Received: 4 May 2020 | Revised: 27 January 2021 | Accepted: 5 February 2021

IET Signal Processing

DOI: 10.1049/sil2.12028

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Deep convolutional neural networks for double compressed AMR audio detection

Aykut Büker | Cemal Hanilçi

Department of Electrical and Electronics Engineering, Bursa Technical University, Bursa, Turkey

Correspondence

Aykut Büker, Department of Electrical and Electronics Engineering, Bursa Technical University, Bursa, Turkey.
Email: aykut.buker@btu.edu.tr

Funding information

Scientific and Technological Research Council of Turkey (TÜBITAK) under the project no. 118R071

Abstract

Detection of double compressed (DC) adaptive multi-rate (AMR) audio recordings is a challenging audio forensic problem and has received great attention in recent years. Here, the authors propose to use convolutional neural networks (CNN) for DC AMR audio detection. The CNN is used as (i) an end-to-end DC AMR audio detection system and (ii) a feature extractor. The end-to-end system receives the audio spectrogram as the input and returns the decision whether the input audio is single compressed (SC) or DC. As a feature extractor in turn, it is used to extract discriminative features and then these features are modelled using support vector machines (SVM) classifier. Our extensive analysis conducted on four different datasets shows the success of the proposed system and provides new findings related to the problem. Firstly, double compression has a considerable impact on the high frequency components of the signal. Secondly, the proposed system yields great performance independent of the recording device or environment. Thirdly, when previously altered files are used in the experiments, 97.41% detection rate is obtained with the CNN system. Finally, the cross-dataset evaluation experiments show that the proposed system is very effective in case of a mismatch between training and test datasets.

1 | INTRODUCTION

Recent developments on communication technologies and multimedia processing systems enable generating, editing, sharing and manipulating the information contents (e.g. text, audio, image and video) easily. Most of the internet users share such contents online. Consequently, people produce their own personal media in various forms (audio, image and video) which are mostly available to others. These contents can easily be altered, modified or manipulated by freely available multimedia editing tools. As a result of the widespread usage of information contents in digital forms and availability of the manipulation tools, multimedia contents usually appear as evidence in courts or law-enforcement offices. This addresses the problem of the multimedia forensics which focuses on the integrity and authenticity of a multimedia content [1, 2].

Research on multimedia forensics mostly focuses on image forensics such as camera identification from the images [3], image steganalysis which aims at identifying the existence of the hidden messages in images [4], image forgery detection [5] and detecting the double compressed images [6]. Although, similar problems have been addressed from the audio forensics point of view, the number of studies dealing with audio content is much less in comparison to the image forensics. Recording device identification [7], audio steganalysis [8], audio forgery detection [9] and double compressed audio detection [10] are the fundamental audio forensics problems similar to image forensics.

Among the various audio forensic tasks mentioned above, only a limited number of studies exist on double compressed (DC) audio detection. However, it is a challenging and important problem which requires more attention. Because an audio content can easily be edited, altered or manipulated using the freely available audio editing software and re-compressed back to the original compressing format in order to produce a perceptually indistinguishable altered audio. Therefore, here, the authors address the problem of detecting DC adaptive multi-rate (AMR) audio recordings. The AMR audio compression codec [11] is chosen because of the fact that the majority of the current mobile telephone handsets store the

audio content in AMR format and there exist several freely available programs which enable decompressing, editing and converting AMR audio into other formats. Thus, DC AMR audio files are more likely to appear in the courts as evidence. This requires reliable audio forensic techniques to verify the integrity and the authenticity of the DC AMR files.

Previous studies addressing the DC audio detection problem can be categorised into three groups according to audio compression codec format: (i) DC Moving Picture Experts Group (MPEG)-1 Audio Layer III (MP3) audio detection, (ii) detecting DC MPEG-2/4 advanced audio coding (AAC) audio and (iii) DC AMR audio recognition. Modified discrete cosine transform (MDCT) based features were generally used for detecting DC MP3 audio tracks in most studies. For example, the number of small values of the quantised MDCT coefficients obtained from the single compressed (SC) and DC MP3 files were compared in [12] for defeating the fake quality MP3 files and it was found that these numbers considerably differ between SC and DC MP3 audio files. The same authors proposed to use support vector machines (SVM) classifier using statistical features extracted from the first digits of the quantised MDCT coefficients for DC MP3 detection in [13]. In [14], 64 statistical features extracted from the MDCT coefficients were proposed to use for detecting DC MP3 files using SVM classifier. In [15, 16], authors proposed to use similarity measure between the histograms of the quantised MDCT coefficients of the questionized MP3 file (possibly DC) and its SC counterpart. Then a threshold was applied to the similarity measure for deciding whether it is a SC or DC MP3 audio. Ma et al. used statistical properties of the scale factors extracted from the windowed audio frames to detect DC MP3 files at the same bit-rate (BR) [17]. Statistical features extracted from the MP3 encoder parameters were proposed to use with SVM classifier for detecting multiple MP3 compression in [18]. Similar to DC MP3 audio detection, features extracted from the MDCT coefficients, statistics of the Huffman codebook indexes or the scale factors were used for recognizing DC AAC audio files [19, 20, 21]. A higher recognition accuracy was obtained when the BR of the second compression is equal or greater than that of the first compression, in general [19, 20, 21].

DC AMR audio detection has received great attention in recent years since almost any handheld device uses this codec for audio recordings. In [10], various frequency domain statistical features (subband energy ratio, low-frequency energy ratio, bispectral features and long-term linear predictive coding) were used with SVM classifier for DC AMR detection. Recognition accuracy was found to be superior when the second compression BR is higher than the first compression BR. In the first attempt to use deep learning approach for DC audio detection on AMR files, three different deep neural network (DNN) architectures were proposed, namely multilayer perceptron (MLP) with two hidden layers, stacked autoencoder (SAE) and MLP with dropout layers [22]. In that study, authors divided each 1-second-long audio signal into short non-overlapping segments consisting of 400 samples and then these raw audio frames were applied to the DNN. In the test stage, majority voting strategy was used for decision. In [23], the same authors proposed to use SAE for feature extraction and Gaussian mixture model-universal background model (GMM-UBM) classifier for classification of DC AMR files. Similar to their previous study, the input of the SAE is the raw audio frames consisting of 400 samples and feature vectors extracted from the hidden layers of the SAE were modelled using GMM-UBM. In a more recent study, statistical features extracted from linear prediction (LP) analysis were used with SVM classifier for DC AMR audio detection [24]. In [25], authors extracted long-term average spectrum (LTAS) and long-term average cepstral features and then applied these features as the input of the simple fully connected DNN. We observed that both features yield considerably promising results on DC AMR detection. In a more recent study concentrated on DC AMR audio recognition [26], authors first extracted seven AMR encoder parameters (e.g. LP coefficients, line spectral pairs, pitch gain, pitch lag etc.) and then calculated 657 statistical features from these encoder parameters. Finally, the number of features were reduced by feature selection method and SVM classifier was used for detection.

From the aforementioned short literature review on DC audio detection, one could notice that hand-crafted acoustic or statistical features were proposed to use for the DC audio detection in the majority of the existing studies. Meanwhile, most of these hand-crafted features rely on prior knowledge about encoding and decoding processes of the codec used to compress audio files. However, recent developments in machine learning revealed that DNN, especially deep convolutional neural networks (CNN) in particular, are highly capable of learning the features automatically when simple short-term processing representation of an audio signal is applied as the input [27, 28]. Motivated by this fact, the authors propose to use CNN for DC AMR audio detection. To the best of our knowledge, CNN has not been applied on DC audio detection task previously. However, it was found to give superior performance on various pattern recognition problems. Therefore, intuitively it would be interesting to investigate and analyse its performance on DC audio detection. The authors propose to use CNN for DC AMR audio detection for two different purposes: (i) as an end-to-end DC audio detector which accepts the input audio file and returns the output indicating whether input audio is DC or SC and (ii) as a feature extractor where we extract features from the proposed CNN architecture and then classify these features using SVM [29] classifier.

2 | DC AMR AUDIO

AMR audio codec is an audio compression format optimised for speech signals and developed by 3rd Generation Partnership Project (3GPP) [11]. It was originally developed as a narrowband speech codec which encodes narrowband signals with sampling rate of 8 kHz. AMR codec encodes the speech signal at eight different BR: $BR \in \{4.75, 5.15, 5.9, 6.7, 7.4, 7.95, 10.2, 12.2\}$ kbps (kbit/s) using code excited linear prediction (CELP) on speech frames consisting of 160 samples (frames with 20 ms duration). The CELP model parameters are

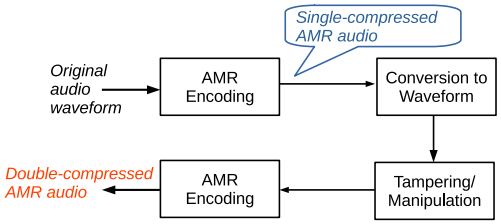

FIGURE 1 DC AMR audio recording generation process

extracted using a 10th order LP analysis and LP coefficients are converted into line spectral pairs (LSP). During the encoding, the CELP model parameters are encoded and transmitted. At the decoding stage, the encoded parameters are decoded first and then speech signal is synthesised from the reconstructed excitation signal by applying LP synthesis filter. For more details about the AMR codec and its encoding/decoding procedures, readers are requested to refer to [30].

