Journal of Intelligent Systems and Internet of Things
Vol. 17, No. 01, PP. 89-105, 2025
DOI: https://doi.org/10.54216/JISIoT.170107

A Deep Learning-Based Guidance for Stuttering Prediction

Rajeswary Nair[1,*], K. S. Kannan[2]
[1]Department of Computer Applications, Kalasalingam Academy of Research and Education, Srivilliputtur, India
[2]Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Srivilliputtur, India
Email: rajeswarynr@gmail.com; saikannan2012@gmail.com

Abstract

Advanced stuttering detection and classification using artificial intelligence is the main emphasis of this work. Determining the degree of stuttering for speech therapists, providing an early patient diagnosis and facilitating communication with voice assistants are just a few of the uses for an efficient classification of stuttering and its subclasses. This work's first portion examines the databases and features utilized, along with the deep learning and classical methods used for automated stuttering categorization. The Bayesian Bi-directional Long Short Memory with Fully Convoluted Classifier model (BaBi-LSTM) is a deep learning model in conjunction with an available stuttering information set. The tests evaluate the impact of individual signal features on the classification outcomes, including pitch-determining variables, different 2D speech representations, and Mel-Frequency Cepstral Coefficients (MFCCs). The suggested technique turns out to be the most successful, obtaining a 95% F1 measure for the entire class. When detecting stuttering disorders, deep learning algorithms outperform classical methods. However, the results differ amongst stuttering subtypes because of incomplete data and poor annotation quality. The study also examines the impact of the number of thick layers, the magnitude of the training information set, and the division apportionment of data into training and evaluation groups on the effectiveness of stuttering event recognition to offer insights for future technique improvements.

Received: November 24, 2024 Revised: January 16, 2025 Accepted: March 13, 2025

Keywords: Stuttering; Prediction; Deep learning; Cepstral coefficients; Speech representation

1. Introduction

Because of our fundamental propensity for participation and contact, humans are social creatures at heart. Speaking to others verbally is a special quality of humans that is essential to their ability to share their ideas, worries, and points of view with others [1]. However, when it comes to interacting with their communities, people with speech impairments face serious social, psychological, and intellectual obstacles. It seems that there are more and more people living with disability. Approximately 1.3 billion disabled individuals worldwide require assistive technology (AT) [2], following WHO recommendations. Approximately 2 billion citizens could be on this list by 2030. One important human right that has been confirmed is the AT provision: The United Nations Convention on the Rights of Persons with Disabilities, a key international treaty promoting disability rights [3]. Diagnosis of speech impairments is linked to numerous interventions. Certain research has given treatments for therapy or aid for people with speech impairments according to the fundamental reasons for the irregularities in speech [4]. Multitudes of persons, ranging in age from 11 to 15, employ deep learning and statistical modelling techniques to identify, categorize, prognosis, and evaluate speech impairments. Robust advances in both industry and academia have been made in statistical modelling (ML), a well-known area within artificial intelligence (AI) [5] – [6]. Applications for text-to-speech and voice-to-text powered by AI have shown to be highly beneficial in terms of improving communication tools for those with speech difficulties. These tools improve speech recognition accuracy and accessibility as well as word predictability. Furthermore,

statistical modelling offers an array of potent, automated algorithms that can manage enormous volumes of data from numerous disciplines, such as core HCI technologies: speech, language, vision, and human-computer interaction interface including vocabulary context-aware prediction, recommender systems, health informatics, and more [7]. Recent studies have shown that employing statistical modeling approaches for thorough voice signal analysis to identify problematic speech has yielded positive results by identifying important components such as Spectro Temporal utterances and Mel-frequency Cepstral Coefficients (MFCCs) from these signals. The results are more dependable when these two qualities are combined. From notably using statistical modelling approaches for voice synthesis, tailored language patterns, speech recognition and enhanced communication, voice synthesis, and accessibility and user experience improvement [8]. Users of ATSs with ML capabilities can teach and modify ML patterns. Identifying the speech patterns and subtleties of individual users would be possible for statistical modelling algorithms through the collection, annotation, and analysis of huge and varied information sets of disordered speech samples. In order to create customized patterns that are integrated into assistive technology, like voice recognition or speech producing systems, this is helpful. Additionally, in the diagnosis and treatment of speech impairments, statistical modelling techniques, being data-driven techniques, can be very beneficial [9]. It is important to recognize the following disadvantages of the SLR, even with the aforementioned benefits. The majority of suggested SLRs, which exclusively examine aphasia and dysarthria, respectively, concentrated on one kind of speech defect. A patient's age that of children is taken into consideration by other SLRs. The employment of assistive technologies is the main topic [10].

Instead of concentrating on a single illness or speech analysis tool, as was the case in earlier studies, the current systematic literature review examined several speech disorders that are suited for all age groups [11]. Finding, classifying, and contrasting efficient speech disorder detection methods was the aim. The goal of the inclusive systematic review that is being suggested is to investigate how deep learning techniques might be used to recognize, categorize, and assess these illnesses. In addition, the study is centred on the use of DL to the treatment of various diseases, considering its possible environmental, psychological, or biological roots, even in the absence of impairments caused by stuttering [12]. With an emphasis on the difficulties and constraints, this paper attempts to thoroughly examine statistical modelling approaches for speech impairment detection because of stuttering. The following imply our work’s main contribution:

- ✓ This work offers an overview of the newest deep learning methods, strategies for extracting features, model performance measures, and the properties of the information sets that are acquired with an emphasis on identifying the researchers' choices for these approaches.
- ✓ Determining the categories of speech impairments that currently exist and explaining how various learning techniques target these issues.
- ✓ To propose a novel Bayesian Bi-directional Long Short Memory with a Fully Convoluted Classifier model (BaBi-LSTM) to evaluate the impact of individual signal features including pitch-determining variables, different 2D speech representations, and MFCCs.
- ✓ Analysing the shortcomings and difficulties in the assessment, classification, and detection systems for speech disorders that are now DL-based. Finding areas of weakness and possible avenues for additional study and advancements.

The investigation is drafted as: Section 2 summarizes the feature representation connected to stuttering and the methodological explanation is provided in Section 3. The results of the study are reported in Section 4. Conclusion and future directions in Section 5

2. Related works

The quantity of features that can raise computation time costs and system performance is one of the main issues facing any ASR system. Feature selection is a solution that can help minimize the number of features by improving system efficiency and eliminating features that are superfluous or unnecessary [13]. The progress of speech analysis and its applicability to people with motor impairments has received more attention in recent years. According to earlier studies, speech interaction apps for the elderly, blind, and people with low hand dexterity have made use of deep learning (DL) and statistical modeling (ML) [14] – [15]. The application of ML and DL techniques in SED has only

been skimmed at in previous academic endeavors. The standard of living for the 70 million PWS globally could be impacted in a variety of ways by the use of SED in various applications. First off, they might comprehend stuttering speech better if a strong SED is integrated with the speech assistance technology that is available today. Second, SED is crucial in simplifying and facilitating the assessment of stuttering severity in CWS, which may have an effect on the treatment strategy for CWS [16].

