IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 28, 2018, accepted January 8, 2019, date of publication January 31, 2019, date of current version February 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2896453

Native Language Identification in Very Short Utterances Using Bidirectional Long Short-Term Memory Network

FARAH ADEEBA[1] **AND SARMAD HUSSAIN**[2]
[1]Department of Computer Science and Engineering, University of Engineering and Technology, Lahore 54890, Pakistan
[2]Center for Language Engineering, Al-Khwarizmi Institute of Computer Science, University of Engineering and Technology, Lahore 54890, Pakistan
Corresponding author: Farah Adeeba (fadeeba@gmail.com)

ABSTRACT Native language identification (NLI) is the task of identifying the first language of a user based on their speech or written text in a second language. In this paper, we propose the use of spectrogram- and cochleagram-based features extracted from very short speech utterances (0.8 s on average) to infer the native language of an Urdu speaker. The bidirectional long short-term memory (BLSTM) neural networks are adopted for the classification of utterances among the native languages. A set of experiments is carried out for the network architecture search and the system's accuracy is evaluated on the validation data set. Overall accuracy of 74.81% and 71.61% is achieved using the Mel-frequency cepstral coefficients (MFCC) and Gammatone frequency cepstral coefficients (GFCC), respectively. Moreover, the optimized MFCC feature-based BLSTM network and GFCC feature-based BLSTM network are merged together to take advantage of both the feature sets. The experiments show that the performance of the merged network surpasses the individual BLSTM networks and accuracy of 75.69% is achieved on the evaluation data. The effect of test data duration is also analyzed (from 0.27 s to 1.5 s); in addition, it is observed that with very short duration as 0.4 s, an accuracy of over 50% can be achieved.

INDEX TERMS Native language identification, BLSTM, RNN, Urdu L2.

I. INTRODUCTION

Native language identification (NLI) is the task of identifying the first language (L1) of a user based on their speech or written text in a second language (L2). Native language identification from speech data is a relatively new research area and there is limited research on this problem. For NLI from textual data, however, a number of approaches have been explored using lexical and structural features such as character, word, part-of-speech, n-grams and dependency relations [1]–[3]. Usually, multilingual speakers lack thorough acquisition of L2, resulting in common artefacts such as certain pronunciation differences and distinct foreign accents [4]. Accurate detection of native language can be useful for numerous human-machine voice interface applications, such as computer assisted language learning (CALL) system, automated speech assessment system, speaker forensics and adaptation in automated speech recognition (ASR) systems. NLI can also facilitate a spoken dialog system by suggesting a user's cultural background.

Majority of the research in the area of NLI is focused on identifying the native language of speakers learning English as a second language. However, the NLI shared task of 2013 [6] boosted interest in this area by making available a large corpus of non-native English [7] for public use. This shared task was focused on NLI identification using textual information covering 11 native languages; Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu and Turkish. In NLI shared task of 2017 [47] evaluation is carried out for three tracks: (1) NLI using textual information, (2) NLI using speech information and (3) NLI using both textual and speech information. For NLI, the native language speech corpus (NLSC) [5] and TOEFL11 corpus [7] recorded from non-native English speakers are widely used. These corpora consist of written responses and speech utterances recorded during English language proficiency tests. Moreover, Norwegian NLI has also been explored by using textual information of learners of Norwegian language [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Mei.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

In text-based NLI, lexical and syntactic features are widely used. Word n-grams and character n-grams based features are used to extract lexical information, whereas, part of speech tags and dependency parse trees are used to extract syntactic information from written text. These features are modeled using different classifiers to classify the L1 of a writer. Jarvis *et al.* [48] used word n-grams, lemmas and part of speech tags for NLI. Support vector machine (SVM) is used for the classification of L1 languages. Proposed technique is evaluated on corpus of non-native English [7] and an accuracy of 83.6% is achieved. This system best performed in 2013 NLI shared task. Ionescu *et al.* [2] used character level n-grams to identify the native language. String kernels technique is used for classification purpose. Proposed technique is evaluated on TOFEL11 corpus [7] and maximum accuracy of 85% is achieved. Cimino and Dell'Orletta [49] used classifier stacking approach i.e. sentence level classification prediction are used in document level classification. Logistic regression model trained on stylistic, syntactic and lexical features is used for sentence level classification. SVM is used for document-level classification. SVM is trained using the lexical, syntactical, stylistic and sentence level prediction features. This system best performed in 2017 NLI shared task with F1-measure of 0.88.

For speech-based NLI, both segmental and supra-segmental level features are explored in the literature. Segmental level features deal with the phoneme level information i.e. L2 phoneme replacement with a similar L1 phoneme (similar manner or place of articulation). This is mostly due to the absence of the L2 phoneme in the L1 phonemic inventory. Supra-segmental features are mostly related to L2 intonation patterns, which are relatively difficult to acquire for adults. In literature, both segmental and supra-segmental features are explored for NLI. Mel-frequency cepstral coefficients (MFCC), perceptual linear prediction (PLP) and Gammatone frequency cepstral coefficients (GFCC) are widely used acoustic-based NLI features.

In a study by Rajpal *et al.* [10], both spectral- and source-based features are used to identify 11 languages as L1 from non-native English speech. The NLSC corpus is used for training and evaluation of the system, accuracy of 40.2% is achieved. In [11], i-vectors [12] extracted from spectral features, i.e. Mel frequency cepstral coefficients (MFCC) are used for NLI from a set of 11 languages. The authors used support vector machines and neural networks for classification of speech utterances. Each utterance in the test data consists of 45 seconds speech. I-vector with neural networks outperformed the baseline resulting in un-weighted average recall (UAR) [13] of up to 67.4%. Abad *et al.* [14] used phone posterior probabilities-based i-vector system for NLI and achieved UAR of 81.3%. Proposed system is evaluated on test data consists of 45 seconds long speech utterances. This system best performed in 2017 NLI shared Task [47]. MFCCs feature-based Gaussian super vectors are used by [15] to identify the native language using NLSC corpus, achieving UAR of 69.72%.

Identification of native language from speech utterance is similar to the language, speaker, accent and dialect identification tasks. Therefore, similar features and techniques are used for these tasks. State-of-the-art identification systems are based on the exploitation of acoustic [16]–[18], phonotactic [19], [20] and prosodic [21], [22] features. Acoustic feature-based approaches explore how a given language sounds; phonotactic feature-based approaches [20] use possible phone combinations of each language to infer the language from a speech utterance; and prosodic feature-based techniques focus on intonation patterns. For speech utterance classification, Gaussian mixture models with universal background model (GMM-UBM) [23], joint factor analysis (JFA) [24] and i-vector [12] framework are widely used.

A resurgence of deep learning-based techniques has revived the use of neural networks for speech processing. Deep neural networks (DNN) yield state-of-the-art performance in classification tasks. DNNs have been used for language identification [25], [26] and evaluated on 3 seconds long utterances. DNNs outperformed i-vector framework and a substantial improvement in accuracy was observed. Recurrent neural networks (RNN) with long short-term memory (LSTM) memory cells outperformed i-vector in the task of language identification [27] for short utterances. Seven MFCC along with SDC with configuration of (7-1-3-7) are used as LID feature set. Comparative analysis between the i-vector framework and LSTM has been carried out. The study showed that for 3-second long utterances, LSTM outperformed the i-vector system by up to 26%. In addition, the effect of the test utterance duration is also analyzed on the limited duration test data (from 0.1 seconds to 2.5 seconds). The system's accuracy deteriorates as the data duration decreases and an overall accuracy of 50% is achieved for 0.5 second long utterances. Usually, a combination of different features or approaches tends to provide better accuracy of the system [28]. In a study by [29], in addition to DNNs, RNNs are also explored for accent identification and a fusion of DNNs and RNNs is experimented. Fusion of networks is evaluated using the NLSC corpus on test data with 45-second utterances. It is observed that a combination of networks performed better as compared to individual networks.

