Deep Learning Bidirectional LSTM based Detection of Prolongation and Repetition in Stuttered Speech using Weighted MFCC

Sakshi Gupta[1], Ravi S. Shukla[2], Rajesh K. Shukla[3], Rajesh Verma[4]
Department of Computer Science and Engineering, Invertis University, Bareilly, Uttar Pradesh, India[1, 3]
Department of Computer Science, Saudi Electronic University, Kingdom of Saudi Arabia[2]
Department of Electrical Engineering, King Khalid University, Kingdom of Saudi Arabia[4]

***Abstract*—Stuttering is a neuro-development disorder during which normal speech flow is not fluent. Traditionally Speech-Language Pathologists used to assess the extent of stuttering by counting the speech disfluencies manually. Such sorts of stuttering assessments are arbitrary, incoherent, lengthy, and error-prone. The present study focused on objective assessment to speech disfluencies such as prolongation and syllable, word, and phrase repetition. The proposed method is based on the Weighted Mel Frequency Cepstral Coefficient feature extraction algorithm and deep-learning Bidirectional Long-Short term Memory neural network for classification of stuttered events. The work has utilized the UCLASS stuttering dataset for analysis. The speech samples of the database are initially pre-processed, manually segmented, and labeled as a type of disfluency. The labeled speech samples are parameterized to Weighted MFCC feature vectors. Then extracted features are inputted to the Bidirectional-LSTM network for training and testing of the model. The effect of different hyper-parameters on classification results is examined. The test results show that the proposed method reaches the best accuracy of 96.67%, as compared to the LSTM model. The promising recognition accuracy of 97.33%, 98.67%, 97.5%, 97.19%, and 97.67% was achieved for the detection of fluent, prolongation, syllable, word, and phrase repetition, respectively.**

Keyword—Speech; stuttering; deep learning; WMFCC; Bi-LSTM

I. Introduction

For communication between human beings, speech proves to be the most habitually and widely used verbal means to precise feelings, ideas, and thought. Not all human beings are blessed with normal means of speech. The potency of speech in delivering data during communication depends on fluency. Fluency is defined by normal speech flow, which connects different phonemes to make a message [1]. Speech is fluent if continuity among semantic units, rhythm, speed, and energy applied for flow is normal. Any kind of disruption in fluency is known as dysfluency. Stuttering is a complex type of dysfluency. In stuttering, there is a disturbance in continuity and rhythm due to pauses and blocks, the rate is much slower, and efforts are higher than normal. Researchers have categorized the factors that lead to stuttering as of three types, namely, development, neurogenic, and psychogenic.

People who stutter (PWS) may have three sorts of disfluencies: repetition of a sound, syllable, word or phrase, sound prolongation during which a sound is sustained for a markedly more extended period that may be traditional and silent blocks at starting of vocalization or word or within the middle of a word. Johnson [2] introduced this classification for the first time. It has been used by clinicians and researchers ever since.

Even though stuttering may not be considered as a disability by many people, it incites a speech constraint. People who stutter loses not only their confidence but also generate a negative attitude towards their communication skills. Furthermore, it ruins their self-confidence, relationship with others, employment opportunities, and opinions of others about them [3]. Stuttering influence individuals of all ages, culture, and races irrespective of their intelligence and financial status. Many pieces of research have stated that stuttering affects approximately 1% of the world population and is more common in males as compared to females [4]. Therefore, this area is mainly a knowledge base field of analysis for different domains like speech pathology, psychology, speech physiology, acoustics, and signal analysis.

Stuttering is one of the intense issues found in speech pathology. Speech-Language Pathologists (SLP) diagnoses the individual who stutters and measures the fluency to gauge the response of the stutterer throughout the treatment process. Traditionally SLPs used to assess the extent of stuttering manually. They counted and divided the frequency of stuttered events with total spoken words. Such sorts of stuttering assessments are arbitrary, incoherent, lengthy, and error-prone. Over the past two decades, SLPs gave great attention to objective assessment techniques for assessing the stuttered events, as discussed in our previous work [5].

Automatic evaluation of stuttered speech is therefore necessary, to automate the count and classification of stuttered events. The proposed work has employed Weighted Mel Frequency Cepstral Coefficients (WMFCC) feature extraction method and deep-learning-based classification method Bi-directional Long-Short Term Memory (Bi-LSTM) for the automatic assessment of four forms of disfluency prolongation and syllable, word, and phrase repetition. The efficacy of the Bi-LSTM model is assessed as compared to other classification models, based on the accuracy of the classification of stuttered events.

In this paper, the University College London Archive of Stuttered Speech (UCLASS) database is utilized for analysis. The experimental analysis in this study reveals that WMFCC and Bi-LSTM based proposed method performs more efficiently as compared to other models.

The results elucidate that the model proposed has improved performance and advantages compared with other models. This study makes two significant contributions.

- Firstly, it uses WMFCC instead of traditional MFCC for feature extraction. WMFCC includes the dynamic information of the speech samples, which increases the detection accuracy of stuttered events; and also reduces the computational overhead to the classification stage.
- Secondly, it employs Bi-LSTM rather than traditional RNN and LSTM. Bi-LSTM provides the solution for gradient disappearance in RNN, as well as overcomes the unidirectional flow of information of LSTM.

The paper is structured according to the following. Section 2 reviews the work related to automatic detection of stuttering speech disorders. Section 3 elaborates on the framework for the system proposed. It also includes brief descriptions of the database used, feature extraction, and classification techniques applied. Section 4 consists of experimental results and a comparative analysis of the classification model. Section 5 provides a conclusion.

II. Related Works

This section reviews work relating to recognition systems designed to detect or classify stuttering speech disorders; previous research has presented various methods and algorithms that have been applied to recognizing stuttering events from speech signals. Table I displays a comprehensive comparative analysis of various feature extraction and classification methods based on the dataset used, type of disfluency, and accuracy. The previous works conducted signifies the importance of feature extraction and classification methods in the stuttered events detection.

Traditional machine learning techniques are being gradually replaced by Deep learning technology. Deep learning provides a more accurate representation of objects and can automatically obtain objects features from a vast amount of data [26]. These are progressively used to further refine computers' capacities in order to understand what humans can do, including speech recognition. Deep structured learning models based on these functional attributes include convolutional neural network (CNN) [27], recurrent neural network (RNN) [28][24], and long-short term memory (LSTM) [25]. The conventional machine learning techniques for recognition employed shallow structured architectures such as hidden Markov model (HMM), Support Vector Machines (SVM), Artificial Neural Network (ANN), and linear and non-linear dynamical system [29]. These architectures are ideally suited for simple or constrained problems, since their limited capabilities can cause problems in complicated large-scale real-world problems [30]. Such real-world problems involve human speech, language recognition, and visual scenes, requiring a more profound and layered architecture to extract the complex information.

Tian Swee et al. [6] and Thiang and Wanto [9] trained Hidden Markov Model (HMM) model to classify speech samples as fluent and non-fluent. The HMM model determines the likelihood of being in a state depends on its prior state at (t-1) while disregarding all other dependence. It also requires a large number of parameters and data for building and training the model [31]. In [8] and [14], Ravikumar et al. and Hariharan et al. discussed the classification of extracted features through Support Vector Machines (SVM). However, SVM deals with only fixed-size input are not efficient for large databases as well as its computational cost is directly proportional to the number of classes to be classified. Savin et al. [19] employed an ANN for classification. ANN does not have structured methodology as well as time-consuming for large networks [32].

The deep learning technique CNN performs very well on non-sequential data while fails in interpreting temporal information. However, the RNN is good at modeling the temporal data but suffers from the problem of short-term memory caused by vanishing gradient [33][34][35]. Thus, LSTM was created as a solution to short-term memory [36]. They are capable of learning long term dependencies [37]. Based on the above considerations, this paper applies Bi-LSTM for the classification of a vast amount of speech data [38]. Bi-LSTM model processes the information in two directions and links them to obtain the output class of stuttering.

