Engineered Science
DOI: https://dx.doi.org/10.30919/es1180

A New Gate Control Unit-Recurrent Neural Network Structure for Audio-Based Sentiment Analysis

Sunil Thimmaiah,[1, 2,*] and Raghu Jayaramu[1]

Abstract

Sentiment analysis, a crucial task in audio processing, involves the classification of emotions expressed in spoken language. The proposed work is based on a novel Extreme Gradient Boosting (XGBoost)-based structure for emotion-based sentiment analysis from dialect speech samples and the results are compared with traditional techniques to prove its uniqueness in improving the performance. We extract relevant acoustic features from the audio signals, such as Mel-frequency cepstral coefficients (MFCC) coefficients and pitch, and utilize them as input to train an XGBoost classifier. The XGBoost algorithm is an ensemble of decision trees with gradient boosting to learn the sentiment patterns in the audio data. The results demonstrate the effectiveness of the XGBoost model for feature optimization which results in the improvement of the classification of sentiment in audio data. After optimizing the features, the classification of three types of sentiment: positive, negative, and neutral is done by using a novel Recurrent neural network (RNN) structure that incorporates a new Gate Control Unit (GCU) specifically designed for audio-based sentiment analysis, because it has gating mechanisms that regulate the information flow within the RNN, enabling the model to selectively focus on relevant acoustic features and effectively capture sentiment-related patterns in the audio data.

Keywords: Gate Control Unit (GCU); Recurrent Neural Networks (RNNs).
Received: 17 November 2023; Revised: 12 March 2024; Accepted: 08 May 2024.
Article type: Research article.

1. Introduction

Sentiment analysis, also known as emotion recognition, is a fundamental task in audio processing that involves understanding and classifying emotions expressed through speech. It has applications in various fields, including call center analytics, voice assistants, and emotional speech recognition systems. Recurrent Neural Networks (RNNs) have shown great potential in capturing temporal dependencies and have been successfully applied to sentiment analysis tasks. In this study, we propose a novel RNN structure that incorporates a new Gate Control Unit (GCU) specifically designed for audio-based sentiment analysis.

[1] *Department of Electronics & Communication Engineering, The National Institute of Engineering, Mysuru, Visvesveraya Technological University, Belagavi, Karnataka, 590018, India.*
[2] *Department of Electronics & Communication Engineering, Nagarjuna College of Engineering & Technology, Bengaluru, Visvesveraya Technological University, Belagavi, Karnataka, 590018, India.*
**Email:* suneelthimmaiah@gmail.com (S. Thimmaiah)

The GCU is a crucial component of the proposed RNN structure, which enhances the ability of the model to capture sentiment-related patterns in the audio data. By introducing gating mechanisms, the GCU enables the model to selectively attend to relevant acoustic features, focusing on the most informative aspects of the speech signal for sentiment analysis. This gating mechanism effectively regulates the flow of information within the RNN, allowing the model to extract meaningful representations of emotions from the audio data.
In summary, this study introduces a novel RNN structure with a specialized GCU for audio-based sentiment analysis. By incorporating the GCU, we aim to enhance the ability of the model to capture sentiment-related patterns in audio data and thereby improve the accuracy of sentiment classification. The following sections will detail the methodology, experimental setup, and results of our proposed GCU-based RNN structure, highlighting its effectiveness and potential applications in the field of audio-based sentiment analysis.

1.1 Related work

The multimodal sentiment analysis based on an adaptive modality-specific weight fusion network (AdaMoW)[1] focuses on the task of multimodal sentiment analysis and proposes a novel approach that utilizes an adaptive modality-specific weight fusion network. In multimodal sentiment analysis, the goal is to analyze and understand the sentiment or emotions expressed through multiple modalities such as text, images, videos, or audio. By leveraging information from different modalities, the aim is to improve the accuracy and robustness of sentiment analysis. The proposed approach in this paper introduces an adaptive modality-specific weight fusion network to effectively combine information from different modalities. The key idea is to assign adaptive weights to each modality based on their specific relevance and importance for sentiment analysis. To analyze the test-based sentiment, the transformer-based feature fusion approach for multimodal visual Sentiment recognition using tweets in the wild was proposed in Ref. [2] and this work focuses on the task of multimodal visual sentiment recognition using social media data, specifically tweets. The paper proposes a novel approach that utilizes a transformer-based feature fusion technique for this task. To improve the performance of sentiment analysis in different target domains, cross-domain sentiment analysis involves training a sentiment analysis model on a source domain and applying it to predict sentiment in a different target domain was proposed in Ref. [3] where labeled data is scarce or unavailable. Here the authors propose a small in-domain fine-tuning approach for cross-domain sentiment analysis. By leveraging a pre-trained model and fine-tuning it on a smaller labeled dataset specific to the target domain, the proposed approach aims to improve sentiment prediction in different domains where labeled data is limited or unavailable. A model based on bidirectional Long Short-Term Memory (BiLSTM) with an attention mechanism was proposed in Ref. [4] to improve the accuracy of sentiment classification from Chinese text. Sentiment classification involves determining the sentiment expressed in text data, such as comments or reviews, and categorizing them into positive, negative, or neutral sentiments. Chinese mixed text comments refer to comments that contain a mixture of different languages, characters, or dialects. The proposed model utilizes BiLSTM architecture, which is a type of RNN that processes sequential data in both forward and backward directions. The BiLSTM can capture contextual information and dependencies in the text data. In Ref. [5], smart analysis approach that combines linguistic features and transformer-based models to improve sentiment analysis in the economics domain has been proposed. The task of economics sentiment analysis involves understanding and classifying the sentiment expressed in economic texts, such as news articles, financial reports, or social media discussions related to economics. The goal is to determine whether the sentiment expressed is positive, negative, or neutral, and to provide insights into the overall sentiment trends in the economics domain from Spanish text. The cross-lingual sentiment analysis involves analyzing sentiment across multiple languages and understanding the polarity (positive, negative, or neutral) of the expressed sentiments.[6] The challenge arises from the linguistic and cultural differences between languages. The proposed work is likely to discuss the process of employing lexicons and dictionaries in the cross-lingual sentiment analysis task. The lexicons and dictionaries can serve as valuable resources for identifying sentiment-bearing words or phrases in different languages. The adopted methodology involves mapping words or phrases from one language to another using translation techniques or language-specific features. This allows sentiment annotations from one language to be applied to corresponding words or phrases in another language. The proposed methodology utilizes lexicons and dictionaries, which are resources that contain words or phrases annotated with their sentiment polarity. As technology grows the adaption of speech in treating several neurological-related diseases like Alzheimer's. In Ref. [7], the detection of dementia based on speech involves analyzing speech patterns, linguistic features, and other speech-related characteristics to identify potential signs of cognitive decline. The goal is to develop a reliable and non-invasive method for early detection of dementia. The paper likely discusses various types of information that can be extracted from speech, such as acoustic features (*e.g.*, pitch, intensity), linguistic features (*e.g.*, vocabulary richness, grammatical errors), and prosodic features (*e.g.*, speech rate, intonation). Each type of information provides valuable insights into the cognitive abilities of an individual. The proposed approach in Ref. [8] aims to capture both syntactic and lexical semantic information to better understand the sentiment expressed towards different aspects. Syntactic structure information refers to the grammatical relationships and dependencies between words in a sentence, while lexical semantic information refers to the meanings and associations of words or phrases. A multi-task learning framework for hate speech detection,[9] jointly learns hate speech detection and sentiment analysis tasks to improve the performance of hate speech detection.

Hate speech detection involves identifying and categorizing offensive or discriminatory language in text data. Sentiment analysis, on the other hand, focuses on determining the sentiment or emotion expressed in text, such as positive,

negative, or neutral. The proposed multi-task learning approach leverages the relationship between hate speech detection and sentiment analysis by jointly training models for both tasks. By simultaneously learning these related tasks, the model can capture shared representations and dependencies, leading to enhanced performance in hate speech detection.

2. Methodology

The detailed architecture of the proposed methodology is shown in Fig. 1.

The steps involved in emotion recognition is shown in Fig. 1, initially the raw dialect speech is taken and its pre-processed to remove un-wanted or noise signals (if any). Then the samples under through windowing where the speech samples are splitted into frames, later the speech samples are and time domain using VMD technique. The time domain analysis is followed by energy estimation using Teager Energy Operator (TEO) as given in Eq. (1). To study the frequency characteristics of speech the samples are transformed into frequency domain using Discrete Fourier Transform (DFT) technique to perform pitch based and spectrum-based analysis which helps to identify the voiced and un-voiced regions in the sample. Then to filter the redundant part of the samples, the processed samples are passed the triangular filter bank. To normalize the variation of speech samples in feature extraction step logarithmic is applied and it is followed by Mel frequency representation and features are optimized using XGBoost technique. Based on optimized features, the samples classified to match with different emotions using NGCU-RNN technique. The process is explained in deep using suitable equations in the following sections.

(i) Data collection and preprocessing

A diverse dataset of audio recordings containing speech samples was collected expressing different sentiments.[18] In Pre-processing the audio data is pre-processed by applying techniques such as audio normalization, noise removal, and resampling to ensure consistency and quality.

The collected Dataset splits the preprocessed audio data into training and testing sets, ensuring a balanced representation of different sentiment classes in each set. Assign sentiment labels to the audio samples based on the intended sentiment or use annotations if available.

(ii) Feature extraction

Extract relevant acoustic features from the preprocessed audio data. Commonly used features for sentiment analysis include Mel-frequency cepstral coefficients (MFCCs): These capture the spectral characteristics of the speech signal, the fundamental frequency of the speaker's voice is represented by pitch, the loudness of the speech signal is estimated based on energy using TEO technique, the prosodic features like rhythm, intonation, and duration information are extracted.