Even while voice assistants have become more and more popular over time, current technologies have not yet been able to generalize to grasp stuttering speech. Fluent speech data was used to train the early versions of assistive technology, including SIRI, Google, and Alexa [17]. Consequently, the current assistants will be pruning major stuttering behaviours, including prolongation, audible block, and repetitions, which may impair their performance. PWS will gain from these technologies, and combining the existing ASR with SED can enhance their capacity for generalization in order to comprehend instances of stuttering. Another critical application requiring an effective SED is stuttering evaluation [18]. It is necessary to have a speech-language pathologist (SLP) evaluate stuttering in early infancy (preschool age), particularly between the ages of three and five. According to [19], it might make stuttering easier to spot and enable 20% of CWS to get treatment before it gets worse. However, stuttering evaluation is expensive for many people, and involves a substantial investment of time and energy for SLPs and PWS. To assess the severity of stuttering, SLPs typically employ two techniques: perceptual scaling and counting process [20] – [21]. The percentage of stuttering words (%SW) or stuttering syllables (%SS) that occur most frequently (%SW) are manually counted by the SLP during the counting method. As a result, automatic SED could facilitate can make the process of evaluating severity easier while also assisting SLPs with stuttering evaluation sessions [22].

The author in [23] suggested fusing a residual network with a Bi-LSTM built on hybrid deep neural networks. The model was trained using various repeat patterns and interruptions, with STFT serving as the only auditory characteristic. The author in [22], proposed a model initially, the input signal was converted to a 256-frequency-bin STFT, indicating an end-to-end SED network. This was done by clipping the signal into a fixed-size audio clip lasting four seconds, sampled at 16 kHz. Frame-level spectral representations are obtained from these spectrograms by using a squeeze-and-excitation residual network [23]. In the last recurrent layer, to call attention to the speech's more noteworthy passages, they added a global attention mechanism. To differentiate between four stuttering episodes, using a neural network, the author of [24] suggests integrating MFCC with phoneme classifications and probabilities. A different section of the model is employed to categorize dysfluency. The 20 × 47 vector, which contained the extracted MFCC coefficients, phoneme class probabilities (18x299 dimensions), and phoneme information for speech recognition estimate with a 1D array with 299 elements, was created by downsampling the 3-second speech signal to 8 kHz [25]. Following feature extraction, three simultaneous patterns are given to the data. Each model has a temporal distribution, rectified linear Activation Function (ReLU), thick multiple layers, after which a bidirectional LSTM component is utilized. Each model's output is normalized using batch normalization and regularized with dropout layers following their merger and passage through two thick layers. Two distinct classifiers are used at the network's end to identify and categorize stuttering events. The under-sampling technique was employed by the author to address the imbalance present in the stuttering speech data. This was achieved by randomly eliminating speech samples [26]. The relevant papers demonstrate the importance of various features in the temporal, frequency, and ASR domains for accurate stuttering episode recognition.

However, the majority of research only employed one perspective on the speech signals, such as time-domain, ASR, or spectral features based on auditory input. MFCC was the only acoustic feature employed by the authors of [27]. While the authors made use of time-domain characteristics such as fundamental frequency, envelope parameters, duration energy peaks, and autocorrelation (ACF). To identify particular stuttering episodes, such as prolongation, these characteristics are crucial. Wav2Vec2.0 contextual embeddings in SED were first introduced, improved, and used by the authors of [28]. To identify recurrence occurrences and improve SED performance, these embeddings are crucial. The literature contains very little study on fused multi-feature in SED. The author in [29] added pitch articulatory elements to the frequency domain and temporal domain data for detection. The author in [30] discovered four stuttering cases by combining MFCC, phoneme classes, and probability. This work develops a multi-modal deep learning model with attention-based feature learning that learns appropriate feature dimensionality reduction by

considering a set of features encompassing temporal dynamics, pitch, and auditory spectral properties, along with speech-to-text capabilities to enhance the performance of sound event detection systems [30].

3. Methodology

Automatic speech recognition is a method where a computer can understand and respond to spoken words or utterances or stuttering is the primary element of assistive technology for those with speech difficulties. DL approaches used for ASR as shown in Fig 1 may be employed for analyse and recognize speech patterns data to produce typed content from voice. Analysing different speech signals about different the ASR system's primary objective is to recognize phonemes, syllables, words, and phrases. Patients to identify voice disorders about speech impairments can use aSRs and voice pathologists can intelligently assess patients in this regard. The training information set is randomly selected to create training and testing sets from observations of both healthy and sick voices is primarily responsible for the ASR's performance. The measure of the model's goodness of fit and assess the effectiveness and transferability of the statistical modelling model after it has been built using the learning set. To apply DL algorithms, the user's voice needs to be analysed and turned into a set of features. The most important and initial phase of automatic speech recognition is pre-processing must be applied to the raw audio signal before any features can be extracted. In this step, speech activity is detected, the vocal tract's length is normalized and the voice signal is cleaned of unwanted and ambient noise. By applying several techniques such as noise reduction, enhancing the optimization of speech recognition systems for low-latency processing is the goal of pre-processing a speech signal, which includes vocal tract length normalization, audio activity detection identification, speech pre-emphasis, audio framing, windowing, and feature extraction. The method of feature extraction selects the most discriminative characteristics parts of the audio stream that can be used to filter out background noise and unnecessary information while detecting language content. The general term for the digital process of producing the voice signal is feature extraction. Linguistic, contextual, auditory, and hybrid features can be broadly divided into four groups. This first phase might involve the use of a variety of feature extraction approaches, including:

- ✓ Acoustic analysis is the process of analysing spoken sound data to extract characteristics of voice quality, prosody, articulation, and phonation, among other areas.
- ✓ Vowel quality, tongue movement, occlusion weakening, speech time, and the synchronization of supralaryngeal and laryngeal activity are a few examples of articulation elements. Pitch, loudness, and length are examples of prosodic features. On the other hand, pitch variability, amplitude variability, and spectral noise ratio were the first three formants associated with vocal quality.
- ✓ The timbral information of sounds is recorded and the frequency-domain analysis of an audio signal structure is represented employing Mel-filterbank-based cepstral coefficients. A Mel-cepstral feature extraction is formed by the set of coefficients known as the MFCCs. With just 12 characteristics about frequency amplitude, MFCCs offer an adequate amount of frequency channels for audio analysis.
- ✓ One major challenge for the number of characteristics that any ASR system has might raise computation time costs and system performance. By eliminating features are superfluous and unnecessary while improving system performance, feature selection serves as a remedy to decrease the overall feature count.

3.1. Information set

The quality and diversity of the information set employed for model development DL and AI patterns is essential to the efficacy of these systems' stuttering detection. The importance of comprehensive information sets that span a comprehensive range of stuttering patterns, spoken input variants, and linguistic settings is highlighted by this crucial dependency. Researchers have employed a variety of information sets in the literature, including SEP-28k, and FluencyBank. In addition, some researchers have even created their own customized information sets to meet their particular study requirements. This joint project highlights the value of large and well-curated data in enhancing deep learning and artificial intelligence approaches for stuttering detection technology.

3.2 Feature-Extraction approach

In voice recognition systems, feature extraction is a crucial stage that transforms unprocessed audio signals into meaningful information for further processing. It is essential to the process of converting spoken language into digital

information, which makes communication between people and technology easier. Earlier research has focused on a number of speech characteristics, including the Mel frequency Cepstral Coefficient (MFCC).

