Derin Öğrenme Mimarilerinde Akustik ve Fonotaktik Öznitelikleri Kullanan Türkçe Ağız Tanıma

Araştırma Makalesi/Research Article

Gültekin IŞIK[1]*, Harun ARTUNER[2]

[1]Bilgisayar Mühendisliği, Iğdır Üniversitesi, Iğdır, Türkiye
[2]Bilgisayar Mühendisliği, Hacettepe Üniversitesi, Ankara, Türkiye
gultekin.isik@igdir.edu.tr, artuner@hacettepe.edu.tr
(Geliş/Received:31.12.2019; Kabul/Accepted:05.05.2020)
DOI: 10.17671/gazibtd.668023

Özet— Ağızlar, standart dilden belli oranda ayrılan yerel konuşma biçimleridir. Ağız tanıma, konuşma tanıma alanında çalışılan popüler konular arasındadır. Özellikle, büyük ölçekli konuşma tanıma sistemlerinin başarımlarını arttırmak için konuşmanın ağzının öncelikli olarak belirlenmesi istenmektedir. Konuşmanın fonetik farklılıkları, fiziksel düzeyde akustik özellikleri incelenerek tespit edilebilmektedir. Log mel-spektrogram gibi öznitelikler bu amaçla kullanılmaktadır. Bununla birlikte, fonotaktik terimi, bir dilde/ağızda, fonemlerin bir araya gelme kurallarına karşılık gelmektedir. Fonem dizilimleri ve bu dizilimin sıklığı ağızdan ağza değişiklik göstermektedir. Fonem dizilimleri fonem tanıyıcılar yardımıyla elde edilmektedir. Son yıllarda popüler olan diğer bir konu derin öğrenme sinir ağlarıdır. Derin öğrenme sinir ağlarının özel bir çeşidi olan Evrişimli Sinir Ağları (CNN) özellikle görüntü ve konuşma tanımada sıklıkla kullanılmaktadır. Uzun Kısa-Dönem Bellekli Sinir Ağları (LSTM), dil modellemede n-gram modellerden daha başarılı sonuçlar üreten bir derin öğrenme sinir ağı modelidir. Bu çalışmada Türkçe ağızların akustik ve fonotaktik özellikleri bakımından CNN ve LSTM-türü sinir ağlarıyla sınıflandırılması ele alınmıştır. Ayrıca LSTM sinir ağları fonotaktik yaklaşımda dil modelleme için kullanılmıştır. Deneysel çalışmada önerilen yaklaşımlar, tarafımızca toplanan Türkçe Ağızlar Veri Kümesi üzerinde kullanılarak sınanmış ve yorumlanmıştır. Çalışma sonucunda, kullanılan yaklaşımların Türkçe ağız tanıma için %85,1 doğruluk oranı verdiği gözlenmiştir.

Anahtar Kelimeler— türkçe ağız tanıma, evrişimli sinir ağları, akustik ve fonotaktik, log mel-spektrogram, yinelemeli sinir ağları dil modelleri

Turkish Dialect Recognition Using Acoustic and Phonotactic Features in Deep Learning Architectures

Abstract— Dialects are local forms of speech separated by a certain rate from a standard language. Dialect recognition is one of the popular topics studied in speech recognition. In particular, the spoken dialect is asked to be identified first in order to improve the performance of large scale speech recognition systems. The phonetic differences of speech can be determined by examining the acoustic properties at the physical level. Features such as Log mel-spectrograms are used for this purpose. In addition, the phonotactic term corresponds to the arrangement rules of phonemes in a language/dialect. Phoneme sequences and the frequency of this sequence vary from dialect to dialect. Phoneme sequences are obtained by phoneme recognizers. Another topic that has become popular in recent years is deep learning neural networks. Convolutional Neural Networks (CNN), which is a special kind of deep learning neural networks, are often used in image and speech recognition. Long Short-Term Memory Neural Networks (LSTM) is a deep learning neural network model that produces more successful results than n-gram models in language modeling. In this study, the classification of Turkish dialects with CNN and LSTM type neural networks in terms of acoustic and phonotactic features were discussed. Also, LSTM neural networks are used for language modeling in phonotactic approach. In the experimental study, the proposed approaches were tested and interpreted on the Turkish Dialects Dataset that we collected. As a result of the study, it has been observed that the approaches used reaches 85.1% accuracy rate for Turkish dialect recognition.

Keywords— turkish dialect recognition, convolutional neural networks, acoustics and phonotactics, log mel-spectrogram, recurrent neural network language models

1. INTRODUCTION

Spoken language recognition is simply a process of identifying the language of a given speech data. The distinction of spoken languages is a natural skill for people. Simulating of this ability in the computer environment constitutes the subject of this study. People recognize languages as the result of perceptual processes in the hearing system. Perceptual cues used by people inspired automatic spoken language recognition studies [1]. Most of the studies conducted in computer environment are mostly originated from language recognition through text. The text-based language recognition approach, which varies according to speech, is based on literary features such as word or sub-word units. Languages having Latin alphabets, text-based language recognition has been widely solved. However, spoken language recognition is completely different compared to the text based language recognition.

In listening experiments performed on humans, it has been found that two clues are used in general to determine the language classes: prelexical and lexical [1]. Phonetic inventory, phonotactic, rhythm, and intonation properties constitute prelexical information [2]. These are separated from lexical information such as words and syntactic. It was found out that infants successfully use prelexical cues to distinguish languages [2]. It is clear here that the infants distinguish languages by their phonemes, rhythm, and intonation characteristics. Adults who have no knowledge about languages also decide only with their prelexical knowledge to distinguish languages.

Studies of spoken language recognition have revealed that acoustic and phonotactic features provide the most important language cues [3, 4]. Acoustic features are concerned with the physical properties (frequency spectrum, formants, etc.) of phonemes, while phonotactic features determine constraints of allowed syllable structures (sequence of phonemes) in a language.