(iii) Model architecture

Design the proposed RNN structure with a GCU for sentiment analysis includes an Input Layer, which takes the extracted acoustic features as input to the model. GCU Layers, introduce GCU layers to capture sentiment-related patterns in the audio data. The GCU employs gating mechanisms to selectively attend to relevant features. Recurrent Connections, utilize recurrent connections (*e.g.*, LSTM or GRU) to capture temporal dependencies and sequential information in the audio data. Fully Connected Layers are added to layers for further process, the extracted features and enhance the representation

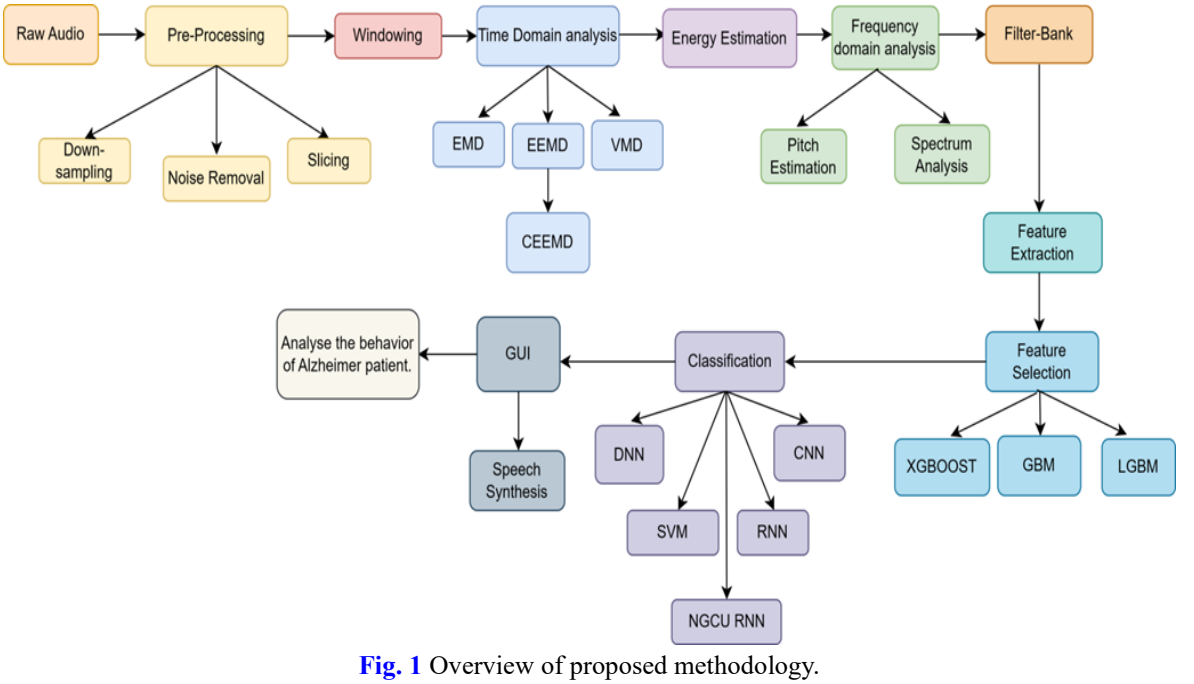


Fig. 1 Overview of proposed methodology.

of model capabilities. The Output Layer consists of a softmax activation function to classify the sentiment into different categories.

(iv) Model training

In this step, the model's parameters and hyperparameters are initialized and later in the second step, GCU-based RNN architecture is trained on the training set using optimization algorithms such as stochastic gradient descent (SGD) or Adam. Then in the third step, Optimize the model by minimizing a suitable loss function, such as categorical cross-entropy, between the predicted and true sentiment labels. In the fourth step, regularize the model to prevent overfitting by applying techniques like dropout or L2 regularization. Finally, experiment with different hyperparameters (*e.g.*, learning rate, batch size) to optimize the model's performance.

(v) Model evaluation

After the training, the Model is evaluated based on accuracy, precision, recall, and F1 score to measure the effectiveness of the model in sentiment classification and also analyze the model's performance across different sentiment classes and examine any potential biases or limitations.

(vi) Comparison and analysis

The evaluated results are compared with existing sentiment analysis models and baselines. Conduct statistical tests (*e.g.*, t-tests) to determine the significance of performance differences. Perform an error analysis to identify common misclassifications and gain insights into areas of improvement.

(vii) Generalization and deployment

Validate the trained model on unseen audio data to ensure its generalization capabilities. Deploy the sentiment analysis model in real-world applications, such as sentiment monitoring in customer support systems or emotion recognition in voice assistants.

(viii). Pre-processing

The raw audio is pre-processed to normalize the frequency variations,[10] identification of voiced and un-voiced regions, segment lengthier speech into frames of smaller lengths 10 ms with a shift of 5 ms, and filter out the discontinuity at the ends of samples windowing is applied that to a hamming window function is used. Pre-processing includes normalization, voiced region detection, framing, and windowing. In the pre-processing step, another important part is the estimation of the energy of the samples, In this proposed work energy of each frame is estimated by using the Teager Energy Operator as given in Eq. (1) [11]

$$\tau = \Psi_c\big(S(n)\big) = S^2(n) - S(n+1)\cdot S(n-1) \quad (1)$$

where, $\Psi_c(s(n))$- is Cross energy operator of speech samples S(n).

The Teager energy values lead to a better representation of energy features for further process.

In the Variational Mode De-composition (VMD) technique, the features of one sequence will not depend on any other features in the sequence hence it is known as a non-recursive technique and it decomposes the infinite real-time signal into sub-signals of finite length. This unique characteristic, makes the VMD technique to overcome the error correction backward path. The decomposed signals of VMD are bagged around its center frequency and the bandwidth of the VMD signal is calculated based on the following parameters:

Hilbert transform is used to estimate the frequency of half sided spectrum and it is obtained by using Eq. (2).

$$\hat{S}(K) = \begin{cases} -jS(k), K = 1 \ldots \ldots \frac{N}{2} - 1 \\ jS(k), K = \frac{N}{2} + 1 \ldots \ldots N - 1 \end{cases} \quad (2)$$

(ii) The frequency shifting $\left(e^{-j\omega kn}\right)$ property of modulation is adopted to shift the frequency of each sub-band to its relevant base-bands.

(iii) By using the H-1 Gaussian technique is used to compute the bandwidth of each decomposed signal of sub-bands.

Finally, the VMD for the signal is formulated as:

The combination of Hilbert transforms and filters, gives the set of important characteristic components $a_k(k = 0,1,2 \ldots\ldots K)$within a limited range of bandwidth. The speech signal S(t) is decomposed into N number of sub-signals. Since, speech is a non-stationary signal the amplitude and frequency of the speech will vary from maximum to minimum values, to control such abrupt variations Lagrange expression is introduced and it is widely used for constrained optimization problems. The Lagrange expression is obtained by Eq. (3):[11]

$$L(\{a_k\},\{\omega_k\}) = \beta \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{i}{\pi t}\right) * a_k(t) \right] e^{-i\omega_k t} \right\|_2^2 + \|f(t) - \sum_k a_k(t)\|_2^2 + \langle \lambda(t), s(t) - \sum_k a_k(t) \rangle \quad (3)$$

where, s(t)-input signal, $a_k(t)$-a subcomponent of the input, β-penalty factor, λ-Lagrange factor

When, $\sum_k a_k(t) = s(t)$, then the simplified equation is as given in Eq. (4)

$$\min_{\{a_k\},\{\omega_k\}} \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{i}{\pi t}\right) * a_k(t) \right] e^{-i\omega_k t} \right\|_2^2 \right\} \quad (4)$$

Algorithm 1: VMD calculation

Input: Signal x, Number of modes K, Regularization parameter lambda, Convergence threshold tol, Max iterations MaxIter

Step 1: Initialization

- Compute the empirical mode decomposition of x using any suitable method such as EMD.
- Initialize the mixing matrix W and the centers u of the Gaussian kernels randomly or using some predefined rule.
- Set the counter i = 0 and the difference variable diff = 1.

Step 2: VMD Algorithm
- While (i < MaxIter) and (diff > tol):
- Update the Fourier coefficients by solving the following optimization problem:
$minimize\ ||X - WH||\_2^{\ 2} + lambda\ ||H||\_1$
$subject\ to\ ||S\_k||\_2 = 1\ for\ k = 1, 2, \ldots, K$
where X is the Fourier transform of x, and S is the matrix of Fourier coefficients of the K modes.
- Update the mixing matrix W and the centers u of the Gaussian kernels by solving the following optimization problem:
$minimize\ ||X - W\ S||\_2^2 + lambda\ ||W||\_1$
where, the regularization term encourages sparsity in the mixing matrix.
- Compute the difference between the current and previous solutions:
$diff = ||S - S\_{prev}||\_F + ||W - W\_{prev}||\_F$
- Update the counter: i = i + 1.
Step 3: Output
- Return the K modes as the inverse Fourier transform of the Fourier coefficients S.

2.1 Signal features

The signal features like mean, standard deviation, entropy, and kurtosis are analyzed for the classification of sentiment speech as mentioned in Fig. 1.

2.1.1 Information and entropy

In general, information is a message carried from source to destination. The term information in speech recognition indicates the level of uncertainty in its probability and represents the occurrence of each syllable in the speech sample and the logarithmic of each probability of sample gives information.[12]

Let us consider the number of random variables in the sample $(S_1, S_2 \ldots\ldots\ldots S_n)$. The information carried by each random variable is obtained from Eq. (5):

$$\mathrm{I_i} = \log\left(\frac{1}{\mathrm{P(S_i)}}\right) \quad \mathrm{i} = 1 \text{ to } \mathrm{N} - 1 \tag{5}$$

The average information carried by each random variable is known as Entropy. It is calculated using Eq. (6):

$$\mathrm{E_i} = \sum_{i=1}^{N-1} \mathrm{P(S_i)} \cdot \log\left(\frac{1}{\mathrm{P(S_i)}}\right) \tag{6}$$

If the value of information and entropy is high then the probability of random variable is less and vice versa., This helps to analyze the complexity of the signals hence it is used widely in the analysis of real-time signals like medical signals, and fault analysis.

2.1.2 Mean and standard deviation

The peak value of the signal at the middle of the frequency of the signal gives the value of Mean and standard deviation (SD) indicates the starting and ending point of the signal.
Mean of the signal is estimated using Eq. (7):

$$\mu = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=0}^{\mathrm{N}-1} \mathrm{S_i} \tag{7}$$

Concerning the mean of the signal, the standard deviation is calculated using Eq. (8):

$$\mathrm{SD} = \frac{1}{\mathrm{N}-1} \sum_{\mathrm{i}=0}^{\mathrm{N}-1} (\mathrm{S_i} - \mu)^2 \tag{8}$$

By (8), SD is the square of the sum of the difference of speech random variable and mean and one major difference between μ and SD is that μ is calculated based on the power of the signal, and SD is calculated based on the amplitude of the speech signal.

2.2 Feature extraction

After estimating the related information related to time series, the features of sentiment speech are extracted using the most widely used Mel-Frequency Cepstral Coefficient (MFCC). The MFCC steps include Pre-processing, framing, windowing, FFT, Energy estimation, Mel-scale operation, and transforming back to time domain Discrete Cosine Transform (DCT).