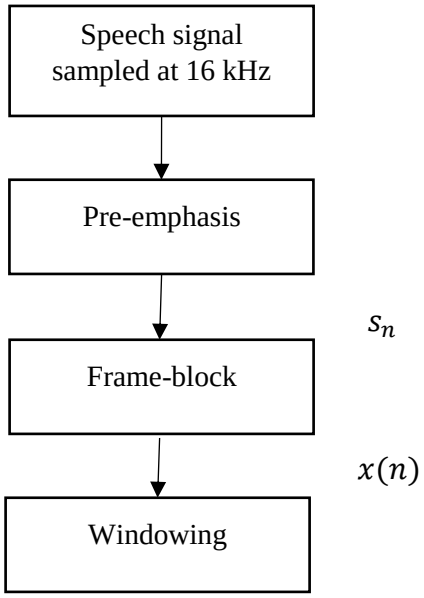

Figure 1. Speech signal pre-processing

Table 1 lists every category along with a description. In the realm of speech technology, particularly speaker identification and voice recognition, MFCC is a well-liked parameterization technique that was developed and applied extensively in recent decades. Additionally, research on speech stuttering with MFCC has been conducted. This method's key component is the MFCC acoustic properties, which are said to be reliable for voice recognition tasks. Even if MFCC analysis is twisted according to Mel-scale, its frequency is the same as cepstral analysis. Mel-frequency Cepstrum, as opposed to linearly spaced frequency bands utilizing standard Cepstrum, more closely approximates the response of the human hearing system. Logarithm is employed in MFCC to distinguish between the vocal system spectrum and the excitation spectrum. The word forms of the vocal tract are measured by the Cepstrum findings that are uttered and show gradual fluctuations in the signal spectrum. Fig 1 provides an illustration of cepstral feature vector computation. First, for every time step of N samples, the windowed signal's temporal representation is transmuted to spectral representation using the FFT. According to [20], the Critical band filter, commonly known as a triangle bandpass filter bank, is filtered before the power coefficients by the FFT block. There are two types of frequency spacing in the Mel-frequency scale: logarithmic above 1 kHz and linear below 1 kHz.

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

Table 1 shows the filter bank for the mel scale. Eq. (1) illustrates how mel-frequency is converted to linear frequency. Mel-cepstral feature extraction Coefficients (MFCC) are obtained by applying the DCT to translate the temporal log Mel spectral representation.

Table 1: Mel-scale filtering

Index	**Low freq. (Hz)**	**Middle freq. (Hz)**	**Upper freq. (Hz)**	**Bandwidth (Hz)**
1	0	50	100	100
2	100	150	200	100
3	200	250	300	100
4	300	350	400	100

5	400	450	500	100
6	500	550	600	100
7	600	650	700	100
8	700	750	800	100
9	800	850	900	100
10	900	950	1000	100
11	1000	1070	1148	146
12	1140	1230	1318	168
13	1300	1450	1510	193
14	1500	1620	1730	224
15	1700	1850	1990	256
16	1900	2130	2280	294
17	2280	2450	2620	335
18	3000	2800	3000	385
19	3450	3200	3450	445
20	3800	4200	3900	510
21	4500	4800	4540	585
22	5200	5570	5200	670
23	5900	6500	5900	770
24	6800	6400	6850	880
25	7800	7300	7860	1000

3.3. Bayesian Bi-directional LSTM

Let M represent the collection of all known ASR appliances. Let E be the total voice signal measured (E_{total}). Assuming discrete times $p(t_n) = p(nT) = p(n)$, where the sampling interval is denoted by $T = t_n - t_{n-1}$. From the M available, the active power load of the j^{th} appliance will now be shown as $p_j(n)$. Furthermore, it is thought that we have M distinct patterns, each of which is associated with an appliance. This results in a modular architecture where a new network is formed each time an appliance is added where (n) can be expressed as:

$$p(n) = \sum_{j=1}^{M} p_j(n) + e(n) \tag{2}$$

where, $e(n)$ represents the measurement's additive noise. Since there are no installed input voice, the metrics $p_j(n)$ are not available in an ASR modeling framework. Rather, just (n) is provided. Thus, the task involves estimating $p_j(n)$ given (n). Every input has a distinct spectral signature. We utilize this fundamental idea to break down the aggregate signal (n) into its constituent parts, $p_j(n)$. In reality, the spectral characteristics of a signal are obtained by integrating its values over a period of time. Therefore, to get the estimates $\hat{p}_j(n)$ of $p_j(n)$, measurements of the aggregate signal $p(n)$ must be gathered over a period of time $K + 1$. As a result, $p(n) = [p(n) \cdots p(n - K)]^T T$. After that, a non-linear relationship of (n) can be used to express the values of $p_j(n)$. Consequently, we have that

$$p_j(n) = f(p(n) + e(n) = \hat{p}_j(n) + e(n) \tag{3}$$

3.4. Bidirectional Long Short-Term model

A fully connected neural network is one method for approximating the unknown relationship $f(\cdot)$.

$$\hat{p}_j(n) = u_j(n)^T . v_j \tag{4}$$

$$u_j(n) = \begin{bmatrix} u_{j,1}(n) \\ \vdots \\ u_{jL}(n) \end{bmatrix} = \begin{bmatrix} \tanh(w_{j,1}^T . p(n)) \\ \vdots \\ \tanh(w_{j,L}^T . p(n)) \end{bmatrix} \tag{5}$$

The hyperbolic tagent is denoted by $tanh$, and the weights $w_{j,i} =, i = 1, \dots, L$ are those that connect that gathers each and every response from the hidden layer $u_{j,i}$. The appliance in which the regressor is constructed is denoted by index j. Using a set of weights v, the estimate of $\hat{p}_j(n)$ is obtained by linearly combining these non-linear adjustments. For the sake of convenience, the author has omitted the subscript j in the following since we are referring to a specific j^{th} regressor. The state vector (n) of the signals is dependent on its prior values because they randomly become dynamically active or inactive. Thus, it follows that:

$$u_i(n) = g(w_i^T . p(n) + r_i^T . u(n-1)) \tag{6}$$

where, r_i is a set of parameters that measure how much $u(n-1)$ contributed to the values of the current state. In actuality, Eq. (6) represents a short-range recurrent regression. Input signal dependencies should be considered in respect to both past and future states, since they are in fact non-causal signals. The model is to be divided into two sections: the forward pass (which relates to the past) and the backward pass (which relates to the future). In other words:

$$u_i(n) = g(w_i^T . p(n) + r_i^T . u(n-1) + \tilde{r}_i^T . u(n+1)) \tag{7}$$

Certain applications exhibit recurring patterns over extended durations, suggesting that short-range reliance is insufficient. For example, goes through multiple operating cycles are connected to one another over an extended period of time. Because of this, in this work, a bidirectional LSTM network serves in the form of foundational voice estimation using linear regression. As seen in Fig 2, each node, or memory module, is composed of three separate parts: the forgetting mechanism, input command, processing logic, and output response.

Forget gate: This part is to remove any information that isn't needed from the memory cell. The output has a range of 0 to 1; numbers near 0 signify that the entering data should be discarded, while values around 1 denote information that should be remembered.

Input node/gate: The matching state that the input node appropriately activates determines whether the "$tanh$" activation is true or false. As an alternative, the input gate controls the sigmoid function of the regression model by deciding if the associated hidden state is "significant enough."