A possible way of generating a DC AMR audio file is depicted in Figure 1. An original audio waveform is encoded using AMR codec so that SC AMR audio recording is generated. If an attacker or fraudster aims at tampering or manipulating the SC AMR file, he/she first needs to decompress the SC AMR audio to obtain the pulse code modulation (PCM) audio waveform. Once the PCM audio waveform is obtained, any audio editing software can be used for tampering (e.g. inserting/deleting an audio segment into the waveform) or manipulating (e.g. changing the gender of the speaker or adding noise to the waveform to suppress the speech content) the audio file. Finally, he/she re-compresses the manipulated audio file with the AMR codec to masquerade the tampering/manipulation so that the DC AMR audio file is obtained. Thus, an original/authentic AMR audio is not necessarily DC unless a fraudster re-compresses it for the second time in order to masquerade the footprints of the tampering. Based on this fact, detection of DC AMR recording is an essential and important audio forensic application.

3 | AUDIO FEATURES

Spectrogram is an important visualisation tool to reveal the hidden information in an audio or speech file and widely used in speech processing applications [31]. It visualises the variation of the spectral content of the audio signal over time. It is obtained using short-term processing of an audio signal. In short-term processing, an audio signal is first divided into short overlapping frames consisting of N samples $x[n, t]$, where $n = 0, 1, \ldots, N-1$ is the sample index and $t = 1, 2, \ldots, T$ is the frame index. Each frame is then windowed using a data-tapering window (generally Hamming or Hanning window) $w[n]$. Finally, discrete Fourier transform (DFT) is taken to obtain the power spectrum of each frame:

$$X(k,t) = \left| \sum_{n=0}^{N-1} w[n]x[n,t]e^{-j2\pi nk/N} \right|^2 \quad (1)$$

where k is the discrete frequency index and the audio frame $x[n, t]$ is assumed to be zero outside of the interval $[0, N-1]$. The logarithmic power spectrum of all frames ($10 \log X(k, t)$) obtained by the short-term Fourier transform (STFT) is known as the spectrogram of the audio signal and visualised by a heat-map.

In order to understand the differences between SC and DC AMR audio files, we analyse and compare the spectrograms of the same recording compressed at different BRs in Figure 2. The columns of the figure correspond to BR values of 4.75, 6.7 and 12.2 kbps, respectively. The last row of the figure shows the differential spectrograms for each BR obtained by subtracting the SC and DC spectrograms in order to highlight the frequency regions most affected by double compression. From the figure it can be seen that considerable differences between SC and DC AMR audio files occur at high frequencies irrespective of the BR used for compression. SC AMR audio files have much higher energy variation at high frequencies (approximately above 2 kHz) than DC AMR files. This can easily be verified from the differential spectrogram images (the third row of the figure). While the formant patterns are still visible in the differential spectrograms, noise-like variations can be observed approximately above 2 kHz. Another interesting observation is that as the compression BR increases, larger energy variations occur at high frequencies for both SC and DC recordings. Similar observations hold for other BR values. However, one would argue that visualising the spectrogram of a single audio file may not be useful for making a general statement about acoustical changes caused by double compression. To this end, long-term average spectra (LTAS) is utilised. LTAS is widely used in audio forensics [32] and for computing the speech intelligibility index which measures the audibility of an audio signal [33]. LTAS is computed by time averaging the spectrogram over all frames:

$$LTAS = \frac{1}{T} \sum_{t=1}^{T} X(k,t). \quad (2)$$

We compute the average LTAS of SC and DC AMR audio files using 100 audio signals compressed at three different BRs. For each BR, the same 100 audio signals (the same speakers and the same contents) are used. Figure 3 displays the LTAS computed using SC and DC AMR audio files. Similar to the findings on spectrogram images, we observe that both SC and DC audio power show similar trends below 2 kHz independent of the compression BR. For $f > 2$ kHz, the differences between SC and DC audio files become larger. Although increasing the compression BR reduces the differences between SC and DC LTAS graphs, the power of the audio files increases in comparison to smaller BR values. These fundamental differences observed in the spectrogram images and the LTAS of the SC and DC AMR audio recordings make the spectrogram a potentially good representation to use with CNN for DC AMR audio detection.

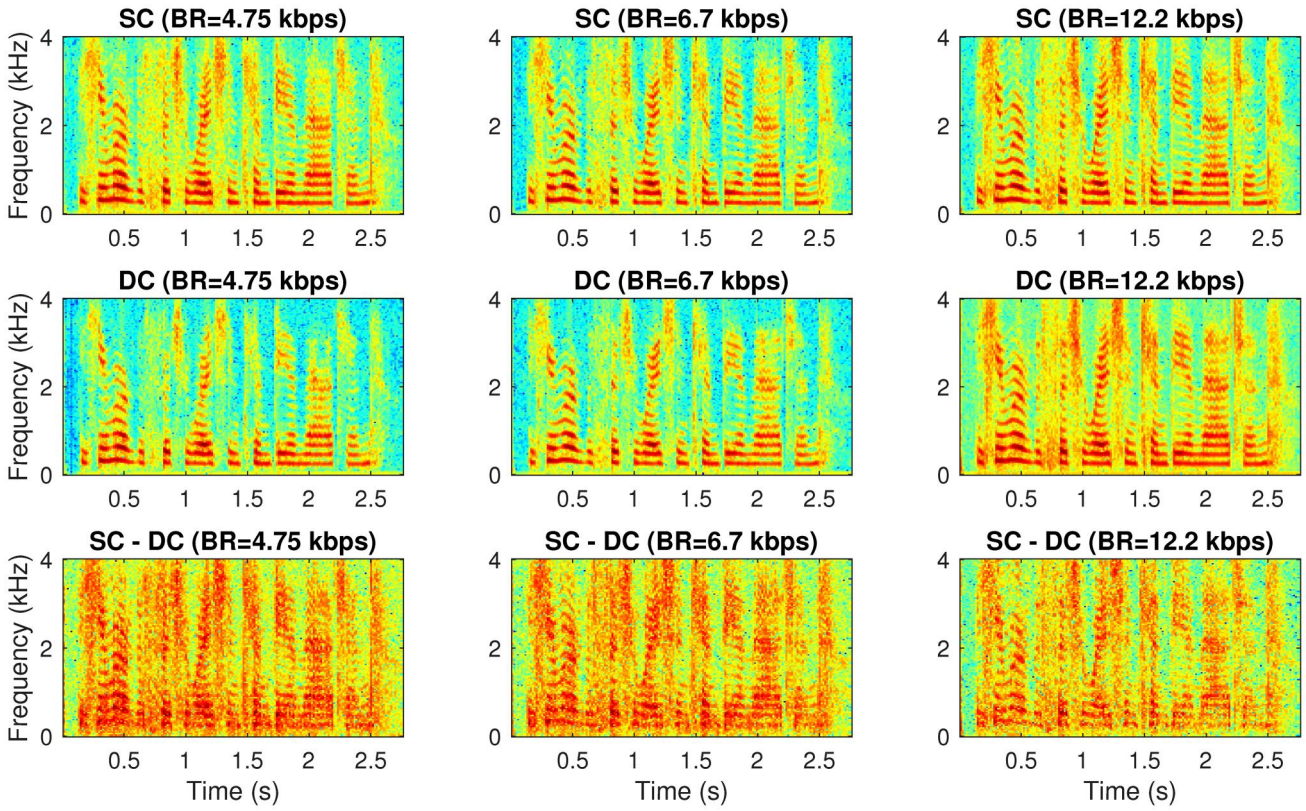

FIGURE 2 Spectrograms of the SC and DC AMR audio files compressed at 4.75 kbps (first column), 6.7 kbps (second column) and 12.2 (third column) kbps. The same audio file is used for each BR in order to avoid differences induced by either speaker variability or content of the audio. The last row corresponds to the differential spectrograms for each BR obtained by subtracting the SC and DC spectrograms in order to highlight the differences

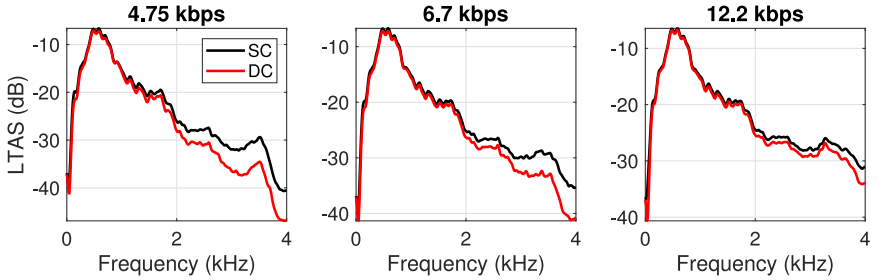

FIGURE 3 Comparison of the long-term average spectra (LTAS) for SC and DC AMR recordings

4 | CNN FOR DC AMR DETECTION

Given an audio signal, s, DC AMR audio detection can be performed as a hypothesis test aiming at deciding between two hypotheses:

- H_0 : s is a DC AMR audio
- H_1 : s is a SC AMR audio.

