Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Towards robust audio spoofing detection: a detailed comparison of traditional and learned features

BALAMURALI B T[1], KIN WAH EDWARD LIN[1], SIMON LUI[2], (Member, IEEE), JER-MING CHEN[3], and DORIEN HERREMANS[1,4], (Senior Member, IEEE)

[1]Information Systems, Technology, and Design Pillar, Singapore University of Technology and Design, Singapore
[2]Tencent Music Entertainment, Shenzhen, China
[3]Science Pillar, Singapore University of Technology and Design, Singapore
[4]Institute of High Performance Computing, A*STAR, Singapore

Corresponding author: Balamurali B T (e-mail: balamurali_bt@sutd.edu.sg).

This research is supported by the Education Research Funding Programme, National Institute of Education (NIE), Nanyang Technological University, Singapore, under grant number AFD 05/15 SL. The views expressed in this paper are the authors' and do not necessarily represent the views of the host institution. Further, this work is also partly supported by SUTD SRG ISTD 2017 129.

ABSTRACT Automatic speaker verification, like every other biometric system, is vulnerable to spoofing attacks. Using only a few minutes of recorded voice of a genuine client of a speaker verification system, attackers can develop a variety of spoofing attacks that might trick such systems. Detecting these attacks using the audio cues present in the recordings is an important challenge. Most existing spoofing detection systems depend on knowing the used spoofing technique. With this research, we aim at overcoming this limitation, by examining robust audio features, both traditional and those learned through an autoencoder, that are generalizable to different types of replay spoofing. Furthermore, we provide a detailed account of all the steps necessary in setting up state-of-the-art audio feature detection, pre-, and postprocessing, such that the (non-audio expert) machine learning researcher can implement such systems. Finally, we evaluate the performance of our robust replay spoofing detection system with a wide variety and different combinations of both extracted and machine learned audio features on the 'out in the wild' ASVspoof 2017 dataset. This dataset contains a variety of new replay spoofing configurations. Since our focus is on examining which features will ensure robustness, we base our system on a traditional Gaussian Mixture Model-Universal Background Model (GMM-UBM). We then systematically investigate the relative contribution of each feature set. The fused models, based on both the known audio features and the machine learned features respectively, have a comparable performance with an Equal Error Rate (EER) of 12. The final best performing model, which obtains an EER of 10.8, is a hybrid system that contains both known and machine learned features, and is trained on an augmented dataset, thus revealing the importance of incorporating both types of features when developing a robust spoofing prediction model.

INDEX TERMS Audio Classification, Audio Spoofing, Autoencoders, Countermeasures, Replay Attacks, GMM-UBM.

I. INTRODUCTION

SINCE the dawn of Hacker Culture in the 50s and 60s [1], enthusiasts have challenged themselves to overcome the limitations of software systems in order to achieve clever and creative outcomes [2]. Security hackers in particular, have posed a threat by breaching defenses and exploiting the weaknesses in networks and systems. Biometric authentication systems such as automatic speaker verification (ASV) are not exempt from this threat. For instance, in audio spoofing attacks, fraudsters alter an audio recording of a voice, such that it mimics a target speaker's voice to access a system protected by speaker verification [3], [4]. In replay attacks, imposters present speech samples recorded from a genuine client to a verification system [5]–[7].

Given the recent advances in audio processing technology, it is becoming easier to synthesize speech such that it sounds like a given target speaker. These technologies can be used by security hackers to break into ASV systems [8], [9]. In

addition to synthesizing speech, one could also use voice conversion methods that enable the conversion of utterances of one speaker to make them sound as if spoken by another speaker [10]–[12]. Given these advances, it is important to investigate whether we can discriminate original speech from spoofed speech recordings, which is the problem we tackle in this research. More specifically, our research contributes to a robust system for audio spoofing detection without knowing which spoofing technique is used in an attack. This challenge gets even more complicated if the discrimination has to be done only using only the audio data, without any other meta data [13].

Existing countermeasures that try to detect specific spoofing attacks typically make use of prior knowledge about the used spoofing algorithm [14]–[17]. As a result, these countermeasures are not generalizable to varying spoofing attacks [18]. A countermeasure for audio spoofing detection typically consists of two parts (see Figure 1). A first part dealing with features extraction and pre/postprocessing of the audio signal, and a second part consisting of a model that determines whether the audio is genuine or spoofed. During the system development, the spoofing prediction accuracy of the model is often compared for different audio features, so as to reach the highest accuracy [19], [20]. We make this comparison explicit in our experiment section.

Developing spoofing prediction systems that are generalizable to a varying range of spoofing techniques, hinges on incorporating audio features that are robust, such that they require little recalibration to detect novel spoofing attacks [13], [21]. One way to achieve this robustness is by investigating the spoofing detection performance when using various audio features. This is the key contribution of this manuscript. We leverage the dataset from the Second Automatic Speaker Verification Spoofing and Countermeasures Challenge, ASVspoof 2017 [22], and investigate the performance a wide variety of audio features on different types of replay spoofing attacks. In addition using a variety of traditional audio features, we train an autoencoder and use it to augment the training dataset.

The second step in developing a spoofing countermeasure is to use the extracted audio features to train a spoofing prediction model. The models typically used in speech/speaker related applications have evolved a lot over the past 40 years. In the past, researchers often used systems based on Discrete Vector Quantization (VQ) [23]. The state-of-the-art then moved to Gaussian Mixture Model (GMM) solutions [24], and more recently into factor analysis based on i-vector frameworks [25]. In this research, we explore the effect of different audio features in a GMM system.

The ASVspoof 2017 database was designed mainly to assess the detection accuracy of replay spoofing attacks especially for 'out in the wild' conditions, meaning without knowing the exact spoofing technique configuration used in the replay attack. In order to do so, the majority of testing data included in this database originates from different, unseen configurations of the spoofing algorithms compared to the training and development set [13]. Such 'out in the wild' conditions often necessitate a new audio feature space to accommodate for different replay spoofing techniques, and typically results in inferior performance when evaluating the models. This again confirms the need for a generalized countermeasure. Such a countermeasure could be achieved either by identifying robust audio features, or by improving an automatically learned feature space of spoofed recordings through an autoencoder. In this research, we explore both approaches in depth.

The remainder of this paper is organized as follows. The developed spoofing detection system is described in Section II and III. These sections include an overview of the audio features used in this investigation and description of the automatic feature extraction process. We provide a very thorough description of, not only how to extract and learn audio features, but also which pre- and postprocessing steps are required for obtaining the best results, thus providing guidance for the audio laymen to tackle this type of spoofing challenge. In addition, this section describes the implemented autoencoder, and how it can be used to augment the dataset. Details about the experimental setup can be found in Section IV, which describes the dataset used in this investigation. The next section describes the results (Section V), in which we discuss the most robust features. Finally, Section VI contains our conclusions and final remarks.

II. A ROBUST HYBRID REPLAY SPOOFING DETECTION ARCHITECTURE

The hybrid spoofing detection system developed in this paper consists of two branches that each have a unique way of processing the input audio files (see Figure 1). Large sets of selected audio features are first extracted from the preprocessed audio files. These features are less redundant and more compact than the original audio signal. In the first arm of the algorithm, the audio features are postprocessed and passed along to the prediction module. In the second arm, however, they are first fed to an autoencoder before postprocessing. For both arms of the algorithm, a GMM-UBM model is built (see Section II-B) that can predict the authenticity of a given audio file. Finally, both model outputs are fed to a fusion model, which will calculate a hybrid estimation of authenticity.

In what follows, we describe the different modules of our spoofing detection system in more detail, followed by an in-depth account of the extracted audio features.

A. FEATURE PROCESSING

The first part of our system relates to audio feature processing and consists of a number of steps, as described below. Figure 2 zooms in on the audio processing module in arm 1, without autoencoder.

