TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 23, No. 6, December 2025, pp. 1626∼1634
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v23i6.26893

Analyzing temporal properties of speech trajectory using graph structures towards speech recognition

Parabattina Bhagath[1], Malempati Shanmukha[2], Gnana Nagasri Puthi[2]
[1]Department of Artificial Intelligence and Data Science, Lakireddy Bali Reddy College of Engineering, Mylavaram, India
[2]Department of Computer Science and Engineering, Lakireddy Bali Reddy College of Engineering, Mylavaram, India

Article Info

Article history:

Received Jan 4, 2025
Revised Aug 25, 2025
Accepted Sep 10, 2025

Keywords:

Graph eigenvalues
Graph signal processing
Speech analysis
Structural processing
Speech trajectory

ABSTRACT

Speech signal analysis aims to identify patterns within data to develop effective recognition algorithms. This process primarily utilizes feature extraction techniques such as linear predictive coding (LPC), linear predictive cepstral coefficients (LPCCs), and Mel-frequency cepstral coefficients (MFCCs). These features are crucial for constructing recognition algorithms that leverage both statistical and deep learning methods. While deep learning models require extensive datasets, they often prove unsuitable for low-resource languages. The Hidden Markov model (HMM) is the most widely adopted statistical framework in speech processing. However, HMMs are characterized by state-dependent models, where each state interacts only with its neighboring states. This limitation restricts HMMs from capturing long-term signal properties, highlighting the need for addressing these constraints at the feature extraction stage. Most feature extraction methods rely on short-term signal processing, which further limits the comprehension of speech utterances. To overcome these limitations, alternative methods are necessary to capture more comprehensive patterns. This paper presents a graph-based approach for analyzing speech trajectories and their temporal properties, which are subsequently validated using HMMs in speech recognition tasks. Graph-based representations on a low-resource Telugu dataset improve recognition accuracy by 13% while reducing processing time compared to traditional LPC.

Corresponding Author:

Parabattina Bhagath
Department of Artificial Intelligence and Data Science, Lakireddy Bali Reddy College of Engineering
Mylavaram, Andhra Pradesh, India
Email: bhagath.2014@alumni.iitg.ac.in

1. INTRODUCTION

Speech signal analysis is a field that aims to understand speech signals and provide valuable insights for researchers working on developing recognition algorithms and feature extraction techniques. This area has progressed by taking inspiration from various domains such as signal processing, cognitive science [1], linguistics [2], [3], and deep learning [4]. Recently, graph signal processing (GSP) [5] has emerged as a new area of research that focuses on graph-based analysis for understanding patterns of underlying data. It provides a wide variety of tools and techniques for computer networks, speech processing, and data science. Graphs are an efficient way to represent multi-dimensional properties of data, which in turn helps to extract appropriate features. Traditional automatic speech recognition (ASR) systems use various techniques such as linear predictive coefficients (LPCs) and Mel-frequency cepstral coefficients (MFCCs) to extract features from

raw audio signals and transform them into representative feature vectors that capture essential characteristics of speech [6]. One promising avenue for feature extraction is the use of graph eigenvalues, which provide insights into the complex temporal patterns present in spoken utterances. By representing speech signals as graphs and utilizing the eigenvalues of these graphs, we can encode valuable information about the underlying structure and dynamics of speech patterns. In this paper, we investigate the graph-based representation of speech signals to give crucial features for speech recognition tasks. The paper is organized as: in section 2, we present the historical developments of GSP and related research for time series analysis. The section 3 discusses the proposed methodology. In section 4, we discuss the environmental setup used for developing the proposed approach. Finally, section 4 elaborates on the findings whereas the paper is concluded in section 5.

