International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.110, May 2026

Automatic Multi-Label Stuttering Detection from Speech using Attention-Enhanced Deep Neural Networks

Ali Diyaa
Modern Academy for Computer Science and Management
Cairo, Egypt

Engy Refaai
Modern Academy for Computer Science and Management
Cairo, Egypt

Alyaa Tamer
Modern Academy for Computer Science and Management
Cairo, Egypt

Soher Mohamed
Modern Academy for Computer Science and Management
Cairo, Egypt

Aya Adel Muhammed Hassan
Phoniatrics Unit, Otolaryngology Dept.
Faculty of Medicine, Ain Shams Univ.
Cairo, Egypt

Rana Ehab
Teaching Assistant at Modern Academy for Computer Science and Management
Cairo, Egypt

Mohamed AbdelFattah
Assistant Professor at Modern Academy for Computer Science and Management
Cairo, Egypt

ABSTRACT

Speech disorders like stuttering interfere with normal speech patterns. Repeating sounds, syllables, or words; prolonging sounds for an excessive amount of time; becoming trapped in silent blocks where no sound is produced despite the speaker's best efforts to speak; or employing interjections. The speech muscles don't work properly, even though the speaker usually knows exactly what they want to say. Stuttering, which affects almost 80 million people globally, can make everyday communication feel challenging and frustrating. If left untreated, it frequently causes problems with social connections and self-confidence. A hybrid deep learning system for automatically identifying stuttering disfluencies in speech recordings is presented in this work. The method combines bidirectional long short-term memory (BiLSTM) layers, an attention mechanism (AM), and convolutional neural networks (CNN) for local acoustic feature extraction. Thirteen Mel-frequency cepstral coefficients (MFCCs) and their first-order delta and second-order delta-delta derivatives are among the many acoustic features used in the model. Evaluations on benchmark datasets, such as SEP-28K and FluencyBank, reveal F1 scores of 97.3% to 98.9% for important disfluency types and accuracy between 97.0% and 98.2%,these results are comparable to human expert agreement.

General Terms

Deep Learning, Speech Processing, Stuttering Detection.

Keywords

Stuttering Detection, Deep Learning, CNN, BiLSTM, Attention Mechanism, Speech Disfluency Classification.

1. INTRODUCTION

Stuttering is a neurological condition caused by anomalies in brain function that disrupt speech fluency. This condition can have significant psychosocial impacts, including negative self-image, adverse perceptions from others, anxiety, and, in some cases, depression. It affects 5% to 10% of preschool-aged children. Early diagnosis is vital, enabling timely intervention while the brain's compensatory mechanisms are still developing. Early therapy can reduce the likelihood of associated challenges such as social anxiety, impaired social skills, maladaptive coping strategies, and negative attitudes toward communication. Nevertheless, stuttering can persist despite early intervention, affecting approximately 1% of adults. In addition, the educational context of this study highlights that speech disorders, particularly stuttering in preschoolers, are not only medical or speech therapy issues. They are also significant factors affecting the quality of the educational process in preschool institutions.

Children between the ages of 3 and 6 undergo intense speech, social, and emotional development. At this stage, the presence of stuttering can lead to difficulties in communication with peers and educators. It may also result in lowered self-esteem and limit the child's active participation in educational activities [1].

The field of automated speech analysis has seen considerable advances in recent years, primarily due to progress in deep learning and machine learning technologies. Early research on computerized stuttering detection relied mainly on traditional signal-processing techniques and manually designed features. Common examples include mel-frequency cepstral coefficients (MFCCs), pitch variations, and energy-related measures. Typically, these features were used in conjunction with standard machine learning algorithms such as support vector machines (SVM). When tested on real-world speech data, they often failed due to differences in speakers.

Recently, deep learning approaches have attracted more interest due to their capability of directly learning complex representations from raw audio data with few hand-crafted features. Compared to other neural networks, convolutional neural networks (CNN) are more effective at capturing local acoustic patterns and realizing abrupt irregularities in speech signals [2]. Recurrent neural networks (RNNs), in particular long short-term memory (LSTM) types and bidirectional long short-term memory (BiLSTM), are particularly suited for modeling the temporal structure of speech and whether sounds change over time [3]. The combination of these architectures has proven to improve the performance of automated stuttering detection systems.

However, many existing models still process speech segments in a uniform way, without adaptively focusing on the specific moments where disfluencies actually occur. Because stuttering events are often irregular and relatively sparse within continuous speech, treating all parts of the signal equally can reduce the model's sensitivity to subtle disruptions. As a result, the performance of such systems may decrease when disfluencies are infrequent or when the model is tested on speech conditions that differ from those used during training.