1) Audio preprocessing

The audio preprocessing module consists of two parts: pre-emphasizing and enhancing the audio file. The audio recordings present in the corpus are first pre-emphasized by ap-

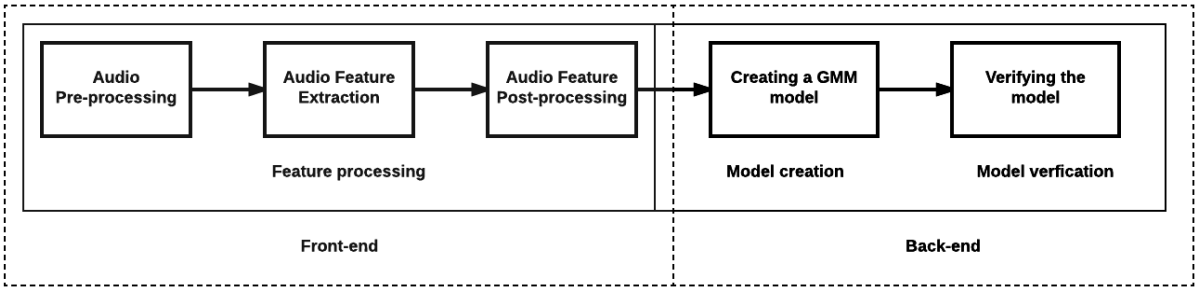

FIGURE 1. Architecture of proposed spoofing detection system.

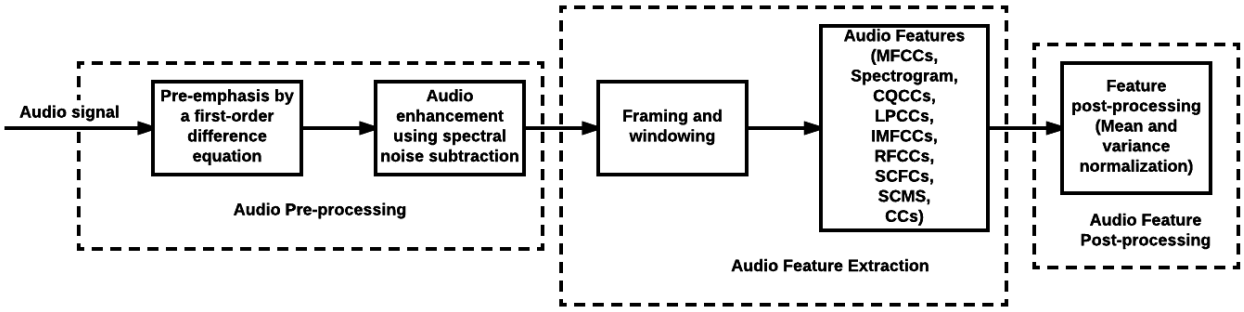

FIGURE 2. Architecture of the feature processing unit, which includes audio preprocessing, audio feature extraction, and audio feature postprocessing.

plying a first order difference equation [26]. The main goal of pre-emphasis is to boost the amount of energy present in the high frequencies. The recordings are further enhanced by removing the noise using a spectral noise subtraction method whereby an estimate of the noise spectrum is subtracted from the speech power spectrum and the negative differences are set to zero. This new power spectrum is then recombined with the original phase of the audio signal so as to reconstruct an enhanced version of the time waveform [27], [28]. This preprocessing module is implemented using the 'Voicebox tool box for speech processing' in Matlab [29].

2) Audio feature extraction

The enhanced audio signals are divided into frames, so as to form segments which are more stationary. Next, a hamming window is applied, with 50% overlap. A large number of audio features, 11 feature sets in total, are then extracted from every frame. These feature sets include mel-frequency cepstral coefficients (MFCCs), spectrogram, constant Q cepstral coefficients (CQCCs), linear predictive cepstral coefficients (LPCCs), inverted mel-frequency cepstral coefficients (IMFCCs), rectangular filter cepstral coefficients (RFCCs), linear filter cepstral coefficients (LFCCs), sub-band centroid frequency coefficients (SCFCs), sub-band centroid magnitude coefficients (SCMCs), and complex cepstral coefficients (CCs). A more detailed explanation of these features can be found in Section III.

3) Autoencoder

In the second arm of the system, we insert a pretrained autoencoder before feature postprocessing, as is displayed in Figure 3. An autoencoder is a special type of feedforward neural network with fully connected layers that is trained by matching its input to its output [30], [31]. The encoder compresses the input into a lower-dimensional space (also called 'code'), which can then be used by the decoder to reconstruct the output. After training, the output of an autoencoder will typically not be exactly the same as the original input, but it will be a closely resembling, slightly degraded version of the input [30], [31]. In addition to learning new features (i.e., the code), the autoencoder is also used to augment the dataset, as explained in Section III-C.

4) Audio feature postprocessing

In order to further reduce any distortion of the data caused by noise contamination, the original audio features (arm 1) or the encoder features (arm 2) are mean and variance normalized. This is achieved by linearly transforming the features such that they have the same statistics within the segment. It was reported that such postprocessing can significantly improve the performance of speaker/speech recognition systems [32], [33]. Figure 2 outlines the entire audio preprocessing and feature extraction process in arm 1.

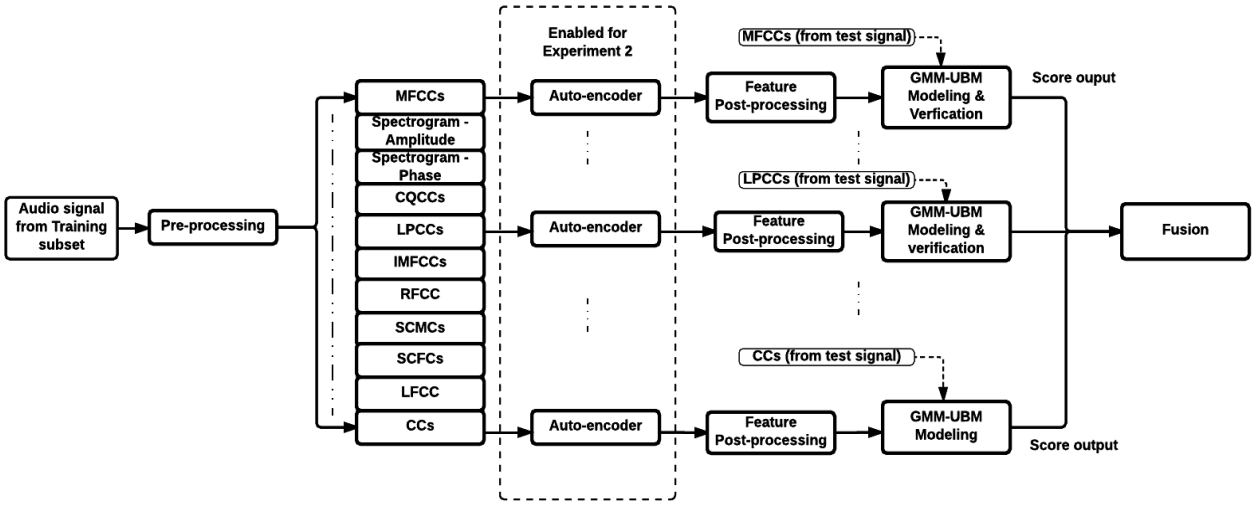

FIGURE 3. The autoencoder arm of the proposed robust hybrid spoofing detection system.

B. FEATURE MODELLING AND PREDICTION

1) GMM-UBM

The second module of the system performs the actual spoofing detection. We use a GMM-UBM for this investigation, which is created for each of the 11 feature sets in a two-stage process. Firstly, a universal background model (UBM) is created using the entire training set for a given feature set. As this pool includes both genuine and spoofed data, the resulting background model captures the feature space of both types of recordings. UBM is a Gaussian Mixture Model created using a binary splitting expectation-maximization (EM) procedure [24], [34]. In the second stage, two GMM models, one for genuine and one for spoofed recordings, will be created by adapting the UBM using maximum a posteriori (MAP) estimation to fit the respective data. Finally, in the prediction phase, the likelihood of the audio features extracted from test segments can be calculated based on each of the two models. The ratio of both of these likelihood probabilities is indicative of the predicted class.