Optimum decision rule for deciding between H_0 and H_1 hypotheses which guarantees the minimum error rate is Bayes decision rule which relies on the posterior probabilities $P(H_0|s)$ and $P(H_1|s)$ and it is defined as:

$$Decision = \begin{cases} \text{Accept } H_0, & P(H_0|s) > P(H_1|s) \\ \text{Accept } H_1, & P(H_1|s) > P(H_0|s) \end{cases}. \quad (3)$$

Therefore, the fundamental task is to choose a reliable technique to compute the posterior probabilities ($P(H_0|s)$ and $P(H_1|s)$) for a given audio signal s. This optimum decision rule reduces to computation of the likelihood functions ($p(s|H_0)$ and $p(s|H_1)$) by following Bayes' rule. Statistical pattern recognition methods aim at estimating the likelihood functions using the training data of each class (SC and DC AMR audio classes). To this end, first, a suitable feature extraction method which is highly capable of discriminating the SC and DC audio files needs to be determined. Then, an effective statistical modelling technique is required to estimate the likelihood functions. Obviously selecting these two most important components (feature extraction and modelling techniques) is the heart of the DC audio detection problem and is a challenging task. To this end, our motivation is based on the fact that deep learning approach can be used for both purposes (feature extraction and determining the posterior probabilities)

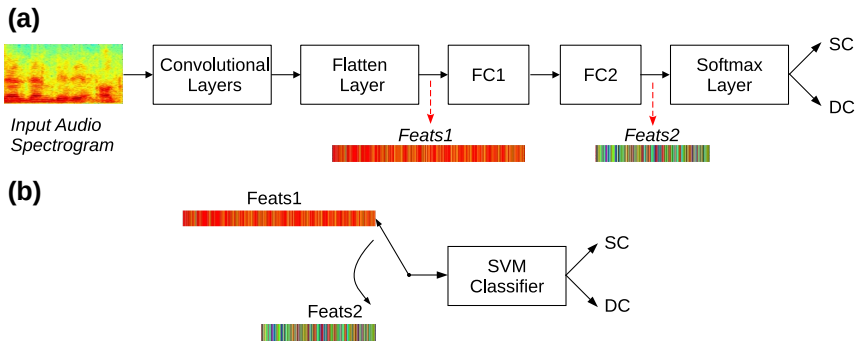

FIGURE 4 The proposed DC AMR audio detection system. (a) The proposed end-to-end DC AMR detection system and (b) SVM system using deep features extracted from the CNN. The input of the CNN is the spectrogram image of the audio signal

TABLE 1 Details of the CNN architecture used in the experiments and the parameters of each layer

Type	Filter size	Output shape	Number of parameters
Input		257 × 498	
Conv1	3 × 3	255 × 496 × 32	320
Batch Normalisation	-	255 × 496 × 32	128
Max Pooling1	2 × 2	127 × 248 × 32	-
Dropout1	0.25	127 × 248 × 32	-
Conv2	3 × 3	125 × 246 × 32	9248
Max Pooling2	2 × 2	62 × 123 × 32	-
Dropout2	0.25	62 × 123 × 32	-
Conv3	3 × 3	60 × 121 × 32	9248
Max Pooling3	2 × 2	30 × 60 × 32	-
Dropout3	0.25	30 × 60 × 32	-
Conv4	3 × 3	28 × 58 × 32	9248
Max Pooling4	2 × 2	14 × 29 × 32	-
Dropout4	0.25	14 × 29 × 32	-
Flatten	-	12992	-
FC1	-	512	6652416
Dropout4	0.5	512	-
FC2	-	256	131328
Dropout5	0.5	256	-
Softmax	-	2	514
Total	-	-	6812450

without any prior assumption rather than seeking for a suitable feature extraction technique to extract hand-crafted features and statistical modelling method for estimating the likelihood functions for each class.

With the recent developments in machine learning, CNN have become state-of-the-art technique for various pattern recognition tasks. CNN based techniques are generally used either for feature extraction or classification. For feature extraction, usually raw data or a simple two-dimensional representation of the data is applied to the CNN and the output of the intermediate layers are extracted as the feature representation of the input data. For end-to-end classification in turn, CNN receives the input data and returns the predicted class label of the input. CNN is a powerful tool for both cases since it is capable of learning the discriminative features automatically without any prior knowledge about the task. Inspired from this, the authors used CNN for DC AMR audio detection as an *end-to-end classifier* and as a *feature extractor* as shown in Figure 4. The detailed CNN architecture used for DC AMR audio detection is described in Table 1. The end-to-end DC AMR audio detector system receives the input audio spectrogram and returns the predicted class label (SC or DC audio) of the input audio signal. As a feature extractor in turn (Figure 4(b)), we use CNN for extracting deep feature representation of the input audio. Given an input audio spectrogram, two different embeddings are extracted from the

intermediate layers of the CNN. The embeddings are extracted from (i) the output of the flatten layer (**Feats1**) and (ii) the output of the last fully connected layer (**Feats2**). These two deep feature representations are then modelled using SVM classifier [29]. Hence, using CNN for two different purposed will help us to reveal the power of the CNN for both tasks (feature extraction or end-to-end classification) and compare their performances on DC AMR audio detection.

To use CNN for DC AMR audio detection problem, the input signal should be organised as a two-dimensional array. Since audio signals are one-dimensional arrays, an appropriate two-dimensional representation of the signal should be chosen with respect to constraints of the problem. Thus, spectrogram representation of an audio signal is a good candidate for this purpose and it has been used in various speech processing applications utilising CNNs such as speaker recognition [27] and speech recognition [28]. However, as described in the preceding section, the spectrogram is computed from the windowed frames of audio signal. Thus, if we treat the spectrogram as an image, then the rows can correspond to the discrete frequency index and the columns to the frame index. Obviously, the total number of frames obtained from audio signals of different duration would naturally be different. Therefore, as the duration of the audio signal changes, the number of columns of the spectrogram image will also change. In order to unify the time-frequency shape of the spectrogram, usually the total number of the frames of an audio signal is fixed to an upper bound [34, 35]. This is accomplished by either truncating the spectrogram along the time axis (axis corresponds to the frames) with a fixed size or concatenating the frames of the short audio files which has less number of frames than the upper bound in order to match the frame numbers.

5 | EXPERIMENTAL SETUP

5.1 | Database and spectrogram computation

DC AMR audio detection experiments are conducted on four different databases, namely TIMIT [36], *Multicodec Invdec Tampering Dataset* (MITD) [37], *MIT Mobile Device Speaker Verification Corpus* (MDSVC) [38] and a Turkish audio database obtained from freely available VoxForge open speech dataset[1]. These datasets are selected to investigate the performance of the proposed DC AMR detection system under various conditions. For example, TIMIT dataset consists of clean audio recordings collected from 630 speakers under controlled conditions without any channel or environment variation. Therefore, we use TIMIT database for preliminary experiments and analysis, in general. MITD dataset contains 39 conversational audio recordings between two speakers recorded using Samsung Galaxy S4 mobile phone [37]. However, 38 out of the 39 recordings were generated from one original recording. The original recording was encoded and decoded using four different codecs (MP3, AAC, HE-AAC and mp3PRO) with various BRs. Therefore, MITD database includes previously encoded and decoded audio recordings. MDSVC database in turn, contains audio recordings from 48 speakers collected using two different microphones at three different locations [38]. With this we aim at investigating the DC AMR detection performance of different microphones and various locations. Finally, VoxForge database consists of speech recordings from 90 speakers collected under uncontrolled conditions. Because VoxForge is an open speech database and participants record their audio using their own setup (generally using their microphone equipped personal computer at their home). Thus, VoxForge dataset includes microphone and environment variations as well as noisy recordings. To sum up, using these four different audio datasets will help us to investigate the DC AMR audio detection performance from different aspects. As an example, Figure 5 shows the distributions of estimated signal-to-noise-ratios (SNR) of the speech recordings in each database[2]. From the figure, we can see that the majority of the speech recordings in TIMIT and MITD datasets have relatively high SNR level. Although there exists recordings with SNR levels below 20 dB in TIMIT database, most of the recordings have SNR value exceeding 40 dB whereas recordings in MITD dataset have SNR values above 52 dB. In contrast to TIMIT and MITD databases, the VoxForge dataset consists of recordings with very high (SNR > 70 dB) and very low (SNR < 0 dB) SNR levels. However the most of the recordings have SNR values within the range [0, 50] dB. Finally, the majority of the recordings in the MDSVC corpus has the lowest SNR values (SNR values between 0 and 30 dB) in comparison to other three datasets. Thus, each dataset consists of recordings with different noise levels and it will be interesting to investigate their behaviours on DC AMR audio detection.

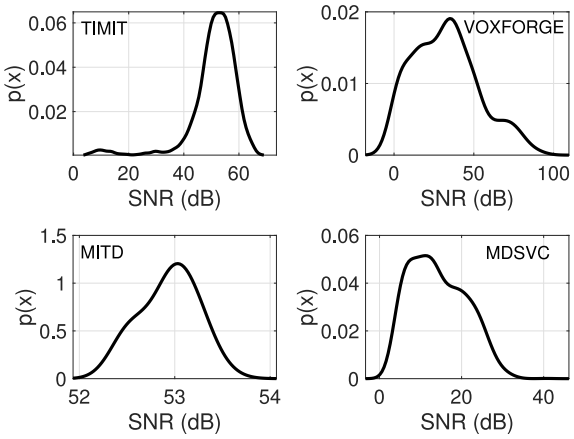

FIGURE 5 Distributions of estimated SNR values for each dataset used for DC AMR detection experiments

[1] http://www.voxforge.org/

[2] SNREval toolkit from https://labrosa.ee.columbia.edu/projects/snreval/ is used to estimate the SNR levels.

Audio recordings duration of 1 s are used in the experiments for all datasets except for the preliminary experiments. In the preliminary experiments and analyses conducted on TIMIT dataset, 5 s long audio recordings are used. If an audio signal is shorter than 5 s, its content is extended to 5 s by copying the samples. For TIMIT database, a total of 6000 1 s long audio signals are used. If the duration of the signal is longer than 1 s, it is cropped to obtain 1 s long audio clips. Similarly, a total of 5185 audio signals of duration 1 s are selected from MDSVC dataset. For VoxForge and MITD datasets in turn, the total number of audio signals are considerably less than TIMIT and MDSVC datasets. However the durations of the signals are much longer. Therefore each audio signal in VoxForge and MITD datasets are divided into 1 s long clips. This gives a total of 21099 and 4405 audio recordings for MITD and VoxForge databases, respectively.