Output gate: if the appropriate memory cell's response is "significant enough" to have an impact on the memory this indicator determines cell right in front of it.

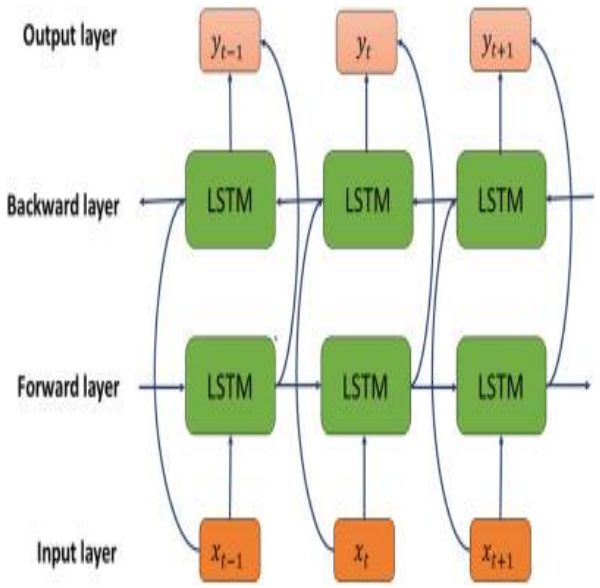

Figure 2. BaBi-LSTM

3.5. Bayesian Optimization

The choice of the recommended network's (BaBi-LST) setup settings is a crucial component of our design. To improve the model configuration parameters, this study provides a probabilistic Bayesian framework rather than using the conventional manual-based tuning method. Assume for the purposes of this discussion that a specific quantity of setup parameters, represented by the symbol π, are accessible. These characteristics include things like learning rates and the quantity of memory cells. Upon receiving the data (p), which comprises an aggregate voice signal collection across a specified time window in the form of a temporal-time series by the network, it can evaluate the error (p, d, π) that it produces. It can also be used to compare the output of the network with the specified π model configuration (i.e., target) outputs (d). This can be done by building a set of Q distinct configurations, for instance, $D1{:}\,Q = \{\pi_1 \cdots \pi_Q\}$. Here, they are referring to any time occurrence, hence index n has been omitted. The Mean Square Error (MSE) is an assumption made by researchers. Let us indicate the minimum for all Q setups as $Emin$. Next, the following improvement function is provided:

$$I(p, d, \pi) = \max\{0, E_{min} - E(p, d, \pi)\} \tag{8}$$

Given a probabilistic framework, we calculate:

$$Expect(I(p, d, \pi)) = Expect(max\{0, E_{min} - E(p, d, \pi)\} \tag{9}$$

When the error function's probability distribution is met, $P(E|D_{1:Q})$ is known can we solve Eq. (10). With the use of Bayes' conditional probability formula, the probability distribution is characterized by:

$$P\big(E\big|D_{1:Q}\big)\ \alpha\ P\big(D_{1:Q}\big|E\big)P(E) \tag{10}$$

A stochastic process with Gaussian increments with population mean μ(π) and population standard deviation Σ can be used to express $P(D_{1:Q}|E)$, given that $P(E)$ usually has a Gaussian distribution:

$$\Sigma = \begin{bmatrix} k(\pi_1, \pi_1) & \dots & k(\pi_1, \pi_Q) \\ \vdots & \ddots & \vdots \\ k(\pi_Q, \pi_1) & \dots & k(\pi_Q, \pi_Q) \end{bmatrix} \quad (11)$$

Whereas, $k(\cdot)$ represents a kernel function. Our goal in optimizing is to identify an improved configuration $\pi^* \equiv \pi_{Q+1}$ that will boost the improvement $I(p, d, \pi^*)$ or further minimize the MSE. After that, $P(D_{1:Q+1} | E)$ will once more be characterized by a Gaussian process with standard deviation for the newly extended set $D_{1:Q+1}$, which includes $\pi^* \geq \pi_{Q+1}$.

$$\begin{bmatrix} \Sigma & b \\ b^T & k(\pi_{Q+1}, \pi_{Q+1}) \end{bmatrix} \quad (12)$$

Here, $b = [k(\pi_{Q+1}, \pi_1) \dots k(\pi_{Q+1}, \pi_Q)]$. The $(E_{Q+1} | D_{1:Q}, \pi_{Q+1})$ can also demonstrated to be a Gaussian whose standard deviation and mean are related to earlier variables. Eq. (12) is employed to calculate the new configuration π^* which is the probability that $I(\cdot)$ will occur. In actuality, it is the integral of $D(\cdot)$ and $D(D_{Q+1} | D_{1:Q}, \pi_{Q+1})$.

4. Numerical results and discussion

The ensuing section details the empirical methodology findings from the suggested framework for detecting stuttering occurrences. Some existing authors used FluencyBank information sets to train and assess the model on stuttering core behaviors. Furthermore, detailed are the experimental design, assessment criteria, and outcomes. Also included are the ramifications and importance of the results for the identification of stuttering, as well as an assessment of performance differences between the suggested pattern and current patterns. The holdout test and k-fold CV were used in this work to assess the recommended model's performance. The separate-test-set method technique separated fluency disorder information sets into test and training collections so that the trained model may be evaluated on stuttering episodes that are not immediately visible. The cherry-picked debate was avoided by using random splitting with test and training sets that were 10% and 90% split. Ten-fold CV selects ten folds at random from the training set. Single-cross-validation fold is used to verify the model's mean squared error while in contrast other folds are utilized for training purposes. The manuscript's conclusions are computed using the test information set's mean of the 10 folds. The statistical metrics listed below are examined in this study.

The weighted average of $Recall$ and $P\ \boldsymbol{U}ecisMon$ is known as the F1-Score. The impact of imbalance on performance information sets is measured using it, and it is calculated as:

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (13)$$

Recall stands for the sensitivity of the model. It calculates the proportion of all stuttering occurrences in the real class, i.e. to the number of correctly predicted positive events, y = p. Recall is computed as follows:

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

Sensitivity ($TPTP+FN$) and Specificity ($TNFP+T$) are combined to get Unweighted Average Recall (UAR). The ratio of all predicted positives to all positives in the sensitivity in a binary classification problem is defined as follows. It is also known as the positive class recall at times. Specificity is defined as recall on the negative class; however, recall measures the number of true positives relative to entire number of correctly identified expected negatives to the aggregate number of negatives.

$$UAR = \frac{specificity + sensitivity}{2} \quad (15)$$

A statistical metric called the EER assesses how incorrectly the model has classified its predictions. It can be used to compute the UAR. To compute it, divide by the total number of accurate $(FP + FN)$ forecasts, the total number of incorrect $(FP + FN + TN + TP)$ projections is found. This yields Eq. (10).

$$EER = \frac{FP + FN}{FP + FN + TN + TP} \tag{16}$$

4.1. Implementation process

The model that has been suggested is constructed and trained using the TensorFlow 2.6 framework using the parameters. Researchers can effectively create and implement statistical modelling patterns with the help of TensorFlow, a Google statistical modelling framework that is available as open source. To extract sound-based attributes spectrum, melody, rhythm, and tempo aspects from the phonetic signals mentioned utilize the Librosa Python package. This library is used in audio and voice analysis. In addition, the k-fold cross-validation technique is implemented using the Python module for Sklearn. Furthermore, to give the voice signals context, a pre-trained Wav2vec2 base model is supplied. By utilizing these instruments, voice sounds may be processed and detected with great accuracy. The training sets of labels and temporal regions that go with them are specifically used to train the model. Using balanced class weights and shuffled data before each fold, it uses ten folds of k-fold cross-validation. By using the k-fold class in the sklearn package, during the training phase, there are model development and validation datasets. Learning makes use of the Adam optimizer, initialized with a learning rate of 0.0001, and binary cross-entropy as the loss function cost function. To reduce over-fitting, model training is stopped early after 5 epochs without improvement, and the optimal model weights from training are maintained. For the suggested method to be used in practical applications, processing speed and temporal complexity are essential. Inference has been carried out on 500 speech data samples using the Psutil Python package on a 12.68 GB total memory Google Colab Tesla T4 GPU. Table 2 explains the techniques used to determine the typical time needed to process data using each feature extraction method.