TABLE I. COMPREHENSIVE ANALYSIS OF VARIOUS ON RESEARCH ACTIVITIES ON STUTTERING DETECTION, DESCRIBING THE FEATURES USED, CLASSIFIER EMPLOYED, NUMBER OF SUBJECTS, TYPE OF CLASSIFICATION AND EXPERIMENTAL RESULTS

Year	Feature Used	Classifier Used	Dataset Used	Type of Classification	Result
2007 [6]	MFCC	HMM	Malay language-based 20 normal and 15 artificial speech samples	Repetition, Prolongation, and Blocks	Normal data- 96%, Artificial Stutter Speech data- 90%
2009 [7]	Kohonen Network	Multi-layer Perceptron and RBF	59 800ms samples of 8 stuttering Polish speakers	Repetition and Prolongation	MLP- 92% RBF- 91%
2009 [8]	MFCC	SVM	12 training and 3 testing samples of 15 adults who stutter	Syllable Repetition	94.35%
2010 [9]	LPC	HMM	5, 10, 15, 20 samples per command and 40-50 observation symbols of HMM	-	5 samples-93.75%, 10- 98.75%, 15- 100% and 20-97.5%
2010 [10]	MFCC	KNN and LDA	10 samples of 8 males and 2 females (11 to 20 years) from UCLASS	Repetition and Prolongation	90%
2010 [11]	LPCC	KNN and LDA	10 samples of 8 males and 2 females (11 to 20 years) from UCLASS	Repetition and Prolongation	88.05%
2011 [12]	12, 13, 26 and 39 Dimensional MFCC	DTW	8 training and 2 testing samples	Repetition	12 D- 80.69%, 13 D- 68.4%, 26 D- 84.01%, 39 D- 84.58%,
2012 [13]	MFCC and LPCC	KNN and LDA	UCLASS database	Repetition and Prolongation	MFCC- 92.55%, LPCC- 94.51%
2012 [14]	Spectral Entropy using Bark, Mel and Erb Scale	SVM	UCLASS database	Repetition and Prolongation	Average accuracy- 96%. Beat result of 96.84% in Erb scale
2013 [15]	MFCC, PLP, and LPC	KNN, LDA, and SVM	UCLASS database	Repetition and Prolongation	Best average classification accuracy is given by SVM using the WLPCC, PLP, and MFCC features- 95%
2013 [16]	SOM	Hierarchal ANN, MLP	153 recordings of 19 PWS	Blocks, syllable repetition and syllable initial prolongation	Blocks- 96% Syllable Repetition- 84% and Prolongation-99%
2014 [17]	MFCC	SVM	UCLASS database	Repetition and Prolongation	97.6%
2015 [18]	MFCC	KNN	80 speech samples for training and 20 for testing	Repetition with 0db to 10db babble noise	60-95% depending on the sound used
2016 [19]	MFCC, Formant, Pitch, ZCR, and Energy	ANN	78 recordings of 4 PWS (25-40 years)	Repetition and Prolongation	88.29%
2016 [20]	MFCC, Formant and Shimmer	DTW	50 repetition events	Repetition	94%
2016 [21]	MACV	Thresholding	5 Stuttering person speech samples from UCLASS database	Repetition and Prolongation	73.29%
2016 [22]	MFCC and PLP	Cross-correlation, Euclidean distance using Morphological Image Processing	UCLASS database	Prolongation, word repetition, and phrase repetition	Prolongation- 99.84%, Word repetition- 98.07% and Phrase repetition- 99.87%
2017 [23]	MFCC	I-Vector	1380 segments of 18 PWS from UCLASS. 80% used for training and 20% for testing	Repetition, Prolongation, and Repetition-Prolongation	Normal- 52.43%, Repetition- 69.56%, Prolongation- 40%, Rep-Pro- 50%
2020 [24]	MFCC	Gated Recurrent CNN	UCLASS database	Prolongation and Repetition	Prolongation- 95% Repetition- 92%
2020 [25]	MFCC	LSTM	UCLASS database	Prolongation, Blocks, and Repetition	4% and 6% higher than ANN and SVM

III. Construction of Model

The proposed work has employed the WMFCC feature extraction method and deep-learning-based classification method Bi-directional Long-Short Term Memory (Bi-LSTM) for the automatic assessment of four forms of disfluency prolongation and syllable, word, and phrase repetition. The process for detection of repetition and prolongation in stuttered speech is split into five stages: signal pre-processing, disfluent speech sample segmentation and labeling, labeled sample splitting into training, validation and test sets, feature extraction and classification using network training and model (Fig. 1). The University College London Archive of Stuttered Speech (UCLASS) database is utilized for analysis [39]. The study evaluates the efficacy of Bi-LSTM model, based on the accuracy of the classification of stuttered events.

A. Signal Pre-Processing

A signal is pre-processed by removing the silence regions [40][41]. There is no excitation in the vocal tract during the silence region, hence no speech production. Thus, pre-processing reduces not only the amount of processing but also enhances the overall efficiency and accuracy of the system proposed. The combination of two widely known approaches, namely Short Time Energy (STE) and Zeros Crossing Rate (ZCR) (Fig. 2), has been used in this work [42][43]. It is a fast and straightforward approach and gives a better result of classifying the speech into voiced/unvoiced.

The short-term energy is the energy-related to short term region of speech [41]. The total energy of a speech frame is determined by the following (1).

$$E(n) = \sum_{m=-\infty}^{\infty} \left(s(m).w(n-m)\right)^2 \qquad (1)$$

Where w(n) represents the windowing function, and n is the shift in the number of samples. The voiced region energy is high in comparison with the unvoiced region. The silent region displays marginal energy content.

Zero-Crossing Rate specifies the number of zero crossings in a given signal [41]. The zero-crossing rate of a stationary signal is calculated by (2):

$$ZCR = \sum_{n=-\infty}^{\infty} \left| sgn\left(s(n)\right) - sgn\left(s(n-1)\right) \right| \qquad (2)$$

Where $sgn(s(n))$ is a signum function and is described as by the (3).

$$sgn\left(s(n)\right) = \begin{cases} 1\; if\; s(n) \geq 0 \\ -1\; if\; s(n) < 0 \end{cases} \qquad (3)$$

The zero-crossing rates in unvoiced sounds are comparatively high as compared to the voiced sounds. The combination of these two features overcome the issue of categorizing the speech into a voiced/unvoiced signal (Fig. 3).

B. Disfluent Speech Sample Segmentation and Labeling

The disfluent speech signals are obtained from the University College London Archive of Stuttered Speech (UCLASS) [39]. It is released in version 1 and version 2, consisting of three types of recording: monologues, reading, and spontaneous conversation. Version 1 has 138 "monologue" recordings contributed by 81 speakers. The database used in this work refers to 20 samples of speech for experimentation [44]. It comprises two female speakers and 18 male speakers aged 7years 8 months to 17 years 9 months. The selection of speech signals aims at covering a wide variety of stuttering rate and age. The samples provided with text script are only included in the database.

This paper investigates only four forms of disfluencies, prolongation, and syllable, word, and phrase repetition. They are easily detectable in monosyllabic words. After pre-processing the selected speech samples, disfluent speech samples were marked and segmented manually by listening to the pre-processed signals. The segmented samples were labeled as five classes, namely, Fluent, Prolongation, Syllable Repetition, Word Repetition, and Phrase Repetition (Fig. 4).

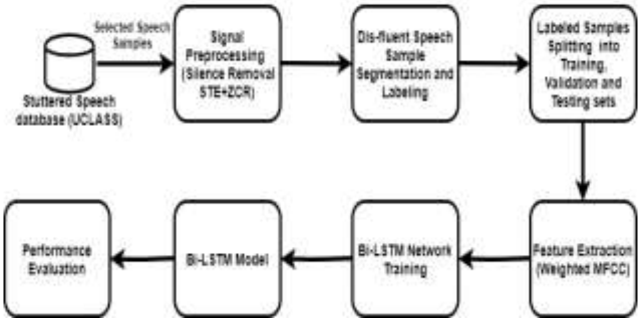
Selected Speech Samples
Stuttered Speech database (UCLASS)
Signal Preprocessing (Silence Removal STE+ZCR)
Dis-fluent Speech Sample Segmentation and Labeling
Labeled Samples Splitting into Training, Validation and Testing sets
Feature Extraction (Weighted MFCC)
Bi-LSTM Network Training
Bi-LSTM Model
Performance Evaluation

Fig. 1. Block Diagram of the Proposed Model.