2. LITERATURE REVIEW

The proposed approach is motivated by the fact that a signal can be represented as a graph. GSP has evolved as a research domain studying various areas such as signal processing, communication networks, and big data [7]. GSP adapted the methods from graph theory and digital signal processing and it is providing tools and techniques to solve different problems in the afore-mentioned areas. Graphs are generally used to represent relationships between components in a physical network or data elements [8]. The usage of graphs to represent the relationships is two-fold: causality and invariance. The causality property helps to retain the relationship between the data components whereas the invariance provides a way to hold the relationships in a varying environment. These properties are further useful in speech signal analysis for two reasons. They are useful properties to capture the temporal dynamics and represent the relationships between various acoustic events [9]. Researchers are trying to use graph-based methods to study speech signals and use the concepts such as temporal GSP, graph fourier transform (GFT) [10], and speech modeling techniques.

A graph is a component that has a set of vertices, edges, and weights. It can be represented as $G = (V, E, W)$, where V is a set of vertices, and E represents the edges between the vertices. The weights associated with each edge are given by W. For a signal s represented by a random vector given by $S = x_1, x_2, x_3, ..., x_n$ with n samples, there are two approaches for constructing the graphs for studying the underlying patterns. The first one is a data-driven approach in which the signal is modelled by understanding the statistical distribution. In this method, statistical components such as mean and covariance are commonly used. The second method for graph construction is an intuitive method in which a set of heuristic rules are defined for choosing the components of a graph [11]. A causal graph was proposed by Mei and Moura [12] to estimate the time series. For a discrete-time signal $S[n]$, a causal graph is given by (1). Here $y[n]$ represents the statistical noise, P_i is the filter on matrix A, and N is the order of the model.

$$S[n] = y[n] + \sum_{i=1}^{N} P_i(A)S[n-i] \tag{1}$$

GFT is a technique that helps in understanding the frequency of graphs. GFT for a matrix A is defined based on laplacian L which is given by (2). Here U is the matrix for which the columns are eigenvectors of A and V is the diagonal matrix of eigenvalues [13].

$$L = UVU^T \tag{2}$$

Stationarity and harmonics of a signal are two important properties to find out the relationship between a time series and its associated graph. Understanding these components together provides necessary clues to the temporal and harmonic properties of a signal. Loukas and Foucard [14] defined joint fourier transform (JFT) that uses the principles of discrete fourier transform (DFT) and GFT. JFT gives the frequency content of a signal $S[n]$ w.r.t both time and graph domains. The relationship of a graph represented as $G = \langle V, E, W \rangle$ and a signal $S[n]$ JFT is given by (3) where V_G is the $N \times N$ left eigenvector matrix and XV_T^T is the result of $DFT\{X\}$. GFT provides a way to estimate the varying nature of a signal over the graph and is a useful way of measuring the abrupt changes in a temporal graph.

$$JFT(S, G) = V_G S V_T^T \tag{3}$$

The study of graph structures in the context of speech signal analysis is proven to be effective and it was used for identifying phoneme boundaries. A method to segment the phonetic units using the graph

eigenvalues is proposed by Bhagath and Das [15] and it was proved to capture the temporal properties of speech signals. Gaussian graph signals use the cumulative sum (CUSUM) procedure to acquire knowledge of an event by estimating its prior probability and posterior probability. Generalized likelihood ratio (GLR) is used for estimating post-changes [16], [17]. The domain of speech signal analysis also adapts to graph structures to understand and model the speech data. The methods have been shown to be effective in noisy environments as well. An improved Wiener filter approach is proposed by Wang *et al.* [18] for noisy speech signals. In addition to using the graph structures for representation, they have been used for modeling using deep neural network architectures. Shirian and Guha [19] proposed a graph-based method to recognize emotions using graph neural networks (GNNs). The study of graph structures in the context of speech signal analysis has been proven to be effective, and it was used for identifying phoneme boundaries. A method to segment the phonetic units using the graph eigenvalues is proposed by Bhagath and Das [20], and it was proved to capture the temporal properties of speech signals. The ASR problem for under-resourced languages is challenging, especially for Indian languages. We can find a lot of researchers working in these languages, developing both the corpus and methods. In the literature, there have been numerous research works found to address the speech recognition problem for low-resourced languages [21]. However, these developments are limited to a few languages and domains. Telugu is a language for which there a minimal contribution in the field of speech recognition.

