Digital Signal Processing 23 (2013) 1720–1730

Contents lists available at SciVerse ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

# Audio codec identification from coded and transcoded audios

Samet Hicsonmez [a], Husrev T. Sencar [a,*], Ismail Avcibas [b]

[a] *Computer Engineering Department, TOBB Economics and Technology University, Ankara, Turkey*
[b] *Electrical and Electronics Engineering Department, Turgut Ozal University, Ankara, Turkey*

ARTICLE INFO

*Article history:*
Available online 19 April 2013

*Keywords:*
Audio codecs
Codec identification
PSTN
VoIP
GSM
Compression

ABSTRACT

A novel technique is presented to identify the codec of a coded audio. The technique does not perform decoding, utilize any coding metadata, or assume information about the structure describing the bit stream format of a codec. The underlying idea of the technique is that the design choices governing the compression level, audio quality and complexity of a codec will reveal themselves on the coded audio. To exploit this, the technique samples 2–4 kilobytes of data from a coded audio and analyzes the randomness and chaotic nature of the sampled data to build statistical models that represent encoding process associated with different codecs. In experiments, we utilize 16 of the most popular audio codecs used for high quality audio compression and in PSTNs, cellular networks, and VoIP networks by setting encoding parameters of each codec to its most commonly used values. Results show that the codec of an encoded audio can be identified with an accuracy of more than 95 percent. Other experiments considering several transcoding scenarios were also performed. Those results show that the scheme can even discriminate the first encoder of a doubly-encoded audio with an accuracy range of around 80 to 90 percent or more as long as the second codec operates on higher bit rates than the first one.

## 1. Introduction

There are more than one hundred audio codecs used for the encoding and decoding of digital audio. These codecs are used for a variety of tasks like compressing high quality sound and music for more efficient storage, streaming audio over the Internet, and transmission of voice communications over public switched telephone networks (PSTNs), cellular networks and voice-over-IP (VoIP) networks. Despite their variety, codecs essentially differ in the trade-off they make between numbers of competing design objectives. These include a codec's ability to balance compression with sound quality, provide robustness and error correction against noise and network glitches, and adapt to varying transmission bandwidth.

The ability to quickly identify the codec used in coding of an audio without relying on any encoding metadata could provide new ways to tackle some existing challenges. For instance, as multimedia becomes a larger part of network traffic, accurate and fast characterization of audio traffic in its different forms (e.g., streaming, file transfers, VoIP applications, etc.) becomes more and more important [1]. Although it may seem that descriptive information pertinent to encoded data (e.g., file or application metadata or content itself) is sufficient to identify the encoder used in generation of the audio, obtaining this information may not be viable as it requires accessing the payload of packets at lower layers of the network protocol stack. In a high-speed network with many active network flows this level of inspection will require significant computational resources.

Another challenge that can be addressed by codec identification relates to identifying provenance of live phone calls to detect voice spam and voice phishing attacks [2]. In today's vastly diversified and non-centralized telephony infrastructure, there are no mechanisms to determine or verify the origin of an incoming call as the voice signal might have been routed over many networks. In order to obtain some information on what networks the call has traversed, not only the most recently used codec but also codecs used during transcoding have to be determined. Similarly, evaluation of quality of audio files and detecting fake-quality ones, i.e., low bit rate audio files transcoded at higher bit rates pretending to be in high quality, is another application area that can benefit from the ability to identify audio codec and corresponding coding parameters involved in generation of an audio [3]. In both cases of call origin identification and audio quality evaluation, audio content can be analyzed to detect traces of transcoding but this can be a computationally intensive task for a real-time system.

There has been a limited amount of work in audio codec identification. The earliest work in this field attempted to detect the type of coding present on a telephony channel [4]. This method involved placing a least mean squares adaptive filter across the communication channel and obtaining statistics from the filter coefficients, which are then used to train a neural network. Considering LPC,

[*] Corresponding author.
*E-mail addresses:* shicsonmez@etu.edu.tr (S. Hicsonmez), htsencar@etu.edu.tr (H.T. Sencar), iavcibas@turgutozal.edu.tr (I. Avcibas).

1051-2004/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.dsp.2013.04.005

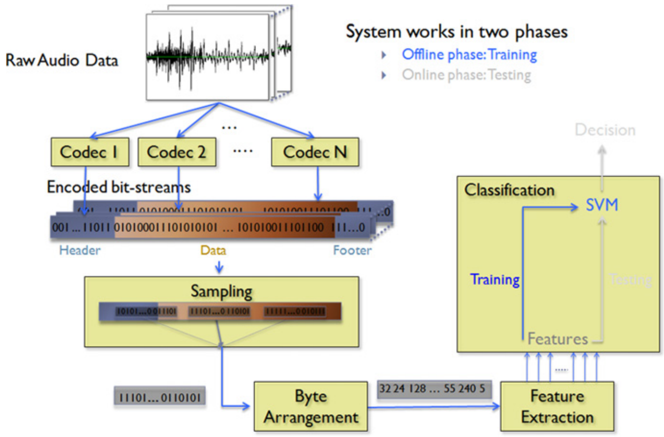

**Fig. 1.** System's operation during the offline phase.

a-law and ADPCM type of coding, an average identification accuracy of 90% was achieved for different learning rates. Later work focused on decoded audio to accomplish the same goal. In Ref. [5], authors proposed utilizing the noise component of speech signals, assuming each codec will have a different impact on it. To exploit this, spectral harmonic-plus-noise decomposition was used to estimate the noise characteristics. Tests performed on six speech codecs, that include ADPCM, AMR, GSM 6.60, GSM 6.20, G.723.1 and G.711, show that although some codecs can be identified without an error, others yield error rates of around 20%. In a similar manner, in [6], authors created a multi-dimensional profile for each speech codec that includes features obtained from noise spectra and time-domain amplitude histogram of coded speech signals. Tests performed considering seven different codecs, including G.711, G.726, G.728, G.729, iLBC, AMR and Silk, show that except for Silk and iLBC codecs, which yielded error rates of 21% and 15%, respectively, speech codecs can be perfectly identified.