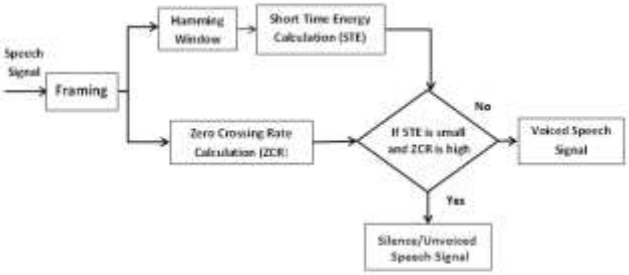
Speech Signal
Framing
Hamming Window
Short Time Energy Calculation (STE)
Zero Crossing Rate Calculation (ZCR)
If STE is small and ZCR is high
No
Voiced Speech Signal
Yes
Silence/Unvoiced Speech Signal

Fig. 2. Speech Pre-Processing by Silence Removal.

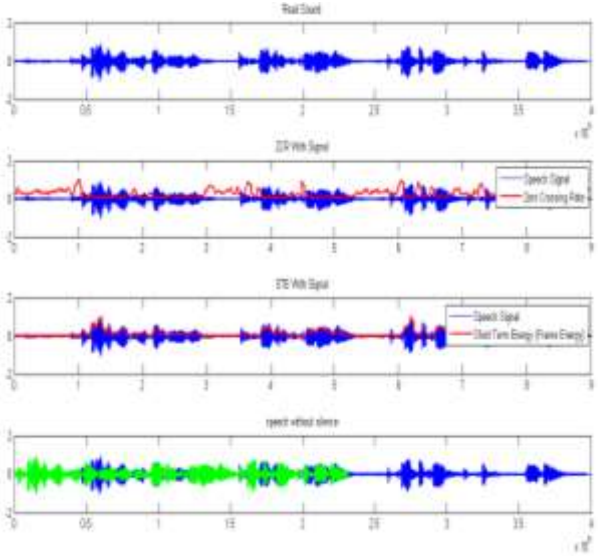

Fig. 3. Silence Removal using STE-ZCR Method.

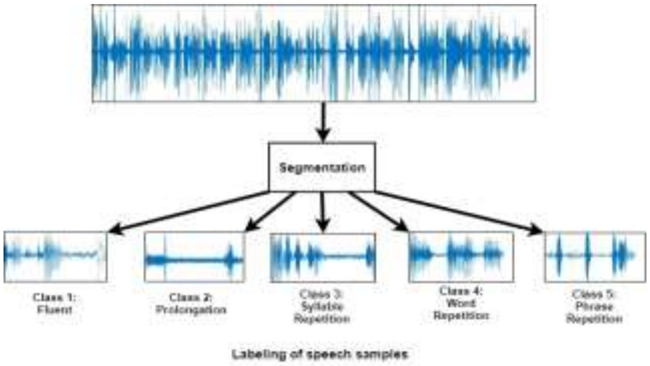
Segmentation
Class 1: Fluent
Class 2: Prolongation
Class 3: Syllable Repetition
Class 4: Word Repetition
Class 5: Phrase Repetition
Labeling of speech samples

Fig. 4. Disfluent Speech Sample Segmentation and Labeling.

C. Labeled Samples Splitting

The segmented disfluent speech samples were divided into three sets for training, validation, and testing. The training set is a subset of labeled stuttered speech samples used to train the model. The validation set evaluates the performance of the model with different hyperparameter values. It is smaller than the training set. The test set determines the final accuracy of the model and analyses the performance of different models. In this study, the datastore of disfluent speech samples is split into training, validation, and test set in the ratio of 60%, 20%, and 20%, respectively.

The process of pre-processing, segmentation, labeling, and sample splitting is described through an algorithm in Table II.

TABLE II. ALGORITHM OF SPEECH SAMPLE PRE-PROCESSING AND SEGMENTATION

```
Input: Selected speech samples of UCLASS database $T_{unpreprocess}$
Output: Pre-processed and labeled speech samples dataset $T_{train}$, $T_{valid}$, and $T_{test}$

1.  Loading the selected speech samples of UCLASS database $T_{unpreprocess}$.
2.  For each sample $S \in T_{unpreprocess}$
3.      Divide the samples into frames of 30msec.
4.      For each frame $f \in$ sample $S$
5.          Calculate STE and ZCR of frame $f$.
6.          If $(STE \geq 0.01)$ and $(ZCR \leq 0.2)$
                Append the frame $f$ to new pre-processed sample $\in S_{preprocess}$
            Else discard the frame.
7.      End for.
8.  End for.
9.  The set of pre-processed samples $T_{preprocess}$ is derived.
10. For each sample $S \in T_{preprocess}$
11.     Manually divide the speech sample into a set of segments $T_{unlabelled}$.
12.     For each segment $St \in T_{unlabeled}$
13.         Identify and label the segment as fluent, prolongation, syllable repetition, word repetition, or phrase repetition.
14.     End for.
15. End for.
16. The set of labeled samples $T_{labelled}$ is derived.
17. Split the set $T_{labelled}$ into training $T_{train}$, validation $T_{valid}$, and testing $T_{test}$ datasets in the ratio 60%, 20% and 20% respectively.
```

D. WMFCC Feature Extraction

The extraction of speech features is a sort of dimension reduction technique that is employed to minimize the data that is giant to be processed by an algorithm. The key objective of feature extraction is to upbraid the speech signal into the various acoustically recognizable elements and to get the feature vectors with a nominal amendment to keep the processing efficient. In our previous work [45], a comparative analysis of extensions of MFCC feature extraction techniques [46], namely Delta MFCC, Delta-delta MFCC, and Weighted MFCC [47] was conducted. Its experimental results displayed, WMFCC slightly outperforms Delta-delta MFCC and significantly outperforms Delta MFCC and MFCC in all situations of frame length, alpha values, and frame overlap percentage [45]. The proposed work has applied frequency-domain based Weighted Mel Frequency Cepstral Coefficients. WMFCC is a fusion of MFCC and its derivatives delta and delta-delta. The resultant vector contains both static as well as dynamic information of the signal. Moreover, the feature vector is of size 14; thus, incur less computational overhead to the classification stage. Table III describes the WMFCC feature extraction algorithm, and the results of the algorithm are displayed in Fig. 5.

TABLE III. ALGORITHM OF WMFCC FEATURE EXTRACTION

```
Input: Pre-processed and labeled speech samples dataset $T_{train}$, $T_{valid}$, and $T_{test}$.
Output: WMFCC feature vector of $T_{train}$, $T_{valid}$, and $T_{test}$ datasets.

1.  Load the datasets $T_{train}$, $T_{valid}$, and $T_{test}$.
2.  Initialize the parameters of the WMFCC feature extraction method.
3.  For each labeled sample $S \in (T_{train}, T_{valid}$, and $T_{test})$
4.      Pre-emphasize $S$ using $\alpha$ filter as 0.98.
5.      Divide the sample into 30msec frames with an overlapping percentage of 75%.
6.      For each frame $f \in S$
7.          Apply the Hamming window function to frame $f$.
8.          Calculate the power spectrum of the windowed signal using FFT.
9.          Calculate the Mel spectrum by passing the power spectrum through 20 Mel filters.
10.         Calculate the log-energy of each filter bank part.
11.         Calculate MFCC by applying energies to DCT.
12.         Compute Delta MFCC as:
                $\Delta c_t = \frac{\sum_{K=1}^{M}(c_{t+k} - c_{t-k})}{2\sum_{K=1}^{M} k^2}$
13.         Compute Delta-Delta MFCC as:
                $\Delta\Delta c_t = \frac{\sum_{K=1}^{M}(\Delta c_{t+k} - \Delta c_{t-k})}{2\sum_{K=1}^{M} k^2}$
14.         Compute 14-dimensional WMFCC as:
                $wc_t = c_t + p \cdot \Delta c_t + q \cdot \Delta\Delta c_t, \ q < p < 1$
```