The process of creating the initial GMM models, one for each feature set, and predicting the class for a new instance is shown in Figure 4. In a first step, the postprocessed audio features from the training set are used to create a universal background model (λ). From this λ, a genuine recording model (λ_G) and a spoofed recording model (λ_S) are created using the maximum a posteriori (MAP) adaptation procedure with genuine and spoofed audio features, respectively. These λ_G and λ_S can then be used to calculate the log-likelihood of a test audio file u (see Figure 4). For λ_G this results in $p(u/\lambda_G)$) and for λ_S the probability is $p(u/\lambda_S)$. The ratio of both of these probabilities, $\frac{p(u/\lambda_G)}{p(u/\lambda_S)}$ can be used to predict the class.

A number of parameters need to be set when creating UBM models. These include the number of Gaussian components and the number of expectation maximization (EM) iterations in the final binary split. Even though the number of Gaussian components required to model various audio features is different, we decided to set this number to 64, based on our investigation of audio samples from the development set. The number of EM iterations was set to 30.

2) Fusion model

Given that we create $22(11 + 11)$ individual GMM-UBM models, 11 for each feature set in arm 1 and 11 for each feature set after the autoencoder in arm 2, a fusion model is needed to merge the individual prediction results. We included a logistic regression fusion procedure [35], [36] to make a final prediction.

For this procedure, we determine the fusion parameters (i.e., weights for the likelihood of each separate model, and a shifting factor), using the development set. These fusion parameters are then used to combine multiple likelihoods, one corresponding to the model for each feature set, to produce a final likelihood, which will determine if the audio file is spoofed or genuine.

III. KNOWN VERSUS LEARNED AUDIO FEATURES

The extracted audio features, together with the learned encoded representation from the autoencoder are discussed below. Finally, we describe how the autoencoder was used to augment the dataset and create a new set of instances for training.

A. EXISTING AUDIO FEATURES

A total of 11 sets of known features are extracted from the dataset. Below we discuss each of these feature sets.

1) Mel-frequency cepstral coefficients (MFCCs)

The first set of features, consisting of MFCCs, focuses on the perceptually relevant aspects of the speech spectrum. They are arguably the most commonly used speech features in the speech/speaker recognition arena [37], [38]. The extraction

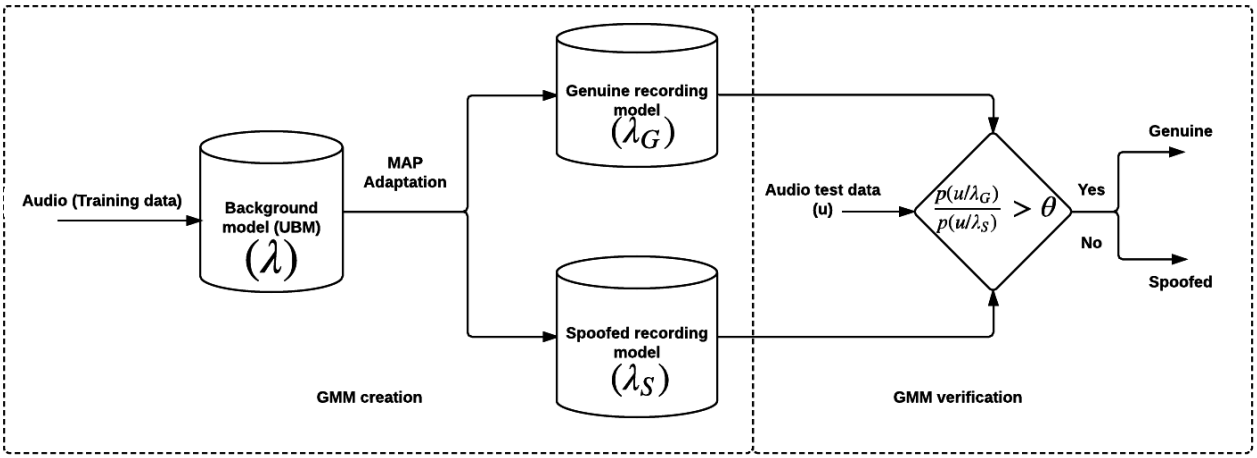

FIGURE 4. Spoofing prediction using a GMM-UBM.

process of MFCCs can be summarized as follows. First, the Discrete Fourier transform (DFT) of the input audio frame is taken to obtain the audio spectrum. This spectrum will indicate the amount of energy present in the various frequency bands. As the human hearing is very sensitive to lower frequencies, but less able to distinguish adjacent high frequency sounds, it was chosen to use a representation that accounts for this: mel-spectrum. To calculate this representation, the spectrum is warped into a mel scale, by using a bank of triangular filters (i.e., a set of overlapped non-linear mel-filter banks whose bandwidth gets narrow at low frequencies and wider at higher frequencies). Once the spectrum is warped, the logarithm of the energy present in the various regions of the speech spectrum is estimated. Finally, this log spectrum is transformed back to the time domain, thus resulting in MFCCs [39].

In the speech recognition arena, it is common to use the first 12-14 MFCCs extracted from a stationary speech frame along with their deltas (first derivatives) and delta-deltas. The first and second derivatives of the MFCCs carry information about speech dynamics. In this paper, we used 24 mel-filters and extracted 13 MFCCs together with the energy of every frame (i.e., the zeroth MFCC corresponds to energy present in the frame). By including the deltas and delta-deltas (i.e., the first and second order derivatives of MFCCs) there will be a total of 42 features per frame (i.e., 14 MFCCs, 14 deltas and 14 delta-deltas). Mathematically, MFCC feature extraction can be summarized as follows [39], [40]:

$$\mathrm{MFCC}(q) = \sum_{m=1}^{M} \log\left[\mathrm{MF}(m)\right] \cos\left\{\frac{q(m-0.5)\pi}{M}\right\} \quad (1)$$

$$\mathrm{MF}(m) = \sum_{k=1}^{K} \left|X_{\mathrm{DFT}}(k)\right|^2 H_m(k) \quad (2)$$

whereby $H_m(k)$ is the m^{th} mel-filter bank, $\mathrm{MF}(m)$ is the mel-frequency spectrum, M is the number of mel-filter banks, q the number of MFCCs, k is the DFT index and K is the total number of DFT indices, X_{DFT} is the DFT of input audio frame.

2) Other Ceptral Coefficients related to MFCCs

In addition to MFCCs, we also included Rectangular Filter Cepstral Coefficients (RFCCs), Linear Filter Cepstral Coefficients (LFCCs) and Inverted Mel-frequency Cepstral Coefficients (IMFCCs). The extraction process of these feature sets is very similar to that of MFCCs. There are, however, subtle differences in the selection of filters and the chosen frequency scale [19]. For instance, RFCCs use a bank of 24 uniform non-overlapping rectangular filters distributed over a linear frequency scale. The procedure for calculating LFCCs is similar to that of RFCCs [41], however, it uses triangular filters instead of rectangular filters [19], [42]. Finally, to calculate IMFCCs, overlapping triangular filters are linearly placed over an inverted-mel scale [43]. This means that IMFCCs will emphasize the higher frequency region of the spectrum, i.e., opposite to human perception. Similar to MFCCs, we have also used the deltas and delta-deltas of these audio features in our experiments.

3) Spectrogram

Spectrograms provide a visual representation of the spectrum of frequencies present in an audio signal over time. They are nowadays widely used as features in image/audio classification and separation systems [44], [45]. When tested on a sound event or speech classification task, they have shown to provide a significant improvement in classification performance when supplemented with other traditional audio features such as MFCCs and LPCCs [46], [47]. In this paper, we obtain the spectrum of an audio signal by calculating the short-time Fourier transform (STFT) of the audio input. This can then be unwrapped to get the amplitude and phase information [18]. Both of these are considered as separate features for this investigation.

