Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Novel Hybrid GRU-LSTM Model for Robust Speech Recognition of Indian Languages in Noisy Environments

Astha Gupta
Chitkara University Institute of Engineering & Technology, Chitkara University
Rakesh Kumar
Chandigarh University
Meenu Gupta
Chandigarh University
Durgesh Srivastava
Chitkara University Institute of Engineering & Technology, Chitkara University
Amit Garg
amit.garg@jaipur.manipal.edu
Manipal University Jaipur

Research Article

Keywords: ASR, Hybrid GRU-LSTM, Feature Extraction, Spectrogram, STFT

Posted Date: May 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9454210/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Novel Hybrid GRU-LSTM Model for Robust Speech Recognition of Indian Languages in Noisy Environments

Astha Gupta[1], Rakesh Kumar[2], Meenu Gupta[3] , Durgesh Srivastava[4], Amit Garg[5]

Department of Computer Science and Engineering, Chitkara University Institute of Engineering & Technology, Chitkara University, Rajpura, Punjab, India[1,4]
Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India[2,3]
Department of Computer Science and Engineering, Manipal University Jaipur, Jaipur, Rajasthan, India[5]

astha.gupta26d@gmail.com[1], rakesh77kumar@gmail.com[2], gupta.meenu5@gmail.com[3] , drdkumar.ptu@gmail.com[4], amit.garg@jaipur.manipal.edu[5]

Abstract

Speech is a naturally occurring phenomenon that depends on the speaker, surroundings, language, accent, and tone. During the speech, the waveforms change over time, making it challenging for Automatic Speech Recognition (ASR) system to analyze and identify these signals in a noisy environment. In this work, A hybrid model GRU-LSTM a combination of Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM), is proposed to recognize the speeches of the five Indian languages (i.e., Bangla, Malayalam, Odia, Telugu, and Urdu). A total of 8682 speech samples of the abovementioned languages are used from which various features are extracted using techniques such as Spectrogram, Short Term Fourier Transform (STFT), Spectral Rolloff, Spectral Bandwidth, Mel-Frequency Cepstral Coefficient (MFCC), and Chroma feature. In addition, the proposed model is tuned at three different learning rates such as 0.1, 0.001, and 0.0001, where each learning rate is analyzed over 0.2, 0.3, and 0.4 dropouts, respectively, to obtain the highest accuracy. By this, the highest validation accuracy of 97.02% with a loss of 0.41 is achieved by tuning the model at 0.001 learning rate at 0.3 dropout. Next, the performance of proposed and SOTA (State of The Art) models for each language is evaluated based on Training Accuracy, Loss, Validation Accuracy, and Loss with Precision, Recall, F1-Score, and Confusion Matrix. Finally, an analogy of the proposed with other SOTA models is also depicted. The performance of each model for each language is evaluated based on Precision, Recall, F1-Score, Accuracy, Loss, and Confusion Matrix. Finally, it is concluded that the proposed model attained the validation accuracy of 96.56%, 96.56%, 98.56%, 94.59% and 98.95% is achieved with 0.469, 0.599, 0.469,0.469 and 0.059 loss for Bangla, Malayalam, Odia, Telugu and Urdu languages.

Keywords: *ASR; Hybrid GRU-LSTM; Feature Extraction; Spectrogram; STFT.*

1 INTRODUCTION

Speech is a verbal form of communication used by humans to express thoughts and feelings and share information. It is an artistic type of biometric feature used for communication and automated applications such as emotion recognition, speech recognition, speaker identification, language identification, speaker verification, multimedia dialog system, and so on [1]. While humans can produce an infinite variety of sounds, only a subset carry linguistic meaning. Speech Recognition (SR) is the process of identifying human speech, or the machine's ability to determine the spoken words and change them into readable text [2]. The steps involved in recognizing the speech are as follows; firstly, the speech is analysed and divided into parts; further, it is transformed into a machine-readable format; and finally, an algorithm is used to match the most suitable pattern. Various speech patterns, speaking styles, languages, dialects, accents, and phrasings are used to train software algorithms that interpret and organize audio into text.

The SRS is a user-friendly interface that helps humans to interact with machines through speech. It converts the speech signal into various patterns, extracts features, and allows the machine to recognize the speech or the speaker [3]. An SRS is made of three main components: Vocabulary, Acoustic Model, and Language Model. *Vocabulary,* is the number of speech samples or phrases which are used for designing a system. Depending upon the size of the vocabulary, the recognition accuracy is affected. *Acoustic Model works like a translator where*

spoken speech is converted into words by recognizing the linguistic features and patterns in the speech sample. *Language Model is used to make predictions by understanding the words and* phrases based on what has been stated [4]. It is of two types, i.e., speaker-independent and speaker-dependent. The speaker-dependent systems are the model trained for each word to recognize speech patterns and expressions, whereas the speaker-independent systems do not require specific training and identify the speech, no matter who the speaker is [5]. These systems are easy to use, more reliable, and give humans a virtual assistant. It performs various tasks such as performing operations on the computer, assisting people who are blind or deaf, controlling Medical and Home appliances, and so on.

The first Speech Recognition System (SRS), named "Audrey," was built in the late 1950s; a speaker-dependent model capable of recognizing the digits from zero to nine. This model was able to achieve an accuracy of about 80%-90%. In the 1960s, the model was improvised and capable of recognizing some words such as “minus”, “plus”, “total”, “off”, and “false”. Also, in the 1970s and 1980s, the SRS was built to recognize more than 20,000 words [6]. Further, in the 1990s and 2000s, when Artificial Intelligence (AI) and Machine Learning (ML) methods such as Neural Network, Deep Neural Network (DNN) came into the picture, it led to a new generation for the development of SRS. Incorporating these techniques in SRS helps enhance the capabilities of SRS by being more accurate, efficient, and robust. The ML algorithms such as Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), and DNN are used for Acoustic Modelling in SRS. In contrast, RNN, LSTM, Transformer model, and n-gram techniques are used as Language Models for SRS. These multimodal approaches have significantly enhanced accuracy while addressing persistent challenges like background noise, accent variation, and vocabulary differences [7]. Various investigations have been done in this area, making it an exciting technology that people use today.

In addition, the SRS plays a significant role in many domains such as Military, Education, Translation (one language to another), Automotive, Bio-metrics, etc. There is a wide scope of ASR system regarding application, functionality, and limitations. Some aspects where ASR system is used are: it analyzes the input audio and represents it in a textual or graphical format. It allows users to be interactive as voice assistants or voice-controlled systems help them do many tasks, such as making calls, playing music, setting reminders, controlling home appliances, and many more. It also helps to convert live speech or recording into textual documents. A user can search for many things using voice-based search engines. The main use of ASR system is for individuals with disabilities, people suffering from speech impairments, Sightless, limb disability, and motor disabilities. Despite the fact that this technology has been successfully developed, several problems still reduce the accuracy of the system.

The significant factors affecting these systems' accuracy are background noise, environment, weather conditions hampering a person’s speech, similar words, speaker accents, vocal strains, and illness. Among these, environmental noise remains the primary challenge, necessitating robust solutions for real-world applications.

The paper proposes a novel "Hybrid GRU-LSTM" model that combines Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) architectures to recognize speech in noisy environments for five Indian languages (Bangla, Malayalam, Odia, Telugu, and Urdu). The model processes 8,682 speech samples using multiple feature extraction techniques including Spectrogram, Short Term Fourier Transform, Spectral Rolloff, Spectral Bandwidth, Mel-Frequency Cepstral Coefficient, and Chroma feature. The hybrid model is optimized by testing different learning rates (0.1, 0.001, 0.0001) and dropout rates (0.2, 0.3, 0.4), achieving the highest validation accuracy of 97.02% with 0.41 loss at 0.001 learning rate and 0.3 dropout. When evaluated against state-of-the-art models (RNN, GRU, LSTM, and Simple RNN-GRU), the proposed hybrid model demonstrates superior performance, achieving validation accuracies

of 96.56%, 96.56%, 98.56%, 94.59%, and 98.95% for Bangla, Malayalam, Odia, Telugu, and Urdu languages respectively, marking a significant advancement in multi-lingual speech recognition under noisy conditions.

Further the rest of the paper is organized into 6 sections, where, Section 2 enlightens literature review. Section 3 focusses on Materials and Methods. Section 4 and Section 5 highlights proposed methodology and shows experimentation, results and analysis respectively. And lastly, section 6 concludes the paper with future scope.

2 LITERATURE REVIEW

Over the last few decades, ASR has made huge demand in the world for communication between a computer and human beings. Due to the increase in the continuous need for ASR technology in various applications, several researchers have worked in this area. Some of the significant research work is included in this section for Speech recognition techniques which primarily focuses on Bangla, Malayalam, Odia, Telugu, and Urdu languages.

San, N., et al [8] developed an ASR system for the low-resources languages such as Punjabi, using multilingual pre-trained models which is fine-tuned by the high-resources(donor) languages such as Hindi, Gujrati, Bengali and so on. It uses an Acoustic Token Distribution Similarity (ATDS) metric which helps to tune the model and improves the overall performance of an ASR system.

Dharmale, G., et al [9] implemented a phonetic system which is used to improve the accuracy of an ASR system. The system is implemented using Hindi and English language where the improved accuracy of 97.34% and 99.25% respectively is achieved.

Dalai, T. et al. [10] presented different statistical and deep-learning based methodology such as Conditional Random Field (CRF), CNN and BLSTM respectively for developing POS tagging in Odia language. The model is trained using Indian Languages Corpora Initiative (ILCI) corpus with Bureau of Indian Standards (BIS) and Universal Dependencies (UD) tagset. The accuracy achieved using CRF, CC and BLSTM was 92.08%, 94.48% and 94.58% respectively.