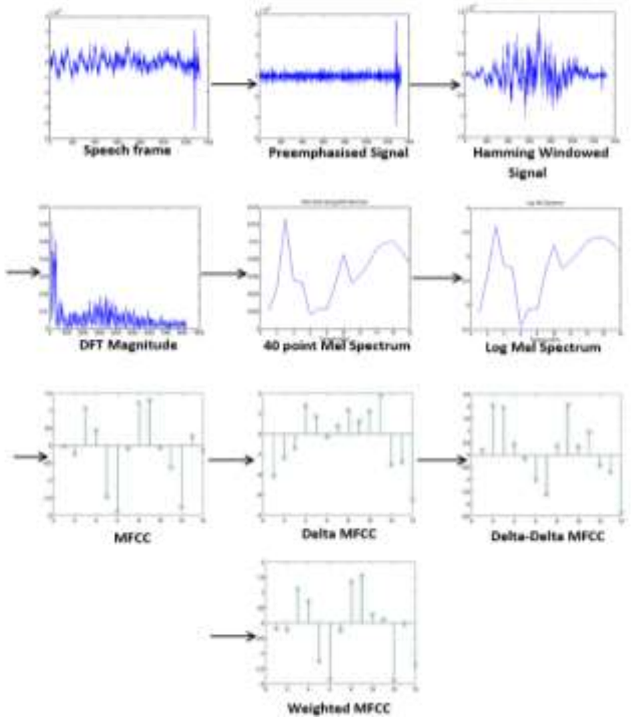

Fig. 5. WMFCC Feature Extraction Process.

E. Bi-Directional Long-Short Term Memory

Deep learning Bi-LSTM is applied for the classification of stuttered speech samples. It is composed of LSTM cells (Fig. 6). The set of features vectors discussed in the above section are set as input to the classifier. The model is trained and validated with 60% and 20% of the speech samples of the datastore, respectively. The remaining of the samples are used for testing the model.

1) Long-Short Term Memory: LSTM is a specialized Recurrent Neural Network (RNN) architecture, competent in learning long term dependencies [48]. RNN suffers from short-term memory, caused by vanishing gradient problem. To mitigate this problem, LSTM has a hidden layer known as the LSTM cell. LSTM cells are built with various gates and cell state that can regulate the flow of information. Like RNNs, at each time iteration, t, the LSTM cell has the layer input, $\mathrm{x_t}$, and the layer output, $\mathrm{h_t}$. The cell also takes the cell input state, $\tilde{\mathrm{C}}_\mathrm{t}$, the cell output state, $\mathrm{C_t}$, and the previous cell output state, $\mathrm{C_{t-1}}$. LSTM architecture has three gates, namely, forget, input, and output gate denoted as $\mathrm{f_t}$, $\mathrm{i_t}$, and $\mathrm{o_t}$, respectively.

The cell state act as the network memory, conveying valuable information across the entire sequence. The gates are specific neural networks that determine which information is permitted on the cell state. Throughout the training, the gates will learn which information is essential to retain or forget. The value of gates and cell state can be determined by using the following (4) to (7):

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{4}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6}$$

$$\tilde{C}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{7}$$

where W_f, W_i, W_o, and W_c are the weights connecting the hidden layer input to all the gates and input cell state. The U_f, U_i, U_o, and U_c are the weight matrices mapping previous cell output state to all the gates and input cell state. The b_f, b_i, b_o, and b_c are bias vectors. The σ and $tanh$ are the sigmoid and tanh activation function, respectively. The cell output state, C_t, and the layer output, h_t, at each time iteration t, can be calculated as in (8)-(9):

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{8}$$

$$h_t = o_t * \tanh(C_t) \tag{9}$$

The result of the LSTM layer should be a vector of all the outputs, represented as $Y_T = [h_{T-n}, \dots, h_{T-1}]$.

2) Bidirectional LSTM: The Bi-LSTM are originated from bidirectional RNN [50]. It processes sequential data with two different hidden layers, in both forward and backward directions, and links them to the same output layer. Across certain areas, bidirectional networks are considerably stronger than unidirectional ones, such as speech recognition [51].

Fig. 7 represents an unfolded Bi-LSTM layer structure containing a forward and a backward LSTM layer [52]. The output sequence of the forward layer, $\vec{h}$, is determined iteratively using inputs in a definite sequence, while the output sequence of backward layer, $\overleftarrow{h}$, is determined using the reversed input. The forward and backward layer outputs are computed using standard LSTM by (4) - (9). The Bi-LSTM layer produces an output vector, Y_T , which defines each element by the following Equation (10).

$$y_t = \sigma\left(\vec{h}_t, \overleftarrow{h}_t\right) \tag{10}$$

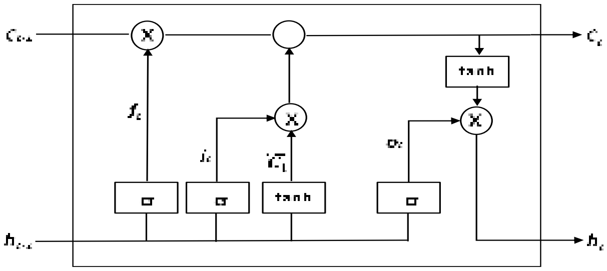

Fig. 6. LSTM Cell [49].

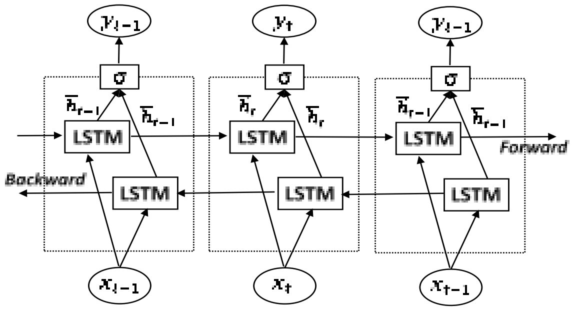

Fig. 7. Structure of an unfolded Bi-LSTM Layer [52].

where σ function combines the two output sequences. It can be a summation function, a multiplication function, a concatenating function, or an average function. Similar to the LSTM layer, a vector, $Y_T = [h_{T-n}, \dots., h_{T-1}]$, represents the final output of a Bi-LSTM layer.

F. Bi-LSTM Model Training and Testing

Although LSTM can acquire long speech sequence information but only takes one direction into consideration. It assumes that only previous frame affects the current frame. But not considers that the next frame is also related to current state. This signifies that there is a two-way relationship and the next speech frame should also be considered. Bi-LSTM provides the solution for this problem (Fig. 8).

Bi-LSTM is capable of solving the relationship between two speech frames. It also strengthens the two-way relationship between the current and next speech frame. Due to the bi-directional time structure of Bi-LSTM, it captures more structural information. Hence gives better classification accuracy as compared to one-way LSTM [53].

From Fig. 8, it can be seen that speech features vectors are obtained through the WMFCC feature extraction technique, and then the feature sequences are passed through Bi-LSTM for training and testing. The Bi-LSTM links the output of the feature extraction module to the further layers. Table IV describes the complete training and testing algorithm.

1) Sort data for padding: During training, the training feature vectors are split into mini-batches. The training data is padded so that they all have the same length. However, a large amount of padding degrades network performance. In order to prevent too much padding in the training process, the training data is sorted by sequence length.

