Attention based Residual-Time Delay Neural Network for Indian Language Identification

by

Tirusha Mandava, Anil Kumar Vuppala

in

12th International Conference on Contemporary Computing (IC3)
(*IC3-2019*)

Noida, India

Report No: IIIT/TR/2019/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
August 2019

Attention based Residual-Time Delay Neural Network for Indian Language Identification

Tirusha Mandava and Anil Kumar Vuppala
Speech Processing Laboratory
International Institute of Information Technology, Hyderabad, India
mandava.tirusha@research.iiit.ac.in, anil.vuppala@iiit.ac.in

***Abstract*—India is a multilingual society having more than 1600 languages. Most of these languages are having an overlapping set of phonemes. This makes developing language identification (LID) framework difficult for Indian languages. In this paper, the above challenge is addressed using phonetic features. To model the temporal variations in phonetic features, attention based residual-time delay neural network (RES-TDNN) is proposed. This network effectively captures long-range temporal dependencies through TDNN and attention mechanism. The proposed network has been evaluated on IIITH-ILSC database using phonetic and acoustic features. The database consists of 22 official Indian languages and Indian English. Attention based RES-TDNN outperformed the other state-of-the-art networks such as deep neural network, long short-term memory network and produced an equal error rate of 9.46%. Further, the fusion of shifted delta cepstral and phonetic features have improved the performance.**

***Index Terms*—Attention based residual-time delay neural network, Equal error rate, Language identification system, Multihead**

I. Introduction

Language identification (LID) is a task to identify the language from a spoken utterance. With an efficient LID system, human-computer interface applications can be more productive and reach multilingual socities [1]. LID system employed at front-end for a broad range of multilingual speech systems, such as spoken language translation, multilingual speech recognition, service customization, and forensics [2]. Performance of the LID system depends on the efficient representation of language information and effective methods for language classification.

The early-stage research on LID systems uses statistical models like Gaussian mixture models, hidden Markov models, and support vector machines (SVM) at the model level [3]–[5]. The i-vector techniques with conventional classifiers such as SVM, probabilistic linear discriminant analysis have been demonstrated their effectiveness and obtained state-of-the-art performance [6]. Later, deep neural networks (DNN) have been explored and shown excellent performance [7]. However, in DNN, frame level decisions are averaged over an utterance to predict language ID instead of utterance level decision (language information is present more precisely at utterance level than frame level). Sequential networks such as recurrent neural networks (RNN) and long short-term memory networks (LSTM) are used to predict language ID through modeling the long temporal information [8]. Even though these sequential networks process the whole input signal at a time, they cannot be parallelized and are computationally intensive. Recently self-attention networks and Bi-directional DNN with gated recurrent neural units are proposed in [9], [10]. End to end LID systems using convolution neural network, LSTM, and attention based hierarchial gated recurrent units are explored [11]–[13].

At the feature level, shifted delta cepstral (SDC) coefficients are state-of-the-art acoustic features for LID [14]. These features are obtained from augmentation of the conventional Mel-frequency cepstral coefficients (MFCC) to capture long-term temporal context. Features which are embedded with contextual information learned from a network in a non-linear discriminant fashion can effectively represent the language information and are robust to noise as compared to conventional acoustic features [15]. In this context, log-likelihood ratios, stacked bottleneck, and multilingual tandem bottleneck features have been used for LID, and these outperformed the standard features [15]–[17]. Time delay neural network (TDNN) acoustic model is used to convert the acoustic sequence into a phoneme sequence [18]. In [19], senone based LSTM-RNN framework is investigated to extract phonetic features.

Most of the works in the literature on Indian LID have been focused on prosody and spectral features [20]. In [21], language specific features are extracted from CV transition regions and steady vowel regions. Hilbert envelops and phase information of linear prediction residual are explored in [22]. Power normalized features [23], implicit excitation source features [24], and phase information related features [25] are studied. Recently, DNN with attention architecture is explored using MFCC features [9]. This paper explores the phonetic features and deep neural networks for LID which are not well explored in the Indian scenario.

We proposed attention based RES-TDNN for Indian LID task. The motivation for this work is that TDNN better models the long-range temporal dependencies [26] which play a vital role in LID. Frame level log likelihood scores obtained from the acoustic model of ASR are used as phonetic features. IIITH-Indian language speech corpus (ILSC) is used to carry out LID experiments [27]. To the best of our knowledge LID with attention based RES-TDNN using phonetic features is the first time explored.