This study aims to address these challenges by developing a hybrid deep learning architecture designed specifically for detecting stuttering disfluencies. The proposed model integrates convolutional layers to capture local acoustic irregularities, BiLSTM layers to model the temporal and contextual relationships within speech sequences, and a multi-head attention mechanism (AM) that dynamically highlights the most informative segments of each utterance. By allowing the model to concentrate computational focus on the parts of speech where disfluencies are more likely to occur, the system aims to improve both detection accuracy and robustness across diverse speech conditions.

2. BACKGROUND AND RELATED WORK

One of the earliest approaches was introduced in [4] , where a Hidden Markov Model(HMM) based approach for detecting fricative phoneme prolongations with an accuracy of 80%. With the advancement of deep learning, more sophisticated neural architectures were introduced for speech disfluency detection. In [5], the authors proposed a combined Deep Residual Networks (ResNet) with (BiLSTM) to detect multiple stuttering types using the UCLASS dataset. By utilizing raw acoustic features instead of language models, their approach achieved an average accuracy of 91.15% refers primarily to the training performance during the Leave-One-Speaker-Out (LOSO) cross-validation process. A deeper analysis of the results reveals that the model achieved an F1-score of only 36.5%, as stated in the StutterNet paper[6].In [6], the authors introduced StutterNet, a deep learning framework designed for stuttering detection directly from acoustic signals. The architecture is based on a Time Delay Neural Network (TDNN). For feature extraction, the system employs MFCC and Linear Prediction Cepstral Coefficients (LPCC) to represent the spectral characteristics of the speech data.The model was evaluated using the UCLASS dataset for multi-class classification of disfluencies. In terms of performance, the framework achieved an overall average F1-score of 38%.To improve end-to-end speech disfluency detection, FluentNet was proposed in [7], an end-to-end deep learning architecture for the multi-class detection of speech disfluencies. The model processes Short-Time Fourier Transform (STFT) spectrograms through a Squeeze-and-Excitation Residual Network (SE-ResNet) to extract frame-level spectral features, followed by BiLSTM layers and an attention mechanism (AM).To mitigate data scarcity, the authors introduced LibriStutter.Evaluated on the UCLASS dataset using LOSO cross-validation, the framework achieved an average accuracy of 88.50% . On the LibriStutter dataset, the model reached an average accuracy of 85.30%.In [8], a multi-branch BiLSTM architecture was developed for real-time stuttering detection and classification. They used a combination of standard (MFCC) with phoneme class probabilities to better capture the phonetic structure of disfluent speech. The architecture processes audio inputs through separate BiLSTM branches for each feature type, which are then concatenated to classify speech into categories: word repetitions, sound repetitions, interjections, and prolongations. The model's performance was validated on the SEP-28K dataset, On this dataset, the model achieved accuracies of 70% for word repetitions, 76% for sound repetitions, 78% for interjections, and 73% for prolongations, while maintaining an accuracy of 71% for non-stuttered (fluent) speech.In [9] TranStutter was introduced, transformer-based architecture designed for the classification of stuttered speech.this approach utilizes a Vision Transformer (ViT) framework to process speech represented as 2D Mel-spectrograms. The model segments these spectrograms into fixed-size patches, which are then processed through multi-head self-(AM)to capture global dependencies and long-range contextual relationships within the acoustic signal.The model's performance was validated independently across two major datasets to demonstrate its robustness. On the SEP-28K dataset, TranStutter achieved an overall accuracy of 91.13%, with specific accuracies of 89.61% for sound repetitions, 87.43% for word repetitions, 87.91% for interjections, 85.09% for prolongations, and 84.32% for blocks. On the FluencyBank dataset, the model reached an overall accuracy of 83.21%, with per-class accuracies of 82.25% for sound repetitions, 81.57% for word repetitions, 80.60% for interjections, 79.93% for prolongations, and 79.46% for blocks. More recently, the work presented in [10] proposed a CNN-based framework for automated speech disfluency classification. The methodology involved the use of upsampling techniques to balance the SEP-28K and FluencyBank datasets. For feature extraction, the system employed 13 (MFCC) as the input representation for the audio signals. the system achieved accuracies of 98.7% for Word Repetitions , 97.6% for Sound Repetitions, 97.2% for Interjections, 95.8% for Prolongations, and 95.7% for Blocks.