In literature, there is paucity of research studies related to the NLI using speech information of learners of languages other than English. Moreover, the need of speech corpus for languages other than English is also highlighted in report of 2017 NLI shared task [47]. The availability of the publically available bench mark speech corpus will facilitate the researchers to analyze the characteristics of a language, compare features and different classification techniques.

This paper is first attempt to study speech-based NLI for five native languages i.e. Balochi, Pashto, Punjabi, Sindhi and Saraiki. To address data paucity challenge, publically available speech corpus [9] is used to infer native language of Urdu (L2) speakers. This dataset is also used to develop the automatic speech recognition system for district names of Pakistan [50].

Motivated by the outstanding performance of LSTM-RNN in related fields, in this study we use the Bidirectional LSTM (BLSTM) model for NLI task of very short utterances (0.27s – 2s). Performance of NLI using two different types of acoustic features i.e. spectrogram and cochleagram is investigated. The training procedure of LSTM networks, particularly BLSTM, takes more time and tuning than feed-forward networks. In this study, many aspects of BLSTM training are explored such as number of hidden layers, size of hidden layers and regularization methods. Moreover, spectrogram feature-based BLSTM-RNN and cochleagram feature-based models are merged together to utilize the strengths of both features.

The organization of the paper is as follows. Section 2 discusses the input features of the NLI system. Section 3 discusses the state-of-art i-vector system whereas Section 4 is on the BLSTM- RNN system with corresponding network optimization experiments. Section 5 describes the proposed system that combines spectrogram and cochleagram feature-based BLSTM-RNN models. Section 6 briefly describes the dataset and Section 7 describes the experimental setup. The evaluation results of the experiments are shown and discussed in Section 8 and Section 9 concludes the work findings.

II. FEATURES

Acoustic characteristics of speech segments are used as input to the NLI systems. In this study, spectrogram-based features i.e. Mel frequency cepstral coefficients (MFCC) and cochleagram-based features i.e. Gammatone frequency cepstral coefficients (GFCC) are used to represent the acoustic characteristics of speech. MFCC and GFCC features are widely used in different speech processing applications e.g. automatic speech recognition (ASR) [45], deceptive speech detection [46], speaker identification, language identification and native language identification etc. The major difference between spectrogram and cochleagram is that spectrogram features are based on Mel scale and cochleagram features are based on equivalent rectangular bandwidth (ERB) scale which has better resolution at low frequencies. These features are computed along with the shifted delta cepstral (SDC) coefficients. They are an extension of delta-cepstral coefficients i.e. stacked version of delta coefficients over several frames. SDCs are used to enhance the accuracy of speaker recognition, language recognition and native language detection systems [11], [30]–[32]. For a given N-dimensional cepstral feature vector, SDC are calculated by concatenating K blocks of delta coefficients as shown in Equation 1.

SDC features are typically written as N-d-P-k where:

- N: number of cepstral coefficients in each frame
- d: time advance and delay for delta computation
- P: time shift between consecutive frames
- K: number of frames to be concatenated to form the final vector

The detailed shifted delta computation is reported by Campbell *et al.* [33]. MFCC features are extracted with a hamming window of 20ms with 10ms frame shift, filtered through a Mel-scale filter bank. Seven MFCC features are used and SDC parameters are computed with the configuration of 7-1-3-7 and concatenated with static coefficients, resulting in a 56-dimensional MFCC input vector.

$$\Delta c\,(t, i) = c\,(t + iP + d) - c(t + iP - d) \tag{1}$$

Extracted MFCC features are stored in vector as shown in Equation 2.

$$F_M = \{F_{M1}, F_{M2}, F_{M3}, \ldots F_{M56}\} \tag{2}$$

The GFCC features are auditory feature based on a set of Gammatone filters which simulate the frequency response of human ears. Gammatone filter bank outputs frequency-time representation of a signal which is called a cochleagram. This cochleagram representation is required for computation of GFCC features. Gammatone filter can be computed using the Equation 3.

$$g(t) = at^{n-1}e^{-2\pi bt}\cos\left(2\pi f_c + \varphi\right) \tag{3}$$

where φ is the phase (usually set to zero), n represents the order of filter, a denotes value of amplitude, f_c is the central frequency (in KHz) and b is the bandwidth or rate of decay. The factor b is defined as:

$$b = 1.019 * 24.7 * (\frac{4.37 f_c}{1000} + 1) \tag{4}$$

In experiments, we used filter of order (n) 4 and calculated the Gammatone filter with 64 channels. For the GFCC feature vector, the first 10 channels of Gammatone filters which correspond to frequency range less than 200 Hz are excluded. Seven GFCC features are used and SDC parameters with configuration of 7-1-3-7 are computed which results in a 56-dimensional vector of GFCC-SDC.

Extracted GFCC features are stored in vector shown in Equation 5

$$F_G = \{F_{G1}, F_{G2}, F_{G3}, \ldots F_{G56}\} \tag{5}$$

III. I-VECTOR MODEL

Inspired by the use of joint factor analysis (JFA) [24], Dehak *et al.* [12] proposed a new approach for front-end factor analysis, termed i-vectors. JFA is based on separate speaker and channel dependent subspace, whereas i-vector is based on a single space, referred as the ''total variability space''. This new space includes both speaker and channel variabilities simultaneously, without making any distinction between channel and speaker effects. I-vector system development is divided into three phases; (1) i-vector extraction (2) variability compensation and (3) scoring. Each i-vector based system differs in terms of variability compensation used to perform identification. In i-vector approach, speaker and session dependent Gaussian mixture model (GMM) super-vector $\boldsymbol{\mu}$ is extracted. This super vector is represented as

$$\boldsymbol{\mu} = \mathbf{m} + \mathbf{Tw} \tag{6}$$

Here m is a speaker and session independent universal background model (UBM) super-vector, T is a rectangular low rank total variability matrix representing the variation across a large collection of training data and w is an independent vector based on normally distributed random vector $N(0, I)$. The w vector is represented by the Baum Welch (BW) statistics N and F for a given utterance u, which are extracted using the UBM.

For a sequence of T frames and an UBM m composed of C mixture components defined in some feature space of dimension D. The BW statistics for a given speech utterance u are obtained by

$$N_c(u) = \sum_{n=1}^{T} P(C|y_n, m) \tag{7}$$

$$F_c(u) = \sum_{n=1}^{T} P(C \mid y_n, m)(y_t, m_c) \tag{8}$$

Here c $= 1, .., C$ is the Gaussian mixture index, mc is the mean mixture of component c and $P(c|y_n, m)$ is the posterior probability of mixture component c given the observation y_n at time n. The i-vector w is obtained by using the following equation

$$w = (I + T^t \sum^{-1} N(u)T)^{-1} . T^t \sum^{-1} \widetilde{F}(u)$$

$N(u)$ is a CF $\times$ CF dimension diagonal matrix with diagonal blocks $N_c I(c = 1, C)$. $\sum$ is a CF $\times$ CF dimensional diagonal covariance matrix obtained during factor analysis training. Whereas, $F(u)$ is a CF $\times$ 1 dimension vector acquired by concatenating all F_c statistics for a given utterance u. Details of the derivation of these parameters can be found in [50].

During the enrollment phase, UBM is trained using the available training data; same training data is used for training of total variability matrix i.e. T-matrix.

In i-vector based LID system, channel compensation is carried out in total variability space rather than GMM super vector and low dimensional vectors are extracted. Number of channel compensation techniques exist e.g. within-class covariance normalization (WCCN) [52], probabilistic linear discriminant analysis (PLDA), nuisance attribute projection (NAP) [53] etc. A heavy-tailed PLDA (HTPLDA) approach [54] is also investigated for speaker recognition and it is observed that HTPLDA performs better as compared to GPLDA. Garcia-Romero *et al.* [55] used length-normalized approach that transforms the HTPLDA towards a more Gaussian distribution resulting into similar performance but less computational complex. In this study, length normalized GPLDA technique is used for eliminating the nuisance effects. The motivation for using this technique is that, GPLDA reduce the dimensionality of i-vector and attempts to find out new orthogonal axes to maximize variance between different classes.