Telugu has multiple dialects for different geographical areas, such as Andhra, Telangana, and Coastal. Developing dialect-specific spoken models is a challenging task because of the availability of the datasets. Yadavalli *et al.* [22] proposed a unified model to recognize words irrespective of the dialect. In the research work, Transformer-based language models were used with block size 16. The proposed approach is discussed in the next section. In the present paper, we develop a low-cost feature extraction method that uses graph structures.

3. PROPOSED FRAMEWORK FOR SPEECH SIGNAL ANALYSIS

The proposed approach consists of four major components: identifying structural components, graph construction, eigenvalue computation, and modeling as outlined in Figure 1. At first, a speech signal is normalized to ensure that the amplitudes are uniform across the dataset. Each speech segment is further analyzed to extract the primary components of a graph's peaks and valleys. The amplitude changes of a spoken utterance form a structural representation in a waveform. This essentially helps to understand the properties of the signal's shape and can be used as features for modeling. Each peak and valley is associated with the location, which tells the sequence of how they are presented in a waveform. A series of graphs will be constructed for a spoken utterance. The construction of the graph is explained in the next subsection. For example, p_0 refers to the peak presented at location 0 in the respective waveform. This is true for the valley also. The presence of a peak and a valley in a location is mutually exclusive, which means a peak and a valley cannot be in the same location. With these components, a graph is constructed for each speech segment, usually having a 10-ms to 20-ms duration. To understand the procedure, we first define the primitive elements of a graph: peaks and valleys. A peak in a speech segment is a sample s_p that satisfies (4), where p is an integer and the position of a peak. A valley in a speech segment is a sample s_v that satisfies (5), where v is an integer and the position of a valley. A graph designated as G is a set of vertices V and edges E. Each vertex in a graph can be a peak or a valley. The edges between the vertices are defined as a link between a peak and a valley, designated as p_i and v_i. The individual steps in the proposed approach are explained in the subsequent subsections.

$$s_{p-1} < s_p < s_{p+1} \tag{4}$$

$$s_{v-1} > s_v > s_{v+1} \tag{5}$$

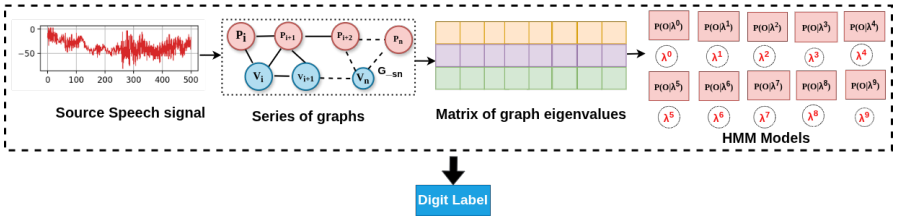

Figure 1. Proposed framework using graph eigenvalues

3.1. Graph construction

At first, a set of peaks and valleys are found using the primitives given by (4) and (5). An example graph structure constructed this way is shown in Figure 2. In this structure, a link between a peak and a valley is created if they are both positioned in subsequent locations. Based on the starting component of a graph, there are three possible structures that contain a varied number of peaks and valleys. There are three types of edges possible in a graph as follows:

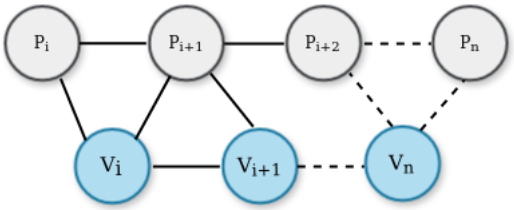

Figure 2. Proposed graph structure

- An edge is defined between two consecutive peaks, defined as (p_i, p_{i+1})
- An edge is defined between two consecutive valleys defined as (v_i, v_{i+1})
- An edge is defined between a peak and a valley positioned in consecutive locations defined as (p_i, v_i)

Once the graphs are constructed for a speech signal, graph eigenvalues are computed for each graph. The procedure to compute graph eigenvalues is discussed in the next subsection.