In this paper, we extend on our earlier work [7] where we introduced a new technique that can reliably classify encoded audio byte streams generated by a particular audio codec. In the literature, a few approaches have been proposed to statistically characterize encoded data. In [8], considering audio steganalysis problem, Böhme et al. proposed a procedure to determine the encoder used to encode MP3 files (i.e., a certain implementation of the MP3 format) so that an appropriate steganalysis method can be applied. For this purpose, a set of 10 features is designated to capture implementation specific details of an MP3 encoder. These features are then used in conjunction with a machine learning classifier to discriminate between 20 encoders. Although high identification accuracy is reported (87% on a random sample of MP3's), generalization of this approach to other encoders is not trivial as the feature extraction process relies on the knowledge of MP3 file structure.

Alternatively in [1], authors proposed a method to characterize the content of a network flow based on its statistical properties. Rather than determining the codec used in encoding of data, the primary goal of that study is to classify network packet content as belonging to one of a set of data types like text, image, audio, video, encrypted data, etc. The underlying idea of the approach is that byte streams associated with different types of encoded data will exhibit different randomness and redundancy characteristics. Using a set of 6–25 features extracted from a randomly sampled 32 KB of data from each flow in the test dataset, an average accuracy of 90% is achieved in distinguishing seven types of data, which included WMA and MP3 audio file formats as two different types.

Although our approach shares similarities with these techniques, it differs from both of them in two aspects. First, we don't assume the knowledge of coding structure and bit stream format associated with any of the codecs. Second, distinguishing audio encoded with different codecs will be more difficult than distinguishing different types of encoded data; therefore, more reliable statistics are needed. The crux of our technique lies in the fact that the net effect of the design choices inherent to the encoding process reflects on the encoded byte stream. The method exploits this by sampling a small window of data, i.e., a byte vector, from the audio byte stream to characterize the codec in terms of the inherent randomness vs. determinism, entropy, and chaotic properties of the byte stream. Since the technique operates on the encoded audio byte stream and decoding is not required, it is computationally very efficient. In our tests, we utilize 16 of the most popular audio codecs used for high quality audio compression and in PSTN's, cellular networks, and VoIP networks.

The remainder of this paper is organized as follows: In the next section, we present an overview of the system and introduce features that are used to build classification models capable of distinguishing between byte streams generated by different audio codecs. Main characteristics of selected audio codecs and the differences between them are described in Section 3. Experimental results are given in Section 4, and we conclude with further discussion in Section 5.

## 2. Methodology and features

The method described in this paper aims at identifying the codec used in generation of an audio. In achieving this, it neither utilizes any coding metadata, nor assumes any knowledge of structure of the coded data. Instead, it attempts to characterize coded audio in terms of statistical properties that mainly relate to encoding process. Since encoding involves striking a balance between compression, quality and other practical concerns, it is natural for audio encoded with a specific codec to exhibit certain characteristics that do not depend on the audio content itself. To exploit this, our technique measures the inherent randomness and chaotic characteristics of encoded data and uses these measurements in conjunction with a classification system to generate a statistical model.

The system works in two phases, namely, the offline phase and the online phase. In the offline phase, the system is initially built from scratch through a process called training. Operation during this phase can be broken down into four steps. In the first step, all uncoded (raw) audio is encoded by the selected codecs. As a result, many encoded versions of each original audio are obtained. Then, during the second step, all encoded audios undergo sampling which is performed by pulling out a continuous chunk of data from encoded audios. The sampling position is determined randomly for each of the encoded audio to prevent the possibility of any structure in the bit stream biasing the results. In the third step, the sampled bit-sequence is organized as a byte vector, by interpreting each eight-bit as an unsigned integer value between 0 and 255. In the last step, features needed for statistical characterization are extracted from the resulting byte vector, and a multi-class classification system is built. During the online phase, the system is tested against given audio files, and its performance is measured as average accuracy in discriminating between audio files encoded by the same codec. Figs. 1 and 2 illustrate the operation during online and offline phases, respectively.

Obviously, the most important aspect of the technique is the selection of features that will be used for statistical characterization. The features we used to build a model can be grouped into two broad categories as follows.