Score of test utterance is computed as log likelihood ratio of the same versus different language model hypothesis. Given two i-vectors V_{target} and V_{test}, the likelihood ratio can be computed as follows:

$$\ln \mathrm{P}(\mathrm{V_{target}}, \mathrm{V_{test}}|\mathrm{H_1})/\mathrm{P}(\mathrm{V_{target}}|\mathrm{H_0})\mathrm{P}(\mathrm{V_{test}}|\mathrm{H_0}) \tag{9}$$

Here $\mathrm{H_1}$ represents the hypothesis that the I-vector belongs to the same language and $\mathrm{H_0}$ denotes the hypothesis that they do not.

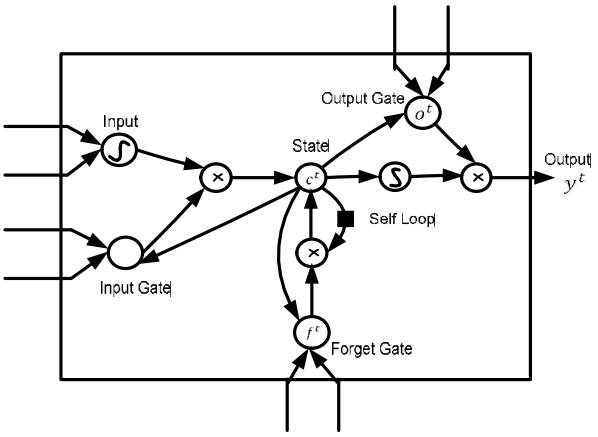

FIGURE 1. Block diagram of a long short-term memory cell.

IV. BIDIRECTIONAL LSTM RNN MODEL

Long short term memory (LSTM) network [34] is a special type of recurrent neural network (RNN) with the capability of learning long-term dependencies. Each LSTM cell has inputs, outputs and a system of gating units to control the information flow. Internal state unit ($\mathrm{c^t}$) is the key component of the cell which is regulated by the multiplicative units called gates i.e. input gate ($\mathbf{i^t}$), output gate ($\mathbf{o^t}$) and forget gate ($\mathbf{f^t}$). A block diagram of an LSTM cell is shown in Fig. 1. Equation 10 represents the LSTM block input [27]. Equations of LSTM inputs, outputs, state unit and gates are provided in Equation 11, Equation 14, Equation 13 and Equation 12, respectively, more details of which can be found in [35].

$$\mathbf{Z^t} = \mathbf{tanh}(\mathbf{W_z x^t} + \mathbf{R_z y^{t-1}} + \mathbf{b_z}) \tag{10}$$

$$\mathbf{i^t} = \boldsymbol{\sigma}\left(\mathbf{W_i x^t} + \mathbf{R_i y^{t-1}} + \mathbf{P_i} \odot \mathbf{c^{t-1}} + \mathbf{b_i}\right) \tag{11}$$

$$\mathbf{f^t} = \boldsymbol{\sigma}\left(\mathbf{W_f x^t} + \mathbf{R_f y^{t-1}} + \mathbf{P_f} \odot \mathbf{c^{t-1}} + \mathbf{b_f}\right) \tag{12}$$

$$\mathbf{c^t} = \mathbf{i^t} \odot \mathbf{z^t} + \mathbf{f^t} \odot \mathbf{c^{t-1}} \tag{13}$$

$$\mathrm{o^t} = \boldsymbol{\sigma}\left(\mathrm{W_o x^t} + \mathrm{R_o y^{t-1}} + \mathrm{P_o} \odot \mathrm{c^t} + \mathrm{b_o}\right) \tag{14}$$

$$y^t = o^t \odot \tanh(c^t) \tag{15}$$

where σ is the logistic sigmoid function, i^t, o^t, f^t, *and* c^t are the input, output, forget gate and cell internal state vectors, respectively. Here all b are bias vectors, the P are peephole weight vectors and R are recurrent weight matrices. x^t is the input vector of size 56 (MFCC-SDC or GFCC-SDC) at time t, W are input weight matrices, *tanh* is the hyperbolic tangent activation function and $\odot$ is the element-wise product of the vectors.

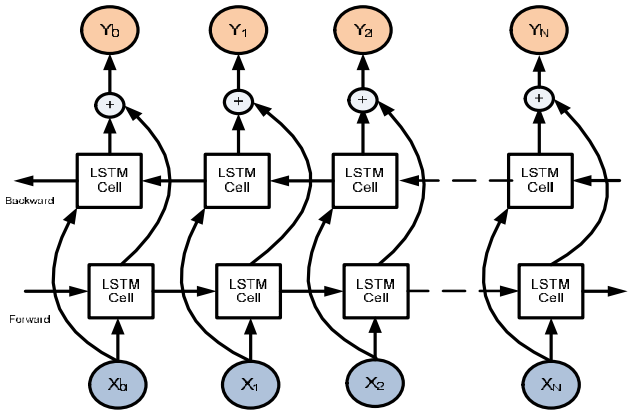

FIGURE 2. General architecture of bidirectional LSTM RNN.

In this study, a multi-layer bidirectional LSTM recurrent neural network is being used in order to utilize previous and future context of a speech frame. Each layer of the bidirectional LSTM network consists of two separate hidden layers i.e. forward layer and backward layer. The first one takes input sequences as-is and the second one the reversed copy of the sequence. Output values from these separate layers are concatenated to generate the output y^t as illustrated in Fig. 2.

Multiple layers of LSTM RNN are stacked on top of each other resulting in a deep network architecture. Output sequence of one layer serves as the input sequence of the next layer, ensuring that the next layer receives input from both backward and forward layers of the level below, as illustrated in Fig. 3.

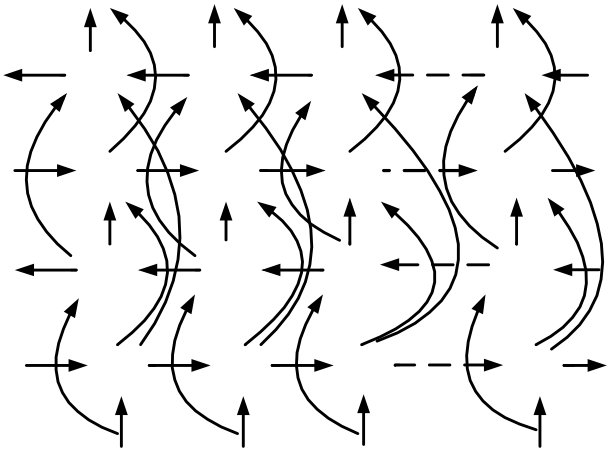

FIGURE 3. General architecture of deep bidirectional LSTM RNN.

Output layer of the network is configured as Softmax, to map input x_j to a class probability P_j defined as

$$P_j = \exp(x_j) \Big/ \sum_i \exp(x_i) \tag{16}$$

where i is an index over all the classes. Output layer of size five is used (one for each native language). Cross Entropy (CE) function is used as a cost function for backpropagating gradients in the training stage, defined as

$$C = -\sum_j t_j \log P_j \tag{17}$$

where t_j denotes the target probability of the class j against the current utterance.

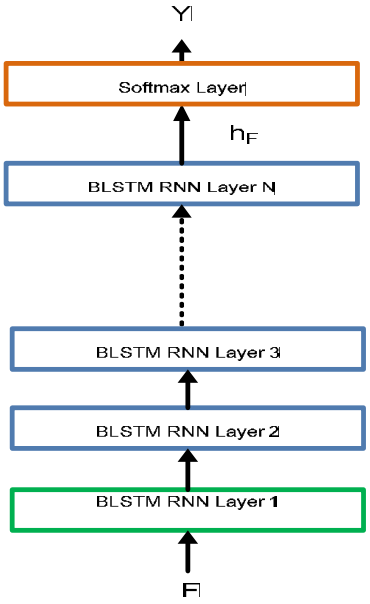

FIGURE 4. Architecture of BLSTM network.