3.2. Graph eigenvalue computation

The procedure of eigenvalue computation involves three important matrices such as the adjacency matrix, the diagonal matrix, and the Laplacian matrix. For a graph G, the adjacency matrix designated as A_G is defined as the matrix consisting of either '0' or '1'. A_G contains 1 in $A_G(i, j)$ if $node_i$ and $node_j$ are adjacent. D_G is the diagonal matrix, and it contains the vertex degrees of graph G. The Laplacian matrix L_G is given by (6) and (7). Finally, the eigenvalues of G can be computed by solving (8). These steps are outlined in Algorithm 1. The solution to equations give the eigenvalues that represent a speech signal. For a speech signal $S[n]$ with samples n and window length k, $\frac{n}{k}$ number of frames can be formed which is designated as n_{frames} in the algorithm. This factor decides the number of graphs $|G|$ formed from a source signal. Subsequently, each graph is used to compute an eigenvalue that represents a feature for each speech segment. The graph eigenvalues are computed for all the corresponding graphs using lines 8 to 10 in the algorithm. Once this step is completed, we need to model each spoken word with a modeling technique. A sample set of graph eigenvalues computed for digits zero through eight are shown in Figure 3.

Algorithm 1 Graph eigenvalue computation

1: **procedure** GEV(G_k) ▷ G_k is a series of graphs
2: $i \leftarrow 0$
3: **while** $i \neq n_{frames}$ **do** ▷ n_{frames} is the number of segments
4: $G_i \leftarrow$ Construct graph(s_i) ▷ Map each s_i to g_i
5: **end while**
6: $i \leftarrow 0$
7: **while** $i \neq |G|$ **do**
8: $L(g_i) \leftarrow D(g_i) - A(g_i)$ ▷ Laplacian of graph g_i
9:
10: $ge_i \leftarrow \frac{U^T(g_i)A(g_i)U(g_i)}{U^T(g_i)U(g_i)}$
11: **end while**
12: **return** ge_i ▷ Set of graph eigenvalues for S[n]
13: **end procedure**

For verifying the proposed features, we used a Hidden Markov model (HMM) for modeling the spoken data, which is discussed in the next subsection.

$$L_G = D_G - A_G \tag{6}$$

$$L_G(i,j) = \begin{cases} degree_i, & \text{if i=j} \\ -A_G(i,j), & \text{if } i \neq j \end{cases} \tag{7}$$

$$|L_G - \lambda(G)| = 0 \tag{8}$$

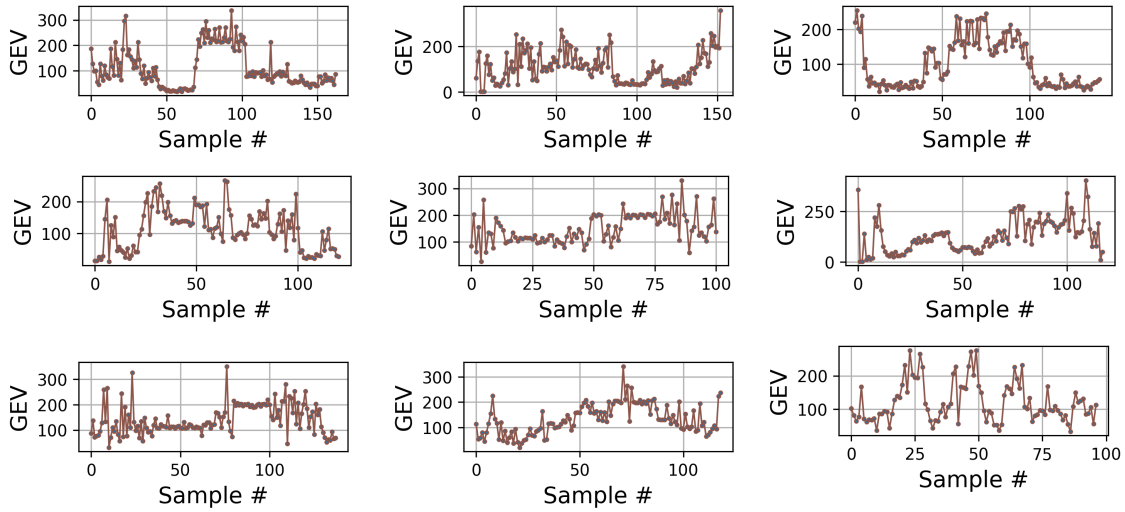

Figure 3. Graph eigenvalues for the digits from 'zero' through 'nine' (one speaker)

