



# Noise Robust ASV Spoof Detection Using Integrated Features and Time Delay Neural Network

Nidhi Chakravarty<sup>1</sup> · Mohit Dua<sup>1</sup>

Received: 11 September 2022 / Accepted: 11 December 2022 / Published online: 26 December 2022  
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2022

## Abstract

Recent advancements in research for development of countermeasure systems for Spoofed Audio detection has helped in building more robust Automatic Speaker Verification (ASV) System. However, available countermeasure systems are not able to generalize well against unknown attacks. The lack of context-dependent information extracted from the given speech at fine grained level is the dominating reason for poor performance of these systems against unknown attacks. To build a noise robust anti-spoof system, in this paper, we propose a Time Delay Neural Network (TDNN)-based countermeasure system that captures context-dependent information well. We devise a three-stage design where at first audio is pre-processed to extract useful information using three different types of features, that are, Mel Frequency Cepstral Coefficients (MFCC), noise robust Gammatone Cepstral Coefficients (GTCC) features and integration of MFCC-GTCC features. These features are then input to proposed Deep Neural Network (DNN) model that uses Long Short-Term Memory (LSTM) network for recurrent aggregation of layer wise generated shallow features in TDNN. Finally, the output is passed through context-dependent pooling layer to generate fixed-length representation that is further used at third stage to classify speech as genuine or spoofed. The proposed system is tested on Logical Access (LA) track of ASV Spoof 2019 dataset, and achieves performance improvement of about 59.7% and 65.9% relative to earlier proposed Linear-Frequency Cepstral Coefficients-Gaussian Mixture Model (LFCC-GMM) and Constant Q Cepstral Coefficients-Gaussian Mixture Model (CQCC-GMM) baseline models, respectively.

**Keywords** ASV · Deep neural network · X-vector · MFCC · GTCC · LSTM

## Introduction

Speech of an individual is prevalently being used as means of his/her authentication in modern security systems. With advancement in audio processing techniques, systems are able to synthesize speech that is able to bypass audio-based security systems. Synthetic speech can be generated using Text-to-Speech (TTS) systems as well as Voice Conversion (VC) systems. The quality of synthetic speech produced

by available state-of-the-art systems is quite hard to distinguish from the real one. Such systems are able to produce converted voice signals with significant naturalness and much similarity to given speaker, thus posing considerable threats to reliability of verification systems [1]. This has led to research for development of countermeasure systems as a part of Automatic Speaker Verification (ASV) Systems for detection of spoofed speech [2]

ASV spoof challenge [3] is catering to need of development of such countermeasure systems by providing a common platform and datasets to evaluate performance of system. Datasets periodically released under the challenge includes spoofed audio with latest algorithms along with the genuine one. In this series of challenges, the dataset released under ASV Spoof 2019 [4] challenge is divided in two categories: Logical Access (LA) and Physical Access (PA). LA involves spoofed audio synthesized from the genuine data using latest available TTS and VC technologies including number of different spoofing algorithms, whereas PA includes replay attacks. The LA dataset includes genuine

---

This article is part of the topical collection “Enabling Innovative Computational Intelligence Technologies for IOT” guest edited by Omer Rana, Rajiv Misra, Alexander Pfeiffer, Luigi Troiano and Nishtha Kesswani.

---

✉ Nidhi Chakravarty  
nidhi\_62000061@nitkkr.ac.in

Mohit Dua  
er.mohitdua@nitkkr.ac.in

<sup>1</sup> Department of Computer Engineering, National Institute of Technology, Kurukshetra, India

as well the spoofed audio from 107 speakers [5]. In building generalized countermeasure system existing efforts that are based on deep neural networks focus on three categories: feature extraction as front end, system modelling as back end, effective loss function. At front ends, features like Mel Frequency Cepstral Coefficients (MFCC) [6], Linear-Frequency Cepstral Coefficients (LFCC) [7], Constant Q Cepstral Coefficients (CQCC) [8, 9] have been dominated such classification task. These front ends employ some sort of filters to model non-linearity of human ear's sensitivity to frequency. Mel Scale captures pitch as perceived by humans comparatively well thus explaining the dominance of MFCC features in such task. Gammatone Cepstral Coefficients (GTCC) [10] features are less explored in this domain, they employ ERB frequency scale with GT filters instead of Mel scale that has proven to be more effective in noisy environment. Motivated by the previously used front-end techniques, the front end of our system has been built using integrated traditional feature MFCC and noise robust GTCC features [11].

Time Delay Neural Network (TDNN) [12] is a layered feed forward neural network that demonstrates the ability to learn invariant relationships between events that are in form of frame level spectral coefficients across time. Using TDNN, we take advantage of temporal nature of acoustic signals by varying the context across hidden layers thus converting complex acoustic signal into abstract representation. X-vectors [13] are such representations that have been proven effective in speaker verification tasks. Subsequent improvements to TDNN in x-vector architecture have further enhanced the quality of final fixed representation produced [14]. The backend of our work is built upon original x-vector architecture with induction of various independent modules as an improvement in topology. We have also utilised skip connections across layers to capture deeper level features and then efficiently aggregate them using LSTM aggregator [15].

The remaining part of the paper is laid out as follows. The related work is presented in “[Related Works](#)”. “[Preliminaries](#)” provides a brief discussion about the approaches utilized to develop the proposed system, while “[Proposed System](#)” describes the proposed system architecture in detail. The experimental setup and results are given in “[Experimental Setup and Results](#)” of the paper. “[Discussion and Comparative Analysis](#)” of the paper contains the discussion and comparative analysis, whereas “[Conclusion and Future work](#)” concludes the paper.