This paper is structured as follows: Section II describes

978-1-7281-3591-5/19/$31.00 ©2019 IEEE

the fundamental architectural aspects of attention based RES-TDNN. The experimental setup is explained in Section III. In Section IV results and discussions are reported. The conclusion is presented in Section V.

II. Attention based Residual-Time Delay Neural Network

The architecture of RES-TDNN with attention is described in Fig. 1.

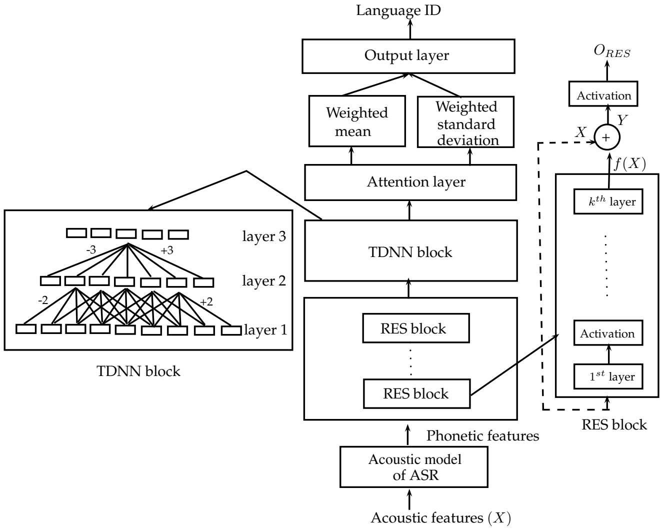

Fig. 1. Block diagram of attention based RES-TDNN. The input to the network is either acoustic or phonetic features.

The network contains four blocks, i.e., (i) stacked RES blocks (ii) TDNN block, (iii) attention statistics layer, (iv) and output layer. The entire network is trained with a single objective function in an end to end fashion to maximize language identification accuracy.

- **Stacked RES blocks** are act as feature extractors which takes an acoustic sequence as an input and transform these features to a higher level representation. Each RES block contains several feed forward layers followed by a non-linear activation function. This stacked block structure allows skip connections between layers.
- **TDNN** effectively captures long-range temporal dependencies through learning affine transformations on different length context windows. In general, initial layers learn these transformations on narrow context window and deeper layers on a wider context window.
- **Attention statistics** layer aggregates frame level features by selecting prominent frames. It computes a scalar value for every frame that specifies the relative importance of each frame. The utterance level representation is obtained by concatenating the mean and standard deviation of the weighted hidden layer representations.
- **Output** layer is a feed forward layer which takes utterance level representation and computes softmax probability scores for each language.

The pipeline of architecture from input acoustic/phonetic sequence to language ID is shown in Fig. 2.

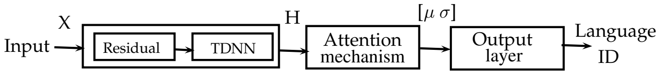

Fig. 2. Input flow in attention based RES-TDNN.

Let $X = [x_1, x_2, x_3, ..., x_L]$ be an input acoustic sequence. Here L is the length of input sequence. Data flow in RES block can be explained using the below two equations:

$$Y = f(X) + X \tag{1}$$

$$O_{RES} = ReLU(Y) \tag{2}$$

where, $f(X)$ is the output of last layer in RES block without non-linear activation function and O_{RES} is the output of RES block. The final RES block output is passed through TDNN block and $H = [h_1, h_2, h_3, ..., h_L]$ be the corresponding output. From these hidden activations (H), attention statistics layer computes a scalar value for every frame as follows:

$$e_t = tanh(W_a H^T) \tag{3}$$

where W_a are attention layer parameters. The values of e_t are normalized using a softmax function as follows:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{t=1}^{L} \exp(e_t)}. \tag{4}$$

The mean and standard deviation of weighted hidden activations are computed as given below:

$$\mu = \sum_{t=1}^{L} \alpha_t h_t \tag{5}$$

$$\sigma = \sqrt{\sum_{t=1}^{L} \alpha_t h_t \odot h_t - \mu \odot \mu} \tag{6}$$

where $\odot$ represents Hadamard product. Both weighted mean and standard deviation vectors are concatenated and given as an input to the output layer to predict the language ID.