Each audio signal from each dataset is compressed using the AMR codec [11, 30] with the randomly selected compression BR (the first compression bit-rate -BR1) varying from 4.75 to 12.2 kbps to obtain the SC audio clips (the same number of SC audio files is generated per each BR). Then we decode the SC AMR audio files and re-compress back using a randomly selected BR (the second compression bit-rate -BR2) to generate the DC AMR audio files. The number of DC AMR audio recordings for each BR1-BR2 combination is the same since we select the compression BR value from a uniform distribution. For each dataset, approximately 25% of the audio recordings are used to train the CNN and the remaining audio files are used for testing. The training set includes the same number of SC and DC AMR audio recordings in order to prevent class imbalance problem. We would like to note that there is no speaker or audio recording overlap between training and test sets for all datasets except MITD database. Because MITD database is composed from 39 single-channel conversational audio recordings between two speakers and the same speakers appear in each recording. In TIMIT database in turn, while each speaker has ten recordings, two of the utterances (SA1 and SA2 sentences) are spoken by all speakers. Therefore, SA1 and SA2 utterances were excluded from the training set in order to avoid our detection system to be biased towards these utterances. This implies that while the training set includes eight recordings per speaker, there are ten audio clips for each speaker in the test set for the TIMIT dataset. Similarly, mutually exclusive training and test sets were used in the experiments for the VoxForge and MDSVC datasets.

Next, the spectrograms are computed from the audio recordings. To this end, each audio signal is divided into 25 ms frames using 10 ms frame shift. Each pre-emphasised audio frame is then windowed using a Hamming window of 200 samples. The power spectrum of each windowed frame is then computed using 512-point DFT. Due to the symmetry property, only the first 257 samples of the power spectra are retained. For a 1 s long audio recording sampled at 8 kHz, this yields a spectrogram image of 257 × 100 because 10 ms frame shift gives a frame-rate of 100 frames per second.

5.2 | CNN and SVM systems for DC AMR audio detection

The CNN architecture used in the experiments consists of four convolutional layers where each convolutional layer is followed by a Max Pooling layer and a Dropout layer in order to avoid overfitting while training the network. Similarly, a batch normalisation layer is used after the first convolutional layer to avoid overfitting. The flatten layer converts the last convolutional layer output into a vector thus it can be fed into the fully connected layers. The output layer of the CNN architecture is a softmax layer with two nodes where the first output unit corresponds to the SC AMR class and the second output node corresponds to the DC AMR class. Rectified linear unit (ReLu) nonlinearity is used for the convolutional layers, whereas sigmoid activation function is used in fully connected layers. Adadelta optimization algorithm is used for optimising the network. Dropout rate of 0.25 is chosen for all convolutional layers whereas 0.5 dropout rate is used in fully connected layers. All these parameters were optimised according to our initial experiments using various numbers of layers, kernel sizes, Dropout rates, optimisers, activation functions and the number of fully connected layers with different number of units. Based on such a thorough initial analysis and experiments, the network summarised in Table 1 was found to yield the best DC AMR audio detection performance. For each dataset, a development set consisting of 500 SC and 500 DC audio recordings which are not included in the training or test sets are used during training to tune the network hyperparameters and to determine the threshold for early stopping.

In the SVM experiments using two different deep feature representations extracted from the CNN architecture (namely, Feats1 and Feats2 as described in Section 4), three different kernel functions (linear, radial basis function (RBF) and sigmoid kernels) are used to compare their performances on DC AMR audio detection. The LIBSVM package [39] is used for SVM training and testing.

5.3 | Performance criterion

Since we aim at distinguishing DC AMR audio from SC AMR recordings, DC AMR audio class is treated as positive class, and SC AMR audio class is treated as negative class. Therefore we measure the performance of the systems in terms of true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Then the detection accuracy is defined as

$$Accuracy[\%] = \frac{TP + TN}{TP + FP + TN + FN} \times 100. \quad (4)$$

In the experiments, we report TN and TP rates for SC AMR audio detection and DC AMR audio recognition, respectively. The TN rate (also known as the specificity) is the probability (a *posteriori*) that a SC AMR audio trial is determined to belong to SC AMR audio class and it is computed by:

		Test BR (kbps)								
		4.75	**5.15**	**5.9**	**6.7**	**7.4**	**7.95**	**10.2**	**12.2**	**Avg.**
Training BR (kbps)	4.75	99.80	99.83	99.80	98.78	97.05	97.23	94.90	89.75	97.14
	5.15	98.93	99.88	99.85	99.86	99.80	99.90	99.71	99.02	99.61
	5.9	98.81	99.71	99.90	99.72	99.42	99.76	99.60	96.91	99.22
	6.7	98.95	99.95	**99.98**	**99.96**	**99.96**	**100**	99.90	99.62	99.79
	7.4	99.82	99.93	99.95	99.96	99.93	99.91	99.71	98.12	99.66
	7.95	98.63	99.85	99.95	99.88	99.90	99.98	99.85	99.95	99.74
	10.2	**99.93**	**99.97**	99.98	99.91	99.92	99.95	**99.98**	99.97	**99.95**
	12.2	99.65	99.86	99.91	99.91	99.95	99.98	99.97	**99.98**	99.90
Pooled		100	100	100	100	100	100	100	100	100

TABLE 2 DC AMR audio recognition rates (%) obtained using end-to-end CNN system. The last row shows the detection rates obtained when a single model is trained by pooling all training data of each BR. The last column shows the average detection rates for each training BR value averaged over all test BR values. The best numbers for each test BR are shown in boldface and the second best numbers are underlined in the table for the ease of comparison

$$TN[\%] = \frac{TN}{TN + FP} \times 100 \tag{5}$$

Hence, higher the value of TN rate, the less likely our system produces FP results. Similarly, the TP rate (also known as the sensitivity) is the probability that a DC AMR audio trial is decided to be a DC AMR audio by the system and it is computed by:

$$TP[\%] = \frac{TP}{TP + FN} \times 100 \tag{6}$$

6 | EXPERIMENTAL RESULTS

6.1 | Preliminary results and analyses

As previously mentioned, TIMIT database is used for DC AMR audio detection for preliminary experiments and analysing the system behaviour. To this end, we first use 5 s long audio recordings from TIMIT database. Thus, 257 × 498 spectrogram images obtained from each audio clip of duration 5 s are applied to the input of the CNN. A total of 6000 SC and 6000 DC AMR audio recordings per each BR value are used in these preliminary experiments. DC AMR audio files are generated using the same first and second compression BRs (BR1 and BR2, respectively). A different CNN model was trained (eight different CNN models were trained in total) for each compression BR using the 25% of the audio recordings. The detection accuracy values obtained using end-to-end CNN system are summarised in Table 2. In the table, besides reporting the results when training and test BRs are the same (diagonal entries of the table), the results obtained when there is a mismatch between training and test BR values are also given (off-diagonal entries of the table). Using a different compression BR values to test the system will give some insight into DC AMR audio detection performance of the proposed system in the face of previously unseen compression BR value. The average detection rates of each row are given in the last column of the table. From the table, the end-to-end CNN system shows great performance on DC AMR audio detection task for all cases irrespective of the compression BR. Detection rates are above 99% for most cases. In general, a slightly higher detection accuracy is obtained for high BRs than low BRs. When there is a mismatch between training and test recordings in terms of the compression BR values (the off diagonal elements of the table), the recognition accuracy decreases as the gap between training and test BRs increases especially when the system is trained using audio files compressed at a low BR. As seen from the last column of the table, the end-to-end system yields average accuracy values higher than 99% except for the case of 4.75 kbps. When the system is trained using audio files compressed at 4.75 kbps, an average detection rate of 97.14% is obtained which is much lower than other training BR values.

One would argue that training a different model for each compression BR value is not a reasonable approach unless prior information regarding the compression BR value of the test signal is provided. Thus, a single system that is capable of recognising audio signals compressed at any BR would be more preferable. To this end, we train a single CNN system using all the training data consisting of SC and DC AMR audio files compressed at eight different BRs. The results are summarised in the last row of the Table 2. As expected, pooling all the training data for all BRs and training a single model yields great detection accuracy independent of the compression BR value. The only drawback of such system is the fact that, since the amount of training data is eight times larger than the case of training one model for each BR separately, training time is considerably increased. However, since training the system is performed off-line in contrast to the test stage it can be acceptable.

The number of misidentified trials per each test BR value for the end-to-end CNN system is given in Figure 6. From the figure, as the compression BR increases, the number of misclassified trials generally reduces. The CNN detection system correctly detects all DC AMR audio files for four BRs (5.15, 6.7, 7.95 and 12.2 kbps). In the case of 5.9 kbps, the system fails to detect only two DC AMR trials whereas for 10.2 kbps, the system misclassifies only one DC AMR audio trial.