3. DATASET

Automatic Stuttering Detection (ASD) research benefits from several publicly available datasets, including UCLASS, SEP-28k, FluencyBank, and LibriStutter. Among these, SEP-28k has become a preferred choice in recent studies, due to its large size and detailed annotations. Critically, the reliability of its labels—each sample verified by at least three professionals—provides strong confidence in the data and supports the development of more robust models [11].SEP-28k was combined with FluencyBank to construct a larger dataset. Since both datasets share a consistent labeling system, the integration was straightforward,enabling the construction of a larger and more diverse training set.

The UCLASS dataset was used for a final round of independent testing, serving as a real-world check to ensure the model's ability to generalize to unseen data rather than memorize training patterns.

3.1 SEP-28K Dataset

In 2021, Apple introduced SEP-28K, built from 28,177 annotated clips pulled from the Stuttered Events Podcasts series. It breaks

speech down into granular categories like prolongations, repetitions, blocks, and interjections, and accounts for fluent speech and non-speech events such as pauses or low-quality audio [11].

3.2 FluencyBank Dataset

The FluencyBank dataset is a collaborative effort between Nan Bernstein Ratner (University of Maryland) and Brian MacWhinney (Carnegie Mellon University), originally designed to track how speech fluency evolves over time [12]. Each clip was carefully categorized into five core types: prolongations, blocks, sound repetitions, word repetitions, and interjections.

3.3 UCLASS Dataset

Created by Peter Howell and his team, UCLASS is one of the most established clinical archives in the field, featuring recordings from speakers—primarily children and teenagers—recorded in clinical settings [13].A processed version was used where the audio is sliced into 3-second intervals, the same format used by the other datasets, maintaining a consistent evaluation standard across all experiments.

4. DATA PREPROCESSING

While the SEP-28K dataset was originally supposed to contain 28,177 recordings, some files ended up missing due to technical issues, so the actual number of usable records was dropped to 21,855. To make up for this shortfall,the SEP-28K dataset were merged with the FluencyBank dataset, which helps it capture the subtle details that really matter when it comes to accurately classifying complex speech traits.

4.1 Merging Datasets

The annotation approach in this study combined methods from both the FluencyBank and SEP-28K datasets. The same labeling standards used in SEP-28K were followed. FluencyBank added another 4,144 three-second clips, and these clips were annotated according to the same rules used in SEP-28K. The merged dataset consisted of 32,000 clips. table Table 1 gives a preview of what the data looked like before preprocessing—specifically, how the annotations were recorded. Each clip was reviewed by three different people, and the labels (shown as 0, 1, 2, or 3) indicate how many of those reviewers assigned a particular annotation to that clip.

Name	Prolongation	Block	SoundRep	WordRep	Interjection
HeStutters_5_121	3	2	1	0	1
StutterTalk_42_8	1	0	3	0	1
FluencyBank_107_3	1	0	0	0	3

Table 1. : Samples of merged records and annotations after merging the datasets.

4.2 Label Encoding

To ensure that the model learned from reliable examples,a voting-based labeling approach was used to handle the annotations from the three reviewers. Each audio clip was originally scored from 0 to 3 across five types of disfluencies where the number reflected how many reviewers spotted that particular issue.A "majority-rule" approach was used.where a disfluency was marked as present only if at least two of the three reviewers identified it. 0 means there's no disorder, and 2 and 3 became 1 present. This careful approach produced a cleaner, more reliable dataset, following common best practices for combining multiple annotations. By focusing on agreement rather than individual judgment.

Name	Prolongation	Block	SoundRep	WordRep	Interjection
HeStutters_5_121	1	1	0	0	0
StutterTalk_42_8	0	0	1	0	0
FluencyBank_107_3	0	0	0	0	1

Table 2. : Sample of Labels After Encoding.

4.3 Upsampling

Both the SEP-28K and FluencyBank datasets suffer from class imbalance. Figure 1 illustrates the distribution of disfluency types in the merged dataset, highlighting this uneven representation.
After encoding, each disfluency type was represented using a binary label (0 or 1), indicating the absence or presence of the disorder in a clip. The distribution of these labels shows that negative samples (0) dominate most categories, particularly for disfluencies such as Block. This imbalance makes it difficult for models to learn patterns.
Similar challenges have been reported in previous work using the SEP-28K dataset, where models struggled to detect underrepresented disfluency types.

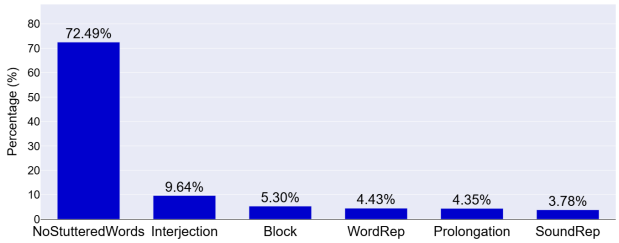

Fig. 1: Distribution of label annotations in the dataset before upsampling.