2) Define Bi-LSTM network: Bi-LSTM network is a layered architecture shown in Fig. 8. The first layer embedding layer is also called as the sequence input layer. It takes the sorted 14-dimensional WMFCC feature vector as input. The second and third layers are the hidden forward and backward LSTM, forming the Bi-LSTM layer with 100 hidden units. Due to these two layers, the current input is related to the previous and next sequence. The input sequence reaches the model in both directions through the hidden layer. After the processing of the hidden layers, the outputs are combined to obtain the final output of the Bi-LSTM layer. The output from both the LSTM layers can be computed by the following (11):

$$h_t = \alpha h_t^f + \beta h_t^b \tag{11}$$

where h_t^f and h_t^b represents the output of forward and backward LSTM layer, when it takes sequence from x_1 and x_T as input. α and β are to control the factors of Bi-LSTM. h_t is the sum of two unidirectional LSTM elements at time t.

The output of the Bi-LSTM layer is the input to the fully connected layer of size equal to the number of classes, i.e., five. This layer links each piece of input feature information with a piece of output information for classification by the next layers.

Finally, the softmax and classification layers categorize speech frames into various disfluencies classes such as prolongation, syllable repetition, word repetition, and phrase repetition. The softmax layer applies the softmax function as an activation function that converts the real vector values into a vector with values between 0 and 1, so it can be interpreted as probabilities. The probability of classifying x into class k in the softmax regression [54] can be defined by (12).

$$P\left(y^{(i)} = k \middle| x;\, \theta\right) = \frac{exp(\theta^{(k)\mathrm{T}}x)}{\sum_{j=1}^{K} exp(\theta^{(j)\mathrm{T}}x)} \tag{12}$$

where Krepresents the number of classes and θ are the model parameters.

In the classification layer, the model receives the values from the softmax function and assigns each input to one of the classes using the cross-entropy function (13).

$$loss = -\sum_{i=1}^{N}\sum_{j=1}^{K} t_{ij} \ln y_{ij} \tag{13}$$

where N represents the number of samples, K is the number of classes, t_{ij} indicates that ith sample belongs to jth class and y_{ij} represents the value obtained from the softmax function.

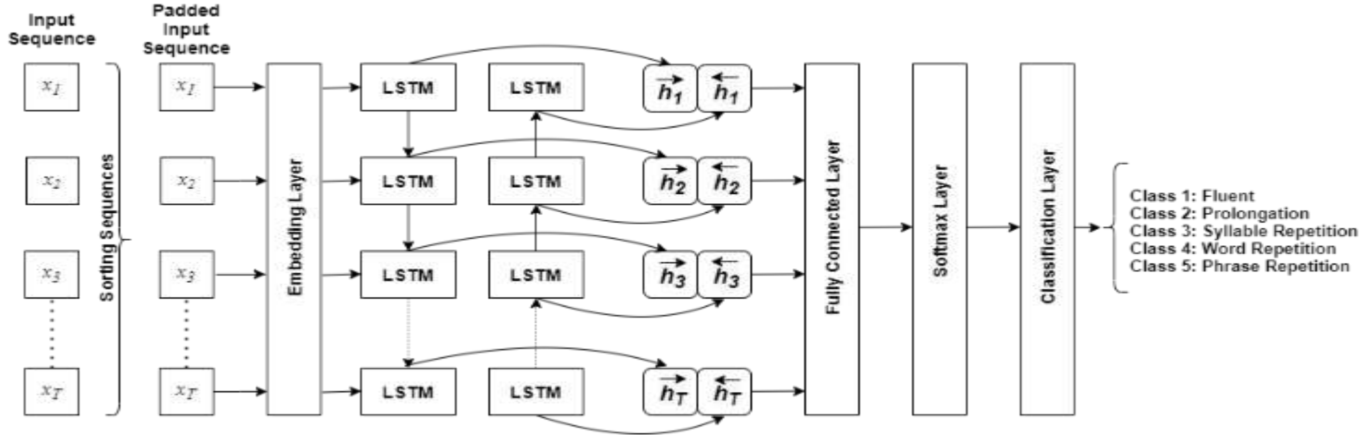

Fig. 8. Bi-LSTM Network.

TABLE IV. ALGORITHM OF BI-LSTM CLASSIFICATION

```
Input: WMFCC feature vector of $T_{train}$, $T_{valid}$, and $T_{test}$ datasets.
Output: Stuttered events classification accuracy

1.  Load the training $T_{train}$ and validation $T_{valid}$ dataset.
2.  Sort the datasets by sequence length.
3.  Build the Bi-LSTM network.
4.  Initialize the Bi-LSTM training hyper-parameters.
5.  Specify the training options.
6.  Train the Bi-LSTM network with $T_{train}$ dataset.
7.  Validate the Bi-LSTM network with $T_{valid}$ dataset.
8.  If Bi-LSTM network is not optimized
        then reinitialize the hyperparameters from step 4.
9.  Load the testing dataset $T_{test}$.
10. Classify the $T_{test}$ samples using a trained Bi-LSTM model.
11. Match the similarity between the test labels and predicted labels.
12. Evaluate the stuttered events classification accuracy of the model.
13. If classification accuracy is optimal
        then output classification accuracy
        else rebuild the model from step 3
```

3) Initializing the hyper-parameters of the network: Once the network is defined, the hyper-parameters of the network are initialized. Model hyper-parameters are properties on which the entire training process depends [55]. They are divided into two categories: Optimizer and model-specific hyper-parameters. The optimization parameters determine how the network is trained and is more related to optimization, such as the number of epochs, batch size, and learning rate. In contrast, the model-specific parameters are variables that determine the model structure, such as the number of hidden units and hidden layers. These parameters should be defined before training.

Hyper-parameter directly controls the training algorithm's behavior and thus have a significant difference in improving model performance [55]. Therefore, choosing appropriate parameters is an integral part of the optimization of the learned model. The process of selecting good hyper-parameters involves a large number of experiments, which is a time-consuming and tedious task. Most researchers rely on their experience of selecting appropriate parameters for a deep neural network.

In order to determine appropriate hyper-parameters, the classification accuracy of the validation set is used for evaluation. This work applies a diagnostic approach, in which various hyperparameters performance is investigated on both training and validation datasets. The analysis determines how a given configuration performs and how to be adjusted to obtain better performance. The hyper-parameters such as learning rate, batch size, number of epochs, and number of hidden units are taken into consideration for analysis.

4) Training and testing of the datasets: Once the Bi-LSTM model and its hyper-parameters are defined, the model is trained by using the training dataset. After the training process is over, the model is validated through the validation dataset. If the classification accuracy of the model is optimized, then its performance is tested; otherwise, the model hyper-parameters are reconfigured. The parameters such as learning rate, batch size, number of epochs, and number of hidden units are considered for reconfiguration. These parameters are tested for various ranges of values. The process of reconfiguration of hyper-parameters is repeated until the model is optimized, as represented in Fig. 9.

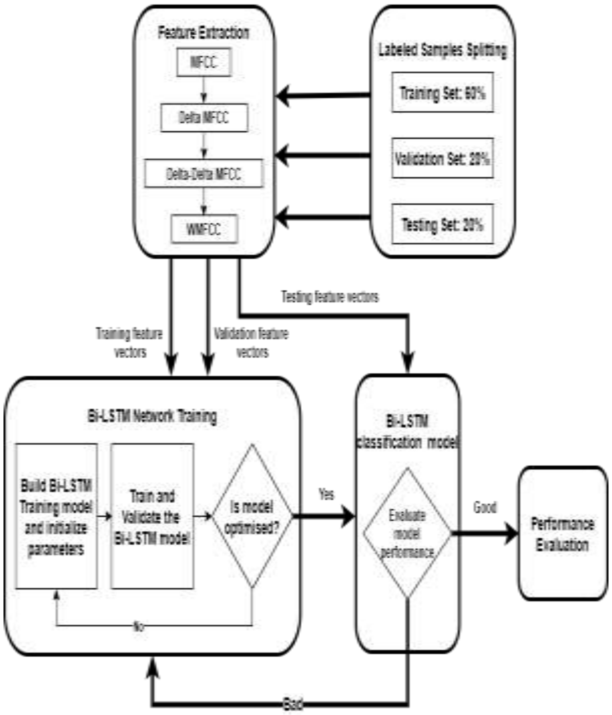

Fig. 9. Bi-LSTM Training and Testing Process.