## Related Works

In this section, we discuss some of the recent works that are related to our proposed work. Kumar et al. [16] utilize X-vector architecture for the spoof detection task. Authors

proposed a time-delay shallow neural network (TD-SNN) with four hidden layers for the spoof classification task [17]. Also, demonstrated their model with various acoustic features extracted at front end including CQCC, LFCC, and used the focal loss function. Their work concluded that TD-SNN-based Spoof Detection System (SDS) outperforms all the GMM-based SDS in the case of PA track of ASV Spoof 2019 dataset, whereas for LA track, GMM-based SDS performs well for some cases. Ray et al. [18] propose a two-stage hybrid model to classify audio input as spoofed or bonafide. Researchers have employed genuinization transformer in the first stage that makes features to be used by the classifier more discriminable. In the second stage Residual Squeeze and Excitation networks are used for classification. Experiments conducted with various ResNet models and concluded that ResSENet-101 demonstrates comparatively better performance for the given task. Zheng et al. [19] employs a densely connected convolutional network at back end with various features extraction techniques such as MFCC, LFCC, CQCC at front end. Their work make use of fusion module in which output of previous single modules are together fused to obtain final fusion model. Their results conclude that their proposed method outperforms available state-of-the-art alternatives for LA track of ASV Spoof 2019 dataset.

Mittal et al. used CQCC with Convolutional Neural Network (CNN), CQCC with LSTM and static-dynamic feature combination of CQCC with LSTM-CNN ensemble in their proposed works of [20, 21] and [8], respectively. However, the issue of noise remains open with MFCC and CQCC features.

The authors in [22] introduced Audio Deep Synthesis Detection (ADD) challenge that has been divided into three sections: a false audio game (Track 1), a low-quality fake audio detection (Track 2), and a partially fake audio detection (Track 3). The primary theme of the presented research is to handle Track 1 and Track 2. To extract features at the frontend, LFCC features were used. In the backend, two Machine Learning (ML) models, Cross-Lingual Speech (XLR-S), and the Emphasized Channel Attention, Propagation, and Aggregation-Time Delay Neural Network (ECAP-TDNN) were used. These backend models were trained using audio data from Track 1, and are trained models were used to categorise audio files from Track 2. The results show that the LFCC-XLR-S model outperformed the other proposed model.

In [23], authors proposed fused features comprising of 14 dimensional MFCC, GTCC with 1 dimensional spectral centroid and flux extracted at front end with BiLSTM [24] model for classification at back end. System is evaluated on ASV Spoof 2019 dataset with EER as performance metrics. Authors claim to have achieved 6.52% lower EER than baseline models. In [25] authors proposed two phase approach

in which firstly speaker specific spoof print model is created using the ResNet18-L-FM model and after enrolment of the speaker cosine-based similarity score is computed during testing for classification. Authors make use of large margin cosine loss function. Additionally, x-vectors-based speaker embeddings are utilized in combination with spoof embeddings thus suggesting the importance of x-vectors in spoofing detection.

From the literature, it is clear that while designing a system model at the backend emphasis is given to capturing discriminative features at the utterance level of the given audio for further effective binary classification. At the front-end, combination of various cepstral coefficients aids further compared to individual features or filter bank energies for the task at hand. X-vectors demonstrate [13] this ability to capture speaker as well as channel characteristics well. These reasons were enough to motivate us to further build the countermeasure system upon x-vectors at the back end along with the use of hybrid features at the front end. Our contribution to this paper is:

1. The proposed work focuses on spoofed attack detection in clean and noisy conditions.
2. The work implements a system that uses integrated GTCC and MFCC at the front end to learn speaker characteristics such that the proposed system can discriminate between genuine and faked speech.
3. This work introduces a new backend model that is TDNN with X-vector layer for classification.
4. The performance of proposed system evaluated in clean and noisy environment by adding Babble noise to testing data at different signal to noise ratio (SNR) values of 0 dB, 5 dB and 10 dB.
5. The work evaluates the system on LA track of ASV Spoof 2019 dataset with evaluation metrics such as Equal Error Rate (EER) and tandem distance cost function (t-DCF). The work evaluates the proposed systems against unseen LA attacks, also.

### Preliminaries

The section discusses some of the preliminaries concerned with proposed system:

#### Acoustic Front-End Model

The aim of Acoustic front-end model is to extract information out of spoken utterance. The input of front-end techniques is a frame of signal usually a size of 20–25 ms. Proposed work uses two feature extraction algorithm such as traditional feature MFCC and noise robust feature GTCC.

- Mel frequency cepstral coefficients

Figure 1 shows the steps involved in extraction of MFCC. MFCC is an audio feature extraction method that deemphasizes all other information while extracting speech characteristics that are similar to those utilised by humans to understand speech. A time frame is first created from an arbitrary number of samples using the spoken signal. In the majority of systems, frame overlapping is utilised to make the transition between frames seamless. Then, using a Hamming window, each time frame is windowed to remove discontinuities at the boundaries. Perhaps, the three most crucial steps in the extraction of MFCCs are to perform a discrete Fourier transform (DFT) to extract frequency domain information, convert a linear power spectrum to a Mel-scale power spectrum using overlapping triangular bandpass filters, and then convert the signal back to time domain using an inverse DFT (IDFT) [26]. The Mel frequency  $mel(f)$  is expressed in term of frequency  $f$  in hertz is given by Eq. (1).

$$mel(f) = 1127 * \ln 1 + \frac{f}{700} \tag{1}$$

- Gammatone cepstral coefficients

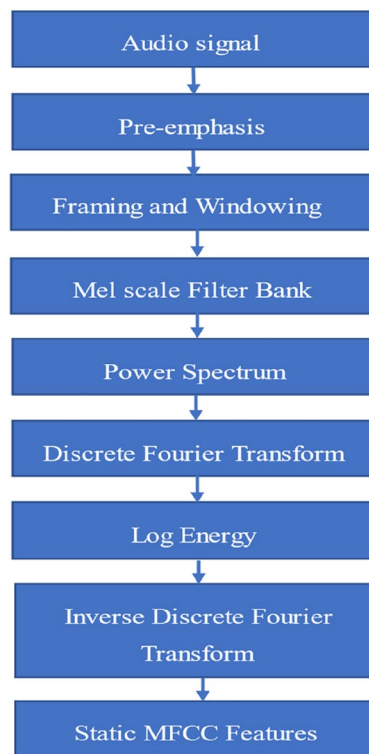


Fig. 1 Steps involved in extraction of MFCC

The Gammatone cepstral coefficients extraction scheme is analogous to the calculating MFCC process for the proposed shown in Fig. 1. The audio signal is first windowed into brief frames, typically lasting between 10 and 50 ms. There are two goals for this approach. The spectro-temporal signal analysis is made easier in two ways: first, the non-stationary audio signal can be assumed to be stationary for such a short period; second, the feature extraction method is made more effective [27]. After that, the fast Fourier transform (FFT) of the signal is applied to the GT filter bank, which emphasises the perceptually significant sound signal frequencies. This filter bank is made up of the frequency responses of several GT filters. In fact, the design of the GT filter bank is the focus of this work, which takes into account factors like the total filter bank bandwidth, the order of the GT filters, the ERB model, and the number of filters  $N$ . Finally, to replicate the human loudness perception and decorrelate the outputs of the logarithmic-compressed filter, the discrete cosine transform (DCT) is used. This results in better energy compaction. The total cost of the computation is very similar to the MFCC computation [28].