4.2. Ablation study

The current research aims to, a hybrid approach combining multiple features as the proposed strategy that combines spectral characteristics of audio features, ASR embedding, and time-series features retrieved from speech signals. To investigate the sensitivity of the model to different feature extraction techniques on the functionality of the suggested pattern, four sets of a series of experiments were carried out as shown in Table 3. Spectral flux onset (SFO), Zero Crossing Rate (ZCR), logits derived from the Wav2vec2 model, Mel-Frequency Cepstral Coefficients (MFCCs), and fundamental frequency were some of the experimental feature representations. Every trial raised questions about the impact of attention mechanisms. The MFCC stated in process 1 is the only acoustic characteristic of a baseline model with BaBi-LSTM block employed in the first experiment. The goal of this experiment was to use MFCC to investigate the baseline model's behavior. As a result, the Bi-LSTM block is made up of input, output, and forgot gates. Following that, MFCC-based Bayesian optimizer maps are used to process the feature maps. The bidirectional LSTM layer with 64 units, the global average pooling layer, and the time-distributed dense layer make up the RNN block. A dense layer using sigmoid activation functions processes the signal and generates a probability for each stuttering event. To observe how attention affects model performance, the baseline model which is the one shown in group 1 was tested without MFCC.

Table 2: Time complexity and processing time analysis

Features	**Time complexity**	**Processing time (ms)**
MFCC	$O(M * N \log N)$	2.2
ZCR	$O(N)$	19.5
SFO	$O(M * N + S$	39.3
FF	$O(M * N)$	312.2
ASR	$O(T * L * L)$	368.2

Table 3: Experimental feature analysis

Group	Experimentation	Features	Model	Attention
1	1	MFCC	LSTM	With feature mapping
	2			Without feature mapping
2	1	Wav2Vec2.0 logits	BaBi-LSTM	With feature mapping
	2			Without feature mapping
3	1	ZCR	Conv+RNN	With feature mapping
	2			Without feature mapping
4	1	FF	CBAM	With feature mapping
	2			Without feature mapping

Table 4: Experimental outcomes

Experiment	Prolongation	Block	Sound	Word	Interjection	F1	ERR	UAR	Recall
G1+TI	96.4	98.8	98.3	93.7	95.4	96.4	86.4	91.3	93.5
G1+T2	96.6	90.3	93.7	95.2	95.1	96.2	86.7	91.05	93.2
G2+TI	97.5	91.5	92.9	97.5	96.4	99.1	83.1	91.5	96.8
G2+T2	98.2	91.2	92.3	97.7	97.1	99.3	82.9	91.5	97.05
G3+TI	98.6	95.3	95.6	96	97.7	90.4	81.5	91.8	98.5
G3+T2	99.8	96.9	96.9	96.1	98.1	91.5	80.3	92.5	99.4
G4+TI	92.2	99.8	92.8	92.06	99.4	94.5	83.4	93.5	96.6
G4+T2	93.1	98.1	89.9	93.05	98.4	95.6	85.5	94.2	96.5

Table 5: F1-score of proposed with existing approach comparison using SEP-28k

SEP-28k	**F1-score**					**Avg. F1-score**
	Prolongation	**Block**	**Sound**	**Word**	**Interjection**	
CNN	66	58	68	63	75	66
RNN	68	61	72	67	77	69
DNN	69	62	76	69	78	71
LSTM	68	55	63	60	71	63
BiLSTM	67	58	66	53	79	64
ResNet	73	68	72	71	79.5	70
BaBi-LSTM	**92**	**98**	**89**	**92**	**99**	**94**

Table 6: F1-score of proposed with existing approach comparison using FluencyBank

SEP-28k	**F1-score**					**Avg. F1-score**
	Prolongation	**Block**	**Sound**	**Word**	**Interjection**	
CNN	25	44	47	48	52	42
RNN	42	40	54	52	53	47
DNN	41	44	45	46	54	46
LSTM	44	45	49	49	52	55
BiLSTM	60	33	60	61	84	56
ResNet	73	63	61	62	60	64
BaBi-LSTM	**97**	**96**	**94**	**95**	**92**	**91**

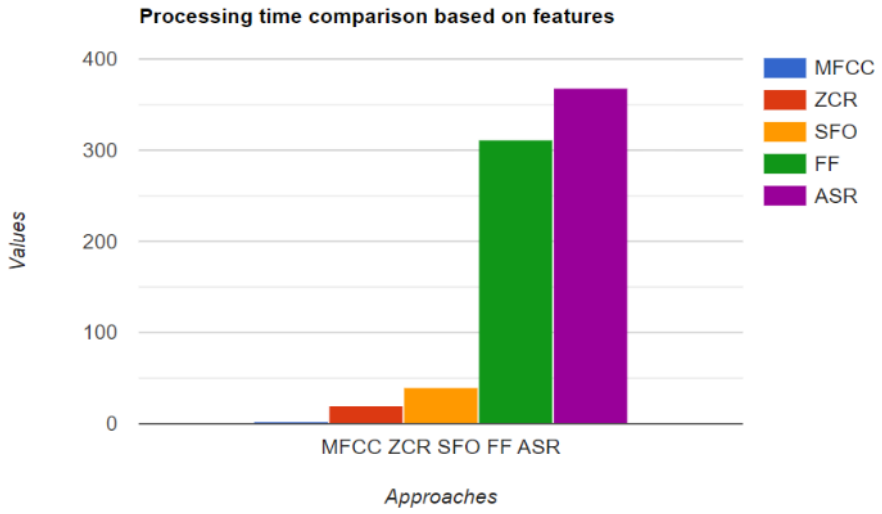

Figure 3. Processing time comparison based on features

In Fig 4a and Fig 4b, the baseline architecture and the proposed architecture without the MFCC mechanism's behavior F1-scores are shown. It was shown that the F1 scores for block (from 88% to 90%) and prolongation (from 96.41% to 96.68%) improved significantly, when the MFCC module was used. However, an investigation showed that over-fitting resulted from MFCC's significant relationship with other classes. The F1 score for sound repetition dropped as a result, going from 88% to 93%. Reducing the dimensionality of the feature collection and implementing regularization helped to address this anomaly. Additionally, there was no discernible variation in the F1-score for interjection that held steady at 85.5% and 95%, correspondingly. These findings suggest that the MFCC module can significantly accelerate the block's throughput but it may have a detrimental impact on the sound repetition task. Additionally, using other features as an exclusive acoustic feature instead of just a handcrafted MFCC may improve BaBi-LSTM performance.