3.3. Hidden Markov model modeling

The HMM are finite state machines in which a process is represented by a set of components. The elements of HMM are a set of states, a transition matrix, and a special start state. HMM is often called a stochastic process, which means an unknown process can be observed through a known process [23]. This approach is suitable for solving the speech recognition problem for one reason. In speech recognition, the spoken utterance is known, and the vocal tract system that utters the message is hidden. By using the HMM, we are supposed to estimate the properties of the vocal tract system. The HMM solves a speech recognition problem in three stages. The present approach uses a 4-state HMM model in the recognition module as shown in Figure 4.

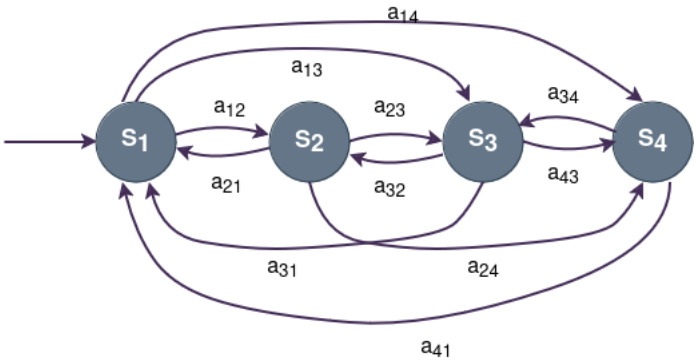

Figure 4. 4-state HMM model

The first step focuses on knowing the parameters of the model by understanding the patterns hidden in the observation sequence, often called as features. In the second stage, the states of the HMM are decoded to find out the optimal state sequence that produces the observation sequence. Finally, the model is evaluated

by estimating how well the model can generate the given observation sequence. This is usually computed by maximum likelihood estimation. In general, the number of states in an HMM is decided based on the number of phonetic units in the targeted words. The proposed approach is tested on a Telugu Digit dataset, and the number of states in the HMM is appropriate.

4. EXPERIMENTAL SETUP AND RESULTS

In this section, we detail the environment used for module development and outline our observations from the experiments. The complete dataset comprises 17,600 speech samples of ten distinct words, spoken by 88 different speakers. These words represent Telugu digits, ranging from zero (0) to nine (9), with each digit being articulated 20 times. The dataset is available at the following link: `https://tinyurl.com/57rsz5xs`. In our study, 70% of the dataset was designated for training, while the remaining 30% was utilized for model evaluation. Each speech utterance underwent pre-processing, which involved removing the direct current (DC) component and normalizing the audio to a range of {-1, +1}. This normalization ensures consistent amplitude values across all speakers. The details of the dataset are summarized in Table 1. We employed several libraries, including scientific Python (SciPy), Python graph signal processing (PyGSP) [24], and NetworkX [25], to enhance our work on HMM modeling, the implementation of graph structures, and the computation of graph eigenvalues. SciPy provides a range of functions for processing matrix data structures with its built-in capabilities. PyGSP is a library that enables the application of signal processing concepts to graphs, whereas NetworkX facilitates the representation of graph structures.

Table 1. Summary of the dataset

S. no	Attributes	Values
1	Total # of distinct words	10
2	Total # of speakers	88
3	Total # of sessions	80
4	Total # of files	17,600
5	Total duration of all the files	587 mins (9.8 hours)

The recognition rates of 3 different features built with HMM are shown in Table 2. Our findings reveal that the recognition rate for the linear predictive coding (LPC)+HMM model ranges from 24% to 76%, linear predictive cepstral coefficients (LPCC)+HMM model gives recognition rates varying from 22% to 82% whereas the recognition rates for the graph EigenValues (GEV)+HMM model span from 37% to 100%. These experiments illustrate that the proposed approach achieves a superior recognition rate when utilizing structural features. In addition to the better recognition rates, there is a reduction in the computational cost since a single eigenvalue needs to be computed for each speech segment. In contrast, the conventional LPC and LPCC methods compute 13 coefficients for each speech segment, resulting in a higher dimensionality. This new method reduces the length of the feature vector by a factor of 12, offering significant computational advantages during the modeling phase. The results are presented in Figure 5.