$$GTCC = \sqrt{\frac{2}{N}} \sum_{n=1}^N \log(E_n) \cos\left[\frac{\pi n}{N}\left(g - \frac{1}{2}\right)\right] \text{ where } 1 \leq g \leq G, \quad (2)$$

where  $E_n$  represents energy of signal in  $n$ th spectral bandwidth and  $G$  shows the number of GTCC.

### Acoustic Back-End Model

An overview of all the models utilised to implement the system's back end is provided in the present section.

- **X-Vector:** In general, x-vector architectures utilizes a deep neural network (DNN) to extract fixed size embeddings from variable size acoustic segments. The DNN employs few TDNN layers that enhances frame-level representation by varying context across hidden layers. A statistics pooling layer aggregates all frame-level representations computing mean and standard deviation vectors once for each input segment. Subsequent layers after pooling usually are fully connected layers that deals with segment-level representation. Finally, a SoftMax layer outputs the posterior probabilities for each individual speaker. This trained DNN is then used to extract fixed dimensional x-vector during testing. These x-vector are then compared with data embeddings from training phase usually by some scoring approach. X-vectors are highly scalable with training data in terms of performance, can handle short duration utterances comparatively well, efficiently captures speaker as well as channel characteristics. Thus, making them suitable for audio spoof detection task.

- **Squeeze- Excitation (SE) Res2Block:** SE-Res2Block is the combination of dilated Res2Net Block [29] with preceding and succeeding dense convoluted layer of context one followed by an SE block. Squeeze-Excitation blocks are computer vision approach that is successfully proven to model feature interdependencies across frames (global channels). In SE Block Squeeze operation simply includes calculating the mean vector  $z$  of features at frame-level across time domain using Eq. (3). \*

$$z = \frac{1}{T} \sum_t h_t \quad (3)$$

Then, the excitation operation calculates a vector  $s$  containing weights between 0 and 1 that is used to rescale each frame (channel):

$$s = \sigma(W_2 f(W_1 z + b_1) + b_2), \quad (4)$$

where  $\sigma(\cdot)$  denotes the sigmoid function  $f(\cdot)$  a Rectified Linear Unit (ReLU) [27]-based non-linearity,  $W_1 \in \mathbb{R}^{R \times F}$  and  $W_2 \in \mathbb{R}^{F \times R}$ . This operation provides a bottleneck in which  $F$  refers to number of frames (channels) and  $R$  refers to reduced dimension.

Finally, a frame-wise multiplication of resulting  $s$  vector is done with original input:

$$\hat{h} = s.h \quad (5)$$

- **LSTM-based Aggregator:** This work makes use of recurrent aggregation using long short-term memory (LSTM). This aids in retrieving more expressive features. Mapping nodes are used to make the dimensions similar before inputting to LSTM-based aggregator. Features generated at lower layers to higher layers of TDNN with varied context forms a sequence with different degree of granularity of information. Recurrent aggregation explores these relationships across layers of TDNN in order. This aids in preserving context with spatial and channel wise information. Though receptive fields of different convolutional layers in TDNN varies but the feature map sizes produced is similar thus no mapping nodes are required to transform into vectors of same dimension before input to LSTM. LSTM employs a memory mechanism that is capable of retaining temporal information of all the inputs together while detecting correlation among all sequence of input. LSTM consists of input gate, output gate and forget gate that regulates the flow of information to realize long short-term memory. This naturally introduces attention mechanism that is context depend-

ent. The input to LSTM-based aggregator is a matrix  $V \in RN \times C$  where  $N$  is number convolutional block and  $C$  is the output features with same dimensions of convolutional blocks. The output vector of LSTM  $U$  is calculated as:

$$U = FLSTM(V), \tag{6}$$

where FLSTM denotes standard LSTM structure with forget gates. The input is connected to convolutional blocks through mapping nodes. Gradient is back propagated directly to all blocks. This aids in extracting more expressive sequential features. The output vector  $U$  is passed through ReLu [27] non-linear function to utilize the obtained abstract features.

$$u = \sigma(WU). \tag{7}$$

- Additive Angular Margin SoftMax (AM-SoftMax): AM-SoftMax used to obtain highly discriminative feature of audio for speaker verification. It is used to improve within class variation by putting focus on target labels. AM-SoftMax loss can be written as follows as a classification loss for training speaker discriminative DNNs:

$$L_{AAM} = -\frac{1}{B} \sum_{i=1}^B \log \frac{e^{s_f(\cos(\theta_i) - A_m)}}{e^{s_f(\cos(\theta_{i,i}) + \sum_{v \neq i} e^{s_f(\cos(\theta_{v,i}))})}, \tag{8}$$

where  $s_f$  is scaling factor for training stability,  $l_i$  label of  $i$ th sample,  $B$  denotes the batch size and  $A_m$  additive margin.  $\theta_{v,i}$  is angle between vector of  $i$ th sample  $s_i$  and  $w_v$  that represents vector of class  $v$

$$\cos \theta_{v,i} = \frac{s_i^T w_v}{\|s_i\| \|w_v\|}. \tag{9}$$

During implementation,  $s_i$  vector of  $i$ th sample can be read from the activation of the fully connected layer and weight associated with  $j$ th output unit can be denoted as  $w_v$ . Cosine similarity is calculated using Eq. (9) and is used to compare two speech features.