Madhavaraj, A. et al. [11] developed an ASR for a multilingual language such as Tamil, Telugu, Kannada, Hindi and Gujarati. The author proposed data pooling and knowledge-driven DNN approaches for training and recognizing the speech samples. The experimental results were combined using settings i.e., Low-resourced and Medium-resourced settings where the low-resourced shows WER of 9.66%, 7.2% and 15.21% for Tamil, Gujarati and Telugu language respectively and Medium-resourced shows 13.94%, 10.28% and 27.24% for Tamil, Kannada and Hindi, respectively.

Sultana, S. et al. [12] represented a concise report on major advancements done in ASRS for the Bangla language. A total of 24 research papers have been addressed based on different deep-learning approaches. It depicts some challenges faced while developing ASRS and states that the system's performance is increased by introducing natural Bangla speech samples in large numbers.

Noman & Cheng [13] proposed an algorithm for Bangla SRS using ANN. The study explores four phases: collection of Bangla speech samples, conversion of samples to frequency domain, filtering & normalization and implementing back propagation algorithm to recognize the speech samples. The model achieved an accuracy of 95.23%.

Sharmin et al. [14] proposed a Bengali spoken digit SRS based on CNN. The model achieved an accuracy of 98.37% using dataset spoken by ten different people having different age, gender and dialect. In addition, the author concluded that the system performance can be increased using large corpus covering all the dialects from different age-groups.

Swain et al. [15] designed an entirely new classifier made up of SoftMax and Support Vector Machine for speech emotion recognition in Odia languages using Deep Convolutional

Recurrent Neural Network. The model was tested using a dataset of seven emotional states of Odia dataset and eight states of RAVDESS dataset, it shows an accuracy of 85.31% and 77.54% respectively.

Rajisha et al. [16] discussed a methodology to recognize four emotions from Malayalam speech samples such as anger, sad, happy and neutral by using various feature extraction and classification techniques. The author used MFCC and Short Time Energy technique to extract features and analyzed those features using SVM and ANN classifier. The experimental results show the recognition accuracy for ANN is better than SVM as 88.4% and 78.2% respectively.

Bilal et al. [17] presented a traditional transformer-based hate speech detection model for Urdu language. The model is trained using the largest Roman Urdu data set of 173,714 messages. The author compares the proposed model with other DL models such as LSTM, BLSTM, and CNN. The model performance was evaluated on various standard parameters and shows an accuracy of 96.70% with respect to other models.

Shivaprasad et al. [18] proposed a methodology for dialect identification in the Telugu language. The spectral and prosodic features from the speech are extracted and passed through two classification models Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM). The author achieved accuracy for GMM, and HMM was 88.40% and 86.95%, respectively.

Naeem, S. et al. [19] presented the work in the Urdu language. Kaldi Toolkit was used to train the model using 100 hours of transcribed data and achieved a word error rate of 9%.

Moncy et al. [20] proposed an Acoustic modeling technique and feature extraction mechanism using DNN and MFCC, respectively, for the Malayalam language. The model was trained and tested using the KALDI toolkit. A Word error rate of 3.01% was observed, and a comparative analysis of the technique was done with the GMM-HMM technique. The author concluded that the DNN-MFCC technique results better than the GMM-HMM technique.

Rajan et al. [21] presented a comparative analysis of the methods which help in Part-of-Speech (POS) tagging in the Malayalam language. Two corpora were used for tagging Malayalam speech using the Decision tree and Bi-directional Long Short-Term Memory (BLSTM) algorithms. Finally, it was concluded that BLSTM is better than the Decision Tree algorithm for Tagging Malayalam speech.

Mohanty et al. [22] proposed an independent speech recognition system using MFCC and HMM for the Odia language. The system's performance was calculated by exploring different levels of HMM, and the model output was calculated using the n-gram approach.

Rakib et al. [23] proposed a methodology to predict the appropriate word coming next and sentence in the Bangla language. The technology used for building this method was GRU-based RNN. This research work focuses explicitly on the Bangla language. The overall accuracy was calculated using the n-gram model like 1-gram, 2-gram, tri-gram, etc.

Islam, J. et al. [24] proposed a speech recognition model using Convolution Neural Network (CNN) and RNN for the Bangla language. About 86.05% accuracy was achieved by using the CNN model.

Kumar et al. [25] presented a DL approach based on Part-Of-Speech tagging in Malayalam. The tagging data was evaluated using various methods such as GRU, RNN, LSTM, and BLSTM. The system produced significant results and an error rate of 13.2%. The BLSTM offers an accuracy of 87.57% as compared to other models.

Nahid, M. et al. [26] represented a network using deep LSTM and RNN to model for Bengali Speeches. The system achieved a word and phoneme error rate of 13.2% and 28.7%, respectively.

Kumar, A. et al. [27] proposed a speaker-independent system for the Telugu language to recognize continuous words. Discrete Wavelet Transformation (DWT), MFCC, HMM, and DNN techniques were used. The speech signal is pre-processed using DWT, and MFCC is used

for extracting the features, whereas HMM -DNN is used to classify the extracted speech signals and identify the speaker. The model achieved 91% accuracy in recognizing the speech, accent, and speaker.

Ali, S. et al. [28] discussed the model to translate Urdu into English language using three-module pipelines based on DNN. The model achieved a learning rate of 0.01 and a dropout rate of 5%.

Ashraf, J. et al. [29] proposed a CMU Sphinx-4 for Urdu Language. The framework used by the author showed improved and better results than the model developed earlier. The model was used for the small Vocabulary in the Urdu speech recognition system and produced a word error rate of 10.66%.

Mannepalli, K. et al. [30] presented a speech recognition system for the Telugu language using MFCC and GMM algorithms. The model was built for various coastal accents such as Rayalaseema, Andhra, and Telangana in Telugu, and their accuracy was calculated. The speech samples were collected from native speakers and extracted features using MFCC. Also, the speech was then classified by using GMM. Further, the author concluded with the performance evaluation using the n-gram approach.

Paul, B. et al. [31] presented the work on the regional languages. The author proposed an upgraded model of traditional speech recognition by incorporating noise distillation block. Using one corpus of the regional language, the author showed the result in terms of WER and CER in three different conditions such as firstly for clean data, secondly data collected in the noisy environment and lastly using noisy data and cleaning it through noise distillation block and thereafter showing the results.

The literature review identifies key challenges in ASRS, such as speech recognition in noisy environments, out-of-vocabulary issues, limited labeled data, and speaker-specific variations. Some researchers show the use of tools and algorithm to recognize speech whereas some uses various SOTA models for implementing their work.

3 MATERIALS AND METHODS

3.1 Materials

3.1.1 Speech Data Collection and Preparation

The speech corpus consists of different Indian languages, such as Bangla, Malayalam, Odia, Telugu, and Urdu are considered in this work to train the model for recognizing speech samples. The audio speech samples are taken from the Kaggle repository [32],[33].

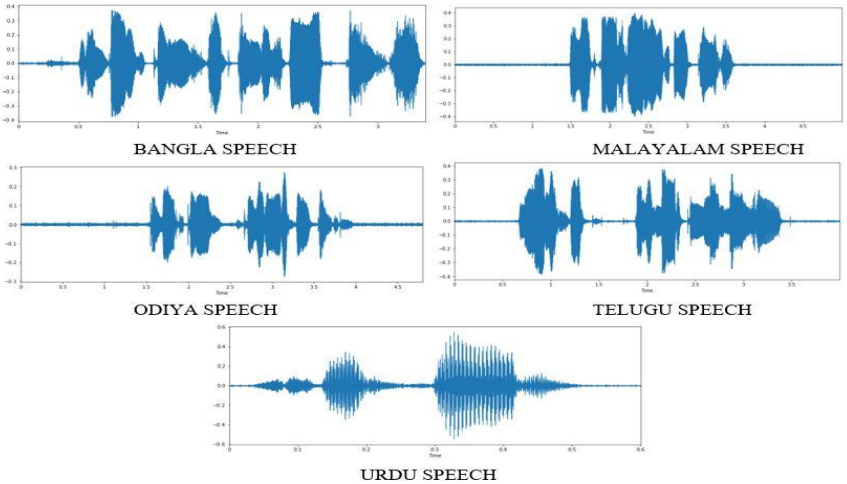


Figure 1. Visualization of Speech Sample

The first dataset contains the speech sample of 8 speakers (4 Native and 4 Non-native) and 2 Metropolitan Indian accent speakers for Bangla, Malayalam, Telugu, and Odiya languages. The second dataset contains the speech samples of 8 male and 2 female speakers (both native

and non-native) for the Urdu language. The visualization of these audio samples for each language is shown in Figure 1. Further, an EDA is applied to the collected dataset to extract the number of Speech samples based on different Indian languages, as shown in Table 1. These speech samples are investigated to analyze the main attribute, which helps the model to differentiate the speech samples.

