire set. In contrast, interjections amount to 21.2% of the collection. Such a disproportion of data in particular classes may result in difficulties in training a machine learning model capable of correctly classifying all disorder events. Classes unrelated to speech disorders make up a small percentage of the data. The only marking that frequently occurs, in as much as 8% of the samples, is natural pausing, which is insignificant in sample quality.

**Table 1.** Distribution of classes in the SEP28-k dataset [14].

| Classes Related to Stuttering | Frequency of Occurrence [%] |
|---|---|
| Breaks | 12 |
| Prolongations | 10 |
| Repetitions of sounds | 8.3 |
| Repetitions of words or whole phrases | 9.8 |
| Any disruption of speech | 44.1 |
| Interjection | 21.2 |
| **Classes not related to speech impairment** | **Frequency of occurrence [%]** |
| Natural pause in speech | 8.5 |
| Unintelligible speech | 3.7 |
| Uncertainty of the evaluator | 0.1 |
| Lack of speech | 1.1 |
| Poor audio quality | 2.1 |
| Audible music | 1.1 |

SEP28-k was more extensively described as it was used in the present investigation; however, there are other databases available, including UCLASS [10], LibriSpeech [29], TORGO [30], and FluencyBank [31]. The FluencyBank database contains many recordings of children and adults with various types of speech disorders, including stuttering [31]. The labels for these recordings are transcriptions saved in CHAT format. Unfortunately, not all have text transcripts and are labeled with speech disorders. The disadvantage of this collection is that the pathologies in speech, unlike SEP-28k, are labeled phonetically. To use the data from FluencyBank, the CLAN program must be employed. There are also inconsistencies and inaccuracies in the labeling provided by the original data authors [14].

Repeating the labeling process of the FluencyBank data, analogous to the labeling process in SEP28-k, resulted in 4144 short clips (3.5 h) annotated according to the following categories [14,31]:

- Classes related to speech disorders: breaks (10.3%), prolongations (8.1%), repetitions of sounds (13.3%), repetitions of words or whole phrases (2.5%), any speech disorder (46.9%), and interjections (27.3%);
- Classes not related to speech disorders: natural pauses in speech (2.7%), unintelligible speech (3.0%), and uncertainty of the evaluator (0.4%).

The UCLASS database contains recordings of people who stutter [10]. The authors have made both phonetic and orthographic transcriptions available. Files containing information about the recordings, such as the environment and the person's condition, are also available. Markers are provided in CHAT SFS and PRAAT TextGrids formats. Three versions of the data are available. The first is from 2004—in which only one category is available, a monologue. A total of 138 people were recorded, 18 of whom were women. The second edition is from 2008—three types of recordings are available: reading, monologue, and conversation. There are 128 conversational recordings of various people available, of which 18 are women. For reading, there are 107 people (15 women, 93 men), while for monologue, there are 82 people (6 women, 76 men). Finally, the third release refers to modifications to version one by changing the recording frequency.

The LibriSpeech dataset represents a corpus of English for automatic speech recognition (ASR) research [29]. The LibriStutter collection is a modified subset of the data included in LibriSpeech. It contains recordings of 23 men and 27 women. Synthetically generated speech disorder events were added and annotated appropriately. The impairment involves repetitions of words, sounds, and phrases; prolongations; and interjections. A total of 15,000 excerpts with speech disorders were generated, 3000 for each previously mentioned class. This collection was created for experiments based on the deep FluentNet model [32].

### *2.2. Machine Learning Application for Stuttering Detection*

Hidden Markov models (HMM) are widely used in event sequence classification. Therefore, they are particularly well suited for research on speech recognition and related problems. Among the numerous examples of research in speech disfluency is the work of Wisniewski et al. [16]. The authors attempted to detect two subclasses of speech disorder, which were prolongations and pauses in speech. The work included several experiments based on MFCC features derived from speech, the most successful of which was a model trained on isolated samples. It achieved an average accuracy for both studied events of 80%. In conducting the experiments, the authors had at their disposal a database of 500 short utterances recorded by two people with speech disorders. The researchers emphasized that HMM can be a universal method in classifying speech impairment due to the lack of a need to integrate different languages.

Tan et al., in their work [33], sought to distinguish between correct and impaired speech without categorizing stuttering. In their study, they used recorded utterances from 10 students, who were asked to record the same short utterance. In addition, they created 15 samples employing a computer-generated speech disorder. Of the 35 samples created, 20 served as a training set for estimating model parameters, while the rest were used as a test set. An accuracy of 93% was achieved using a model based on 12 MFCC features.

It is also worth mentioning the article by Nöth et al. [15], who used the HMM algorithm to find similarities between speech signals containing speech disorders and those from speakers without disfluencies in speech. For the data collection, 37 stuttering patients of ages between 12 and 45 were recorded. They were asked to read excerpts from the English story "The Northwind and the Sun." The authors concluded that there was a strong correlation between the recordings studied. They were able to distinguish between individuals of different subclasses of speech disorders based on the number and length of breaks in the signal.

The Support Vector Machine (SVM) method can be used for both regression and classification [34]. In 2013, Mahesha and Vinod used a multiclass SVM to classify three categories of stuttering, i.e., prolongation and word and syllable repetitions [17]. Experiments were conducted using three different types of acoustic features. These were LPCCs (linear predictive cepstral coefficient) [35], MFCCs (Mel-Frequency Cepstral Coefficients) [36], and LPCs (linear predictive coding). They used data from the UCLASS collection. The authors selected 20 recordings of people between the ages of 11 and 20 for the experiments. The efficiency the authors achieved was 92%, 88%, and 75%, respectively. The SVM algorithm was also used by Ravikumar and colleagues [37]. They classified syllable repetition using MFCC features and dynamic time warping (DTW). They trained and tested the algorithm on a dataset containing the recorded utterances of 15 different adults with speech disorders. In the study, the authors achieved an accuracy of 94.35%. Pálfy and Pospíchal [38] compared the accuracy of SVM using two different kernels of SVM, linear and radial basis function (RBF). A more favorable result was achieved using the linear kernel; its accuracy was 98%. While using the RBF kernel, an accuracy of 96.4% was obtained. In the experiments, the MFCCs coefficients were used as features.

Among the classical machine learning methods, one can also see methods using the k-nearest neighbor (k-NN) method, including a model presented by Chee et al. based on a feature vector containing MFCC coefficients developed for detecting repetitions and prolongations in speech [39]. The model achieved an accuracy of 90.91%. The authors used ten recorded reading passages from the UCLASS collection. The samples were selected to cover a wide age range of speech disorders. Among the data chosen, manual segmentation of the impairment was performed, extracting 55 prolongations and 55 repetitions. The same authors also investigated the effectiveness of LPCC features for the same speech disorders, as in the previously mentioned work [40]. The data used in this experiment also overlap with their previous work. An accuracy of 87.5% was achieved for the linear discriminant analysis (LDA) model and 89.77% for k-NN [40].

A noteworthy paper using these models is a 2017 paper by Ghonem et al. [41]. The presented model was capable of distinguishing between several speech disorders simultaneously. The model was trained on selected data from the UCLASS collection. Thirty-nine recordings with 18 participants were used. After averaging, the accuracy was 52.9% for disorders such as repetitions, prolongations, and fluent speech.

Neural networks were used in speech disorder detection even before today's level of technology and computing power were available. As early as 1997, Howell and his team designed a system for stuttering detection consisting of two stages [42]. The first was a model whose task was to segment speech into linguistic units, and then, in the second stage, another model assigned the created segments to the appropriate categories. Recordings made while reading the fairy tale "Arthur the Rat" were used as a dataset. Twelve children with speech disorders participated in the study. The story contains 376 words, of which 90% are monosyllabic. An accuracy of 78.01% was achieved on average in classifying the words spoken incorrectly. Geetha et al. used neural networks to detect stuttering among children [43]. The dataset contained 51 study participants. The model was trained based on information such as the type of disorder, its duration, gender, age, family information, and the child's behavioral assessment. In the study, the authors declared an accuracy of 92%. The detection of individual categories of stuttering was also carried out by Szczurowska and colleagues in 2014 [18]. The result obtained was an accuracy of 76.67% for pauses in speech. They used 40 fragments of 4 s containing disordered speech and the same number of recordings of correct speech. Kohonen self-organizing maps (SOM) and a multilayer perceptron were used for classification.

In 2013, Mahesha and Vinod presented a new type of MFCC based on the linear prediction-Hilbert transform-based MFCC (LH-MFCC) [44]. The authors declare that the improved features contain more information related to voice, articulation, and pitch. They reported a 1.79% improvement in performance over classic MFCC coefficients, measuring the average accuracy of the model on a test dataset.