In the VMD technique, the frames of the sample are decomposed based on the center frequency and peak amplitude, and these values are considered for the analysis of features in the frequency domain.[13] The center frequency and peak amplitude are calculated by using Eqns. (9) and (10), respectively.

$$\omega_i^{m+1} = \frac{\int_{i=0}^{\infty} \omega |S_i(\omega)|^2 d\omega}{\int_{i=0}^{\infty} |S_i(\omega)|^2 d\omega} \tag{9}$$

$$S_i^{m+1}(\omega) = \frac{S_i(\omega) - \sum_{k \neq i} S_k(\omega) + \frac{\mu(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k)} \tag{10}$$

As an initial step, the desired parameters like modes (i), center frequency (ω), and balancing parameter (α) are reset. In general, the center frequency-based analysis depends majorly on the number of mode values I, in this proposed work the mode values are i=1,3,5,7,9, and maximum performance is obtained with i = 7.[13]

The pre-processing, framing, windowing, and energy estimation using TEO are explained in Section 2.1. To estimate the frequency-related parameters like the pitch of speech, Fast Fourier Transform (FFT) is applied and normalizes the variation of speech in the frequency domain the processed signals are passed through the Mel-scale the output of the scale is estimated using Eq. (11):

$$f_{Mel} = 2595\log_{10}\left(\frac{f}{100}+1\right) \quad (11)$$

2.3 Feature optimization

Consider a 4 net of features of signal where the first set contains 2 positive features (oval shape) on one side and 5 negative features (Star shape) other side. These features are separated by a split line but due to some errors, 3 positive features get mixed up with negative features as shown in Fig. 2(a). Likewise in the second set of features, there are 5 positive features and 2 negative features which are separated by a split line, but again due to some error, 3 negative features get mixed up with 5 positive class features as shown in Fig. 2(b). The third set separates the positive features and negative features as shown in Fig. 2(c). In the XG boosting technique, the weighted classifiers are added to its results as shown in Fig. 2(d).

Let the dataset d= $\{a_m, b_m\}$, where m =1,2,3……, N. The model is learned or trained using P trees. The outcome of a model ($\hat{a}_j$) is expressed as shown in Eq. (12).[16,17]

$$\hat{a}_j = \emptyset(b_j) = \sum_{Q=1}^{Q} F_Q(b_j), F_Q \epsilon J \quad (12)$$

where, J indicates the hypothesis space regression tree signified as p(b) and it is obtained using Eq. (13)

$$J = \{F(y) = v_{p(b)}\} \quad (13)$$

with,

v representing the leaf node and

F(y) represents the leaf node function of the both samples.

The output of n^{th} iteration is determined using Eq. (14):

$$\hat{a}_j{}^n = \hat{a}_j{}^{n-1} + F_n(b_j) \quad (14)$$

The fitness function-based mathematical expression is formulated using Eq. (15):

$$S(F_n) = \sum_{j=1}^{N} L(a_j, \hat{a}_j{}^{n-1}) + F_n(y_j) + \Delta(F_n) \quad (15)$$

where,

L is the loss function, $\Delta(F_n)$ is the complexity of a model that contains a score and it is calculated by using Eq. (16):

$$\Delta(F_n) = \lambda . R_n + \gamma \, {}^1/_2 \sum_{i=1}^{R} v_i^2 \quad (16)$$

with,

R represents the leaf node

Then the following formula is simplified using 2nd order Tailor series as shown in Eq. (17)

$$G(F_n) = \sum_{j=1}^{N}[v(a_j, \hat{a}_j{}^{n-1}) + b_j F_n(y_j) + {}^1/_2 \, r_j F_n^2(y_j) + \Delta(F_n) \quad (17)$$

where, b_j and r_j indicate the first and second-order derivatives of the loss function and they are given in Eqs.(18) and (19) respectively:

$$b_j = \frac{\partial v(a_j, \hat{a}_j{}^{n-1})}{\partial \hat{x}_j{}^{n-1}} \quad (18)$$

$$r_j = \frac{\partial^2 v(a_j, \hat{a}_j{}^{n-1})}{\partial \hat{a}_j{}^{n-1}} \quad (19)$$

From the above equations, the fitness function is defined as given in Eq. (20):

$$G(F_n) = \sum_{j=1}^{N}[b_j v_p(b_j) + {}^1/_2 \, r_j v_p^2(b_j) + \lambda R + \gamma \, {}^1/_2 \sum_{j=1}^{R} v_i^2 \quad (20)$$

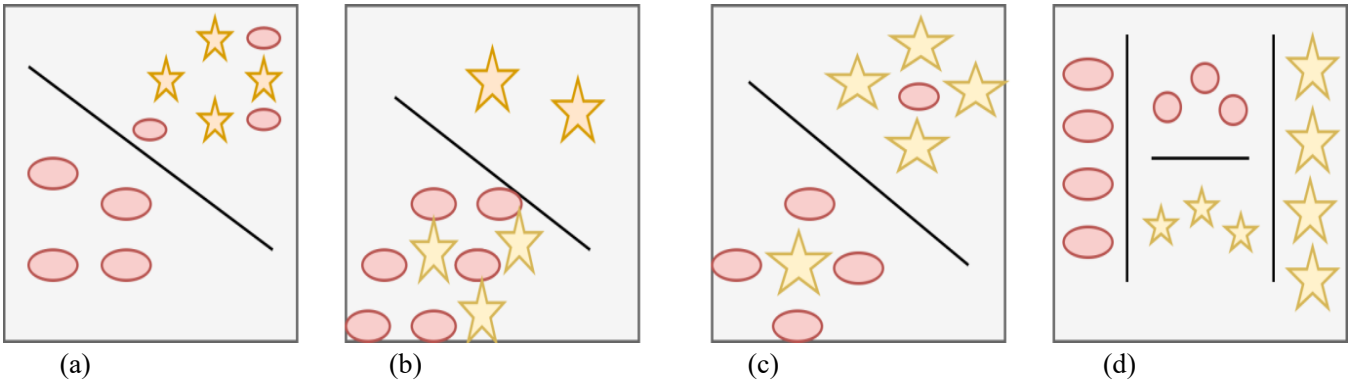
where, $v_p(y_j)$ represents the training instant set in j[th] leaf.

The appropriate leaf provided the structure of a desired tree v_i^* and it is mentioned in Eq. (21):

$$v_i^* = \frac{-\sum_{j \epsilon K_i} b_j}{-\sum_{j \epsilon K_j} r_j + \gamma} \quad (21)$$

The features of the fitness function which are obtained from eqn. (20) are optimized and the simplified fitness function is defined as mentioned in Eq. (22):

$$G(F_n) = -1/2 \sum_{i=1}^{R} \frac{\left(\sum_{j \epsilon K_i} b_j\right)^2}{\sum_{j \epsilon K_i} r_j + \gamma} + \lambda R \quad (22)$$



(a) (b) (c) (d)

Fig. 2 Working illustration of XGBOOST (a) with minimum positive charges below the threshold (b) with minimum negative charges above the threshold (c) Equal deposition of charges (d) Distribution of positive and negative charges below the threshold, above a threshold and in between the threshold.

2.4 Classification

The time-series signals are dependent signals and the dependency will be carried to the frame or data of the signal. So, while classifying the dependent signals the contextual information should not vanish. Thus, for the classification of this kind of long dependency signal, RNN is more suitable, but the major disadvantage of RNN is, that the gradient of a present or previous frame is overlapped by next upcoming frames. To overcome this issue much research was carried out and finally, the researchers came up with a Long-Short Term Memory (LSTM) based RNN that operates on the Gates principle. The basic structure of the LSTM-based RNN is shown in Fig. 3. In a neural network, the sigmoid function transforms any real value signal between '0' and '1', the activation (tanh) will decide that the value of a neuron or a cell is required for the computation of output.

In basic structure, the complete operation of a network including memorizing the data and updating or deleting of data is controlled by three gates namely input which receives the features as input for classification, forget gate- which stores the desired data or feature and forget the unwanted data, output gate gives the output of NN.[4] The mathematical expression of LSTM is defined from Eqn. (23.a) to (23.f):

$$f_t = \sigma(W_f[h_{t-1}, S_t] + S) \quad (23a)$$

$$i_t = \sigma(W_i\,[h_{t-1}, S_t] + b_i) \quad (23b)$$

$$\tilde{C}_t = \tanh(W_C\,[h_{t-1}, S_t] + b_C) \quad (23c)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (23d)$$

$$o_t = \sigma(W_o\,[h_{t-1}, S_t] + b_o) \quad (24e)$$

$$h_t = o_t * \tanh(C_t) \quad (24f)$$

In (23. a), the input vector is represented by S_t At the 't' steps, each input vector is multiplied with weight vectors represented by W_f, the multiplication is linear at 't-1' steps indicated by h_{t-1} and after multiplying all values the forget gate f_t decides which values to be retained and which have to be forgotten based on '1' and '0' respectively. Accordingly, in (23. b), the input gate i_t decides which input values need to get bias by b_i after multiplying the input vector linearly with weight vectors. The movement of the vector's current state to the next state is indicated by $\tilde{C}_t$ and C_t respectively without vanishing the gradients of present and previous frames with the help of 'tanh.' The output gate operation is represented by o_t in (23. e) selects the value or data that has to move from the present to the next state and the size of the data will be selected based on the value of h_t show in (23. f).

Though the LTSM structure is efficient in maintaining the dependency between each frame, the training and testing of all three gates in every instance is a time-consuming process. So, the researchers have decided to reduce the number of gates from three to two and named the new RNN as Gated Recurrent Unit (GRU). The basic structure of GRU is shown in Fig. 4, In this structure the input gate and output gate of LSTM are merged and named as Update gate.[15] So, the GRU structure has two gates namely the Reset gate and Update gate. The modeling of these two gates is illustrated in (24) and (25) respectively.