To investigate the impact of contextual factors on the trial outcomes, in the second trial (group 2). The MFCC was utilized to fuse the logits of the pre-trained basic LSTM model. Researchers examined the model structure in the absence of MFCC. The outcomes show that, aside from extension, the model's performance increased in every class. The F1 score saw the largest gain, rising from 88% to 95% (4% increase) in sound repetition. This suggests that the model might be capable of efficiently learning and generalizing to sound repetition. Furthermore, there was a notable improvement observed in both the block and word repetition, with increases of 3% from 88% to 91% and 93% to 97.52%, respectively. Additionally, there was a minor but significant improvement in the interjection class, going from 75% to 76%. In contrast, prolongation went from 66% to 67% with a small rise of less than 1%. The results of the experiment show that the inclusion of MFCC had a negligible impact on the model's performance. More precisely, there was a 0.16% increase in the model's F1-score. Nonetheless, there was a 1% enhancement in F1 metric for both the delay and the interjection. As shown in Table 4, in the third trial (group 3), by combining several characteristics, auditory-based, contextual, and time-domain spectral information may be represented by the model developed a temporal understanding feature representation at the frame level. The five concurrent streams, which have followed the same structural design as the fundamental model, inherit these traits. The findings show that a multi-feature model utilizing MFCC performs better than the others in each of its five classes. The group that improved the most on stuttering instances was the sound repetition class (4.61%) suggesting that the model had difficulties identifying sound repeat in the early groups that lacked pitch and temporal data.

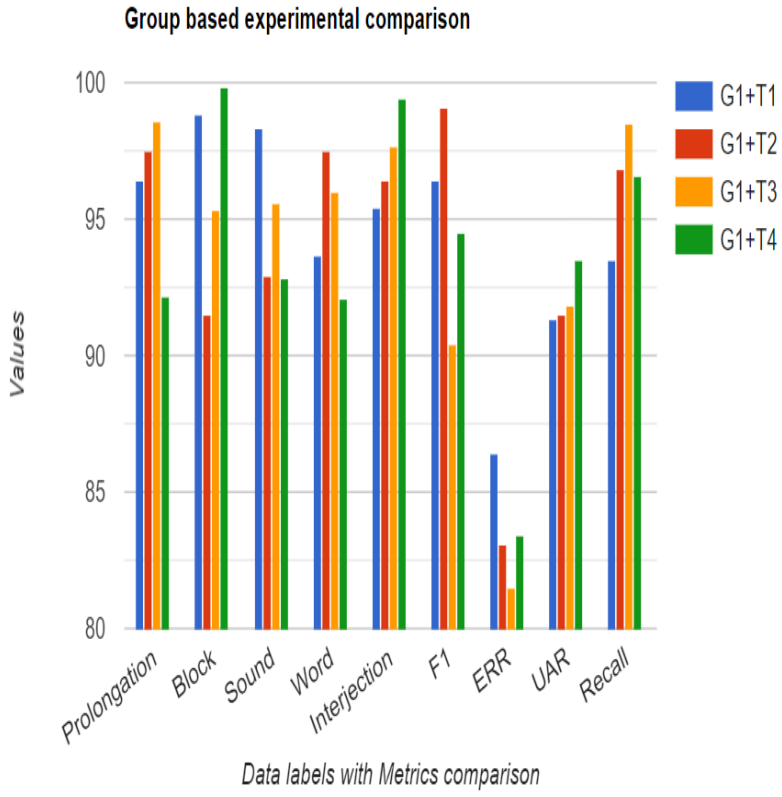

Figure 4a. Group_1 based experimental outcomes

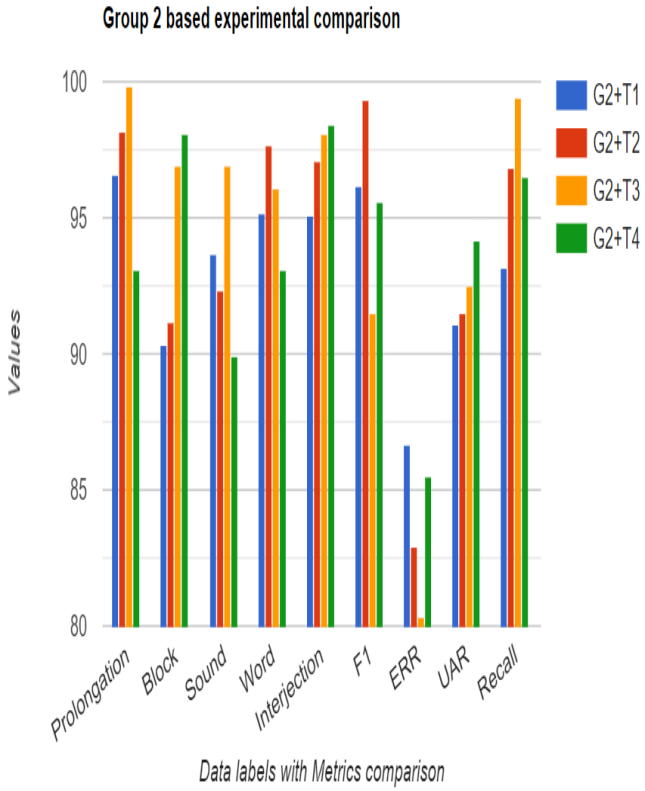

Figure 4b. Group_2 based experimental outcomes

Nonetheless, there was a 1% increase in the interjection F1-score. Additionally, there was a 1.5% increase in the block's F1 score for word extension and repetition. In the fourth trial (group 4) to address the issue of class imbalance, by using the sigmoid activation function integrated with Focal Loss (FL) and α = 0.25 and γ = 2.0, a focal weight was added in contrast to the standard cross-entropy loss function employed in the multi-class prediction algorithm. The testing results' overall F1-score grew by 3%, from 71.39% to 74.45%, in tandem with FL hyper-tuning. As shown in Fig 5a and Fig 5b, apart from the improvement in the F1 score, there was a 2% gain in UAR, a 7% decrease in EER, and an approximately 7% increase in recall.

4.3. Comparative analysis

Table 5 shows that on the SEP-28k information set, the recommended pattern achieves a mean harmonic F1-score of 94%, which represents a 4% absolute improvement over existing approach. F1-score of 91% with a high effect size. In comparison to existing works, the model using the Fluency Bank information set obtains a moderate effect size and the average F1 score increased by 3% absolute (91.41% versus 98.2%). The difference has an effect size of 0.85 ($p < 0.01$) according to Cohen's d, indicating statistical significance. Significant improvements over the previous state of the art are shown by statistical testing, with impact sizes ranging from moderate to large (0.4 to 0.7 for Cohen's d). This shows notable increases in detecting performance in real-world applications. On both information sets, the model performs better in terms of block and word repetition than the cutting-edge methods used today. To be more exact, the block event scored 66% on FluencyBank and 69% on the SEP-28k. However, the F1-score for word repetition on the SEP-28k and FluencyBank was 83% and 90%, respectively. The outperforming of current patterns indicates that the suggested model has the potential for stuttering event identification and shows a discernible improvement in the block. Three key aspects account for the model's increased performance on word and block repetition in the two information sets. First, as the data section states, there is a 25% consensus on this event, and SLPS is not the information set's annotator. As a result, the suggested model uses an annotator consistency analysis with a minimum of three independent raters to address the information set’s reliability issues.