Dialect recognition is a special case of language recognition. The dialect is defined as a local speaking style that can be separated at a certain rate from the standard language having the same root [5]. People who live in the same region have similar dialect characteristics. Dialects differ more in terms of phonetic, morphological, sometimes lexical, and syntactic aspects compared to the language they belong to [6]. In addition to features such as gender and age, dialect differences are important factors affecting the performance of automatic speech recognition systems [7]. Thus, dialect differences must be addressed in order to build large scale, speaker independent speech recognition systems [8].

The dialect of Istanbul has been adopted as the standard dialect of Turkish language. However, different dialects are spoken in various regions of Turkey and they are different in many ways from Istanbul dialect. By processing these differences, the performances of Turkish speech recognition systems can be increased. In this study, Turkish dialect recognition was made based on the above mentioned acoustic and phonotactic features.

Language recognition methods can be used here since dialect recognition is similar in many ways to language recognition. There are studies in the literature where acoustic features such as cepstral coefficients are used with Gaussian Mixture Model (GMM) for language recognition [9]. There are classification studies with GMM using acoustic features for Arabic dialect recognition [8]. Studies were performed by using GMM and Support Vector Machines (SVM), in which acoustic and phonotactic features are examined for English dialect recognition problems [10]. In addition, phoneme sequences were modeled by n-gram models for dialect [11] and language recognition [12]. In recent years, studies in which acoustic features are used in deep learning architectures have been observed [13]. In addition, there are various studies in which deep learning architectures are used for language recognition [14-17].

Deep Learning can be described as a whole of artificial neural networks based on deep architecture and methods developed for them [18]. In this approach, the quantity of hidden layers that perform the actual learning function of artificial neural networks were increased and new techniques were developed. Initialization of network weights, optimization algorithms, and activation functions can be given as examples to these techniques. Deep neural networks can be considered to combine classification and feature extraction processes [19]. In each layer, features that are specific to the input data are learned and these learned features are used in the next layer. In this way, an architecture is established in which simplest to most complex features are learned starting from the input layer to the output layer [20].

Convolutional Neural Networks (CNN), a special type of deep neural networks, have been successfully applied in image classification [21] and speech recognition [22] systems. In this study, log-mel coefficients (log mel-spectrogram) of speech samples were extracted and classified at acoustic level with CNN architecture. In speech samples, the stress at the end of the sentences was found to be effective in determining the Turkish dialects. For this, it is proposed that final words of the sentences are isolated and classified by their spectral features.

Studies in the dialectology are limited since the datasets are not sufficient and the analysis processes are long [23]. In this study, the Turkish Dialects Dataset [24], which is also expected to be helpful for dialectology studies, was used.

The phonotactic approach to the dialect recognition problem is based on the fact that the phonemes in the dialects differ in terms of sequence. The phoneme sequence in one dialect may not be on the other, and this feature can be used to distinguish one dialect from the other [8]. Parallel Phone Recognition followed by Language Model (PPRLM) [4] method is a frequently used method

in language and dialect recognition problems. As a language model in PPRLM architecture, the probabilities of phoneme sequence are modeled using a statistical n-gram models. It is proposed here to use phoneme-based LSTM RNN (Long Short-Term Memory Recurrent Neural Networks) language model instead of n-gram model in PPRLM.

Language models obtained with LSTM networks [25] have become popular in speech recognition systems in recent years. Language models created with LSTM networks produce more successful results than statistical N-gram models and classical RNN language models [26]. Language models operate on a word or character basis and provide prediction of the next word or character based on a context. In this study, it was aimed to train the LSTM recurrent networks on a phoneme basis and predict the next phoneme. The LSTM language models obtained for each dialect are used in the PPRLM architecture to find the probabilities of the given samples.

In the second section of the study, the dataset used was introduced, and in the third section the methods proposed for the acoustic and phonotactic approaches were explained. In the fourth section, experiments performed after the extraction of the features and in the fifth section, results and discussions of the experiments took place. The sixth section contains the conclusion part of the study.

2. DATASET

Turkish Dialects Dataset was used in the study. It is known that such a data set about the Turkish dialects has not been established to date. Existing datasets are mostly for linguists.

2.1. Turkish Dialects Datasets

A dataset was prepared by arranging the content obtained from linguists who study the Turkish dialects. Records collected from the Turkish dialect regions of Ankara, Kıbrıs, Trabzon and Alanya were used, and studies were carried out on them. Speech records were collected by face to face interviews. In the regions having these dialects, people who are considered to have the characteristics of the dialect were selected. Attention was paid to the fact that the selected people are old, that most of them have not left the area they live in and that their level of education is low. People who provide these properties are more likely to have dialect-specific phones [27].

The collected records has been made noise-free and the long silences between sentences have been removed. Thus, 2.7 h data was obtained for four dialect regions in total including Ankara 0.8 h, Kıbrıs 0.65 h, Trabzon 0.55 h and Alanya 0.7 h. The records of four people were selected for each dialect. The sampling frequency of all recordings was converted to 16 KHz. Later, all recordings were labeled on word basis by using Praat [28] software. Speech records were separated into sentences (utterances), so that about 400 sentences were specified for each dialect region. Each sentence was approximately 2-3 seconds long. This dataset was based on spontaneous speeches and was not based on text.

3. ANALYSIS METHODS

Dialect recognition systems are generally implemented in two approaches: Acoustic approach and Phonotactic approach.

3.1. Acoustic Approach

The acoustic approach for dialect recognition is based on the fact that the sounds in the dialects differ in terms of their spectral distributions. It was demonstrated that Turkish dialects differ in the vowel-consonant context [29]. For this reason, the spectral distribution differences can distinguish the Turkish dialects from each other.

In order to investigate the speech signal from acoustic point of view, the characteristics of the signal at physical level must be examined. A pressure wave occurs when people speak and two speech events can be distinguished at the acoustic level according to the amplitude or frequency components of the waves. Information at acoustic level is obtained by parameter extraction methods from the raw signal. The speech signal is passed through stages such as Fast Fourier Transform (FFT) and Mel filters to obtain parameters. In recent years, mel-spectrogram features which represent acoustic information well, have been frequently used in CNN-type deep neural networks [22].