Generic architecture of BLSTM network shown in Fig. 4 is used to develop MFCC based NLI system having input F as F_M. Similarly, GFCC features based NLI system is also developed by using input feature vector F_G. N is the number of layers which is optimized experimentally for each system. In addition, several experiments are carried out to optimize number of hidden units and regularization methods. Details of these experiments are provided in subsequent sections.

A. NUMBER OF HIDDEN LAYERS

Theoretically, an adequately wide neural network with only one hidden layer can approximate any function when trained on a sufficient amount of data [36]. But, an extensively wide network may end up memorizing the corresponding output value which is not useful for practical applications because every input value may not be part of the training data. Multiple hidden layers are better because they can learn hierarchical, more complex internal representations. Therefore, a number of experiments are conducted to find out the optimal depth i.e. number of hidden layers. During these experiments, hidden units are fixed to 256 for each forward and backward direction LSTM. Dropout of 0.2 and L2 input weight regularization of 0.01 is applied at each layer.

B. SIZE OF HIDDEN LAYERS

The number of hidden units in neural network can influence its performance significantly. Fewer hidden units can cause under-fitting result in high errors on training and validation data. On the other hand, a large number of units

may cause over-fitting and can result in higher testing error. Optimal numbers of units are required to minimize the effect of under-fitting and over-fitting. Different studies have been carried out to find the various rules for the determination of optimal number of units of the different layers of a neural network [37]. In most of the cases, researchers optimized the neural network with different configuration in terms of number of hidden units and layers. Nagendra and Kher [38] suggest to obtain the best number by iteratively adjusting the number of neurons while considering the error during neural network testing. We used iterative adjustment of number of hidden units, started from the lowest number i.e. 128 and gradually increased upwards in power of two up to 600 due to memory limitations.

C. REGULARIZATION METHODS

In deep learning, regularization methods are helpful to decrease model complexity by minimizing weights' values, since small values result in smoother hypothesis functions. In this study, two regularization methods are used: dropout [39] and weight regularization i.e. standard L2 regularization. L2 regularization is applied to input connection of each LSTM unit at each layer of the network. Different dropout and L2 regularization combinations are further investigated using grid search to find the optimal configuration. Dropout values are considered in the range of 0.0-0.5 with increment of 0.1, whereas L2 regularization values in the range of 0.00-0.02 with increment of 0.01 are used in experiments. Different combinations are tried and model performance is evaluated on validation data.

V. MERGED BLSTM RNN MODEL

Wide variety of acoustic features is available with different strengths and weaknesses. A combination of several different features as an input to the single system may result in more accurate results. Different approaches can be used to concatenate the features, (1) concatenation of computed features from input speech, (2) concatenation of prioritized features by applying different speech processing and machine learning techniques on computed raw features. Li and Stern [40] combined MFCC and perceptual linear prediction (PLP) features to develop GMM-HMM based ASR system. The combined features reduced word error rate (WER) significantly. Similarly, a study by Tjandra et al. [41] showed that convolution neural network(CNN) with concatenated MFCC and GFCC features performed better as compared to CNN with one feature set. Zhang *et al.* [42] used a combination of different features including MFCC, PLP, linear frequency cepstral coefficients (LFCC) and GFCC for language recognition. It is observed that combined feature sets result in lower equal error rate (EER). These studies show that combination of feature sets perform better. In addition, combination of processed feature set (with the help of machine learning algorithm) improves system performance significantly as compared to the system having combined feature set directly. Therefore, in this study an NLI system is also developed in which feature sets are processed and combined using architecture shown in Fig. 5.

In this architecture, two independent BLSTM neural networks are used to process MFCC-SDC and GFCC-SDC features separately. The optimized BLSTM networks (discussed in Section VIII) are used and merged together using the merge layer. These two independent BLSTMs models return their final output sequence i.e. $\boldsymbol{h_M}$ and $\boldsymbol{h_G}$, thus dropping the temporal dimension (i.e. converting the input sequence into a single vector). These two vectors are concatenated using the formula given below:

$$H_F = h_M h_G \tag{18}$$

where h_M is output of optimized BLSTM trained on MFCC feature set (F_M) and h_G is the output of the optimized BLSTM trained on GFCC feature set (F_G). H_F is the feature set having concatenated h_M and h_G, i.e.

$$H_F = \{h_{m1}, h_{m2}, h_{m3} \ldots, h_{m56} h_{g1}, h_{g2}, h_{g3}, \ldots, h_{m56}\}$$

This $\boldsymbol{H_F}$ vector is forwarded to fully connected layers, to learn mappings from both high level feature vectors to the output classes, as shown in Fig. 5. The last layer of this network is a softmax layer which outputs probabilities for each class. In Fig. 5 K and L represents number of hidden layers for MFCC based BLSTM network and GFCC based BLSTM network, respectively.

Similar to the optimization of independent BLSTM models, a numbers of experiments are carried out to find the optimal number of fully connected layers i.e. N and size of layers for merged BLSTM RNN model.

VI. DATASET

Single word utterances speech corpus [9] is used for the NLI of an Urdu speaker. This speech corpus consists of Pakistan's district names (139 district names) recorded from more than 300 speakers, with 11 different L1 backgrounds. It is comprised of about 15 hours of speech, sampled at 8 KHz, collected in different background noises, with varying qualities of mobile phones and network operators. Data is publically available at CLE store.[1] More details of dataset are available in [9].

Although, the speech corpus contains recorded speech from speakers of 11 different L1 backgrounds, only five languages have a sufficient amount of data. Selected corpus is comprised of 10,261 speech utterances (from 0.27s to 2s) recorded from speakers having L1 as Balochi (bal), Pashto (pus), Punjabi (pan), Saraiki (skr) and Sindhi (snd), on average each utterance is 0.8 second long.

Selected data is randomly divided into three sets i.e. training (train), validation (val) and testing (test) with non-overlapping speakers and their utterances.

Training set contains 7517 utterances (80% of dataset) and validation set comprises 1370 speech samples