To address the class imbalance present in the dataset, a two-stage resampling strategy was utilized. First, all fluent speech samples were identified as those in which none of the five disfluency labels were present. These samples were grouped into a single non-stuttered subset. To ensure Enough representation during model training, this subset was upsampled using random sampling with replacement until it reached 20,000 samples.
In the second stage, the dataset was balanced across the different combinations of stuttering types. Since each audio segment can contain multiple disfluency types at the same time, all possible binary combinations of the five labels were considered. For each observed combination in the dataset, samples were gathered and upsampled using random sampling with replacement to reach a target size of 1,000 samples per combination. Combinations that were not available in the dataset were neglected.
Following the implementation of this balancing technique, the ultimate dataset comprised 48,000 samples, encompassing 28,000 instances of stuttered speech and 20,000 instances of non-stuttered speech, thereby mitigating bias during the model's training phase and enhancing the model's capacity to learn all varieties of stuttering behaviors with equal efficacy.

The data illustrated in figure2 show how labels were distributed both before and after resampling the existing data.

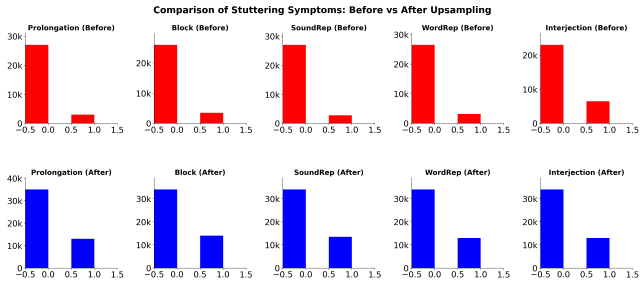

Fig. 2: Impact of Upsampling on Data Distribution.

5. FEATURE EXTRACTION

This work focuses on features that capture both the short-term spectral characteristics of speech and the dynamic changes that often signal disfluencies such as repetitions, prolongations, or blocks. The core feature used is MFCCs, which provide a compact representation of the spectral envelope of speech that closely models human auditory perception [14]. A total of 13 coefficients were extracted. For each 3-second audio clip (resampled to 16 kHz), MFCCs were calculated using a window size of 2048 samples (≈128 ms) and a hop length of 512 samples (≈32 ms), resulting in a time sequence of 94 frames per clip.

Also, first-order deltas (Δ) and second-order delta-deltas (ΔΔ) of the MFCCs were added to track fast transitions and dynamic behavior characteristic of stuttering. Delta features are the rate of change of the cepstral coefficients between frames and delta-deltas capture the acceleration (second derivative), giving information about how fast these changes vary over time [15].

Before feeding the features into the model, global standardization was performed on the entire training set. Each of the 39 feature dimensions was scaled to have zero mean and unit variance. This step ensures that no single coefficient dominates the learning process due to differences in magnitudes and also helps the model converge faster.

6. CLASSIFICATION MODEL

In this study, two hybrid deep learning architectures were utilized in this research: a combination of CNN and BiLSTM, and a second architecture that additionally integrates an attention mechanism. The CNN convolution operation is shown in the following equation [2]:

$$Z_{i,j}^{(k)} = \sum_{m=0}^{K-1} \sum_{n=0}^{C-1} W_{m,n}^{(k)} \cdot X_{i+m,\, j+n} + b^{(k)} \tag{1}$$

Where:

—$W^{(k)}$ = convolution kernel of filter k

—K = kernel size

—C = number of channels

—$b^{(k)}$ = bias

—$Z^{(k)}$ = output feature map

Each LSTM unit contains a cell state, which holds information from previous units, allowing the network to learn temporal relationships. This cell state is part of the LSTM memory unit, where several gates control which information from the inputs, previous cell state, and hidden state will be used to generate the current cell and hidden states. Namely, the forget gate f_t and input gate i_t are utilized to determine what information should be retained within the current state C_t [16]. This is shown in the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

The attention weights and context vector are computed as follows:

$$e_t = v^T \tanh(W_h h_t + b_h) \tag{4}$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} \exp(e_k)} \tag{5}$$

$$c = \sum_{t=1}^{T} \alpha_t h_t \tag{6}$$

where h_t is the hidden state at time step t, W_h and b_h are the weight matrix and bias, v is the attention weight vector, α_t represents the normalized attention weight, T is the sequence length, and c is the resulting context vector [17].