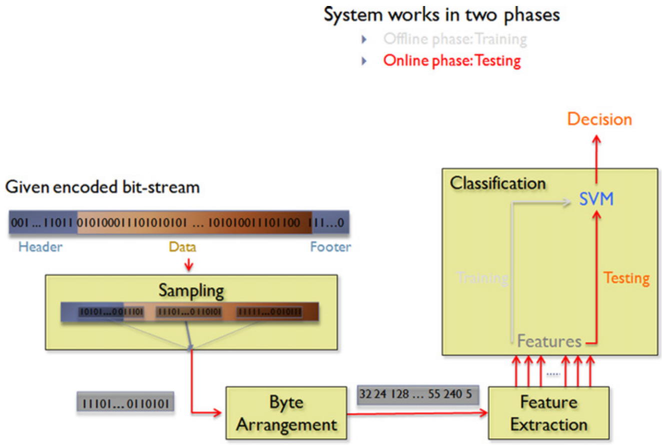

**Fig. 2.** System's operation during the online phase.

### 2.1. Chaotic features

There is theoretical and experimental evidence for the existence of chaotic phenomena in speech signals unexploited by linear models [9]. Assuming that an audio signal is produced by a chaotic system, different codecs will have different impacts on the chaotic structure of an audio signal. The main concept of the chaotic features is based on the neighborhood of the audio signal vectors in the phase space. The phase space vector of a signal $\mathbf{s}(n) = [x(n), x(n+T) \ldots x(n+(D_E-1)T)]$ is reconstructed according to Takens embedding theorem [10], where $x(n)$ is the $n$th sample of the signal, $T$ is the time delay, and $D_E$ is the embedding dimension of the phase space. Takens time delay embedding theorem states that the phase space can reveal useful information about the original unknown chaotic dynamics of the signal if appropriate $D_E$ is selected. The false neighbors method is commonly used for finding the appropriate embedding dimension [11]. The method provides False Neighbors Fraction (FNF) for a given dimension, $D$, and finds the appropriate $D_E$ by increasing $D$ until the FNF reaches zero. The criterion of labeling a neighborhood as true or false is formed by comparing two distances of nearest neighbor points, $\mathbf{s}(n)$ and $\mathbf{s}(m)$, embedded in successively increasing dimensions. If the distance

$$d_D\big(s(n), s(m)\big) = \sqrt{\sum_{k=0}^{D-1} \big(x(n+k\times T) - x(m+k\times T)\big)^2}$$

in $D$-dimensional space is significantly different from

$$d_{D+1}\big(s(n), s(m)\big) = \Bigg(\big(x(n+T\times D) - x(m+T\times D)\big)^2 + \sum_{k=0}^{D-1} \big(x(n+k\times T) - x(m+k\times T)\big)^2\Bigg)^{1/2}$$

in $(D+1)$-dimensional space, then they are considered to be a pair of false neighbors [10]. After labeling all neighbors as true or false, the FNF is defined as the ratio of false neighbors to all neighbors.

The Lyapunov exponent (LE) quantifies the predictability of a chaotic signal [12]. The LE is a global measure for the divergence of nearby trajectories in phase space. A positive exponent means that the trajectories which are initially close to each other move apart over time (divergence). The magnitude of a positive exponent determines the rate of how rapidly they move apart. A system with greater magnitude of LE is said to be more unpredictable. The LE is calculated for each embedding dimension $D_E$ as:

$$\lambda = \lim_{N\to+\infty} \frac{1}{N} \sum_{n=1}^{N} \ln \frac{d(s(n+1), s(m+1))}{d(s(n), s(m))}$$

where $\mathbf{s}(n)$ is the reference point and $\mathbf{s}(m)$ is the nearest neighbor of $\mathbf{s}(n)$ on a nearby trajectory. The Lyapunov exponent is the average rate of divergence (or convergence) of two neighboring trajectories. There are $D_E$ Lyapunov exponents, i.e., $\lambda_1, \lambda_1, \ldots, \lambda_{D_E}$ in descending order. $\lambda_{D_E}$ is known as the largest LE and a positive largest LE is the indicator of chaos. After calculating LEs for all the nearest neighbor pairs on different trajectories, the LE for the whole signal is calculated as the average of these LEs.

Compression algorithms exploit the redundancy present in the signals and their performance is directly related to the decorrelation of their output. As there are no practical perfect compression algorithms, unexploited correlations still remain at their output stream. Essentially, chaotic type features obtained in the phase space simply measure the leftover multi-dimensional correlations in the output stream. Our main hypothesis here is that for each of the compression algorithms these features are different in a statistical sense, and residue correlations in the encoded data are unique enough to design successful classifiers. To capture these differences, we have calculated the FNF and the LEs of the signals with TISEAN [13] software packages. The feature vector $F_{D_E}$ for FNF has three components: The fraction of false neighbors, the average size of the neighborhood, and the root-mean-squared size of the neighborhood:

$$F_{D_E} = \Big[FNF, mean\big(d_{D_E}(s(n), s(m))\big), RMS\big(d_{D_E}(s(n), s(m))\big)\Big]$$

The complete chaotic feature vector, $F$, consists of 26 components:

$$F = \{F_{D_E} \mid D_E = 3, 4, 5, 6, 7\} \cup \{\lambda_i \mid i = 1, 2, \ldots, 11\}$$

### 2.2. Randomness features

These features are primarily inspired by the randomness tests devised by NIST to evaluate randomness properties of cryptographic primitives [14]. We utilize a subset of these features to characterize randomness properties of the sampled byte vectors. Randomness features can be broadly categorized as time or frequency domain depending on how they are computed.

In time domain, simple statistics, like mean, variance, autocorrelation, entropy, and higher order statistics like bicoherence, skewness and kurtosis are computed. Since each codec uses a different compression algorithm and supports variable bit rates and various sample rates, it is expected that these differences will significantly influence the encoded data. For instance, variance is related to variability or spread of values in the data, and presence of specific patterns in data will affect the measured variance. In fact, we observe that samples taken from compressed audio are more likely to be uniformly distributed. This is in line with the fact that compression removes any structure in the data. Auto-correlation is another measure that can be used to reveal repeating patterns in the encoded data. To utilize this, the first 21 coefficients of autocorrelation function are included as features.

Entropy is a measure that quantifies the degree of randomness or uncertainty in the data. Therefore, it can be used to differentiate codecs on the basis of their ability to compress audio. Bicoherence is a higher order statistic that detects and quantifies the non-linearity and non-Gaussianity present in the byte vector. Hence, similar to entropy, it can be helpful in distinguishing between different levels of compression. To capture the impact of different codecs, we compute average bicoherence as another feature. Kurtosis and skewness are two other higher order statistics related to distribution of the data, and they denote two possible ways to quantify how the distribution of data deviates from the normal distribution. Crucially, skewness is a measure of the lack of symmetry in a distribution and kurtosis is the degree of peakedness of a distribution. Both features are computed from probability mass function of the sampled byte vector data.

Frequency domain features are rather simple and provide statistics related to the distribution of energy in several spectral bands. For this, frequency spectrum is divided into four equal sub-bands and mean, variance and skewness of each sub-band are computed as features. Overall, there are 39 features with 27 computed in time domain and 12 in frequency domain.