Table 2. Recognition rates (%) obtained for different speakers

S. no	LPC	LPCC	GEV	S. no	LPC	LPCC	GEV	S. no	LPC	LPCC	GEV
1	44	64	46	13	41	50	62	25	48	48	54
2	52	52.66	46	14	54	52	66	26	62	68	70
3	54	60	54	15	37	36	68	27	56	48	52
4	52	22	48	16	54	36	88	28	46	58	56
5	46	82	52	17	52	30	88	29	34	34	37
6	54	82	58	18	24	24	96	30	44	50	46
7	60	62	52	19	76	48	100	31	52	48	66
8	54	30	54	20	44	56	46	32	54	72	88
9	38	48	54	21	60	48	90	33	48	28	40
10	58	58	54	22	52	48	54	34	56	18	96
11	56	46	56	23	54	44	100	35	36	38	44
12	52	40	58	24	60	36	90	36	46	40	66

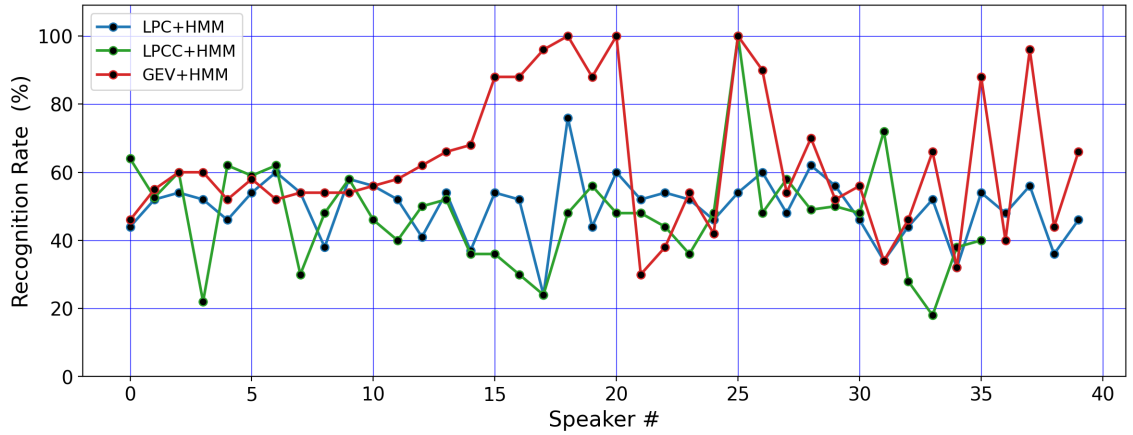

Figure 5. Comparison between LPC and GEV features for 40 speakers

5. CONCLUSION

This paper introduces a graph-based method for the analysis of speech signals. The methodology employs graph structures as feature representations within the temporal domain, effectively capturing the inherent patterns of a speech signal through this graphical representation. The proposed feature extraction method was tested in recognition tasks using HMM to assess its efficacy. The performance of the models developed for the dataset was compared to traditional approaches using LPC+HMM and LPCC+HMM. The results indicate a 13% improvement in recognition rates over LPC and LPCC features, along with a 12-fold reduction in computational costs. Future work will focus on analyzing the scalability of the proposed features with larger datasets.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Parabattina Bhagath	✓	✓			✓	✓			✓	✓	✓			
Malempati Shanmukha		✓	✓	✓		✓		✓		✓				
Gnana Nagasri Puthi		✓	✓	✓		✓			✓					

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available on request from the corresponding author, [initials: PB]. The data, which contain information that could compromise the privacy of research participants,

are not publicly available due to certain restrictions.