[1] http://www.cle.org.pk/clestore/speechcorpus.htm

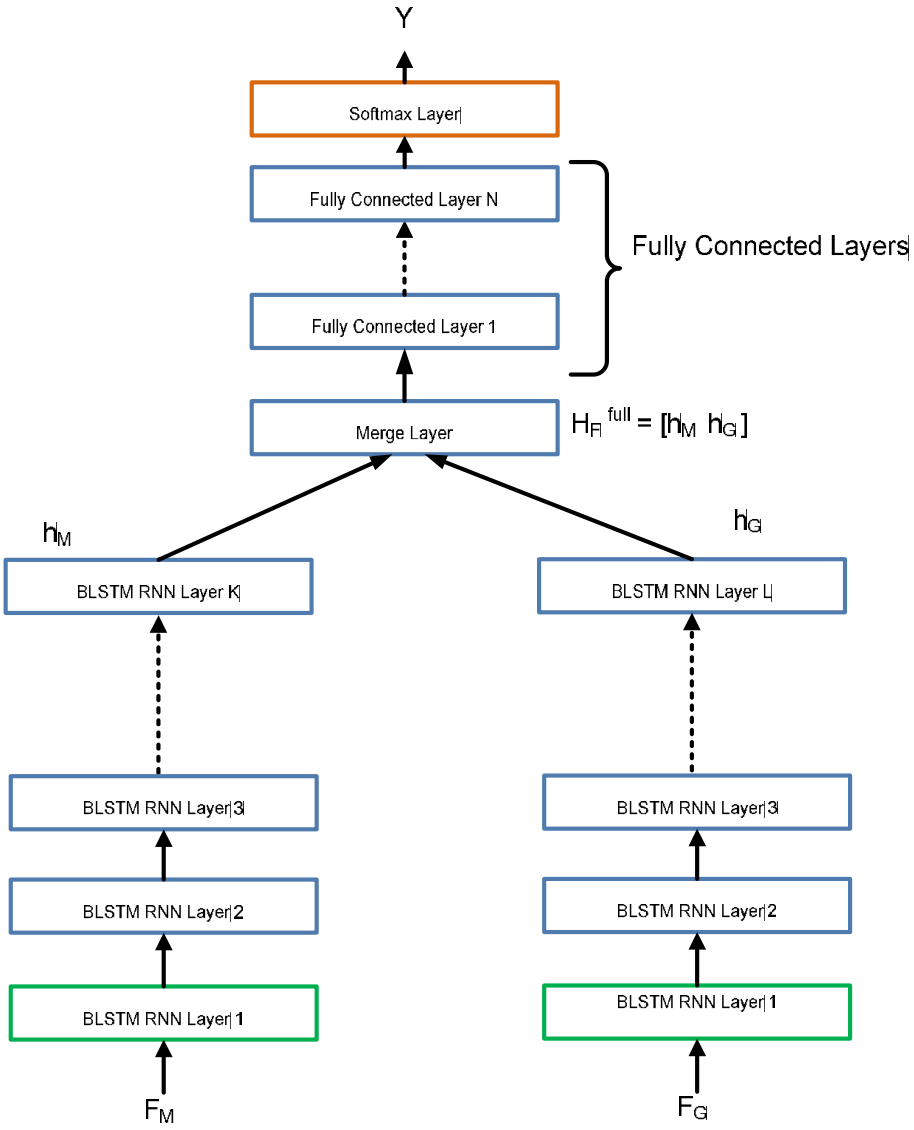

FIGURE 5. Architecture of merged BLSTM RNN.

TABLE 1. Language wise training, validation and test data distribution.

Language	Training Utterances	Training Speakers	Validation Utterances	Validation Speakers	Testing Utterances	Testing Speakers
Balochi (bal)	1507	213	274	61	274	130
Punjabi (pan)	1499	459	274	126	275	155
Pashto (pus)	1502	324	274	136	275	143
Saraiki (skr)	1504	535	274	119	277	195
Sindhi (snd)	1505	193	274	72	273	119

(10% of dataset), whereas test data consists of 1374 utterances (10% of dataset). It is ensured that instances of each vocabulary item should be present in each set i.e. training, validation and testing. Moreover, instances of each vocabulary item are randomly distributed according to the defined ratio i.e. 80:10:10. Language wise data distribution is described in Table 1.

VII. EXPERIMENTAL SETUP

Each BLSTM-RNN network is optimized using Adam optimizer [43], a widely used optimization method with an initial learning rate of 10^3. Several experiments are carried out to find a best architecture in terms of number of layers, size of hidden layers and regularization methods. For all GFCC feature-based experiments, 60 epochs are used for training of

the network, whereas for MFCC feature-based experiments 30 epochs are used. Each network configuration is evaluated using the validation set and the model with the best validation accuracy is used for later experiments. In merged BLSTM models approach, ReLU is used for activation of fully connected layers and dropout of 0.4 is applied at each layer.

BLSTM-RNN models are implemented in Python, using the Keras [44] neural network library, running on top of Tensorflow. Experiments are conducted on a machine with NVIDIA GeForce GTX 1080 GPU with 8GB of memory and 32GB RAM.

VIII. RESULTS & DISCUSSION

A number of experiments are carried out for i-vector model, BLSTM-RNN model using MFCC and GFCC as input features and model accuracy on validation data is calculated. In addition, network architecture search is also performed for merged BLSTM models. Details of experiments are given in subsequent sections.

A. I-VECTOR MODEL RESULTS

In this study, different configurations of i-vector are explored in terms of number of Gaussian components and i-vector length and performance of NLI system is evaluated using test data in terms of accuracy. Combination of i-vector length and Gaussian components are shown in the form given below.

$$\text{i-vector}_{vec\_length}\text{\_GMM}_{components}$$

Whereas

$$vec\_length = \{400, 600, 800\}$$
$$components = \{64, 128, 256, 1024\}$$

Table 2 summarizes i-vector based NLI system performance in terms of accuracy with different configurations of Gaussian components and i-vector dimensions.

TABLE 2. I-vector system accuracy.

Configuration	MFCC	GFCC
i-vector400_GMM64	56.75	57.96
i-vector400_GMM128	56.81	60.09
i-vector400_GMM256	57.05	59.00
i-vector400_GMM1024	56.44	55.59
i-vector600_GMM128	56.14	60.46
i-vector800_GMM128	56.44	**60.88**

It is evident from Table 2 that maximum accuracy of 60.88% is achieved using i-vector of length 800 with 128 Gaussian components.

B. BLSTM RNN NETWORK ARCHITECTURE SEARCH

1) NUMBER OF HIDDEN LAYERS

Results of experiments for number of hidden layers are shown in Table 3. During experiments, the effect of number of hidden layers in terms of training accuracy (train Acc), validation accuracy (val Acc), training loss (train loss) and validation loss (val loss) is also examined. Validation loss shows the value obtained from same epoch as the validation accuracy, whereas training loss is obtained from the last epoch.

It is observed that training and validation accuracy of MFCC feature-based BLSTM network increases with the use of more than one hidden layers and highest accuracy is achieved with three hidden layers. Highest validation accuracy of GFCC feature-based BLSTM network is achieved with two hidden layers. Therefore, for later experiments three hidden layers will be used for MFCC feature-based BLSTM and two hidden layers will be used for GFCC feature-based BLSTM network.

2) SIZE OF HIDDEN LAYERS

After finding out the number of hidden layers, experiments are carried out to find out the optimal number of neurons in each layer. During experiments equal numbers of neurons are used at each layer. Three layers are used in MFCC feature-based BLSTM and two layers are used in GFCC feature-based BLSTM. Dropout of 0.2 and L2 input weight regularization of 0.01 is applied at each layer. Table 4 shows the results of comparison of hidden layer size. It is observed that with larger number of neurons, the number of trainable parameters rises and the validation accuracy start deteriorating. It is observed that three layers BLSTM model with 256 neurons performs better for MFCC features set and BLSTM model of two layers with 512 neurons yields higher validation accuracy for GFCC feature set. This network configuration is further improved by using the dropout and L2 regularization.

3) REGULARIZATION METHODS

Table 5 shows validation accuracy against different configurations of dropout and L2. It is observed that dropout of 0.4 with L2 of 0.00 yields the best result with MFCC as feature vector, whereas for GFCC feature set, dropout of 0.2 with L2 of 0.01 yields best validation accuracy.

The NLI accuracy attained by optimized MFCC feature-based BLSTM model on the training and validation data at different training epochs is shown in Fig. 6. Similarly, optimized GFCC feature-based BLSTM model accuracy on the training and validation data at different training epochs is illustrated in Fig. 7. Loss curves of MFCC and GFCC feature-based models are shown in Fig. 8 and Fig. 9, respectively. A comparison between the accuracy curves suggests that MFCC feature-based network is faster to train and yields highest validation accuracy than GFCC feature-based network. It also suggests that there is slightly more over-fitting in MFCC feature-based model than the GFCC feature-based network even with higher dropout and L2 regularization values.

C. MERGED BLSTM RNN MODELS ARCHITECTURE SEARCH

After the network optimization of MFCC feature-based BLSTM model and GFCC feature-based BLSTM model, their outputs are combined. Outputs of BLSTM models

TABLE 3. Effect of number of hidden layers on training and validation accuracy using MFCC and GFCC features, BLSTM network with 256 neurons/layer and dropout of 0.2, L2 of 0.01.

#Layers	#Params[M]	Feature Set							
		MFCC				GFCC			
		val Acc(%)	train Acc(%)	train loss	val loss	val Acc(%)	train Acc(%)	train loss	val loss
1	0.6	70.36	96.34	0.116	1.14	66.13	82.31	0.71	0.932
2	2.2	71.17	95.18	0.31	0.88	**69.34**	83.51	0.51	1.034
3	3.7	**71.46**	97.15	0.10	1.55	69.27	83.32	0.48	1.018
4	5.3	71.27	95.68	0.15	1.08	63.35	70.00	0.83	1.001

TABLE 4. Validation data accuracy with different number of hidden units, 3 layers are used in MFCC feature-based BLSTM network and 2 layers are used in GFCC feature-based BLSTM network, dropout of 0.2, L2 of 0.01 is applied at each layer of both models.