### Proposed System

The architecture of proposed model described in this section. Figure 2 shows the detailed view of the system. As described earlier ASV system as two important phases. The first phase uses feature extraction algorithm to extract information from

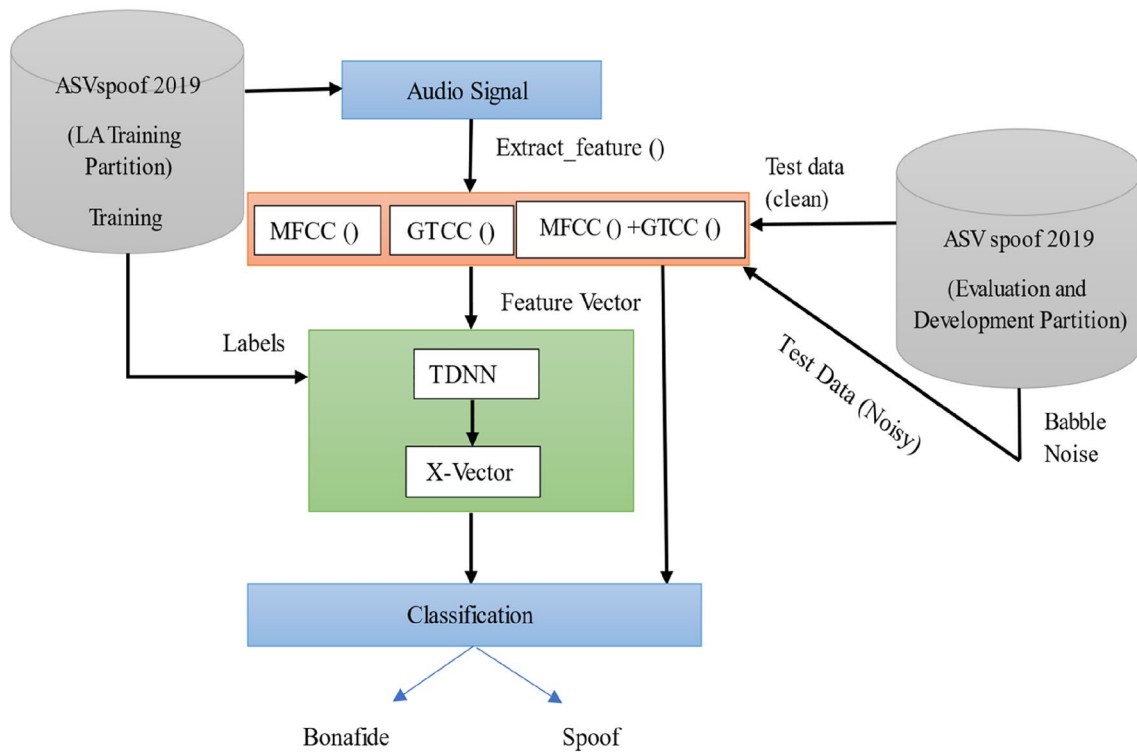


Fig. 2 Proposed architecture

audio, while the second uses acoustic models to classify the audio. The feature extraction section displays three different types of features employed in the proposed work: MFCC produces 39 features, GTCC produces 36 features, and the augmented MFCC-GTCC produces a feature vector of size 75D. In our model, LA partition of ASV spoof 2019 has been used for training the model after first phase. For testing the performance of the model evaluation and development partition of the dataset is used. During implementation of the model, we have done two experiments. In first experiment performance of proposed model evaluated in clean environment and in the second experiment model's performance evaluated in the noisy environment. In the last phase, binary classification is done.

### Acoustic Front-End Model

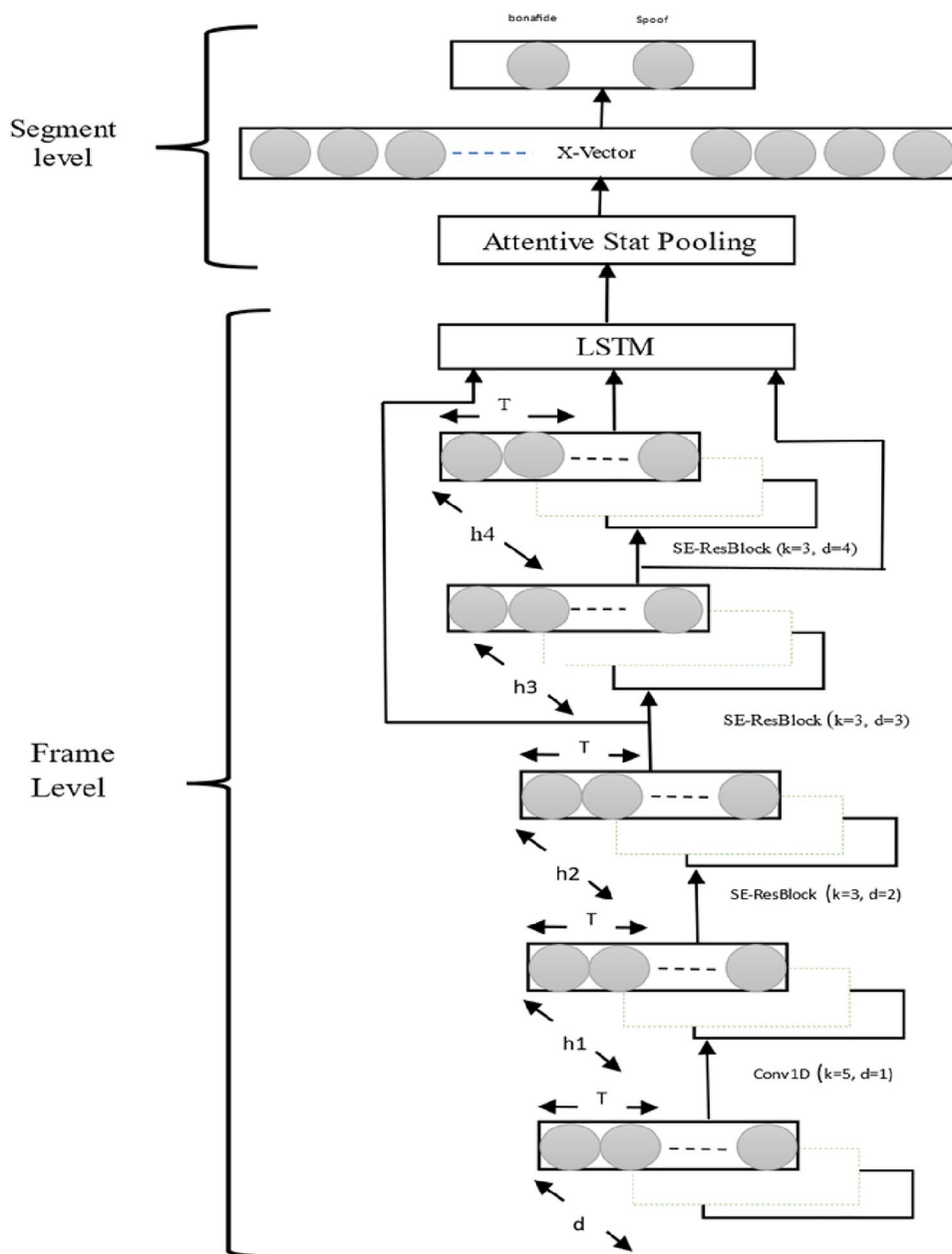
The feature extraction process plays a vital role in ASV systems. This section describes the algorithms used for the feature extraction. Before starting the process of feature