The significant increase in available computing power over the past few years has led to a considerable increase in interest in deep learning, including in the field related to signal processing. A number of methods were developed that can effectively handle many types of tasks, including the classification of speech disorders. In 2021, research work was published on the development of a deep neural network for classifying subtypes of speech disorders called StutterNet [25]. The authors used the first release of the UCLASS dataset for the study, from which they selected recordings of 128 persons. StutterNet contains five layers of time delay that precede pooling layers, three layers of fully connected artificial neurons, and a softmax layer. The last layer returns predictions for all diagnosed classes. In addition, the ReLU activation function and data normalization were used after each of the mentioned elements. Moreover, after the first two fully connected layers, a dropout with a likelihood ratio of 0.2 was included. To prepare the data, the authors determined 20 MFCCs from each sample. A *k*-fold cross-validation method was used, with a *k* equal to 10. Ten experiments were conducted, in each of which 80% of the samples were randomly assigned to the training set, 10% to the validation set, and 10% to the test set. The authors declared that, at the time of publication of the paper, they achieved the highest score on the UCLASS dataset. The values of the F1-score were 0.27, 0.16, 0.46, and 0.63 for repetitions, prolongations, blocks, and fluent speech, respectively.

Another paper that used deep learning was published by Alharbi and colleagues [1]. In their work, they employed data from the UCLASS (Release Two) dataset, 25 recordings of children reading the fairy tale "Arthur the Rat," 14 recordings of people reading the text "One more week to Easter," and 27 recordings from other sources. They investigated two methods for classifying speech disorders, conditional random fields (CRF), and bidirectional long short-term memory (BLSTM) to detect stuttering events in transcriptions of stuttering speech. The researchers obtained the highest averaged F1 measure of 0.76 for the BILSTM model for six different subclasses of speech disorders. Another publication is the paper by Bhatia and colleagues [3]. They created an automated system for detecting stuttering and selecting possible therapies based on classification results. They consider prolongations, repetitions, and abnormal pauses as classes of speech disorders. The authors created and released their own data labels, classifying passages as prolongations or repetitions. They performed segmentations on each audio sample using a Python language library, then calculated a 13-dimensional MFCC-based feature vector. The authors noticed that repetitions differ strongly from prolongations, so they trained two separate models. Both used an architecture built from modified recurrent blocks, i.e., a gated recurrent convolutional neural network (GRCNN). They introduced several different types of machine learning models. The best was GRCNN, trained separately for prolongations and repetitions, achieving accuracy results of 95% and 92%, respectively. However, these were the results obtained on a validation set.

There are several publications that measure model performance on the SEP-28k dataset. Bayerl et al. [45] used the wav2vec2 model and achieved an F1-score equal to 0.22 for blocks, 0.69 for interjections, 0.41 for prolongations, 0.43 for sound repetitions, and 0.46 for word repetitions. An essential element of their publication is that they added semi-automatically generated samples to the SEP-28k dataset to address the class imbalance problem. Sheikh et al. [46] investigated the impact of multi-task and adversarial learning. They achieved F1-scores equal to 0.32, 0.34, 0.12, 0.51, 0.71 for repetitions, prolongations, blocks, interjections, and general class. The same team of scientists published another paper related to stuttering detection [47]. They managed to achieve a mean F1-score for all classes in SEP-28k equal to 0.46 by addressing the class imbalance problem with augmentations and weighting the contribution of classes in the loss function.

## 3. Method

### *3.1. Assumptions*

It was assumed that the SEP-28k database was to be employed. It was divided into a training set and a test set, with 22,523 samples in the training set and 5632 in the test

set. It was also decided to test the SVM and k-nearest neighbor algorithms, which belong to the group of classical algorithms. Both algorithms were trained for classes such as prolongation, blocking, voicing repetition, word repetition, interjection, and a general class containing all those previously mentioned. MFCCs were derived as features from speech signals. Before applying the SVM algorithm, a principal component analysis [48] was used to reduce the dimensionality of the features. The number of components to keep during the dimensionality reduction was chosen so that the amount of variance that needs to be explained was greater than 0.85. This resulted in reducing the number of features to 1025, which is 33% of the original vector. The data were also standardized to obtain a mean expected value close to zero and a standard deviation of 1. The algorithms were tested for different values of hyperparameters; the best value of the number of $k$ nearest neighbors in the case of k-NN was 5, while in the case of SVM, the most promising results were obtained by using the polynomial kernel function.

The block diagram of the experiments performed is shown in Figure 1.

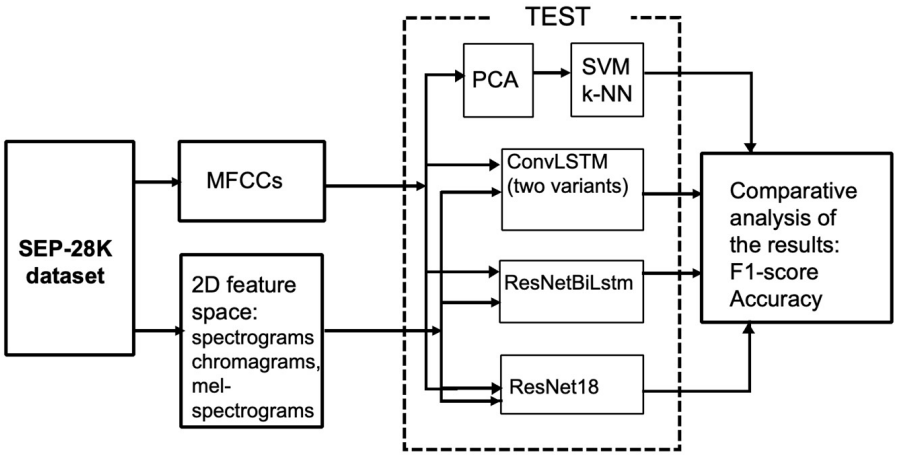


**Figure 1.** Block diagram of the experiments.

### *3.2. Classical Approach to Stuttering Detection*

Figures 2 and 3 present the F1 measure and accuracy results obtained on the test set of the best-trained versions of the k-nearest neighbor algorithm and the support vector machine for each of the classes mentioned in the assumptions of the experiments.

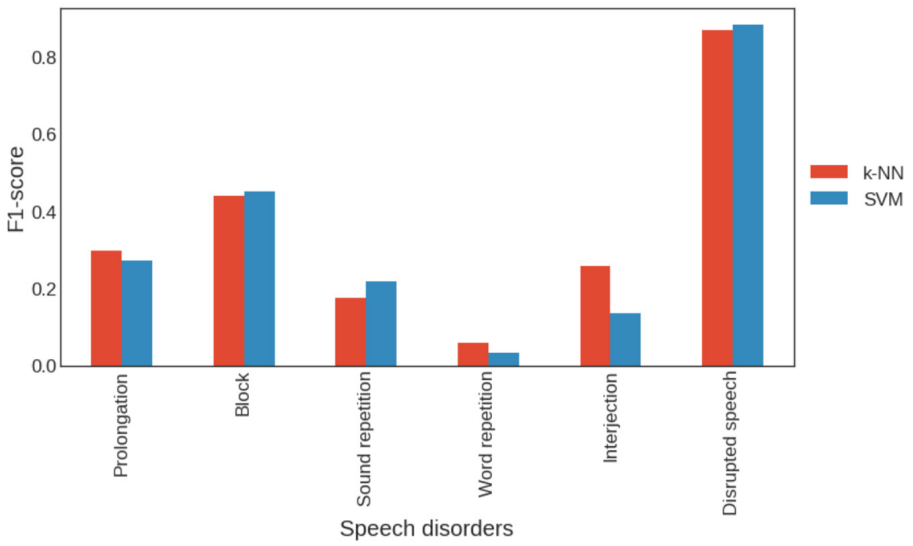


**Figure 2.** F1 measure values for SVM and k-NN algorithms.

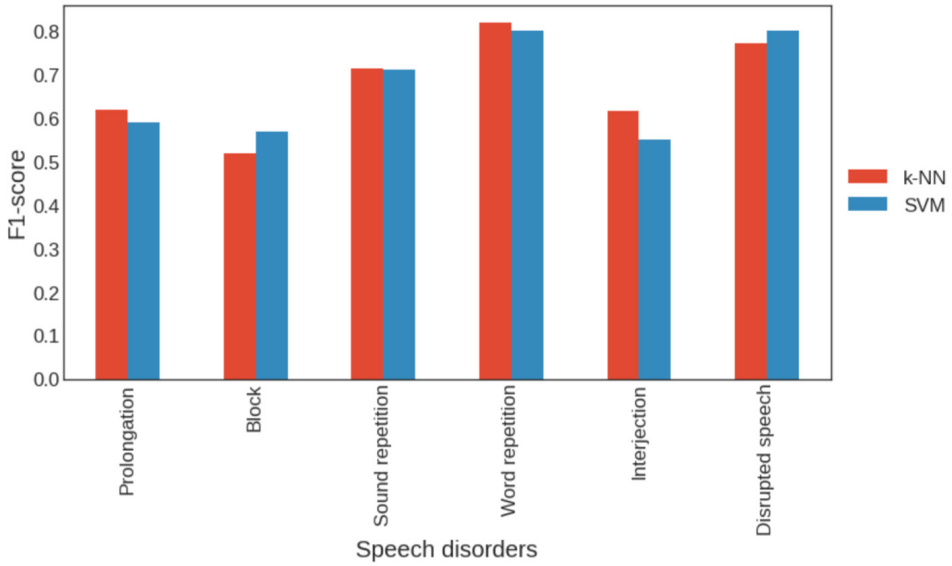


**Figure 3.** Accuracy values for SVM and k-NN algorithms.

Analyzing Figures 2 and 3, it can be seen that despite high accuracy, the values of the F1-score for most classes are quite low. This means that the models tend to label positive samples as negative, a very unfavorable phenomenon in medical pathology detection tasks. The models achieved the highest results for the general class. High accuracy was achieved for classes such as word repetition and sound repetition, but the F1 measure score was very low for them. This may be because the number of these classes in the set was very low, as presented in Table 1. Interjections and blocks were detected at a similar level, while prolongations were classified at the F1-score of about 0.6. Both algorithms achieved similar measures for most classes, but SVM appears slightly more accurate in the speech disorder classification task.