## 3. Audio codecs

A codec can be evaluated in terms of three design goals. These are the quality (accuracy) of decoded audio, compression rate, and complexity of encoder/decoder [15]. In practice, however, codec design is driven mostly by bandwidth efficiency considerations. Therefore, the degree of compression provided by an audio codec and its adaptability to varying conditions is an important factor. This is expressed by compression bit rate or data rate of a codec, which is determined by multiplying audio sampling rate by the average number of bits needed to code each sample. Typically, speech codecs operate at 8 KHz sampling rate, while music codecs operate at sampling rates of 44.1 KHz or lower. Depending on the choice of coding technique, each sample can be coded with as low as a few bits or a few bytes.

Audio codecs are optimized for coding of either music or speech. Typically, the main concern in music coding has been high quality compression for efficient storage of files; whereas in speech coding, it has been the real-time communication applications. Compared to speech, music has a very wide frequency range and, therefore, music codecs are expected to provide a higher fidelity experience which, in effect, translates to higher sampling and data rates. Aside from the content type, bandwidth requirements of transmission medium is another factor that dictates the choice of a codec. Codecs used in GSM communication have the least bit rate due to the low bandwidth provided by wireless data communication. On the contrary, codecs used in PSTN networks offer the highest bit rate, best audio quality and least complexity as compared to other speech codecs. Codecs used in VoIP communication fall in between these two codec groups in terms of both audio quality and data rates they offer.

From a theoretical standpoint, the most distinctive aspect of a codec design relates to the technique used for coding of raw audio samples. Over time a number of very successful techniques have been developed and subjected to rigorous scientific study. These techniques are based on a few approaches depending on whether speech or music has to be coded. Most generally, speech coders are divided into two categories known as waveform coders and model-based coders. The main difference between the two is that the coders in the latter category are based on a model of speech production; therefore, they provide much better compression. Pulse code modulation (PCM) is the most simple and established waveform coding technique. A variant of this scheme, adaptive differential PCM (ADPCM) [15], is widely used for high quality speech coding. The most successful model-based coding technique is the linear predictive coding (LPC) [16]. In practice, most speech codecs use code-excited linear prediction (CELP) [17], which is based on an LPC model of speech and provide a better speech quality.

As opposed to speech coders, the basic approach used in coding of non-speech audio is based on reducing the redundancy of an audio signal via time to frequency mapping [15]. Due to their ability to perfectly reconstruct the signal, a variety of block-based transformation approaches have been proposed for audio coding. Among these, modified discrete cosine transform (MDCT) is the most popular one as it delivers excellent coding efficiency. Because of this MDCT is widely used in music coders like MP3, AAC, OGG, WMA and AC-3.

**Table 1**
A comparison of audio codecs.

| Codec group | Codec | Default bit rate (Kbps) | MOS[1]<br>ODG[2] | Enc. tech. | MIPS | Delay (ms) |
|---|---|---|---|---|---|---|
| PSTN | a-law [20] | 64 | 4.44 | PCM | 0.01 | 0.125 |
| | u-law | 64 | 4.45 | PCM | 0.01 | 0.125 |
| | PCM | 32 | N/A | ADPCM | ∼ 0 | ∼ 0 |
| GSM | AMR [21] | 12.2 | 4.14 | ACELP | 20 | 25 |
| | AWB [22] | 12.65 | 4.20 | ACELP | 40 | 25 |
| | GSM [23] | 13 | 3.5 | LTP | 5 | 20 |
| | GSM (WAV) | 18 | 3.9 | RPE-LTP | 6 | 20 |
| VoIP | G.729 [24] | 8 | 4.1 | CS-ACELP | 20 | 15 |
| | G.726 [25] | 32 | 4.3 | ADPCM | 2 | 0.125 |
| | iLBC [26] | 13.33 | 4.1 | LPC | 18 | 40 |
| | Speex [27] | 22 | 3.84 | CELP | 40 | 30 |
| High quality compression | AAC [28] | 128 | −0.975 | MDCT | N/A | N/A |
| | MP3 [28] | 128 | −1.179 | MDCT | N/A | N/A |
| | OGG | 128 | −0.485 | MDCT | N/A | N/A |
| | FLAC | lossless | N/A | Linear | N/A | N/A |
| | WMA | 128 | −0.661 | MDCT | N/A | N/A |

[1] MOS values taken from http://www.vocal.com/speech_coders/psqm_data.html.
[2] ODG values taken from [19].

Another aspect that differentiates audio codecs is the quality of the coded audio as compared to that of the uncoded original. Many objective and subjective audio quality measures have been proposed to evaluate the impact of compression on quality of both music and speech. Most widely used subjective speech quality measurement metric is the *mean opinion score* (MOS) [18]. MOS of a codec is a number between 1 and 5 where the above toll quality sound has a MOS score of at least 4. To replace the subjective measures by objective ones, *perceptual evaluation of audio quality* (PEAQ) [19] method is proposed. The output of PEAQ method is known as *objective difference grade* (ODG) that takes values in the range −4 to 0, where higher scores correspond to higher sound quality.

Complexity of a codec, measured in MIPS (millions of instructions per second) numbers, and coding delay, which refers to total algorithmic delay caused by the actual process of encoding and decoding audio samples, are other factors of consideration in evaluating codecs. It must be noted that delay is an important issue in real-time audio transmission; therefore, it is important that GSM, VoIP and PSTN codecs achieve as little delay as possible.