In both cases, the characteristics of the previous layers were utilized for their respective advantages: CNN layers capture the underlying spatial characteristics in the audio, while BiLSTM layers process the sequence in both forward and backward directions. The attention mechanism was also used to focus on important temporal features within the input sequence.

The CNN + BiLSTM model first passes the input feature set through CNN layers to detect and extract localized audio features. The features extracted are then fed into the BiLSTM layers that analyze the features in forward and backward directions based on the sequence of features. The bidirectional processing of features gives the model context on what is happening at a particular time point and how it relates to past and future events. The temporal context necessary for identifying stuttering patterns like prolongations and repetitions, which typically occur across several frames [16], is thus maintained.

The second model, CNN + BiLSTM + AM, builds upon the first architecture by adding a Multi-Head Attention layer after the BiLSTM layers. The attention mechanism is used to attend significant temporal regions, which helps the model to learn features needed for stutter classification. The attention layer helps the model to attend to important segments along with reducing the influence of unrelated temporal regions. This improves the model accuracy and sensitivity to detect subtle or infrequent disfluency events.

Subsequently, the implementation details, training configurations, and experimental results for each of the proposed architectures are discussed along with the reasoning behind the adopted design choice.

6.1 CNN + BiLSTM Model

The primary architecture is a hybrid CNN and BiLSTM model that detects stuttering by capturing both short-term acoustic features and long-range temporal dependencies.

The input is a reshaped feature matrix of shape (*batch_size*, *time_steps*, 1). The architecture consists of three stacked Conv1D layers with filter counts increasing from 64 to 128 and 256, a kernel size of 3, ReLU activation, and same padding. Each convolutional layer is followed by max-pooling to reduce temporal resolution:

$$P_i = \max(A_i, A_{i+1}, \ldots, A_{i+p-1}) \tag{7}$$

Additionally, dropout layers (rate 0.2) are applied after each convolutional block to reduce overfitting. The dropout operation is expressed as:

$$\tilde{h}_i = h_i \cdot r_i \tag{8}$$

$$r_i \sim \text{Bernoulli}(1-p) \tag{9}$$

where h_i is the original activation, $\tilde{h}_i$ is the activation after dropout, r_i is a binary mask, and $p = 0.2$.

The CNN output is passed to a two-layer BiLSTM, which processes the sequence in both forward and backward directions. This design is important for stuttering classification, as disfluent events such as prolongations and repetitions often span multiple frames and depend on surrounding context.

Following the BiLSTM, a dropout layer (0.2) is applied before a fully connected layer (128 units, ReLU). Another dropout layer is applied prior to the classification head, which produces the final predictions.

Each stuttering type has its own classification head, consisting of a single-unit dense layer with sigmoid activation, producing a binary output per label. The backbone (CNN + BiLSTM) is shared across all labels, while only the classification head is label-specific, allowing efficient learning of shared speech representations while adapting to each disfluency type.

The model is compiled using the Adam optimizer with binary cross-entropy loss, and is trained independently for each label.

6.2 CNN + BiLSTM with Self-Attention Model

Building on the Convolutional BiLSTM model described in the previous section, a second architecture was developed by incorporating a multi-head attention mechanism.

A simple self-attention layer was added immediately after the BiLSTM. The AM functions as an adaptive weighting system: it computes relevance scores between all time steps and learns to assign higher importance to those frames most indicative of disfluency. In practice, this allows the model to dynamically focus on the critical short segments — such as the tiny pauses or repeated bursts in sound repetition — while down-weighting irrelevant fluent regions. By focusing computational emphasis on the most informative parts of the utterance, attention reduces the influence of background noise and improves sensitivity to sparse or subtle events across all disfluency types. The rest of the architecture follows the same backbone as the first model: three Conv1D layers (64 $\rightarrow$ 128 $\rightarrow$ 256 filters, kernel size 3, ReLU, same padding, max-pooling after each, dropout 0.3), followed by the BiLSTM (64 units), then attention, flattening, dropout (0.3), a dense layer (128 units, ReLU), and another dropout (0.3). For each label, a single-unit sigmoid head is added for binary classification. Training was performed individually per label using Adam optimizer, binary cross-entropy loss.

6.3 Model Evaluation

Both the CNN + BiLSTM and CNN + BiLSTM + Attention models were trained for 100 epochs using a learning rate of 0.001 and a batch size of 64. All experiments were conducted on a system equipped with an NVIDIA GTX 1660 GPU. The models were evaluated on the 10% test set using four standard classification metrics: accuracy, precision, recall, and F1-score.