### *3.3. Deep Machine Learning Models Employed in Stuttering Detection*

As already mentioned, MFCCs are widely used in speech recognition tasks. It is believed that they can effectively extract signal features associated with speech [49,50]. In the experiments, four deep models available in the literature, i.e., ConvLSTM in two architecture variants (baseline and the one originally proposed by [14]), ResNetBiLstm, and ResNet18, were tested as backbones. The general deep model architecture is shown in Figure 4. Since the source code and weights for none of them were made publicly available, it was decided to implement it following the information contained in publications presenting the architectures.

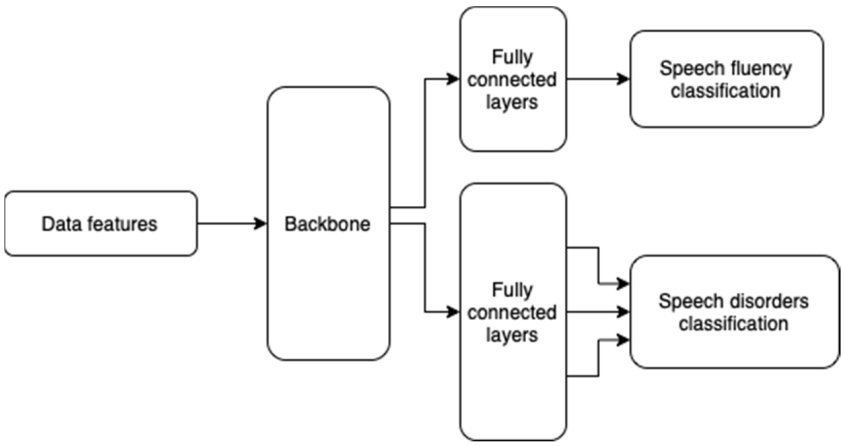


**Figure 4.** General architecture of the used models.

The baseline architecture consists of the LSTM layer, whose purpose is to model the dependencies between successive time samples and map these dependencies to particular types of stuttering. The result of this layer is a sequence of neurons, which is then fed in parallel to two dense layers. One is responsible for a binary decision on whether a given sample contains fluent speech or not. At the same time, the other is responsible for assigning a speech sample to the subclass of speech disorder in the case of dysfluency. The ConvLSTM model differs from the baseline in the cost functions used, but also in the number and type of neural layers used. It has a sequence of convolutional layers before the recursive layer. Their task is to detect complex patterns in the data received at the input. The weights thus assigned to individual pixels and their groups are then processed analogously to the basic model by recurrent networks and dense layers. Another difference between these two backbones is the cost function used. The baseline uses binary cross-entropy (BCE), shown in Equation (2). The extended version uses a separate cost function for each output type. The weighted cross entropy combined with the Focal loss function is used for speech fluency classification. In contrast, speech impairment classification uses the concordance correlation coefficient (CCC) shown in Equation (3). ResNet18 uses only convolutional layers, but they are more advanced since residual connections are used. ResNetBiLstm combines the ResNet network with two bidirectional long short-term memory blocks.

$$\mathrm{BCE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \times \mathrm{LOG}(p(y_i)) + (1 - y_i) \times \mathrm{LOG}(1 - p(y_i)) \tag{2}$$

$$\mathrm{CCC} = \frac{2\rho_{xy}\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + \left(\mu_x - \mu_y\right)^2}, \tag{3}$$

where:

$y$—label for the given class;
$N$—number of classes;
$p(y)$—predicted probability for the given class;
$\rho$—Pearson coefficient correlation between $x$ and $y$;
$\sigma$—standard deviation;
$\mu$—mean value.

As part of the data preprocessing, it was decided to employ MFCCs since they appeared in the literature in the largest number of papers. For each data sample, 47 MFCCs were determined and calculated with a window size of 25 ms. All trained models are capable of simultaneously returning results for both the general class and separately for each stuttering subclass, i.e., prolongations, blocks, voiced repetitions, word repetitions, and interjections. The sample size of the data for each model was set to 32, with the threshold limit at 0.5. Data from the SEP28-k collection were used. They were randomly divided into training, validation, and test sets, with 22,540 samples in the training set, 2816 in the validation set, and 2818 in the test set. The ratio of the general class to subclasses of stuttering in the value of the cost function is also an important parameter. It was decided to assign the general category the same weight as each subclass during the first test. Subsequent experiments investigated the effect of different types of data features on the model performance, changes in model architectures related to the number of neural layers at the end of the model, the effect of the amount of data in the training set on performance, and the effect of how the data were partitioned.

The best models for evaluation were selected on the basis of the average score of the F1 measure for all classes. Table 2 presents the values of the F1 measure and the accuracy of the models on the test set. In addition, Figures 5 and 6 present a summary of the results.

All the presented models were trained with different values of the learning rate, which were $3 \times 10^{-4}$, $1 \times 10^{-5}$, $1 \times 10^{-6}$, and $1 \times 10^{-4}$, respectively, for baseline, ConvLstm, ResNet18, and ResNetBiLstm. The models were trained for 40 epochs with 32 samples in each batch. There were two fully connected layers in each classification head, which consisted of 512 neurons in the hidden layer. The best ratio of the dropout was determined

between 0.1 and 0.2, depending on the model. The proposed models achieved the highest values of the F1 measure for a given architecture. After analyzing the presented results, we concluded that the models are much better at classifying the general class than specific disorders. The model that seems to be the most promising for further experiments is ResNetBiLstm. It achieved the highest F1 measure for each class, and achieved comparable accuracy to other architectures.

**Table 2.** Values of basic model measures on the test set.

| **Baseline** | **Prolongation** | **Block** | **Sound Repetition** | **Word Repetition** | **Interjection** | **Disrupted Speech** |
|---|---|---|---|---|---|---|
| F1-score | 0.258863 | 0.436660 | 0.067591 | 0.109167 | 0.002066 | 0.486504 |
| Accuracy | 0.699432 | 0.583392 | 0.805891 | 0.828957 | 0.657204 | 0.432931 |
| **ConvLSTM** | **Prolongation** | **Block** | **Sound repetition** | **Word repetition** | **Interjection** | **Disrupted speech** |
| F1-score | 0.245714 | 0.390687 | 0.007286 | 0.083751 | 0.268041 | 0.512953 |
| Accuracy | 0.718950 | 0.619233 | 0.806600 | 0.828957 | 0.672463 | 0.466288 |
| **ResNetBiLstm** | **Prolongation** | **Block** | **Sound repetition** | **Word repetition** | **Interjection** | **Disrupted speech** |
| F1-score | 0.283465 | 0.519135 | 0.256705 | 0.203620 | 0.438048 | 0.845312 |
| Accuracy | 0.716290 | 0.549493 | 0.697584 | 0.725643 | 0.650039 | 0.746687 |
| **ResNet18** | **Prolongation** | **Block** | **Sound repetition** | **Word repetition** | **Interjection** | **Disrupted speech** |
| F1-score | 0.307692 | 0.472831 | 0.120635 | 0.020202 | 0.349515 | 0.835989 |
| Accuracy | 0.670304 | 0.568979 | 0.784100 | 0.848792 | 0.634451 | 0.727202 |