Table 1. Number of Speech samples

Language	Number of Audio Samples
Bangla	1527
Malayalam	2393
Odia	747
Telugu	1515
Urdu	2500

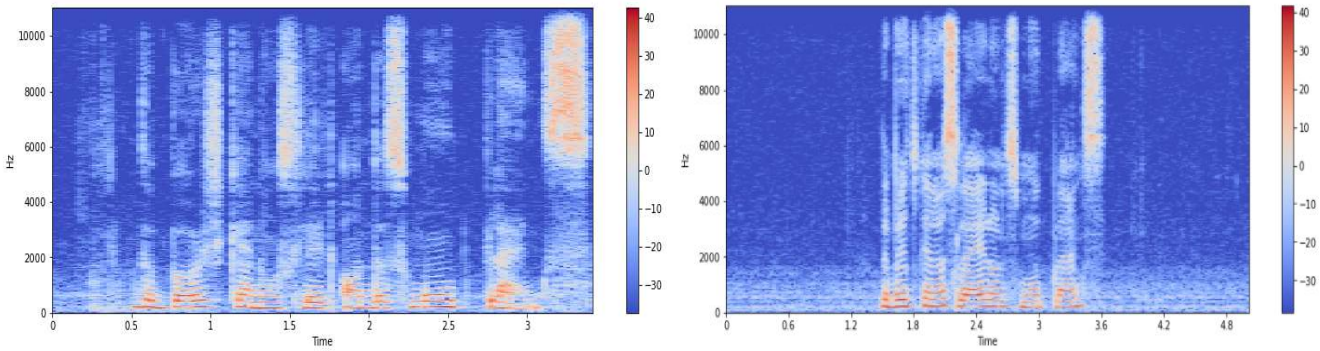
3.1.2 Data Pre-processing

The primary aim of this step-in speech recognition is to visualize the speech signal in various forms and make it more specified so that it can be adequately modeled. Various Feature Extraction Techniques such as Spectrogram, STFT, Spectral Rolloff, Spectral Bandwidth, MFCC, and Chroma feature are used for obtaining Speech features. Specific characteristics like strength, shape, frequency, and bandwidth of lower and upper frequencies of each speech sample are represented using above mentioned techniques. Here, Librosa package is used to extract and analyze the speech signals.

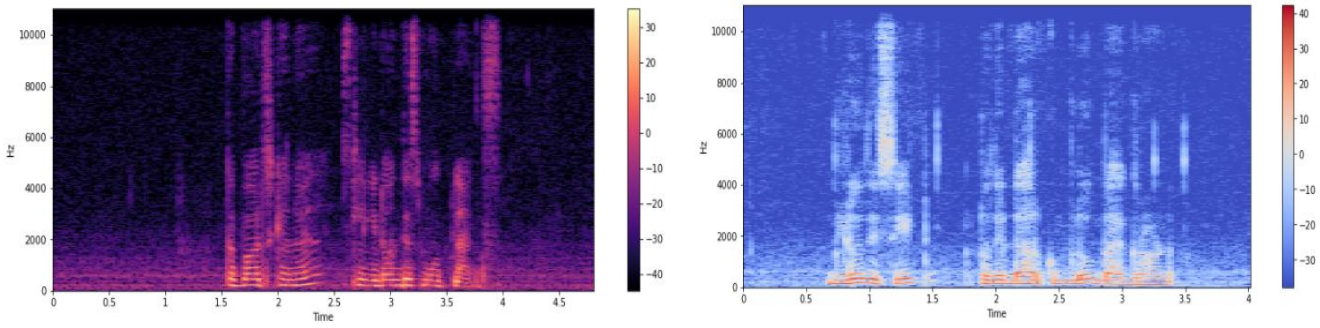
The feature extraction technique mentioned above are elaborated as under:

3.1.2.1 Spectrogram

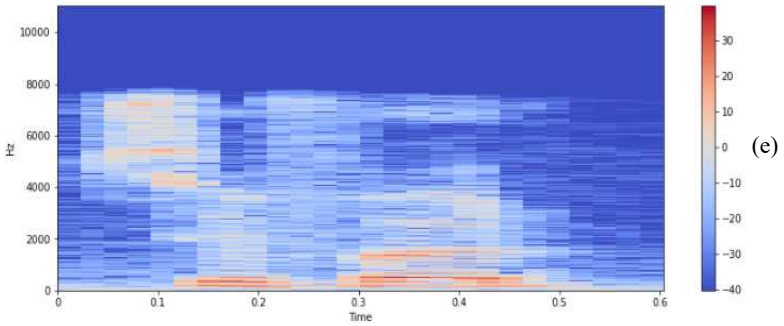
The signal strength is visually represented using the Spectrogram technique, which interprets the intensity of the actual data over time at different frequencies. Figure 2 shows the Spectrogram of each language's speech sample where time is on the x-axis, frequency is on the y-axis, and the color bar shows amplitude. The spectrogram of the speech shows the amplitude change for every frequency over time in the signal.



(a) (b)



(c) (d)



(e)

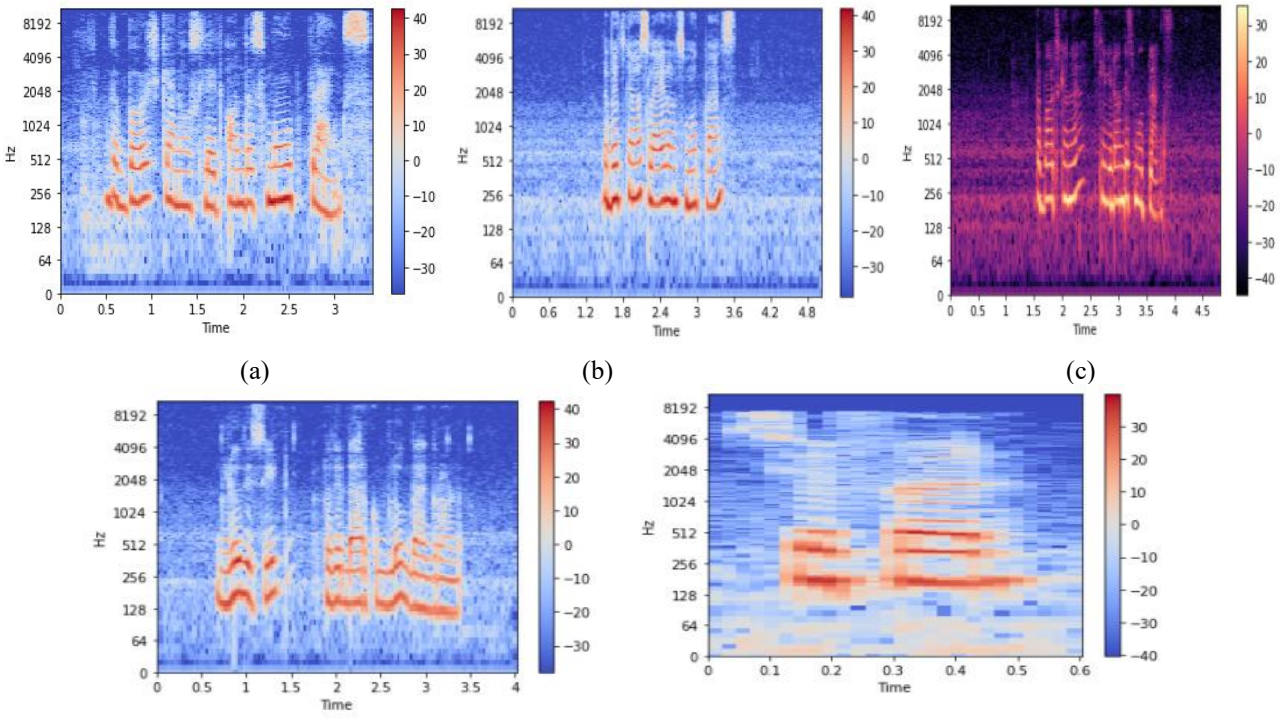
Figure 2. Spectrogram of Speech (a) Bangla (b) Malayalam (c) Odiya (d) Telugu (e) Urdu

3.1.2.2 STFT

STFT is a time-frequency distribution graph. The signal gets emphasized whenever the speech signal has some meaning, as shown in Figure 3. Equation 1 shows the mathematical calculation of STFT.

$$X(n,w) = \sum_{m=-\infty}^{\infty} x[m]w[n-w]e^{-jwm} \qquad (1)$$

$$where, x(n)\ is\ signal\ on\ time\ and\ w(n)\ is\ the\ window\ frame$$



(a) (b) (c)

(d) (e)

Figure 3. Short-term Fourier Transform of Speech (a) Bangla (b) Malayalam (c) Odiya (d) Telugu (e) Urdu

3.1.2.3 Spectral Rolloff

It is described as a filter intended to roll off frequencies outside a particular range. It measures the frequency below 85% of the total energy preserved. Figure 4 depicts the common spectral Rolloff frequency for each frame obtained for each language. The function used to compute spectral Rolloff is $librosa.feature.spectral_rolloff$.