The dialect recognition problem can be modeled in acoustic means as in Eq. (1) [8]. Let $D = \{D_1, D_2, \dots, D_n\}$ be the set of dialects that are to be classified as and $\vec{a} = \{\vec{a}_1, \vec{a}_2, \dots, \vec{a}_F\}$ be frame based feature vectors that gives the spectral information of the speech sample. The aim here is to find the dialect class $\widehat{D}$ which gives the highest probability by using the frame-based spectral feature vectors obtained from a given speech sample.

$$\widehat{D} = argmax_i \, P_{D_i}(D_i|\vec{a}) \qquad (1)$$

Convolutional Neural Networks (CNN)

CNN architecture is frequently used in image recognition and acoustic signal processing [22]. These networks operate in two stages. In the first step, it allows local features related to each other to be extracted and in the second step, it allows classification to be made by using multi layered neural networks.

Local feature maps are obtained by convolving the small sized filters (kernels) on the input data of the network. Since a filter is applied to all input data, this characteristic of the network is called shared weights. The parameters of the filters (weight matrix) are updated during training with the back propagation algorithm so that the filters are

learned. Feature maps are obtained by applying the activation function (tanh, sigmoid, etc.) to the output of the layer. Feature maps are then subject to pooling process. Pooling, which is an important part of CNN architecture, reduces dimensions of feature maps, thus reduces variety of features. Pooling is generally done with the *max* operator (aka *max-pooling*). Largest element is found among features in the selected sized pooling window and thus the size of the feature maps is reduced. Since there are no parameters of the pooling process, there is no learning at this stage.

The classification stage of the CNN architecture consists of multilayer deep neural networks. The feature maps obtained from the pooling process are flattened and transformed into vectors to be input to the multilayer neural network. In general, sigmoid activation function is used in hidden layers and softmax function is used in output layer. The posterior probability for each class in the output layer is estimated. The one with highest probability is selected among them and the classification is completed.

3.2. Phonotactic Approach

Phonotactic deals with allowed phoneme sequences of a language/dialect. It is based on the principle that the phonemes in the dialects differ in terms of sequence and frequency. If a phoneme sequence which exists in one dialect that doesn't exist in the other, then this phoneme sequence can be used to distinguish the two dialects from each other.

If the study conducted is actually considered as a classification problem, dialect recognition can be probabilistically modeled as in Eq. (2) [8]. Let $D = \{D_1, D_2, \dots, D_n\}$ be a dialect set that is wanted to be recognized and $C = \{c_1, c_2, \dots, c_K\}$ be a sequence of phonemes. The aim here is to find the dialect class $\widehat{D}$ that gives the highest probability for the phoneme sequence obtained from a given speech sample.

$$\widehat{D} = argmax_i \, P_{D_i}(D_i|C) \qquad (2)$$

Standard phonotactic approaches estimate the probabilities of phoneme sequences obtained from one or more phoneme recognizers [30]. The best known phonotactic method is the PPRLM method [4].

Speech samples in the PPRLM method are separated into phonemes using more than one (m) phoneme recognizer in parallel. N-gram models are then trained on these phoneme sequences at the number of dialects (n) that are wanted to be recognized. At the end of this process, $m \times n$ N-gram models are trained in total and the probability distributions of the phoneme sequences of all dialects are extracted. In the recognition process, the phoneme sequence is obtained by passing the given speech sample through the phoneme recognizers. The trained n-gram models are applied to the phoneme sequence. The n-gram model that produces the highest probability within these models gives the dialect class of the speech sample.

The phoneme recognition part of the phonotactic system is important. In language recognition, phoneme recognizers do not have to be trained in the target language to be recognized, but they are expected to cover the phonemes in the target language. Therefore, instead of a single phoneme recognizer, multiple phoneme recognizers are used to capture phonemes in the target language. Because, a phoneme that is not in the training set of one recognizer can be in the training set of the other recognizer. Here, the consistency rate of the phoneme recognizer is important [30]. Consistency means obtaining same result in each situation.

In this study PhnRec software [31] was used as a phoneme recognizer. This software has four phoneme recognizers (PRs) trained on English, Russian, Hungarian and Czech languages. In the study, English and Hungarian phoneme recognizers which showed more consistent and accurate results for the Turkish data set, were used. For example, the <giyerdik> ("we were wearing") utterance is separated into its phonemes by these recognizers as follows:

<giyerdik> → English PR. → iy – eh – er – d – ih

<giyerdik> → Hungarian PR. → g – i – j – e: – d_: – i

Also in this study, LSTM RNN language models are used in PPRLM architecture. The LSTM model stores information such as variable and long term history rather than N-gram which stores fixed and short history.

3.2.1. Long Short-Term Memory (LSTM) Neural Networks

Unlike classical neural networks, in Recurrent Neural Networks, the output of the hidden layer is connected both to the input of the next layer and to the input of the layer itself (Fig 1a). The layer's output activation (h_t) is again connected to its input to model the temporal dependencies that are important for speech data.

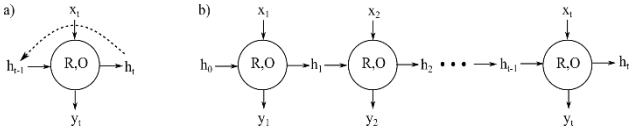

Figure 1.a) RNN architecture, b) The unfolded RNN [32].

RNN takes a sequence of $x_{1:n} = (x_1, \dots, x_n)$ input vectors and h_0 initial state vector, then returns a sequence of $y_{1:n} = (y_1, \dots, y_n)$ output vectors together with sequence of $h_{1:n} = (h_1, \dots, h_n)$ state vectors. The R function defined as recurrent, computes the new state vector h_i using the state vector h_{i-1} and the input vector x_i. This is usually a sigmoid function. The function O also converts the h_i state vector to the output vector y_i. Here h_i stores the symbols seen up to that time [33]. t represents the time step from 1

to T and corresponds to the number of layers when RNN is unfolded in Fig. 1b. Information about the history stored in the memory is as much as T value.

$$RNN(h_0, x_{1:n}) = h_{1:n}, y_{1:n} \quad (3)$$

$$h_i = R(h_{i-1}, x_i) \quad (4)$$

$$y_i = O(h_i) \quad (5)$$

RNN causes the vanishing gradient problem during back propagation [34, 35]. For this reason it is difficult to train recurrent neural networks in practice [36]. LSTM recurrent networks are designed to eliminate this disadvantage of the RNNs. This is achieved by the memory cells and various gates added to the existing RNN architecture [37]. With the help of these gates, it is controlled which information the cell will accept, which one will forget and which will be output. Thus, this structure can learn short or long patterns of temporal data. The calculations described in [34] can be referred to for updating the memory cell gates and their outputs.