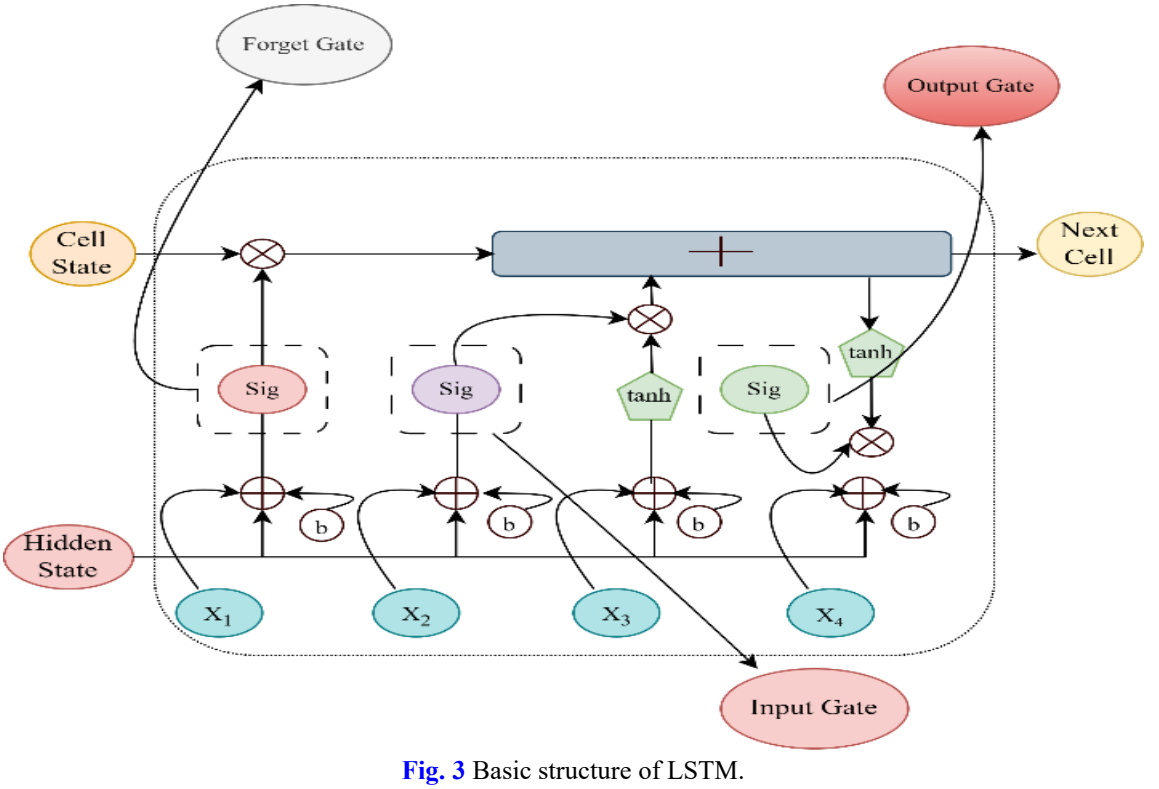


Fig. 3 Basic structure of LSTM.

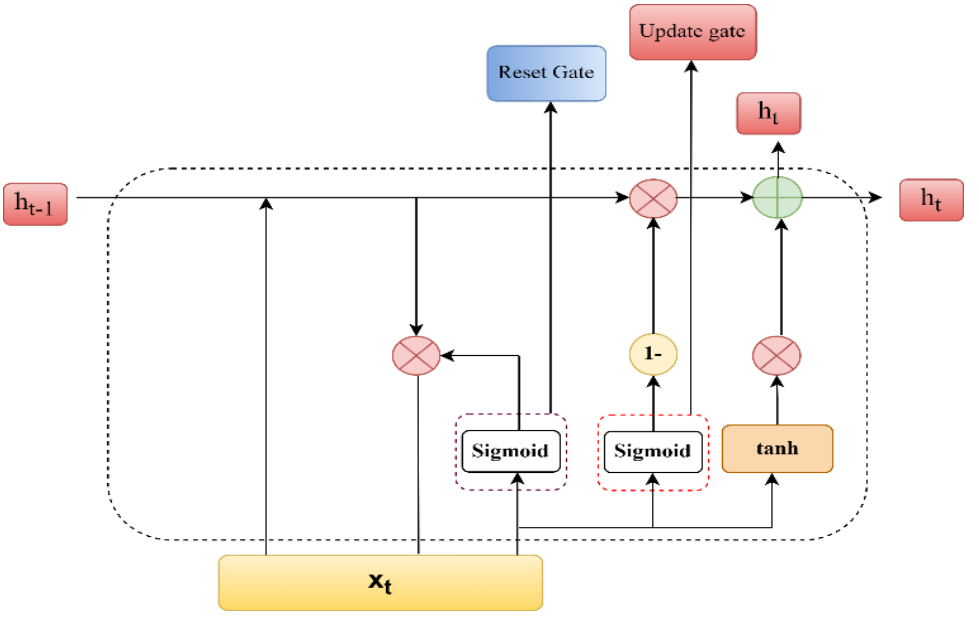


Fig. 4 Gated recurrent unit structure.

The memorization and retention of data in the network is controlled by the Update gate (U_t) and it is as mentioned in Eq. (24):

$$U_t = \sigma(W_{Uh}h_{t-1} + W_{Ux}X_t + b_U) \quad (24)$$

In updating the value of the previous frame vector 'h_{t-1}' and input vector 'X_t'is multiplied linearly with their respective weights 'W_{Uh}' and 'W_{Ux}' and finally added with the suitable bias values 'b_U' of update gate.

The deletion of remaining or redundant data after storing by the update gate is controlled by the Reset gate and it is as given in Eq. (25):

$$R_t = \sigma(W_{Rh}h_{t-1} + W_{Rx}X_t + b_R) \quad (25)$$

In updating the value of the previous frame vector 'h_{t-1}' and input vector 'X_t' is multiplied linearly with their respective weights 'W_{Rh}' and 'W_{Rx}' and finally added with the suitable bias values 'b_R' of reset gate.

The new transformed data or signal is generated from GRU '$\hat{h}_t$' using the activation function 'tanh.' as shown in Eq. (26):

$$\hat{h}_t = \tanh(W_{xh}x_t + R_tW_{hh}h_{t-1}) \quad (26)$$

In (26), the weight matrices of the input vector 'W_{xh}' is multiplied linearly with input vector 'X_t'and it is added with the product of the output of reset gate 'R_t', weight matrices of the previous state 'W_{hh}' and previous state vector 'h_{t-1}'.

Finally, the output of GRU is defined in Eq. (27):

$$h_t = U_th_{t-1} + (1 - U_t)\hat{h}_t \quad (27)$$

In Eq. (27), the present memory is represented by 'U_th_{t-1}' is added to the previous memory state $(1 - U_t)\hat{h}_t$. The GRU is an improved or optimized form of LSTM-based RNN, but the main drawback of this system is slow convergence and less efficient. So, in the proposed an improved version of RNN is used to increase the classification efficiency of Sentiment speech which is a New Gate Control Unit (NGCU) based on RNN.[15-19] The structure of NGCU is shown in Fig. 5.

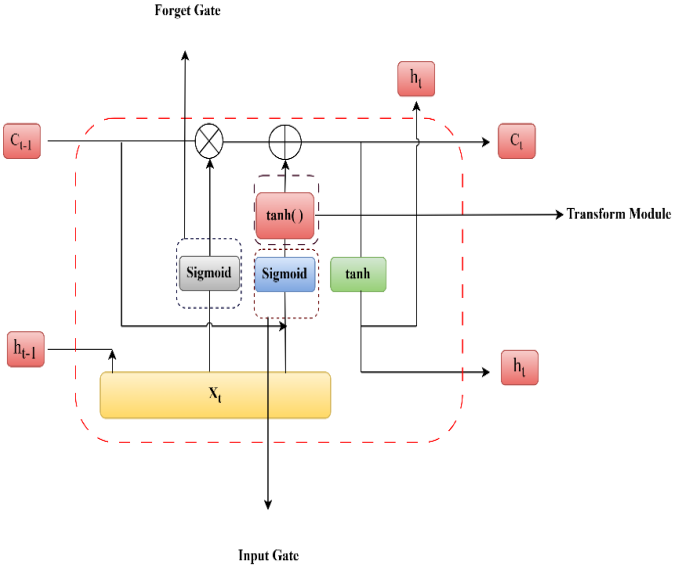


Fig. 5 NGCU structure.

NGCU RNNs aim to address the limitations of traditional RNNs by incorporating additional gate control mechanisms and units. The introduction of NGCUs enhances the ability of the model to handle long-term dependencies and preserve context. NGCU RNNs utilize novel gate control mechanisms that go beyond traditional sigmoid activations, resulting in improved memory representation and information integration. By addressing the gradient vanishing or exploding problem more effectively, NGCU RNNs provide a more stable and robust training process. This leads to better performance on tasks that require capturing and retaining long-term

dependencies, such as machine translation, speech recognition, sentiment analysis, and other sequential data tasks.

The NGCU structure is included with the transform module (tanh (C_t)) to improve the sensitivity of the RNN structure for the classification of multiple classes in the signal along with this module it has two gates namely: input and forget gate.[15]

In this structure, the role of the forget gate is to calculate the amount of desired data that needs to be retained from the previous state for the present state of the cell. The forget gate is loaded with the input vector 'X_t' and the output of forget gate is taken completely with the data of the previous state 'C_{t-1}' for further process through the input gate. It is given in Eq. (28):

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}X_t + b_f) \quad (28)$$

In (28), the weight vectors 'W_{fh}'multiplied linearly with 'h_{t-1}' at t-1 steps and stores the result, the input vector 'X_t' multiplied linearly with its weight vector 'W_{fx}'. The results of two products are added with the bias values and results of forgetting 'f_t'stored in the form '0' and '1' using sigmoid function.

The input of NGCU fetches more part of the 'C_{t-1}' In the whole data frame, the main role of the input gate is to calculate how much data needs to be retained to produce the desired output for the next state. It is given in Eq. (29):

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}X_t + C_{t-1} + b_i) \quad (29)$$

In (29), the weight vectors 'W_{ih}'multiplied linearly with 'h_{t-1}' at t-1 steps and stores the result, the input vector 'X_t' multiplied linearly with its weight vector 'W_{ix}'. The results of the two products are added to the 'C_{t-1}' Data and bias values and results of forget 'i_t'stored in the form '0' and '1' using sigmoid function.