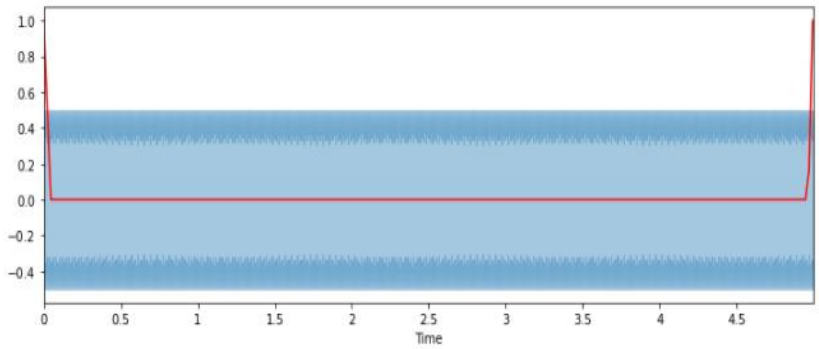


Figure 4. Speech Spectral Rolloff of Speech

3.1.2.4 Spectral Bandwidth

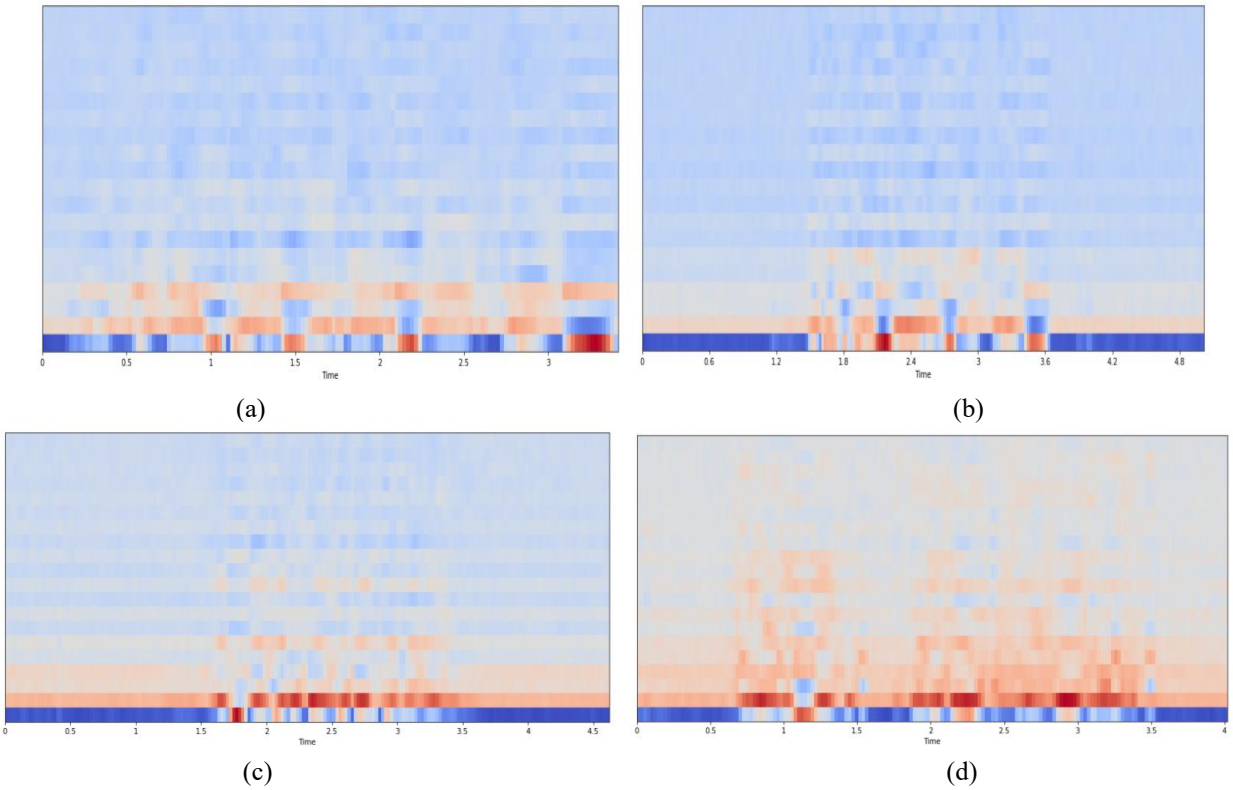
It is also known as a spectral spread which is derived from the spectral centroid. The audio signals of each language with the most energetic sound are extracted using Spectral bandwidth. The weighted mean of the frequency bins is mathematically calculated using equation 2.

$$(\sum_{k} S(k)\,(f(k)) - f_c)^p)^{1/p} \qquad (2)$$

$$where\ s(k)\ is\ spectral\ magnitude\ of\ frequency\ bin\ k$$
$$f(k)\ is\ the\ frequency\ bin\ at\ bin\ k$$
$$f(c)\ is\ the\ spectral\ centroid$$

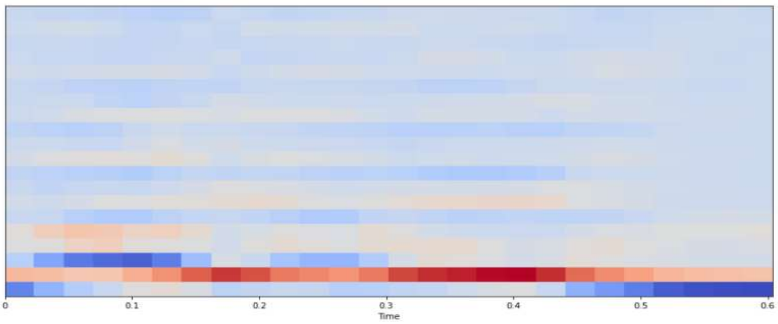
3.1.2.5 MFCC

It evenly distributes the frequency bands on the Mel scale. These are the small set of features that concisely describe the spectral envelope's overall shape. The $librosa.feature.mfcc$ function is used to calculate the MFCC of each speech sample. Figure 5 represents the MFCC feature extraction of speech, where one axis represents time, and the other represents a range.



(a) (b)

(c) (d)

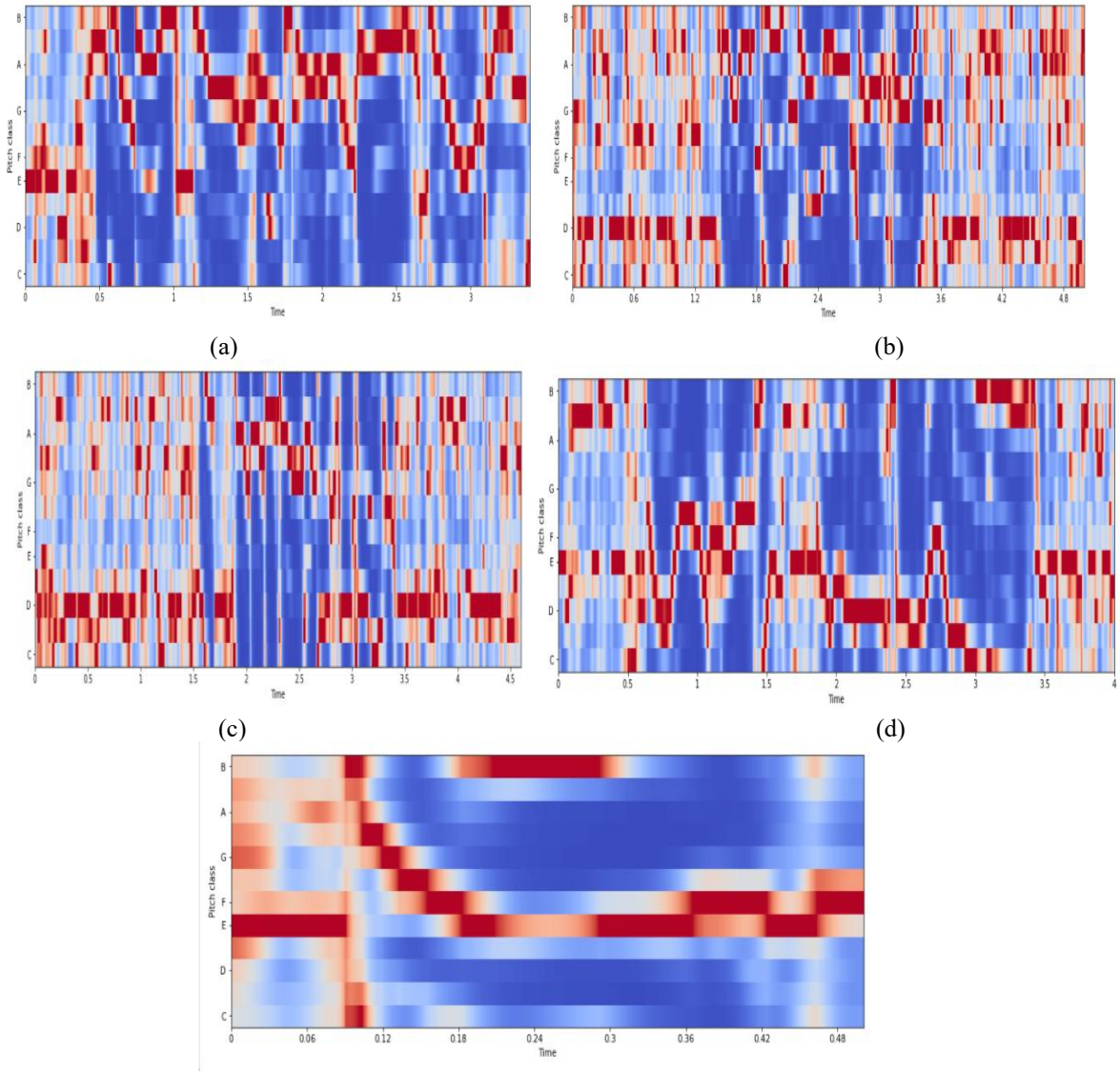


(e)

Figure 5. MFCC of Speech (a) Bangla (b) Malayalam (c) Odiya (d) Telugu (e) Urdu

3.1.2.6 Chroma Feature

It is a 12-element feature vector that indicates the amount of energy in each pitch class present in the signal. Figure 6 depicts each speech sample's chroma feature considered in this proposed work. The $librosa.feature.chroma_stft$ function is used to compute the Chroma features.



(a) (b)

(c) (d)

(e)

Figure 6. Chroma Feature of Speech (a) Bangla (b) Malayalam (c) Odiya (d) Telugu (e) Urdu

3.2 Methods

The methodology used to design the system model is GRU and LSTM. A hybrid GRU-LSTM model is proposed for recognizing speech samples of all the five languages.

GRU is an updated version of RNN and is used to analyze time-series data, text, audio, and other sequential data. The network consists of two gates, i.e., the Update gate and the reset gate with the Hidden and Candidate hidden State. Figure 7 represent a framework of GRU.