In the proposed network, multi-head attention is also explored with the intuition that each head captures unique discriminant information. Multi-head attention can be implemented with ease by increasing the number of attention statistics layers. Final utterance wise representation is obtained by concatenating each attention layer representation. In the case of multi-head, the objective function (cross-entropy) is penalized by the factor (ϵ) as defined below to ensure each head captures unique information [28].

$$\epsilon = ||W_a W_a^T - I||_F^2 \tag{7}$$

where $||.||_F$ represents the Frobenius norm of the matrix and T represents transpose of a matrix. Implementation details of the architecture have been explained in Section III.

III. Experimental setup

This section briefly gives details of the database, baseline and proposed LID systems.

A. Database

To evaluate the performance of the proposed network, we have considered IIITH-ILSC database. It consists of 23 languages, Assamese, Bengali, Bodo, Dogri, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Maithili, Malayalam, Manipuri, Marathi, Nepali, Odia, Punjabi, Sanskrit, Santali, Sindhi, Tamil, Telugu, Urdu, and Indian English. Details of the database are listed in Table I. It contains data from both noisy and clean environments. Sample wave files from the database are available in the link: https://researchweb.iiit.ac.in/~mandava.tirusha/LID_IC3_19.html.

TABLE I
Details of IIITH-ILSC database.

IIITH-ILSC	
Number of speakers	50 (25 male and 25 female for each language)
Amount of data	4.5 hours for each language
Duration of utterances	5-10 sec
Sampling frequency	16kHz

B. Baseline Features

This sub-section briefly explains the extraction of different baseline features used for developing the LID systems. In this study, standard features like MFCC, SDC, i-vector, and phonetic are used as baseline features.

- **MFCC** features are extracted from 20 ms windowed speech signal with an overlap of 10 ms having dimension 40. First and second order derivatives are added to these features resulting in a 120-dimensional feature vector.
- **SDC** features are computed from MFCC by concatenating delta cepstral coefficients over multiple frames. Extraction of these features involves four parameters N (number of cepstral coefficients), d (delta distance between acoustic vectors), p (distance between blocks), k (total number of successive blocks used to compute SDC features). This work uses widely used 7-1-3-7 (N-d-p-k) configuration to compute SDC features resulting in a 56-dimensional acoustic vector [29].
- **i-vector** features are computed using Kaldi i-vector script. Extraction involves universal background model, which consists of 2048 Gaussian mixtures trained with 40-dimensional MFCC features and dimension of the extracted i-vector is 100. For every 10 frames, one i-vector is extracted.
- **Phonetic features** are computed from the 40-dimensional MFCC vector as follows. For every MFCC vector (x_t), phone posterior probabilities are computed from the acoustic model which is trained on Microsoft data [30] using DNN. These posterior probabilities are considered as phonetic features in this work and dimension is 70.

C. Baseline LID systems

In this study, three different networks are used as baseline LID systems namely DNN, DNN with residual connections (DNN-RES), and LSTM. Training of all networks used Adam as an optimizer. Learning rate is halved upon observing an increase in validation cost. The training is halted upon encountering an increase in validation cost over three successive epochs. In DNN, DNN-RES a symmetric 4 frame window and in LSTM a symmetric 2 frame window is used to splice adjacent frames. In all networks, hidden units are followed by ReLU activation function and networks are trained with cross-entropy objective function. All the above networks are implemented using pytorch.

Finer details of all networks are explained below.

- **DNN** architecture used in this work is a fully connected feed forward neural network with the same architecture in [27]. It contains four hidden layers and the number of units in each layer is set to be 1024.
- **DNN-RES** network contains four residual blocks, and each block contains two hidden layers with the same architecture as described in [31]. The first hidden layer has 1024 units and the second hidden layer has a number of units equal to the input dimension. The output of the second hidden layer is added to the input and is given as input to the next residual block.
- **LSTM** network contains two hidden layers in which each layer is followed by a projection layer of dimension equal to the hidden layer. Each layer contains 320 cells.