Accuracy represents the overall proportion of correct predictions (both fluent and disfluent) made by the model and is calculated as the ratio of correctly classified records to the total number of records [18]. *Precision* quantifies the fraction of instances classified as positive (stuttering) that are actually positive, indicating the model's ability to reduce false positives [19]. *Recall* (sensitivity) measures the proportion of actual positive (stuttering) samples that are correctly identified by the model [19]. In stuttering detection, high recall is important because missing true disfluencies (false negatives) may delay intervention and support. The *F1-score*, which is the harmonic mean of precision and recall, provides a balanced measure that accounts for both false positives and false negatives [19]. Because stuttering datasets often suffer from class imbalance, the F1-score is particularly useful for evaluating performance on the minority class.

7. RESULTS AND DISCUSSION

The CNN + BiLSTM model achieved an average accuracy of 97.18% and an average F1-score of 98.06% across all labels, reflecting strong overall performance with some variation across disfluency types. The CNN + BiLSTM + Attention model improved these figures to an average accuracy of 97.48% and an average F1-score of 98.18%, demonstrating the benefit of the attention mechanism in refining focus on disfluent segments.

Despite these gains,an overfitting was observed on the prolongation class in the attention-enhanced model. This likely occurred because prolongations are longer, sustained events with relatively consistent spectral characteristics, making them easier for the model to memorize during training. On the independent set of 49 unseen audio files, the CNN + BiLSTM model showed lower sensitivity to sound repetition, missing several instances due to its limited ability to isolate very short inter-repetition gaps. The CNN + BiLSTM + Attention model detected sound repetitions more reliably but exhibited reduced precision on prolongation events, consistent with the overfitting tendency. Of the 49 audios, the attention model flagged prolongation in 34 cases while original labels indicated only 6 prolongation instances.

Table 3. : Model 1 — CNN+BiLSTM: Precision and Recall Values.

Disorder	Precision		Recall		F1 Score	Test Acc.
	[0]	[1]	[0]	[1]		
Prolongation	0.9745	0.9698	0.9896	0.9296	98.3%	97.4%
Block	0.9664	0.9749	0.9906	0.9139	98.0%	97.1%
Sound Rep.	0.9780	0.9784	0.9922	0.9407	98.6%	98.0%
Word Rep.	0.9669	0.9877	0.9956	0.9115	98.3%	97.5%
Interjection	0.9475	0.9752	0.9907	0.8687	97.1%	95.9%

Table 4. : Model 2 — CNN+BiLSTM+Attention: Precision and Recall Values.

Disorder	Precision		Recall		F1 Score	Test Acc.
	[0]	[1]	[0]	[1]		
Prolongation	0.9759	0.9763	0.9917	0.9344	98.1%	97.0%
Block	0.9680	0.9750	0.9906	0.9183	98.3%	97.4%
Sound Rep.	0.9806	0.9816	0.9954	0.9477	98.9%	98.2%
Word Rep.	0.9732	0.9863	0.9950	0.9289	98.3%	97.5%
Interjection	0.9611	0.9724	0.9893	0.9041	97.3%	97.3%

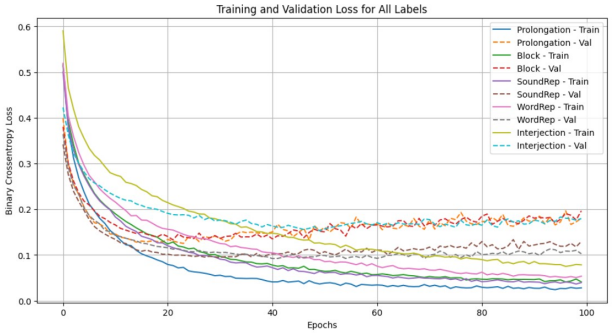

Fig. 3: Training and validation loss of the CNN–BiLSTM model.

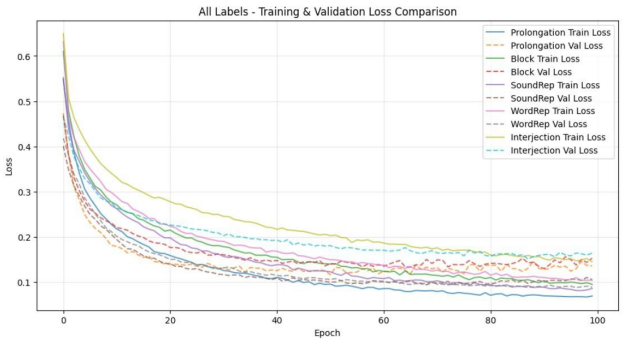

Fig. 4: Training and validation loss of the CNN–BiLSTM–Attention model.