3.2.2. LSTM Neural Network Language Model And The Proposed Architecture

LSTM neural networks were used for language modeling. Here, the LSTM language model was trained at the phoneme level. The aim here is to model the probability of phoneme sequences obtained from phoneme recognizers for each dialect sample. The probability of the sequence consisting of K phonemes is calculated as follows:

$$P(c_1, \dots, c_K) = \prod_{k=1}^{K} P\left(c_k \middle| c_{1:(k-1)}\right) \quad (6)$$

Here, 1: $(k-1)$ demonstrates the sequence from phoneme with index 1 up to the phoneme with index $k-1$. The LSTM network can estimate the next phoneme using the probability distribution of the available phonemes. Each of the $m \times n$ LSTM models trained in this way produces a high probability for the phoneme sequence in that dialect, whichever is trained on which dialect.

Figure 2 shows the position of the LSTM language models used in this study in the PPRLM architecture. This design was inspired by the architecture in [8]. Speech samples were preprocessed and made ready for phoneme recognizers. Two ($m = 2$) phoneme recognizers were used for each of the four dialects ($n = 4$). An LSTM language

model was trained for each dialect using the phonemes generated by these recognizers.

Here, 39 phonemes in English and 61 phonemes in Hungarian were obtained from phoneme recognizers. For this reason, the input and output vectors of the LSTM neural network are of the same size, 39 and 61 dimensions for English and Hungarian respectively. The input and output vectors are in one-hot encoding structure. That is, for each phoneme there is a sparse matrix structure with 1 in the corresponding phoneme position and 0 in the others. For example, in the phoneme sequence <*iy–eh–er–d–ih*> obtained from English recognizer, the <*eh*> phoneme is the output label while <*iy*> phoneme is given to the input layer with one-hot vector structure at $T = 1$ time step. In the same way, when the <*eh*> phoneme is given to the input layer, <*er*> phoneme forms the output label at $T = 2$ time step. In this way, only one phoneme is processed at each time step and the process continues until the entire phoneme sequence of the relevant utterance is processed (Fig. 1b). Thus, with the trained LSTM neural network, the phoneme after each phoneme is predicted and the language model of the corresponding dialect is constructed.

The cross-entropy loss function was used to train the LSTM network. This loss function maximizes the probability that the network estimates (Eq. 7). The performance of language models is measured by a special parameter called perplexity (PP). A good language model should give a low PP value to a sentence in that language. Perplexity metric of each LSTM model for a given sentence was calculated as in Eq. (8).

$$Loss = -\frac{1}{M}\sum_{i=1}^{M} \ln P_{c_i} \quad (7)$$

$$H(C) = \frac{1}{N}\log P(C_1^N) \quad (8)$$

$$PP(C) = 2^{H(C)} \quad (9)$$

Here, P_{c_i} are the posterior probability values of the phonemes in the output layer. Loss is found by taking the mean of the negative logarithms of these probability values. M is the number of phonemes of the utterance being processed, N is the number of phonemes of the utterance

being tested. Softmax Regression was used to decide which language model will be chosen according to PP values produced by LSTM models. It should be noted here that, calculated scalar PP values is given to input of the softmax regression directly. Thus, there are 8 nodes in the input and 4 nodes in the output of the softmax regression. The softmax function is used to obtain the target classes in the output layer. LSTM network setup and softmax regression calculations were made with the help of Keras Library [38]. Keras is written in Python and is one of the most popular deep learning platforms.

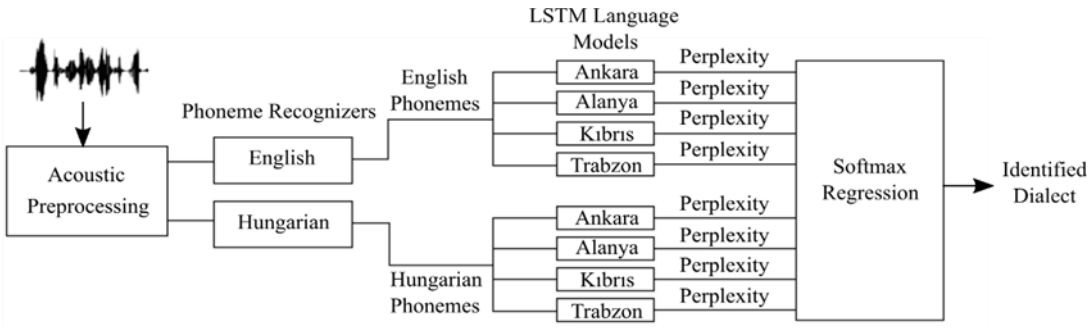

Figure 2. PPRLM architecture consisting of phoneme recognizers and LSTM language models.