In machine learning, the sigmoid and tanh are two activation functions used to analyze the non-linearity in the signal. The mathematical definition of sigmoid and tanh function is defined in Eqs. (30.a) and (30.b), respectively:

$$\sigma = \frac{1}{1+a^{-x}} \quad (30a)$$

$$\tanh(x) = \frac{2}{1+a^{-2x}} \quad (30b)$$

The only difference between these two activation functions is, that σ varies from '0' to '1' as $\tanh(x)$ varies between $\pm j$ ($j = 1,2,3 \ldots\ldots\ldots, N$). So, when the value 'x' is in the range: $-j < x < j$ then 'σ' enters the supersaturated zone where the variation of input is limited significantly resulting in a decrease in the sensitivity of learning by input gate. In NGCU, it is called an anti-oversaturation conversion represented by a Transform module. It is defined Eq. (31):

$$tr = \tanh(i_t) \quad (31)$$

where, 'tr' indicates the Transform module. As shown in Fig. 5, the value of 'tr' is added with the product of the output of the forget gate 'f_t' and previous state value 'C_{t-1}' results in generating of the value of present state 'C_t'. The value 'C_t' gives information about how much data can be memorized for the next upcoming state. It is defined mathematically as given in Eq. (32)

$$C_t = tr + C_{t-1} \times f_t \quad (32)$$

Finally, the output of the present state is given by Eq. (33):

$$h_t = \tanh(C_t) \quad (33)$$

3. Results and Discussion

In the time domain, speech can be represented as a waveform as shown in Figs. 6(a) and (b). The waveform represents the changes in air pressure over time that occur when we speak. It is a continuous signal that captures the variations in vocal cord vibrations and the resulting sound waves. When speech is represented in the time domain, the x-axis typically represents time, while the y-axis represents the amplitude or intensity of the sound wave. As we speak, the waveform fluctuates, showing the patterns of speech sounds, including vowels, consonants, and pauses. The time-domain representation of speech allows us to analyse various aspects of the signal, such as the duration of phonemes, the presence of pauses, and the overall rhythm and cadence of speech. By examining the waveform, we can observe the different characteristics that makeup spoken language.

A spectrogram is a visual representation of the frequencies present in a speech signal over time as shown in Figs. 7(a) and (c). It provides valuable insights into the spectral content of the speech signal. While spectrograms typically display the magnitude of the frequency components, they can also be represented in decibels (dB) to provide a logarithmic scale of the signal power or intensity. In a decibel spectrogram of speech, the x-axis represents time, the y-axis represents frequency, and the intensity of the colors or shading represents the power or intensity of the signal in decibels. Darker or brighter areas indicate higher power or intensity at specific frequencies and time intervals. By representing the spectrogram in decibels, we can visualize the relative loudness or amplitude of different frequency components within the speech signal. This allows us to observe the energy distribution across different frequencies and track changes in the spectral content over time. Decibel spectrograms are commonly used in various speech and audio processing applications. They help in analyzing and understanding speech sounds, identifying phonetic features, detecting spectral patterns, and diagnosing speech-related issues such as

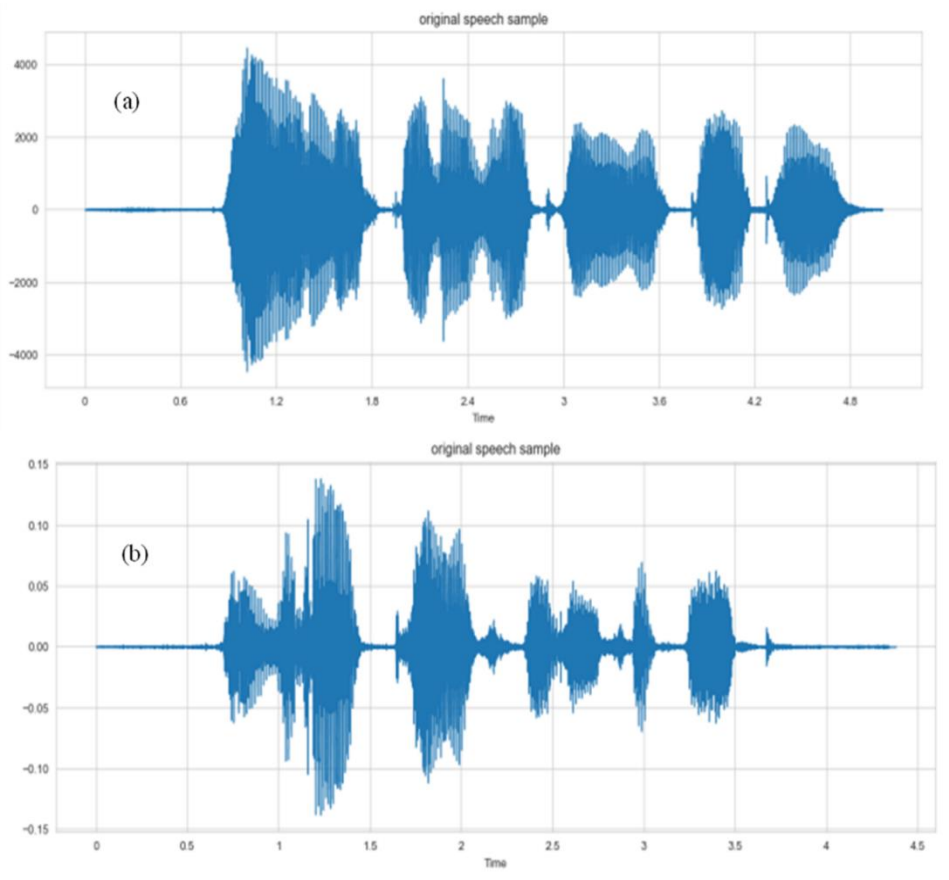


Fig. 6 Time domain representation of speech. (a) Speech sample 1 (b) Speech sample 2.

articulation disorders or vocal abnormalities.

A Mel-spectrogram is a specific type of spectrogram that utilizes the Mel scale to represent frequencies, which corresponds more closely to how humans perceive sound as shown in Figs. 7(b) and (d). When represented in decibels (dB), a Mel-spectrogram provides a logarithmic scale of the power or intensity of the speech signal in Mel-frequency bins.

To generate a Mel-spectrogram in decibels, the speech

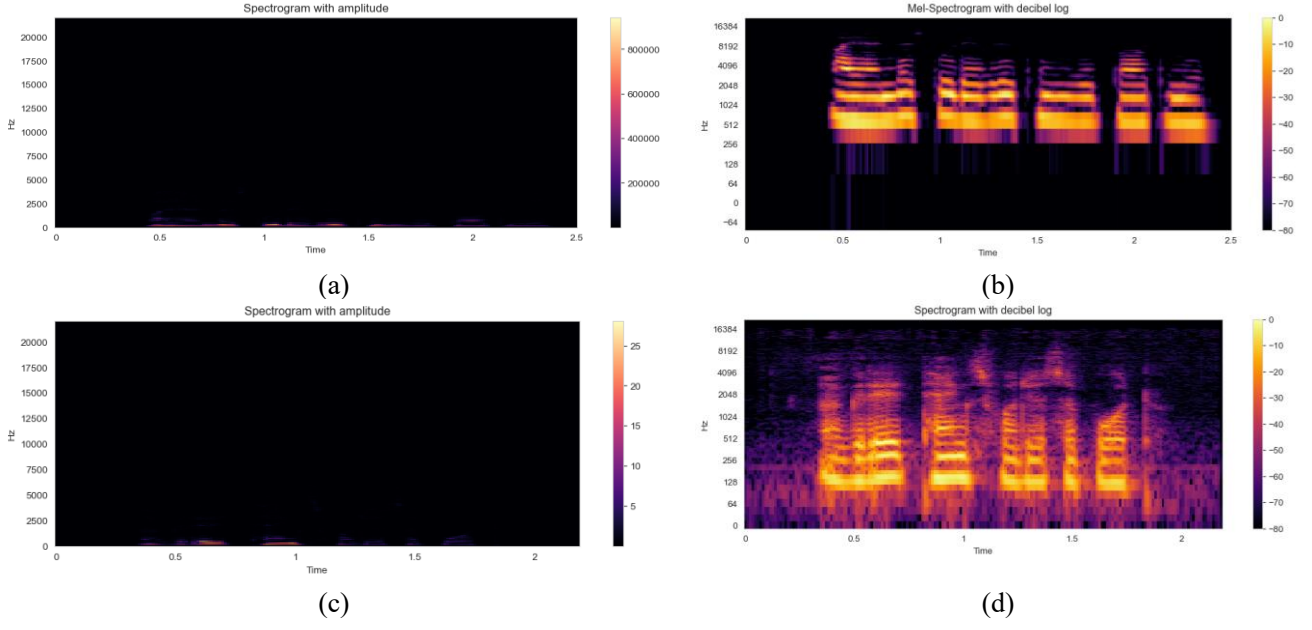


Fig. 7 Spectrogram of speech. (a) Spectrogram of speech sample 1 (b) Mel Spectrogram of speech sample 1 (c) Spectrogram of speech sample 2 (d) Mel Spectrogram of speech sample 2.

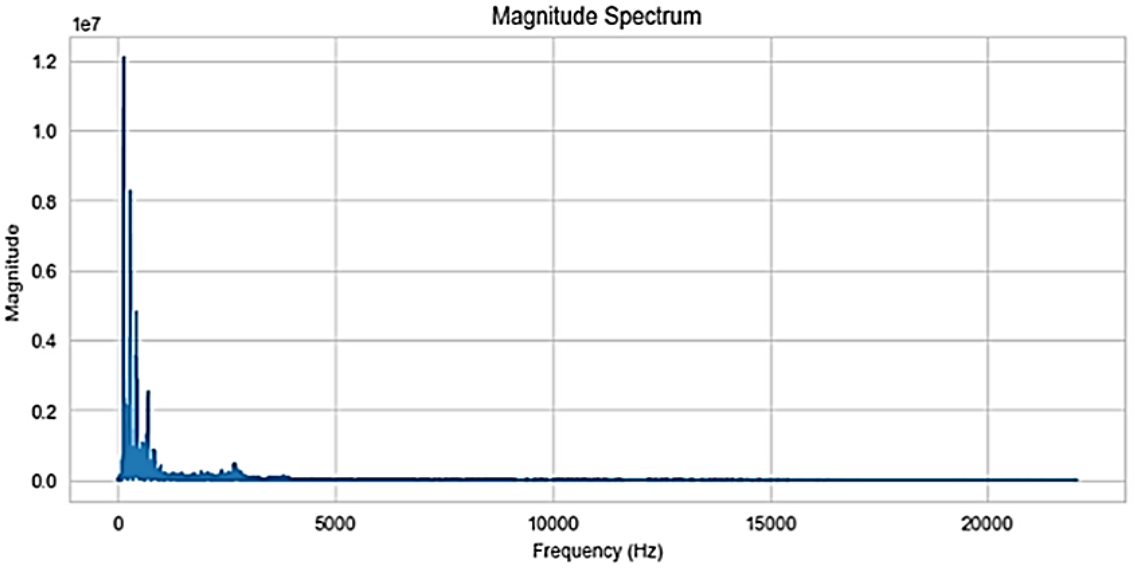


Fig. 8 Magnitude Spectrum of the speech sample.

signal is first divided into short overlapping frames. Each frame is then transformed from the time domain to the frequency domain using techniques such as the FFT.