# Hidden Units	# Params[M]		val Acc (%)	
	MFCC	GFCC	MFCC	GFCC
128	0.97	0.58	70.22	68.54
256	3.79	2.2	**71.46**	69.34
512	14.92	8.6	66.35	**71.61**
600	20.44	11.80	68.98	71.51

TABLE 5. Validation accuracy (%) with different combinations of dropout and L2, MFCC feature-based BLSTM network with 3 layers and 256 neurons/layer, GFCC feature-based BLSTM network with 2 layers and 512 neurons/layer.

Dropout	L2					
	0.00		0.01		0.02	
	MFCC	GFCC	MFCC	GFCC	MFCC	GFCC
0	68.03	68.83	67.30	71.09	67.08	70.88
0.1	70.8	65.84	71.16	70.43	70.29	70.80
0.2	71.67	69.92	71.46	**71.61**	71.89	70.43
0.3	74.3	68.69	72.18	62.56	73.13	71.09
0.4	**74.81**	70.72	73.5	59.56	73.72	61.24
0.5	72.91	67.87	72.18	70.29	73.43	65.91

TABLE 6. Effect of number of fully connected layers in merged models on validation accuracy, layer size 256, dropout of 0.3.

#Layers	val Acc (%)	train Acc (%)	train loss	val loss
1	72.85	85.97	0.56	0.97
2	73.13	86.84	0.42	0.96
3	**73.80**	**85.06**	**0.51**	**0.92**
4	71.75	83.25	0.55	0.95

trained on MFCC and GFCC features are merged and forwarded to the fully connected layers as shown in Fig. 5. Experiments are conducted to find out the optimal number of fully connected layers and size of layers. Experiments results are provided in subsequent sections.

1) NUMBER OF FULLY CONNECTED LAYERS

Results of experiments for number of fully connected layers are given in Table 6. It is evident from Table 6 that best performance is achieved with three fully connected layers. Experiments are performed with fully connected

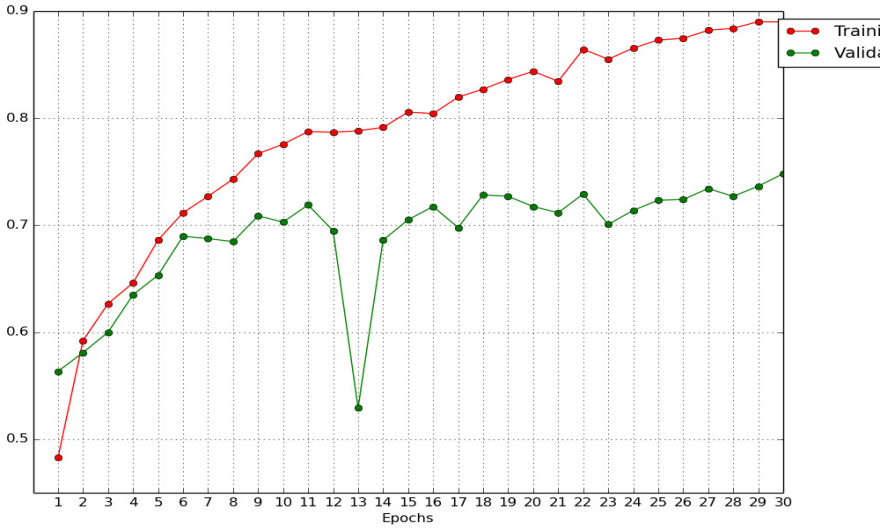

FIGURE 6. MFCC feature-based BLSTM-RNN model accuracy on the training and validation data over 30 Epochs, 3 layers BLSTM model with 256 neurons/layer, dropout of 0.4 and L2 of 0.00.

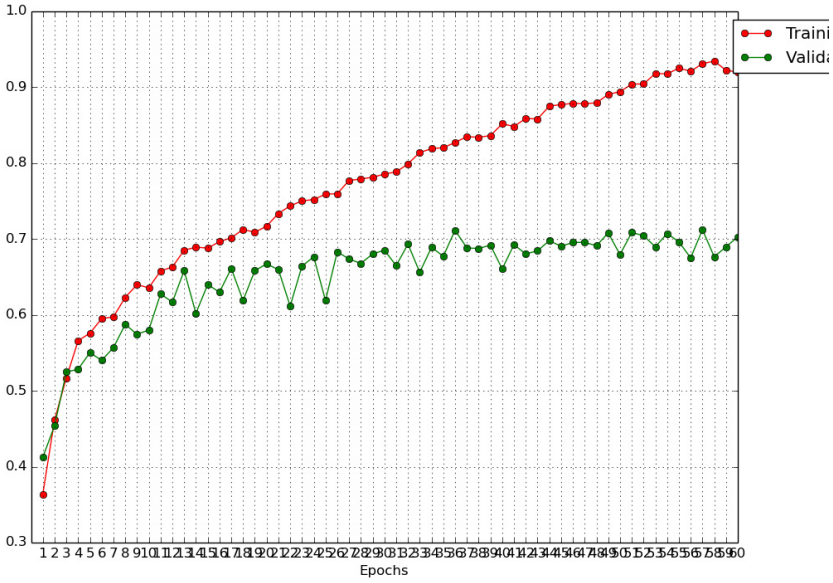

FIGURE 7. GFCC feature-based BLSTM-RNN model accuracy on the training and validation data over 60 Epochs, 2 layers BLSTM model with 512 neurons/layer, dropout of 0.2 and L2 of 0.01.

layers of 256 neurons and dropout of 0.3 is applied at each layer.

2) SIZE OF FULLY CONNECTED LAYERS

After finalization of number of fully connected layers, experiments with three layers and varying layers size are conducted and validation accuracy is computed. During experiments, equal numbers of neurons are used at each layer. Table 7 shows the validation accuracy for merged BLSTM trained with different sizes of fully connected layers along with number of parameters (in millions). It is observed that merged BLSTM models network with three fully connected layers of size 256 yields the highest validation accuracy.

TABLE 7. Comparison of fully connected layer size.

#Size of layer	# Params [M]	val Acc(%)
128	12.64	72.91
256	12.94	**75.69**
512	13.73	72.04

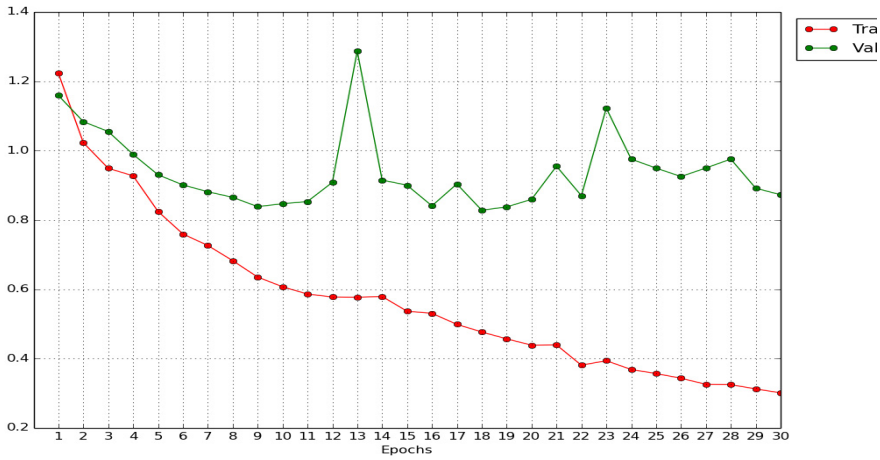

FIGURE 8. MFCC feature-based BLSTM model loss over 30 epochs.