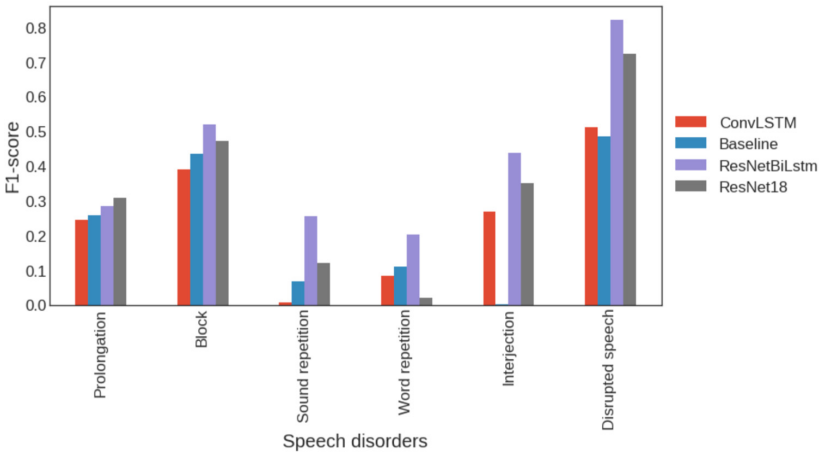


**Figure 5.** Values of F1 measure of models trained on MFCC features.

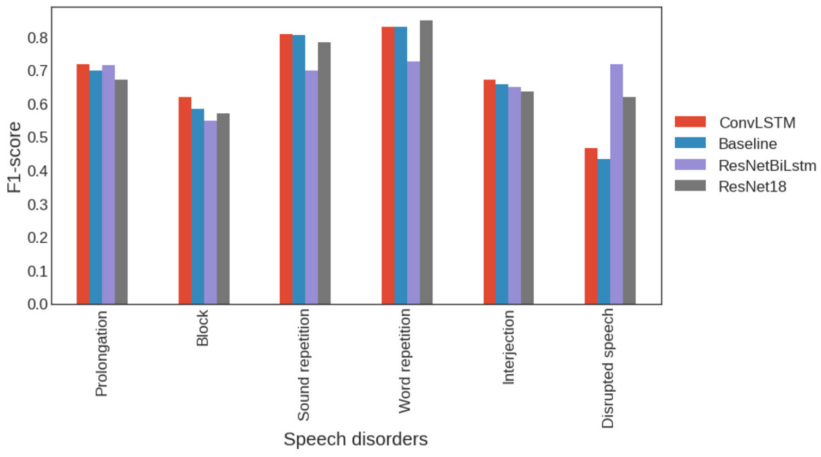


**Figure 6.** Values of the accuracy of models trained on MFCC features.

### *3.4. Impact of Feature Types*

Spectrograms and mel spectrograms, i.e., 2D feature spaces, are widely used in speech classification [50], so it is worth employing them as one of the representations incorporated into the deep models. Moreover, even though chroma features are primarily dedicated to music analysis [51], such a representation in the form of 2D feature space, i.e., chromagrams, is employed in speech analysis as well [50], but with less success [52]. One of the properties of chroma features is that they capture harmonic and melodic characteristics of music while being robust to changes in timbre and instrumentation [53]. In stuttering detection, this can be important because of the lack of impact caused by different voice timbre in speakers, so chromagrams are to be used as well.

Therefore, the performance of the models was examined using selected features as input and their combination. Spectrograms were created using the libros Python language library. The window size was set to about 25 ms, taking into account the sampling rate of 16 kHz. The parameters of the learning rate were chosen based on the results of the previous study. To perform this experiment, the size of the input data of each model had to be changed slightly. For the baseline model, which has the LSTM cell first, only the number of features in the layer input was changed, but the number of consecutive time steps remained the same. In the case of architectures that have weave layers in the first place, it was not necessary to change the size of the filters. Only one of the input dimensions changed, which only affected the output height. The chromagram was considered a complementary feature, but it was employed to see how deep learning models would perform with only this information. Finally, mel spectrograms were also tested combined with the models proposed. Variations in the configurations of model architectures were analogous to the experiment with spectrograms. An identical strategy was adopted for the selection of the learning rate. The results are shown in Figures 7–11.

The use of only chromagrams yielded the weakest results, probably because these features only carry information about the pitch in a given time window. While this is insufficient data to classify speech disorders effectively, it can be a valuable addition to the input information.

In addition, the performance of the models was checked using mel-scale-based spectrograms as input. Such a representation is often used in speech recognition [50,52], so it was deemed worth testing (see Figure 11).

After analyzing the results, we can see that the ResNetBiLstm and ResNet18 models can more accurately recognize speech disfluencies, confirming previous conclusions. The models seem to classify better when trained on a combination of all features, which are MFCCs, spectrograms, and chroma features. The exception here is the ResNet18 model, which is already a complex enough model that it may have been subject to some degree of overtraining.