To generate a spectrogram in decibels, the speech signal is first divided into short overlapping frames. Each frame is then transformed from the time domain to the frequency domain using techniques such as the FFT. The power or intensity of the frequency components is calculated, and the resulting spectrogram is displayed with a logarithmic scale in decibels. The magnitude of the speech sample is shown in Fig. 8.

Next, the power or magnitude spectrum of each frame is calculated which is shown in Fig. 9. The Mel scale is then applied to convert the linear frequency scale into Mel frequencies, which are perceptually more uniform. This conversion is achieved using a set of filter banks that approximate the human auditory system's response to different frequencies. The filter banks are triangular-shaped and spaced according to the Mel scale.

The autocorrelation technique and the pitch contour technique are two common methods used for pitch estimation in speech processing. Here's a comparison between the two techniques:

3.1 Autocorrelation technique

The autocorrelation technique estimates the pitch by finding the periodicity in the speech signal through autocorrelation analysis. It calculates the autocorrelation function of a speech frame and identifies the lag at which the maximum autocorrelation occurs, representing the pitch period. The pitch frequency is then derived by converting the pitch period to Hz using the sampling rate. The autocorrelation technique is relatively simple and computationally efficient. However, it can be sensitive to noise and other factors that affect the accuracy of autocorrelation peaks, leading to potential errors in pitch estimation.

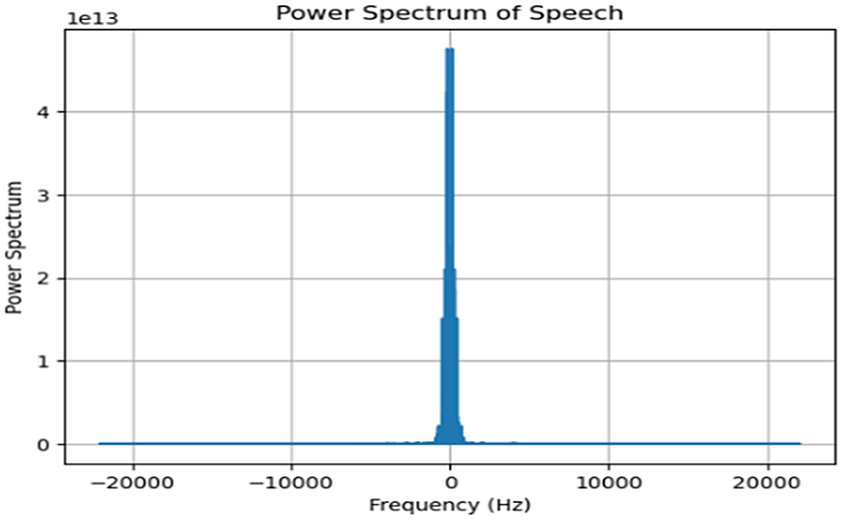


Fig. 9 Power spectrum of the speech sample.

3.2 Pitch contour technique

The pitch contour technique estimates the pitch by analyzing the pitch-related characteristics of the speech signal across multiple frames. It divides the speech signal into overlapping frames and performs pitch estimation for each frame using various algorithms or methods. The estimated pitch values for each frame are then combined to create a continuous pitch contour or representation of the pitch variations over time. The pitch contour technique allows for more robust pitch estimation by considering temporal patterns and smoothing effects across frames. It is often based on advanced algorithms such as cepstral analysis, harmonic product spectrum, or machine learning models. The pitch contour technique can provide more accurate pitch estimation results, especially in the presence of noise, variability, and complex speech signals. The difference between the two techniques can be observed in Fig. 10, where the Fig. 10(a) represents the estimated pitch frequency using the auto-correlation technique is 140.445 Hz whereas Fig. 10(b) gives the estimated pitch frequency using the pitch-contour technique is 3897.42 Hz

In the context of machine learning and deep learning models, loss, accuracy, and validation accuracy are commonly used metrics to evaluate the performance of a model during training and testing. Here's an explanation of each metric:

(i) Loss

The loss represents the error or discrepancy between the predicted output of the model and the actual ground truth labels. It is a numerical value that quantifies how well the model is performing in terms of its ability to minimize the error during training. The goal of training a model is to minimize the loss by adjusting the model's parameters through techniques like gradient descent. Different types of loss functions are used depending on the nature of the problem, such as mean squared error (MSE) for regression tasks or categorical cross-entropy for classification tasks. Lower values of the loss indicate better performance, with zero being the ideal value (perfect prediction).

(ii) Accuracy

Accuracy is a performance metric used to evaluate classification models. It represents the percentage of correctly predicted labels out of the total number of samples in the dataset. It is calculated by dividing the number of correct predictions by the total number of predictions. Accuracy is useful for balanced datasets where the classes are approximately equally represented. However, accuracy can be misleading in cases where the dataset is imbalanced or the cost of misclassifications varies across classes.

(iii) Validation accuracy

During training, a model is typically evaluated on a separate validation dataset to monitor its generalization performance. Validation accuracy is the accuracy achieved by the model on this validation dataset. It helps to assess how well the model is performing on unseen data and can indicate if the model is overfitting (performing well on the training data but poorly on new data) or underfitting (not capturing the underlying patterns in the data). Validation accuracy is calculated using eqn.(34) is a crucial metric for model selection, hyperparameter tuning, and detecting issues like overfitting or convergence problems.

The analysis shown in Fig. 11 and Fig. 12, says that the accuracy of classifying sentiment from dialect speech is better in the proposed techniques. Also, the performance is validated in validation accuracy calculation using the proposed technique helps to classify the over-fitted data.

$$\text{Accuarcy} = \frac{\text{Preciction of True positive+Preciction of True Neagtive}}{\text{All Types of prediction(True positive+True Negative+False Positive+False Negative)}} \quad (34)$$

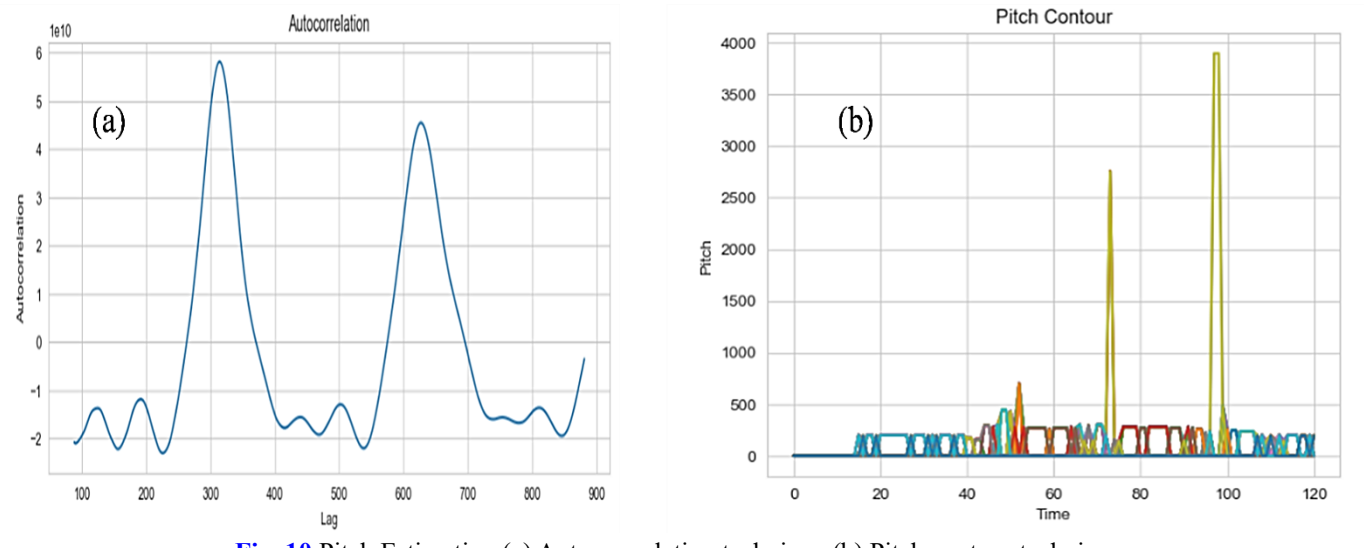


Fig. 10 Pitch Estimation (a) Auto-correlation technique (b) Pitch contour technique.

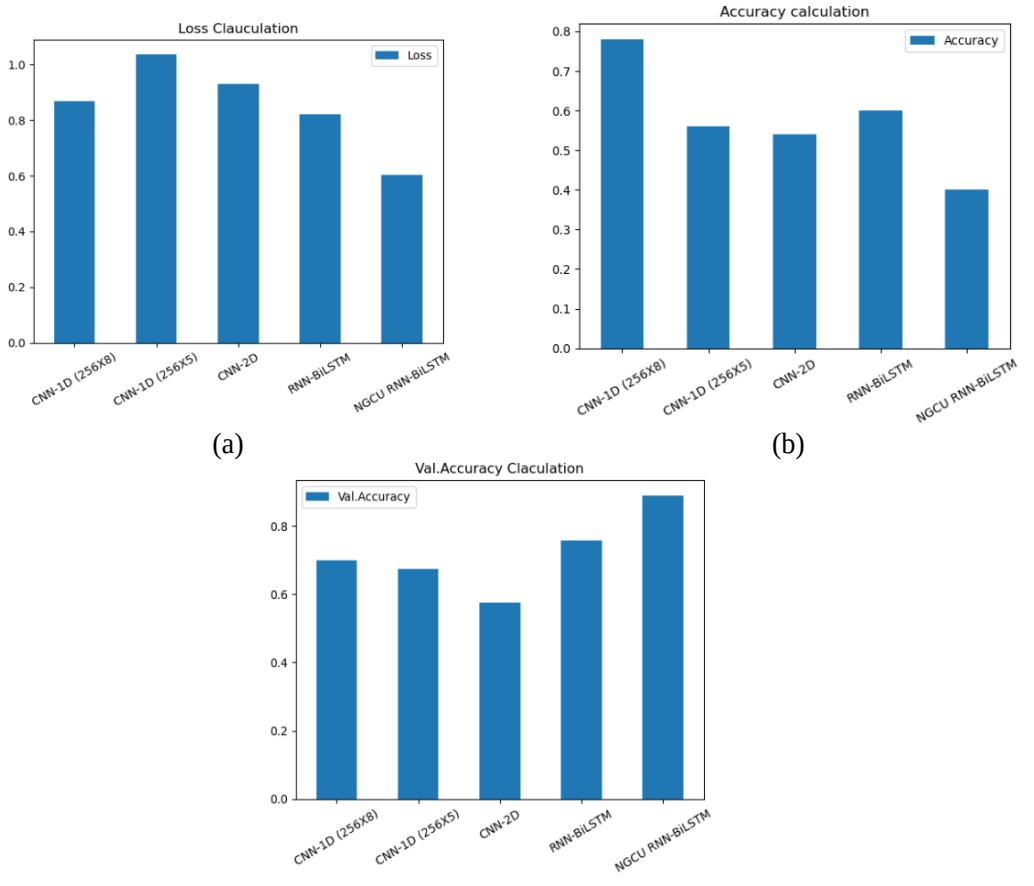


Fig. 11 Performance analysis of 1st sample (a) Loss Calculation (b) Accuracy calculation (c) Validation accuracy calculation.