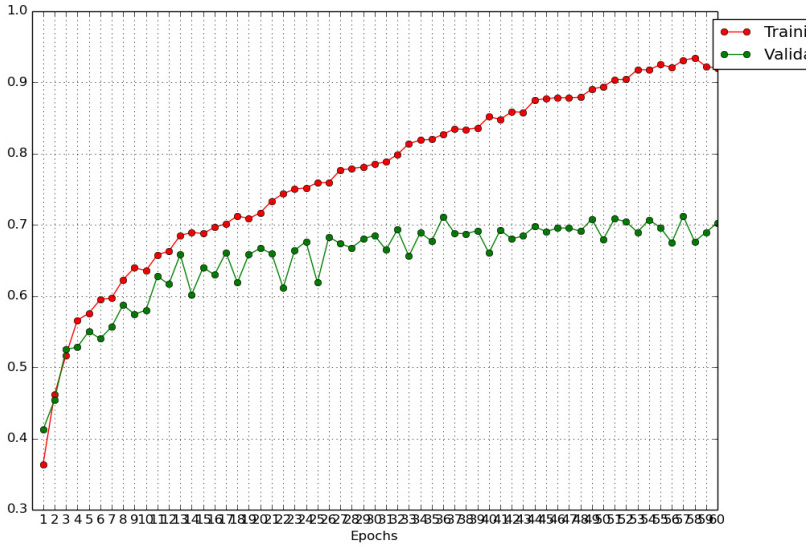

FIGURE 9. GFCC feature-based BLSTM model loss over 60 epochs.

TABLE 8. Confusion matrix for the test set using the merged BLSTM models, rows are reference and columns are hypothesis.

	bal	pan	pus	skr	snd
bal	**171**	22	26	34	21
pan	40	**145**	61	18	11
pus	8	18	**242**	3	4
skr	46	8	1	**206**	16
snd	51	20	5	26	**171**

The best accuracy achieved on the validation data is 75.69% by merging 3-layers MFCC feature-based BLSTM model, 2-layers GFCC feature-based BLSTM model in a layer and using 3 fully connected hidden layers (excluding merge and output layer).

TABLE 9. Recall for each class and the unweighted average recall (UAR) on the test set (%).

bal	pan	pus	skr	snd	UAR
62	53	88	74	63	68

This model is later evaluated on the test set and an accuracy of 68.05 % is achieved. It is observed that proposed model outperforms the i-vector model.

Fig. 10 shows the relationship between system accuracy and duration of test utterances. Test data set is divided into six groups on the basis of utterance duration and accuracy of each group is computed. It is evident from Fig. 10 that system accuracy increases with the duration of test utterances so a reliable system can be developed when longer test utterances are available.

Confusions among languages are shown as a confusion matrix in Table 8 and recalls in Table 9. It's evident from

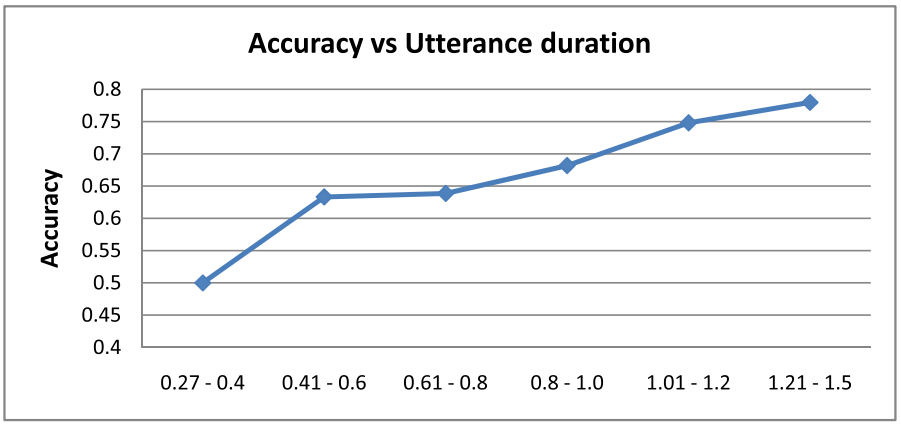

FIGURE 10. Merged BLSTM model performance on very short utterances.

Table 8 that Pashto (pus) language is identified more correctly than the rest of the languages and confusions are most frequent among Sindhi and Balochi.

IX. CONCLUSION

In this paper, we explored the use of spectrogram and cochleagram features for native language identification from very short speech utterances (0.8s on average) for Urdu (L2) speakers. Bidirectional long short-term memory (BLSTM) models are adopted to solve this complex problem of NLI for limited duration speech data. Several configurations of BLSTM models are explored and compared. This study indicates that MFCC features are more robust than GFCC features for speech data recorded in various acoustical environments, with various quality mobile phones and network operators. Results of BLSTM model are compared with state-of-the-art i-vector model and it is observed that BLSTM model performs better as compared to the i-vector model.

In addition, BLSTM models trained using MFCC and GFCC features are also merged together to take advantage of both feature sets. We found that the merged models approach outperforms the individual models. However, by looking at the confusion matrix given in Table 8, it is observed that the system confuses among languages which are acoustically and geographically close, such as Punjabi and Pashto and Sindhi and Balochi.

In the future, it will be worthwhile to investigate prosodic features to improve the performance of the system. In addition, hierarchical classification can be adopted to initially classify languages into family groups and then make fine-grained between them.

REFERENCES

[1] S.-M. J. Wong and M. Dras, "Exploiting parse structures for native language identification," in *Proc. Conf. Empirical Methods Natural Lang.* Edinburgh, U.K.: Association for Computational Linguistics, 2011, pp. 1600–1610.

[2] R. T. Ionescu, M. Popescu, and A. Cahill, "Can characters reveal your native language? A language-independent approach to native language identification," in *Proc. EMNLP*, 2014, pp. 1363–1373.

[3] V. Kríž, M. Holub, and P. Pecina, "Feature extraction for native language identification using language modeling," in *Proc. RANLP*, 2015, pp. 298–306.

[4] J. Lopes, I. Trancoso, and A. Abad, "A nativeness classifier for TED talks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 5672–5675.

[5] B. W. Schuller *et al.*, "The INTERSPEECH 2016 computational paralinguistics challenge: Deception, sincerity & native language," in *Proc. INTERSPEECH*, 2016, pp. 2001–2005.

[6] J. R. Tetreault, J. Burstein, and C. Leacock, "A report on the first native language identification shared task," in *Proc. Building Educ. Appl. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (BEA @NAACL-HLT)*, J. R. Tetreault, J. Burstein, and C. Leacock, Eds. Stroudsburg, PA, USA: The Association for Computational Linguistics, 2013, pp. 48–57.

[7] D. Blanchard, "TOEFL11: A corpus of non-native English," Educ. Test. Service, Princeton, NJ, USA, ETS Res. Rep. i-15, 2013, vol. 2.

[8] S. Malmasi, M. Dras, and I. Temnikova, "Norwegian native language identification," in *Proc. Int. Conf. Recent Adv. Natural Lang. Process. (RANLP)*, 2015, pp. 404–412.

[9] S. Rauf, A. Hameed, T. Habib, and S. Hussain, "District names speech corpus for pakistani languages," in *Proc. Int. Oriental Conf. COCOSDA/CASLRE*, Shanghai, China, Oct. 2015, pp. 207–211.

[10] A. Rajpal, T. B. Patel, H. B. Sailor, M. C. Madhavi, H. A. Patil, and H. Fujisaki, "Native language identification using spectral and source-based features," in *Proc. INTERSPEECH*, San Francisco, CA, USA, 2016, pp. 1–5.

[11] M. Senoussaoui, P. Cardinal, N. Dehak, and A. L. Koerich, "Native language detection using the I-vector framework," in *Proc. INTERSPEECH*, 2016, pp. 1–5.

[12] N. Dehak, P. J. Kenny, R. Dehak, D. Pierre, and O. Pierre, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[13] A. Rosenberg, "Classifying skewed data: Importance weighting to optimize average recall," in *Proc. INTERSPEECH*, 2012, pp. 2242–2245.

[14] A. Abad, E. Ribeiro, F. Kepler, R. Astudillo, and I. Trancoso, "Exploiting phone log-likelihood ratio features for the detection of the native language of non-native English speakers," in *Proc. INTERSPEECH*, 2016, pp. 2242–2245.