extraction, some pre-processing has been done by setting frame size to 25 ms, frame overlapping set to 50%, and hamming window size set to 30 ms. After pre-processing, feature extraction process has been done.

Function 1 `extract_features()` shows the implementation of front end of the proposed system. Function takes audio as input. In line 2, `MGCC()` used to extract mfcc features for the respective audio. In line 4, an `audioread()` function has been applied, with audio serving as its input. This function returns a frame  $y$  whose size is determined by the sampling rate  $s_r$ . In the next line, the built-in `MFCC()` function of MATLAB has been used to extract MFCC features. `MFCC()`, which accepts  $y$  and  $s_r$  as input and returns coefficients, the double delta and delta features, which display the change in coefficients from one data frame to the next. In line 7, similar process followed to extract gtcc features. At end, both the features combined sequentially to create a new feature vector which contains both kinds of features.

<b>Function1:</b> <code>extract_features()</code>
<i>Input: audio file in .flac format</i>
<ol style="list-style-type: none"> <li>1. <i>Begin:</i></li> <li>2. <code>extract_MGCC(audio_file)</code></li> <li>3. {</li> <li>4. <code>y, s<sub>r</sub> = audioread(audio_file);</code></li> <li>5. <code>coeffs1, delta1, deltadelta1 = mfcc(y, s<sub>r</sub>) // MFCC feature vector</code></li> <li>6. <code>X = return coeffs1, delta1, deltadelta1</code></li> <li>7. <code>coeffs2, delta2, deltadelta2 = gtcc(y, s<sub>r</sub>) // GTCC feature vector</code></li> <li>8. <code>Y = return coeffs2, delta2, deltadelta2</code></li> <li>9. <code>Z = horzcat(X, Y) // MFCC and GTCC sequentially concatenated features.</code></li> <li>10. }</li> <li>11. <i>End</i></li> </ol>
<i>Output: returning MGCC 75D feature matrix</i>

**Fig. 3** Architecture of proposed backend model



**Acoustic Backend Model**

For binary classification, TDNN with X-Vector have been used as shown in proposed architecture in Fig. 3. In Figure  $T, d, h$  represents number of frames, input feature vector dimension and size of hidden layer. TDNN has two layers: Frame level layer and Segment level layer.

The feature map of the most recent frame-layer is the only one used by the original x-vector system to calculate the pooled statistics. These deeper level features are the most complex ones and ought to have a substantial

correlation with the speaker identities given the hierarchical structure of a TDNN. Our suggested technique concatenates all of the SE-Res2Blocks' output and initial CNN feature maps for each frame. Following this, the concatenated data are processed by an aggregate LSTM layer to produce the features for the attentive statistics pooling.

At segment layer, firstly attentive statistics pooling has been used which takes output from the aggregate LSTM layer. Using this layer, different weights are assigned to different frame to capture long term variation in speaker characteristic. The last layer of proposed backend

**Table 1** LA partition of ASVspoo2019 Database

Samples	Logical Access	
	Genuine Samples	Spoof Samples
Training	2580	22,800
Development	2548	22,296
Evaluation	25,445	38,378

architecture is X-vector and at end audio classified as bonafide of spoofed according to speaker embeddings.

## Experimental Setup and Results

The experimental setup details of proposed ASV system, and details on obtained results given in this section. The implementation of frontend has been carried out using MATLAB R2021, and a windows 11 operating system with an intel core i5 processor. The inbuilt function mfcc(), gtcc() have been used to extract frontend features from audio. Speech brain AI tool kit has been used to implement backend of the proposed model. To analyse the performance of the proposed model for speaker verification, the model is trained and evaluated using Logical Access (LA) partition of ASV spoof 2019. Subpart of LA partition are training set, evaluation set and development set. In the proposed model, training set is used for training, development set used to optimise speaker countermeasure and evaluation model is used to evaluate the performance of the proposed model.

## Dataset

The suggested system has been trained, developed, and evaluated using the ASVspoo2019 LA dataset. To defend against attacks like Speech Synthesis (SS), VC, and replay, this dataset was developed. The dataset's LA portion includes TTS and

VC fabricated speech. Speeches are recorded by 8 male and 12 female. The collection contains samples that are 2–8 s long and recorded in English. However, there are only a maximum of 4–6 s of audio in each set. Table 1 shows the specifics of the training and evaluation samples for each type of attack in the used dataset.

## Evaluation Metric

To measure the performance of the proposed model, evaluation metrics such as Equal Error Rate (%) and tandem detection cost function (t-DCF) has been used.