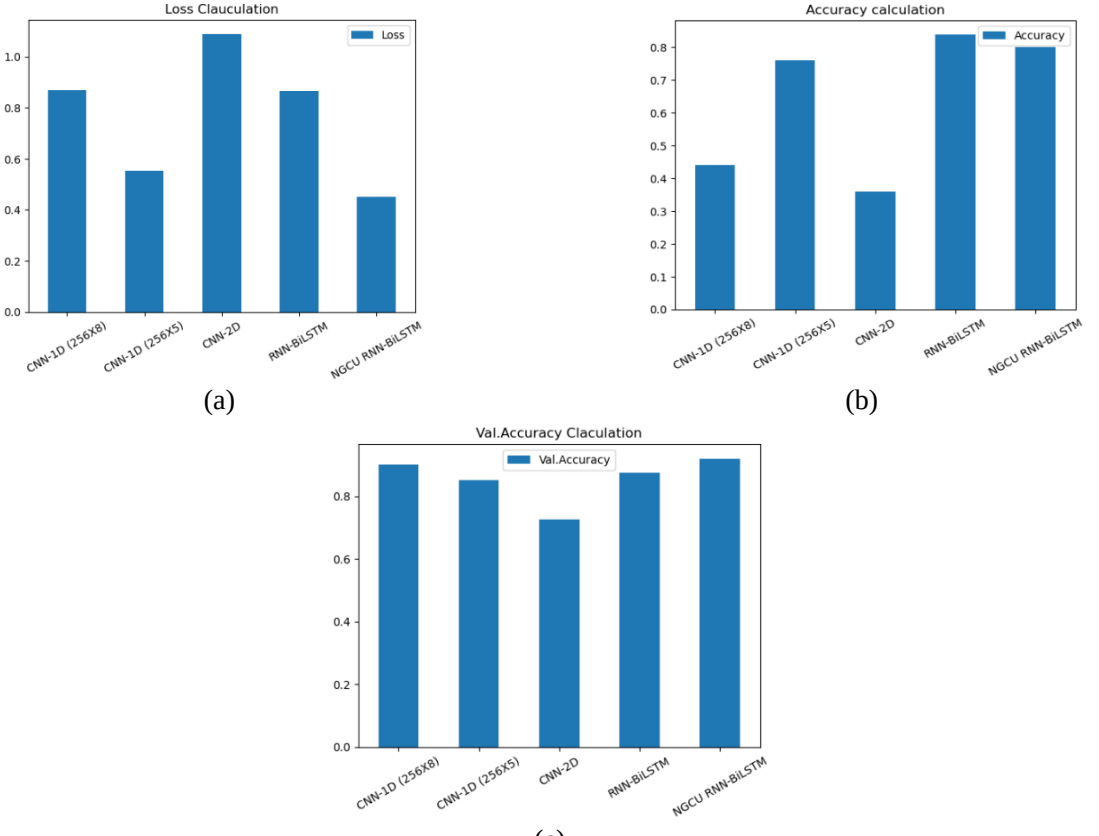


Fig. 12 Performance analysis of 2nd Sample (a) Loss Calculation (b) Accuracy calculation (c) Validation accuracy calculation.

The classification rate sentiment from dialect speech concerning various parameters like peak amplitude, Entropy, Energy, and Center frequency is shown in Fig. 13. In the peak amplitude case, the fluent voice amplitude of the frame is more (79.18) compared to sentiment speech (59.16) and the accuracy of recognizing this is 70.12%. Similarly, the average information content of the frame for continuous speech is slightly high (79.19) whereas for the sentiment frame, the entropy is calculated using eqn.(6) slightly less (79.18), and the difference between these is much less (0.01). The proposed methodology gives good results in the two classifiers that are energy and center frequency, The energy gap between the fluent and sentiment voiced is greater (1.24) compared to entropy entropy-based classification, and concerning center frequency the classification rate is better. The overall accuracy of classifying the sentiment speech w.r.t various VMD-based parameters is 77%. The variation of other VMD-based parameters like Mean(eqn.(7)), Standard deviation(eqn.(8)),

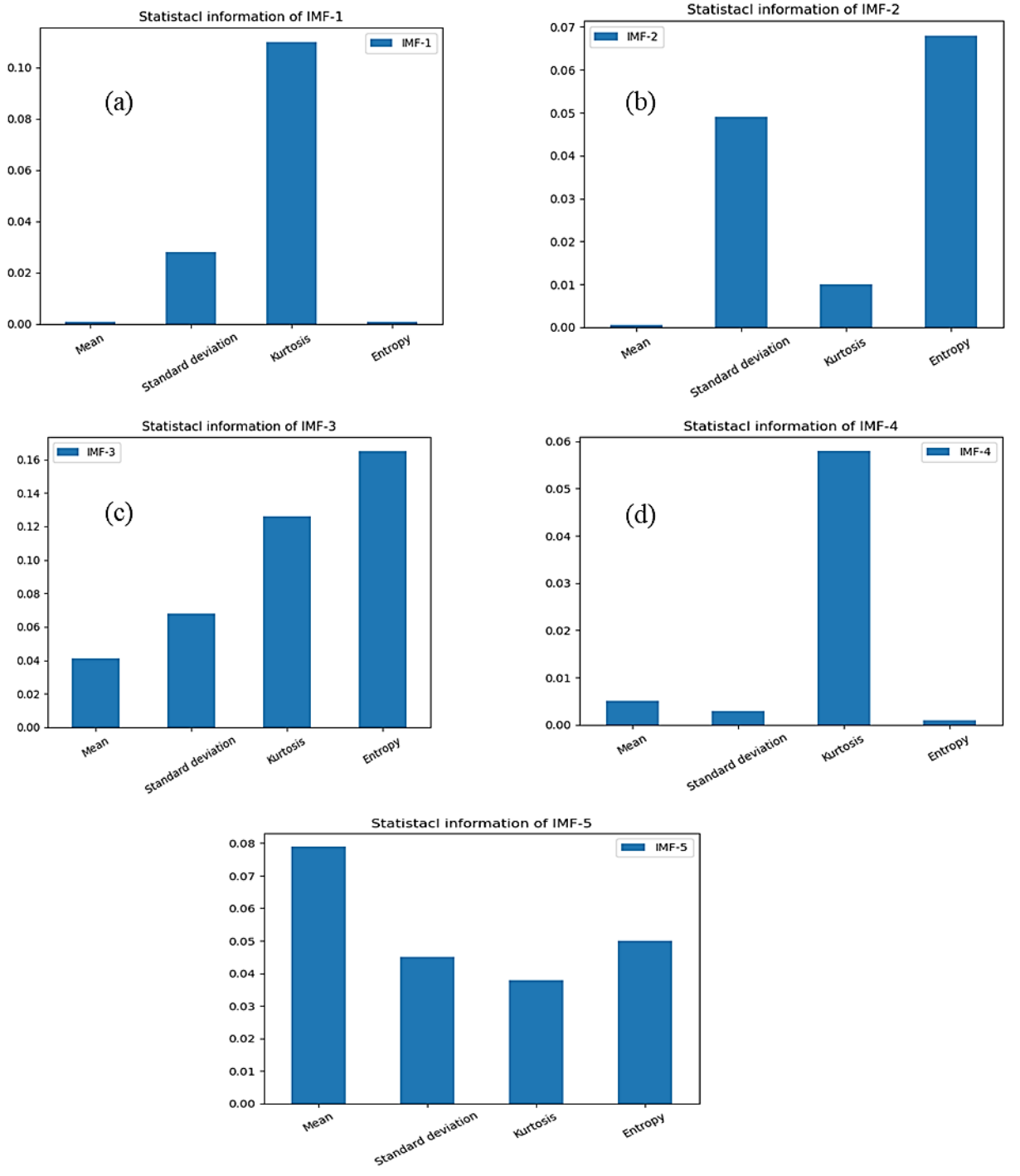


Fig. 13 Variation of other VMD-based parameters like Mean, Standard deviation, Kurtosis, and Entropy (a) Statistical information of IMF-1 (b) Statistical information of IMF-2 (c) Statistical information of IMF-3 (d) Statistical information of IMF-4 (e) Statistical information of IMF-5.

Kurtosis, and Entropy of each intrinsic mode function (IMF) is explained in Fig. 13 for Release One (Monolog). The complete VMD is divided into 5 IMFs, these IMFs are calculated based on the occurrence of the number of "0" in between the successive maxima values and local minima. Also, in the proposed kurtosis statistical analysis is considered, this analysis helps to identify the percentage of peak values in that particular IMF's.

In Fig. 13(a), the variation in terms of SD is more deviated from the mean value, which indicates that the sentiment is speeded widely in IMF-1 and the same observation is found in Fig. 13 (b). In Fig. 13 (a) the variation between local maxima and minima is more indicated by kurtosis but this variation is less in Fig. 13(b). The average information content (Entropy) is more in Fig. 13(a) compare to Fig. 13(b). The variation of all the parameters is increased linearly as indicated in Fig. 13(e), which indicates that the IMF-3 has a sentiment part but not more compared to any other IMF. However, the parameters in IMF-5 look a bit different compared to any other cases from Fig. 13(a) to (d), because the parameters are not much deviated from local maxima and minima values hence the information in IMf-5 is not much affected by sentiment.