D. Proposed RES-TDNN with attention

RES-TDNN with multi-head attention network contains five RES blocks and one TDNN block. Each RES block contains three layers, in which the first and third layer having a number of units equal to the input feature dimension and the second layer having 1024 units. TDNN block contain three layers with temporal context [-1, 1], [-2, 2], [-3, 3] with 256 units respectively. Attention layer is a single feedforward layer which computes utterance wise representation with dimension equal to the dimension of last hidden layer of TDNN block. The output layer contains a number of units equal to the number of language classes.

IV. Results and Discussion

Our work aims to improve the performance of LID systems in the Indian scenario. In this connection, attention based RES-TDNN architecture is proposed and is compared with state-of-the-art networks using phonetic and acoustic features. The multi-head attention mechanism is investigated in the proposed architecture. Further, using this network fusion of acoustic and phonetic features are studied at the network level and feature level. In this paper, the performance of the LID system is presented in terms of the equal error rate (EER).

A. LID system using Multi-head attention based RES-TDNN

The performance of proposed attention based RES-TDNN and baseline architectures using different features are listed in Table II. The experimental results have shown that the proposed network outperformed the other networks. The best performance of attention based RES-TDNN can be attributed to two factors. The first factor is that efficient long-term temporal modeling capability of TDNN and second is that aggregation of frame level features by attention layer to predict language ID instead of taking frame level decisions in other networks. At the feature level, phonetic features have better performance compared to the other features. From these observations, it is speculated that long temporal information is playing a vital role in LID either at feature level or network level.

TABLE II
BENCHMARK COMPARISON (EER IN %) OF LID SYSTEMS.

Network	Features			
	MFCC	SDC	i-vector	Phonetic
DNN	22.42	17.95	14.72	**13.34**
DNN-RES	21.87	17.12	14.25	**12.56**
LSTM	20.05	16.59	14.08	**12.22**
Attention based RES-TDNN	**15.45**	**13.81**	**13.68**	**9.46**

The multi-head attention mechanism is investigated in the proposed network, and the results are tabulated in Table III. It can be seen that using multi-head attention has improved the performance of LID systems. The improvement in the performance can be associated with its better utterance wise representation (each head captures distinct information) compared to the single-head attention.

TABLE III
RESULTS (EER IN %) OF LID SYSTEMS TRAINED USING MULTI-HEAD RES-TDNN.

Number of heads	SDC	Phonetic
1-head	13.81	9.46
2-head	14.05	9.47
3-head	**12.82**	**8.82**

B. *Analysis of phonetic and acoustic (SDC) features using attention based RES-TDNN*

Combination of SDC and phonetic features at the network level (fuse the scores obtained from the individual models) and at feature level (train a network by concatenating SDC and phonetic features) has a significant improvement compared to the individual models. These results are presented in Table IV.

TABLE IV
RESULTS (EER IN %) OF LID SYSTEMS WITH THE FUSION OF ACOUSTIC AND PHONETIC FEATURES.

Number of heads	Fusion at network level	Fusion at feature level
1-head	7.75	8.55
2-head	7.94	8.48
3-head	**7.42**	**8.22**

Further, we briefly discuss the confusion between languages in acoustic (SDC) and phonetic space. Fig. 3 and 4 represents confusion matrix for 23 languages using phonetic and acoustic features respectively. The following observations are noted from the confusion matrices.

1) In acoustic space,
   a) Marathi is confused with Gujarati.
   b) Kokani is confused with Sindhi.
   c) Kannada is confused with Manipuri.
   d) Malayalam is confused with Sindhi.

   However, in phonetic space above language pairs are distinguishable. This may be due to these language pairs have similar sound units in acoustic space, but the characteristics of these sound units are different (due to phonotactic constraints) in phonetic space. This assumption is proved in the case of Marathi and Gujarati since these languages have similar sounds [32].
2) Hindi is confused with Urdu in phonetic space indicating that both the languages are phonetically similar structure. It is also observed that Hindi (92%) and Urdu (78%) accuracy is high in acoustic space.
3) The accuracy of some languages such as Telugu (33% to 92%), Tamil (49% to 90%), and Malayalam (44% to 81%) is significantly improved in phonetic space as compared to the acoustic space.