All these factors that contribute to the design of a codec will, directly or indirectly, reveal themselves on the encoded audio. Among these, obviously, encoding technique will have the most prominent effect as it determines the compression level, data rate, and audio quality. Since complexity and delay are more related to the encoding and decoding modules, their effect on the encoded data will be limited at most to bit stream framing and formatting. The fundamental premise of this paper is that these effects will be consistent, will not significantly depend on the audio content itself, and can be modeled in terms of the statistical properties of the coded audio. In our experiments, we used 16 widely utilized codecs that are categorized into four distinct groups according to their main usage area such as communication over PSTN, GSM, VoIP networks and high quality compression. Distinguishing characteristics of these codecs are summarized in Table 1. The second column of the table shows the most commonly used data rate, the third column provides information on the audio quality under ideal conditions in terms of MOS (for speech codecs) and ODG (for high quality compression codecs). The fourth column shows the underlying coding technique, and the fifth column gives a measure of

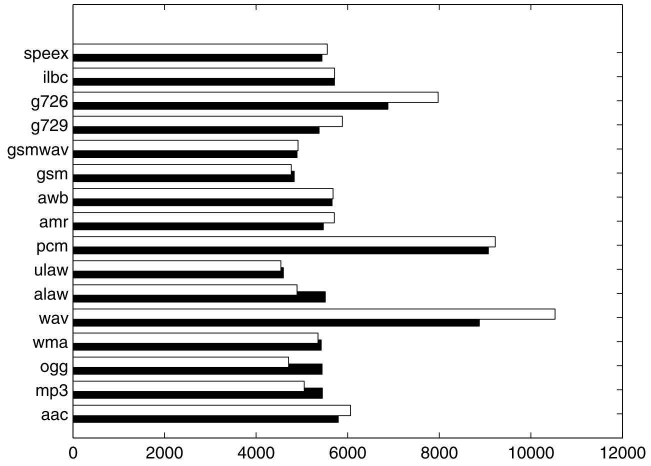

**Fig. 3.** Average variance values of audios coded with different codecs.

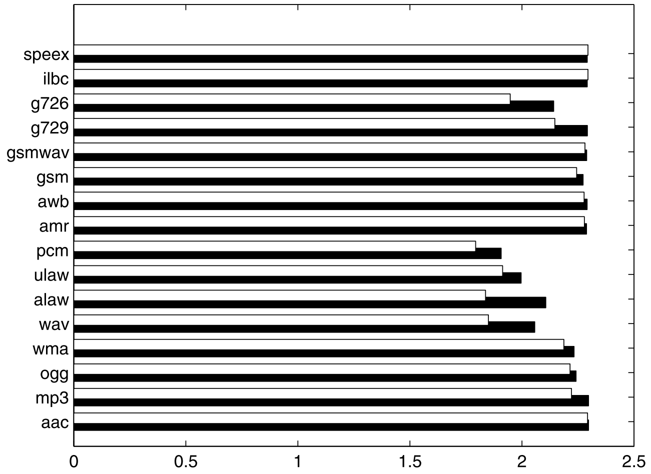

**Fig. 4.** Average entropy values of audios coded with different codecs.

complexity of codecs in terms of MIPS. The last column indicates the codec delay where it applies.

## 4. Experiments

In the experiments, four data sets are used. The first set consists of 1000 audio samples taken from 500 different songs, at audio CD quality (1411 Kbps), of different genres. All the samples are five seconds long and they are obtained by carving out two randomly determined non-overlapping and non-contiguous segments. The second data set consists of 2000 different speech samples taken from the VoxForge speech corpus [29]. These speech samples are 1–13 seconds long and have 256 Kbps bit rate. In the rest of the paper, we will refer to the first data set as Music-I data set and second one as Speech-I data set. To verify the validity of the results, we also generated two larger speech and music data sets, each consisting of 4000 samples. The samples in the music data set are obtained as before with the only difference being that each sample is taken from a different song. The second speech data set is also composed of samples from VoxForge corpus and it has no overlap with Speech-I data set. These latter sets will be referred to as Music-II and Speech-II data sets.

Samples in all data sets are encoded with 16 different codecs using bit rates given in Table 1. After encoding, fixed length byte vectors are randomly extracted from each encoded sample and 65-dimensional feature vectors are computed. Figs. 3–8 demonstrate the variation for six of the 65 features for all the codecs. These six features are the common features selected by a feature selection algorithm by running it on both music and speech data sets. It must be noted however that because feature selection algorithms try to maximize the overall accuracy, some of the selected features that may not seem individually very discriminative may perform very well when combined with other selected features.

In all the figures, white and black bars represent the average values computed, respectively, on Music-II and Speech-II data sets. Figs. 3 and 4 represent average variance and entropy values obtained from encoded music and speech data with different codecs. Both of these features can be attributed to lack or presence of some structure in the coded data and can be intuitively related to the way how different codecs implement compression. Similarly, although they are difficult to directly link to physical phenomena, it can be seen in Figs. 5–8 that average values for auto-correlation function coefficients, FNF ratios, and Lyapunov exponents show significant variation depending on the type of encoding. False Neighbors Fraction is a chaotic type feature and average values of two FNF features with respect to 16 codecs are displayed in Figs. 6 and 7. More clearly these two vectors are the average size of the neighborhood and the average of the squared size of the neighborhood of fifth dimension, respectively. Average values for the LE, which is another chaotic type feature, is shown in Fig. 8. Values displayed in the figure are belong to logarithm of the stretching factor of first iteration.