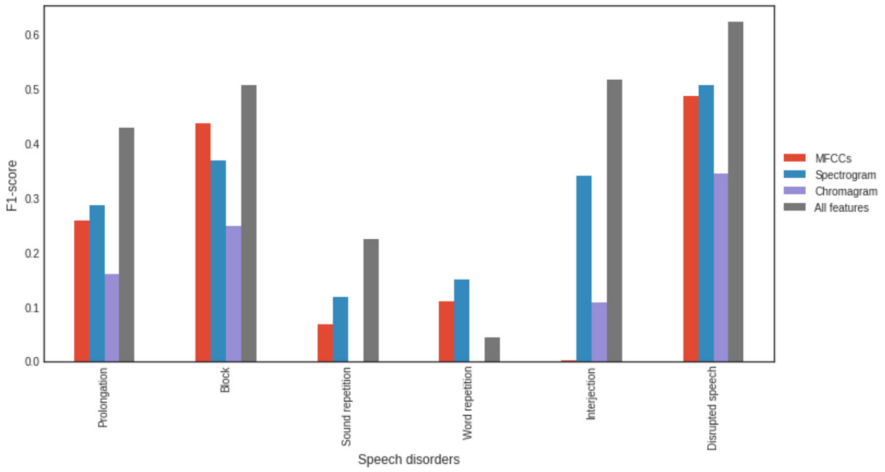


**Figure 7.** Values of the F1 measure for the baseline model on test set after training on specific features.

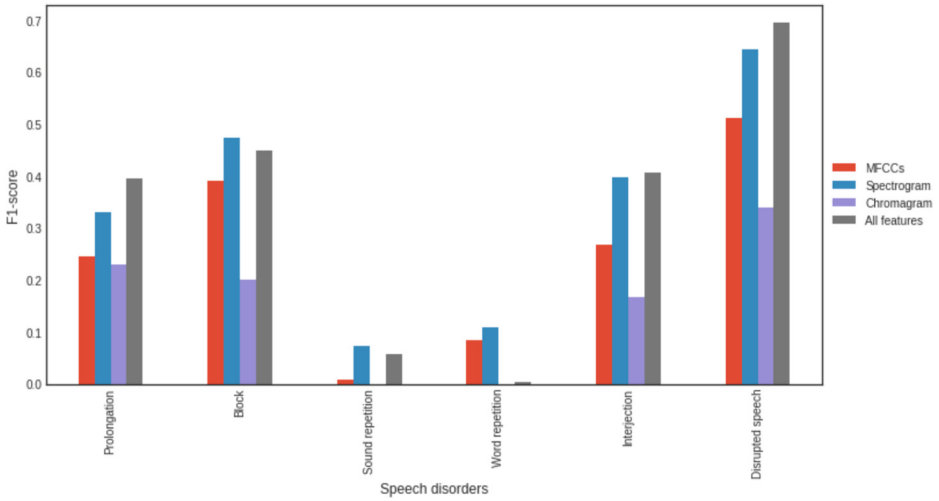


**Figure 8.** Values of the F1 measure for the ConvLSTM model on the test set after training on specific features.

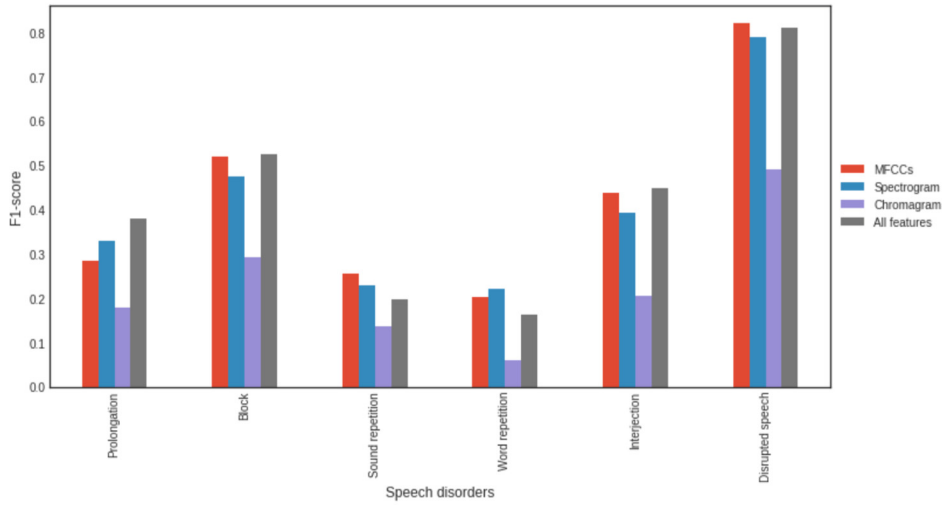


**Figure 9.** Values of the F1 measure for the ResNetBiLstm model on the test set after training on specific features.

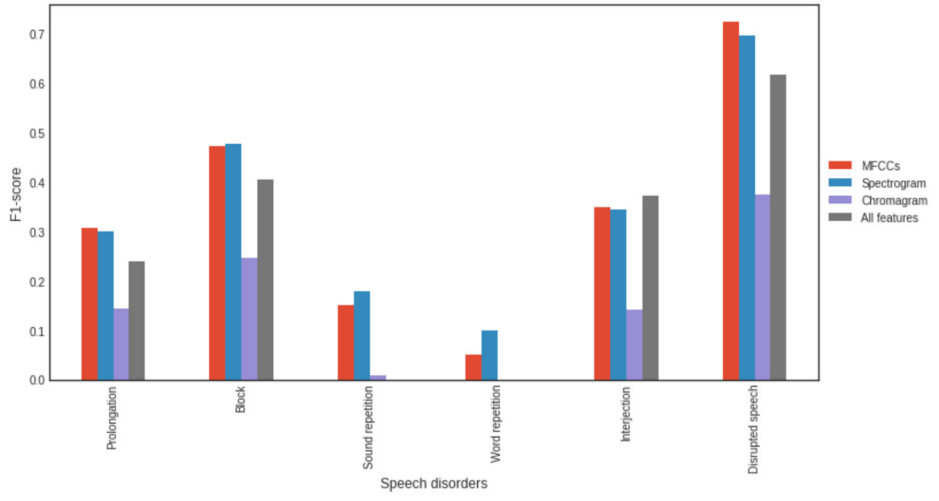


**Figure 10.** Values of the F1 measure for the ResNet18 model after training on specific features.

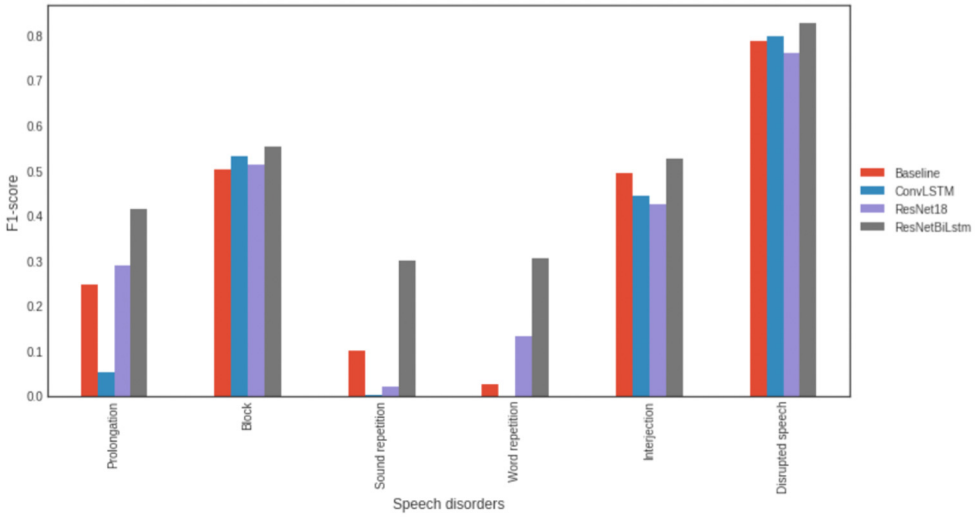


**Figure 11.** Performance of the models checked using mel-scale spectrograms.

### *3.5. Impact of the Number of Dense Neural Layers at the End of the Model*

In the previous experiments, each model output had two parallel dense neural layers. One was tasked with binary classification, determining whether the speech in a given passage was impaired, while the other classified the subcategories of stuttering. It was decided to investigate modifying the architectures by removing one of the dense layers, specifically the one responsible for general class classification. In such a variant, the overall class score is estimated based on the stuttering subclasses. Figure 12 shows this modification performed on the architectures.

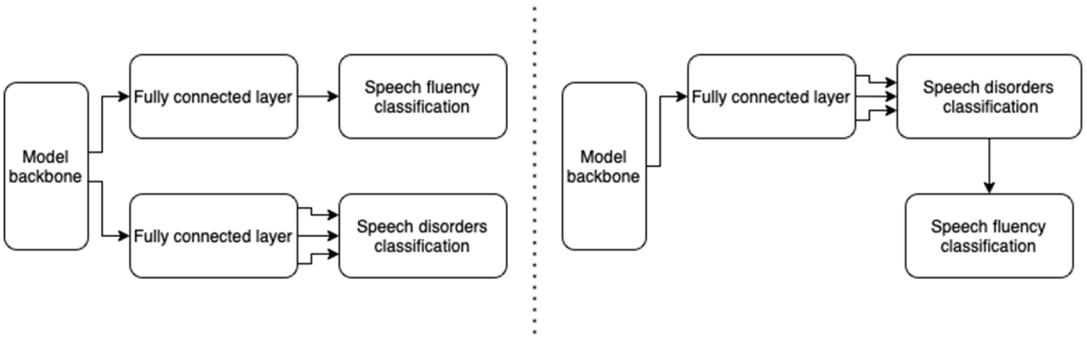


**Figure 12.** Model architecture modification.

Another modification in this experiment is the cost function, from which the factor responsible for classifying correct speech was removed. The results are shown in Figure 13.

Figure 13 clearly shows that each model achieved a significantly higher average F1 measure for stuttering class classification. This leads to the conclusion that those without a factor in the cost function responsible for general categories can better learn the detection of individual subclasses.

### *3.6. Impact of the Amount of Data in the Training Set*

In a publication presenting the SEP-28k collection [28], the authors conducted an experiment to study the effect of the amount of data in the training set on the model results. We decided to repeat this experiment and verify its assumptions. The average values of the F1 measure for the subclasses of speech disorders are shown in Figure 14. The red-color-marked SEP-6k indicates the presence of about 6000 random samples in the training set. Similarly, SEP-12k and SEP-18k denote 12,000 and 18,000 samples, respectively. The test set was the same as in the previous experiments. Hence, Figure 14 contains the

average results of the F1 measure for the models, depending on the number of training samples in the dataset.

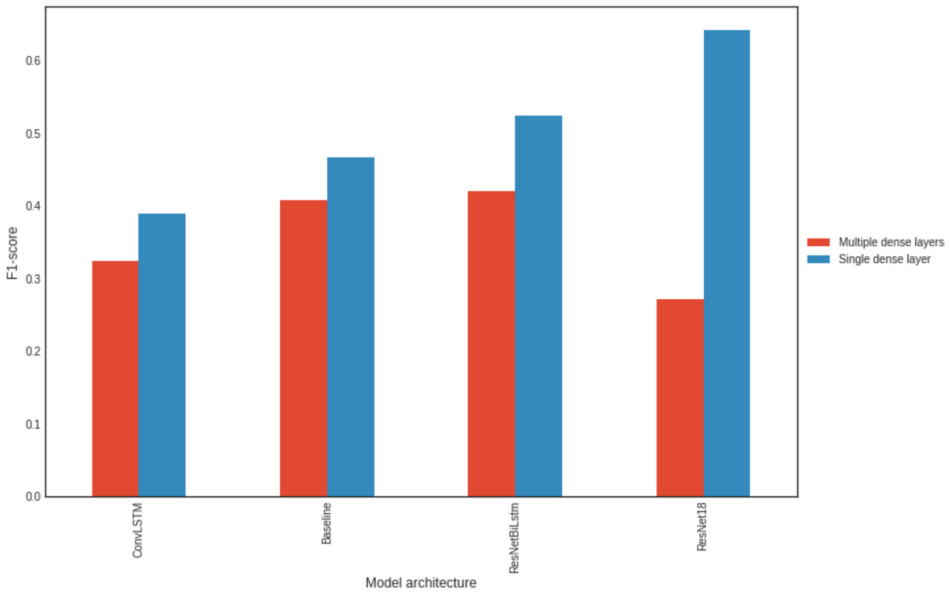


**Figure 13.** Average values of the F1 measure achieved during the experiment related to the number of dense layers at the end of the model.

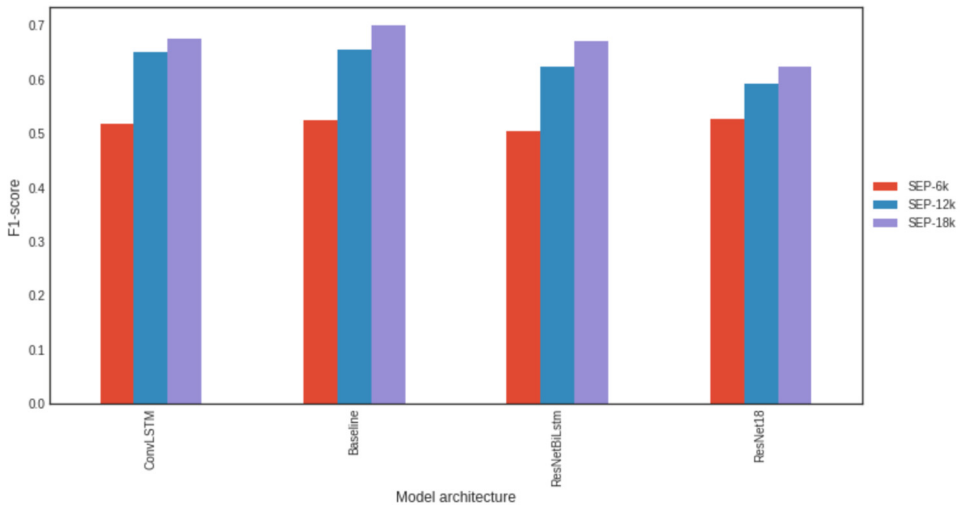


**Figure 14.** Result of the F1 measure for the general class depending on the amount of data in the training set.

Analyzing the measure obtained led to the conclusion that we were able to replicate the results of the experiment included in the publication of the SEP-28k set [14]. However, each of the models obtains a significantly higher score on the F1 measure, which indicates higher accuracy in classifying speech disorders. This confirms the need for a dataset as large as SEP-28k.