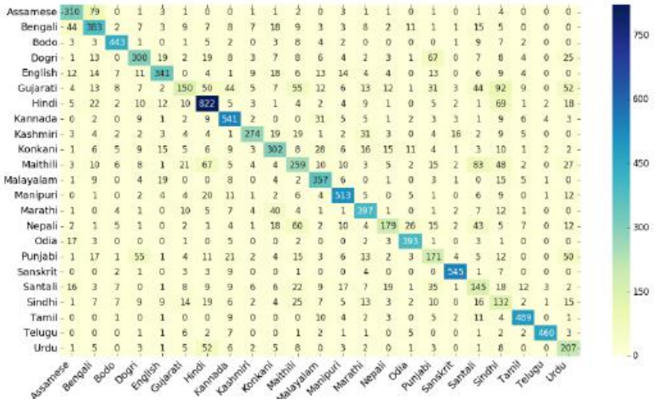

Fig. 3. Confusion matrix for attention based RES-TDNN using phonetic features.

V. CONCLUSION

In this paper, attention based residual-time delay neural network is proposed for Indian language identification. This network is trained using phonetic features extracted from the acoustic model of an automatic speech recognizer. The proposed network outperformed state-of-the-art methods and produced an equal error rate of 9.46%. Multi-head attention further improved performance. The best performance noticed in this work is with the fusion of shifted delta cepstral and phonetic features at the network level having an EER of 7.42%. The consistency of the network has to be studied with short duration utterances. Our future studies are targetted in the presence of noise and other mismatched conditions.

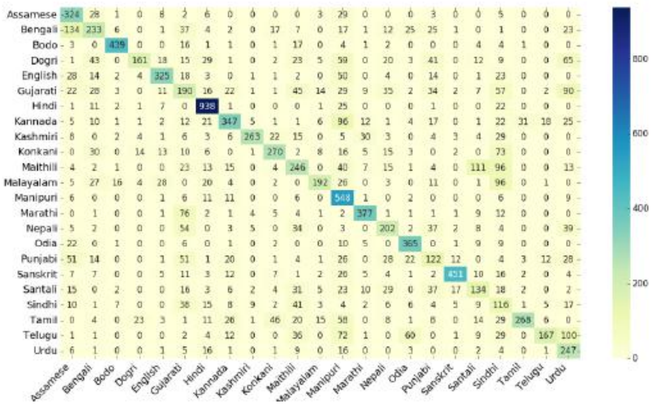

Fig. 4. Confusion matrix for attention based RES-TDNN using acoustic (SDC) features.

VI. Acknowledgements

The authors would like to thank Science & Engineering Research Board (SERB) for funding Language Identification in Practical Environments (YSS/2014/000933) project.

References

[1] F. Pellegrino and R. André-Obrecht, “Automatic language identification: an alternative approach to phonetic modelling,” *Signal Processing*, vol. 80, no. 7, pp. 1231–1244, 2000.

[2] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, “Language identification: a tutorial,” *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82–108, 2011.

[3] P. A. Torres-Carrasquillo, D. A. Reynolds, and J. R. Deller, “Language identification using Gaussian mixture model tokenization,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2002, pp. 757–760.

[4] M. A. Zissman, “Automatic language identification using gaussian mixture and hidden markov models,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1993, pp. 399–402.

[5] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, and D. A. Reynolds, “Language recognition with support vector machines,” in *Proc. Odyssey*, 2004, pp. 285–288.

[6] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, “Language recognition via i-vectors and dimensionality reduction,” in *Proc. INTERSPEECH*, 2011, pp. 861–864.

[7] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, “Automatic language identification using deep neural networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 5337–5341.

[8] G. Gelly and J. Gauvain, “Spoken language identification using LSTM-based angular proximity,” in *Proc. INTERSPEECH*, 2017, pp. 2566–2570.

[9] K. Mounika, S. Achanta, H. Lakshmi, S. V. Gangashetty, and A. K. Vuppala, “An investigation of deep neural network architectures for language recognition in indian languages.” in *Proc. INTERSPEECH*, 2016, pp. 2930–2933.

[10] L. Mateju, P. Cerva, J. Zdansky, and R. Safarik, “Using deep neural networks for identification of Slavic languages from acoustic signal,” in *Proc. INTERSPEECH*, 2018, pp. 1803–1807.

[11] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, “Application of convolutional neural networks to language identification in noisy conditions,” in *Proc. Odyssey*, 2014, pp. 287–292.