After, classifying the sentiment from dialect XGBoost technique is applied to compare the proposed NGCU RNN technique, and the results are shown in Fig. 14.

$$\text{Precision} = \frac{\text{Prediction of True positives}}{\text{Predicted all positives (True positive+False positive)}} \tag{35}$$

$$\text{Recall} = \frac{\text{Prediction of True Neagtives}}{\text{Prediction of all actual positives (True positive+False negative)}} \tag{36}$$

$$\text{F1 score} = \frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}} \tag{37}$$

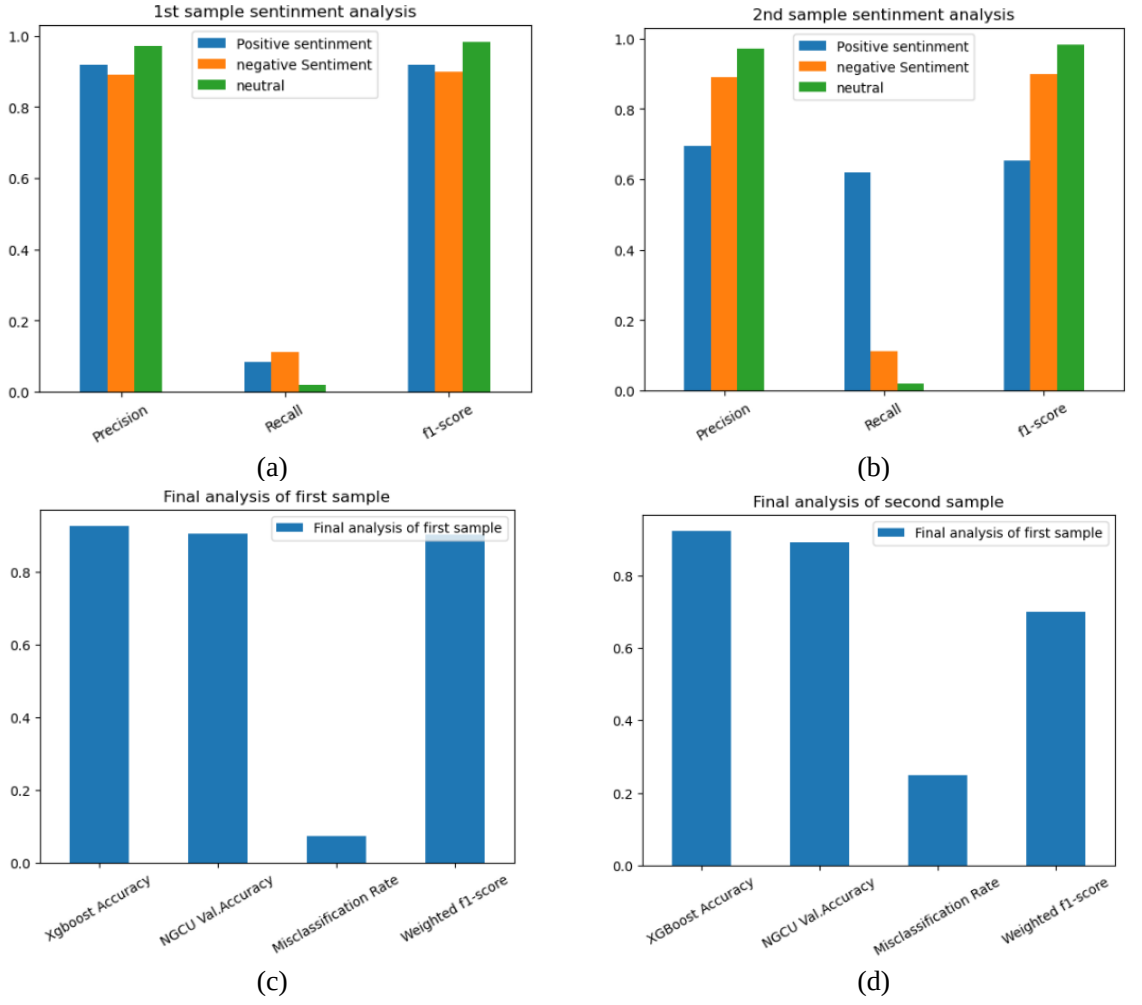


Fig. 14 Performance analysis of the proposed technique after applying the XGBoost technique (a) Sentiment classification from first sample (b) Sentiment classification from second sample (c) Accuracy comparison of the first sample (d) Accuracy comparison of the second sample.

XGBoost provides feature importance rankings, indicating the relative importance of each input feature in predicting sentiment. This information can help identify key features that contribute most significantly to sentiment analysis, enabling better understanding and interpretation of the model's predictions. XGBoost has built-in support for handling missing values in the dataset. It can automatically learn the optimal direction to handle missing values during the training process, saving effort and time in preprocessing steps. Hence the accuracy is improved by 1.46% for the first sample as shown in Fig. 14(c) and 3.06% for the second sample.

4. Conclusion

The proposed approach of utilizing a new gate control unit-recurrent neural network (GCU-RNN) structure in combination with XGBoost for automatic audio-based sentiment analysis shows promising results. The integration of the GCU-RNN allows capturing temporal dependencies and patterns in audio data, while XGBoost leverages its ensemble of weak learners to enhance the overall predictive performance. By leveraging the strengths of both the GCU-RNN and XGBoost, the sentiment analysis model achieves accurate predictions and robust performance. The GCU-RNN effectively captures the temporal dynamics and patterns present in audio signals, enabling a more comprehensive understanding of the sentiment expressed in the audio data. XGBoost, with its powerful ensemble learning and regularization techniques, further enhances the model's predictive capabilities and helps address challenges such as overfitting, feature importance, and handling imbalanced data. The combination of the GCU-RNN and XGBoost leverages the advantages of both techniques, providing a robust and accurate sentiment analysis solution for audio data. This approach can have various practical applications, including sentiment analysis in voice recordings, audio-based social media content, customer service interactions, and more. Further research and experimentation can be conducted to explore different variations and extensions of the proposed approach, fine-tune hyper-parameters, and evaluate its performance on diverse audio datasets. Additionally, incorporating additional features or combining audio features with textual or other modalities could potentially improve the sentiment analysis performance even further. Overall, the integration of the GCU-RNN structure with XGBoost for automatic audio-based sentiment analysis demonstrates a promising direction for research and applications in the field, offering a robust and effective solution for analyzing sentiments expressed through audio data.

Acknowledgements

I would like to acknowledge my institution for providing me an opportunity to carry out my research and also the Research Centre for the technical support and database provided by the Kaggle.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

References

[1] J. Zhang, X. Wu and C. Huang, Transformer-based feature fusion approach for multimodal visual sentiment recognition using tweets in the wild, *IEEE Access*, 2023, **11**, 48410–48420, doi: 10.1109/ACCESS.2023.3276932.

[2] F. Alzamzami, A. E. Saddik, Transformer-based feature fusion approach for multimodal visual sentiment recognition using tweets in the wild, *IEEE Access*, 2023, **11**, 47070–47079, doi: 10.1109/ACCESS.2023.3274744.

[3] A. V. Kotelnikova, S. V. Vychegzhanin, E. V. Kotelnikov, Cross-domain sentiment analysis based on small in-domain fine-tuning, *IEEE Access*, 2023, **11**, 41061-41074, doi: 10.1109/access.2023.3269720.

[4] L. Xiaoyan and R. C. Raga, BiLSTM model with attention mechanism for sentiment classification on chinese mixed text comments, *IEEE Access*, 2023, **11**, 26199–26210, doi: 10.1109/ACCESS.2023.3255990.

[5] J. A. García-Díaz, F. García-Sánchez, R. Valencia-García, Smart analysis of economics sentiment in spanish based on linguistic features and transformers, *IEEE Access*, 2023, **11**, 14211–14224, 10.1109/ACCESS.2023.3244065.

[6] A. Raza, A. Habib, J. Ashraf, B. Shah, F. Moreira, Semantic orientation of crosslingual sentiments: employment of lexicon and dictionaries, *IEEE Access*, 2023, **11**, 7617–7629, doi: 10.1109/ACCESS.2023.3238207.

[7] C. Zheng, M. Bouazizi, T. Ohtsuki, An evaluation on information composition in dementia detection based on speech, *IEEE Access*, 2022, **10**, 92294–92306, doi: 10.1109/ACCESS.2022.3203068.

[8] F. M. Plaza-Del-Arco, M. D. Molina-González, L. A. Ureña-López, M. T. Martín-Valdivia, A multi-task learning approach to hate speech detection leveraging sentiment analysis, *IEEE Access*, 2021, **9**, 112478–112489, doi: 10.1109/ACCESS.2021.3103697.

[9] Q. Yin, R. Zhang, X. Shao, CNN and RNN mixed model for image classification, *MATEC Web of Conferences*, 2019, **277**, 02001, doi: 10.1051/matecconf/201927702001.

[10] B. Xu, H. Li, A novel empirical variational mode decomposition for early fault feature extraction, *IEEE Access*, 2022, **10**, 134826–134847, doi: 10.1109/ACCESS.2022.3232553.

[11] E. Kvedalen, Signal processing using the Teager Energy Operator and other nonlinear operators, Master, University of Oslo Department of Informatics. 2003.
[12] S. Hadiyoso, I. Wijayanto, A. Rizal, S. Aulia, Biometric systems based on ecg using ensemble empirical mode decomposition and variational mode decomposition, *Journal of Applied Engineering Science*, 2020, **18**, 181–191, doi: 10.5937/jaes18-26041.
[13] S. Deb, S. Dandapat, J. Krajewski, Analysis and classification of cold speech using variational mode decomposition, *IEEE Transactions on Affective Computing*, 2020, **11**, 296–307, doi: 10.1109/TAFFC.2017.2761750.
[14] M. Sundermeyer, H. Ney, R. Schluter, From Feedforward to Recurrent LSTM Neural Networks for Language Modeling, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2015, **23**, 517–529, doi: 10.1109/TASLP.2015.2400218.
[15] J. Wang, X. Li, J. Li, Q. Sun, H. Wang, NGCU: A new RNN model for time-series data prediction, *Big Data Research*, 2022, **27**, 100296, doi: 10.1016/j.bdr.2021.100296.
[16] S. Gupta, L. Goel, A. Singh, R. K. Singh, TOXGB: Teamwork optimization based XGBoost model for early identification of post-traumatic stress disorder, *Cognitive Neurodynamics*, 2022, **16**, 833–846, doi: 10.1007/s11571-021-09771-1.
[17] X. Wang, X. Chen, Q. Wang and G. Chen, Early diagnosis of parkinson's disease with speech pronunciation features based on XGBoost model, 2022 IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI), 2022.
[18] S. Gupta, Audio speech sentiment dataset, Kaggle, 2021.
[19] S. Seo, S. Na, J. Kim, HMTL: Heterogeneous modality transfer learning for audio-visual sentiment analysis, *IEEE Access*, 2020, **8**, 140426-140437, doi: 10.1109/ACCESS.2020.3006563.