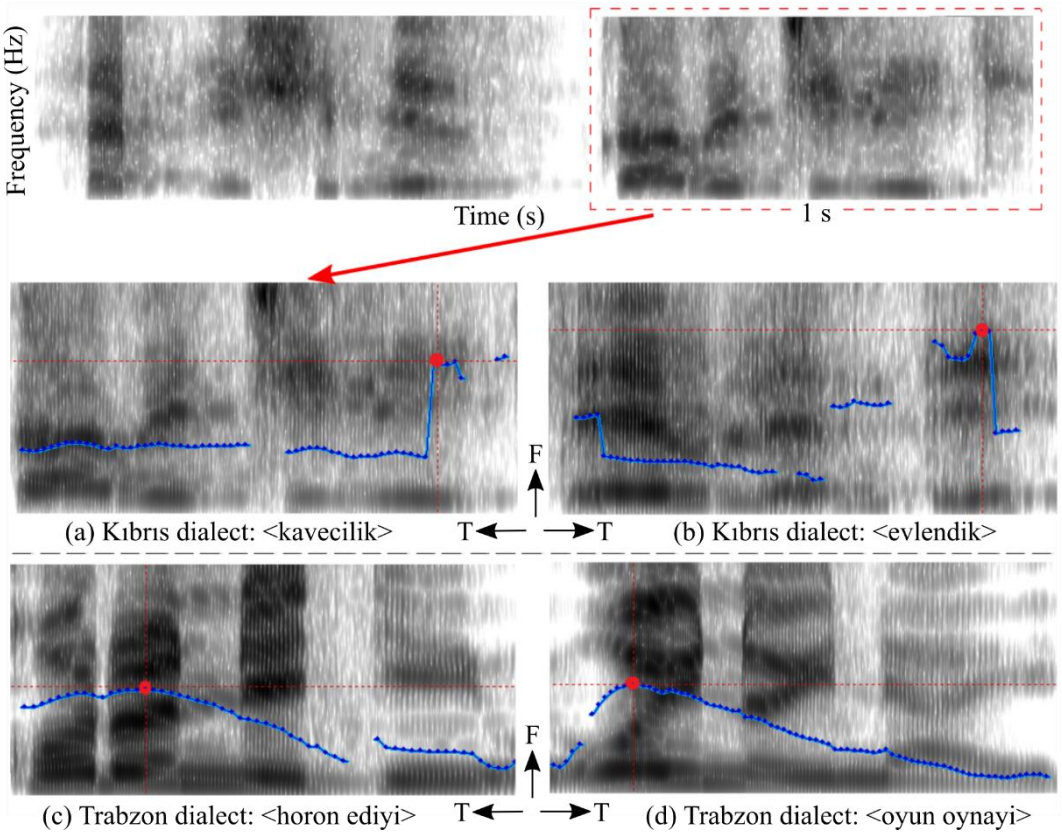

Figure 3. Spectrogram representation for the end of two sentences of Cyprus and Trabzon dialects.

4. EXPERIMENTS

In signal processing, the frequency spectrum of the speech signal is extracted with fast Fourier transform. Mel-scaled frequency spectrum is obtained by applying non-linear mel-filters to this spectrum. By taking the logarithm of this, the power spectrum is calculated. A spectrogram visually expresses the time-dependent variation of the frequency spectrum of the signal. The log mel-spectrogram shows the time-dependent variation of the power spectrum. Log mel-spectrogram features are widely used in language recognition. In this study, these features were the inputs of CNN architecture.

Speech samples in the Turkish dataset were separated into phonemes using phoneme recognizers. With these phoneme sequences, LSTM neural networks were trained to build the language model of the related dialect. Trained neural network language models were used in the PPRLM architecture. According to the perplexity values obtained here, softmax regression was trained to decide the dialect class.

In examinations made on the dialect samples in the Turkish data set, it was generally determined that sentence finals (sf) were stressed more. The shape of this stress at the end of the sentence varies from dialect to dialect (Figure 3).

In Figure 3, pitch variations (blue curve) at the sentence finals the most stressed part (red dot) are shown. In speech samples, the pitch frequency of the Kıbrıs dialect generally increases towards the end of the sentence, on the contrary it fades out in the Trabzon dialect. Because of this detected

property, the dialects can be acoustically distinguished from each other only by using final parts of sentences.

The last 0.5 s and 1 s lengths of each sentence were used to capture the acoustic variation caused by the stress at the sentence finals. Log mel-spectrogram features were extracted from these parts. Thus, the performance of the proposed method on the Turkish dataset was measured.

For acoustic methods, sentences in the dataset were divided by k-fold cross validation as training and test sets. Here $k = 10$ was selected. Thus, the whole dataset was divided into 10 parts and 9 of these were used for training and 1 for testing. The separation of these parts into training and testing was carried out 10 times in a row, and the output score was obtained by taking the average of these. The reason for doing so is to reduce the randomness and to ensure the consistency of the output score.

In the following sections of this chapter, the parameters of CNN and LSTM networks were given with the proposed form and the results were reported. Accuracy rate was used as a performance criteria for all methods. Figure 4 summarizes the steps for obtaining the input forms used for acoustic and phonotactic methods and the tools used for them.

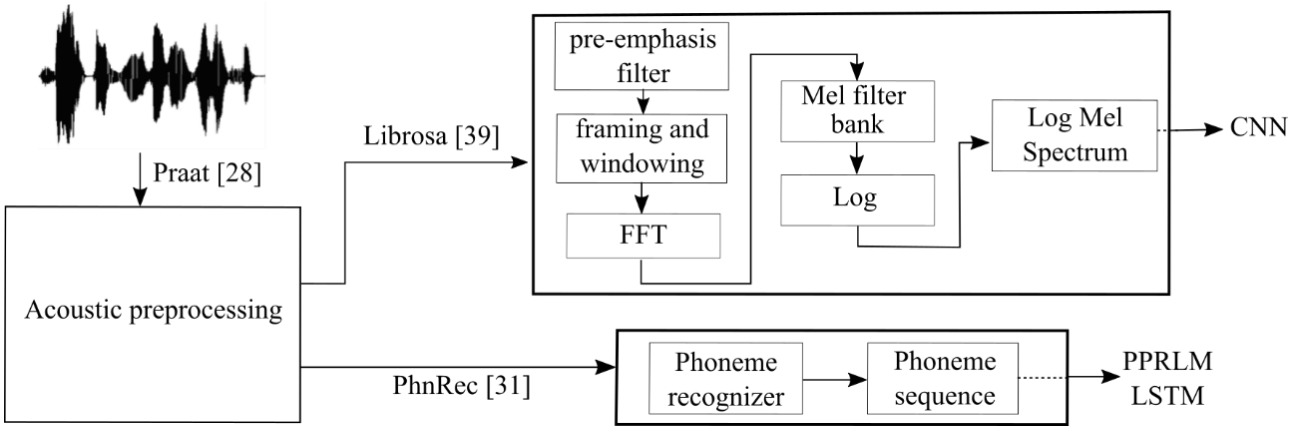

Figure 4. The process of obtaining inputs and the tools used.