### *3.7. Impact of the Data Division into Training, Validation, and Test Sets*

The effect of the level at which the data are divided between the training, validation, and test sets was also examined. All the previous training was conducted on data divided into subsets at the level of individual samples. This was an assumption consistent with the experiments described in the baseline and ConvLSTM model information publication [14]. Splitting the data at the episode level means that recordings belonging to specific programs

are not repeated in the training and test sets. This limits the model's ability to learn a particular person's voice or way of speaking. Such a restriction should force the model to better generalize the knowledge it acquires. The last option is to split the data at the level of the entire algorithm. In this variant, the absence of recordings belonging to the presenter in the training and test collections is guaranteed. As described in the publication on the SEP28-k collection, in some series, not only did the recorded individuals have speech disorders, but the presenter was also affected. In addition, such a division makes it possible to test the model on recordings created under entirely different conditions. It is likely that each series was recorded in different rooms, and different microphones were used. All of these factors may be present in the recorded signals, which means that deep learning models can learn the features associated with them. Partitioning was performed to replicate the previous sizes of specific subsets as much as possible. In the end, there were 18,449 samples in the training set, 4643 samples in the validation set, and 5063 samples in the test set. The series chosen as the validation set were "My Stuttering Life" and "Strong Voices." The test collection included a program called "StutterTalk." Table 3 shows the distribution of classes in every split.

**Table 3.** Class distributions after different splits.

| Split Variant | Class | Train | Val | Test |
|---|---|---|---|---|
| Split by samples | Prolongation | 6849 | 850 | 847 |
| | Block | 9626 | 1183 | 1161 |
| | Sound repetition | 4485 | 582 | 547 |
| | Word repetition | 3694 | 445 | 482 |
| | Interjection | 7770 | 921 | 967 |
| Split by episode | Prolongation | 6941 | 787 | 715 |
| | Block | 9522 | 1156 | 1132 |
| | Sound repetition | 4507 | 529 | 515 |
| | Word repetition | 3678 | 441 | 444 |
| | Interjection | 7680 | 981 | 873 |
| Split by show | Prolongation | 5723 | 1486 | 1353 |
| | Block | 7848 | 2093 | 2027 |
| | Sound repetition | 4120 | 589 | 903 |
| | Word repetition | 3270 | 420 | 931 |
| | Interjection | 6780 | 1016 | 1862 |

Figure 15 shows the results of the described study. After analyzing them, we concluded that the way the data are divided has a significant impact on the results of the models. Unfortunately, in the experiment, each of the test sets has different samples. Nonetheless, it can be seen that the models perform best when split at the sample level. This may be due to the fact that in such a situation, the training set has the greatest diversity in samples. Another interpretation could be that the models are learning how specific individuals speak, which would have a negative impact on the generalization of such a system.

### *3.8. Implementation of Wav2Vec*

Recently, transformers have outperformed RNNs and CNNs in many tasks related to natural language processing and computer vision. It was decided to check the performance of one of the transformer-based architectures, named Wav2Vec2, developed and pre-trained by Facebook [54]. The model was implemented in the same way as other deep learning backbones, with the exception of extracting features. Wav2Vec2 has its own built-in feature extractors, so it needs only a raw audio signal as input. The results compared to the best-

performing ResNetBiLstm and ResNet18 are shown in Figure 16. The best model out of all the Wav2Vec2 experiments was trained for 40 epochs with 32 samples in every batch. The initial weights come from Facebook's pretraining, so the model was only fine-tuned for this specific classification task. The learning rate was set to 5e-06. According to the results, the transformer-based model did not outperform previous methods. It achieved better results only for word and sound repetitions, which is promising for further research as there were issues with satisfactory results in those classes.

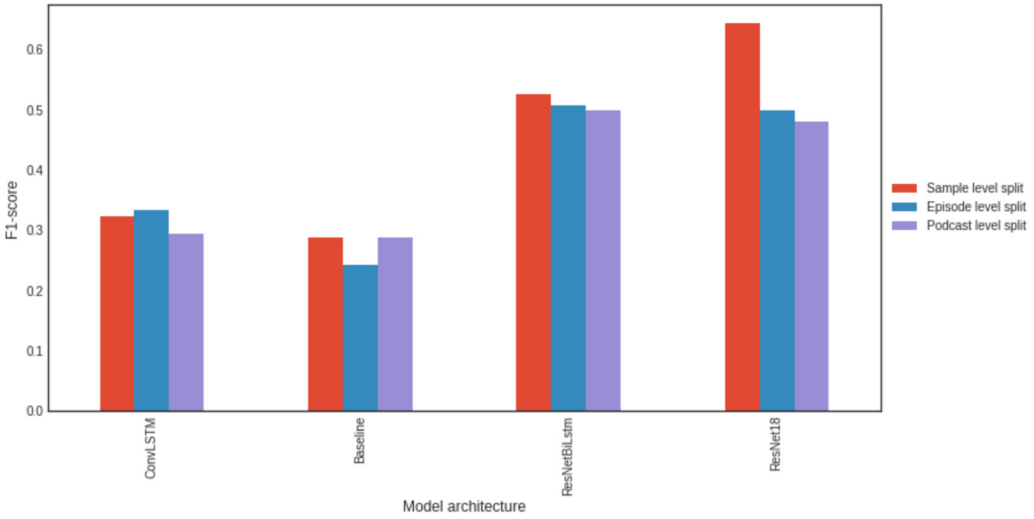


**Figure 15.** Values of F1 measure for models trained on split subsets at different levels.

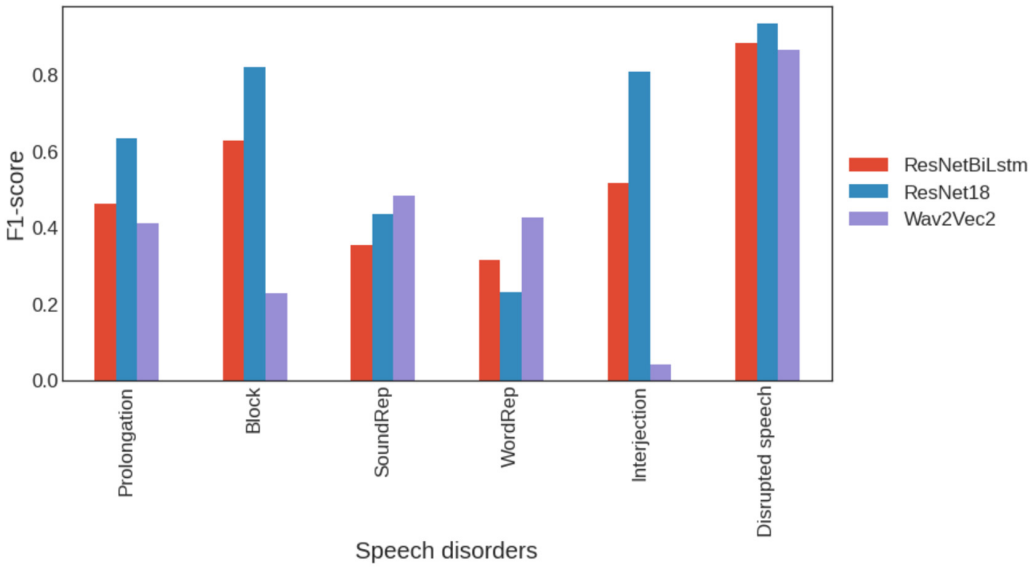


**Figure 16.** Comparison of RNN, CNN, and transformer-based methods.

It is worth noting that the ResNet18 model achieved an F1 measure score of 0.932 for the overall (disrupted) class. This is the highest score obtained in the studies conducted that has been seen in the literature.

### *3.9. Testing Algorithms on the FluencyBank Dataset*

While working on deep learning models, it is important to check their performance on an external dataset that was excluded from the training set. The FluencyBank dataset was chosen for this purpose as it has annotations made by the authors of the SEP-28k paper. The best-performing models from the previous experiments were selected for this

benchmark. Those models are ResNet18 and ResNetBiLstm with the modified architecture from Section 3.5 and Wav2Vec2 from Section 3.8. The F1-scores for specific speech disorders are shown in Figure 17. The performance of ResNet18 is the worst among the other models. ResNetBiLstm achieved the highest mean F1-score, but this is because Wav2Vec2 performed poorly in detecting blocks. Nevertheless, Wav2Vec2 was the best model in four out of six speech disorders.