REFERENCES

[1] R. Voleti, J. M. Liss, and V. Berisha, "A Review of Automated Speech and Language Features for Assessment of Cognitive and Thought Disorders," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 2, pp. 282–298, Feb. 2020, doi: 10.1109/JSTSP.2019.2952087.

[2] L. Xue, F. K. Soong, S. Zhang, and L. Xie, "ParaTTS: Learning Linguistic and Prosodic Cross-Sentence Information in Paragraph-Based TTS," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2854–2864, 2022, doi: 10.1109/TASLP.2022.3202126.

[3] Z. Zhang, E. Coutinho, J. Deng, and B. Schuller, "Distributing recognition in computational paralinguistics," *IEEE Transactions on Affective Computing*, vol. 5, no. 4, pp. 406–417, Oct. 2014, doi: 10.1109/TAFFC.2014.2359655.

[4] D. F. O. Onah and A. Ibrahim, "Evaluating Speech Emotion Recognition through the lens of CNN; LSTM Deep Learning Models," *in 2023 IEEE International Conference on Big Data (BigData), IEEE*, Dec. 2023, pp. 6245–6247, doi: 10.1109/BigData59044.2023.10386881.

[5] J. Wang and X. Pan, "The signal feature extraction of graph Fourier transform on the constructed graph," in *Proceedings - 2021 6th International Conference on Communication, Image and Signal Processings, CCISP 2021, IEEE*, Nov. 2021, pp. 379–383, doi: 10.1109/CCISP52774.2021.9639270.

[6] A. Chowdhury and A. Ross, "Fusing MFCC and LPC Features Using 1D Triplet CNN for Speaker Recognition in Severely Degraded Audio Signals," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1616–1629, 2020, doi: 10.1109/TIFS.2019.2941773.

[7] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, Sep. 2014, doi: 10.1109/MSP.2014.2329213.

[8] E. Ceci and S. Barbarossa, "Graph signal processing in the presence of topology uncertainties," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1558–1573, 2020, doi: 10.1109/TSP.2020.2976583.

[9] X. He, Y. Xie, S.-M. Wu, and F.-C. Lin, "Sequential Graph Scanning Statistic for Change-point Detection," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers, IEEE*, Oct. 2018, pp. 1317–1321, doi: 10.1109/ACSSC.2018.8645505.

[10] A. G. Veeravalli, W. D. Pan, R. Adhami and P. G. Cox, "A tutorial on using hidden Markov models for phoneme recognition," *Proceedings of the Thirty-Seventh Southeastern Symposium on System Theory*, 2005, pp. 154-157, doi: 10.1109/SSST.2005.1460896.

[11] C. Zhang, D. Florencio, and P. A. Chou, "Graph Signal Processing – A Probabilistic Framework," *Microsoft Research Technical Report*, p. MSR-TR-2015-31, 2015.

[12] J. Mei and J. M. F. Moura, "Signal Processing on Graphs: Causal Modeling of Unstructured Data," in *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2077-2092, 2017.

[13] N. Perraudin, A. Loukas, F. Grassi, and P. Vandergheynst, "Towards stationary time-vertex signal processing," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 3914–3918, 2017, doi: 10.1109/ICASSP.2017.7952890.

[14] A. Loukas and D. Foucard, "Frequency analysis of time-varying graph signals," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE*, Dec. 2016, pp. 346–350, doi: 10.1109/GlobalSIP.2016.7905861.

[15] P. Bhagath and P. K. Das, "Phoneme Boundary Analysis Using Graphs," in *IEEE Region 10 Annual International Conference, Proceedings/TENCON, IEEE*, Oct. 2019, pp. 1764–1768, doi: 10.1109/TENCON.2019.8929673.

[16] A. Buciulea and A. G. Marques, "Graph Learning from Gaussian and Stationary Graph Signals," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, IEEE, Jun. 2023*, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10096413.