The classification performance of the optimized model is compared with the traditional LSTM model using the testing dataset. After the testing process is over, the process of performance evaluation is carried out. If the results of the evaluation are optimal, then the process is stopped; otherwise, the complete model is redefined, and the complete training process is repeated until the model is optimized.

IV. EXPERIMENTS AND RESULTS

This section discusses the efficacy and performance of the proposed algorithm based on WMFCC feature extraction and Bi-LSTM classification for four forms of disfluencies. This study evaluates the stuttered events recognition model using the stuttered samples obtained from the UCLASS database. The dataset used in this work refers to 20 samples of speech from UCLASS for experimentation. It comprises two female speakers and 18 male speakers aged 7 years 8 months to 17 years 9 months. The stuttered speech samples are manually identified and segmented from the selected speech samples. The segmented samples were labeled as five classes, namely, Fluent, Prolongation, Syllable Repetition, Word Repetition, and Phrase Repetition. The speech samples were split into training testing and validation datasets. Firstly, the signals are pre-processed by removing the silent regions from the samples using the combination of STE and ZCR techniques. Then 14-dimensional acoustic features were extracted from the segmented samples using the WMFCC feature extraction algorithm. Finally, the extracted feature vectors are inputted to the deep learning Bi-LSTM model. The Bi-LSTM model is trained and optimized through training and validation sets by reconfiguring the hyperparameters. The performance of the proposed model is compared with the traditional LSTM model by using the test set.

A. Adjustments of Parameters

In the training model, various hyperparameters of deep learning classification such as learning rate, batch size, number of epochs, and number of hidden units, also play a vital role in the performance of the learned model.

When training the Bi-LSTM network, these parameters are tuned, and their accuracy on the validation set is observed. The experiments were performed based on the hyperparameters' configuration tabled in Table V.

For the first experiment, the best value of the initial learning rate was determined while fixing the typical values for mini batch-size as 16, the number of epochs as 100, and the number of hidden units as 100. The learning rate was varied from 10^{-2} to 10^{-4} for analysis, and the result is presented in Fig. 10. It can be seen that 10^{-2} as the initial learning rate, generated better classification accuracy of 86.67% for available stuttered data.

In the second experiment, the effect of batch-size values 4,8,16 and 32 was determined by fixing the initial learning rate to the best value obtained in the last experiment while the other two with their typical values. The average classification accuracy versus batch size is represented in Fig. 11. The experiment showed that the model produced the highest classification accuracy of 96.67% for the value of mini batch-size as 8.

The effect of the number of epochs was analyzed in the third observational study by fixing the learning rate and batch size as their best values while the typical value for the number of hidden units. The study discussed the effect of different values of epochs, such as 5, 10, 30, 50, and 100. The results are presented in Fig. 12. It can be figured out that number of epochs as 50 outputs best recognition accuracy of speech disfluencies with a value of 96.67%.

Finally, the last experiment was carried out to determine the effect of the various number of hidden units by using the best parameters obtained from the last three experiments. The number of hidden units was varied from 50 to 200 for analysis, and the result is presented in Fig. 13. It can be seen that hidden units as 100 generated better classification accuracy of 96.67%.

TABLE V. EXPERIMENTS OF HYPER-PARAMETERS CONFIGURATION

Experiments	Learning Rate	Batch-Size	Epochs	Hidden Units
Learning Rate	10^{-2} to 10^{-4}	16	100	100
Batch-Size	10^{-2}	4 to 16	100	100
Epochs	10^{-2}	8	5 to 100	100
Hidden Units	10^{-2}	8	50	100

From the experiments, it was determined that the optimal value for learning rate, batch size, number of epochs, and number of hidden units was 10^{-2}, 8, 50, and 100, respectively.

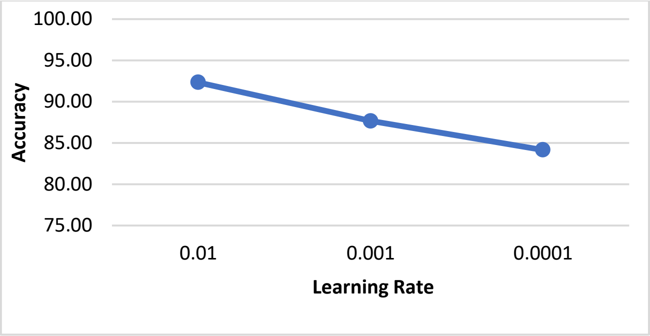

Fig. 10. Changes between Learning Rates and Accuracy.

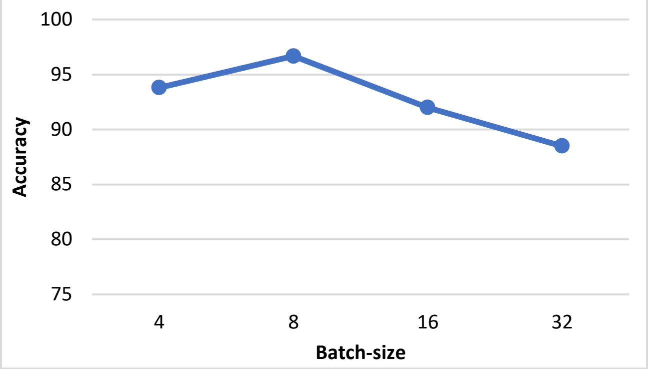

Fig. 11. Changes between Batch Size and Accuracy.

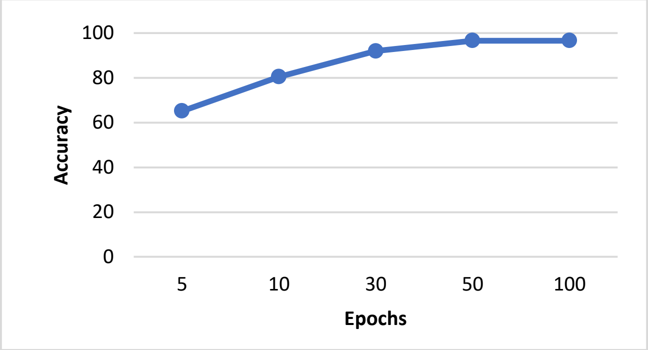

Fig. 12. Changes between Epochs and Accuracy.

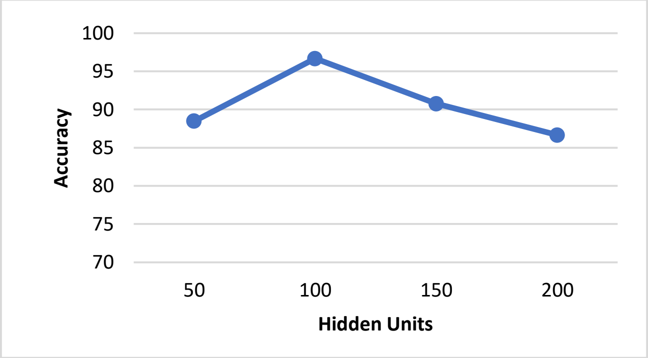

Fig. 13. Changes between the Hidden units and Accuracy.

B. Analysis of Experimental Results

The classification efficiency of the proposed WMFCC and Bi-LSTM based model is verified by carrying out the comparison experiments of the proposed model and unidirectional LSTM. During the experiment, the dimension of the WMFCC feature vector was 14, the frame length was 30ms with overlapping of 75%, the pre-emphasis factor alpha was 0.98, the single Bi-LSTM layer with 100 hidden units, the activation function was Adam, the epochs was 50, the batch-size was 8, and the learning rate was set to 10^{-2}.

The accuracy and loss function of Bi-LSTM and LSTM is represented in Fig. 14 and 15. From Fig. 14, it can be observed that the Bi-LSTM model has slow convergence speed and high accuracy as compared to the LSTM model. From Fig.15, it can be seen that the Bi-LSTM model decreases the loss value to a shallow stable value as compared to LSTM. Thus, it is concluded that the proposed model accomplished a stronger convergence effect.