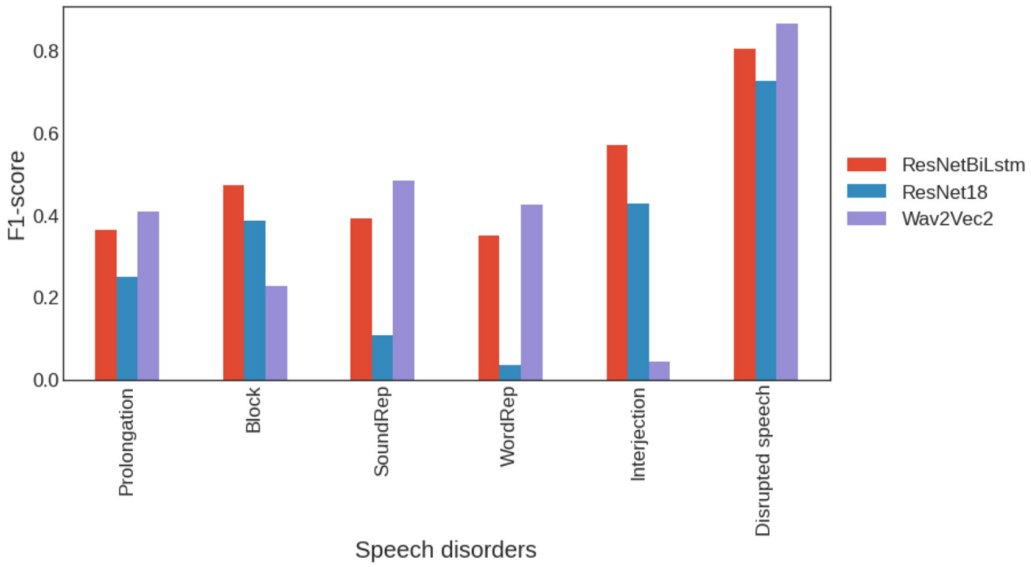


**Figure 17.** Performance of deep learning models on the FluencyBank dataset.

## 4. Conclusions

The results of the experiments indicate that SVM proved to be more effective than k-NN. Among the deep models tested, ResNet18 achieved the highest values on the SEP-28k dataset, and the ResNetBiLstm model achieved the best results on the FluencyBank dataset. It should be noted that the deep models are capable of simultaneously making predictions for each of the classes, unlike the SVM algorithm, which had to be trained separately for each class. The deep learning model achieved significantly higher values for the F1 measure. The most challenging classes to discern were found to be repetitions of both sounds and whole words. One reason may be the small number of positive samples for these classes in the dataset. Interjections and blocking were recognized with similar success. Prolongations are also among the more difficult subcategories to detect. The reason behind this may be the annotations available in the SEP-28K collection. It should be remembered that they were made by individuals who do not work with stuttering people daily. They only received training under the guidance of clinicians. However, their annotations are independent of each other, and the final grade assigned to a given fragment resulted from averaging these labels. A real benefit of making the SEP-28K collection available is the amount of data it contains. The experiments show that the number of samples in the test set significantly impacts the final result. Therefore, this is one of the constraints that was uncovered in the study. Moreover, some other limitations of such a study can be discerned. The most important seems to lie in the nature of the stuttering phenomenon. Some classes of disfluencies are less commonly seen than others, so the datasets contain imbalanced data. Moreover, if such stuttering-like events are searched automatically, then pauses, repetitions, hesitation in talking, etc., will also be treated as stuttering occurrences, even if they are not. One solution seems to be a multimodal approach in which disfluency events in speech are checked against blocks appearing on the speaker's face.

The study found that the way the data are divided has a significant impact during training. An experiment on the effect of particular features showed that they carry additional information about the data, which translates favorably into the final result of speech disorder detection. Therefore, this is another constraint limiting the study. There are a

plethora of features and feature/algorithm combinations that will influence the outcome of the study.

It was also found that it was not necessary to separate the model into two branches, one to make a binary prediction of the speech disorder and the other to classify specific subcategories of stuttering. A better result was achieved on the test set by models with a single path for predicting particular classes, with the prediction for the general class determined based on that. This led to a reduced model, which translates into a shorter inference time. Moreover, less training time and computational resources are required. All the experiments were performed on a laptop with a single NVIDIA RTX3070-class graphics card with 8 GB of built-in RAM. It should, however, also be mentioned that each model could be trained for a much larger number of epochs. Moreover, the ResNet18 model could be extended to include more convolutional layers, depending on the resources available.

## 5. Summary

In this work, a series of algorithms were implemented to classify speech disorder in a speech signal for categories such as prolongation, blocking, sound repetition, word repetition, interjection, and a general type of stuttering, the last one including all those previously mentioned. The experiments tested algorithms such as SVM, the k-nearest neighbor algorithm, the ResNet18 residual network, the ResNetBiLstm deep network, and Wav2vec2 and ConvLSTM in two variants. The ResNet18 deep network architecture was found to be the best among the tested algorithms. It achieved an F1 measure value of 0.93 on the test set. Even though results from state-of-the-art works cannot be directly used (different algorithms, datasets, data partitions, etc.), the result is still high compared to the literature, and it outperforms earlier outcomes. In addition, several experiments were conducted to study the impact of elements such as model architecture, data partitioning, amount of data, and use of individual features to check which MFCCs combined with the given algorithm handled classification the best.

**Author Contributions:** Conceptualization, P.F. and B.K.; methodology, P.F.; software, P.F.; validation, P.F. and B.K.; investigation, P.F.; data curation, P.F.; writing—original draft preparation, B.K.; writing—review and editing, B.K.; visualization, P.F.; supervision, B.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** We provided the code on github concerning the ResNet and Wave2Vec model training, as well as determining features and samples, as csv files with data division, i.e., https://github.com/filipovvsky/stuttering_detection (accessed on 10 May 2023). Links to publicly archived datasets analyzed are contained in References and cited in the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Alharbi, S.; Hasan, M.; Simons, A.; Brumfitt, S.; Green, P. Sequence Labeling to Detect Stuttering Events in Read Speech. *Comput. Speech Lang.* **2020**, *62*, 101052. [CrossRef]
2. Arnab, A.; Jayasumana, S.; Zheng, S.; Torr, P. Higher Order Conditional Random Fields in Deep Neural Networks. *arXiv* **2016**. [CrossRef]
3. Bhatia, G.; Saha, B.; Khamkar, M.; Chandwani, A.; Khot, R. Stutter Diagnosis and Therapy System, Based on Deep Learning. *arXiv* **2020**, arXiv:2007.08003.
4. Sheikh, S.; Sahidullah, M.; Hirsch, F.; Ouni, S. Machine Learning for Stuttering Identification: Review, Challenges and Future Directions. *Neurocomputing* **2022**, *514*, 385–402. [CrossRef]
5. Korzekwa, D.; Lorenzo-Trueba, J.; Drugman, T.; Kostek, B. Computer-assisted pronunciation training—Speech synthesis is almost all you need. *Speech Commun.* **2022**, *142*, 22–33. [CrossRef]
6. Li, J. Recent Advances in End-to-End Automatic Speech Recognition. *arXiv* **2021**, arXiv:2111.01690. [CrossRef]