4) Constant Q Cepstral Coefficients (CQCCs)

CQCCs make use of a perceptually motivated time-frequency analysis known as constant Q transform (CQT). The frequency bins in Fourier-based approaches are often regular spaced and will result in a variable Q factor $\left(Q = \frac{centerfrequencyofband}{bandwidth}\right)$ [18], [48], [49], based on the center frequency of each particular band. In CQT, however, the bins are spaced geometrically, in order to ensure a constant Q factor. In stark contrast to the Fourier approach, CQT offers a higher frequency resolution at lower frequencies and a higher temporal resolution at higher frequencies. The extraction process of CQCCs begins by taking the constant-Q-transform of the input audio frame. The power spectrum is then computed and its logarithm is calculated. Since the k bins in a constant Q transforms are each on a different scale, the log power spectrum has to be resampled before applying the discrete cosine transform (DCT). This resampling is achieved by converting the geometric space to linear space. This involves a down-sampling operation over the first k bins (i.e., the low frequency part) and an up-sampling operation for the remaining bins (i.e., the high frequency). This linearization of the frequency scale of the CQT further preserves the orthogonality of the DCT outputs [48]. In this investigation, 20 CQCCs, their deltas and delta-deltas have been extracted from each audio frame, thus forming a total of 60 features.

5) Linear predictive cepstral coefficients (LPCCs)

Another standard set of features widely used in speech recognition are LPCCs [39], [50]. They are computed from the smoothed auto-regressive power spectrum of an audio frame. LPCC extraction begins with the estimation of the linear predictive coding (LPC) coefficients of an audio frame. These linear predictive coefficients are converted to LPCCs by using a recursion algorithm as shown below [51]. In this investigation, 16 LPCCs have been extracted from every audio frame.

Consider the speech input $s(n)$ to an all-pole LPC filter, which has p linear predictive coefficients, and a prediction error signal of $e(n)$. Let E_e be the power of this error signal. Now the p coefficients $[a_0, a_1, \ldots, a_{p-1}]$ can be recursively converted to n LPCCs $[L_0, L_1, \ldots, L_{n-1}]$ as follows for each LPCC.

For $m = 0$:

$$L_m = \ln(E_e) \tag{3}$$

For $1 \leq m \leq p$:

$$L_m = -a_m + \frac{1}{m}\sum_{k=1}^{m-1}\{-(m-k)a_k L_{m-k}\} \tag{4}$$

For $p < m < n$:

$$L_m = \frac{1}{m}\sum_{k=1}^{p}\left\{-\frac{(m-k)}{m}a_k L_{m-k}\right\} \tag{5}$$

6) Spectral sub-band Centroid Coefficients

Spectral sub-band centroids, as the name suggest, represent the centroids of selected sub-bands of the spectrum. Both spectral sub-band centroid magnitude (SCM) and spectral sub-band centroid frequency (SCF) can be extracted from a given sub-band, whereby the characteristics of the latter one are similar to that of formant frequency [19], [52]. Sub-band centroid features, supplemented with cepstral features, have shown to work very well for speech recognition [53].

The SCMC and SCFC extraction process starts by creating the spectrum for a given speech frame and dividing this into k sub-bands, whereby each sub-band is defined by a fixed lower edge and an upper edge frequency. SCFC is then calculated from the average frequency for each sub-band, and weighted by the normalized energy of each frequency component in that sub-band. Similarly, SCMC is calculated as the average magnitude of each sub-band, weighted by the frequency of each magnitude component in that sub-band. For calculating the SCMC, we then take the logarithm of this result and apply DCT [19], [52]. For our experiments, we have used 8,192 bins (sub-bands) of spectrum extracted from every 20ms audio frame.

7) Complex Cepstral cofficients (CCCs)

Cepstral analysis, often referred to as homomorphic filtering [54], can be used to separate out various components in a speech production (i.e., source-filter) model. The Complex Cepstrum is defined as the inverse Fourier transform (IFT) of the logarithm of the Fast Fourier Transform (FFT) of a signal [55]. By taking the FFT of the speech signal, the convolution between the source and filter components in the time domain will be converted into their product in the frequency domain. The logarithm operator then transforms this product operation into a sum operation of both components. Finally, an IFT is taken to bring these summed components back into the time domain (quefrency domain) [56].

The resultant complex cepstral coefficients (CCs) characterize the slow and fast varying components of speech. Slow-varying components (e.g. contribution of pitch) are concentrated in the upper part of the cepstral domain, whereas the fast-varying components (e.g. contribution of the vocal tract filter) are concentrated in the lower part [55]. Since CCs carry more speech specific information than most of the other cepstral coefficients that we discussed above, they typically result in higher speaker recognition performance [57]. In the experiments below, we have used the lower 50 cepstral coefficients extracted from every frame.

B. LEARNED FEATURE REPRESENTATION

An autoencoder is trained for each feature set, such that a dense presentation of all original audio features is learned. Each of these resulting dense representations (one per feature set) are then concatenated to form the input for the final classification model.

To be able to train the autoencoders, a number of hyper-parameters are set, including encoding-size (code-size =

100), number of layers in encoder and subsequently the decoder (one layer for each), number of nodes per layer (ten per layer), and loss-function [58], [59]. Figure 5 shows the autoencoder architecture used in our experiments for a particular feature vector of size $1 \times n$.

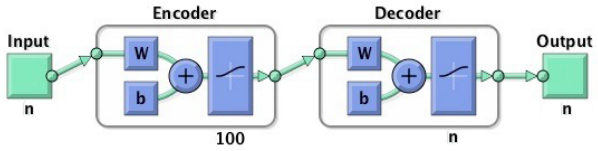

FIGURE 5. Architecture of the autoencoder used in this investigation, for a feature vector of size n.

In the figure, the size of the input feature vector is $1 \times n$. The code-size is set to 100, and the encoder/decoder each consist of 1 layer. Mean squared error (MSE) is used as the loss function. MSE is the recommended loss function for cases in which the input values are in the range of zero to one [31]. In this example, the encoder compresses the original $1 \times n$ input to a lower dimensional code (in this case of size 100) and the decoder reconstructs the output of size $1 \times n$ using this code. The encoder will be trained until it reaches the performance target. In our case, this in turn corresponds to achieving the lowest gradient, which corresponds to the lowest loss in prediction error. To finish the training in a finite time, the maximum number of epochs in the training phase is set to 200. This training cutoff was set after careful investigation of the evolution of the MSE loss when using different training samples, corresponding to each of the feature vectors. We found that the MSE is very low for 200 epochs and did not change too much in subsequent epochs.

C. DATA AUGMENTATION

In addition to creating a new, dense representation, the autoencoder trained on genuine recordings is also used to augment our dataset (see Figure 6). This is achieved by using the autoencoder to reconstruct the feature sets for the $1,508$ genuine recordings in the original training set. These $1,508$ reconstructions are then added to the training set. A similar procedure is performed on the spoofed recordings.

IV. EXPERIMENTAL SETUP

We investigate the influence of both the known and learned feature representations on spoofing detection accuracy. The setup of our experiment, including dataset, evaluation methodology, are discussed below.

A. AUDIO DATASET - ASVSPOOF 2017

The 'out in the wild' ASVspoof 2017 dataset (protocol V2) is used for our experiments [60]. This corpus originated from RedDots corpus3 and contains recordings collected by researchers using Android mobile phones [61], [62]. The recordings include both replayed (spoofed) and non-replayed (original) utterances. The former (i.e., replayed) are captured versions of the original RedDots recordings, meaning that an utterance of an original target speaker was replayed through transducers of varying quality, and recorded using a mobile phone), whereas the latter (i.e., non-replayed utterances) are original recordings. Replayed utterances can be used to model a 'stolen voice'.

We chose to work with this dataset, as it contains audio from different, unseen configurations of the spoofing algorithms in the testing set. The aim of this paper is to assess the validity of different audio features when building a robust spoofing detection model, which remains effective without knowing which spoofing technique was used.

The ASVspoof corpus is divided into three subsets: training, development and testing set. The training and development set are used to train and validate our spoofing detection system, whereas the testing set is used to test the performance. Since heterogeneity in the data is highly essential for developing reliable spoofing countermeasures, no two-same-speaker recordings are included in any of the three subsets. Further, the data collection sites are also chosen to be distinct for all the three subsets [13], [60]. More information about corpus is shown in Table 1.