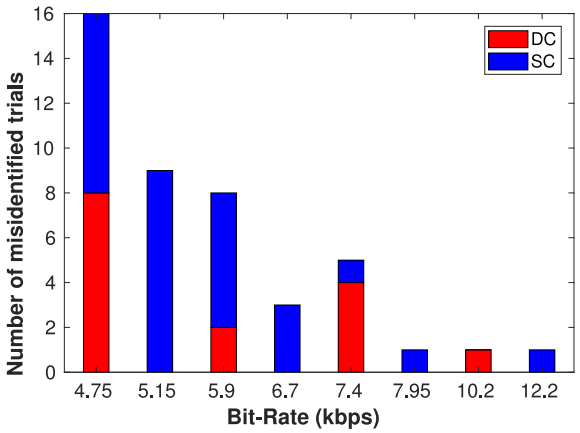

FIGURE 6 Number of misidentified trials per each test BR for end-to-end CNN system

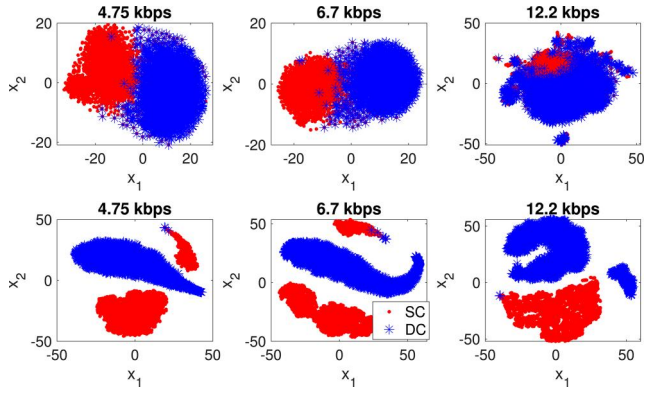

FIGURE 7 Scatter plot of the Feats1 (the first row) and Feats2 (the second row) features for SC and DC audio files compressed at 4.75 kbps (first column), 6.7 kbps (second column) and 12.2 kbps (third column)

After the preliminary experiments and observing the great success of the end-to-end CNN system, we analyse the output of the bottleneck layers of the CNN architecture in order to gain a better understanding the system's behaviour. To this end, we extract two different embeddings from the intermediate layers of the CNN model, namely Feats1 and Feats2 as described in Section 4. Although, these two embeddings are later used as the feature vectors with SVM classifier for DC AMR audio detection, we first analyse the two representations. We applied t-distributed stochastic neighbour embedding (t-SNE) non-linear dimensionality reduction method [40] to the feature vectors in order to embed high dimensional features into two-dimensional space for visualisation. Figure 7, shows the scatter plot of the Feats1 (the upper row of the figure) and Feats2 (the lower row of the figure) feature vectors extracted using SC and DC AMR audio files compressed at 4.75, 6.7 and 12.2 kbps. All 6000 recordings from TIMIT database are used to generate the scatter plots. The scatter plots in the figure show that the two audio classes (SC and DC audio classes) are mostly well separated for both feature representations (Feats1 and Feats2). The two classes overlap when the deep feature vectors are extracted from the flatten layer (Feats1) using the audio signals compressed at 12.2 kbps. The contradiction between the two deep embeddings is possibly because of the fact that for Feats1 we reduced 12992 dimensional data into two-dimensional space whereas in the latter case (Feats2) 256 dimensional feature vectors were reduced onto the two-dimensional space.

In the last set of preliminary experiments on 5 s long audio recordings using TIMIT dataset, we use deep feature vectors obtained from the CNN model with SVM classifier. Results obtained with SVM classifier using three different kernel functions are given in Table 3. The last column of the table shows the average detection rate averaged over all BRs for each kernel function. SVM classifier using deep features yields great detection performance irrespective of the BR used to compress test recordings. Linear kernel function is superior to RBF and Sigmoid kernels in most cases. This observation is expected because in Figure 7, it is shown that the two classes are almost linearly separable even in the two-dimensional space in most cases. When comparing the two deep feature representations (Feats1 and Feats2), Feats1 gives better recognition rates than 256 dimensional fully connected layer features (Feats2) in general. This is possibly because of the fact that SVM classifier usually finds the optimum separating hyperplane easily and therefore performs better with high dimensional feature vectors [41]. Comparing the SVM results reported in Table 3 and end-to-end CNN system results (Table 2) we can observe that SVM classifier shows slightly better performance than the CNN system. The reason of the superior performance of the SVM classifier might be because CNN system actually uses the simple softmax classifier using the 256 dimensional features extracted from the last hidden layer (Feats 2). However, SVM classifier finds the separating hyperplane by projecting the feature vectors onto the higher dimensional space using a kernel function so that the two classes are easier to separate with a hyperplane in that higher dimensional kernel space. Since linear kernel function with Feats1 (the embeddings extracted from the flattening layer of the CNN) is superior to other kernel functions and Feats2 in general, Feats1 feature vectors with linear kernel configuration will be used in the remaining SVM experiments.

6.2 | Results on one-second-long TIMIT recordings

In the preliminary experiments and analyses reported in the previous section, we showed that both end-to-end CNN system and SVM classifier with deep features extracted from the bottleneck layer of the CNN yield great performance on DC AMR audio detection. However, in the preliminary experiments 5 s long audio recordings were used and a different model for each compression BR was trained. First, 5 s long audio recordings are unnecessarily long and reasonable detection rates were obtained using only 1 s long audio files in the previous studies addressing the DC AMR audio detection [22, 23]. Second, training a separate model for each compression BR is unusual and using a single model to detect DC AMR audio irrespective of the compression BR is more desirable.

		Bit-Rrte (kbps)								
		4.75	**5.15**	**5.9**	**6.7**	**7.4**	**7.95**	**10.2**	**12.2**	**Avg.**
Feats 1	Linear	**100**	**99.98**	**99.92**	**99.98**	99.95	**100**	**100**	**100**	**99.98**
	RBF	97.25	99.97	99.92	99.98	**99.96**	100	99.92	100	99.62
	Sigmoid	93.80	99.98	99.92	99.98	99.95	100	100	100	99.20
Feats 2	Linear	**99.80**	**99.95**	**99.91**	99.97	**99.95**	**99.98**	99.98	**100**	99.94
	RBF	99.78	99.91	99.90	**100**	99.93	99.98	**100**	100	99.94
	Sigmoid	99.80	99.88	99.90	99.98	99.93	99.98	100	100	99.93

TABLE 3 DC AMR audio detection rates (%) using SVM classifier and deep features extracted from CNN. Feats1 corresponds to the feature vector extracted from the flattening layer and Feats2 is the feature vector extracted from the last fully connected layer. The best numbers obtained for each BR and features are shown in boldface and the globally best numbers are shown in boldface and underlined

Therefore, in the remaining experiments we use 1 s long audio clips and train a single model for detection task. We would like to note that while training a single model we do not increase the number of DC audio files by eight times in the training set. The same number of SC and DC audio files is used in the training set and the set of DC AMR audio files in the training set includes the same number of files for each compression BR. Since using 1 s long audio clips (a spectrogram image of size 257 × 100) reduces the number of input parameters five times in comparison to 5 s long audio recordings (spectrogram image of size 257 × 498), we use a CNN with three convolutional layer followed by two fully-connected layers each consisting of 512 units in order to avoid overfitting due to smaller input data. All the parameters (kernel sizes, activation functions etc.) are the same with the model shown in Table. 1. This section reports the detection results obtained using 1 s long audio clips from TIMIT database.

Average SC and DC AMR audio detection rates obtained using 1 s long audio clips from TIMIT database are given in Table 4. For SC audio detection, the detection rates obtained for eight compression BR values (TN rates) are averaged in the table. For the DC AMR audio recordings in turn, each possible combination of the first and second compression BRs (BR1 and BR2, respectively) are considered. This leads to 64 possible first and second compression BR configurations ($8 \times 8 = 64$). The average DC audio detection rates reported in Table 4 are computed by averaging these 64 detection rate values (TP rates). The rationale behind reporting the average detection rates rather than providing the detection rate of each possible BR1 and BR2 configuration individually is the fact that both CNN and SVM systems achieve great performance (detection rate of 100%) for the vast majority of the cases. Therefore, we report average detection rates in order to avoid confusion. As seen from the Table 4, both CNN and SVM systems yield 100% average SC AMR audio detection rate which implies that both systems achieve 100% TN rate for all BRs. Thus, end-to-end CNN and SVM system with deep features are both very powerful to detect SC AMR audio recordings. For the case of DC AMR audio detection, although SVM system slightly outperforms the end-to-end CNN system in terms of average detection rate, the performance difference (99.92% vs. 99.94%) between two systems is negligible. Only four cases are found to yield detection rate lower than 100% with end-to-end CNN system. Interestingly, the BR1 value in all of these four cases is found to be 4.75 kbps. The lowest DC AMR audio detection rate (96.40%) is obtained using end-to-end system when both BR1 and BR2 are 4.75 kbps. SVM system using deep features in turn, gives 100% detection rate for 62 cases among the 64 possible BR1 and BR2 combinations. Similar to the results obtained with the end-to-end CNN system, when BR1 and BR2 values are the same and 4.75 kbps, the lowest DC AMR detection rate (97%) is obtained with SVM system. In summary, both end-to-end CNN and SVM systems show great performance even for 1 s long audio recordings from TIMIT database are used.