- ASVspoo2019 proposed a metrics to measure the performance of their baseline. In our work we are using t-DCF to measure and compare the performance of proposed model with the baseline model. The t-DCF function is defined in the following Eq. (10).

$$t - DCF(s) = C_1 P_m(s) + C_2 P_f(s), \quad (10)$$

where  $P_m(s)$  and  $P_f(s)$  are miss rate and false alarm at threshold  $s$ ,  $C_1$  and  $C_2$  are constants used to represent t-DCF cost.  $P_m(s)$  and  $P_f(s)$  are calculated as follows.

$$P_m(s) = \frac{\text{score of bonafide trails} \leq s}{\text{Total bonafide trails}} \quad (11)$$

$$P_f(s) = \frac{\text{score of spoof trails} \leq s}{\text{Total bonafide trails}} \quad (12)$$

- Equal Error Rate (EER) is a threshold  $s_{EER}$  point where miss rate and false alarm rate are equal to each other. The mathematical expression for EER is given in Eq. (13) below:

**Table 2** Performance analysis of proposed model over evaluation dataset

Model	EER (%)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Min t-DCF
MFCC	4.9	94	94	94.4	94	0.0637
GTCC	5.49	91.5	92.9	94.4	91.5	0.0774
MFCC + GTCC	1.13	98	98.5	98.7	98	0.0326

**Table 3** Performance analysis of proposed model over development dataset

Model	EER (%)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Min t-DCF
MFCC	4.3	95.5	95.5	92.9	95.4	0.0574
GTCC	5.34	98.9	98.3	95.5	98.1	0.0744
MFCC + GTCC	0.03	99.3	99	99.5	99	0.0012



$$EER = P_f(s_{EER}) = P_m(s_{EER}). \tag{13}$$

- Accuracy (%), Recall (%), Precision (%), and F1-score (%) as described by Eqs. (14–17), have been used to assess the performance of the proposed ASV system.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{14}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

$$F1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{17}$$

where TP: true positive, TN: true negative, FP : false positive, FN: false negative.

### Experiment 1: Performance Analysis of Proposed Model in Clean Environment

Tables 2 and 3 shows the results of the proposed model that uses MFCC, GTCC and sequentially combined MFCC-GTCC features. The performance of the model is evaluated over both development dataset and evaluation dataset. From Table 2, it can be observed that using evaluation set

Fig. 4 Process of generating Noisy testing data

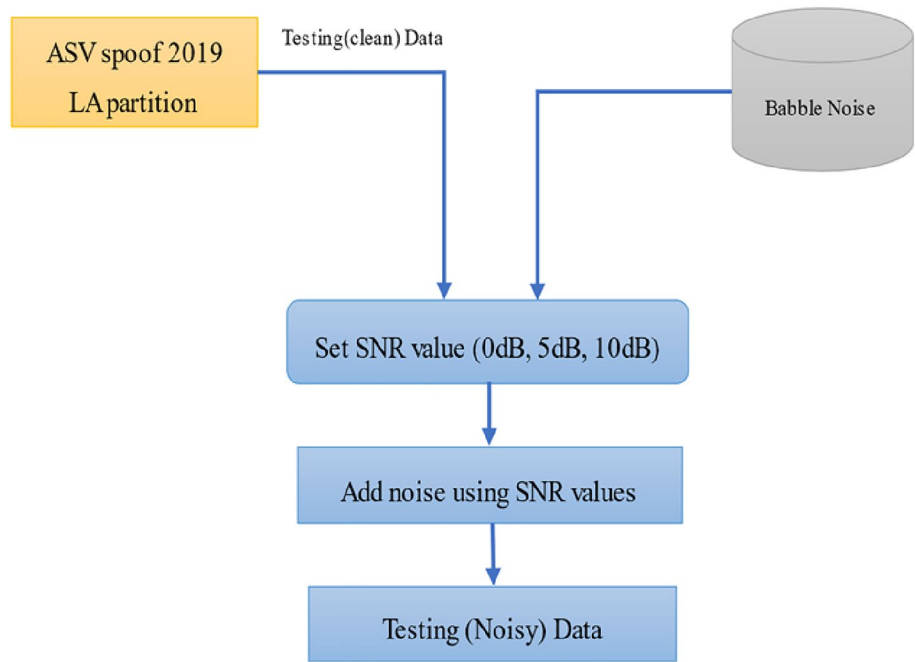


Table 4 Performance analysis of proposed model over evaluation dataset in noisy environment

Model	SNR (dB)	EER (%)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Min t-DCF
MFCC	0	8.37	92.8	92.5	92	92.2	0.0057
	5	9.78	90.5	90	90	91	0.0088
	10	10.9	87	88	87	87	0.0097
GTCC	0	7	92	92	92	92	0.0036
	5	8.5	91.5	91	91	91.2	0.0054
	10	9.4	96	98	97	97	0.0074
MFCC + GTCC	0	6.9	93	93	93.5	93	0.0012
	5	8.7	90	90	90.7	90.3	0.0034
	10	9.2	88	88	88.2	88	0.0056

**Table 5** Performance analysis of proposed model over development dataset in noisy environment

Model	SNR (dB)	EER (%)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Min t-DCF
MFCC	0	7.78	87	88	87	87	0.0047
	5	9.78	85	85	78	83	0.0067
	10	10.9	82	82	82.5	82	0.0087
GTCC	0	7.5	92	92	92	92	0.0026
	5	9.5	91.5	91	91	91.2	0.0034
	10	10.8	96	98	97	97	0.0054
MFCC+GTCC	0	6.3	93	93	93.5	93	0.0019
	5	9.2	90	90	90.7	90.3	0.0028
	10	9.1	88	88	88.2	88	0.0021

our first model achieved EER 4.9% and 0.0637-Min t-DCF, second model that is GTCC has achieved EER 5.49% and 0.0774-Min t-DCF. The last model that is MFCC-GTCC has achieved EER 1.13% and -Min t-DCF 0.0326. From the result, it can be concluded that over evaluation dataset model 3 (MFCC-GTCC) outperformed other models.

From Table 3, it can be observed that using development set our first model (MFCC) achieved EER 4.3% and 0.0547-Min t-DCF, second model that is GTCC has achieved EER 5.34% and 0.0744-Min t-DCF. The last model that is MFCC-GTCC has achieved EER 0.03% and Min t-DCF 0.0012. From the result, it can be concluded that over evaluation dataset model 3 (MFCC-GTCC) outperformed other models. Now after comparing both the results, it can be concluded that our proposed model worked best over development dataset.