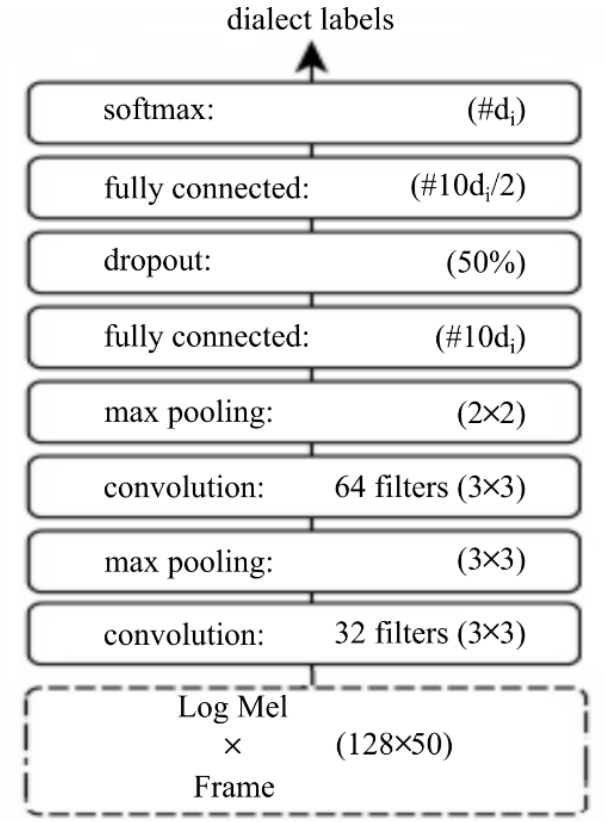

Figure 5. Network parameters of the Log Mel-CNN model.

4.1. CNN and Log Mel-Spectrogram

By using the open sourced librosa library [39], a 128 mel-feature for each frame was extracted. 128×50 and 128×100 sized log mel-spectrograms were obtained from the last 0.5s and 1s length sections of each sentence, respectively, and were given to CNN input (Figure 5). In CNN model two convolution layers consisting of 32 and 64 filters were used, respectively. 32 and 64 feature maps were obtained from these layers. Convolution was performed on both axes (frequency and time) by applying 3×3 sized filters to the mel-spectrogram matrix. Similarly the pooling window was also applied along both axes of the feature maps. The parameters that give the best result for the CNN model were shown in Figure 5. In order to make comparisons in this model, both the sentence finals (sf) and the whole sentence were used. If the whole sentence was used, the sentence was given to the CNN input in 0.5 s length pieces. In this case, the score average of all parts was taken and the output score was obtained.

The dropout [40] step in Figure 5 cuts 50% of the connections in that layer randomly so as to prevent overfitting of the network and reduces the dimensions. ReLU [41] was used as the activation function in the network, and softmax function was used at the output layer. The training was done using the stochastic gradient descent (SGD) [42] algorithm according to the cross-entropy loss criteria. In Figure 5, $d_i = 4$ is the number of the dialect

classes. The Keras library [38] was used to set up and train the CNN architecture.

4.2. LSTM and PPRLM

All speech samples in the dataset were tokenized into phonemes by passing through the two recognizers. Using these phonemes, two LSTM language models were trained for each dialect (Figure 2). The perplexity (*PP*) values of all trained language models were calculated according to Eq. 9. The training of the LSTM network was performed with the SGD algorithm according to the cross-entropy criteria and the gradients were calculated by the Back Propagation Through Time (BPTT) algorithm [43]. In the beginning, the weight matrices were initialized with values close to zero and the learning rate was determined as $\alpha = 0{,}1$. After every 10 samples, learning ability of the network was tested with validation data. In this way, if the probability values of the validation data increase, training continues, otherwise the value of α is reduced to half. If the probability value does not increase significantly, the training is terminated. The input and output layers of the LSTM network are 39 for the English recognizer and 61 for the Hungarian. There are two hidden layers and 50 nodes in each layer. The time step value was taken as $T = 10$. This value indicates the number of phonemes to keep in memory. Speech samples for training, testing and validation were divided by 80, 10, 10 percent, respectively.

The perplexity values produced by the trained LSTM language models were given to the softmax regression network. Softmax function was used at the output of the 8-input 4-output regression network. Thus, the dialect class was identified.

For the implementation of the methods, a system with an Intel i7 processor with a frequency of 2.7 GHz and a memory of 16 GB was used. All experiments were performed on a central processing unit (CPU). For the two approaches, the change of the computation time according to the number of samples is given in Figure 6.

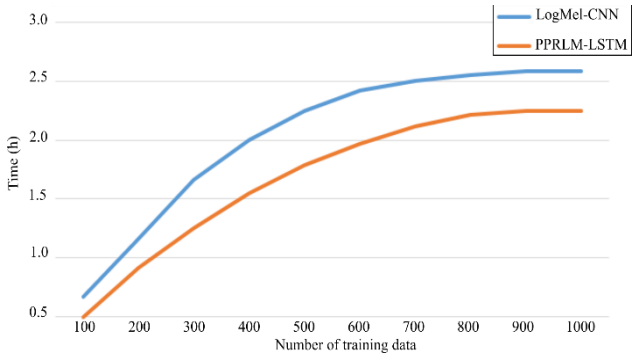

Figure 6. Sample size - time (h) diagram for the methods.

5. RESULTS AND DISCUSSIONS

Utterances with a length of 0.5 s, 1 s and 3 s of the Turkish dataset were tested by acoustic approach and utterances with a length of 1 s and 3 s were tested by phonotactic approach and the calculated accuracy rates were given in Table 1. Best scores are written in bold.

Table 1. Accuracy rates of the methods produced for the Turkish dataset based on test durations.