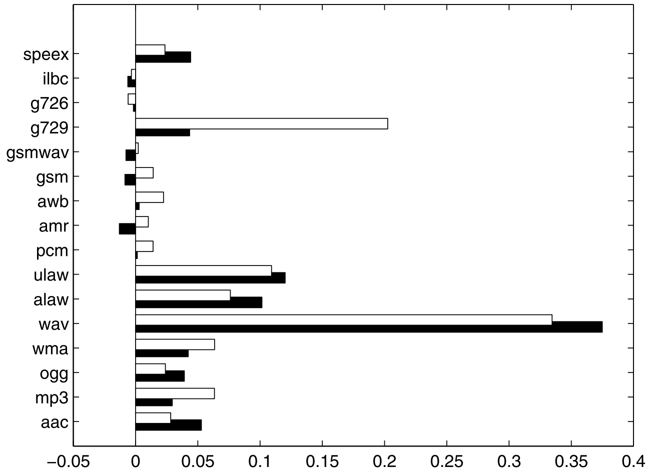

**Fig. 5.** Average values of 11th coefficient of auto-correlation function of audios coded with different codecs.

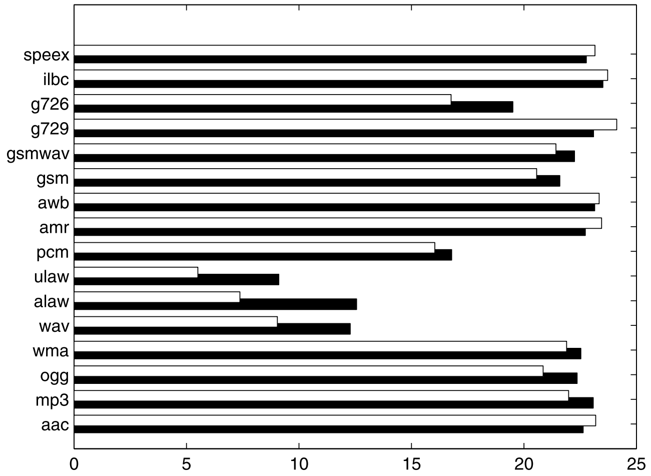

**Fig. 6.** Average values of the average size of the neighborhood for 5th embedding dimension of audios coded with different codecs.

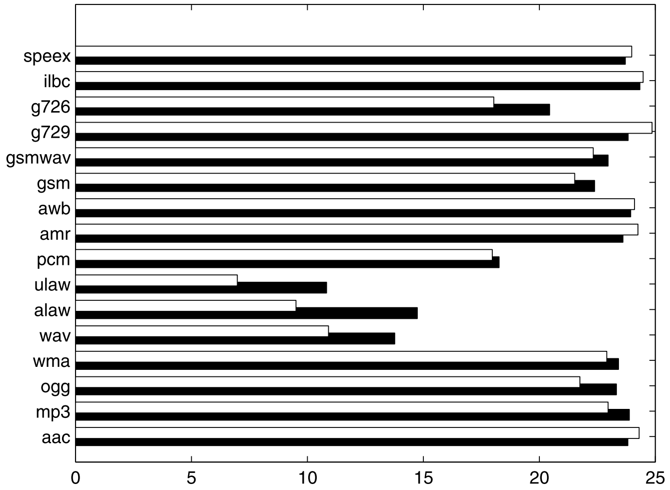

**Fig. 7.** Average values of the average of the squared size of the neighborhood for 5th embedding dimension of audios coded with different codecs.

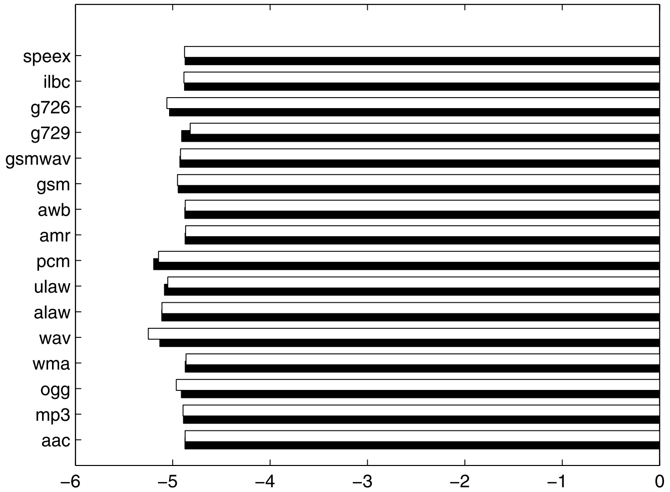

**Fig. 8.** Average of the LE of audios coded with different codecs.

For classification, we use a standard machine learning technique, a support vector machine (svm) implemented in the Libsvm package [30] with radial basis kernel. In all experiments, half of the data set is used for training and remaining half is used for testing. We should note that when working on the Music-I data set we ensured that two samples taken from each song are either placed in the training or test set but are not distributed among the two sets. We conducted two groups of experiments. In the first group, we aimed at identifying codec of a coded audio considering different sampling window sizes. In the second group, we focused on the transcoding scenario where encoding–decoding of an audio is followed by another encoding, and our goal is to identify the first codec, i.e., codec used prior to transcoding.

### *4.1. Single coding scenario*

Several experiments are performed on the four data sets. In all tests, uncoded audio samples are also included as a separate class in the classification, yielding a 17-class classification problem. One of the most important parameters of the scheme is the sampling window size. To determine the optimum size with regard to accuracy and computation time, we made several tests considering window sizes of 1 KB, 2 KB, 4 KB and 8 KB. Table 2 presents average classification accuracies corresponding to these window sizes. Results obtained on Music-I data set show that accuracy increases rather steadily from 1 KB to 8 KB. In terms of computation time, however, feature extraction from 8 KB audio samples took significantly longer than that from 4 KB samples. Speech-I data samples were relatively shorter in duration; because of this encoding with some of the codecs compressed samples to a size less than 4 KB. Therefore, accuracies were obtained only for 1 KB and 2 KB sampling window sizes, which yielded slightly better accuracies as compared to those obtained on Music-I data set. In order to maintain a balance between accuracy and computation time, in the rest of the tests window sizes were chosen as 4 KB and 2 KB for music and speech data sets, respectively.