As shown in Figures 3 and 4, the attention model demonstrated lower validation loss and stable convergence during training, but generalized poorly to real-world data—particularly for Prolongation. This suggests overfitting to validation set patterns rather than learning robust representations. In contrast, the non-attention model, despite higher and more variable validation loss, handled Prolongation more effectively in real-world testing, highlighting an important practical limitation of the attention-enhanced approach.

In addition, several other architectures were implemented and evaluated, including CNN, ConvLSTM, and Audio Spectrogram Transformers (AST).

Table 5. : Test Accuracy of Other Deep Learning and Transformer Models.

Model	Test Accuracy
CNN	75.04%
ConvLSTM	89.14%
AST	63.00%

The ConvLSTM model significantly outperformed CNN and AST in identifying sequential and temporal dependencies of disfluencies. The CNN model achieved an accuracy of 75.04%. The Audio Spectrogram Transformer (AST) showed the least success at 63%, likely because transformer-based architectures require larger and more diverse datasets and pre-training to generalize effectively to stuttering-specific patterns.

A comparison of the two models against the benchmark model [10] is presented in Table 6. the results show improvement following dataset merging, upsampling, and the use of MFCC with delta and delta-delta features.

Table 6. : Performance Comparison Across Models and Stuttering Types.

Model	Disorder	Accuracy	F1 Score
Benchmark [10] MFCC + CNN	Prolongation	95.8%	95.8%
	Block	95.7%	95.6%
	Sound Rep.	97.6%	97.6%
	Word Rep.	98.7%	98.7%
	Interjection	97.2%	97.2%
Model CNN-BiLSTM-AM	Prolongation	97.4%	98.3%
	Block	97.4%	98.3%
	Sound Rep.	98.2%	98.9%
	Word Rep.	97.5%	98.3%
	Interjection	97.3%	97.3%

To ensure a fair evaluation, both models were tested on an independent subset of 49 raw audio clips from SEP-28K that had not been upsampled or preprocessed for training. The benchmark model used a combination-based upsampling strategy producing a highly imbalanced training set biased toward stuttering detection. When tested on the 49 raw audios (only 22 actually containing stuttering), the benchmark flagged nearly all samples as disfluent, producing a high number of false positives. the models, trained on a properly balanced dataset, correctly identified stuttering in 20 out of the 22 true positive cases.

Table 7. : Exact Label Set Accuracy.

Model	Label Set Accuracy
Benchmark [10] MFCC + CNN	44.89%
Model (CNN-BiLSTM-Attention)	90.9%

Figure 5 shows a grouped bar chart comparing the F1-scores obtained by the proposed models against the benchmark method [10] for five different disfluency classes: Prolongation, Block, Sound Repetition, Word Repetition, and Interjection. This visualization enables a fine-grained assessment of model performance across different types of speech disfluencies. By presenting the results the figure shows the relative effectiveness of each model, making it easier to notice performance improvements introduced by the proposed CNN–BiLSTM and CNN–BiLSTM with attention mechanisms over the baseline approach.

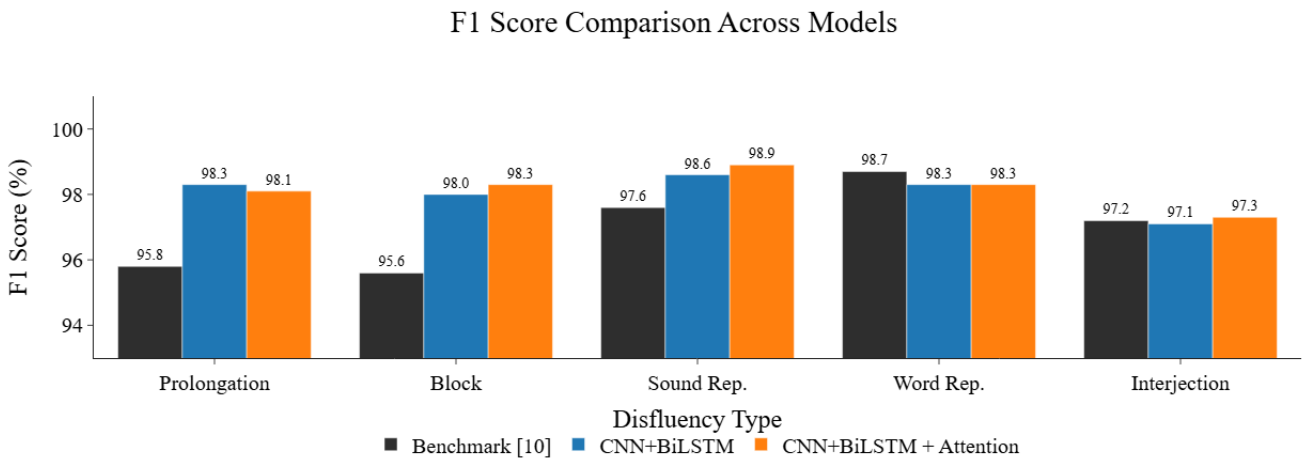

Fig. 5: Comparison of F1-scores across disfluency classes for the benchmark model and the proposed CNN–BiLSTM and CNN–BiLSTM with attention models.