B. EQUAL ERROR RATE (EER) AS THE EVALUATION METRIC

The Equal Error Rate (EER) is used as the primary metric to evaluate the performance of our spoofing detection system [13], [63]. The false positive rate $(p_{fp}(\theta))$ and false negative rate $(p_{fn}(\theta))$ at a particular threshold are required to calculate EER:

$$p_{fp}(\theta) = \frac{\text{Number of replay trials with likelihood} > \theta}{\text{Total number of replay trials}} \quad (6)$$

$$p_{fn}(\theta) = \frac{\text{Number of non-replay trials with likelihood} \leq \theta}{\text{Total number of non-replay trials}} \quad (7)$$

The EER corresponds to the particular value of θ for which $p_{fp}(\theta)$ and $p_{fn}(\theta)$ are approximately equal. The lower the EER value, the better is the performance of the system. In this paper, the term 'performance' refers to EER results, which can be seen as a measure of accuracy.

C. EXPERIMENTAL METHODOLOGY

In order to thoroughly examine the effect of different audio features and the autoencoder, a total of three experiments are conducted, as represented in Figure 3.

In Experiment 1, the audio features explained in Section III-A are extracted from the training set and postprocessed to create individual feature-specific GMM models. This means that a UBM is created for each set of features (e.g. MFCCs, CQCCs, etc.), using the $1,508$ genuine and $1,508$ spoofed audio files in the training set. We evaluate and compare the performance of models based on each of these

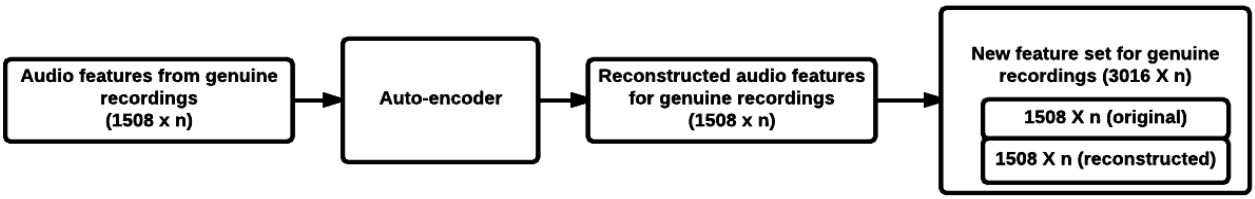

FIGURE 6. Data augmentation (with n number of features, and $1,508$ original genuine recordings). A similar procedure is performed on the spoofed recordings.

TABLE 1. Statistics of ASVspoof 2017 database.

Subset	Number of speakers	Number of replay sessions	Number of replay configuration	Number of non-replay utterances	Number of replay utterances
Training	10	6	3	1,508	1,508
Development	8	10	10	760	950
Testing	24	163	112	1,298	12,008

features separately. In addition, we explore a fused model that takes as input the likelihoods corresponding to the models of all of the feature sets investigated.

In the second experiment, an autoencoder is used to learn a new representation for each of the original feature sets, and to augment the training set. We again evaluate the performance of each feature set, and compare this to a hybrid system that is obtained by fusing the individual likelihoods obtained per feature set.

The third experiment is a fusion of all the results from Experiment 1 and 2, whereby the GMM-UBM likelihoods for each of the models based on individual features sets are combined using a logistic regression fusion.

V. RESULTS

We discuss the performance of the different features through three experiments as described below, followed by a comparison with other state-of-the-art models.

A. EXPERIMENT 1: PERFORMANCE OF KNOWN AUDIO FEATURES

The performance of models built on each of the different feature sets is displayed in Table 2. The model based on CQCCs performs best. This result aligns with previous studies conducted in this arena [40], [48].

Interestingly, the performance of the model based on MFCCs, one of the de-facto features extracted in many of the speech/speaker recognition, is found to be the worst. As explained earlier, MFCCs capture perceptually relevant aspects of the speech spectrum. In a replay attack scenario, the imposter is playing speech samples captured from a genuine speaker, hence, it could be expected that the perceptual/ audible artifacts would be the same as that of the original audio, which explains the low performance when using MFCCs.

Three models based on cepstral co-efficients (RFCCs, LFCCs and IMFCCs), whose extraction process is very similar to that of MFCCs yet uses a different selection of filters and frequency scales, perform well. Among these three, the inverted-MFCCs, which models characteristics opposite to human perception, is the best performer. This falls in line with our previous hypothesis that replayed audio may have inaudible artifacts and that IMFCCs might be capturing cues outside of human perception.

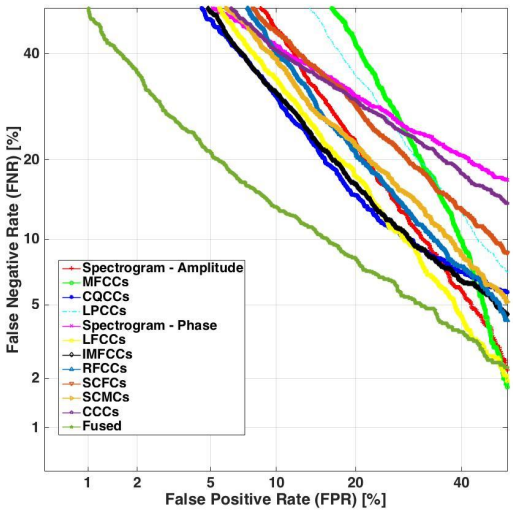

FIGURE 7. DET Curves showing the performance of spoofing the detection models for Experiment 1.

The model based on spectral amplitude is a good candidate to distinguish spoofed recordings from genuine. A related model, using spectral phase information, however, did not perform up to the expectations set in [18].

The performance of models that use features based on spectral sub-band centroids (e.g. SCMCs and SCFCs) is similar, with a slight edge for amplitude based features (i.e., SCMCs) versus frequency based features (i.e., SCFCs). This trend is also observed when comparing spectral amplitude versus spectral phase. The effect can be due to the way

TABLE 2. Results from Experiment 1.

Features	EER
MFCCs	27.2
CQCCs	**17.5**
Spectrogram - Amplitude	20.9
Spectrogram - Phase	26.8
LPCCs	26.1
RFCCs	20.3
LFCCs	19.0
IMFCCs	**18.1**
SCMCs	21.1
SCFCs	24.0
CCCs	25.9
Fused result	**12.2**

The features resulting in the three best performing models are indicated in bold.

that an imposter tries to capture genuine speech samples, as the energy/amplitude of the captured speech will be directly related to the placement of the capturing device with respect to the mouth position of the genuine speaker. This could be reflected in the extracted features.

Finally, models built using complex cepstral coefficients and LPCCs do not perform as to expected. The former are expected to carry more speech specific information. The latter inherit advantages of LPC (i.e., linear prediction coefficients), which are related to the speech production (source-filter) model. Both of these features result in models with a similar performance to those using MFCCs.

In order to achieve better spoofing detection performance and to overcome the limitation in performance of individual features, the results from models based on different feature sets have been merged using logistic regression fusion. As expected, the resulting hyrbid model has a far better overall performance.

The results of Experiment 1 are visualised in Figure 7 by using an detection error trade-off (DET) graph. DET graphs show the performance of detection tasks that involve a trade-off error. In this case, false negative and false positive rates for each set of features are plotted against each other in the DET graph. The EER can be observed from the DET graph as the point where the percentage of false negatives equals the false positive rate [64]. Better performance is reflected by closer proximity to the origin. This graph confirms our previous conclusions and identifies the fused model as the best performing.

B. EXPERIMENT 2: PERFORMANCE WITH AUTOENCODER FEATURES AND DATA AUGMENTATION

Based on the increased popularity of latent neural network representations, we expected a steep increase in performance when building the model using machine learned features, and including data augmentation. According to our results listed in Table 3 and Figure 8, this is, however, not the case for spoofing detection, at least not when using only these new, learned features. None of the models built using the new feature sets and augmented dataset, are able to match the performance of CQCC in Experiment 1. The performance of the model with an autoencoder based on CQCC features in Experiment 2 is slightly below that of Experiment 1. Models with average performance in Experiment 1 (e.g., SCFC, CCCs, LPCCs), keep a similar ranking in experiment 2, however, their absolute EER performance is compromised. The reason for this decreased performance may be related to the fact that there is a higher variance in the machine learned features compared to the original features, which can cause a drop in performance. This warrants further investigation. The models based on spectral amplitude and MFCCs are the only ones that increase their performance in terms of EER when compared to that of Experiment 1.