Models and test durations		**Accuracy rates (%)**		
		0.5 s	**1 s**	**3 s**
#1	LogMel-CNN (sf)	84.0	84.2	-
#2	LogMel-CNN	83.7	84.0	**84.4**
#3	PPRLM-LSTM (T=5)	-	83.8	84.1
#4	PPRLM-LSTM (T=10)	-	84.2	**85.1**
#5	PPRLM-LSTM (T=20)	-	83.9	84.6

For the LogMel-CNN models, both the sentence finals (sf) and the entire sentence were used as input (#1, #2). The LSTM language model was tested with different T time steps and the model with $T = 10$ (#4) gave better accuracy (85.1%). In acoustic methods, as the sentence finals were limited to 0.5 s and 1 s, no experiment was conducted for samples with 3 s duration (because, sentence finals are short segments of speeches). In the same way phonotactic methods did not provide sufficient phoneme sequences in 0.5 s period and therefore experiments were carried out only for samples with a duration of 1 s and 3 s.

For acoustic models in Table 1, models trained with sentence finals (sf) give better results than models trained with entire sentence ($\#1 > \#2$ for 0.5 s and 1 s). This result shows that dialect regions can be classified only by looking at the end of the sentences. It is important that such short speech samples give these rates. Because the processing of only the end of the sentences means both the processing power and the time saving. Such an approach, where only the sentence final is used, has not been investigated before. However, in the long durations (3 s), the model in which whole sentence is used, performed better (84.4%). This supports the hypothesis that the accuracy rate increases as test duration increases [44]. It was also observed that the phonotactic methods gave better scores than the acoustic methods. This result confirms the argument that the phonotactic method provides more discriminative information than the acoustic method [8, 45].

In the following confusion matrices, the classification results produced by the methods used are given. In the tables; the rows indicate the target class (ground-truth) and the columns indicate the output which the model predicts. The confusion matrices of the models that give the best results on the Table 1 are shown in the following tables (Table 2, 3).

The prominent pattern in the confusion matrices is the high possibility of confusion between the Alanya and Kıbrıs dialects. Another pattern is that, Kıbrıs dialect is confused with Trabzon dialect at least rates. These two results

support the fact that the dialect properties of geographically close regions are similar [29].

Table 2. The confusion matrix of the LogMel-CNN model (%).

	Ankara	**Alanya**	**Kıbrıs**	**Trabzon**
Ankara	**83.4**	4.2	5.4	7.0
Alanya	4.1	**85.4**	5.4	5.1
Kıbrıs	5.1	6.5	**84.8**	3.6
Trabzon	4.9	6.1	5.0	**84.0**

Table 3. The confusion matrix of the PPRLM-LSTM model (%)

	Ankara	**Alanya**	**Kıbrıs**	**Trabzon**
Ankara	**84.4**	6.4	4.4	4.8
Alanya	4.7	**85.9**	5.2	4.2
Kıbrıs	4.8	6.6	**84.6**	4.0
Trabzon	6.1	4.8	3.6	**85.5**

6. CONCLUSIONS

In this study, a recognition was made on Turkish dialects by acoustic and phonotactic methods. Deep learning neural networks were used for recognition. In these networks, log mel-spectrogram features and phonemes were used as input data. There is no classification study for the Turkish dialects made by speech processing and machine learning methods in the literature. This makes it impossible to compare these with the results of other studies.

Two approaches were proposed in the study. The use of sentence finals (0.5 s and 1 s) in deep neural networks is one of the suggestions. The ratios given in Table 1 demonstrate that this approach gives good results in dialect recognition. This result shows that the dialect regions can be classified only by looking at the end of the sentence. Such an approach using only sentence finals has not been applied before.

Parallel PRLM is a popular architecture for language / dialect recognition. N-gram models are used for language modelling in this architecture. Since n-grams model fixed and short history information, it is suggested to use LSTM neural network model instead of n-gram in this study. The use of LSTM instead of n-gram in this study is another of the suggestions. There is no other study that used the LSTM language model in PRLM architecture for dialect recognition.

A corpus called the Turkish Dialects Dataset was created and used in this study. This dataset includes speech samples from Ankara, Alanya, Kıbrıs and Trabzon dialects. This dataset will be shared online to support studies on dialect recognition and linguistics. Also this dataset is intended to be made to cover the entire dialect areas of Turkey. It is thought that the classification performance will increase with increasing number of dialects and samples from each dialect. In addition, the results here can be improved by using Bi-directional LSTM (BiLSTM) architectures, especially for phonotactic modeling.

REFERENCES

[1] J. Zhao, H. Shu, L. Zhang, X. Wang, Q. Gong, P. Li, "Cortical competition during language discrimination", *Neuroimage*, 43(3), 624–633, 2008.

[2] F. Ramus, J. Mehler, "Language identification with suprasegmental cues: a study based on speech resynthesis.", *J. Acoust. Soc. Am.*, 105(1), 512–21, 1999.

[3] Y. K. Muthusamy, E. Barnard, R. A. Cole, "Reviewing Automatic Language Identification", *IEEE Signal Process. Mag.*, 11(4), 33–41, 1994.

[4] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech", *IEEE Trans. Speech Audio Process.*, 4(1), 31–44, 1996.

[5] N. Demir, "Ağız Terimi Üzerine", Türkbilig, 105–116, 2002.

[6] A. Etman, A. A. Louis, **American dialect identification using phonotactic and prosodic features**, IntelliSys - Proc. 2015 SAI Intell. Syst. Conf., pp. 963–970, 2015.

[7] S. Safavi, M. Russell, P. Jančovič, "Automatic speaker, age-group and gender identification from children's speech", *Computer Speech & Language*, 50, 141-156, 2018.

[8] F. Biadsy, **Automatic Dialect and Accent Recognition and its Application to Speech Recognition**, PhD Thesis, Columbia Univ., pp. 1–171, 2011.

[9] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, R. Dehak, **Language recognition via Ivectors and dimensionality reduction**, in Proceedings of the Annual Conference of the International Speech Communication Association, Interspeech, 857–860, 2011.

[10] A. Hanani, M. Russell, M. J. Carey, "Human and computer recognition of regional accents and ethnic groups from British English speech", *Comput. Speech Lang.*, 27(1), 59–74, 2013.

[11] H. Soltau, L. Mangu, F. Biadsy, **From modern standard Arabic to Levantine ASR: Leveraging GALE for dialects**, in ASRU, 266–271, 2011.

[12] M. Soufifar, S. Cumani, L. Burget, J. H. Cernocky, **Discriminative classifiers for phonotactic language recognition with ivectors**, in ICASSP, 4853–4856, 2012.

[13] I. Lopez-moreno, J. Gonzalez-dominguez, O. Plchot, D. Martinez, J. Gonzalez-rodriguez, P. Moreno, **Automatic language identification using deep neural networks**, in ICASSP, vol. 1, 2014.

[14] C. Salamea, L. F. D'Haro, R. De Cordoba, R. San-Segundo, "On the use of phone-gram units in recurrent neural networks for language identification", *Proceedings of Odyssey*, 117-123, 2016.

[15] M. Jin, Y. Song, I. Mcloughlin, L.R. Dai, Z.F. Ye, "LID-senone extraction via deep neural networks for end-to-end language identification", *Proceedings of Odyssey*, 210-216, 2016.

[16] A. Lozano-Diez, R. Zazo-Candil, J. Gonzalez-Dominguez, D.T. Toledano, J. Gonzalez-Rodriguez, **An end-to-end approach to language identification in short utterances using convolutional neural networks**, in Interspeech, 2015.

[17] Y. Tian, L. He, Y. Liu, J. Liu, "Investigation of senone-based long short-term memory RNNs for spoken language recognition", *Proceedings of Odyssey*, 89-93, 2016.

[18] L. Deng, D. Yu, "Deep Learning: Methods and Applications", *Found. Trends Signal Process.*, 7, 2014.

[19] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, O. Vinyals, **Learning the speech front-end with raw waveform CLDNNs**, in Interspeech, 2015.

[20] A.-R. Mohamed, G. E. Hinton, G. Penn, **Understanding How Deep Belief Networks Perform Acoustic Modelling**, in ICASSP, 2012.

[21] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber, **Flexible, High Performance Convolutional Neural Networks for Image Classification**, Proc. Twenty-Second Int. Jt. Conf. Artif. Intell., 2011.

[22] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, "Convolutional Neural Networks for Speech Recognition", *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 22(10), 2014.

[23] N. F. Chen, W. Shen, J. P. Campbell, **A linguistically-informative approach to dialect recognition using dialect discriminating context-dependent phonetic models**, in ICASSP, 5014–5017, 2010.

[24] G. Işık, H. Artuner, **A Dataset For Turkish Dialect Recognition and Classification with Deep Learning**, in 26. IEEE Signal Processing and Communications Applications Conference (SIU), 2018.

[25] M. Sundermeyer, R. Schlüter, H. Ney, **LSTM Neural Networks for Language Modeling**, Proc. Interspeech, 194–197, 2012.

[26] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, S. Khudanpur, **Recurrent Neural Network based Language Model**, in Interspeech, 1045–1048, 2010.

[27] N. Demir, "Ağız Araştırmalarında Kaynak Kişi Meselesi", **Folk. Prof. Dr. Dursun Yıldırım Armağanı**, 11, 1998.

[28] Internet: P. Boersma, D. Weenink, Praat program, http://www.praat.org, 03.08.2019.

[29] L. Karahan, **Anadolu Ağızlarının Sınıflandırılması**, Türk Dil Kurumu Yayınları, 1996.

[30] P. Matějka, P. Schwarz, J. Cernock, P. Chytil, **Phonotactic Language Identification using High Quality Phoneme Recognition**, in Eurospeech, 2005.

[31] Internet: P. Schwarz, P. Matejka, L. Burget, O. Glembek, Phoneme recognition based on long temporal context, http://speech.fit.vutbr.cz/software, 27.10.2019.

[32] M. H. S. Segler, T. Kogej, C. Tyrchan, M. P. Waller, "Generating Focussed Molecule Libraries for Drug Discovery with Recurrent Neural Networks", *ACS Central Science*, 4(1), 120-131, 2017.

[33] Y. Goldberg, "A primer on neural network models for natural language processing", *J. Artif. Intell. Res.*, 57, 345–420, 2016.

[34] A. Graves, J. Schmidhuber, "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures", *Neural Networks*, 18(5), 602–610, 2005.

[35] H. Sak, A. Senior, F. Beaufays, **Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling**, Interspeech, 338–342, 2014.

[36] Y. Bengio, P. Simard, P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", *IEEE Trans. Neural Networks*, 5, 157–166, 1994.

[37] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory", *Neural Comput.*, 9(8), 1735–1780, 1997.

[38] Internet: F. Chollet, Github, https://github.com/fchollet/keras. 15.11.2019.

[39] B. Mcfee, C. Raffel, D. Liang, D. P. W. Ellis, M. Mcvicar, E. Battenberg, O. Nieto, **librosa: Audio and Music Signal Analysis in Python**, Proc. 14th Python Sci. Conf., no. Scipy, 1–7, 2015.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *J. Mach. Learn. Res.*, 15, 1929–1958, 2014.

[41] V. Nair, G. E. Hinton, **Rectified linear units improve restricted boltzmann machines**, Proc. 27th Int. Conf. Mach. Learn., 2010.

[42] L. Bottou, **Large-Scale Machine Learning with Stochastic Gradient Descent**, Proc. COMPSTAT'2010, 177–186, 2010.

[43] R. J. Williams, J. Peng, "An efficient gradient-based algorithm for online training of recurrent network trajectories", *Neural Comput.*, 4, 491–501, 1990.

[44] L. Ferrer, Y. Lei, M. Mclaren, N. Scheffer, "Study of Senone-Based Deep Neural Network Approaches for Spoken Language Recognition", *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 24(1), 105–116, 2016.

[45] Z. Tang, D. Wang, Y. Chen, L. Li, A. Abel, "Phonetic temporal neural model for language identification", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1), 134-144, 2017.