8. CONCLUSION

Accurate identification and categorization of speech disfluency types are crucial to understanding and evaluating the characteristics and severity of stuttering. This research introduced and assessed an automated system for recognizing and classifying disfluencies using hybrid deep learning architectures. If further refined and validated within clinical settings, these systems could prove to be beneficial aids phoniatricians in the evaluation and planning of interventions for fluency disorders.

To address class imbalance, this study employed upsampling techniques combined with a hybrid deep learning model integrating CNN, BiLSTM, and attention mechanism components, achieving state-of-the-art performance on SEP-28K and FluencyBank benchmarks.

9. FUTURE WORK

Future work will focus on extending the proposed system by implementing the Stuttering Severity Instrument, Third Edition (SSI-3), a standardized clinical assessment tool used by phoniatricians to measure stuttering severity. By developing and integrating this model,aiming to create a complete automated system that derives an SSI-3 severity score directly from raw speech recordings, eliminating the need for manual annotation by a clinician. Such a system would represent a significant step toward scalable, objective, and real-time stuttering severity monitoring, with practical value in both clinical and remote-care settings.

10. REFERENCES

[1] N. Vasylieva, T. Marieieva, L. Zahorodnia, V. Melikhova, Y. Taniavska, and O. Dzhus, "Examining stuttering in preschool children from the perspective of speech therapy and neurology," *Revista Romaneasca pentru Educatie Multidimensionala*, vol. 17, no. 2, pp. 712–731, 2025.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[3] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[4] M. Wiśniewski, W. Kuniszyk-Jóźkowiak, E. Smołka, and W. Suszyński, "Improved approach to automatic detection of speech disorders based on the Hidden Markov Models approach," *Journal of Medical Informatics & Technologies*, vol. 15, pp. 145–152, 2010.

[5] T. Kourkounakis, A. Hajavi, and A. Etemad, "Detecting multiple speech disfluencies using a deep residual network with bidirectional long short-term memory," in *Proc. ICASSP 2020 - IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 6089–6093.

[6] S. A. Sheikh, Md. Sahidullah, F. Hirsch, and S. Ouni, "StutterNet: stuttering detection using time delay neural network," in *Proc. 2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 426–430.

[7] T. Kourkounakis, A. Hajavi, and A. Etemad, "FluentNet: end-to-end detection of stuttered speech disfluencies with deep learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2986–2999, 2021.

[8] M. Jouaiti and K. Dautenhahn, "Dysfluency classification in stuttered speech using deep learning for real-time applications," in *Proc. ICASSP 2022 - IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 6482–6486.

[9] K. Basak, N. Mishra, and H. T. Chang, "TranStutter: a convolution-free transformer-based deep learning method to classify stuttered speech using 2D mel-spectrogram visualization and attention-based feature representation," *Sensors*, vol. 23, no. 19, p. 8033, 2023.

[10] N. Alhakbani, R. Alnashwan, A. Al-Nafjan, and A. Almudhi, "Automated stuttering detection using deep learning techniques," *Journal of Clinical Medicine*, vol. 14, no. 10, p. 3552, 2025.

[11] C. Lea, V. Mitra, A. Joshi, S. Kajarekar, and J. P. Bigham, "SEP-28K: a dataset for stuttering event detection from podcasts with people who stutter," in *Proc. ICASSP 2021 - IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 6798–6802.

[12] N. B. Ratner and B. MacWhinney, "Fluency Bank: a new resource for fluency research and practice," *Journal of Fluency Disorders*, vol. 56, pp. 69–80, 2018.

[13] P. Howell, S. Davis, and J. Bartrip, "The University College London archive of stuttered speech (UCLASS)," *Journal of Speech, Language, and Hearing Research*, vol. 52, no. 2, pp. 556–569, 2009.

[14] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[15] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, "Fundamentals of speech recognition," in *Robust Automatic Speech Recognition*, pp. 9–40, 2016.

[16] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5–6, pp. 602–610, 2005.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[18] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[19] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.