### Experiment 2: Performance Analysis of Proposed Model Under Noisy Environment

To analyse Proposed model in a noisy setting, the work simply uses babbling noise and an audio sample of both the dataset i.e., development and evaluation dataset as shown in Fig. 4. The work employs babbling noise to development and evaluation dataset at three distinct Signal to Noise Ratios (SNR) of 0 dB, 5 dB and 10 dB. It is vital to notice that the noise is only applied to the testing dataset. However,

training has been done on clean data. In a noisy setting, the performance of all the three model have been checked. The performance of all the model given in Tables 4 and 5.

From Table 4, it can be observed that augmented MFCC-GTCC outperformed other models in noisy environment over evaluation dataset. Table 5 shows the models performance in noisy environment over development dataset. From the results, it can be observed that augmented MFCC+GTCC outperformed other models in noisy condition also.

### Experiment 3: Performance Analysis of Proposed Model Under Unseen Attacks

This experiment is intended to assess the proposed system's effectiveness against unseen LA attacks, namely A07, A08, A09, A10, A17, A18, and A19 [30]. The proposed spoofing system utilises the samples of the LA dataset, which contains 63,895 examples of unseen assaults. We trained the model using genuine and spoof data from the LA training set, and evaluated it using genuine and spoof samples from unseen attacks. We conducted studies to identify unseen voice spoofing assaults created by sophisticated spoofing algorithms such as A07, A08, A09, A17, A18 and A19. The experimental results are shown in Table 6, which demonstrate that the proposed system is also capable in handling unseen attacks, although the value of the parameters degrade in comparison to known attacks.

**Table 6** Performance analysis of proposed model under unseen/novel attacks

Attacks	Feature set	EER (%)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Min t-DCF
A07	MFCC+GTCC	4.01	90	92	93	92	0.0012
A08	MFCC+GTCC	6	95	93	92	93	0.0015
A09	MFCC+GTCC	4.7	94	94	94.9	94.4	0.0471
A10	MFCC+GTCC	5.79	89	88	87	89	0.0643
A17	MFCC+GTCC	12	96.1	96	96	95.8	0.1852
A18	MFCC+GTCC	13	96	95	94	94.9	0.2576
A19	MFCC+GTCC	23	95	78	75	76	0.3042

**Table 7** Comparison with existing methods for spoofing detection

Work	Front-end	Back-end	Development Set		Evaluation Set	
			EER (%)	min t-DCF	EER (%)	min t-DCF
Wang et al. [30]	CQCC-GMM	X-Vector	0.43	0.01283	9.57	0.2366
	LFCC-GMM		2.71	0.0663	8.09	0.2116
Weng et al. [29]	LFCC-CQT (Fusion)	SE-Res2Net50	–	–	1.892	0.0452
Desolanques et al. [31]	MFCC	ECPA-TDNN	0.87	0.1066	1.12	0.1318
Dua et al. [32]	Mel spectrogram-CQCC	CNN-LSTM (Under Noise 0 dB)	–	–	13.5	–
		CNN-LSTM (Under Noise 5 dB)	–	–	14.5	–
Final Proposed Model	MFCC + GTCC (Under Noisy condition)	TDNN	0.03	0.0012	1.13	0.0032
		TDNN (Under Noise 0 dB)	4.29	0.008	4.65	0.0094
		TDNN (Under Noise 5 dB)	4.39	0.0072	5.0	0.0065
		TDNN (Under Noise 10 dB)	5.19	0.0071	5.23	0.0097

## Discussion and Comparative Analysis

The aim of the proposed model is to enhance the performance of automatic speaker verification system in case of Spoofing attacks. In our proposed work, we have implemented a model which is having two parts: front end and backend. For the implementation of front-end MFCC and GTCC features are integrated sequentially. For backend implementation, T-DNN with X-Vector have been used. Tables 2 and 3 show the result of the proposed model in clean condition. Performance of the proposed model evaluated over development dataset and evaluation dataset. It can be concluded from Table 3 that system performed well over development set. The performance of proposed model is also evaluated in noisy environment also by adding babble noise to both development dataset and evaluation dataset. Tables 4 and 5 show the result of the proposed system under noisy condition. It can be concluded from Table 5 that at SNR value 0 system is performing well for development dataset. The proposed model performed well to detect unseen attacks also.

Many researchers have done work in this area and still working to improve the performance of the ASV model against different kinds of attacks. Table 7 shows some existing work done in this field and compare its performance with the suggested approach. In traditional model at front end, mostly MFCC and LFCC have been used. As the research going on new features have been introduced in this area. Following are some works given by authors. Table 6 compares the existing work to proposed study on same dataset, except [31] by Desplanques et al. that uses 2019 VoxCeleb and [32] used VSDC dataset in their work. Based on the comparative analysis, it can be concluded that proposed study outperformed other existing methodologies.

## Conclusion and Future Work

Voice spoofing detection using ASV system is an active area of research. The performance of ASV system plays critical role to detect the difference between spoofed or bonafide audios. Therefore, to optimize the performance of proposed ASV system, a new backend model has been introduced for speaker verification based on TDNN with X-vector. From the results, it can be concluded that proposed backend resulted in significant relative improvements over classification models. Two experiments have been carried out to check the performance of proposed system in clean and noisy environment. In both the conditions, proposed system outperformed and achieved performance improvement of about 59.7% and 65.9% relative to earlier proposed works. In future, this work can be extended by introducing new optimized feature extraction methods at front end and light weight classification model at backend. Also, the proposed system can be used to handle other type of attacks such as replay and deepfake attacks.

**Funding** I, Dr. Mohit Dua, on the behalf of all the authors declare that this study did not receive any funding from any resource.

**Data availability** All the data generated or analyzed during this study are included and referred to in this published article.

## Declarations

**Conflict of Interest** The authors declare that submitted manuscript have no conflict of interest.

**Ethical Approval** This research article does not contain any studies with human participants or animals performed by any of the authors.

**Human and Animal Rights** This article does not contain any studies with human participants or animals performed by any of the authors.