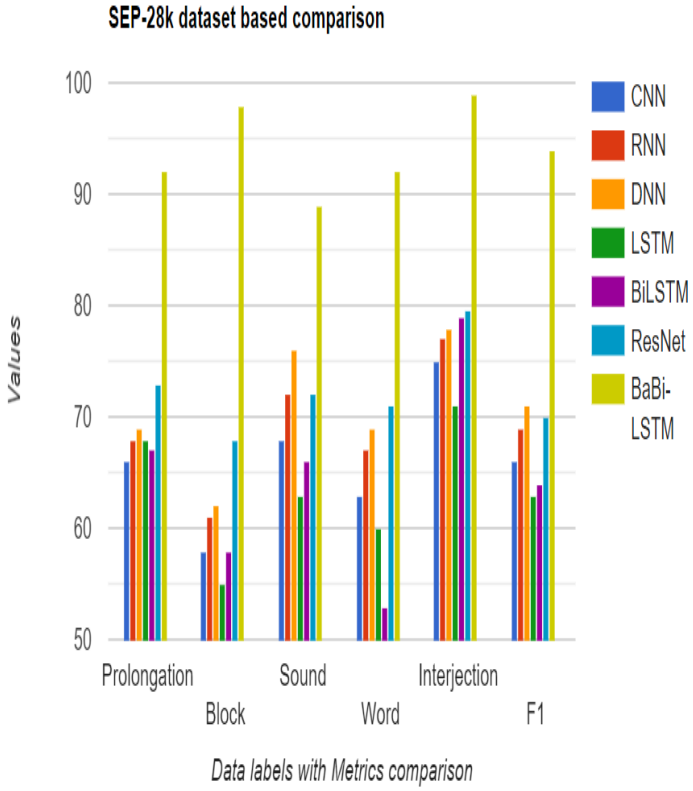

Figure 5a. SEP-28k based comparison

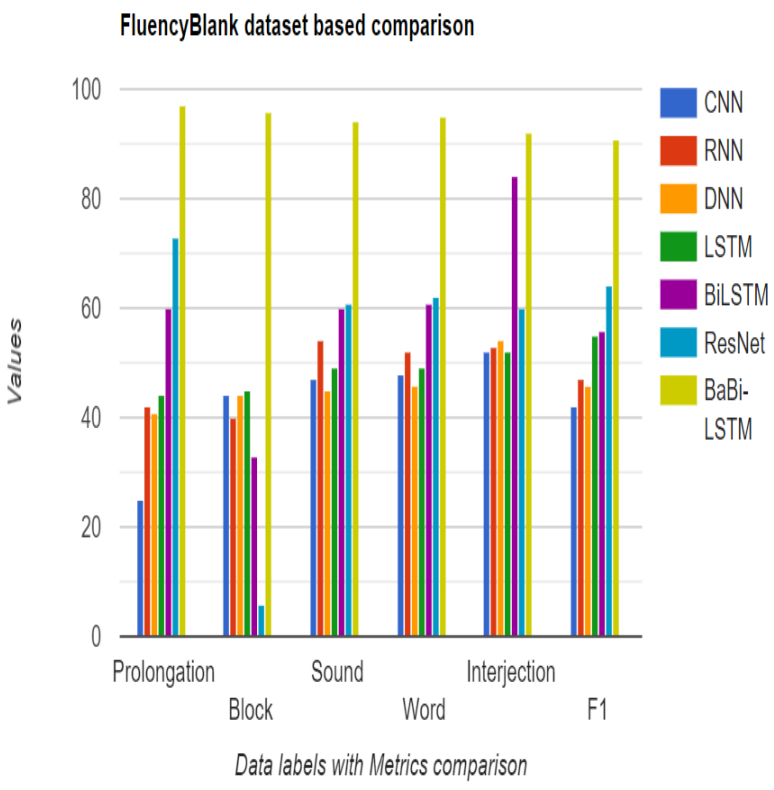

Figure 5b. FluencyBlank based comparison

Although some existing works address this issue using, the same techniques as the recommended model use the intra-rater agreement to explain the results in terms of blocks and other events. The combination of pitch, contextual, and temporal characteristics is the second element. This outcome corroborates other research in the field that suggested the significance of these traits for the detection block. Ultimately, by using the best-fit focal weights determined through extensive testing, the model successfully addresses the issue of class imbalance. The model only achieves state-of-the-art findings and performs well on 8 hours and 1.5 hours, respectively, of trustworthy data on all stuttering events from FluencyBank and SEP-28k. The model beats the most advanced approach by 7.38% from the perspective of prolongation class on the FluencyBank information set; on SEP-28k, however, only barely beats the suggested model by 73.00%. In terms of sound repetition, the model outperforms the prior demonstrated superior performance, with gains of 7.8% on SEP-28k and 66.41% on FluencyBank, relative to previous studies. Since the sound repetition performance is 11% worse than in the previous study, The F1-score was applied to minority groups are both word repetition and sound were merged into a single core behaviour class, which could be the reason for looking at the baseline outcomes of existing approaches. Thus, it could be wise to divide word and sound repetition into different groups. The author demonstrated good sound repetition performance (75%) and interjection performance (83%) on the FluencyBank information set, but it was unable to identify any more stuttering episodes. As a result, these stuttering events are covered by the phoneme probabilities and features of the articulatory vocal tract mentioned. Additionally, in a small sample, misclassification of interjection and sound due to missing data may arise from employing the agreements of three annotators to resolve inter-rater agreements. The proposed model performed worse for FluencyBank than SEP28-k did. Upon more examination, it was shown that SEP28-k had poorer annotation quality, particularly for the block class. After removing low-confidence samples from the model to enhance performance, it was retrained to deal with this.

5. Conclusion

The advantages of fully connected Bayesian Bi-directional Long Short Memory with Fully Convoluted Classifier model (BaBi-LSTM) are integrated into this work to create a more powerful hybrid acoustic model. Speech recognition tasks are used to highlight the power of BaBi-LSTM architecture. Varieties of pooling and weight-sharing mechanisms that are commonly employed in computer vision are investigated. Unfortunately, in the ASR task, no pooling strategy showed significant performance gains. The optimal result is shown to be obtained with the structure consisting of Bayesian Bi-directional Long Short Memory with Fully Convoluted Classifier model (BaBi-LSTM). Researchers also find that MFCCs are the best locally correlated feature set for BaBi-LSTM. Additionally, our team created a novel method of MFCC feature integration with the BaBi-LSTM feature. Relative improvements over the best-performing existing CNN system were 5.8% and the DNN system was 10%, respectively, for the suggested design. The proposed system was able to achieve this high gain because of speaker-adapted features and neuron characteristics. The experimental findings demonstrated the computational efficiency and competitiveness of the BaBi-LSTM system when compared to the current baseline.

References

[1] M. Pagel, "Q&A: what is human language, when did it evolve and why should we care?" *BMC Biology*, vol. 15, pp. 1–6, 2017. doi:10.1186/s12915-017-0405-3.