[15] M. A. Huckvale, "Within-speaker features for native language recognition in the interspeech 2016 computational paralinguistics challenge," in *Proc. INTERSPEECH*, San Francisco, CA, USA, 2016, pp. 2403–2407.

[16] D. Martínez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in iVectors space," in *Proc. INTERSPEECH*, Firenze, Italy, 2011, pp. 861–864.

[17] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via I-vectors and dimensionality reduction," in *Proc. INTERSPEECH*, 2011, pp. 857–860.

[18] M. H. Bahari, N. Dehak, H. Van hamme, L. Burget, A. M. Ali, and J. Glass, "Non-negative factor analysis of Gaussian mixture model weight adaptation for language and dialect recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 7, pp. 1117–1129, Jul. 2014.

[19] Q. Zhang, H. Boril, and J. H. L. Hansen, "Supervector pre-processing for PRSVM-based Chinese and Arabic dialect identification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Nov. 2013, pp. 7363–7367.

[20] P. Matejka, P. Schwarz, J. Cernock, and P. Chytil, "Phonotactic language identification using high quality phoneme recognition," in *Proc. Eurospeech*, 2005, pp. 2237–2240.

[21] L. Mary and B. Yegnanarayana, "Prosodic features for language identification," in *Proc. Int. Conf. Signal Process., Commun. Netw. (ICSCN)*, Jan. 2008, pp. 57–62.
[22] R. W. M. Ng, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Analysis and selection of prosodic features for language identification," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Dec. 2009, pp. 123–128.
[23] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digit. Signal Process.*, vol. 10, nos. 1–3, pp. 19–41, 2000.
[24] N. Dehak, P. Dumouchel, and P. Kenny, "Modeling prosodic features with joint factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 7, pp. 2095–2103, Sep. 2007.
[25] A. Lozano-Diez, R. Zazo, D. T. Toledano, and J. Gonzalez-Rodriguez, "An analysis of the influence of deep neural network (DNN) topology in bottleneck feature based language recognition," *PLoS ONE*, vol. 12, no. 8, 2017, Art. no. e0182580.
[26] 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 Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 5337–5341.
[27] R. Zazo, A. Lozano-Diez, J. Gonzalez-Dominguez, D. T. Toledano, and J. Gonzalez-Rodriguez, "Language identification in short utterances using long short-term memory (LSTM) recurrent neural networks," *PLoS ONE*, vol. 11, no. 1, 2016, Art. no. e0146917.
[28] L. J. Rodríguez-Fuentes *et al.*, "Multi-site heterogeneous system fusions for the Albayzin 2010 language recognition evaluation," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2011, pp. 377–382.
[29] Y. Jiao, M. Tu, V. Berisha, and J. Liss, "Accent identification by combining deep neural networks and recurrent neural networks trained on long and short term features," in *Proc. INTERSPEECH*, 2016, pp. 1–5.
[30] D. R. González and J. R. C. de Lara, "Speaker verification with shifted delta cepstral features: Its pseudo-prosodic behavior," in *Proc. I Iberian SLTech*, 2010, pp. 1–4.
[31] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller, Jr., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," in *Proc. INTERSPEECH*, 2002, pp. 89–92.
[32] M. Van Segbroeck, R. Travadi, and S. S. Narayanan, "Rapid language identification," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 7, pp. 1118–1129, Jul. 2015.
[33] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," in *Proc. Odyssey Speaker Lang. Workshop*, 2006, pp. 210–229.
[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[35] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul./Aug. 2005, pp. 2047–2052.
[36] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," *CoRR*, vol. abs/1512.03965, pp. 1–33, May 2015.
[37] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization," *Neural Netw.*, vol. 22, no. 10, pp. 1448–1462, 2009.
[38] S. M. S. Nagendra and M. Khare, "Artificial neural network approach for modelling nitrogen dioxide dispersion from vehicular exhaust emissions," *Ecolog. Model.*, vol. 190, nos. 1–2, pp. 99–115, 2006.
[39] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, Aug. 2012.
[40] X. Li, *Combination and Generation of Parallel Feature Streams for Improved Speech Recognition*. Pittsburgh, PA, USA: Carnegie Mellon Univ., 2005, p. 125.
[41] A. Tjandra, S. Sakti, G. Neubig, T. Toda, M. Adriani, and S. Nakamura, "Combination of two-dimensional cochleogram and spectrogram features for deep learning-based ASR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4525–4529.
[42] Q. Zhang, G. Liu, and J. H. Hansen, "Robust language recognition based on diverse features," in *Proc. ODYSSEY, Speaker Lang. Lang. Recognit. Workshop*, 2014, pp. 152–157.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, pp. 1–15, Jan. 2014.
[44] F. K. Chollet. *Keras: Deep Learning for Humans*. (2017). [Online]. Available: https://github.com/keras-team/keras
[45] P. Barua, K. Ahmad, A. A. S. Khan, and M. Sanaullah, "Neural network based recognition of speech using MFCC features," in *Proc. Int. Conf. Inform., Electron. Vis. (ICIEV)*, May 2014, pp. 1–6.
[46] M. Sanaullah and K. Gopalan, "Distinguishing deceptive speech from truthful speech using MFCC," in *Proc. 7th Int. Conf. Circuits, Syst. Signals*, 2013, pp. 167–171.
[47] S. Malmasi *et al.*, "A report on the 2017 native language identification shared task," in *Proc. 12th Workshop Innov. Use NLP Building Educ. Appl.*, 2017, pp. 62–75, doi: 10.18653/v1/W17-5007.
[48] S. Jarvis, Y. Bestgen, and S. Pepper, "Maximizing classification accuracy in native language identification," in *Proc. 8th Workshop Innov. Use NLP Building Educ. Appl.* Atlanta, Georgia: Association for Computational Linguistics, 2013, pp. 111–118.
[49] A. Cimino and F. Dell'Orletta, "Stacked sentence-document classifier approach for improving native language identification," in *Proc. 12th Workshop Innov. Use NLP Building Educ. Appl.* Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 1–8.
[50] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 5, pp. 980–988, Jul. 2008, doi: 10.1109/TASL.2008.925147.
[51] M. Qasim, S. Nawaz, S. Hussain, and T. Habib, "Urdu speech recognition system for district names of pakistan: Development, challenges and solutions," in *Proc. 19th Oriental Conf. COCOSDA*, Bali, Indonesia, Oct. 2016, pp. 28–32. [Online]. Available: http://www.ococosda2016.org/
[52] O. A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *Proc. INTERSPEECH*, 2006, pp. 1–4.
[53] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 1, May 2006, p. 1.
[54] P. Kenny, "Bayesian speaker verification withheavy-tailed priors," in *Proc. Odyssey, Speaker Lang. Recognit. Workshop*, Brno, Czech Republic, Jun. 2010, Paper 14.
[55] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of I-vector length normalization in speaker recognition systems," in *Proc. INTERSPEECH*, Florence, Italy, Aug. 2011, pp. 249–252.

FARAH ADEEBA received the M.Sc. degree in computer science from the National University of Computer and Emerging Sciences, Lahore, Pakistan, in 2011. She is currently pursuing the Ph.D. degree in spoken language identification with the University of Engineering and Technology, Lahore, where she has been with the Center for Language Engineering, since 2011. Her research interests are in speech recognition, and text-to-speech synthesis.

SARMAD HUSSAIN received the Ph.D. degree in speech science from Northwestern University, Evanston, IL, USA, in 1997, and the M.Phil. degree in computer speech and language processing from Cambridge University, U.K., in 1993. He also worked at the Oxford University Phonetics Lab, during his doctoral program, where he was involved on phonological aspects of stress in Urdu. He is currently a Professor of computer science, and also the Head of the Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore. His research is focused on developing computing solutions for Pakistani languages, including research in linguistics, localization, language computing standards, speech processing, and computational linguistics. He has been serving on many national and international committees. Some of his current international memberships included the Security and Stability Advisor Committee of ICANN, the Executive Committee of Asian Federation of Natural Language Processing, and the Pakistan representative on the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques.