Journal of Ambient Intelligence and Humanized Computing (2022) 13:1985–2000
https://doi.org/10.1007/s12652-021-02960-0

ORIGINAL RESEARCH

LSTM and CNN based ensemble approach for spoof detection task in automatic speaker verification systems

Mohit Dua[1] · **Chhavi Jain**[1] · **Sushil Kumar**[1]

Received: 12 November 2019 / Accepted: 9 February 2021 / Published online: 19 February 2021

Abstract
Nowadays, fingerprint and retina scans are the most reliable and widely used biometric authentication systems. Another emerging biometric approach for authentication, though vulnerable, is speech-based systems. However, speech-based systems are more prone to spoofing attacks. Through such malicious attacks, an unauthentic person tries to present himself as legitimate user, in order to acquire illegitimate advantage. Therefore, these attacks, created by synthesis or conversion of speech, pose an enormous threat to the reliable functioning of automatic speaker verification (ASV) authentication systems. The work presented in this paper tries to address this problem by using deep learning (DL) methods and ensemble of different neural networks. The first model is a combination of time-distributed dense layers and long short-term memory (LSTM) layers. The other two deep neural networks (DNNs) are based on temporal convolution (TC) and spatial convolution (SC). Finally, an ensemble model comprising of these three DNNs has also been analysed. All these models are analysed with Mel frequency cepstral coefficients (MFCC), inverse Mel frequency cepstral coefficients (IMFCC) and constant Q cepstral coefficients (CQCC) at the frontend, where the proposed ensemble performs best with CQCC features. The proposed work uses ASVspoof 2015 and ASVspoof 2019 datasets for training and testing, with the evaluation set having speech synthesis (SS) and voice conversion (VC) attacked utterances. Performance of proposed system trained with ASVspoof 2015 dataset degrades with evaluation set of ASVspoof 2019 dataset, whereas performance of the same system improves when training is also done with the ASVspoof 2019 dataset. Also, a joint ASVspoof 1519 dataset is created to add more variations into the single dataset, and it has been observed that the trained ensemble with this joint dataset performs even better, while performing evaluation using single datasets. This research promotes the development of systems that can cope with completely unknown data in testing. It has been further observed that the models can reach promising results, paving the way for future research in this domain using DL.

Keywords Deep learning · LSTM · DNN · Audio spoofing · CNN

1 Introduction

The voice of a person can provide different kinds of information about him. Being the primary form of communication, speech is an appealing behavioural and physiological trait that can be used for biometric systems. The sound and way of talking can uniquely identify a person. Authentication for security via audio is getting attention due to the availability of low-cost sensors which can be easily deployed. However, speech biometrics are not yet used much in the real time scenarios. Since such authentication systems perform verification of a claimed person, they are still likely to be fooled through spoofed input. In terms of the networking literature, when one person attempts to exploit the security credentials of another through attacks, it is called as spoofing (Wu 2014; Wu et al. 2015). Voice-based system is used to restrict unauthorized access to a location or service and allowing authorized people (Scardapane et al. 2017). Therefore, it is highly important that this area is researched upon for real time adoption of such systems. Attacking these systems through spoofs needs a firm solution by developing countermeasures

✉ Mohit Dua
er.mohitdua@nitkkr.ac.in

Chhavi Jain
chhavijain2202@gmail.com

Sushil Kumar
sushilkumar9417047@gmail.com

[1] Department of Computer Engineering, National Institute of Technology, Kurukshetra, India

for ASV (Yamagishi et al. 2017). An ASV system simply tells whether a claimed utterance by the user can be accepted as genuine, or it needs to be rejected as spoofed.

There are two tasks which are closely related to ASV, speaker identification (SI) and speaker verification (SV). Speaker identification (SI) is the problem of identifying the person who spoke a given speech utterance. On the contrary, in speaker verification (SV) problem, a person first makes a claim about his identity, which is validated by the SV system using a speech sample. Thus, the system verifies whether the claim is correct or not (Aleksic and Katsaggelos 2006). Based on the uttered speech, there are two types of ASV systems, namely text-independent and text-dependent systems. Text-dependent systems contain a fixed group of words or phrases which are used for enrolling and verification in the system. On the other hand, systems that focus only on the user's voice, irrespective of the utterance are called as text-independent systems. Here, any kind of speech, in any language, can be passed into the system.

This problem of ASV was first introduced in Interspeech 2013 (Evans et al. 2013), in the special session held on spoofing and countermeasures for ASV, wherein future research in ASV was encouraged. Collection of datasets, standard metrics to be used and protocols were other areas discussed in this session. In the year 2015, ASVspoof challenge was organized with the objective of evaluating the state-of-the-art solutions in terms of accuracy and error rates, and to provide a common platform for testing anti-spoofing architectures (Wu et al. 2015; Scardapane et al. 2017; Yamagishi et al. 2017). The challenge involved text-independent verification of data and the attacks taken into consideration in the respective dataset are SS and VC. These challenges confirm that there is still room for improvement in the implemented approach, thereby promoting research.

1.1 Related work

In SS, natural-sounding speech is synthesized from text algorithmically (Dua et al. 2015; Sahu and Dua 2016, 2017). Many approaches that are used to detect synthetic speech are mainly based upon processing artefacts depending on a synthesis algorithm of a particular type. Based on the examination, dynamic variation present in the speech parameters of these synthetic speeches are less than those of actual speech (Dua et al. 2017, 2019b; Sahu et al. 2018). The researchers in (Satoh et al. 2001), investigated the utilization of intra-frame differences as a discriminative feature. This method performs well in detecting hidden Markov model (HMM) based speech synthesis without global variance compensation (Toda and Tokuda 2007). In Chen et al. (2010), higher order of Mel-cepstral coefficients (MCEPs) are employed to detect synthetic speech made by an HMM-based system. Another approach for detecting synthetic utterances involves the use of fundamental frequency statistics (De Leon et al. 2012b). In De Leon et al. (2012a), it is shown that Gaussian mixture model–universal background model (GMM–UBM) or support vector machine (SVM) are effective in protecting systems from spoofing attacks.

VC attacks generally involve modification of one person's voice, so that it can be matched with the target speaker's voice. Such fake voices are used for breaking into the ASV systems. In Alegre et al. (2012, 2013), a new approach is shown to detect two types of attacks, artificial signal attacks and the attacks which preserve real-speech phase. It is shown that SVM classifiers are naturally robust against the artificially generated VC attacks (Shabtai et al. 2011).

After several successful research works in this area, the focus has now shifted towards trying to manifest the tremendous potential of DL in ASV also (Scardapane et al. 2017). DL models have shown to substantially improve the sophisticated systems used in SV. By using the backpropagation algorithm, a DNN learns how its parameters should be altered by making computations from representations in the previous layer. A major breakthrough has been observed through the usage of CNNs in several fields like image processing, speech recognition, SV and SI, processing of video, etc. Apart from image data, CNNs perform really well on sequential data like sound also. Authors of Muckenhirn et al. (2018) have used CNN with multi-layer perceptron (MLP), and CNN-based system with off-the-shelf long term spectral statistics (LTSS) multi-layer to detect presentation attack. The work fed end to end audios to the proposed system. Their best combination achieved 0% EER with the product rule for most of the attacks, however on average it achieved 0.157% EER with all the attacks. The work in Qian et al. (2016) applied linear discrete analysis (LDA) with deep features that achieved 1.1% EER on average with all the attack types.

The works in LeCun et al. (2015), analysed the implementation of DNNs for SV. It is shown that through fine-tuning and standard feature extraction methods these DNNs can reach promising results, paving the way for the research community to explore this area further. Neural networks are unstable predictors. Ensemble methods came into picture to take care of the instability of neural networks. When several models are implemented for a problem, there might be disagreement between them. Ensemble methods tend to give out one final result, based on the results of independently trained models (Cunningham et al. 2000). Robustness of ensemble methodology motivated us to introduce it in the proposed work.

The research discussed in Todisco et al. (2019) explains the ASVspoof 2019 challenge completely. The researcher community gave two baseline GMM model trained with ASVspoof 2019 dataset (Mittal and Dua 2021). Frontend of these models are constant Q cepstral coefficients (CQCC)

and linear frequency cepstral coefficients (LFCC). The system with CQCC coefficients is performing better than the system with LFCC coefficients. This motivates the proposed work to use CQCC features for experiments.

Motivated by the merits of CQCC features, ensemble of deep learning models and availability of latest ASVspoof 2019 dataset, the proposed work in this paper focuses on the use of LSTM networks and different types of Convolutional Neural Networks (CNNs) for implementing countermeasures for the spoofing problem (Scardapane et al. 2017; Saranya et al. 2020). In contrast with Recurrent Neural Networks (RNNs), LSTMs are far better because they do not incur the problems of vanishing or exploding gradient descents. Our work is motivated from the promising results this family of neural networks has shown in the recent past. Another more recent technique incorporated for this task is the use of CNNs. TC and SC are also thought of as one possible way of getting better results (Elbayad et al. 2018). In our work, we have implemented several DNNs which confirm that their improvement over existing recurrent networks is commendable. The first DNN comprises of time-distributed dense layers, along with LSTM layers, and is named as tD–LSTM–DNN. The second model with 1D convolutional layers and LSTM is named as TC–LSTM–DNN, as it is in temporal dimension. The DNN based on time-distributed SC is called as tSC–DNN. The proposed work uses ASVspoof 2015 and ASVspoof 2019 datasets for training and testing, with the evaluation set having SS and VC attacked utterances. Also, a joint ASVspoof 1519 dataset is created to add more variations into the single dataset, and it has been observed that the trained ensemble with this joint dataset performs even better, while performing evaluation using single datasets.

The contents of the paper are arranged as follows. Section 2 briefly explains the preliminaries related to neural networks. The description of the dataset and metrics is in Sect. 3. The proposed architecture is illustrated in Sect. 4 of the paper. All the implementation and experimental setup details are given in Sect. 5. Finally, Sect. 6 presents the results of the experiments carried out. Section 7 concludes the paper and presents the future course of action.

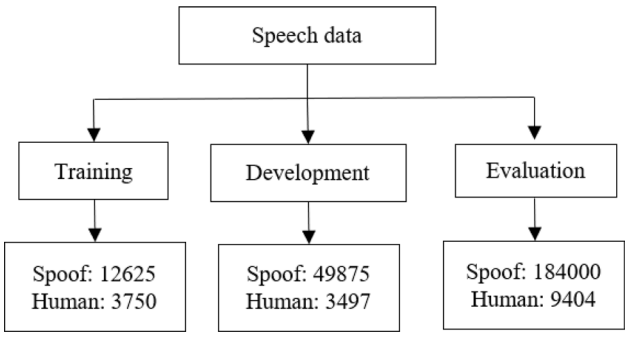


Fig. 1 Categorical split of data between the subsets

2 Preliminaries

2.1 LSTM

RNNs have an internal memory to keep track of the past information processed in the network. Even though they were first designed in 1980s, they weren't widely used due to the lack of computational power of machines. However, with the increase in this computational capability, the real potential of RNNs (and other neural networks, in general) is being identified. Today, they are used to solve problems in almost every domain (Scardapane et al. 2017) such as music, videos, text, weather, etc. RNNs can also be used for finding patterns in images by dividing pixels into a series of data (Graves et al. 2013). But problems like exploding and vanishing gradient descents are associated with RNNs (Bengio et al. 1993).

LSTM is an extension of RNN, which solves the above-mentioned problems. It also makes the network remember the past information for a long time. An LSTM unit contains several gates like Forget, Input and Output gates. LSTM performs better than RNNs in processing, classifying and predicting time sequenced data.

2.2 CNN

CNNs are artificial neural networks (ANNs) which have mostly been used for processing and recognizing images. Recently, the focus has shifted towards using CNNs for times series data.

However, CNNs still take input in the form of images. An input image will be of the form $h \times w \times d$ (h = height, w = width, d = number of channels). A filter belonging to a convolutional layer is convolved over the input image. Feature vectors extracted from a speech signal can be treated just like images and fed into a CNN (Sainath et al. 2015).

A neural network made up of such convolutional layers (also called 2D convolutional layers) is called a spatial convolutional network (SCN). This network uses the process of SC, because the filter moves over the image spatially. On the other hand, there is a category of CNNs called as temporal convolutional network (TCN). Results show that TCNs are

Table 1 Number of spoofed and bona fide utterances dataset

Utterance type	Training	Development	Evaluation
Spoofed	22,800	22,296	63,822
Bona fide	2580	2548	7355

		Decision	
		Accept	**Reject**
Class	**Spoofed (0)**	False Acceptance	True Rejection
	Genuine (1)	True Acceptance	False Rejection

Fig. 2 Possible outcomes of an ASV system

better than Random forests (RFs) and RNNs. This is because TCNs possess longer memory than RNN architectures of the same capacity. Here, the only difference lies in the dimensions of the input data and the filter. 1D filter convolves over 1D input. TCNs have become increasingly popular for sequence modelling tasks. They capture long term dependencies in sequences. The most important advantage of TCNs over RNNs or LSTMs is parallelism. In the recurrent networks, the computation for later time-steps has to wait unit the previous ones are completed. The data is manipulated sequentially. Whereas, in case of TCNs, the same filter is used for all units in a layer, hence an input sequence in TCN will be processed a one whole.

2.3 Ensemble learning

Neural networks possess the immense capability to learn. But a drawback of this flexibility is that they are sensitive to the specifics of the training data. They may result in finding a different set of weights every time they are trained. This in turn produces different predictions. A better approach, which is followed in this work is ensemble method, wherein multiple models are trained instead of just one. Independently trained models can disagree upon their classification. Therefore, the ensemble model gives a final prediction by combining the results of these individual networks. As expected, this method generally outperforms the single models. Max voting and dynamic weighted voting are basic ways to perform ensemble learning.

3 Dataset and metrics

3.1 ASVspoof 2015 dataset

The dataset used in this task is the one released for the ASVspoof Challenge 2015. It is available freely on the Edinburgh DataShare Repository. In this dataset, attention is given to VC and SS attacks. To record the genuine portion of this dataset, 61 females and 45 males are employed. The resulting dataset in then split into three non-overlapping parts. The following figure, Fig. 1 shows the number of spoofed and genuine utterances under each of the 3 subsets—training, development, evaluation. For training the neural network, we used the sound files from both the training and development sub-directories of the ASV 2015 database, in order to increase the input data size (Wu 2014).

Included SS and VC attacks into the dataset are generated by 10 different spoofing attack generation algorithms. The type of spoofing attack can never be known in case of real systems. In order to inculcate this into the dataset, spoofing attacks generated by 5 different algorithms are used in the training part, whereas additional spoofing attacks generated by 5 other algorithms are included in the evaluation set. A total of 5 known attacks, and 5 unknown attacks constitutes the entire evaluation data. Such a division avoids over-fitting the known attacks to spoofing countermeasures.

3.2 ASVspoof 2019 dataset

ASVspoof 2019 dataset was released by ASV community in 2019. It is an English language-based dataset recorded by different groups of males and females (Todisco et al. 2019). Most importantly this dataset has two parts, one contains utterances spoofed by SS and VC attacks and the second contains utterances spoofed by replay attack. Former part is named as logical access (LA) set and the later one is named as physical access (PA) set. Both of these sets have three parts each namely training, development and evaluation. Table 1 gives the information of number of utterances in all parts of LA set. This study focuses on the SS and VC attacks; therefore, LA part is taken into consideration in this paper.

3.3 Joint ASVspoof 1519

To obtain joint ASVspoof 1519 dataset, 70% data of ASVspoof 2015 dataset and 70% data of ASVspoof 2019 are combined. This generated dataset contains more variations in all respects like number of speakers, environmental conditions etc. than the single dataset, and brings new insights for more variations in datasets and large size datasets.

ASVspoof 2019 dataset is derived from VCTK corpus, whereas ASVspoof 2015 dataset is derived from spoofing and anti spoofing (SAS). As both the initial datasets are recorded by different sets of speakers under different environmental conditions, therefore, joint ASVspoof 1519 dataset is phonetically and acoustically enriched. This generated dataset has more variations of spoofing attacks also as the attack systems of ASVspoof 2019 dataset are the modified over the years.

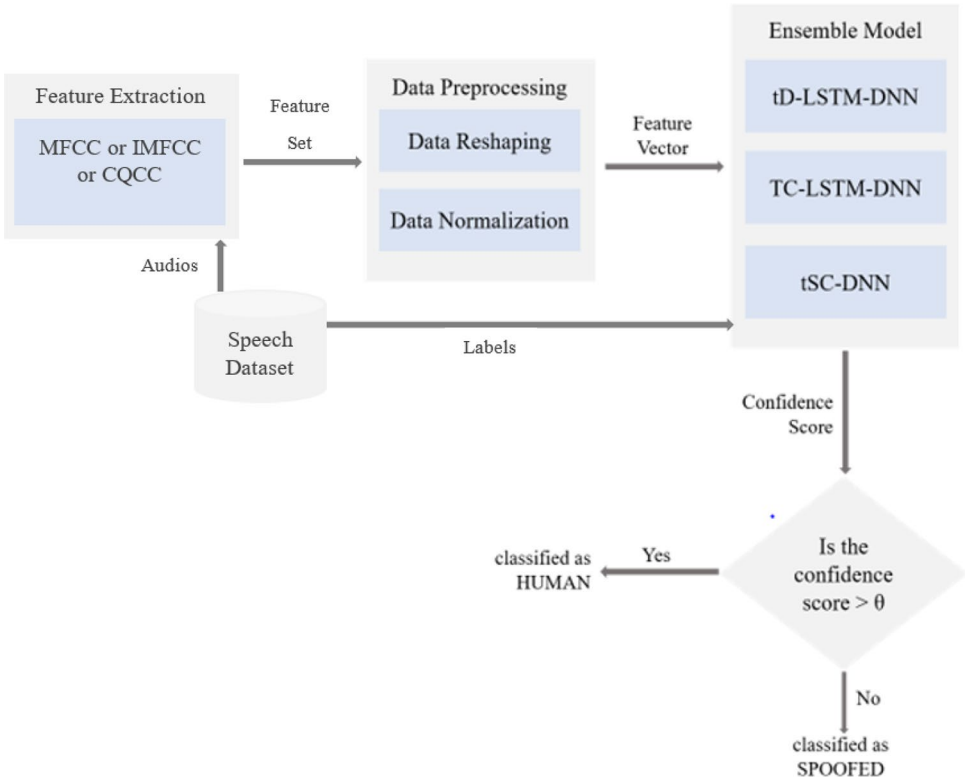


Fig. 3 Proposed system architecture

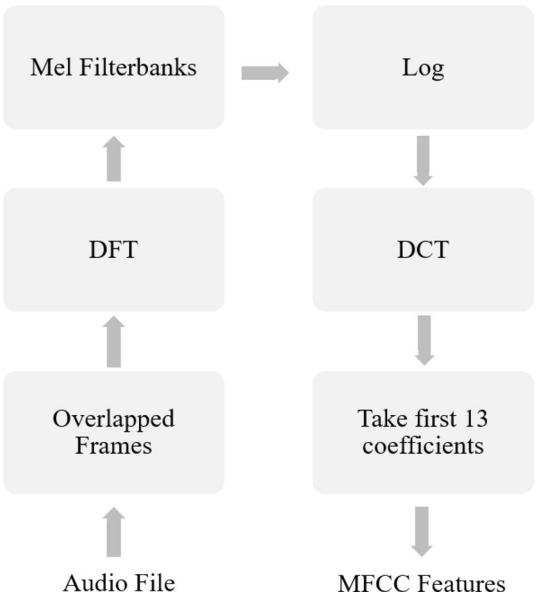


Fig. 4 MFCC feature extraction process

3.4 Metrics

An ASV system can either accept or reject a claim made by a person through the spoken utterance. Hence, for a given ASV system, there is a total of 4 outcomes that can be generated by it. These possible outcomes are shown by Fig. 2.

Once the training of a DNN is completed, the model's accuracy is checked on a common parameter called equal error rate (EER) (Wu 2014). For each test data point, the model results in a confidence score value between 0 and 1, wherein a value closer to 1 indicates high probability of the utterance being genuine and a value tending to 0 referring to spoofed utterance. False acceptance rate (FAR) gives the rate of wrongly classifying a spoofed utterance as genuine. A fraudster is accepted through the system, thereby harming its security.

$$P_{fa}(\theta) = \frac{Number\ of\ spoofed\ trials\ with\ confidence\ score > \theta}{Total\ equation\ number\ of\ spoofed\ trials} \tag{1}$$

P_{fa} denotes the probability of false acceptance. The numerator term here denotes the count of actually spoofed trials which were incorrectly labeled as human, because the confidence score exceeded θ, the threshold.

False rejection rate (FRR), as the name suggests, gives the rate of genuine humans being wrongly classified as spoof, thereby being rejected by the system. This is a measure of how many genuine people were incorrectly denied access. The following value, P_{fr} gives the probability of false rejection.

$$P_{fr}(\theta) = \frac{Number\ of\ genuine\ human\ trials\ with\ \ confidence\ score < \theta}{Total\ equation\ number\ of\ genuine\ trials} \tag{2}$$

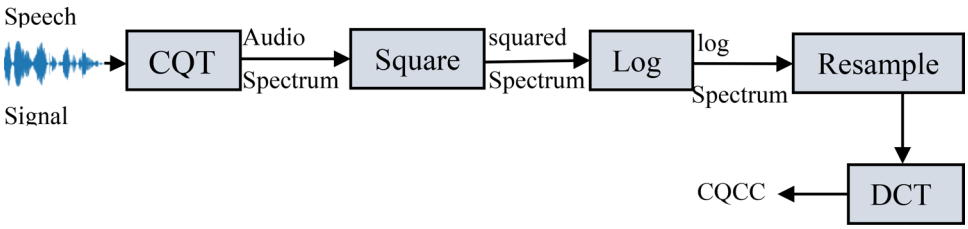


Fig. 5 CQCC feature extraction process

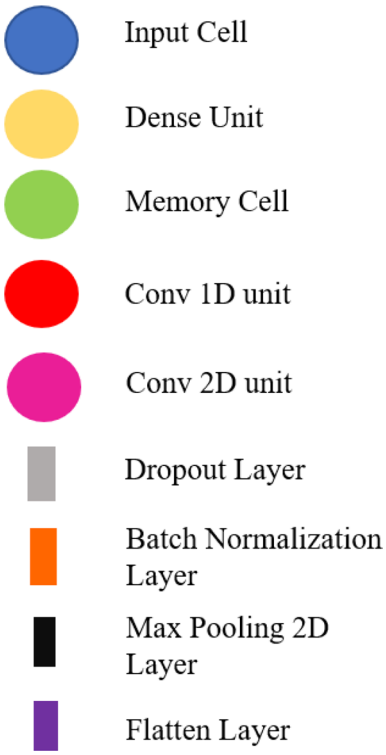


Fig. 6 Symbols with their meanings

As mentioned above, *EER* is selected as the error measure. It is defined by choosing a value of θ, such that $P_{fa}(\theta) = P_{fr}(\theta) = EER$. Thus, the purpose is to find a tradeoff between FAR and FRR, such that the system is optimized to achieve the lowest equal values of P_{fr} and P_{fa}. The lower the value of EER, the better the system. For each of the 10 attacks during testing, *EER* is calculated separately and finally the mean *EER* of the system is evaluated.

4 Proposed work

In the proposed work, ensemble of LSTM and DNNs with temporal convolution and spatial convolution is proposed. Three different features that are MFCC, IMFCC and CQCC are used at the frontend using ASVspoof 2015 dataset for both training and testing to find out the best combination of ensemble and features. The obtained combination is evaluated by training it with ASVspoof 2015 dataset and testing it with completely new ASVspoof 2019 dataset. Also, the combination is trained and tested with ASVspoof 2019 dataset independently. Thirdly, the work uses a new dataset i.e., joint ASVspoof 1519 to add different variation into the single dataset. The combination is trained with the joint

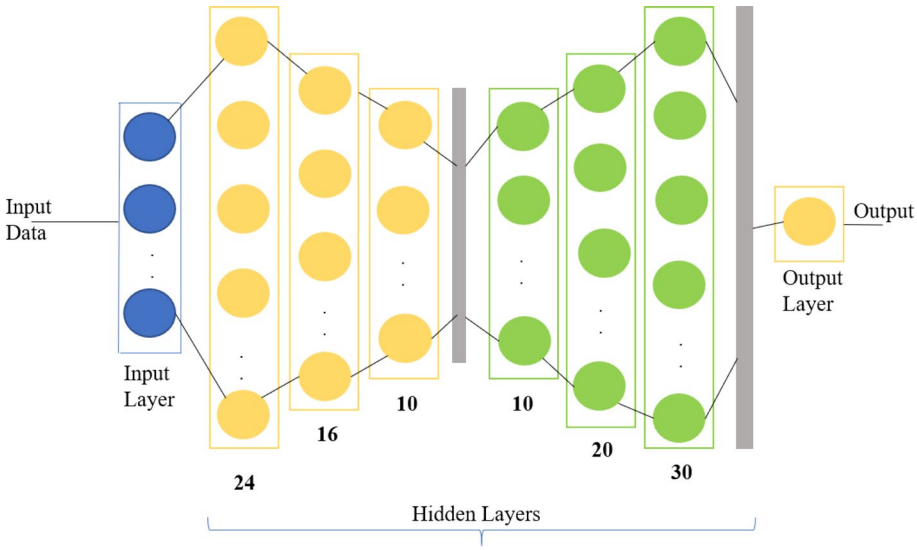


Fig. 7 tD–LSTM–DNN architecture

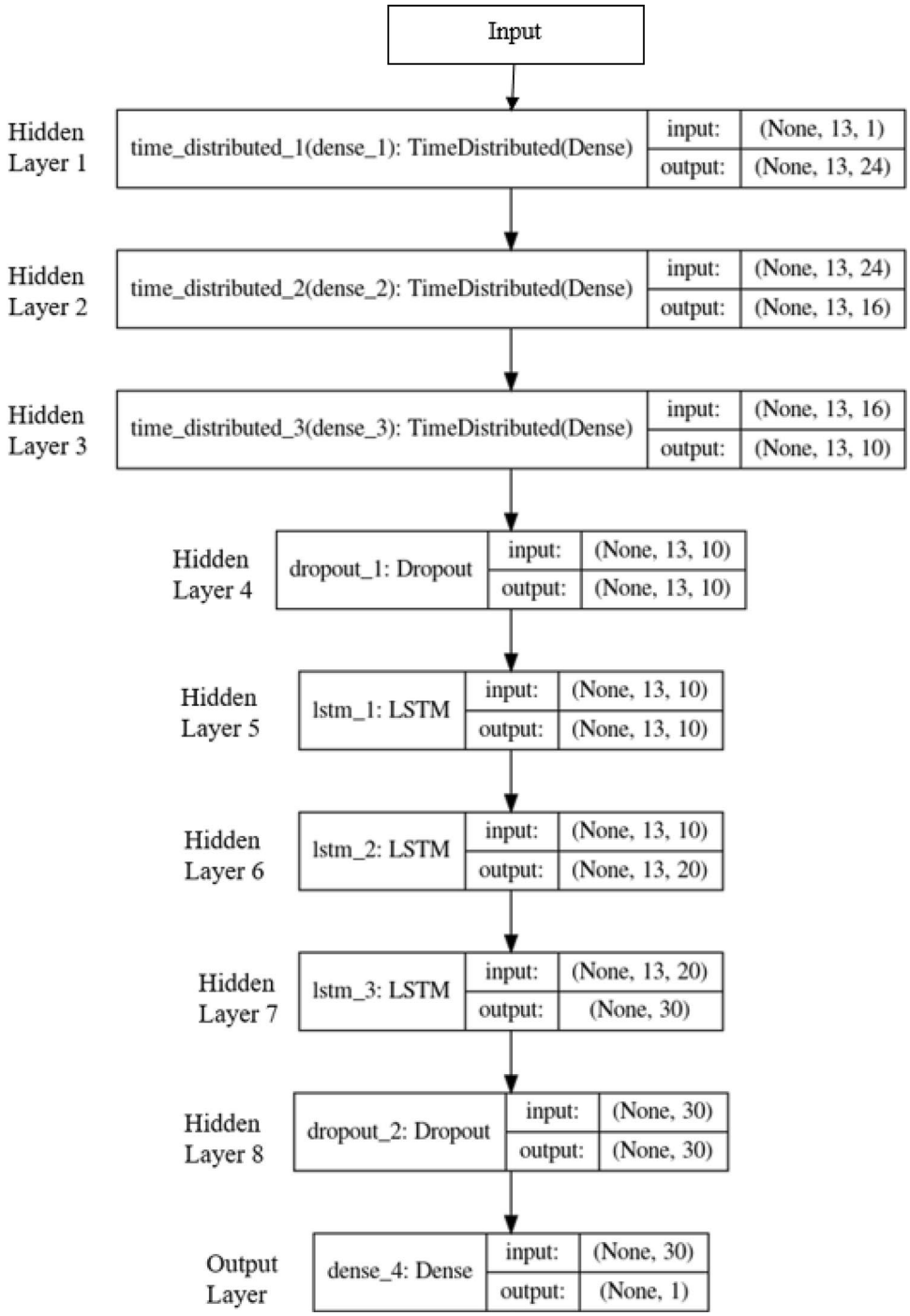


Fig. 8 Detailed architecture of tD–LSTM–DNN

ASVspoof 1519 dataset and performance is evaluated with the ASVspoof 2015 and ASVspoof 2019 datasets.

Figure 3 shows the architecture that has been used to implement the proposed system. From a given test speech signal, firstly, features are extracted. The resulting matrix is then preprocessed by reshaping and normalizing the data. Finally, the feature vector is input to the DNNs separately. Each DNN computes a confidence score. The ensemble model eventually uses these three confidence scores to generate the final confidence score. If this probability value is greater than the threshold θ, the signal is genuine, otherwise it is spoofed. A detailed explanation of each of these components is given in subsequent sections.

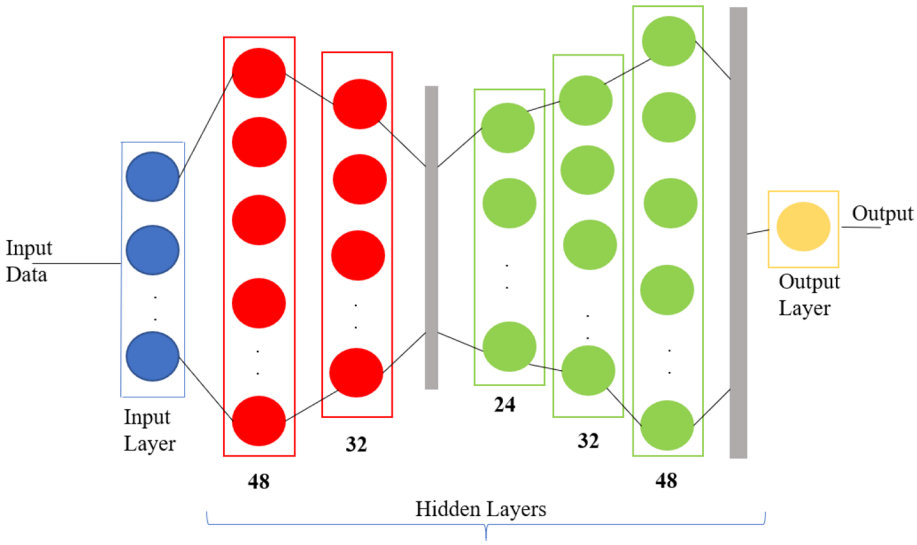


Fig. 9 TC–LSTM–DNN architecture

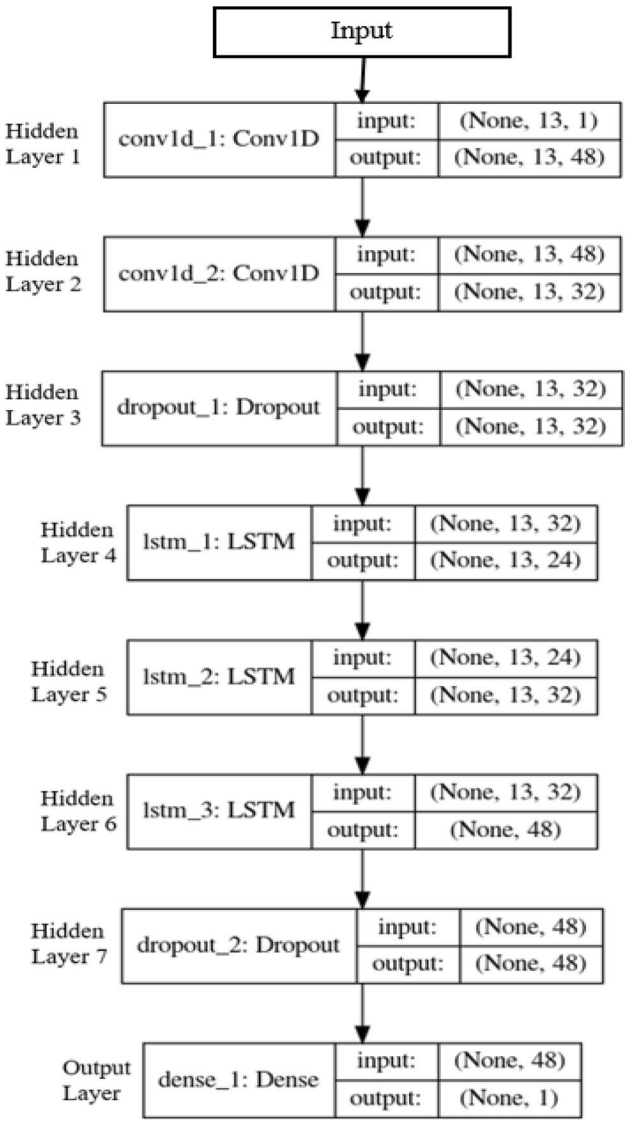


Fig. 10 Detailed architecture of TC–LSTM–DNN

4.1 Feature extraction

The most important step for any speech recognition system is to extract the features that are good at finding linguistic content and discards all other unwanted information like noise, emotions etc. (Dua et al. 2020a).

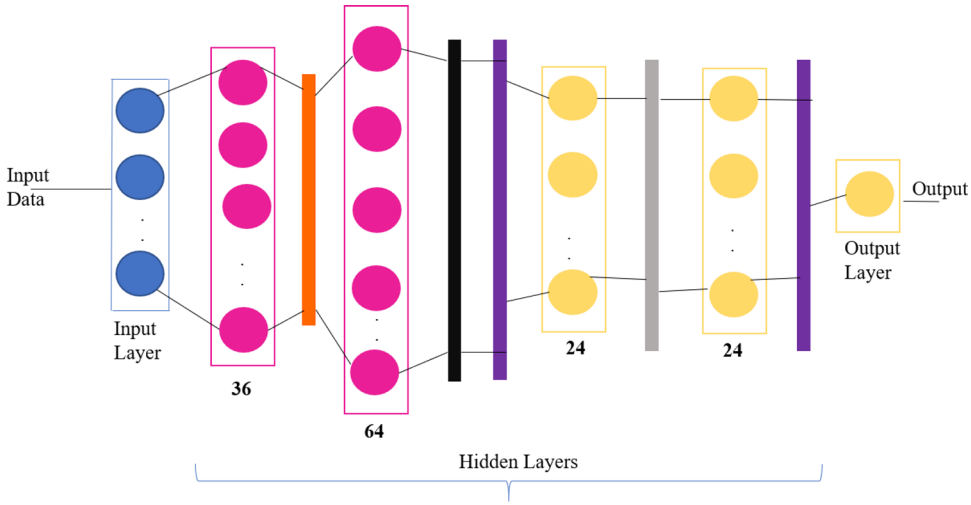
4.1.1 MFCC feature extraction

MFCCs (Hermansky et al. 2000) are the best features at representing human voice and give promising results in clean environments (Dua et al. 2018; Kumar and Aggarwal 2020). Before finding the MFCCs, an analog signal is changed into a digital signal by sampling and quantization at a certain sampling rate. The digital signal then goes through various steps, shown in Fig. 4, which compute the MFCC features. Discrete Fourier transform (DFT) of the signal is computed after dividing the analog signal into overlapped frames. After this, Mel Filterbank and Log filters are applied. Finally, discrete cosine transform (DCT) is applied on the energies (Kuamr et al. 2014a, b; Kumar et al. 2014). Then, the first 13 coefficients are kept, while the higher ones are dropped because they generally reduce the performance of the models (Hossan et al. 2010).

4.1.2 Inverse MFCC feature extraction

MFCC features extract the information of low frequency regions, but the high frequency region is taken care by the inverse Mel frequency Cepstrum Coefficients (IMFCC) features (Mohammadi and Sadegh Mohammadi 2017). Process of extraction of these coefficients is similar to MFCC

Fig. 11 tSC–DNN architecture



extraction only the inverted Mel filter is used instead of Mel filter. For this work 13 IMFCC feature coefficients are extracted.

4.1.3 CQCC feature extraction

Constant Q cepstral coefficients extraction starts with application of constant Q transform (CQT) that maintains the constant Q factor throughout the input audio file (Todisco et al. 2017). Then square of the spectrum is taken before applying the log. CQT leaves the bins in geometric shape that is changed to linear space by doing resampling of the logged spectrum. This process ends with the application of discrete cosine transform (DCT) (Yang et al. 2019). This process gives the desired CQCC features (Fig. 5). This work uses the 30 feature coefficients.

4.2 Architecture of models

This section gives a detailed description of the best three model architectures used for this task. Figure 6 shows the meaning of each entity in the figures that follow it.

The architectures can be best illustrated with supporting diagrams. The following sub-sections describe each model separately.

4.2.1 tD–LSTM–DNN

It consists of three dense layers with 24, 16 and 10 units respectively in the beginning, as shown in Fig. 7. They are followed by a dropout layer. Three LSTM layers with 10, 20 and 30 units respectively are placed sequentially after the dropout layer. Another dropout later is added just before the output layer to avoid overfitting (Srivastava et al. 2014).

This architecture is shown in Fig. 8. It also shows the shape of the input given to each layer along with the shape of the output generated by that layer. First argument (none) gets automatically calculated as the number of audio files in the input. The second argument (13) denotes the 13 MFCC features (Dua et al. 2019a, 2020b) used. The third argument is a variable value used for representing the number of MFCC coefficients in each input time sequence. Proposed model has similar architecture to Fig. 8 when applied with the 13 IMFCC and 30 CQCC features.

4.2.2 TC–LSTM–DNN

It comprises of convolutional 1D layers in the place of dense layers of tD–LSTM–DNN. Two such layers made up of 48 and 32 units respectively, are used in the beginning. They are followed by a dropout layer whose output goes into three sequentially arranged LSTM layers having 24, 32 and 48 units respectively. This model utilizes temporal convolution. Time sequenced data is given as input to the hidden layers. This is shown in Fig. 9.

In Fig. 10, the input and output shapes are also shown along with each layer. These shapes have the same format as described for tD–LSTM–DNN layers. Figure 10 shows the architecture with the application of 13 MFCC features, similar architecture can be obtained for 13 IMFCC and 30 CQCC features too.

4.2.3 tSC–DNN

The first layer is a 2D convolutional layer, followed by batch normalization and another 2D convolutional layer. The next layer is the max pooling layer which works in reducing the dimensionality of the input to lower down the complexity of learning more from the data. Flatten layer simply flattens the data to be given as input to the dense layer. Another dropout layer is added in between, followed by a dense layer

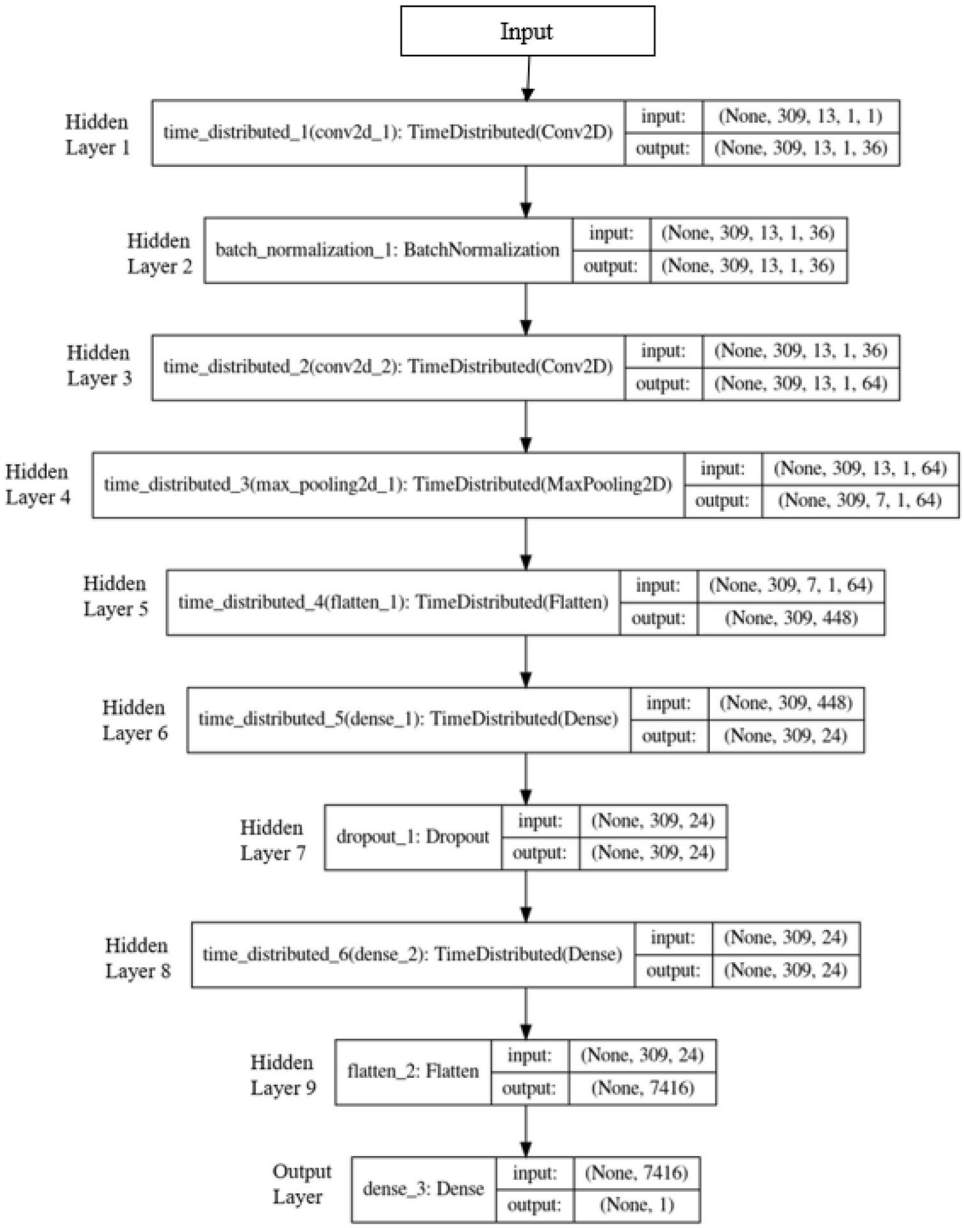


Fig. 12 Detailed architecture of tSC–DNN

and flatten layer. All these architectural details are specified in Fig. 11.

Figure 12 further illustrates this model architecture in a more detailed way. As for the input and output shapes, the first argument still means the same. The second argument (309) represents the height i.e., the number of frames in the feature vector for each audio file. The third argument refers to the 13 MFCC coefficients. Similarly, it can be illustrated for 13 IMFCC and 30 CQCC features too. For time sequencing the data, fourth argument is used. The last argument denotes the number of channels for this image-like data.

4.3 Ensemble model

In this work, the ensemble model comprises of the three DNNs explained above. It uses the predictions and EERs given by these models together to perform a new classification. In order to compute the final result, the ensemble model calculates a weighted confidence score using the scores of the independently trained models. Figure 13 shows the ensemble method.

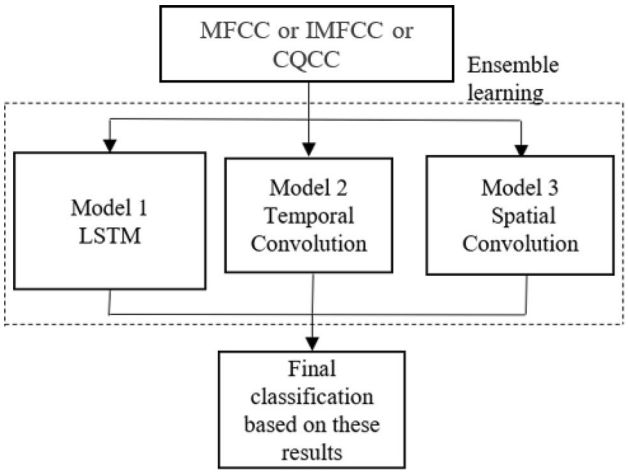


Fig. 13 Ensemble approach

5 Experimental setup

5.1 Feature extraction

For extracting the MFCC features, Librosa module is used. The analog speech signal is first sampled with a sampling rate of 16 kHz. The window size and hop length are 2048 samples and 512 samples respectively (Chakroborty and Saha 2009).

For each audio file, Librosa returns a 2D matrix where each row denotes the MFCC coefficient and each column denotes the frame in that audio file. In order to get a fixed size feature vector for each file, the feature vector is reduced to a 1D matrix consisting of 13 MFCC coefficients. The original 2D vector is first transposed, and then a column-wise mean is taken to get the 1D matrix. This preprocessing is shown in Fig. 14.

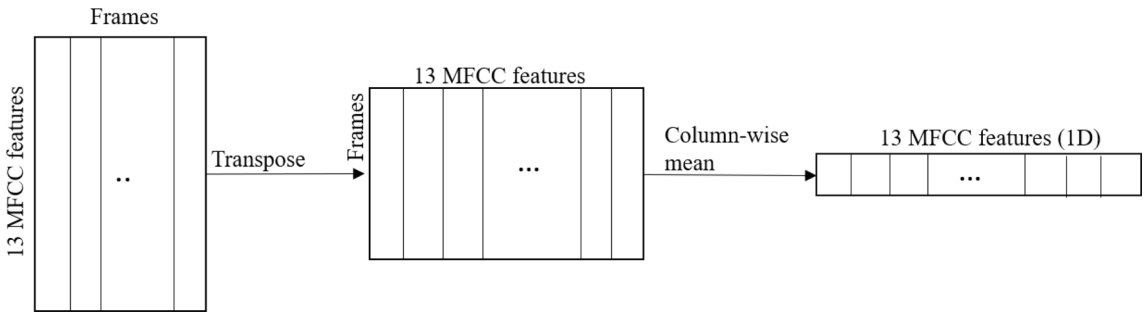


Fig. 14 1D Feature extraction for tD–LSTM–DNN and TC–LSTM–DNN

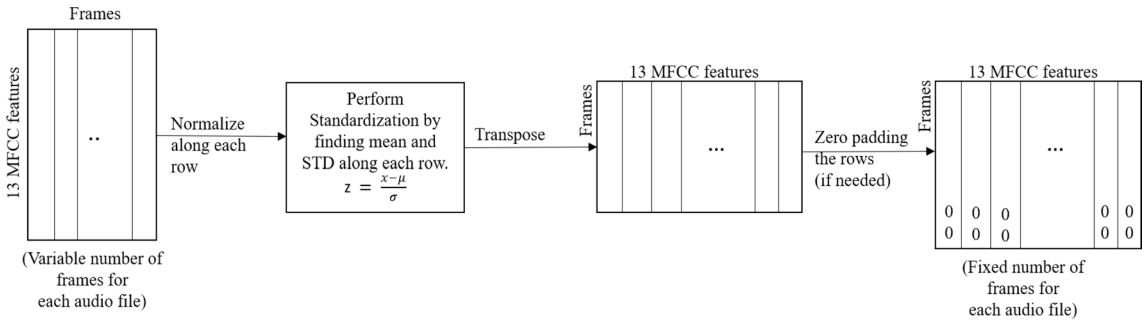


Fig. 15 2D feature extraction for tSC–DNN

Fig. 16 Reshaping of data for tD–LSTM–DNN and TC–LSTM–DNN

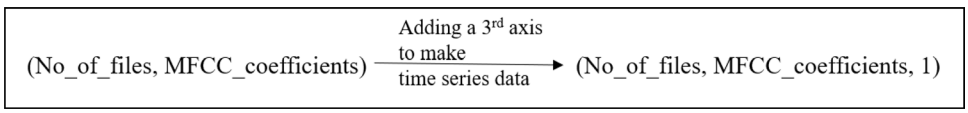


Fig. 17 Reshaping of data for tSC–DNN

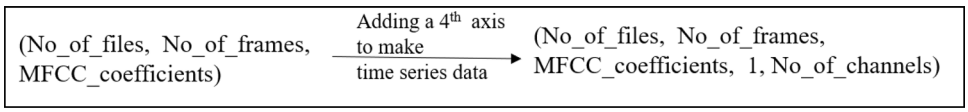


Table 2 Parameters for each DNN

DNN	Parameters		
	Epochs	Learning rate	Dropout
tD–LSTM–DNN	180	0.008	10%
TC–DNN	130	0.002	10%
tSC–DNN	60	0.001	15%

In the ensemble learning method, we perform weighted means on the confidence scores given by the 3 DNNs independently. The weight for each DNN is computed from the respective EER value. After performing the required operations, the resultant weights are – 0.24 for tD–LSTM–DNN, 0.29 for TC–LSTM–DNN and 0.47 for tSC–DNN. The sum of these weights is 1. Therefore, by calculating the weighted average confidence score, the ensemble model further improves the results, when applied on the ASVspoof 2015 dataset. These models are trained by the ASVspoof 2019 (Kamble et al. 2020) and JointASV1519 datasets with the parameters to add fairness to the results

Table 3 EER (%) with MFCC at frontend, best results are shown in bold

Attack type	EER (%)			
	tD–LSTM–DNN	TC–LSTM–DNN	tSC–DNN	Ensemble
S1	0.5	1.6	0.1	0.2
S2	3.3	6.8	0.8	0.7
S3	10.5	9.1	0.4	0.8
S4	10.2	7.3	1.7	1.1
S5	11.8	10.1	1.4	1.8
S6	20.8	12.5	5.2	3.8
S7	11.5	9.5	0.7	1.2
S8	1.1	0.6	0.9	1.7
S9	11.0	7.1	1.1	0.1
S10	16.7	19.3	6.1	5.8
Average	9.7	8.4	1.9	**1.7**

Table 4 EER (%) with MFCC at the frontend, best results are in bold

Model	EER (%)		
	Known attacks	Unknown attacks	Average
tD–LSTM–DNN	7.3	12.2	9.7
TC–LSTM–DNN	7.0	9.8	8.4
tSC–DNN	0.9	2.8	1.9
Ensemble	0.9	2.5	**1.7**

Further, Fig. 18 shows the EER values given by the DNNs for each of the 10 attack types. These results clearly confirm that deep learning can pave the way for market-wide adoption of voice-based authentication systems. As this field attracts even more researchers, deep learning would definitely improve the results in the future

tSC–DNN uses image-like feature vector. Therefore, a 2D matrix is used for training it. However, the MFCC coefficients have to be standardized before the matrix enters the model for training. This is done by finding mean and standard deviation across each row (each MFCC coefficient) to get a standardized value in that row.

$$z = \frac{x - \mu}{\sigma} \tag{3}$$

Each audio file is of varying duration, which in turn implies that the number of frames is also variable. To get a fixed number of frames for all files without losing any information, zero padding is done by adding extra rows in the transposed and normalized matrix. Figure 15 illustrates this processing.

13 IMFCC features are extracted on the Python platform and the process similar to MFCC is used for preprocessing the features (Hourri and Kharroubi 2019) into 1D and 2D vectors. In case of CQCC features, these are extracted on octave platform on Ubuntu system. 30 feature coefficients are extracted for each of 400 frames of an audio file. Both 1D and 2D feature vectors are obtained by the above discussed preprocessing method.

5.2 DNNs

The DNNs are implemented in Python, using the Keras library. Before feeding the processed feature vector into the implemented models, it must be reshaped into the desired format. Figure 16 explains the reshaped format for models tD–LSTM–DNN and TC–LSTM–DNN, whereas Fig. 17 represents the same for tSC–DNN. Models have similar shapes as described in Figs. 16 and 17 for IMFCC and CQCC features also.

While training the DNNs, we minimize mean squared error (MSE) on the training dataset.

We use a batch size of 500 speech signals for updating the weights and biases. A dropout of 10% is used in tD–LSTM–DNN and TC–LSTM–DNN to avoid any kind of overfitting of the data. tSC–DNN uses 15% dropout. We also use batch normalization in tSC–DNN. It normalizes data before it goes into a hidden layer to result in a more accurate and stable model. It leads to reduction in overfitting by providing slight regularization effects. Together with dropout, it gives better results.

The output layer in each model is made up of 1 dense unit, having sigmoid activation function to handle binary classification. All the hidden dense layers in tD–LSTM–DNN have ReLu activation function. All the convolutional layers in TC–LSTM–DNN are also using the ReLu activation function. Table 2 shows the other parameters specific to each model.

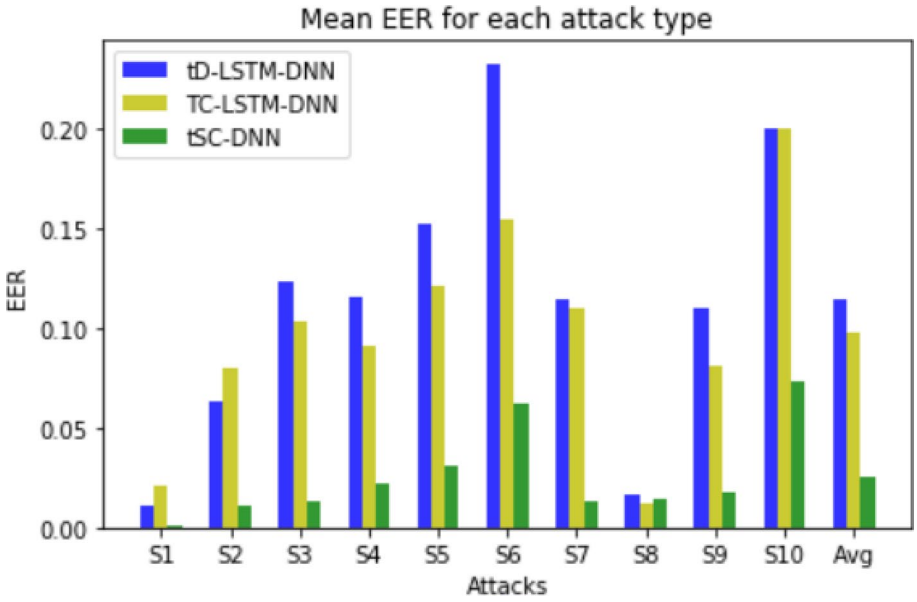


Fig. 18 Bar graph showing the EER values for each attack

Table 5 EER (%) with IMFCC at the frontend, best results are in bold

Model	EER (%)		
	Known attacks	Unknown attacks	Average
tD–LSTM–DNN	7.3	10.9	9.1
TC–LSTM–DNN	6.7	10.1	8.4
tSC–DNN	0.6	2.0	1.3
Ensemble	0.1	1.7	**0.9**

Training of these models with the CQCC features at the frontend is also done. Using CQCC features at the frontend of these proposed models performs the best as shown in Table 6. Therefore, CQCC features are chosen for further work

Table 6 EER (%) with CQCC at the frontend, best results are in bold

Model	EER (%)		
	Known attacks	Unknown attacks	Average
tD–LSTM–DNN	3.5	10.7	7.1
TC–LSTM–DNN	0.9	6.9	3.9
tSC–DNN	0.09	1.31	0.7
Ensemble	0.01	0.19	**0.1**

Table 7 Evaluation of proposed ensemble with different datasets

Model	Training set	Evaluation set	EER (%)
Ensemble	ASVspoof 2015	ASVspoof 2019	24.8
	ASVspoof 2019	ASVspoof 2019	0.81
	JointASVspoof 1519	ASVspoof 2015	**0.09**
	JointASVspoof 1519	ASVspoof 2019	**0.6**

6 Results

Each of the three DNNs gives promising results on the testing audio files. As expected, the ensemble learning model gives the best results as it uses a combination of the results given by each model trained separately.

As per the metrics described for this problem, the EER value is calculated separately for each model. In order to do so, receiver operating characteristic (ROC) curve is plotted between false positive rate (FPR) on the x-axis and true positive rate (TPR) on the y-axis (Devi and Thongam 2019; Tagomori et al. 2020). The FPR value of the intersection point between ROC curve and the thresholds line gives the corresponding EER.

Firstly, all the models are evaluated with the ASVspoof 2015 dataset. All the models are evaluated by taking the MFCC, IMFCC and CQCC features at the frontend. Performances of these models are compared, and the best performing features are considered in further studies. After this, models are trained with the ASVspoof 2019 (Kamble et al. 2020) and JointASV1519 datasets and performances are compared.

6.1 Performance of models with ASVspoof 2015 dataset

For each model, the EER value is evaluated over each of the 10 attacks when MFCC features are used at the frontend. The average EER for that model is calculated by averaging the 10 EERs. For tD–LSTM–DNN, the average EER comes out to be 9.7%. Similarly, for TC–LSTM–DNN, the average EER value is computed to be 8.4%. It is evident that the EER reduced from the first DNN. For tSC–DNN,

Table 8 Comparison of proposed approach with existing techniques

Approach	Feature extraction	Classifiers	Data-set	Performance rate (EER in %)
Qian et al. (2016)	Deep features	Linear discriminant analysis (LDA)	ASVspoof 2015	1.1
Zhang et al. (2017)	Spectrogram features	CNN RNN CNN + RNN Ensemble learning (all above)	ASVspoof 2015	1.47
Muckenhirn et al. (2017)	End to end audio	CNN SLP LTSS MLP Product rule combination	ASVspoof 2015	0.157
Dinke et al. (2018)	CQCC8 k-DD	CLDN (convolutional LSTM neural network, joint)	ASVspoof 2015	4.56
Proposed approach	CQCC 1D CQCC 2D	tD–LSTM–DNN, TC–LSTM–DNN, tSC–DNN, ensemble learning (all above)	ASVspoof 2015	**0.09**

Table 9 Comparison of proposed approach with existing techniques

Approach	Feature extraction	Classifiers	Data-set	Performance rate (EER in %)
Todisco et al. (2019)	CQCC	GMM (baseline system)	ASVspoof 2019	0.95
Chettri et al. (2019)	SDA, MFCC + IMFCC + CQCC + SCMC	CNN, CRNN, 1DCNN, Wave-U-Net, GMMs, SVMs, ensemble learning (all above)	ASVspoof 2019	Ensemble1 (2.64), Ensemble2 (9.57), Ensemble3 (10.63)
Proposed approach	CQCC 1D CQCC 2D	tD–LSTM–DNN, TC–LSTM–DNN, tSC–DNN, ensemble learning (all above)	ASVspoof 2019	**0.6**

the EER value is 1.9%, which is significantly lower than those of the other DNNs. Finally, the ensemble model gives the lowest value of EER as 1.7%. Table 3 shows these EERs.

Table 4 below shows the mean EER separately for the broad category of attacks, i.e., known attacks and unknown attacks. Intuitively, it would be expected that the EER for unknown attacks would be higher than that for known attacks.

Training of these models with the IMFCC features at the frontend gives a better performance than the MFCC features at the frontend. Table 5 shows the performances of these models with IMFCC at the frontend for known and unknown attacks, and average performance is also shown.

6.2 Analysis of proposed system with different datasets

Proposed ensemble (CQCC at frontend) trained with ASVspoof 2015 dataset is used to test the data from ASVspoof 2019 dataset. As the data for testing is completely new for this trained model, performance of the model degrades.

Proposed ensemble is trained with ASVspoof 2019 dataset (Kamble et al. 2020) and joint ASVspoof 1519 dataset, also. Performance results of trained system on various evaluation sets is shown in Table 7, best results are in bold. With the evaluation sets of ASVspoof 2015 and ASVspoof 2019 datasets this ensemble is preforming the best. All the analysis results are shown in Table 7.

6.3 Analysis and comparison with earlier proposed techniques

Studies in the area of ASV systems work towards enabling wide adoption of speech-based authentication systems in the market. Our work is primarily focused on the backend and feature set that explores the performance of neural networks. Our best performing ensemble is the one with CQCC at the frontend. Overall, ensemble trained with JointASVspoof 1519 dataset performs the best with ASVspoof 2015 and ASVspoof 2019 datasets, individually. Tables 8 and 9 shows the comparison of our best proposed architecture with the work from the literature. From any author's work we have chosen the best performing systems on average for all types of attacks for comparison. The EER values are evaluated over the evaluation data.

7 Conclusion and future work

The tSC–DNN model is based on spatial convolution and treats the speech feature vector as an image. The first model proposed in this paper, tD–LSTM–DNN, uses the conventional LSTM layers that have been used for a long time now. But the results given by such a model are somewhat less impressive. Besides that, temporal convolution applied in TC–LSTM–DNN provides better results than tD–LSTM–DNN because of using 1D convolutional layers. MFCC features are used in this experimentation because many studies suggest that this feature set provides the best results on speech data. Along with these IMFCC and latest introduced CQCC features are also used at the front end. IMFCC features at frontend gives better performance than MFCC at frontend. However, proposed ensemble performs the best with CQCC at the frontend, when evaluated with the ASVspoof 2015 dataset. Further, the system trained with ASVspoof 2019 dataset gives better performance than baseline system on evaluation sets. Joint ASVspoof 1519 dataset gives new insights to add more number of speakers, languages, environmental conditions, etc. to the dataset. The ensemble trained with the JointASVspoof 1519 dataset performs the best when evaluated with the ASVspoof 2015 and ASVspoof 2019 datasets. For future work, these same models can be trained on varying feature vectors to analyze their performance. Besides these, deeper CNNs and LSTMs can be implemented to improve the accuracy and lower the EER.

Compliance with ethical standards

Conflict of interest The submitted work does not have any conflict of interest.

References

Alegre F, Vipperla R, Evans N (2012) Spoofing countermeasures for the protection of automatic speaker recognition from attacks with artificial signals. In: 13th annual conference of the international speech communication association 2012, INTERSPEECH 2012, pp 1686–1689

Alegre F, Amehraye A, Evans N (2013) Spoofing countermeasures to protect automatic speaker verification from voice conversion. In: ICASSP, IEEE international conference on acoustics, speech and signal processing—proceedings. IEEE, pp 3068–3072

Aleksic PS, Katsaggelos AK (2006) Audio-visual biometrics. Proc IEEE 94:2025–2044

Bengio Y, Frasconi P, Simard P (1993) Problem of learning long-term dependencies in recurrent networks. In: 1993 IEEE international conference on neural networks. IEEE, pp 1183–1188

Chakroborty S, Saha G (2009) Improved text-independent speaker identification using fused MFCC and IMFCC feature sets based on Gaussian filter. World Acad Sci Eng Technol 35:613–621

Chen LW, Guo W, Dai LR (2010) Speaker verification against synthetic speech. In: 2010 7th international symposium on Chinese spoken language processing, ISCSLP 2010—proceedings. IEEE, pp 309–312

Chettri B, Stoller D, Morfi V et al (2019) Ensemble models for spoofing detection in automatic speaker verification. arXiv

Cunningham P, Carney J, Jacob S (2000) Stability problems with artificial neural networks and the ensemble solution. Artif Intell Med 20:217–225. https://doi.org/10.1016/S0933-3657(00)00065-8

De Leon PL, Pucher M, Yamagishi J et al (2012a) Evaluation of speaker verification security and detection of HMM-based synthetic speech. IEEE Trans Audio Speech Lang Process 20:2280–2290. https://doi.org/10.1109/TASL.2012.2201472

De Leon PL, Stewart B, Yamagishi J (2012b) Synthetic speech discrimination using pitch pattern statistics derived from image analysis. In: 13th annual conference of the international speech communication association 2012, INTERSPEECH 2012, pp 370–373

Devi KJ, Thongam K (2019) Automatic speaker recognition with enhanced swallow swarm optimization and ensemble classification model from speech signals. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-019-01414-y

Dinkel H, Qian Y, Yu K (2018) Investigating raw wave deep neural networks for end-to-end speaker spoofing detection. IEEE/ACM Trans Audio Speech Lang Process 26:2002–2014. https://doi.org/10.1109/TASLP.2018.2851155

Dua M, Kumar A, Chaudhary T (2015) Implementation and performance evaluation of speaker adaptive continuous Hindi ASR using tri-phone based acoustic modelling. In: Proceedings of 2015 international conference on future computational technologies, pp 68–73

Dua M, Aggarwal RK, Biswas M (2017) Discriminative training using heterogeneous feature vector for Hindi automatic speech recognition system. In: 2017 international conference on computer and applications, ICCA 2017. IEEE, pp 158–162

Dua M, Aggarwal RK, Biswas M (2018) Performance evaluation of Hindi speech recognition system using optimized filterbanks. Eng Sci Technol Int J 21:389–398. https://doi.org/10.1016/j.jestch.2018.04.005

Dua M, Aggarwal RK, Biswas M (2019a) GFCC based discriminatively trained noise robust continuous ASR system for Hindi language. J Ambient Intell Humaniz Comput 10:2301–2314. https://doi.org/10.1007/s12652-018-0828-x

Dua M, Wesanekar A, Gupta V et al (2019b) Color image encryption using synchronous CML-DNA and weighted bi-objective genetic algorithm. In: ACM international conference proceeding series, pp 121–125

Dua M, Aggarwal RK, Biswas M (2020) Optimizing integrated features for Hindi automatic speech recognition system. J Intell Syst 29:959–976. https://doi.org/10.1515/jisys-2018-0057

Dua M, Aggarwal RK, Biswas M (2020) Discriminative training using noise robust integrated features and refined HMM modeling. J Intell Syst 29:327–344. https://doi.org/10.1515/jisys-2017-0618

Elbayad M, Besacier L, Verbeek J (2018) Pervasive attention: 2D convolutional neural networks for sequence-to-sequence prediction. arXiv

Evans NWD, Kinnunen T, Yamagishi J (2013) Spoofing and countermeasures for automatic speaker verification. In: Interspeech, pp 925–929

Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: ICASSP, IEEE international conference on acoustics, speech and signal processing—proceedings. IEEE, pp 6645–6649

Hermansky H, Ellis DPW, Sharma S (2000) Tandem connectionist feature extraction for conventional HMM systems. In: ICASSP, IEEE international conference on acoustics, speech and signal processing—proceedings. IEEE, pp 1635–1638

Hossan MA, Memon S, Gregory MA (2010) A novel approach for MFCC feature extraction. In: 4th international conference on signal processing and communication systems, ICSPCS'2010—proceedings. IEEE, pp 1–5

Hourri S, Kharroubi J (2019) A novel scoring method based on distance calculation for similarity measurement in text-independent speaker verification. Procedia Comput Sci 148:256–265. https://doi.org/10.1016/j.procs.2019.01.068

Kamble MR, Sailor HB, Patil HA, Li H (2020) Advances in anti-spoofing: from the perspective of ASVspoof challenges. APSIPA Trans Signal Inf Process. https://doi.org/10.1017/ATSIP.2019.21

Kuamr A, Dua M, Choudhary A (2014a) Implementation and performance evaluation of continuous Hindi speech recognition. In: 2014 international conference on electronics and communication systems, ICECS 2014. IEEE, pp 1–5

Kuamr A, Dua M, Choudhary T (2014b) Continuous Hindi speech recognition using Gaussian mixture HMM. In: 2014 IEEE Students' conference on electrical, electronics and computer science, SCEECS 2014. IEEE, pp 1–5

Kumar A, Aggarwal RK (2020) Discriminatively trained continuous Hindi speech recognition using integrated acoustic features and recurrent neural network language modeling. J Intell Syst 30:165–179. https://doi.org/10.1515/jisys-2018-0417

Kumar A, Dua M, Choudhary T (2014) Continuous hindi speech recognition using monophone based acoustic modeling. Int J Comput Appl ICACEA(1):15–19

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444

Mittal A, Dua M (2021) Constant Q cepstral coefficients and long short-term memory model-based automatic speaker verification system. In: Proceedings of international conference on intelligent computing, information and control systems. Springer, pp 895–904

Mohammadi M, Sadegh Mohammadi HR (2017) Robust features fusion for text independent speaker verification enhancement in noisy environments. In: 2017 25th Iranian conference on electrical engineering, ICEE 2017. IEEE, pp 1863–1868

Muckenhirn H, Magimai-Doss M, Marcel S (2018) End-to-end convolutional neural network-based voice presentation attack detection. In: IEEE international joint conference on biometrics, IJCB 2017. IEEE, pp 335–341

Qian Y, Chen N, Yu K (2016) Deep features for automatic spoofing detection. Speech Commun 85:43–52. https://doi.org/10.1016/j.specom.2016.10.007

Sahu P, Dua M (2016) An overview: context-dependent acoustic modeling for LVCSR. In: Proceedings of the 10th INDIACom; 2016 3rd international conference on computing for sustainable global development, INDIACom 2016. IEEE, pp 2223–2227

Sahu P, Dua M (2017) A quinphone-based context-dependent acoustic modeling for LVCSR. Advances in intelligent systems and computing. Springer, Berlin, pp 105–111

Sahu P, Dua M, Kumar A (2018) Challenges and issues in adopting speech recognition. Advances in intelligent systems and computing. Springer, Singapore, pp 209–215

Sainath TN, Vinyals O, Senior A, Sak H (2015) Convolutional, long short-term memory, fully connected deep neural networks. In: ICASSP, IEEE international conference on acoustics, speech and signal processing—proceedings. IEEE, pp 4580–4584

Saranya S, Rupesh Kumar S, Bharathi B (2020) Deep learning approach: detection of replay attack in ASV systems. Advances in intelligent systems and computing. Springer, Berlin, pp 291–298

Satoh T, Masuko T, Kobayashi T, Tokuda K (2001) A robust speaker verification system against imposture using an HMM-based speech synthesis system. In: EUROSPEECH 2001—SCANDINAVIA—7th European conference on speech communication and technology, pp 759–762

Scardapane S, Stoffl L, Rohrbein F, Uncini A (2017) On the use of deep recurrent neural networks for detecting audio spoofing attacks. In: Proceedings of the international joint conference on neural networks. IEEE, pp 3483–3490

Shabtai NR, Rafaely B, Zigel Y (2011) The effect of reverberation on the performance of cepstral mean subtraction in speaker verification. Appl Acoust 72:124–126. https://doi.org/10.1016/j.apacoust.2010.09.009

Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958

Tagomori T, Tsuruda R, Matsuo K, Kurogi S (2020) Speaker verification from mixture of speech and non-speech audio signals via using pole distribution of piecewise linear predictive coding coefficients. J Ambient Intell Humaniz Comput 1–11. https://doi.org/10.1007/s12652-020-01716-6

Toda T, Tokuda K (2007) A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. IEICE Trans Inf Syst 90:816–824

Todisco M, Delgado H, Evans N (2017) Constant Q cepstral coefficients: a spoofing countermeasure for automatic speaker verification. Comput Speech Lang 45:516–535. https://doi.org/10.1016/j.csl.2017.01.001

Todisco M, Wang X, Vestman V et al (2019) ASVspoof 2019: future horizons in spoofed and fake audio detection. arXiv

Wu ZK (2014) ASVspoof 2015: Automatic speaker verification spoofing and countermeasures challenge evaluation plan. Training 10:3750. https://doi.org/10.7488/ds/298

Wu Z, Kinnunen T, Evans N et al (2015) ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In: Sixteenth annual conference of the international speech communication association

Yamagishi J, Kinnunen TH, Evans N et al (2017) Introduction to the issue on spoofing and countermeasures for automatic speaker verification. IEEE J Sel Top Signal Process 11:585–587. https://doi.org/10.1109/JSTSP.2017.2698143

Yang J, Das RK, Li H (2019) Extended constant-Q Cepstral coefficients for detection of spoofing attacks. In: 2018 Asia-Pacific signal and information processing association annual summit and conference, APSIPA ASC 2018—Proceedings. IEEE, pp 1024–1029

Zhang C, Yu C, Hansen JHL (2017) An investigation of deep-learning frameworks for speaker verification antispoofing. IEEE J Sel Top Signal Process 11:684–694. https://doi.org/10.1109/JSTSP.2016.2647199