7. Michalopoulou, Z.H.; Gerstoft, P.; Kostek, B.; Roch, M.A. Introduction to the special issue on machine learning in acoustics. *J. Acoust. Soc. Am.* **2021**, *150*, 3204–3210. [CrossRef]
8. Piotrowska, M.; Korvel, G.; Kostek, B.; Ciszewski, T.; Czyżewski, A. Machine learning-based analysis of English lateral allophones. *Int. J. Appl. Math. Comput. Sci.* **2019**, *29*, 393–405. [CrossRef]
9. Roch, M.A.; Gerstoft, P.; Kostek, B.; Michalopoulou, Z.H. How machine learning contributes to solve acoustical problems. *J. Acoust. Soc. Am.* **2021**, *17*, 48–57. [CrossRef]
10. Howell, P.; Davis, S.; Bartrip, J. The University College London Archive of Stuttered Speech (UCLASS). *J. Speech Lang. Hear. Res.* **2009**, *52*, 556–569, Erratum in: *J. Speech Lang. Hear. Res.* **2010**, *53*, 1774. [CrossRef]
11. Yairi, E.; Ambrose, N. Epidemiology of stuttering: 21st century advances. *J. Fluen. Disord.* **2013**, *38*, 66–87. [CrossRef]
12. Chu, S.H.; Unicomb, R.; Lee, J.; Cho, K.S.; Louis, K.O.S.; Harrison, E.; McConnell, G. Public attitudes toward stuttering in Malaysia. *J. Fluen. Disord.* **2022**, *74*, 105942. [CrossRef]
13. Wheeler, K. For People Who Stutter, the Convenience of Voice Assistant Technology Remains out of Reach, USA Today (Online). 2020. Available online: https://eu.usatoday.com/story/tech/2020/01/06/voice-assistants-remain-out-reach-people-who-stutter/2749115001/ (accessed on 4 March 2023).
14. Lea, C.; Mitra, V.; Joshi, A.; Kajarekar, S.; Bigham, J. SEP-28k: A Dataset for Stuttering Event Detection from Podcasts with People Who Stutter. *arXiv* **2021**, arXiv:2102.12394.
15. Nöth, E.; Niemann, H.; Haderlein, T.; Decher, M.; Eysholdt, U.; Rosanowski, F.; Wittenberg, T. Automatic stuttering recognition using Hidden Markov models. In Proceedings of the 6th International Conference on Spoken Language Processing, Beijing, China, 16–20 October 2000.
16. Wiśniewski, M.; Kuniszyk-Jóźkowiak, W.; Smołka, E.; Suszyński, W. Automatic detection of disorders with the use of Hidden Markov Model. In *Computer Recognition Systems 2*; Part of the Advances in Soft Computing; Springer: Berlin/Heidelberg, Germany, 2007; Volume 45, pp. 445–453. [CrossRef]
17. Mahesha, P.; Vinod, D. Classification of speech disfluencies using speech parameterization techniques and multiclass svm. In Proceedings of the International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, Greader Noida, India, 11–12 January 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 298–308. [CrossRef]
18. Szczurowska, I.; Kuniszyk-Jóźkowiak, W.; Smołka, E. The application of Kohonen and multilayer perceptron networks in the speech nonfluency analysis. *Arch. Acoust.* **2014**, *31*, 205–210.
19. Czyzewski, A.; Kaczmarek, A.; Kostek, B. Intelligent Processing of Stuttered Speech. *J. Intell. Inf. Syst.* **2003**, *21*, 143–171. [CrossRef]
20. Muñoz, M.; Coto-Jiménez, M. An Experimental Study on Speech Enhancement Based on a Combination of Wavelets and Deep Learning. *Computation* **2022**, *10*, 102. [CrossRef]
21. Doras, G.; Teytaut, Y.; Roebel, A. A Linear Memory CTC-Based Algorithm for Text-to-Voice Alignment of Very Long Audio Recordings. *Appl. Sci.* **2023**, *13*, 1854. [CrossRef]
22. Hariharan, M.; Fook, C.Y.; Sindhu, R.; Adom, A.H.; Yaacob, S. Objective evaluation of speech dysfluencies using wavelet packet transform with sample entropy. *Digit. Signal Process.* **2013**, *23*, 952–959. [CrossRef]
23. Yeh, P.H.; Yang, S.L.; Yang, C.C.; Shieh, M.D. Automatic Recognition of Repetitions in Stuttered Speech: Using End-Point Detection and Dynamic Time Warping. *Procedia Soc. Behav. Sci.* **2017**, *193*, 356. [CrossRef]
24. Banerjee, N.; Borah, S.; Sethi, N. Intelligent stuttering speech recognition: A succinct review. *Multimed. Tools Appl.* **2022**, *81*, 24145–24166. [CrossRef]
25. Sheikh, S.; Sahidullah, M.; Hirsch, F.; Ouni, S. StutterNet: Stuttering Detection Using Time Delay Neural Network. In Proceedings of the 2021 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, 23–27 August 2021. [CrossRef]
26. Zayats, V.; Ostendorf, M.; Hajishirzi, H. Disfluency detection using a bidirectional LSTM. In Proceedings of the Interspeech 2016, San Francisco, CA, USA, 8–12 September 2016; pp. 2523–2527. [CrossRef]
27. Chen, Q.; Chen, M.; Li, B.; Wang, W. Controllable time-delay transformer for real-time punctuation prediction and disfluency detection. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 8069–8073. [CrossRef]
28. Fleiss, J. Measuring nominal scale agreement among many raters. *Psychol. Bull.* **1971**, *76*, 378–382. [CrossRef]
29. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015.
30. Rudicz, F.; Namasivayam, A.; Wolff, T. The TORGO database of acoustic and articulatory speech from speakers with dysarthria. *Lang. Resour. Eval.* **2010**, *46*, 523–541. [CrossRef]
31. FluencyBank Database. Available online: https://fluency.talkbank.org/access/Voices-CWS.html (accessed on 4 March 2023).
32. Kourkounakis, T.; Hajavi, A.; Etemad, A. FluentNet: End-to-End Detection Of Speech Disfluency with Deep Learning. *arXiv* **2020**, arXiv:2009.11394.
33. Tan, T.; Ariff, A.; Ting, C.; Salleh, S. Application of Malay speech technology in Malay speech therapy assistance tools. In Proceedings of the 2007 International Conference on Intelligent and Advanced Systems, Kuala Lumpur, Malaysia, 25–27 November 2007; pp. 330–334.

34. Korvel, G.; Kostek, B. Comparison of Lithuanian and Polish Consonant Phonemes Based on Acoustic Analysis—Preliminary Results. *Arch. Acoust.* **2019**, *44*, 693–707. [CrossRef]
35. Mporas, I.; Ganchev, T.; Siafarikas, M.; Fakotakis, N. Comparison of Speech Features on the Speech Recognition Task. *J. Comput. Sci.* **2007**, *3*, 608–616. [CrossRef]
36. Gupta, H.; Gupta, D. LPC and LPCC method of feature extraction in Speech Recognition System. Proceedings of 2016 6th International Conference—Cloud System and Big Data Engineering (Confluence), Noida, India, 14–15 January 2016; pp. 498–502. [CrossRef]
37. Ravikumar, K.; Rajagopal, R.; Nagaraj, H. An approach for objective assessment of stuttered speech using MFCC. *ICGST Int. J. Digit. Signal Process.* **2009**, *9*, 19–24.
38. Pálfy, J.; Pospíchal, J. Recognition of repetitions using support vector machines. In Proceedings of the IEEE Signal Processing Algorithms, Architectures, Arrangements, and Applications SPA 2011, Poznan, Poland, 29–30 September 2011; pp. 1–6.
39. Chee, L.; Chia, A.O.; Hariharan, M.; Sazali, Y. MFCC based recognition of repetitions and prolongations in stuttered speech using k-nn and lda. In Proceedings of the 2009 IEEE Student Conference on Research and Development (SCOReD), Serdang, Malaysia, 16–18 November 2009; pp. 146–149. [CrossRef]
40. Chee, L.; Chia, A.O.; Hariharan, M.; Sazali, Y. Automatic detection of prolongations and repetitions using LPCC. In Proceedings of the 2009 International Conference for Technical Postgraduates (TECHPOS), Kuala Lumpur, Malaysia, 14–15 December 2009; pp. 1–4.
41. Ghonem, S.; Abdou, S.; Esmael, M.; Ghamry, N. Classification of stuttering events using i-vector. *Egypt. J. Lang. Eng.* **2017**, *4*, 11–19. [CrossRef]
42. Howell, P.; Sackin, S.; Glenn, K. Development of a two-stage procedure for the automatic recognition of dysfluencies in the speech of children who stutter: II. ANN recognition of repetitions and prolongations with supplied word segment markers. *J. Speech Lang. Hear. Res.* **1997**, *40*, 1085–1096. [CrossRef]
43. Geetha, Y.; Pratibha, K.; Ashok, R.; Ravindra, S. Classification of childhood disfluencies using neural networks. *J. Fluen. Disord.* **2000**, *25*, 99–117. [CrossRef]
44. Mahesha, P.; Vinod, D. LP-Hilbert transform based MFCC for effective discrimination of stuttering dysfluencies. In Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22–24 March 2017; pp. 2561–2565.
45. Bayerl, S.P.; Wagner, D.; Nöth, E.; Bocklet, T.; Riedhammer, K. The Influence of Dataset Partitioning on Dysfluency Detection Systems. In *Text, Speech, and Dialogue*; TSD 2022; Lecture Notes in Computer Science; Sojka, P., Horák, A., Kopeček, I., Pala, K., Eds.; Springer: Cham, Switzerland, 2022; Volume 13502. [CrossRef]
46. Sheikh, S.A.; Sahidullah, M.; Hirsch, F.; Ouni, S. Robust Stuttering Detection via MULTI-task and Adversarial Learning. In Proceedings of the 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, 29 August–2 September 2022; pp. 190–194. [CrossRef]
47. 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**, *27*, 2553–2564. [CrossRef]
48. Jolliffe, T.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A* **2016**, *374*, 20150202. [CrossRef]
49. Ganchev, T.; Fakotakis, N.; Kokkinakis, G. Comparative evaluation of various MFCC implementations on the speaker verification task Archived 2011-07-17 at the Wayback Machine. In Proceedings of the 10th International Conference on Speech and Computer (SPECOM 2005), Patras, Greece, 17–19 October 2005; Volume 1, pp. 191–194.
50. Korvel, G.; Treigys, P.; Tamulevicius, G.; Bernataviciene, J.; Kostek, B. Analysis of 2D Feature Spaces for Deep Learning-Based Speech Recognition. *J. Audio Eng. Soc.* **2018**, *66*, 1072–1081. [CrossRef]
51. Müller, M.; Kurth, F.; Clausen, M. Audio Matching via Chroma-Based Statistical Features. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), London, UK, 11–15 September 2005; pp. 288–295.
52. Alías, F.; Socoró, J.C.; Sevillano, X. A Review of Physical and Perceptual Feature Extraction Techniques for Speech, Music and Environmental Sounds. *Appl. Sci.* **2016**, *6*, 143. [CrossRef]
53. Zhu, Y.; Kankanhalli, M.S. Precise pitch profile feature extraction from musical audio for key detection. *IEEE Trans. Multimedia* **2006**, *8*, 575–584. [CrossRef]
54. Baevski, A.; Zhou, H.; Abdelrahman, M.; Auli, M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, 6–12 December 2020; pp. 12449–12460.