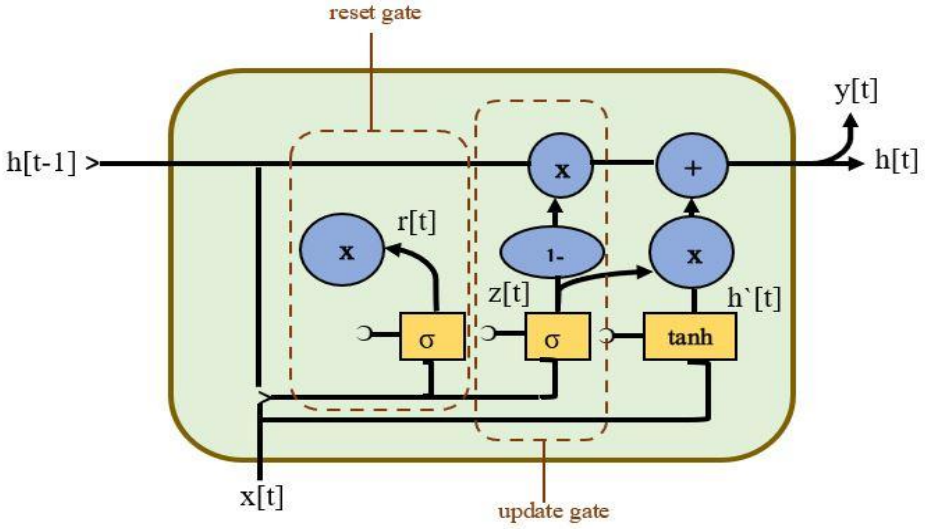


Figure 7. Framework of GRU

It works on a gating mechanism, where a particular gate controls the information flow. At each timestamp t, it takes in the input x_t and hidden state signals h_{t-1} from the preceding timestamp and generates the output. The output generated is then used for the next timestamp. Mathematically, the following set of equations from 3 to 6 are used to calculate GRU gates:

Update Gate:	$z_t=\sigma\ (W_z * [h_{t-1}, x_t])$	(3)
Reset Gate:	$r_t=\sigma\ (W_r * [h_{t-1}, x_t])$	(4)
Candidate Hidden State:	$h`_t=\tanh\ (W_h * [r_t * \{h_{t-1}, x_t\}])$	(5)
Hidden State:	$h_t= (1-z_h) * h_{t-1} + z_t * h`_t$	(6)

where, W_z, W_r and W_h are learnable weight matrices,

x_t is input at t timestamp,

h_{t-1} is preceding hidden state,

h_t is current hidden state.

The update gate z_t determines the amount of hidden state h_t information need to be updated and combined with current input x_t. The reset gate r_t decides the amount of information which is to be forgotten and not used in the calculation. The candidate hidden state $h`_t$ is based on current input $x_t$ and reset gate $r_t$ whereas Hidden state $h_t$ is combination of previous hidden state $h_{t-1}$ and candidate hidden state $h`_t$, controlled by an update gate z_t. The GRU processing works on each timestamp t, initially an input vector *x[t]* with the previous hidden state *h[t-1]* from the previous timestamp is given to the GRU unit. It is passed to the sigmoid activation function to generate output. Typically, the output of GRU is the hidden state h_t produced at each timestamp, it can be used as a final output or used further for processing. In addition, LSTM is a type of RNN and has the same functionality of working upon sequential data. It employs to keep track of crucial information from the past while processing the series of input. It makes use of memory and gating mechanism, where the difference lies in its three gates: the input gate, output gate, and forget gate. Figure 8 represents the architecture of LSTM model.

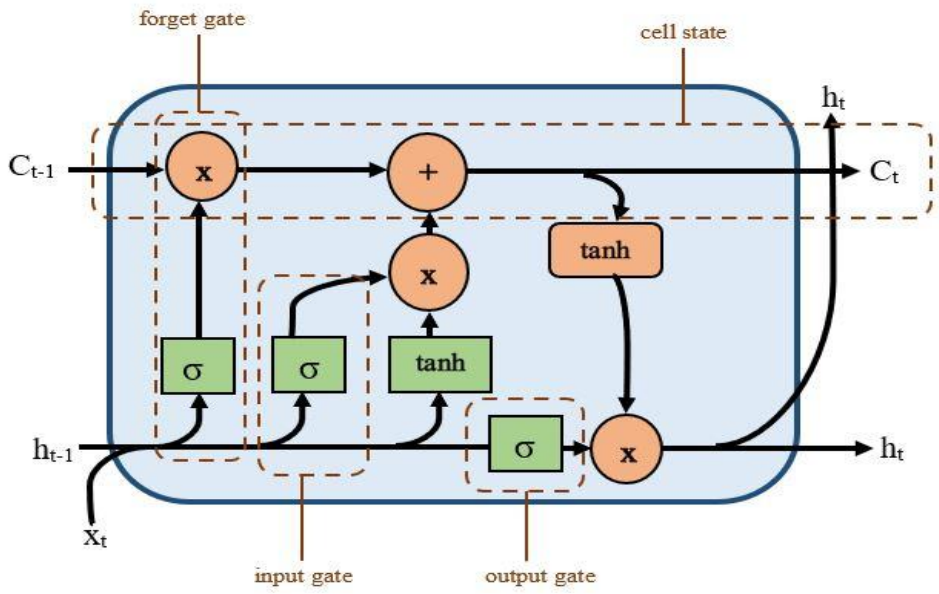


Figure 8. LSTM Architecture

The input gate i_t works to update the state of a particular cell and decides whether the information is essential or not. The output gate o_t is responsible for determining which state should be the next hidden state h_t and forget gate f_t, keeps the necessary information to calculate the state of the cell, and sees if the data can be helpful or not. LSTM is calculated using the following set of the equation from 7 to 9:

$$\textit{Input Gate:} \quad i_t = \sigma\left(w_i\,[h_{t-1}, x_t] + b_i\right) \tag{7}$$

$$\textit{Forget Gate:} \quad f_t = \sigma\left(w_f\,[h_{t-1}, x_t] + b_f\right) \tag{8}$$

$$\textit{Output Gate:} \quad o_t = \sigma\left(w_o\,[h_{t-1}, x_t] + b_o\right) \tag{9}$$

where, w_i, w_f *and* w_o *are learnable weight matrices,*

b_i, b_f *and* b_o *are bias for respective gates,*

x_t *is input at t timestamp,*

h_{t-1} *is output at preceding lstm block at t-1 timestamp.*

LSTM uses these three gates to control the flow of information into and out of the memory. These gates decide, what to forget, what needs to be remembered and what should be the output at processing sequence of each timestamp. In terms of functionality, both GRU and LSTM work in the same manner, but the difference lies in the Gates. The GRU utilizes a hidden state that combines input and forget gate into an update gate, whereas LSTM can remember data for a long time.

4 PROPOSED METHODOLOGY

This section proposes an algorithm and model for recognizing the speech using different speech samples of the Indian Language.

4.1 System Model

The system model is designed using a combination of two types of RNN, i.e., GRU and LSTM. The fusion of both the models allows them to work in tandem. The hybrid model is stacked with both GRU and LSTM unit, to make a fully connected recurrent network. It consists of four layers with 64 and 128 hidden states, respectively. During the processing of speech samples, the model takes input through both GRU and LSTM component. Thereafter, at each timestamp, the updated hidden state is evaluated by combining the output components of both

units. Finally, it is passes to activation function layer for prediction and classification of the speech samples. 'Relu' and 'SoftMax' activation function are used on the dense layer which maps 64 and 5 units to the output layer respectively. Both activation functions are non-linear as they easily adapt the different data and differentiate each output. Beside this, the dataset is divided into training and test set using holdout method. The Hybrid model uses the combination of *equation 3 to 9* for recognizing speech in an efficient manner. It is trained for 50 epochs having 32 batch size with learning rate as 0.001 and dropout as 0.3. Thereafter, the prediction and performance of the model is evaluated on standard parameters such as Precision, Recall, and F1-Score, along with accuracy, loss curves, and Confusion matrix. Figure 9 represents the proposed hybrid deep learning model.

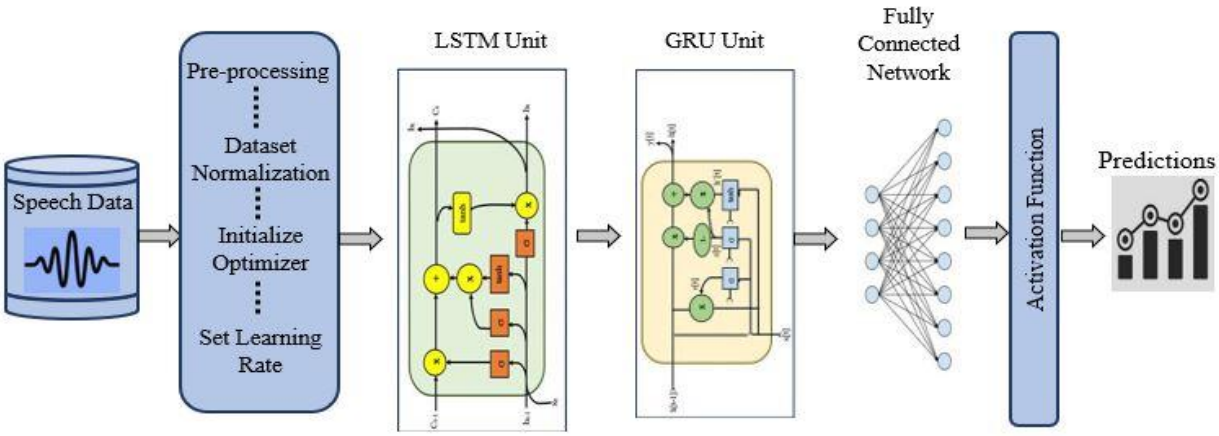


Figure 9. Proposed Hybrid Deep Learning Model

4.2 Architecture and working

The step-wise methodology adopted for the recognition of speech in a noisy environment is specified in the algorithm mentioned below. Initially, the speech dataset is pre-processed using various feature extraction technique. These techniques depict the strength and remove background noise for each speech signal. When the speech samples are pre-processed and features are extracted, the dataset is divided into train and test sub dataset. Then, the model is trained by setting optimizer and learning rate for 50 epochs with 32 batch size and results are evaluated.