TABLE 3. Results from Experiment 2.

Features	EER
MFCCs	24.2
CQCCs	**21.5**
Spectrogram - Amplitude	**20.2**
Spectrogram - Phase	26.5
LPCCs	31.2
RFCCs	27.9
LFCCs	24.0
IMFCCs	21.6
SCMCs	23.0
SCFCs	35.9
CCCs	32.0
Fused result	**12.6**

The features resulting in the three best performing models are indicated in bold.

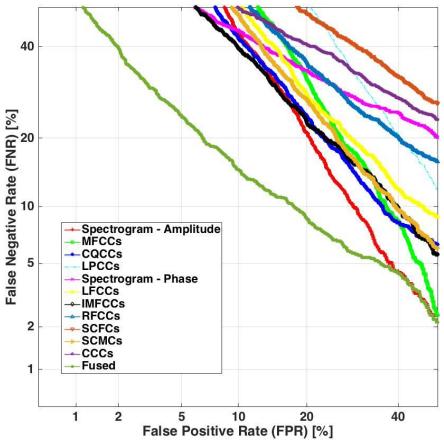

FIGURE 8. DET Curves showing the performance of the spoofing detection models for Experiment 2.

When looking at the performance ranking of models based on the different feature sets (see Figure 9), we see that it is different from the ranking in Experiment 1. For example, models based on spectral amplitude are now found to be the best performing models, whereas they are only the fifth best model in Experiment 1.

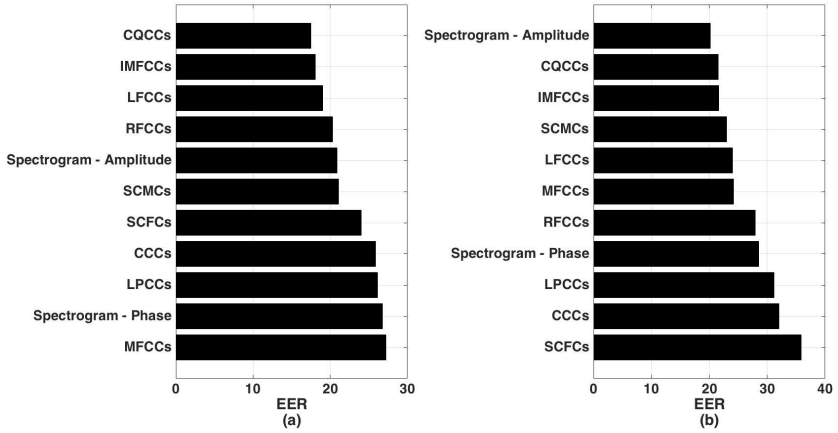

FIGURE 9. Ranking of models built on different sets of audio features based on their EER performance: (a) with known features (Experiment 1) (b) with autoencoder features and data augmentation (Experiment 2).

C. EXPERIMENT 3: PERFORMANCE OF HYBRID SYSTEM

We explore the effectiveness of a hybrid system that combines both the predictions of the models built on known and machine learned features. The results of this experiment are shown in Table 4 and the resulting DET curves are shown in Figure 10. A superior performance is found when using a logistic regression fusion to combining the output of individual models. This could be attributed to the larger, more diverse feature space (coming from multiple models). The hybrid system outperforms all individual models, including the fused results from both Experiment 1 and 2. This confirms that each of the audio feature sets capture relevant aspects of audio signals that, when put together, form the most powerful model.

TABLE 4. Results from Experiment 3.

Experiment	EER (Calibrated)
Experiment 1 (Fused)	12.2
Experiment 2 (Fused)	12.6
Experiment 1 and 2 (Fused)	10.8

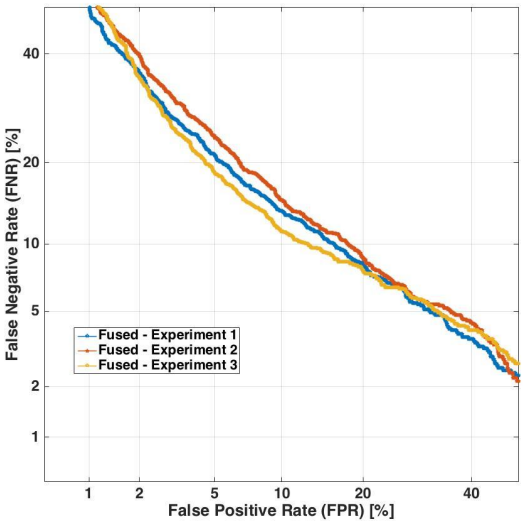

FIGURE 10. DET Curves showing the performance of spoofing the detection models for Experiment 3.

D. COMPARISON WITH STATE-OF-THE-ART SYSTEMS

Table 5 compares the performance of our proposed system with other systems trained on the ASVSpoof2017 dataset in terms of EER. When comparing to other models based on GMM, such as System 3 [65], we achieve better performance due to our unique feature set and dataset augmentation. For instance, with the traditional feature set from Experiment 1, a slightly better performance than System 3 is achieved, which uses the same core model, but less features. In Experiment 3, we further improve the accuracy of the proposed system by integrating an autoencoder. By using this to augment the training set and by fusing the results based on both traditional and learned features, the EER improves by 20%, thus reaching 10.8. The only system that outperforms our proposed architecture integrates, in addition to GMM, a recursive neural networks [66]. It would be conceivable that the performance of said model would also improve when incorporating our feature learning and dataset augmentation method. This is a topic for future research.

VI. CONCLUSIONS

We examine the effect of a large variety of audio features on the performance of a GMM-UBM based replay audio spoofing detection system. One of the goals of this paper is to pinpoint the most important features when building a robust model that works on an 'in the wild dataset', i.e., without having any information about the used replay spoofing technique. In addition to thoroughly comparing the models built on the different features, we also provide a clear procedure for proper pre-and postprocessing of these features, which we hope will be valuable to other researchers.

TABLE 5. EER values obtained for different models on the ASVSpoof2017 dataset with various features.

Systems	EER	Features	classifier	Fusion	Training Dataset
Proposed System	**10.8**	**MFCCs, CQCCs, Spectrogram, LPCCs, RFCCs, LFCCs, IMFCCs, SCMCs, SCFCs, CCCs and Autoencoder reconstructed features**	**GMM**	**Yes**	**Training**
System 1 [66]	6.73	Power Spectrum, LPCCs	CNN, GMM, RNN, Total Variation	Yes	Training
System 2 [67]	12.34	CQCCs, MFCCs, PLP	GMM-UBM, GSV-SVM, ivec-PLDA, GBDT, Random Forest	Yes	Training
System 3 [65]	14.03	MFCCs, IMFCCs, RFCCs, LFCCs, PLPCCs, CQCCs, SCMCs, SSFCs	GMM, FF-ANN	Yes	Training+Development
System 4 [68]	14.66	RFCCs, MFCCs, IMFCCs, LFCCs, SSFCs, SCMCs	GMM	Yes	Training+Development
System 5 [69]	15.97	Linear Filterbank Feature	GMM, CT-DNN with convolutional layer & time-delay layers	Yes	Training
ASVSpoof Baseline (B01) [13]	24.77	CQCCs	GMM	No	Training+Development
ASVSpoof Baseline (B02) [13]	30.6	CQCCs	GMM	No	Training

In our experiments, we explore both known audio features, and those learned by an autoencoder (i.e., using a feed forward neural network). The former includes some de-facto features often used this field, as well as some potentially new features that would be able to distinguish genuine speech from spoofed one. The included features are MFCCs, spectrogram, CQCCs, LPCCs, IMFCCs, RFCCs, LFCCs, SCFCs, SCMCs, and CCCs. Secondly, a novel representation for each of these feature sets is learned by an autoencoder. In addition, this autoencoder is used to augment the training set.