TABLE 4 Average SC and DC AMR audio detection rates (%) obtained by CNN and SVM systems using 1 s long audio recordings from TIMIT database

	Audio type	
System	**SC**	**DC**
CNN	100	99.92
SVM	100	99.94

TABLE 5 SC AMR audio detection rates (TN rates in %) on MITD dataset using end-to-end CNN and SVM systems

	Compression BR (kbps)							
System	**4.75**	**5.15**	**5.9**	**6.7**	**7.4**	**7.95**	**10.2**	**12.2**
CNN	**93.39**	**94.71**	**94.86**	**96.82**	**95.79**	**96.28**	**98.87**	**98.67**
SVM	85.37	88.60	88.60	91.05	89.58	91.68	96.82	95.94

6.3 | Results on MITD database

Next, the DC AMR audio detection experiments are conducted on MITD dataset [37]. As explained in Section 5.1, MITD dataset comprises audio recordings that were previously encoded and decoded using different audio codecs (MP3, AAC, HE-ACC and mp3PRO) with various BRs [37]. Therefore, the experiments using MITD database will help us to investigate the effect of previous audio manipulations on DC AMR audio detection. Each recording is first divided into 1 s long audio clips and then SC and DC AMR audio files were generated using the same procedure described in Section 5.1. SC AMR audio detection rates (TN rates) obtained using end-to-end CNN system and SVM classifier with deep features on MITD database are shown in Table 5. As the compression BR

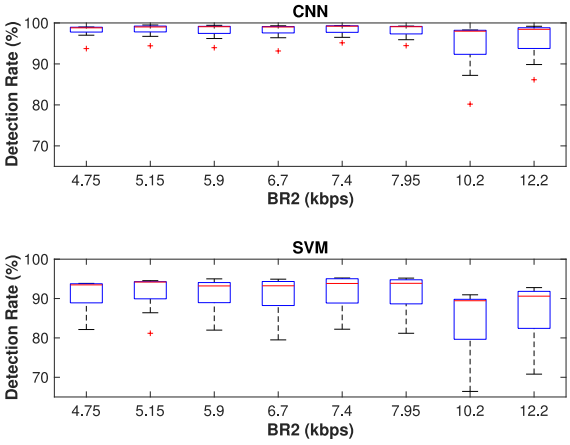

FIGURE 8 Box plots of DC AMR audio detection rates obtained using CNN and SVM classifiers for each second compression bit-rate (BR2) value. The height of the each box represents the variations of the eight detection rates for the first compression bit-rate (BR1) values ranging from 4.75 to 12.2 kbps.

increases, both systems yield better detection rate in general. For example, while a 93.99% SC AMR audio detection rate is obtained for 4.75 kbps using end-to-end system, it improves to 98.67% when the compression BR is 12.2 kbps. The CNN system is superior to the SVM classifier on SC AMR audio detection task independent of the compression BR.

Figure 8 compares the box plots of the DC AMR audio detection rates (TP rates) per each BR2 value for end-to-end CNN and SVM systems. The height of the each box is determined by the variations of the DC AMR detection rates obtained for eight different BR1 values ranging from 4.75 kbps to 12.2 kbps for a fixed BR2 value. While red lines in each box represents the sample median of the eight detection rates, the top and bottom edges of a box correspond to the 75th and 25th percentiles of the detection rates, respectively. From the figure, we can observe that end-to-end CNN system systematically outperforms the SVM classifier with deep features for all BR2 values. Interestingly, for both systems the detection rates show similar trends with small variations as BR2 increases from 4.75 up to 10.2 kbps. However, for 10.2 and 12.2 kbps, larger variation is observed on the detection rates. It is found that as BR2 reaches to 10.2 kbps, detection accuracy drops drastically independent of the value of BR1. When BR1 is 10.2 or 12.2 kbps, the relative performance reduction is much larger than the other BR1 values. For instance, when BR1 is chosen as 10.2 kbps, the detection rate obtained by the end-to-end CNN system drops from 94.46% to 80.19% as BR2 increases from 7.95 to 10.2 kbps. However, when BR1 is 4.75 kbps, 99.19% and 98.16% detection rates are obtained for 7.95 and 10.2 kbps BR2 values, respectively. Similar to end-to-end CNN system, when the BR1 value is lower than 10.2 kbps, better recognition rates are obtained using SVM classifier independent of the BR2 value. Again, when BR2 reaches to 10.2 kbps, detection rates considerably degrade as in the case for end-to-end CNN system.

In [23], SAE+GMM system were found to give 92.30% average detection rate on MITD dataset. However, our proposed end-to-end CNN system achieves 97.41% average recognition rate which considerably outperforms the SAE +GMM system.

TABLE 6 SC AMR audio recognition rates (TN rates in %) on MDSVC dataset obtained using end-to-end CNN system

Compression BR (kbps)							
4.75	5.15	5.9	6.7	7.4	7.95	10.2	12.2
99.58	99.78	99.77	99.33	98.94	100	99.77	100

TABLE 7 DC AMR audio recognition rates (TP rates in %) on MDSVC dataset obtained using end-to-end CNN system

		BR2 (kbps)							
		4.75	**5.15**	**5.9**	**6.7**	**7.4**	**7.95**	**10.2**	**12.2**
BR1 (kbps)	4.75	94.76	97.07	99.79	98.53	99.16	99.16	100	99.79
	5.15	100	100	100	99.78	99.78	99.78	100	100
	5.9	100	100	100	100	100	100	100	99.77
	6.7	100	100	100	100	100	100	100	100
	7.4	100	100	100	100	100	100	100	100
	7.95	100	100	99.78	100	100	100	99.78	100
	10.2	100	99.54	99.32	99.09	99.32	99.54	99.77	100
	12.2	100	98.96	98.34	100	100	100	99.79	100

6.4 | Results on MDSVC dataset

In order to analyse the effect of recording device (channel), recording environment and phrases spoken by the speakers on DC AMR audio detection, MDSVC dataset is now used in the experiments. As described in Section 5.1, MDSVC dataset contains audio recordings from 48 speakers recorded using two different microphones (headset and internal) at three locations (hallway, intersection and office) [38]. Recordings were carried out in two different sessions and database contains a total of 67 different phrases. Thus, MDSVC database is a good candidate for investigating the effects of different parameters such as channel, recording environment and spoken utterance (phrase) on DC AMR audio detection. However, we first report the results without taking these parameters into account to investigate the general performance of the dataset. Tables 6 and 7 summarise the SC and DC AMR audio detection rates obtained using end-to-end CNN system, respectively. From the SC AMR detection results given in Table 6, we observe that for seven compression BRs the detection rate is above 99%. The only exception is 7.4 kbps which gives a detection rate of 98.94%. For DC AMR audio detection case in turn (Table 7), a detection rate below 99% is obtained only for 5 cases out of 64. End-to-end system gives 100% detection rate for most cases. The lowest detection rate (94.76%) is obtained when the audio

TABLE 8 Average SC and DC AMR audio detection rates (%) for different microphone types in MDSVC dataset

Audio Type	Microphone type	
	Headset	Internal
SC	99.89	99.40
DC	99.71	99.67

signals are compressed at 4.75 kbps in the first and second compression stages. Similar observations hold for the SVM classifier with deep features. For example, average DC AMR audio detection rates of 99.69% and 99.81% are obtained using end-to-end and SVM systems, respectively. Thus, both systems show similar DC audio detection performance on average with negligible difference.

Next, we analyse the effect of recording device (channel) on the detection performance and the results are given in Table 8. The detection rates summarised in the table are calculated by averaging the accuracies for all compression bit-rates. From the table, we observe that the detection rates obtained using the audio clips originally recorded using headset microphones are slightly better than internal microphone recordings. However, the performance gap between headset and internal microphones is insignificant. This indicates that the proposed DC AMR audio detection system is robust against channel variations.

The average SC and DC AMR audio detection rates obtained using the audio files recorded at three different locations are shown in Figure 9. The detection rates do not vary considerably as the recording environment changes. Interestingly, for hallway and office recordings, SC detection rates are slightly better than DC AMR audio detection rates. For example, while SC audio files recorded at hallway give 99.91% accuracy, DC AMR audio detection rate for the same location is 99.78%. However, for the intersection location, 99.02% and 99.61% average detection rates are obtained for SC and DC files, respectively.

Finally, we investigate whether SC and DC AMR audio detection rates depend on the phrase spoken during the recording. MDSVC database consists of the audio files where 67 different phrases were spoken by the speakers. Figure 10 shows the average SC and DC AMR audio detection rates for each phrase. We observe that both SC and DC AMR audio detection rates are above 99% for most phrases. Only seven phrases yield SC detection rate below 99%. Similarly, DC detection rate is below 99% for 5 phrases. The lowest SC AMR detection rate of 95.83% is obtained for the phrase "Eugene Weinstein". For the DC case in turn, the phrase "Mitchell Peabody" yields the lowest detection rate of 96.24%.

6.5 | Results on VoxForge dataset

The last database used in the DC AMR audio detection experiments is the VoxForge dataset. SC AMR detection rates obtained on VoxForge database using end-to-end CNN and

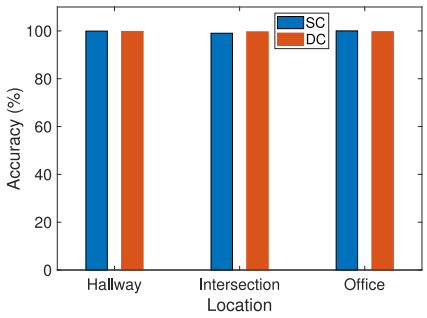

FIGURE 9 SC and DC AMR audio detection rates for different locations

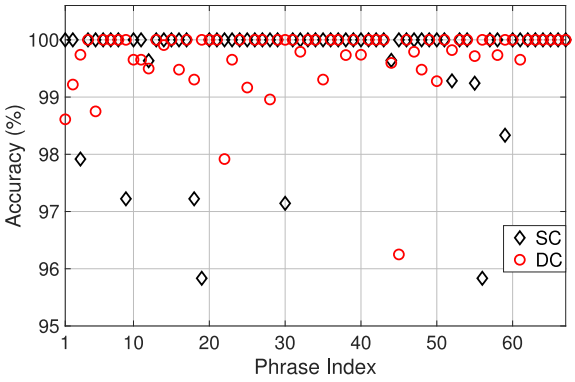

FIGURE 10 SC and DC AMR audio detection rates for each phrase in MDSVC dataset

TABLE 9 SC AMR audio detection rates (TN in %) on VoxForge dataset using end-to-end CNN and SVM classifiers. The best number for each BR is given in boldface

	Compression BR (kbps)							
	4.75	**5.15**	**5.9**	**6.7**	**7.4**	**7.95**	**10.2**	**12.2**
CNN	95.09	94.76	91.13	91.40	95.26	99.50	96.48	97.87
SVM	**95.46**	**96.59**	**94.47**	**94.98**	**97.04**	**99.74**	**97.13**	**98.78**

SVM systems are given in Table 9. When audio files are compressed at 5.9 kbps, both systems yield the smallest SC detection rates (TN rates). The best SC audio detection rates are obtained when audio recordings are compressed at 7.95 kbps. SVM classifier with deep features is superior to end-to-end system for SC AMR audio detection on VoxForge dataset. The DC AMR audio detection rates for both systems are shown in Table 10. From the table we observe that end-to-end CNN system outperforms SVM classifier in most cases. Interestingly, SVM system yields slightly better detection rates only for three cases and the second compression bit-rate BR2 is found to be 4.75 kbps in all these cases. The highest detection rate is obtained when BR1 is 7.4 kbps and the BR2 is either 4.75, 7.4 or 7.95 kbps. When both BR1 and BR2 are very low, the minimum detection rate is obtained. For example, when the audio signals are first compressed at 4.75 kbps, 83.33% and 84.06% detection rates are obtained when BR2 is 4.75 kbps and 5.15 kbps, respectively.