**Informed Consent** Not applicable.

## References

- Tak H, Todisco M, Wang X, Jung J, Yamagishi J, Evans N. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. 2022. arXiv Prepr. arXiv2202.12233
- Wu Z, Evans N, Kinnunen T, Yamagishi J, Alegre F, Li H. Spoofing and countermeasures for speaker verification: a survey. *Speech Commun.* 2015;66:130–53.
- Wu Z, et al. ASVspoof: the automatic speaker verification spoofing and countermeasures challenge. *IEEE J Sel Top Signal Process.* 2017;11(4):588–604. <https://doi.org/10.1109/JSTSP.2017.2671435>.
- Yamagishi J et al. Asvspoof 2019: the 3rd automatic speaker verification spoofing and countermeasures challenge database. 2019.
- Wu Z, Gao S, Cling ES, Li H. A study on replay attack and anti-spoofing for text-dependent speaker verification. *Signal Inf Process Assoc Annu Summit Conf (APSIPA) Asia-Pac.* 2014. <https://doi.org/10.1109/APSIPA.2014.7041636>.
- Hossan MA, Memon S, Gregory MA. A novel approach for MFCC feature extraction. *Int Conf Signal Process Commun Syst.* 2010. <https://doi.org/10.1109/ICSPCS.2010.5709752>.
- Dave N. Feature extraction methods LPC, PLP and MFCC in speech recognition. *Int J Adv Res Eng Technol.* 2013;1:2320–6802.
- Todisco M, Delgado H, Evans N. Constant Q Cepstral coefficients: a spoofing countermeasure for automatic speaker verification. *Comput Speech Lang.* 2017. <https://doi.org/10.1016/j.csl.2017.01.001>.
- Todisco M, Delgado H, Evans NWD. A new feature for automatic speaker verification anti-spoofing: constant Q cepstral coefficients. *Odyssey.* 2016;2016:283–90.
- Valero X, Alías F. Gammatone cepstral coefficients: biologically inspired features for non-speech audio classification. *Multimed IEEE Trans.* 2012;14:1684–9. <https://doi.org/10.1109/TMM.2012.2199972>.
- Ge W, Tak H, Todisco M, Evans N. On the potential of jointly-optimised solutions to spoofing attack detection and automatic speaker verification. 2022. arXiv Prepr. arXiv2209.00506
- Liu H, Zhao L. A speaker verification method based on TDNN–LSTMP. *Circuits Syst Signal Process.* 2019;38(10):4840–54.
- Snyder D, Garcia-Romero D, Sell G, Povey D, Khudanpur S. X-vectors: robust dnn embeddings for speaker recognition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5329–33.
- Snyder D, Garcia-Romero D, Sell G, McCree A, Povey D, Khudanpur S. Speaker recognition for multi-speaker conversations using x-vectors. In: ICASSP 2019–2019 IEEE International conference on acoustics, speech and signal processing (ICASSP), 2019, pp. 5796–800.
- Qin Y, Du J, Wang X, Lu H. Recurrent layer aggregation using LSTM. In: 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–8.
- Kumar MG, Kumar SR, Saranya MS, Bharathi B, Murthy HA. Spoof detection using time-delay shallow neural network and feature switching. In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2019, pp. 1011–17.
- Zhang X, Zhang X, Zou X, Liu H, Sun M. Towards generating adversarial examples on combined systems of automatic speaker verification and spoofing countermeasure. *Secur Commun Netw.* 2022;2022:2666534. <https://doi.org/10.1155/2022/2666534>.
- Ray R, et al. Feature genuinization based residual squeeze-and-excitation for audio anti-spoofing in sound AI. *Int Conf Comput Commun Netw Technol (ICCCNT).* 2021. <https://doi.org/10.1109/ICCCNT51525.2021.9580127>.
- Wang Z, Cui S, Kang X, Sun W, Li Z. Densely connected convolutional network for audio spoofing detection. In: 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2020, pp. 1352–60.
- Mittal A, Dua M. Automatic speaker verification system using three dimensional static and contextual variation-based features with two dimensional convolutional neural network. *Int J Swarm Intell.* 2021;6(2):143–53.
- Mittal A, Dua M. 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, 2021, pp. 895–904.
- Lv Z, Zhang S, Tang K, Hu P. Fake audio detection based on unsupervised pretraining models. In: ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 9231–5.
- Hassan F, Javed A. Voice spoofing countermeasure for synthetic speech detection. In: 2021 International Conference on Artificial Intelligence (ICAI), 2021, pp. 209–12.
- Rupesh Kumar S, Bharathi B. Generative and discriminative modelling of linear energy sub-bands for spoof detection in speaker verification systems. *Circuits Syst Signal Process.* 2022;41(7):3811–31.
- Chen T, Kumar A, Nagarsheth P, Sivaraman G, Khoury E. Generalization of audio deepfake detection. In: Proceedings of Odyssey 2020 The Speaker and Language Recognition Workshop, 2020, pp. 132–7.
- Barai B, Basu S, Nasipuri M, Das D, Das N. VQ/GMM based speaker identification with emphasis on language dependency. 2018.
- Fu Z, Lu G, Ting KM, Zhang D. A survey of audio-based music classification and annotation. *IEEE Trans Multimed.* 2010;13(2):303–19.
- Cheng O, Abdulla W, Salcic Z. Performance evaluation of front-end algorithms for robust speech recognition. *Proc Eighth Int Symp Signal Process Appl.* 2005;2:711–4. <https://doi.org/10.1109/ISSPA.2005.1581037>.
- Li et al. X. Replay and synthetic speech detection with res2net architecture. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 6354–8.
- Wang X, et al. ASVspoof 2019: a large-scale public database of synthesized, converted and replayed speech. *Comput Speech Lang.* 2020;64:101114. <https://doi.org/10.1016/j.csl.2020.101114>.
- Desplanques B, Thienpondt J, Demuynck K. Ecapa-tdnn: emphasized channel attention, propagation and aggregation in tdnn based speaker verification. 2020. arXiv Prepr. arXiv2005.07143
- Dua M, Sadhu A, Jindal A, Mehta R. A hybrid noise robust model for multireplay attack detection in automatic speaker verification systems. *Biomed Signal Process Control.* 2022;74:103517. <https://doi.org/10.1016/j.bspc.2022.103517>.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.