The performance of each of the models built with these different feature sets is reported in terms of EER. When using only known features, or only autoencoder features, the resulting performance is around 12 in terms of EER. When creating a hybrid system that incorporates both types of features, we achieve a superior performance of 10.8. This competes with the current state-of-the-art and reiterates the importance of integrating different types of audio features, both known and machine learned, in order to develop a robust model for replay spoofing detection.

REFERENCES

[1] S. Levy, Hackers: Heroes of the computer revolution. Anchor Press/Doubleday Garden City, NY, 1984, vol. 14.

[2] W. A. Galston, T. C. Hilde, P. Levine, and L. D. Introna, The Internet in public life. Rowman & Littlefield, 2004.

[3] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, T. Leino, and A.-M. Laukkanen, "I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry." in Interspeech, 2013, pp. 930–934.

[4] Y. W. Lau, M. Wagner, and D. Tran, "Vulnerability of speaker verification to voice mimicking," in Intelligent Multimedia, Video and Speech Processing, 2004. Proceedings of 2004 International Symposium on. IEEE, 2004, pp. 145–148.

[5] S. K. Ergünay, E. Khoury, A. Lazaridis, and S. Marcel, "On the vulnerability of speaker verification to realistic voice spoofing," in Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on. IEEE, 2015, pp. 1–6.

[6] D. Mukhopadhyay, M. Shirvanian, and N. Saxena, "All your voices are belong to us: Stealing voices to fool humans and machines," in European Symposium on Research in Computer Security. Springer, 2015, pp. 599–621.

[7] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification." in APSIPA, 2014, pp. 1–5.

[8] P. L. De Leon, M. Pucher, and J. Yamagishi, "Evaluation of the vulnerability of speaker verification to synthetic speech," 2010.

[9] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga, "Evaluation of speaker verification security and detection of hmm-based synthetic speech," IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 8, pp. 2280–2290, 2012.

[10] T. Kinnunen, Z.-Z. Wu, K. A. Lee, F. Sedlak, E. S. Chng, and H. Li, "Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech," in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012, pp. 4401–4404.

[11] F. Alegre, R. Vipperla, and N. Evans, "Spoofing countermeasures for the protection of automatic speaker recognition systems against attacks with artificial signals," in INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, 2012.

[12] Z. Kons and H. Aronowitz, "Voice transformation-based spoofing of text-dependent speaker verification systems." in INTERSPEECH, 2013, pp. 945–949.

[13] T. Kinnunen, N. Evans, J. Yamagishi, K. A. Lee, M. Sahidullah, M. Todisco, and H. Delgado, "ASVspoof 2017: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," Training, vol. 10, no. 1508, p. 1508, 2017.

[14] T. B. Amin, J. S. German, and P. Marziliano, "Detecting voice disguise from speech variability: Analysis of three glottal and vocal tract measures," in Proceedings of Meetings on Acoustics 166ASA, vol. 20, no. 1. ASA, 2013, p. 060005.

[15] J. Villalba and E. Lleida, "Detecting replay attacks from far-field recordings on speaker verification systems," in European Workshop on Biometrics and Identity Management. Springer, 2011, pp. 274–285.

[16] P. L. De Leon, I. Hernaez, I. Saratxaga, M. Pucher, and J. Yamagishi, "Detection of synthetic speech for the problem of imposture," in Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE, 2011, pp. 4844–4847.

[17] Z. Wu, E. S. Chng, and H. Li, "Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition," in Thirteenth Annual Conference of the International Speech Communication Association, 2012.

[18] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," speech communication, vol. 66, pp. 130–153, 2015.

[19] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," 2015.

[20] M. Sahidullah, H. Delgado, M. Todisco, H. Yu, T. Kinnunen, N. Evans, and Z.-H. Tan, "Integrated spoofing countermeasures and automatic speaker verification: An evaluation on asvspoof 2015," 2016.

[21] H. Yu, Z.-H. Tan, Y. Zhang, Z. Ma, and J. Guo, "Dnn filter bank cepstral coefficients for spoofing detection," Ieee Access, vol. 5, pp. 4779–4787, 2017.

[22] N. Evans, M. Sahidullah, J. Yamagishi, M. Todisco, K. A. Lee, H. Delgado, T. Kinnunen et al., "The 2nd automatic speaker verification spoofing and countermeasures challenge (asvspoof 2017) database, version 2," 2018.

[23] F. K. Soong, A. E. Rosenberg, B.-H. Juang, and L. R. Rabiner, "Report: A vector quantization approach to speaker recognition," AT&T technical journal, vol. 66, no. 2, pp. 14–26, 1987.

[24] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," Digital signal processing, vol. 10, no. 1-3, pp. 19–41, 2000.

[25] D. Matrouf, N. Scheffer, B. Fauve, and J.-F. Bonastre, "A straightforward and efficient implementation of the factor analysis model for speaker verification," in Eighth Annual Conference of the International Speech Communication Association, 2007.

[26] R. Vergin and D. O'Shaughnessy, "Pre-emphasis and speech recognition," in Electrical and Computer Engineering, 1995. Canadian Conference on, vol. 2. IEEE, 1995, pp. 1062–1065.

[27] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'79., vol. 4. IEEE, 1979, pp. 208–211.

[28] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," IEEE Transactions on speech and audio processing, vol. 9, no. 5, pp. 504–512, 2001.

[29] M. Brooks, "The voicebox toolkit," http://www/ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html, 2013.

[30] A. Ng, "Sparse autoencoder, cs294a lecture notes, vol. 72, no. 2011, pp. 1-19," 2011.

[31] N. Hubens, "Deep inside: Autoencoders," https://towardsdatascience.com/deep-inside-autoencoders-7e41f319999f, 2017, accessed: 2018-09-02.

[32] O. M. Strand and A. Egeberg, "Cepstral mean and variance normalization in the model domain," in COST278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction, 2004.

[33] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," Speech Communication, vol. 25, no. 1-3, pp. 133–147, 1998.

[34] S. O. Sadjadi, M. Slaney, and L. Heck, "Msr identity toolbox v1. 0: A matlab toolbox for speaker-recognition research," Speech and Language Processing Technical Committee Newsletter, vol. 1, no. 4, pp. 1–32, 2013.

[35] D. A. Van Leeuwen and N. Brümmer, "An introduction to application-independent evaluation of speaker recognition systems," in Speaker classification I. Springer, 2007, pp. 330–353.

[36] G. S. Morrison, "Tutorial on logistic-regression calibration and fusion: converting a score to a likelihood ratio," Australian Journal of Forensic Sciences, vol. 45, no. 2, pp. 173–197, 2013.

[37] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal processing magazine, vol. 29, no. 6, pp. 82–97, 2012.

[38] L. Muda, B. KM, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques," Journal of Computing, vol. 2, no. 3, pp. 138–143, 2010.

[39] L. R. Rabiner and B.-H. Juang, Fundamentals of speech recognition. PTR Prentice Hall Englewood Cliffs, 1993, vol. 14.

[40] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients," in Speaker Odyssey Workshop, Bilbao, Spain, vol. 25, 2016, pp. 249–252.

[41] T. Hasan, S. O. Sadjadi, G. Liu, N. Shokouhi, H. Bořil, and J. H. Hansen, "Crss systems for 2012 nist speaker recognition evaluation," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 6783–6787.

[42] F. Alegre, A. Amehraye, and N. Evans, "A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns," in Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on. IEEE, 2013, pp. 1–8.

[43] S. Chakroborty, A. Roy, and G. Saha, "Improved closed set text-independent speaker identification by combining mfcc with evidence from flipped filter banks," International Journal of Signal Processing, vol. 4, no. 2, pp. 114–122, 2007.

[44] Y. M. Costa, L. S. Oliveira, and C. N. Silla Jr, "An evaluation of convolutional neural networks for music classification using spectrograms," Applied soft computing, vol. 52, pp. 28–38, 2017.