[2] World Health Organization, "Disability," 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/disability-and-health. [Accessed: May 17, 2024].

[3] A. Hair, P. Monroe, B. Ahmed, K. J. Ballard, and R. Gutierrez-Osuna, "Apraxia world: a speech therapy game for children with speech sound disorders," in *Proceedings of the 17th ACM Conference on Interaction Design and Children*, 2018, pp. 119–131.

[4] P. Wang and H. Van Hamme, "Benefits of pre-trained mono-and cross-lingual speech representations for spoken language understanding of Dutch dysarthric speech," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2023, pp. 1–25, 2023. doi:10.1186/s13636-023-00280-z.

[5] Y. Gu, M. Bahrani, A. Billot, et al., "A statistical modeling approach for predicting post-stroke aphasia recovery: a pilot study," in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2020, pp. 1–9. doi:10.1145/3389189.3389204.

[6] D. Mulfari, G. Meoni, M. Marini, and L. Fanucci, "Statistical modeling assistive application for users with speech disorders," *Applied Soft Computing*, vol. 103, p. 107147, 2021. doi:10.1016/j.asoc.2021.107147.

[7] S. Abderrazek, C. Fredouille, A. Ghio, M. Lalain, C. Meunier, and V. Woisard, "Interpreting deep representations of phonetic features via neuro-based concept detector: application to speech disorders due to head and neck cancer," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 200–214, 2023. doi:10.1109/TASLP.2022.3221039.

[8] V. Vashisht, A. Kumar Pandey, and S. Prakash Yadav, "Speech recognition using statistical modeling," *IEIE Transactions on Smart Processing and Computing*, vol. 10, no. 3, pp. 233–239, 2021.

[9] S. Ayanouz and A. Anouar Abdelhakim, "A smart chatbot architecture based on NLP and statistical modeling for health care assistance," in *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*, 2020, pp. 1–6.

[10] A. Zhang, "Human computer interaction system for teacher-student interaction model using statistical modeling," *International Journal of Human-Computer Interaction*, vol. 2022, pp. 1–12, 2022.

[11] A. K. Tyagi and M. Manoj Nair, "Deep learning for clinical and health informatics," in *Computational Analysis and Deep Learning for Medical Care: Principles, Methods, and Applications*, 2021, pp. 107–129.

[12] P. Janbakhshi, I. Kodrasi, and H. Bourlard, "Subspace-based learning for automatic dysarthric speech detection," *IEEE Signal Processing Letters*, vol. 28, pp. 96–100, 2020. doi:10.1109/LSP.2020.3044503.

[13] A. Tripathi, S. Bhosale, and S. K. Kopparapu, "Automatic speaker independent dysarthric speech intelligibility assessment system," *Computer Speech and Language*, vol. 69, p. 101213, 2021. doi:10.1016/j.csl.2021.101213.

[14] C. Sitaula, J. He, A. Priyadarshi, et al., "Neonatal bowel sound detection using convolutional neural network and Laplace hidden semi-Markov model," *Scientific Reports*, vol. 30, pp. 1853–1864, 2022. doi:10.1109/TASLP.2022.3178225.

[15] J.-F. Landrigan, F. Zhang, and D. Mirman, "A data-driven approach to post-stroke aphasia classification and lesion-based prediction," *Brain*, vol. 144, pp. 1372–1383, 2021. doi:10.1093/brain/awab010.

[16] K. Jothi and V. Mamatha, "A systematic review of statistical modeling based automatic speech assessment system to evaluate speech impairment," in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, IEEE, 2020, pp. 175–185.

[17] K. Bharti and P. K. Das, "A Survey on ASR Systems for Dysarthric Speech," in *Proceedings of the 2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST)*, IEEE, 2022, pp. 1–6.

[18] E. Smith, S. Hokstad, and K. A. B. Næss, "Children with Down syndrome can benefit from language interventions; Results from a systematic review and meta-analysis," *Journal of Communication Disorders*, vol. 85, p. 105992, 2020. doi:10.1016/j.jcomdis.2020.105992.

[19] M. J. Page, J. E. McKenzie, P. M. Bossuyt, et al., "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," *International Journal of Surgery*, vol. 88, p. 105906, 2021. doi:10.1016/j.ijsu.2021.105906.

[20] A. Mehrish, N. Majumder, R. Bharadwaj, R. Mihalcea, and S. Poria, "A review of deep learning techniques for speech processing," *Information Fusion*, vol. 2023, p. 101869, 2023.

[21] A. Abeysinghe, M. Fard, R. Jazar, F. Zambetta, and J. Davy, "Mel frequency cepstral coefficient temporal feature integration for classifying squeak and rattle noise," *Journal of the Acoustical Society of America*, vol. 150, pp. 193–201, 2021. doi:10.1121/10.0005201.

[22] I. Kodrasi and H. Bourlard, "Spectro-temporal sparsity characterization for dysarthric speech detection," *Green Engineering*, vol. 28, pp. 1210–1222, 2020. doi:10.1109/TASLP.2020.2985066.

[23] N. D. Cilia, C. De Stefano, F. Fontanella, and A. S. Di Freca, "A ranking-based feature selection approach for handwritten character recognition," *Pattern Recognition Letters*, vol. 121, pp. 77–86, 2019. doi:10.1016/j.patrec.2018.04.007.

[24] S. Hegde, S. Shetty, S. Rai, and T. Dodderi, "A survey on statistical modeling approaches for automatic detection of voice disorders," *Journal of Voice*, vol. 33, pp. 947–e11, 2019. doi:10.1016/j.jvoice.2018.07.014.

[25] H. Azadi, M. R. Akbarzadeh-T, H. R. Kobravi, and A. Shoeibi, "Robust voice feature selection using interval type-2 Fuzzy AHP for automated diagnosis of Parkinson's disease," *Personal Computing*, vol. 29, pp. 2792–2802, 2021. doi:10.1109/TASLP.2021.3097215.

[26] J. Kaur, A. Singh, and V. Kadyan, "Automatic speech recognition system for tonal languages: state-of-the-art survey," *Archives of Computational Methods in Engineering*, vol. 28, pp. 1039–1068, 2021. doi:10.1007/s11831-020-09414-4.

[27] M. H. Franciscatto, M. D. Del Fabro, J. C. D. Lima, et al., "Towards a speech therapy support system based on phonological processes early detection," *Natural Language Processing*, vol. 65, p. 101130, 2021. doi:10.1016/j.csl.2020.101130.

[28] S. R. Shahamiri, "Speech vision: an end-to-end deep learning-based dysarthric automatic speech recognition system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 852–861, 2021. doi:10.1109/TNSRE.2021.3076778.

[29] M. Geng, X. Xie, Z. Ye, et al., "Speaker adaptation using spectro-temporal deep features for dysarthric and elderly speech recognition," *Soft Computing*, vol. 30, pp. 2597–2611, 2022. doi:10.1109/TASLP.2022.3195113.

[30] S. Dudy, S. Bedrick, M. Asgari, and A. Kain, "Automatic analysis of pronunciations for children with speech sound disorders," *Computer Speech and Language*, vol. 50, pp. 62–84, 2018. doi:10.1016/j.csl.2017.