[17] Y. Chen, X. Mao, D. Ling, and Y. Gu, "Change-Point Detection of Gaussian Graph Signals with Partial Information," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, IEEE*, Apr. 2018, pp. 3934–3938, doi: 10.1109/ICASSP.2018.8461397.

[18] T. Wang, H. Guo, X. Yan, and Z. Yang, "Speech signal processing on graphs: The graph frequency analysis and an improved graph Wiener filtering method," *Speech Communication*, vol. 127, pp. 82–91, Mar. 2021, doi: 10.1016/j.specom.2020.12.010.

[19] A. Shirian and T. Guha, "Compact graph architecture for speech emotion recognition," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, IEEE*, Jun. 2021, pp. 6284–6288, doi: 10.1109/ICASSP39728.2021.9413876.

[20] P. Bhagath and P. K. Das, "Graph Eigenvalue based Structural Method towards Phonetic Boundary Detection," in I*EEE Region 10 Annual International Conference, Proceedings/TENCON, IEEE*, Dec. 2021, pp. 591–596, doi: 10.1109/TENCON54134.2021.9707281.

[21] N. Choudhary and D. G. Rao, "The LDC-IL Speech Corpora," in *Proceedings of 2020 23rd Conference of the Oriental COCOSDA International Committee for the Co-Ordination and Standardisation of Speech Databases and Assessment Techniques, O-COCOSDA 2020, IEEE*, Nov. 2020, pp. 28–32, doi: 10.1109/O-COCOSDA50338.2020.9295011.

[22] A. Yadavalli, G. S. Mirishkar, and A. K. Vuppala, "Multi-Task End-to-End Model for Telugu Dialect and Speech Recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, ISCA: ISCA, Sep.* 2022, pp. 1387–1391, doi: 10.21437/Interspeech.2022-10739.

[23] D. Ali, I. Touqir, A. M. Siddiqui, J. Malik and M. Imran, "Face Recognition System Based on Four State Hidden Markov Model," in *IEEE Access*, vol. 10, pp. 74436-74448, 2022, doi: 10.1109/ACCESS.2022.3188717.

[24] M. Defferrard, L. Martin, R. Pena, and N. Perraudin, "PyGSP: Graph Signal Processing in Python," *Zenodo*, no. Version v0.5.0, 2017, doi: 10.5281/zenodo.1003158.

[25] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," in *Proceedings of the 7th Python in Science Conference*, Jun. 2008, pp. 11–15, doi: 10.25080/TCWV9851.

BIOGRAPHIES OF AUTHORS

Parabattina Bhagath is a professor in the Department of Artificial Intelligence and Data Science at Lakireddy Bali Reddy College of Engineering in Mylavaram, India. He earned his M.Tech. and Ph.D. from the Indian Institute of Technology Guwahati, India, in 2010 and 2020, respectively. He specializes in speech processing, particularly for resource-constrained languages and IoT devices. He has developed structural processing methods to understand speech signals using their structural properties. These procedures serve as alternative methods and procedures to address low-resource data, in addition to conventional machine-learning techniques. His research has been published in the proceedings of various esteemed conferences. He can be contacted at email: bhagath.2014@alumni.iitg.ac.in.

Malempati Shanmukha completed his B.Tech. in Computer Science and Engineering at Lakireddy Bali Reddy College of Engineering, located in Mylavaram, India. His research focuses on Telugu spoken language modeling for classifying gender and age groups using convolutional neural networks (CNNs). By analyzing speech patterns and leveraging techniques like MFCC extraction, he has contributed to developing effective solutions for speech-based applications in regional languages. He can be contacted at email: malempatishanmukha@gmail.com.

Gnana Nagasri Puthi completed her B.Tech. in Computer Science and Engineering at Lakireddy Bali Reddy College of Engineering, located in Mylavaram, India. Her research focuses on Telugu spoken language modeling for classifying gender and age groups using convolutional neural networks (CNNs). By analysing speech patterns and leveraging techniques such as MFCC extraction, she has contributed to the development of effective solutions for speech-based applications in regional languages. He can be contacted at email: gnananagasri29@gmail.com.