[45] K. W. E. Lin, B. Balamurali, E. Koh, S. Lui, and D. Herremans, "Singing voice separation using a deep convolutional neural network trained by ideal binary mask and cross entropy," Neural Computing and Applications, Dec 2018. [Online]. Available: https://doi.org/10.1007/s00521-018-3933-z

[46] Q. T. Nguyen et al., "Speech classification using sift features on spectrogram images," Vietnam Journal of Computer Science, vol. 3, no. 4, pp. 247–257, 2016.

[47] J. Dennis, H. D. Tran, and H. Li, "Spectrogram image feature for sound event classification in mismatched conditions," IEEE signal processing letters, vol. 18, no. 2, pp. 130–133, 2011.

[48] M. Todisco, H. Delgado, and N. Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," Computer Speech & Language, vol. 45, pp. 516–535, 2017.

[49] Z. Wu, T. Kinnunen, N. Evans, and J. Yamagishi, "Asvspoof 2015: the first automatic verification spoofing and countermeasures challenge evaluation plan," in IEEE Signal Processing Society Speech and language Technical Committee Newsletter, 2014.

[50] J. Makhoul, "Linear prediction: A tutorial review," Proceedings of the IEEE, vol. 63, no. 4, pp. 561–580, 1975.

[51] Matlab, "DSP System Toolbox, Transforms and Spectral analysis, Linear prediction." http://www.mathworks.com/help/dsp/ref/lpctofromcepstralcoefficients.html, 2017, accessed: 2017-09-02.

[52] J. M. K. Kua, T. Thiruvaran, M. Nosratighods, E. Ambikairajah, and J. Epps, "Investigation of spectral centroid magnitude and frequency for speaker recognition." in Odyssey, 2010, p. 7.

[53] K. K. Paliwal, "Spectral subband centroid features for speech recognition," in Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on, vol. 2. IEEE, 1998, pp. 617–620.

[54] A. Oppenheim and R. Schafer, "Homomorphic analysis of speech," IEEE Transactions on Audio and Electroacoustics, vol. 16, no. 2, pp. 221–226, 1968.

[55] L. R. Rabiner and R. W. Schafer, Theory and applications of digital speech processing. Pearson Upper Saddle River, NJ, 2011, vol. 64.

[56] B. P. Bogert, "The quefrency alanysis of time series for echoes; cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking," Time series analysis, pp. 209–243, 1963.

[57] B. B. Nair, E. A. Alzqhoul, and B. J. Guillemin, "Comparison between mel-frequency and complex cepstral coefficients for forensic voice comparison using a likelihood ratio framework," in Proceedings of the World Congress on Engineering and Computer Science, San Francisco, USA, 2014.

[58] X. Zhang, Y. Fu, S. Jiang, L. Sigal, and G. Agam, "Learning from synthetic data using a stacked multichannel autoencoder," in Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on. IEEE, 2015, pp. 461–464.

[59] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," Neural networks, vol. 6, no. 4, pp. 525–533, 1993.

[60] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," 2017.

[61] T. Kinnunen, M. Sahidullah, M. Falcone, L. Costantini, R. G. Hautamäki, D. Thomsen, A. Sarkar, Z.-H. Tan, H. Delgado, M. Todisco et al., "Reddots replayed: A new replay spoofing attack corpus for text-dependent speaker verification research," in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, 2017, pp. 5395–5399.

[62] K. A. Lee, A. Larcher, G. Wang, P. Kenny, N. Brümmer, D. v. Leeuwen, H. Aronowitz, M. Kockmann, C. Vaquero, B. Ma et al., "The reddots data collection for speaker recognition," in Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[63] J. Oglesby, "What's in a number? moving beyond the equal error rate," Speech communication, vol. 17, no. 1-2, pp. 193–208, 1995.

[64] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The det curve in assessment of detection task performance," National Inst of Standards and Technology Gaithersburg MD, Tech. Rep., 1997.

[65] A. R. Gonçalves, R. P. Violato, P. Korshunov, S. Marcel, and F. O. Simoes, "On the generalization of fused systems in voice presentation attack detection," in 2017 International conference of the biometrics special interest group (BIOSIG). IEEE, 2017, pp. 1–5.

[66] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks." in Interspeech, 2017, pp. 82–86.

[67] Z. Ji, Z.-Y. Li, P. Li, M. An, S. Gao, D. Wu, and F. Zhao, "Ensemble learning for countermeasure of audio replay spoofing attack in asvspoof2017." in INTERSPEECH, 2017, pp. 87–91.
[68] P. Korshunov and S. Marcel, "A cross-database study of voice presentation attack detection," in Handbook of Biometric Anti-Spoofing. Springer, 2019, pp. 363–389.
[69] L. Li, Y. Chen, D. Wang, and T. F. Zheng, "A study on replay attack and anti-spoofing for automatic speaker verification," Proc. Interspeech 2017, pp. 92–96, 2017.

BALAMURALI B T is a postdoctoral research fellow working at the Singapore University of Technology and Design. He received his Ph.D. in Electrical and Computer Engineering from the university of Auckland, New Zealand in 2015. After his Ph.D, he worked as a researcher in Gastro intestinal group in Auckland Bio-engineering Institute and was a lecturer in Auckland university of technology. Prior to his Ph.D endeavor, he was a design and development engineer in Tata Elxsi, India. He is currently passionate about artificial intelligence and trying to solve a variety of problems related to bio-signal processing detection and classification, automatic speech/speaker recognition, spoofed-speech detection, blacklisted speaker identification, blind source separation, music classification, fluid flow classification, fruit ripeness detection etc.

KIN WAH EDWARD LIN is now an AIST Postdoctoral Researcher, working at the National Institute of Advanced Industrial Science and Technology (AIST), Japan. He received his Ph.D. degree from Singapore University of Technology and Design (SUTD) in 2018. Edward has obtained many scholarships to study in top schools and universities in Singapore and Hong Kong. He also has 3 years undergraduate teaching experience and 3 years working experience in Hong Kong IT Industry. He has published 2 WiFi-related papers during his MPhil study and 8 Audio-related paper during his PhD study. His research interests now include Audio-related iOS app and Singing Voice Separation.

SIMON LUI received his Ph.D. degree in Computer Science from The Hong Kong University of Science and Technology (HKUST) in 2011. He is currently the Director of Tencent Music Entertainment Group (TME), and Adjunct Assistant Professor of the Singapore University of Technology and Design (SUTD). Simon was an Assistant Professor of SUTD in 2012-2018, and a visiting scholar of the Massachusetts Institute of Technology (MIT) CSAIL in 2012-2013. His primary research interests are machine learning and music information retrieval. Simon developed several best selling apps in the iOS stores in 7 countries. His business story was reported by CNN International and featured in IEEE Institute.

JER-MING CHEN is an Assistant Professor at the Singapore University of Technology and Design. He received his Ph.D. in Applied Physics from the University of New South Wales in 2010. His primary research interest is in the interaction of coupled resonators in acoustics. Jer-Ming has also written lay-language scientific papers and has featured in the international media and popular press, including newspapers (e.g. New York Times, UK Telegraph, Sydney Morning Herald), TV and radio documentaries (BBC, ABC, Network Ten), and popular science magazines (Physics Today, Scientific American, The Straight Dope).

DORIEN HERREMANS (SM'17) is an Assistant Professor at Singapore University of Technology and Design, with a joint appointment at the Institute of High Performance Computing at the Agency for Science Technology and Research, A*STAR. In 2015, she was awarded the individual Marie-Curie Fellowship for Experienced Researchers, and worked at the Centre for Digital Music, Queen Mary University of London. Prof. Herremans received her PhD in Applied Economics from the University of Antwerp. After graduating as a commercial engineer in management information systems at the University of Antwerp in 2005, she worked as a Drupal consultant and was an IT lecturer at Les Roches University in Bluche, Switzerland. Prof. Herremans' research focuses on the intersection of machine learning/optimization and digital music/audio. She is a Senior Member of the IEEE and co-organizer of the First International Workshop on Deep Learning and Music as part of IJCNN, as well as guest editor for Springer's Neural Computing and Applications.