TABLE 10 DC AMR audio recognition rates (TP in %) on VoxForge dataset using CNN and SVM classifiers. The best number for each BR2 is given in boldface

		BR2 (kbps)							
System	**BR1 (kbps)**	**4.75**	**5.15**	**5.9**	**6.7**	**7.4**	**7.95**	**10.2**	**12.2**
CNN	4.75	83.33	84.06	92.89	89.21	89.21	89.95	93.13	93.38
	5.15	98.80	95.00	96.19	95.47	95.47	95.00	96.19	95.71
	5.9	97.53	95.07	95.32	96.30	95.81	95.81	96.30	97.29
	6.7	99.26	97.54	**98.28**	99.26	99.26	99.26	98.28	99.26
	7.4	**99.76**	**98.57**	97.15	**99.52**	**99.76**	**99.76**	**98.34**	**99.76**
	7.95	99.26	95.32	96.05	99.26	99.01	99.26	97.04	99.01
	10.2	98.82	93.44	93.67	94.84	94.14	93.20	95.31	96.95
	12.2	98.10	85.81	87.47	97.63	98.10	97.39	90.78	97.87
SVM	4.75	79.09	82.11	90.17	88.66	88.91	88.41	92.69	92.69
	5.15	99.02	94.40	94.64	95.13	94.64	94.40	95.37	94.40
	5.9	97.73	93.21	94.22	95.97	94.47	95.72	94.97	96.73
	6.7	99.24	97.49	97.74	99.24	99.24	99.24	97.99	99.24
	7.4	99.73	97.53	96.05	99.50	99.50	99.50	97.53	99.50
	7.95	98.97	93.11	93.62	99.23	98.97	98.97	96.17	98.46
	10.2	99.04	91.88	91.40	93.31	93.31	92.60	93.79	95.94
	12.2	98.04	81.21	82.19	97.56	97.80	96.82	86.58	97.31

TABLE 11 Comparison of the average SC and DC AMR audio detection rates (%) for each dataset obtained using end-to-end CNN and SVM systems

	Audio	Database			
System	**Type**	**TIMIT**	**MITD**	**MDSVC**	**VOXFORGE**
CNN	SC	100	96.17	99.64	95.18
	DC	99.92	97.41	99.69	95.83
SVM	SC	100	90.95	99.72	96.76
	DC	99.84	89.82	99.81	94.87

6.6 | Comparison of the datasets

In order to compare the SC and DC AMR audio detection performance of each database, we summarise the average detection rates for each database and system (end-to-end CNN and SVM classifier with deep features) in Table 11. In general, CNN system is superior to SVM except for DC AMR audio files in MDSVC database. However, the performance gap between CNN and SVM systems on MDSVC dataset is very low (99.69% vs. 99.81%). The best AMR audio detection performance is obtained on TIMIT database for both systems as expected. This is probably because of the fact that TIMIT database consists of clean audio recordings collected under controlled conditions without any channel or recording device variability as mentioned earlier. For end-to-end CNN system, the lowest detection rates are obtained on VoxForge dataset. This is possibly because audio recordings in VoxForge dataset are relatively noisy in comparison to TIMIT database and they were not collected using a unified recording setup which introduces channel and environment variations between each recording. MITD database in turn, consisting of previously encoded and decoded speech files using different codecs (AAC, mp3Pro etc.), yields the lowest detection performance using SVM classifier. This is expected since the recordings were previously manipulated by encoding and decoding using various codecs at different BRs. Hence, generating a DC AMR audio signal using the recordings from MITD dataset will produce an audio recording which is encoded three times in total. Considering the fact that each encoding and decoding process leaves its tell-tale footprints on the recordings, the system fails to recognise such audio signals. However, the proposed end-to-end CNN system considerably outperforms SVM classifier for MITD dataset. This indicates that the end-to-end system is more robust against previous audio manipulations on DC AMR audio detection task.

6.7 | Cross database evaluation on DC AMR audio detection

Finally, we evaluate the DC AMR audio detection using end-to-end CNN system when training and test datasets are completely different. The motivation behind these experiments is the fact that, a desired system should be able to recognise any AMR compressed audio file without any prior knowledge about the recording conditions of the test signal. To simulate this, we train the system using a dataset and test files are selected from another dataset which is completely unknown to the system. The average DC AMR audio detection rates obtained using the cross-database evaluation experiments are

TABLE 12 DC AMR audio detection rates (%) obtained using end-to-end system for cross-database evaluation. The last row shows the detection rates obtained when the system is trained using all training data of all datasets

	Test dataset			
Training Dataset	**MDSVC**	**MITD**	**VOXFORGE**	**TIMIT**
MDSVC	99.68	11.41	96.43	96.65
MITD	44.89	97.35	42.57	17.27
VOXFORGE	99.22	10.63	95.69	94.04
TIMIT	99.29	10.44	94.94	99.93
All	99.78	90.49	96.42	99.80

summarised in Table 12. From the table, for most of the datasets (except MITD) end-to-end CNN system yield reasonable detection rates when the system is trained using a completely different database. In case of the mismatch between training and test dataset, MITD dataset show very low performance as expected since the audio recordings in MITD database were previously altered before generating the AMR files. These results show that end-to-end system potentially is a good candidate for detecting the AMR compressed audio files. Although reasonable detection rates are obtained when a single model is trained using all training data of all datasets (the last row of Table 12), the detection rates are low in comparison to the accuracies obtained when training and test recordings come from the same dataset. An important research question arises from the results reported in Table 12 is detecting AMR compressed audio files using recordings previously manipulated by various techniques. Since it is out of the scope of our current work, we leave it for our future studies.

6.8 | Comparison with the previous studies

Finally, we compare the results obtained here with the results previously reported in various studies. Among the four datasets used by the authors, TIMIT database was widely used in the previous studies. In order to make a fair comparison, we select only the studies where TIMIT database was used for DC AMR audio detection. The number of training and test recordings, duration of the recordings, features and classifiers utilised and the average recognition accuracies reported in the literature and our work are summarised in Table 13. The number of training and test files in the table corresponds to the total number of SC and DC AMR audio files used to train and test the system. The average accuracy column corresponds to the accuracy values averaged over the all BRs. Comparison with the previously reported results show that the proposed systems (end-to-end CNN and SVM) outperform the existing studies addressing the DC AMR audio detection using TIMIT database. From the table we can see that although the majority of the previous studies use larger number of files to train the classifier and less number of audio files for testing, the results obtained in our work are superior. Comparing the results reported in the previous studies that used whole-length TIMIT recordings [24, 25, 26] and our results obtained using only 1 s long audio recordings, both end-to-end CNN system or SVM classifier with deep features considerably outperform the previously reported systems. CNN and SVM system used in this work outperforms the earlier studies because spectrogram image conveys both spectral and temporal information together about the audio signal. Thus, CNN learns the most discriminative information for the DC AMR audio detection task whereas in the majority of the previous studies only handcrafted temporal or spectral features were used.

7 | CONCLUSION

Here, the authors proposed to use CNN for DC AMR audio detection. The CNN approach was used for two different purposes: (i) as an end-to-end DC AMR audio detection system which receives the audio file as an input and returns the output whether the input audio is SC or DC and (ii) as a feature extractor where two different feature representations were obtained from the outputs of the two different layers of the CNN model and then those features were used for SVM-based DC AMR audio detection. First, it was shown that double compressing an audio file highly affects the high frequency region of the audio spectrum (Figure 2) and therefore spectrogram of the audio file was proposed to use as an input image for the CNN. Preliminary experiments conducted on audio recordings of 5 s duration from TIMIT database using a total of 8000 trials for each compression BR showed that end-to-end CNN system yields promising results on DC audio detection (Table 2). It was shown that better detection rate is achieved when the same BR is used for training and testing the system, in general. In case of a mismatch between training and test BR, a slight performance degradation was observed. Especially when the system was trained using much lower BR than the test BR, the performance reduction was found to be larger. When CNN was used as a feature extractor and SVM was used for classification, it was found that features extracted from the output of the flattening layer slightly outperform the features extracted from the last fully connected hidden layer (Table 3). In both cases, the recognition rates were slightly better than end-to-end CNN system. In general, using linear kernel showed better performance than RBF and sigmoid kernels with SVM classifier.