Step 1. *Load, preprocess and normalize the dataset. Further split the dataset into train and test sub datasets.*

Step 2. *Initilize input units, optimizer and learning rate to train model*

Step 3. *for* 50 *epochs and* 32 *batch size to train Hybrid model* :

$$model.add\left(LSTM\left(128, input_{shape} = (None, 13), return_{sequences} = True\right)\right)$$

#Adding layers with the number of input specified and return the last output in the output sequence

$$model.add(GRU(64))$$

#Adding layers with the number of input specified

$$model.add\left(Dense(64, activation = 'relu')\right)$$

 # Add a densely connected layer with 64 units with relu function

$$Dropout(0.3); \quad \#\ Dropout = 30\%$$

$$model.add(Dense(5, activation = 'softmax')$$

 # Add a Softmax layer with 5 output layers

Step 4. *end for*

Step 5. *Evaluation of proposed model using performance matrix such as Accuracy, Recall, Precision, F1 − Score*

This algorithm is used to train and evaluate the proposed model. It is trained for 50 epochs with Adam optimizer using different learning rates and dropouts. The Adam optimization technique requires less parameters and has a faster computation rate than other techniques, such as Gradient Descent, Adagrad, RMSProp, and Momentum. In addition, the model is tuned at three learning rates, i.e., 0.01, 0.001, and 0.0001. It is a manual fine-tuning task, and selection is made to train the model in a better way which shows less training error. At each learning rate, the models are analyzed over 0.2, 0.3, and 0.4 dropouts, respectively. The Dropout is not taken less than 0.2 and greater than 0.4, as the results are inaccurate while training the model and result in overfitting the data. The highest Training and Validation accuracy of 98.91% and 97.04%, respectively, is achieved using learning rate=0.001 and Dropout = 0.3, as shown in Table 2. Further, the models are tuned to achieved learning rate and dropout to recognize speech samples of all the five languages.

Table 2. Training and Validation Accuracy, Loss at different learning rate and Dropout

Learning Rate	Dropout	Training Accuracy (%)	Training Loss (%)	Validation Accuracy (%)	Validation Loss (%)
0.01	0.2	91.70	22.83	90.08	21.82
	0.3	96.51	13.04	93.99	16.90
	0.4	96.35	11.89	96.29	11.78
0.001	0.2	95.92	13.38	96.70	0.06
	0.3	**98.91**	**0.39**	**97.04**	**0.41**
	0.4	98.29	0.53	94.43	1.73
0.0001	0.2	96.24	15.26	97.0	10.14
	0.3	95.06	19.69	96.99	12.11
	0.4	94.91	19.44	96.42	11.45

5 EXPERIMENTATION, RESULTS AND ANALYSIS

5.1 Experimental Setup

The experiment was performed on a Windows system having Intel® Core ™ i5-4300U CPU at 1.90GHz 2.49Ghz with 8 GB RAM. The software used to complete all the experiments is Anaconda, where all the models are trained on Jupyter Notebook by installing the Python libraries. The hyperparameters values used for model training are 50 epochs with batch size 32 at 0.001 learning rate and 0.3 dropout.

5.2 Performance Metrics

The performance analysis of each model is performed using standard metrics such as Confusion Matrix, Loss, Accuracy, Precision, Recall, and F1-Score.

Confusion Matrix is an N*N matrix used to estimate the performance of the classification models. This matrix consists of observation and prediction which are either positive or negative. Loss refers to the wrong predictions or error made by the model. Accuracy represents the number of correct predictions made by the model out of the total number of predictions. It is evaluated by dividing the number of correct predictions, i.e., True Positives (TP) and True Negatives (TN) by total number of TP, TN, False Positives (FP) and False Negatives (FN) predictions. To attain accuracy in percentage, it is then multiplied by 100. The equation 10 depicts Accuracy formula,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

Precision measures the exactness of the speech signal. It is the percentage of those speech signals that the model accurately analyzes. It is calculated by dividing the proportion of correctly predicted positive values by the total number of positive predictions. Mathematically, it is interpreted as shown in equation 11,

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

Recall is used to calculate the strength of the model. It is a fraction of correct predicted outcomes by the total number of positive class (TP and FN) instances. Equation 12 expresses the formula for recall as,

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

F1-Score measure the model's overall performance by balancing precision and recall as a single unit. It is a harmonic mean of both precision and recall. The mathematical formula to calculate it is shown in Equation 13,

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (13)$$

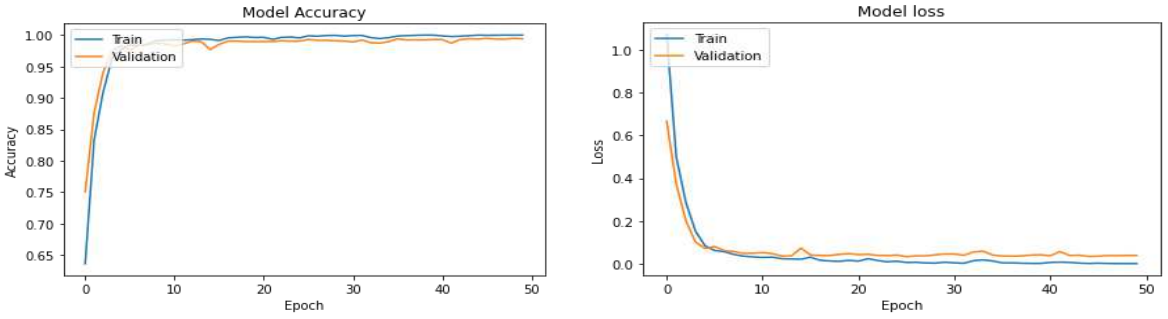
5.3 Results

The performance of the model is calculated by using the standard performance metrics. The model is trained for 50 epochs using the speech dataset mentioned in Table 1. The results obtained under different parameters for each language are mentioned in Table 3. Figures 10 and 11 depict the Accuracy, Loss curves, and Confusion matrix.

Table 3. Training and Validation Accuracy (%), Loss (%) of the Proposed Model with Precision, recall, and F1-Score

Language	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Precision	Recall	F1-Score
Bangla	97.65	0.409	96.56	0.469	98.5	97.5	95
Malayalam	97.65	0.209	96.56	0.599	98.5	97	99
Odia	99.65	0.209	98.56	0.469	96.5	99	97.5
Telugu	99.65	0.409	94.59	0.469	99	98.5	95.5
Urdu	99.95	0.009	98.95	0.059	99	99	96

Further, it is inferred from the Accuracy and loss curve of the proposed model that a stable accuracy is achieved after the fourth epoch, as shown in Figure 10(a), and minimum loss is incurred after the fifth epoch, as depicted in Figure 10(b), as compared to other Models. Figure 11 shows the Confusion Matrix of the Proposed Model.



(a) (b)

Figure 10. Graphs of Proposed Model (a) Accuracy (b) Loss

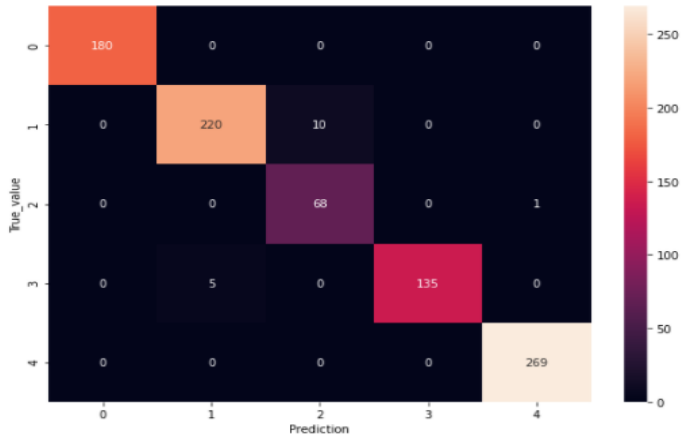


Figure 11. Confusion Matrix of Proposed Model

5.4 Comparative Analysis of Proposed Model to other SOTA models

In this work, the SOTA models such as RNN, GRU, LSTM, and Simple RNN-GRU are used to compare with the proposed model. The experimental results obtained using the performance metrics under each language are shown in Table 4.

Table 4. Training and Validation Accuracy (%), Loss (%) of SOTA Models with Precision, recall, and F1-Score under each language

Models	Language	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Precision	Recall	F1-Score
RNN	Bangla	67.09	0.79	66.89	0.59	87	87	87
	Malayalam	73.09	0.59	72.59	0.54	56	90	69
	Odia	71.09	0.39	70.59	0.86	69	75	72
	Telugu	65.09	0.99	64.59	0.54	56	65	96
	Urdu	69.5	0.60	68.5	0.59	99	99	99
GRU	Bangla	97.5	0.0023	90.69	0.5623	98	96	91
	Malayalam	98.5	0.0003	95.55	0.5903	99	95	99
	Odia	99.3	0.0053	98.46	0.5953	94	99	96
	Telugu	95.5	0.0093	94.53	0.5993	99	99	93
	Urdu	100	0.0010	99	0.5610	99	99	97
LSTM	Bangla	97.8	0.0030	95.89	0.8630	99	99	99
	Malayalam	98.8	0.0010	95.89	0.2310	98	99	99
	Odia	99.7	0.0005	96.75	0.4605	99	99	99
	Telugu	95.8	0.0060	94.88	0.5660	99	98	98
	Urdu	97.8	0.0010	96.8	0.5910	99	99	95
Simple RNN-GRU	Bangla	82.29	0.44	81.59	0.59	92.5	91.5	89
	Malayalam	88.29	0.24	87.59	0.46	77.5	92.5	84
	Odia	86.29	0.24	85.59	0.59	81.5	87	84
	Telugu	80.29	0.64	79.46	0.49	77.5	82	94.5
	Urdu	84.9	0.30	83.9	0.56	99	99	98

In addition, Accuracy and Loss curves of RNN, GRU, LSTM and Simple RNN-GRU are presented in a, b, c and d part of Figure 12 respectively. It is interpreted that the Accuracy curves of the RNN model show unstable behavior during each epoch. GRU, LSTM, Simple RNN-GRU offer some stable accuracy after ten epochs. In contrast, the Loss curves for the RNN model show a fickle loss in both the training and validation phase, and GRU, LSTM, Simple RNN-GRU depict the loss declines over each epoch. Further, the confusion matrix of each model is shown in Figure 13.