[12] W. Geng, W. Wang, Y. Zhao, X. Cai, B. Xu *et al.*, “End-to-end language identification using attention-based recurrent neural networks.” in *Proc. INTERSPEECH*, 2016, pp. 2944–2948.

[13] B. Padi, A. Mohan, and S. Ganapathy, “End-to-end language recognition using attention based hierarchical gated recurrent unit models,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 5966–5970.

[14] M. A. Kohler and M. Kennedy, “Language identification using shifted delta cepstra,” in *Proc. IEEE Midwest Symposium on Circuits and Systems*, 2002, pp. 69–72.

[15] W. Geng, J. Li, S. Zhang, X. Cai, and B. Xu, “Multilingual tandem bottleneck feature for language identification,” in *Proc. INTERSPEECH*, 2015, pp. 413–417.

[16] M. Diez, A. Varona, M. Penagarikano, L. J. Rodriguez-Fuentes, and G. Bordel, “On the use of phone log-likelihood ratios as features in spoken language recognition,” in *Proc. IEEE Spoken Language Technology Workshop*, 2012, pp. 274–279.

[17] R. Fér, P. Matějka, F. Grézl, O. Plchot, and J. Černockỳ, “Multilingual bottleneck features for language recognition,” in *Proc. INTERSPEECH*, 2015, pp. 389–393.

[18] Z. Tang, D. Wang, Y. Chen, L. Li, and A. Abel, “Phonetic temporal neural model for language identification,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 134–144, 2018.

[19] Y. Tian, L. He, Y. Liu, and J. Liu, “Investigation of senone-based long-short term memory RNNs for spoken language recognition,” in *Proc. Odyssey*, 2016, pp. 89–93.

[20] V. R. Reddy, S. Maity, and K. S. Rao, “Identification of Indian languages using multi-level spectral and prosodic features,” *International Journal of Speech Technology*, vol. 16, no. 4, pp. 489–511, 2013.

[21] D. Nandi, A. K. Dutta, and K. S. Rao, “Significance of cv transition and steady vowel regions for language identification,” in *Proc. IEEE International Conference on Contemporary Computing*, 2014, pp. 513–517.

[22] D. Nandi, D. Pati, and K. S. Rao, “Language identification using hilbert envelope and phase information of linear prediction residual,” in *Proc. IEEE International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation*, 2013, pp. 1–6.

[23] A. K. Dutta and K. S. Rao, “Robust language identification using power normalized cepstral coefficients,” in *Prco. IEEE International Conference on Contemporary Computing*, 2015, pp. 253–256.

[24] D. Nandi, D. Pati, and K. S. Rao, “Implicit excitation source features for robust language identification,” *International Journal of Speech Technology*, vol. 18, no. 3, pp. 459–477, 2015.

[25] A. K. Dutta and K. S. Rao, “Language identification using phase information,” *International Journal of Speech Technology*, vol. 21, no. 3, pp. 509–519, 2018.

[26] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, “Phoneme recognition using time-delay neural networks,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[27] R. K. Vuddagiri, K. Gurugubelli, P. Jain, H. K. Vydana, and A. K. Vuppala, “IIITH-ILSC speech database for Indian language identification,” in *Proc. Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2018, pp. 56–60.

[28] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, “Self-attentive speaker embeddings for text-independent speaker verification,” in *Proc. INTERSPEECH*, 2018, pp. 3573–3577.

[29] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell, and D. A. Reynolds, “Acoustic, phonetic, and discriminative approaches to automatic language identification,” in *Proc. European Conference on Speech Communication and Technology*, 2003, pp. 1345–1348.

[30] B. M. L. Srivastava, S. Sitaram, R. Kumar Mehta, K. Doss Mohan, P. Matani, S. Satpal, K. Bali, R. Srikanth, and N. Nayak, “Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages,” in *Proc. Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2018, pp. 11–14.

[31] R. K. Vuddagiri, H. K. Vydana, and A. K. Vuppala, “Improved language identification using stacked SDC features and residual neural network,” in *Proc. Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2018, pp. 205–209.

[32] D. Sengupta, Goutam Saha, “Study on similarity among Indian languages using language verification framework,” *Advances in Artificial Intelligence*, vol. 2015, no. 325703, p. 24, 2015.