After preliminary experiments on 5 s long audio recordings, the proposed CNN system was tested on 1 s long audio recordings using four different datasets. The proposed end-to-end CNN system was found to give great SC AMR audio detection performance (100% detection rate) using TIMIT database whereas 99.92% DC AMR detection rate was obtained. When the proposed system is tested using a dataset consisting of previously manipulated audio recordings (MITD database), 96.17% and 97.41% average recognition rates were obtained for SC and DC AMR files, respectively. Experiments on a dataset composed from audio files recorded at three different locations using two different microphones (MDSVC dataset) showed that the proposed CNN system yield 99.64%

TABLE 13 Comparison of DC AMR detection performance with the previous studies

Related Study	# Training (recordings)	# Test (recordings)	Duration	Classifier	Features	Average accuracy (%)
[10]	8820	3780	5-10 s	SVM	Frequency domain statistical features	84.86
[22]	N.A.	6000	1s	Majority Voting	SAE/NN	91.10
[23]	3000	9000	1s	UBM-GMM	SAE	98.28
[24]	8820	3780	1–8 s	SVM	Statistical features using LP analysis	93.47
[25]	6300	6300	1–8 s	DNN	LTAS	89.12
[26]	8820	3780	1–8 s	SVM	Statistical AMR encoding parameters	99.18
Proposed model	3000	8000	5 s	CNN	STFT	99.92
Proposed model	3000	8000	5 s	SVM	Deep bottleneck features	99.97
Proposed model	3000	8000	1 s	CNN	STFT	99.92
Proposed model	3000	8000	1 s	SVM	Deep bottleneck features	99.84

and 99.69% average detection rates for SC and DC AMR recordings, respectively. It was found that the CNN system is capable of providing great performance independent of the microphone type (Table 8), location (Figure 9) and the spoken phrase (Figure 10). The lowest detection rates were obtained when using the VoxForge dataset which consists of audio recordings recorded under uncontrolled conditions with channel variability and background noise. In general end-to-end CNN system was shown to outperform the SVM classifier with deep features on both SC and DC AMR audio detection (Table 11).

We performed cross database evaluation for DC AMR audio detection and revealed that except for the MITD dataset, in case of a mismatch between the training and test datasets, the proposed system still give reasonably good detection performance with the average detection rate above 95% (Table 12). However, when a previously altered/manipulated audio is compressed using AMR codec and used to test the system trained using a different dataset, the detection rates were found to be very low. So, future works addressing the DC AMR audio detection, should focus on this problem.

Finally, comparison with the previous studies using the TIMIT database showed that the recognition rates obtained by authors in the proposed work were shown to be better than the existing studies (Table 13).

ORCID

Aykut Büker https://orcid.org/0000-0002-6404-1499
Cemal Hanilçi https://orcid.org/0000-0002-9174-0367

REFERENCES

1. Stamm, M.C., Wu, M., Liu, K.J.R.: Information forensics: an overview of the first decade. IEEE Access. 1, 167–200 (2013)
2. Maher, R.C.: Audio forensic examination. IEEE Signal Process. Mag. 26(2), 84–94 (2009)
3. Lukas, J., Fridrich, J., Goljan, M.: Digital camera identification from sensor pattern noise. IEEE Trans. Inform. Forens. Security. 1(2), 205–214 (2006)
4. Boroumand, M., Chen, M., Fridrich, J.: Deep residual network for steganalysis of digital images. IEEE Trans. Inform. Forens. Security. 14(5), 1181–1193 (2019)
5. d. Carvalho, T.J., et al.: Exposing digital image forgeries by illumination color classification. IEEE Trans. Inform. Forens. Security. 8(7), 1182–1194 (2013)
6. Huang, F., Huang, J., Shi, Y.Q.: Detecting double JPEG compression with the same quantization matrix. IEEE Trans. Inform. ForensSecurity. 5(4), 848–856 (2010)
7. Hanilçi, C., et al.: Recognition of brand and models of cell-phones from recorded speech signals. IEEE Trans. Inform. Forens Security. 7(2), 625–634 (2012)
8. Ghasemzadeh, H., Kayvanrad, M.H.: Comprehensive review of audio steganalysis methods. IET Signal Process. 12(6), 673–687 (2018)
9. Imran, M., et al.: Blind detection of copy-move forgery in digital audio forensics. IEEE Access. 5, 12843–12855 (2017)
10. Shen, Y., Jia, J., Cai, L.: Detecting double compressed AMR-format audio recordings. Proc. PCC. (2012)
11. Adaptive Multi-Rate (AMR) Speech Codec. https://www.3gpp.org/ftp/Specs/archive/26_series/26.073/26073-f00.zip Accessed 03 April 2020
12. Yang, R., Shi, Y.Q., Huang, J.: In: Proc. 11th ACM workshop on multimedia and security MM&Sec, pp. 117–124 (2009). Defeating fake-quality MP3
13. Yang, R., et al.: III, E.J.D. (eds.) Proc. SPIE Media Forensics and Security II, vol. 7541, pp. 200–209. International Society for Optics and Photonics. SPIE (2010). Detecting Double Compression of Audio Signal'
14. Liu, Q., Sung, A.H., Qiao, M.: Detection of double MP3 compression. Cognitive Comput. 2, 291–296 (2010)
15. Bianchi, T., et al.: In: Proc. First ACM workshop on information hiding and multimedia security IH&MMSec, pp. 159–164 (2013).Detection and classification of double compressed MP3 audio tracks.
16. Bianchi, T., et al.: Detection and localization of double compression in MP3 audio tracks. EURASIP J. Inform. Security, 10 (2014)
17. Ma, P., et al.: Detecting double-compressed MP3 with the same bit-rate. J. Software. 9(10), 2522–2527 (2014)
18. Yan, D., et al.: Compression history detection for MP3 audio. KSII Trans. Internet Inform. Syst. 12(2), 662–675 (2018)
19. Jin, C., et al.: An efficient algorithm for double compressed AAC audio detection. Multimedia Tools Appl. 75, 4815–4832 (2016)
20. Huang, Q., et al.: In: Sun, X., Pan, Z., Bertino, E. (eds.) Cloud Computing and Security, pp. 347–359. Springer International Publishing (2018). AAC audio compression detection based on QMDCT coefficient.
21. Huang, Q., et al.: AAC double compression audio detection algorithm based on the difference of scale factor. Information. 9(7) (2018)
22. Luo, D., Yang, R., Huang, J.: Detecting double compressed AMR audio using deep learning. Proc. ICASSP., 2669–2673 (2014)
23. Luo, D., et al.: Detection of double compressed AMR audio using stacked autoencoder. IEEE Trans. Inform. Forens. Security. 12(2), 432–444 (2017)

24. Sampaio, J.F.P., Nascimento, F.A.O. In: Proceedings of 22th Brazilian conference on automation (2018). 'Double compressed AMR audio detection using linear prediction coefficients and support vector machine'
25. Büker, A., Hanilçi, C.: Double compressed AMR audio detection using long-term features and deep neural networks. Proc. ELECO., 590–594 (2019)
26. Sampaio, J.F.P., de, O., Nascimento, F.A.: Detection of AMR double compression using compressed-domain speech features. Forens. Sci. Intl. Digital Investig. 33, 200907 (2020)
27. An, N.N., Thanh, N.Q., Liu, Y.: Deep CNNs with self-attention for speaker identification. IEEE Access. 7, 85327–85337 (2019)
28. Abdel-Hamid, O., et al.: Convolutional neural networks for speech recognition. IEEE/ACM Trans. Audio Speech Language Process. 22(10), 1533–1545 (2014)
29. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining Knowledge Discovery. 2, 121–167 (1998)
30. AMR speech Codec: ETSI TS 126 090 V5.0.0 (2002-06). https://www.etsi.org/deliver/etsi_ts/126000_126099/126090/05.00.00_60/ts_126090v050000p.pdf. Accessed 03 April 2020]
31. Rabiner, L., Schafer, R.: Theory and applications of digital speech processing, 1st ed. Prentice Hall Press, Upper Saddle River, NJ, USA (2010)
32. Grigoras, C.: In: Audio Engineering Society conference: 39th international conference: audio forensics: practices and challenges 27–32. (2010). Statistical tools for multimedia forensics
33. Byrne, D., et al.: An international comparison of long-term average speech spectra. J. Acoustic. Soc. Am. 96(4), 2108–2120 (1994)
34. Himawan, I., et al.: In: Marcel, S., Nixon, M.S., Fierrez, J., Evans, N. (eds.) Handbook of Biometric Anti-Spoofing. Advances in Computer Vision and Pattern Recognition, 2nd ed., pp. 391–415. Springer International Publishing (2019). Voice presentation attack detection using convolutional neural networks
35. Lavrentyeva, G., et al.: In: Proceedings of interspeech., pp. 82–86 (2017). Audio replay attack detection with deep learning frameworks
36. TIMIT Acoustic-Phonetic Continuous Speech Corpus. https://catalog.ldc.upenn.edu/LDC93S1
37. Gärtner, D., et al.: In: Audio Engineering Society conference: 54th international conference: audio forensics (2014). A multi-codec audio dataset for codec analysis and tampering detection
38. Woo, R.H., Park, A., Hazen, T.J.: In: IEEE Odyssey - The speaker and language recognition workshop, pp. 1–6 (2006). The MIT mobile device speaker verification corpus: data collection and preliminary experiments 2006
39. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 27 (2011). 1–27:27. http://www.csie.ntu.edu.tw/∼cjlin/libsvm
40. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Machine Learn. Res. 9, 2579–2605 (2008)
41. Muller, K., et al.: An introduction to kernel-based learning algorithms. IEEE Trans. Neural Networks. 12(2), 181–201 (2001)

How to cite this article: Büker A, Hanilçi C. Deep convolutional neural networks for double compressed AMR audio detection. *IET Signal Process*. 2021;15:265–280. https://doi.org/10.1049/sil2.12028