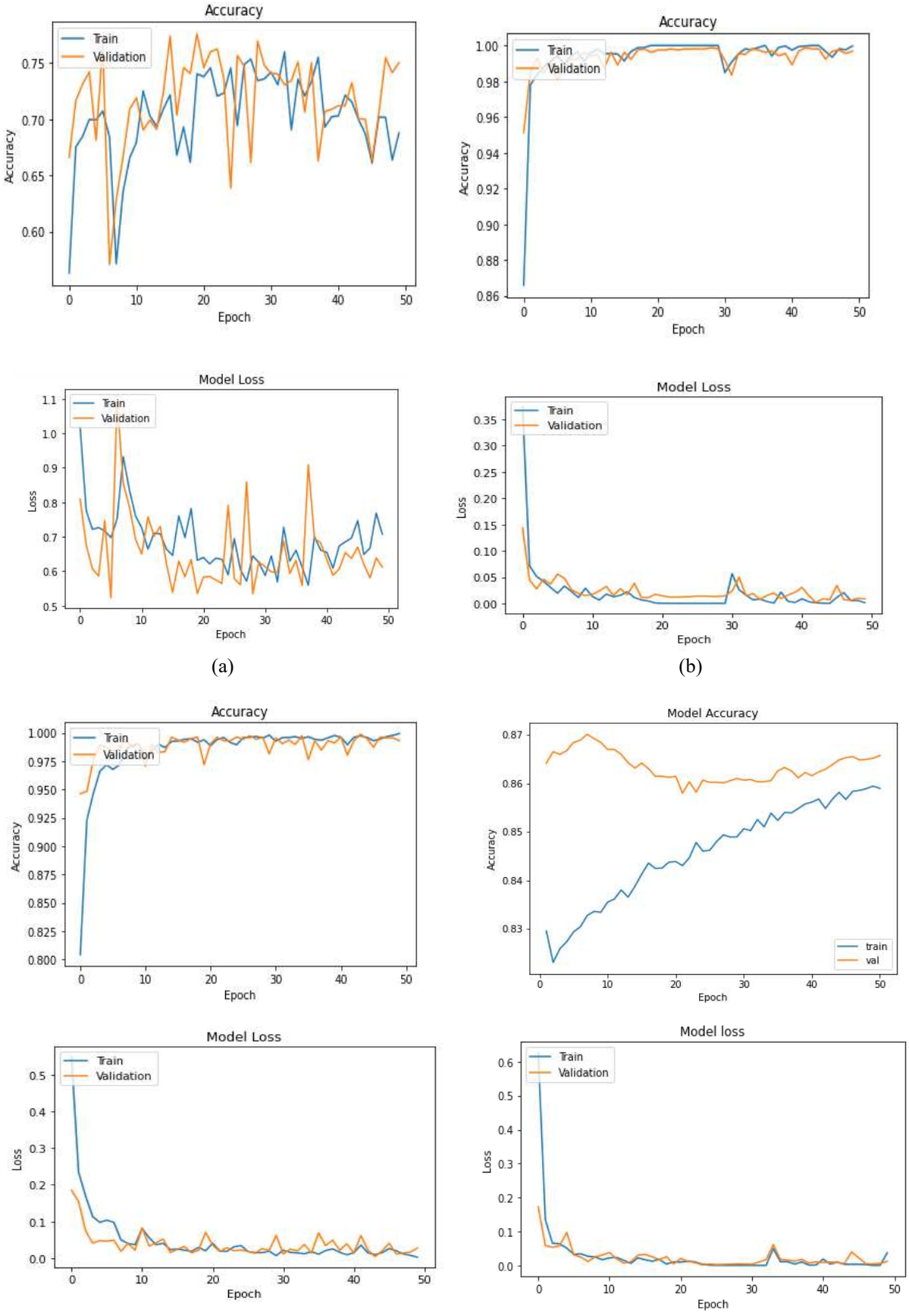


Figure 12. Accuracy and Loss graph of other models (a) RNN (b) GRU (c) LSTM (d) Simple RNN-GRU

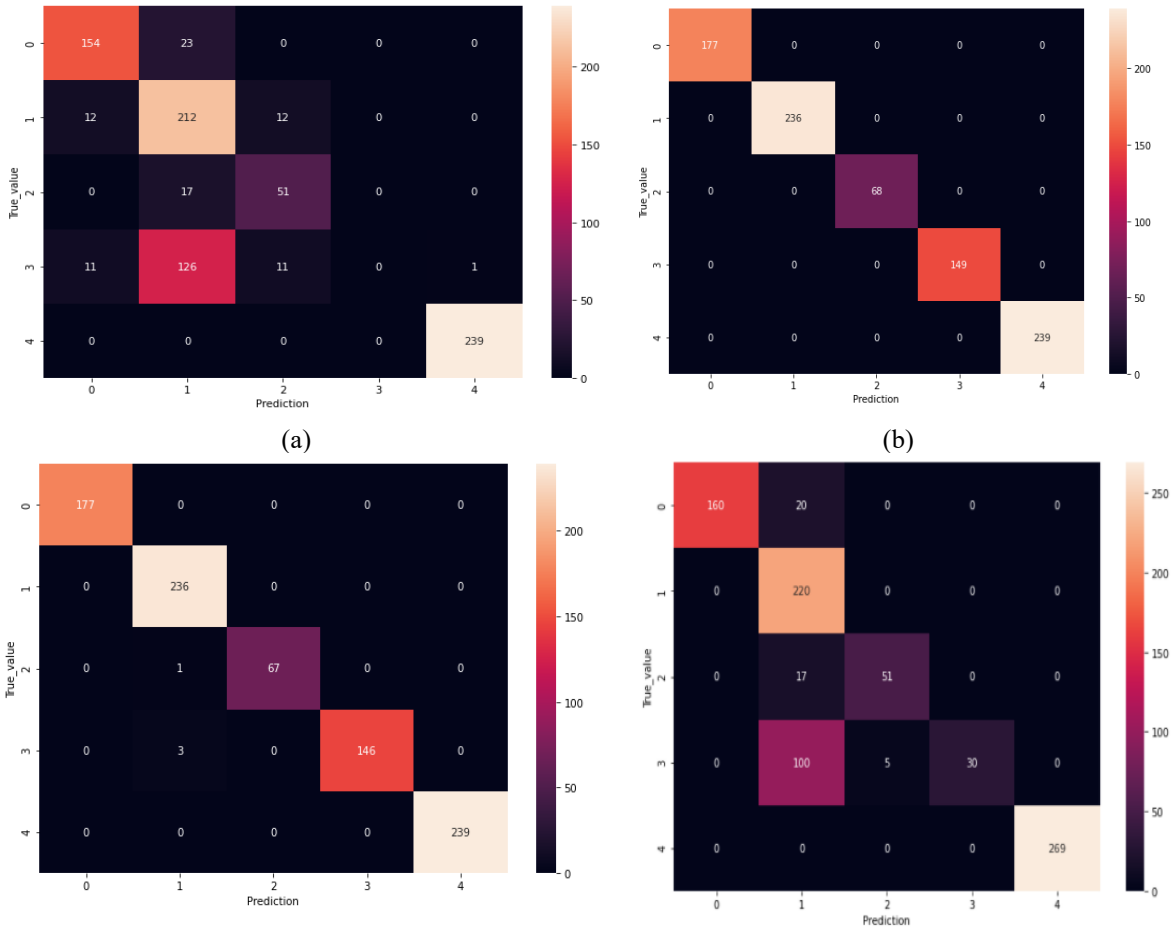


Figure 13. Confusion Matrix of Models (a) RNN (b) GRU (c) LSTM (d) Simple RNN-GRU

5.5 Analysis

In order to determine how good the proposed model work, it is compared with SOTA models such as RNN, GRU, LSTM, and Simple RNN-GRU. Table 3 shows the experimental result obtained in terms of Training Accuracy, Loss, Validation Accuracy and Loss with Precision, recall, and F1-Score under each language of the proposed model. In contrast, Table 4 depicts the result of all SOTA models for each language. Analyzing the results, it is observed that our model shows different training and Validation accuracy for each language as compared to other models. For Bangla language, Training Accuracy of 97.65% with 0.409 loss and Validation accuracy of 96.56% with 0.469 loss is incurred, for Malayalam language, Training Accuracy of 97.65% with 0.209 loss and Validation accuracy of 96.56% with 0.599 loss is achieved, for Odia language, Training Accuracy of 99.65% with 0.209 loss and Validation accuracy of 98.56% with 0.469 loss is incurred, for Telugu language, Training Accuracy of 97.65% with 0.409 loss and Validation accuracy of 94.56% with 0.469 loss is incurred and for Urdu language, Training Accuracy of 99.95% with 0.009 loss and Validation accuracy of 98.95% with 0.059 loss is incurred. Since, the proposed model achieves highest accuracy with minimum loss for all the five languages used in this research, we can say that our model outperforms as compared to other models.