**Table 2**
Effect of sampling window size on accuracy.

| Window sizes | Accuracy (%) | |
|---|---|---|
| | Music-I | Speech-I |
| 1 KB | 85.1 | 87.04 |
| 2 KB | 90.91 | 97.34 |
| 4 KB | 95.88 | N/A |
| 8 KB | 98.13 | N/A |

The proposed technique's ability to differentiate among different codecs is presented as confusion matrices given in Table 13 and Table 14, respectively, obtained on Speech-I and Music-I data sets. As can be seen in the test results concerning speech data set, almost all codecs can be discriminated successfully and the few confused ones are those that are generated by codecs using the same encoding technique at different bit rates. For instance, AMR, AWB and G.729 all use ACELP as encoding technique, and samples from the three were confused only with each other. Similarly, a-law, u-law, PCM, WAV (uncoded) and G.726 coded samples are confused with each other, as all these codecs are based on PCM or ADPCM. Results on music data set exhibit a similar pattern as well. Repeating the same test on Music-II and Speech-II data sets using a sampling window size of 4 KB yielded an accuracy of 97.91% and 97.88%, respectively.

We also performed feature selection to determine the most effective features on classification accuracy in the two tests. For feature selection we used the sequential floating forward search method [31]. This algorithm first determines a feature that singlehandedly yields the best accuracy. Then it tries to find a second feature that when combined with the first one results in the highest accuracy. This procedure continues iteratively until a given number of features are selected. Floating search method offers the flexibility to reconsider features previously discarded. It is found separately on both music and speech data sets that selecting more than 10 features, from the set of 65 features, improves the classification accuracy only slightly. Results show that skewness, entropy, mean, variance, the average size of the neighborhood for 3rd, 4th and 5th embedding dimension, the average of the squared size of the neighborhood for 5th embedding dimension, the fraction of false nearest neighbors for 5th embedding dimension, and 2nd, 5th, 11th, 12th coefficients of auto-correlation function, and logarithm of the stretching factor for first iteration, which is an LE feature, have higher discrimination capability. It must be noted that these 14 distinct features are obtained by combining the 20 features obtained after two feature selection runs on the two data sets, out of which six were common. When we repeated both tests utilizing only the selected 10 features, we observed that accuracy values of 96.15% and 95.80% can be achieved on music and speech data sets, respectively. These results show that the complexity of the method can be reduced considerably by decreasing the number of features with a very slight reduction in the performance.

To ensure that classification accuracy does not only rely on the bit rate of a codec, we repeated the same experiment considering similar bit rate and worst bit rate coding scenarios. In the case

**Table 3**
Within classes test results for all data sets.

| Codec class | Accuracy (%) | | | |
|---|---|---|---|---|
| | Music-I (4 KB) | Speech-I (2 KB) | Music-II (4 KB) | Speech-II (4 KB) |
| GSM | 99.05 | 95.65 | 99.01 | 98.46 |
| PSTN | 98.8 | 97.8 | 99.65 | 98.87 |
| VoIP | 99.95 | 99.02 | 99.46 | 99.38 |
| High quality compression | 95.33 | 100 | 98.59 | 98.93 |

of similar bit rate coding, rather than using default bit rate options, uncoded audio samples are coded at similar bit rate options. For codecs that offer only one bit rate option those bit rates are used. For this reason, to match the speed of PCM-based codecs, which provides only 64 Kbps bit rate option, all audios are coded at 64 Kbps with the high quality compression codecs. For GSM and VoIP codecs, we chose 13 Kbps as encoding bit rate. Hence, overall, PSTN and high quality compression codecs are used for encoding at 64 Kbps and others at their closest bit rate option to 13 Kbps. In the worst bit rate coding scenario, however, we encode the uncoded samples with encoders' worst bit rate options. In both tests, Music-I data set is used with 2 KB as the sampling window size. Corresponding classification results are obtained as 94.13% and 94.37%, respectively. These results show us that the technique's ability to differentiate codecs is not simply based on the level of compression.

We also tested encoders within their codec classes. More clearly, we constructed four different classifiers, first for discriminating GSM codecs alone, second for PSTN codecs, third for VoIP codecs and the last one for high quality compression codecs. Using Music-I data set with 4 KB sampling window size yielded an accuracy of 99.05%, 98.8%, 99.95% and 95.33%, respectively, for the GSM, PSTN, VoIP, and high quality compression codec classes. The same test is repeated with the Speech-I data set using 2 KB byte vectors. Classification results for this test are obtained as 95.65%, 97.8%, 99.02% and 100%, respectively, for the same codec classes. These tests are also performed on Music-II and Speech-II data sets. Corresponding results are given in Table 3. It can be seen from these results that an increase in both the number of samples used for training and the sampling window size improves the accuracy of the technique and shows that audio content has no impact on the performance.

Comparison of the results given in Tables 13 and 14 and the results in Table 3 show that the scheme's ability to discriminate codecs varies slightly depending on the type of data set used for testing. In general, it can be observed that codecs designed to mainly encode speech are more accurately discriminated when tested on a music data set. Likewise, speech data sets yield higher accuracy in discrimination of codecs primarily used for coding music. This is mainly so because codecs designed to encode a given type of audio are better tailored to exploit the underlying characteristics of that audio type. (It can be seen in Table 1 that codecs in the same codec group use similar encoding algorithms and bit rates.) As a result, codecs perform much better and behave more uniformly when the type of audio content matches the codecs' usage intent. On the contrary, when there is such a mismatch, codecs perform worse than expected and the differences among codecs become more apparent, which makes the task of classification easier. The only exception to this observation is the PSTN codecs, and this may be attributed to simplicity of their encoding, i.e., sampling followed by quantization, which does not take into account the specifics of either speech or music. However, even there, it can be seen that as coding bit rate decreases to 32 kbps from 64 kbps, PCM codec can be better differentiated on music data set as it will introduce more coding artifacts on music than speech.