The complete illustration of the validity of the proposed model can be performed by using the evaluation indicators of relevant experiments such as precision, recall, specificity, and F measure according to the confusion matrix, on test datasets.

The comparison of the LSTM model and the proposed Bi-LSTM model is displayed in Table VI. The results elucidated that WMFCC and Bi-LSTM based model proposed in this work provides the best and efficient performance and the average overall classification accuracy as 96.67%.

Table VII displays the accuracy, sensitivity and specificity of various disfluency classes. In terms of detecting stuttered events, prolongation detection, and phrase detection displayed the highest sensitivity of 97.5%. Classification of word repetition samples gave the best specificity of 99.37%. The prolongation detection achieved the highest accuracy of 98.67%.

From the analysis of the above results, it is concluded that the proposed model performs better than other models, thus determining the effectiveness of long term and bidirectional dependence on information for stuttered speech analysis. Further, the feature extraction of WMFCC includes the dynamic information of the speech samples, which increases the detection accuracy of stuttered events; and also reduces the computational overhead to the classification stage.

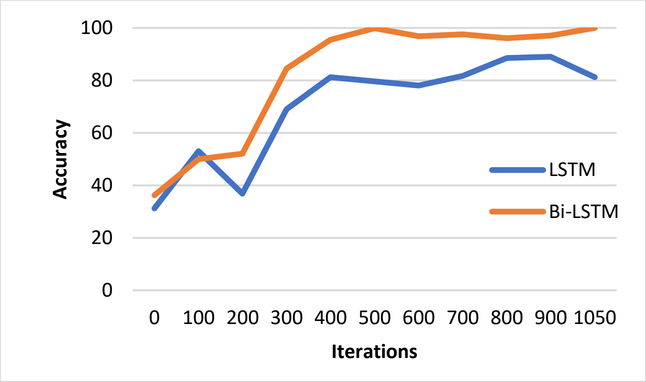

Fig. 14. Accuracy Comparison of LSTM and Bi-LSTM Models.

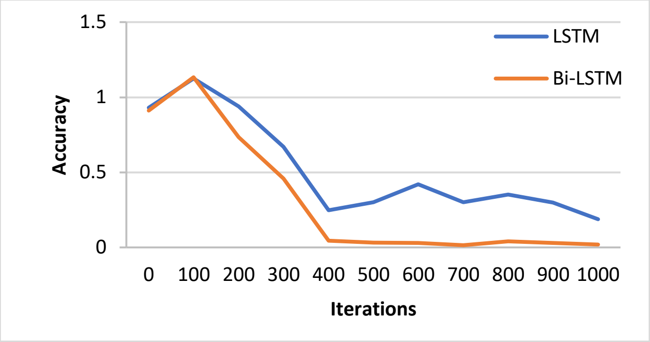

Fig. 15. Loss Comparison of LSTM and Bi-LSTM Models.

TABLE VI. CLASSIFICATION RESULTS OF DISFLUENCY CLASSES

Disfluency type	Accuracy (%)	Sensitivity (%)	Specificity (%)
Fluent	97.33	90	98.75
Prolongation	98.67	97.5	98.12
Syllable Repetition	97.5	92.5	98.12
Word Repetition	97.19	87.5	99.37
Phrase repetition	97.67	97.5	96.87

TABLE VII. COMPARISON OF LSTM MODEL AND PROPOSED MODEL

Model	Precision (%)	Recall (%)	F-Score (%)	Accuracy (%)
LSTM	83.67	84.11	83.88	83.33
Bi-LSTM	96.18	96.31	96.01	96.67

The result summary of this study (Table VII) and previous works results in Table I give comparable results. However, a direct comparison cannot be made due to different languages, different classifiers, and different types, size, and categorical distribution of stuttered speech database, as well as ways of segmentation of database for gathering, stuttered speech samples.

V. CONCLUSION

The present research proposed an automated and efficient method based on the WMFCC feature extraction algorithm and deep-learning Bi-LSTM network for automatic assessment of the stuttered speech. The disfluencies such as prolongation and syllable, word, and phrase repetition are accurately detectable using this method. The speech samples are parameterized into 14-dimensional WMFCC feature vectors. This model can extract static as well as dynamic acoustic features by using WMFCC, which enhances the detection accuracy of stuttered events; and also reduces the computational overhead to the classification stage. The feature vectors are modeled by Bi-LSTM in both forward and backward directions and capable of learning the long dependencies, taking full account of disfluency patterns in speech frames. Experiments show that when the hyper-parameters are reconfigured during the training of the model, results in an optimal configuration of parameters and leads to a highly accurate model. The optimally configured model proposed in this study is compared with the unidirectional

LSTM model. The disfluency classification accuracy of the proposed model has a better classification accuracy of 96.67% than the LSTM model. It can be concluded that the WMFCC and Bi-LSTM based proposed model effectively improves the recognition accuracy of stuttered events.

In the future study, other feature extraction and classification techniques may be applied for improving the process of detection of speech disfluencies.

References

[1] Guitar, Stuttering : an integrated approach to its nature and treatment, Fifth edition. Philadelphia ;;Baltimore: Wolters Kluwer ;Lippincott Williams & Wilkins, 2019.

[2] W. JOHNSON, “Measurements of oral reading and speaking rate and disfluency of adult male and female stutterers and nonstutterers.,” J. Speech Hear. Disord., vol. (Suppl 7), pp. 1–20, Jun. 1961.

[3] S. Erickson and S. Block, “The social and communication impact of stuttering on adolescents and their families,” J. Fluency Disord., vol. 38, no. 4, pp. 311–324, Dec. 2013.

[4] O. BLOODSTEIN and N. B. RATNER, A handbook on stuttering, 6th ed. NY: Thomson Delmar Learning, 2008.

[5] S. Gupta, R. S. Shukla, and R. K. Shukla, “Literature survey and review of techniques used for automatic assessment of Stuttered Speech,” Int. J. Manag. Technol. Eng., vol. IX, no. X, pp. 229–240, 2019.

[6] T. S. Tan, Helbin-Liboh, A. K. Ariff, C. M. Ting, and S. H. Salleh, “Application of Malay speech technology in Malay speech therapy assistance tools,” 2007 Int. Conf. Intell. Adv. Syst. ICIAS 2007, pp. 330–334, 2007.

[7] I. Świetlicka, W. Kuniszyk-Jóźkowiak, and E. Smołka, “Artificial neural networks in the disabled speech analysis,” Adv. Intell. Soft Comput., vol. 57, pp. 347–354, 2009.

[8] K. M. Ravikumar, R. Rajagopal, and H. C. Nagaraj, “An Approach for Objective Assessment of Stuttered Speech Using MFCC Features,” vol. 9, no. 1, pp. 19–24, 2009.

[9] Thiang and Wanto, “Speech Recognition Using LPC and HMM Applied for Controlling Movement of Mobile Robot,” Semin. Nas. Teknol. Inf. 2010, no. Seminar Nasional Teknologi Informasi 2010 SPEECH, 2010.

[10] L. S. Chee, O. C. Ai, M. Hariharan, and S. Yaacob, “MFCC based recognition of repetitions and prolongations in stuttered speech using k-NN and LDA,” SCOReD2009 - Proc. 2009 IEEE Student Conf. Res. Dev., pp. 146–149, 2009.

[11] L. S. Chee, O. C. Ai, M. Hariharan, and S. Yaacob, “Automatic detection of prolongations and repetitions using LPCC,” Int. Conf. Tech. Postgraduates 2009, TECHPOS 2009, 2009.

[12] Ravi Kumar K M and S. Ganesan, “Comparison of Multidimensional MFCC Feature Vectors for Objective Assessment of Stuttered Disfluencies,” Int. J. Adv. Netw. Appl., vol. 860, pp. 854–860, 2011.

[13] O. Chia Ai, M. Hariharan, S. Yaacob, and L. Sin Chee, “Classification of speech dysfluencies with MFCC and LPCC features,” Expert Syst. Appl., vol. 39, no. 2, pp. 2157–2165, 2012.