6 CONCLUSION AND FUTURE SCOPE

Recognizing speech in a noisy environment is one of the exciting and challenging problem. This research proposes a hybrid GRU-LSTM model and the comparative analysis of various SOTA models such as RNN, GRU, LSTM, and Simple RNN-GRU used to recognize speech samples in a noisy environment. The data set of each language used in the research has been collected from the Kaggle repository. To train and validate the model, features of speech samples are extracted using various feature extraction techniques. The proposed model is initially tuned at three learning rates, i.e., 0.1,0.001 and 0.0001, where it is analyzed over 0.2, 0.3, and 0.4 dropouts, respectively. It shows 97.02% accuracy at 0.001 learning rate and 0.3 dropout during the validation phase. Thereafter, each model is analyzed using the same learning rate and dropout, where the Hybrid GRU-LSTM model outperforms other models. The proposed model achieves the validation accuracy of 96.56%,96.56%, 98.56%,94.59%, and 98.95% for Bangla, Malayalam, Odia, Telugu, and Urdu, respectively. Despite predicting the speech samples with the highest accuracy, the model has certain limitations. It may require more time to get trained if the training data is huge and complex. Due to the long training time, there is a risk of overfitting the data.

In the near future work, the models shall be trained for other languages (Indian or Non-Indian), and different noise reduction techniques can be applied for smoothing the speech samples. Moreover, other SOTA models can also be used to recognize the speech precisely.

Declaration:

Ethics Approval: No Human subject or animals are involved in the research.

Consent to Participate: All authors have mutually consented to participate.

Consent to Publish: All the authors have consented the Journal to publish this paper.

Data Availability Statement: Authors declare that all the data being used in the design and production cum layout of the manuscript is declared in the manuscript.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] Rojc, M., & Mlakar, I. (2022). *An LSTM-based model for the compression of acoustic inventories for corpus-based text-to-speech synthesis systems*. Computers and Electrical Engineering, 100, 107942.

[2] Chai, L., Du, J., Liu, Q. F., & Lee, C. H. (2020). *A cross-entropy-guided measure (CEGM) for assessing speech recognition performance and optimizing DNN-based speech enhancement*. IEEE/ACM Transactions on audio, speech, and language processing, 29, 106-117.

[3] Kadiri, S. R., Alku, P., & Yegnanarayana, B. (2023). *Analysis of instantaneous frequency components of speech signals for epoch extraction*. Computer Speech & Language, 78, 101443.

[4] Jain, N., Gupta, V., Shubham, S., Madan, A., Chaudhary, A., & Santosh, K. C. (2021). *Understanding cartoon emotion using integrated deep neural network on large dataset.* Neural Computing and Applications, 1-21.

[5] Badhon, S. S. I., Rahaman, M. H., Rupon, F. R., & Abujar, S. (2020, July). *State of art research in Bengali speech recognition*. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[6] Ponraj, A. S. (2020, November). *Speech Recognition with Gender Identification and Speaker Diarization.* In 2020 IEEE International Conference for Innovation in Technology (INOCON) (pp. 1-4). IEEE.

[7] Papakostas, M., Spyrou, E., Giannakopoulos, T., Siantikos, G., Sgouropoulos, D., Mylonas, P., & Makedon, F. (2017). *Deep visual attributes vs. hand-crafted audio features on multidomain speech emotion*

recognition. Computation, 5(2), 26.

[8] San, N., Paraskevopoulos, G., Arora, A., He, X., Kaur, P., Adams, O., & Jurafsky, D. (2024). *Predicting positive transfer for improved low-resource speech recognition using acoustic pseudo-tokens*. arXiv preprint arXiv:2402.02302.

[9] Dharmale, G., Patil, D. D., Ganguly, T., & Shekapure, N. (2024). *Original Research Article Effective speech recognition for healthcare industry using phonetic system*. Journal of Autonomous Intelligence, 7(5).

[10] Dalai, T., Mishra, T. K., & Sa, P. K. (2023). *Part-of-Speech Tagging of Odia Language Using Statistical and Deep Learning Based Approaches*. ACM Transactions on Asian and Low-Resource Language Information Processing, 22(6), 1-24.

[11] Madhavaraj, A., & Ganesan, R. A. (2022). *Data and knowledge-driven approaches for multilingual training to improve the performance of speech recognition systems of Indian languages*. arXiv preprint arXiv:2201.09494.

[12] Sultana, S., Rahman, M. S., & Iqbal, M. Z. (2021). *Recent advancement in speech recognition for bangla: A survey*. Int. J. Adv. Comput. Sci. Appl, 12(3), 546-552.

[13] Noman, A., & Cheng, X. (2023). *Bengali Isolated Speech Recognition Using Artificial Neural Network.* Mechatronics and Automation Technology: Proceedings of ICMAT 2022, 33, 14.

[14] Sharmin, R., Rahut, S. K., & Huq, M. R. (2020). *Bengali spoken digit classification: A deep learning approach using convolutional neural network.* Procedia Computer Science, 171, 1381-1388.

[15] Swain, M., Maji, B., Kabisatpathy, P. & Routray, A. (2022). *A DCRNN-based ensemble classifier for speech emotion recognition in Odia language*. Complex Intell. Syst. 8, 4237–4249.

[16] Rajisha, T. M., Sunija, A. P., & Riyas, K. S. (2016). *Performance analysis of Malayalam language speech emotion recognition system using ANN/SVM.* Procedia Technology, 24, 1097-1104.

[17] Bilal M, Khan A, Jan S, Musa S, Ali S. *Roman Urdu Hate Speech Detection Using Transformer-Based Model for Cyber Security Applications*. Sensors. 2023; 23(8):3909. https://doi.org/10.3390/s23083909

[18] Shivaprasad, S., & Sadanandam, M. (2021). *Dialect recognition from Telugu speech utterances using spectral and prosodic features.* International Journal of Speech Technology, 1-10.

[19] Naeem, S., Iqbal, M., Saqib, M., Saad, M., Raza, M. S., Ali, Z., ... & Arshad, M. U. (2020, December). *Subspace gaussian mixture model for continuous urdu speech recognition using kaldi*. In 2020 14th International Conference on Open Source Systems and Technologies (ICOSST) (pp. 1-7). IEEE.

[20] Moncy, A. M., Athira, M., Jasmin, H., & Rajan, R. (2020, December). *Automatic speech recognition in Malayalam using DNN-based acoustic modelling.* In 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS) (pp. 170-174). IEEE.

[21] Rajan, R., Joseph, A. J., Robin, E. K., & Fathima, N. T. (2020, November). *Part-of-speech tagger in malayalam using bi-directional lstm*. In 2020 23rd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA) (pp. 22-27). IEEE.

[22] Mohanty, P., & Nayak, A. K. (2019). *N-gram language model based continuous voiced Odia digit recognition*. Int. J. Recent Technol. Eng.(IJRTE), 8(2), 4565-4574.

[23] Rakib, O. F., Akter, S., Khan, M. A., Das, A. K., & Habibullah, K. M. (2019, December). *Bangla word prediction and sentence completion using GRU: an extended version of RNN on N-gram language model*. In 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-6). IEEE.

[24] Islam, J., Mubassira, M., Islam, M. R., & Das, A. K. (2019, February). *A speech recognition system for Bengali language using recurrent neural network*. In 2019 IEEE 4th international conference on computer and communication systems (ICCCS) (pp. 73-76). IEEE.

[25] Kumar, S., Kumar, M. A., & Soman, K. P. (2019). *Deep learning based part-of-speech tagging for Malayalam Twitter data* (Special issue: deep learning techniques for natural language processing). Journal of Intelligent Systems, 28(3), 423-435.

[26] Nahid, M. M. H., Purkaystha, B., & Islam, M. S. (2017, December). *Bengali speech recognition: A double layered LSTM-RNN approach*. In 2017 20th international conference of computer and information technology (ICCIT) (pp. 1-6). IEEE.

[27] Kumar, A. P., Roy, R., Rawat, S., & Sudhakaran, P. (2017*). Continuous telugu speech recognition through combined feature extraction by mfcc and dwpd using hmm based dnn techniques*. International Journal of Pure and Applied Mathematics, 114(11), 187-197.

[28] Ali, S. A., Khan, S., Perveen, H., Muzzamil, R., Malik, M., & Khalid, F. (2017). *Urdu language translator using deep neural network*. Indian J. Sci. Technol, 10(40), 1-7.

[29] Ashraf, J., Iqbal, N., Khattak, N. S., & Zaidi, A. M. (2010, March). *Speaker independent Urdu speech recognition using HMM.* In 2010 The 7th International Conference on Informatics and Systems (INFOS) (pp. 1-5). IEEE.

[30] Mannepalli, K., Sastry, P. N., & Suman, M. (2016). *MFCC-GMM based accent recognition system for Telugu speech signals.* International Journal of Speech Technology, 19, 87-93.

[31] Paul, B., Phadikar, S., & Nandi, U. (2026). *Noise-augmented transformer-based automatic speech recognizer using a novel noise distillation system*. Multimedia Tools and Applications, 85(2), 60.
[32] Gupta, S. *A Dataset of Non-Native English Accent to Assist Neural Speech Recognition*. Available [Online]: https://www.kaggle.com/datasets/imsparsh/accentdb-core-extended.
[33] Ali, H. *Urdu Speech Dataset*. Available [Online]: https://www.kaggle.com/hazrat/urdu-speech-dataset.