[14] M. Hariharan, V. Vijean, C. Y. Fook, and S. Yaacob, “Speech stuttering assessment using sample entropy and Least Square Support Vector Machine,” Proc. - 2012 IEEE 8th Int. Colloq. Signal Process. Its Appl. CSPA 2012, pp. 240–245, 2012.

[15] C. Y. Fook, H. Muthusamy, L. S. Chee, S. Bin Yaacob, and A. H. B. Adom, “Comparison of speech parameterization techniques for the classification of speech disfluencies,” Turkish J. Electr. Eng. Comput. Sci., vol. 21, no. SUPPL. 1, pp. 1983–1994, 2013.

[16] I. Świetlicka, W. Kuniszyk-Jóźkowiak, and E. Smołka, “Hierarchical ANN system for stuttering identification,” Comput. Speech Lang., vol. 27, no. 1, pp. 228–242, 2013.

[17] J. Pálfy, “Analysis of Dysfluencies by Computational Intelligence,” Information Sci. Technol. Bull. ACM Slovakia, vol. 6, no. 2, pp. 45–58, 2014.

[18] S. Jabeen and K. M. Ravikumar, “Analysis of 0dB and 10dB babble noise on stuttered speech,” Proc. IEEE Int. Conf. Soft-Computing Netw. Secur. ICSNS 2015, pp. 0–4, 2015.

[19] P. S. Savin, P. B. Ramteke, and S. G. Koolagudi, “Recognition of repetition and prolongation in stuttered speech using ANN,” in Smart Innovation, Systems and Technologies, 2016, vol. 43, pp. 65–71.

[20] P. B. Ramteke, S. G. Koolagudi, and F. Afroz, “Repetition detection in stuttered speech,” in Smart Innovation, Systems and Technologies, 2016, vol. 43, pp. 611–617.

[21] P. Mahesha and D. S. Vinod, “Automatic segmentation and classification of dysfluencies in stuttering speech,” ACM Int. Conf. Proceeding Ser., vol. 04-05-Marc, 2016.

[22] I. Esmaili, N. J. Dabanloo, and M. Vali, “Automatic classification of speech dysfluencies in continuous speech based on similarity measures and morphological image processing tools,” Biomed. Signal Process. Control, vol. 23, pp. 104–114, 2016.

[23] S. Ghonem, S. Abdou, M. Esmael, and N. Ghamry, “Classification of Stuttering Events Using I-Vector,” Egypt. J. Lang. Eng., vol. 4, no. 1, pp. 11–19, Apr. 2017.

[24] G. Bhatia, B. Saha, M. Khamkar, A. Chandwani, and R. Khot, “Stutter Diagnosis and Therapy System Based on Deep Learning,” Jul. 2020, Accessed: Aug. 16, 2020.

[25] Girirajan S, Sangeetha R, Preethi T, and Chinnappa A, “Automatic Speech Recognition with Stuttering Speech Removal using Long Short-Term Memory (LSTM),” Int. J. Recent Technol. Eng., vol. 8, no. 5, pp. 1677–1681, Jan. 2020.

[26] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, “Speech Recognition Using Deep Neural Networks: A Systematic Review,” IEEE Access, vol. 7, pp. 19143–19165, 2019.

[27] P. J. Lou, P. Anderson, and M. Johnson, “Disfluency detection using auto-correlational neural networks,” in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018, 2020, pp. 4610–4619.

[28] M. Reisser, “Recurrent Neural Networks in Speech Disfluency Detection and Punctuation Prediction,” 2015.

[29] Y. Cho and L. K. Saul, “Kernel Methods for Deep Learning,” Adv. neural Inf. Process. Syst., pp. 342–350, 2009.

[30] G. Zhong, X. Ling, and L. N. Wang, “From shallow feature learning to deep learning: Benefits from the width and depth of deep architectures,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 9, no. 1. Wiley-Blackwell, Jan. 01, 2019.

[31] C. Chakraborty and P. H. Talukdar, “Issues and Limitations of HMM in Speech Processing: A Survey,” Int. J. Comput. Appl., vol. 141, no. 7, pp. 13–17, May 2016.

[32] C. P. Lim, S. C. Woo, A. S. Loh, and R. Osman, “Speech recognition using artificial neural networks,” in Proceedings of the 1st International Conference on Web Information Systems Engineering, WISE 2000, 2000, vol. 1, pp. 419–423.

[33] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, “Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies,” 2001.

[34] Y. Bengio, P. Simard, and P. Frasconi, “Learning Long-Term Dependencies with Gradient Descent is Difficult,” IEEE Trans. Neural Networks, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[35] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” 2013.

[36] C. Olah, “Understanding LSTM Networks.” 2015, Accessed: Aug. 16, 2020.

[37] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” Int. J. Uncertainty, Fuzziness Knowlege-Based Syst., vol. 6, no. 2, pp. 107–116, Apr. 1998.

[38] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM networks,” in Proceedings of the International Joint Conference on Neural Networks, 2005, vol. 4, pp. 2047–2052.

[39] P. Howell, S. Davis, and J. Bartrip, “Europe PMC Funders Group The UCLASS archive of stuttered speech,” vol. 52, no. 2, pp. 556–569, 2010.

[40] Rabiner, L. R. and B. H. Juang, Fundamentals of speech recognition. Prentice-Hall, Inc., USA., 1993.

[41] L. Rabiner, Digital processing of speech signals. Englewood Cliffs N.J.: Prentice-Hall, 1978.

[42] D. Enqing, L. Guizhong, Z. Yatong, and C. Yu, “Voice activity detection based on short-time energy and noise spectrum adaptation,” Int. Conf. Signal Process. Proceedings, ICSP, vol. 1, no. 1, pp. 464–467, 2002.

[43] R. G. Bachu, S. Kopparthi, B. Adapa, and B. D. Barkana, “Voiced/unvoiced decision for speech signals based on zero-crossing rate and energy,” Adv. Tech. Comput. Sci. Softw. Eng., pp. 279–282, 2010.

[44] P. Howell and M. Huckvale, “Facilities to assist people to research into stammered speech.,” Stammering Res., vol. 1, no. 2, pp. 130–242, Jul. 2004, Accessed: Jul. 19, 2020.

[45] S. Gupta, R. S. Shukla, R. K. Singh, and R. K. Shukla, “Weighted Mel Frequency Cepstral Coefficient based feature extraction for automatic assessment of stuttered speech using Bi-directional Long-Short Term Memory”, unpublished.

[46] X. Huang, Spoken language processing : a guide to theory, algorithm and system development. Upper Saddle River N.J.: Prentice Hall PTR, 2001.

[47] S. V.Chapaneri, “Spoken Digits Recognition using Weighted MFCC and Improved Features for Dynamic Time Warping,” Int. J. Comput. Appl., vol. 40, no. 3, pp. 6–12, 2012.

[48] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[49] M. Phi, “Illustrated Guide to LSTM’s and GRU’s: A step by step explanation | by Michael Phi | Towards Data Science,” 2018. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

[50] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, 1997.

[51] A. Graves, N. Jaitly, and A. Mohamed, “Hybrid speech recognition with deep bidirectional LSTM Alex Graves , Navdeep Jaitly and Abdel-rahman Mohamed University of Toronto Department of Computer Science 6 King ’ s College Rd . Toronto , M5S 3G4 , Canada,” pp. 273–278, 2013.

[52] Z. Cui, R. Ke, Z. Pu, and Y. Wang, “Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values,” Transp. Res. Part C Emerg. Technol., vol. 118, Sep. 2020.

[53] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “The Performance of LSTM and BiLSTM in Forecasting Time Series,” in Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019, Dec. 2019, pp. 3285–3292.

[54] Y. Ren, P. Zhao, Y. Sheng, D. Yao, and Z. Xu, “Robust Softmax Regression for Multi-class Classification with Self-Paced Learning,” 2017.

[55] J. Leonel, “Hyperparameters in Machine /Deep Learning | by Jorge Leonel | Medium,” Apr. 2019. https://medium.com/@jorgesleonel/hyperparameters-in-machine-deep-learning-ca69ad10b981.