**Table 4**
Confusion matrix for GSM to PSTN transcoding scenario using a-law codec on the music data set.

| | | | Classified | | | | |
|---|---|---|---|---|---|---|---|
| | | | double coded | | | | single coded |
| | | | AMR | AWB | GSM | GSM (WAV) | a-law |
| Actual | double coded | AMR | 99 | 1 | 0 | 0 | 0 |
| | | AWB | 4.8 | 94 | 0 | 1.2 | 0 |
| | | GSM | 0 | 0.2 | 98.6 | 0 | 1.2 |
| | | GSM(WAV) | 0.6 | 26.2 | 0 | 72.6 | 0.6 |
| | single coded | a-law | 0.8 | 0.6 | 0.6 | 0.8 | 97.2 |

**Table 5**
GSM to PSTN transcoding test results.

| Transcodec | Accuracy (%) | |
|---|---|---|
| | Music-I | Speech-I |
| a-law | 92.28 | 89.78 |
| PCM | 84.48 | 90.20 |
| u-law | 85.20 | 90.66 |

### 4.2. Transcoding scenario

When two parties engage in voice communication, most of the time, the audio must traverse different communication networks. For instance, when a VoIP user calls a GSM phone, audio data is first encoded with appropriate VoIP codec and then re-encoded with a GSM codec. In this and similar situations, where the audio must be encoded more than once, the ability to identify the first encoder may reveal information on the network that originated the call or on the relay networks. This capability can be further simplified to distinguishing between singly-encoded and doubly-encoded audio. To test the applicability of our technique to such problems, we considered three different transcoding scenarios. In the experiments, since only Music-I and Speech-I data sets are used with 4 KB as the length of the byte vector, after this point we will refer to them as music and speech data sets, respectively.

First, we consider the scenario concerning a call made from a GSM network to PSTN network. In this case, audio is first encoded with one of the four possible GSM codecs and later re-encoded with one of the three PSTN codecs. Since there are three PSTN codecs, three different tests are performed: one for a-law codec, one for PCM codec and the last one for u-law codec. In all the three tests, we first encode a raw audio sample with four GSM codecs, decode it, and later encode it with one of the three PSTN codecs separately. We also include the singly-encoded audio in these tests to ensure that the technique can also distinguish between singly-coded and transcoded audio. This essentially results in a 5-class classification test. The test concerning GSM to PSTN transcoding with the a-law codec yielded an accuracy of 92.28% on music and 89.78% on speech data sets. Corresponding confusion matrix is presented in Table 4 as an example. When the transcodec is changed to PCM codec the accuracy is obtained to be 84.48% and 90.2%, respectively, for the music and speech data sets. Similarly, for the u-law codec the accuracies are 85.2% and 90.66% for the two data sets. These results also can be seen in Table 5.

Second, we consider the GSM to VoIP transcoding scenario. Here, audio is initially encoded with one of the four GSM codecs and then re-encoded with one of the four VoIP codecs. Since there are four VoIP codecs, four tests are performed, namely, GSM to VoIP transcoding with G.729, G.726, iLBC and Speex codecs. As before, in all cases, singly-encoded audio is included as a separate class, thereby yielding five-class classification in all cases. Classification accuracies on the music data set are found to be 62%,

**Table 6**
GSM to VoIP transcoding test results.

| Transcodec | Accuracy (%) | |
|---|---|---|
| | Music-I | Speech-I |
| G.729 | 62.00 | 45.46 |
| G.726 | 87.44 | 81.30 |
| iLBC | 95.00 | 87.54 |
| Speex | 91.20 | 79.22 |

**Table 7**
Confusion matrix for GSM to VoIP transcoding scenario using G.729 codec on the speech data set.

| | | | Classified | | | | |
|---|---|---|---|---|---|---|---|
| | | | double coded | | | | single coded |
| | | | AMR | AWB | GSM | GSM (WAV) | G.729 |
| Actual | double coded | AMR | 33.4 | 11.5 | 6.1 | 36.2 | 12.8 |
| | | AWB | 0.2 | 89.5 | 0.6 | 2.5 | 7.2 |
| | | GSM | 4.7 | 10.8 | 20.3 | 27.4 | 36.8 |
| | | GSM(WAV) | 9 | 12.4 | 13.4 | 35.2 | 30 |
| | single coded | G.729 | 8.2 | 7.2 | 9.9 | 26.8 | 47.9 |

**Table 8**
VoIP to PSTN transcoded scenario: Confusion matrix of VoIP to u-law test using music data set.

| | | | Classified | | | |
|---|---|---|---|---|---|---|
| | | | double coded | | | single coded |
| | | | G.729 | G.726 | Speex | u-law |
| Actual | double coded | G.729 | 99.2 | 0 | 0 | 0.8 |
| | | G.726 | 0 | 98.4 | 1.6 | 0 |
| | | Speex | 0 | 10 | 90 | 0 |
| | single coded | u-law | 0.4 | 0 | 0 | 99.6 |

**Table 9**
VoIP to PSTN transcoding test results.

| Transcodec | Accuracy (%) | |
|---|---|---|
| | Music-I | Speech-I |
| a-law | 97.15 | 88.52 |
| PCM | 99.20 | 88.67 |
| u-law | 96.80 | 98.52 |

87.44%, 95%, and 91.2% respectively, for the four VoIP codecs as listed above. On the speech data set, corresponding accuracies are measured as 45.46%, 81.30%, 87.54%, and 79.22%. These results are given in Table 6. Results show that when G.729 is used as the transcodec, it yields very low accuracy. Confusion matrix for the test corresponding to GSM to VoIP transcoding with G.729 codec is given in Table 7. Examining this result closely shows that this may be attributed to the fact that G.729 codec has a lower bit rate than all of the GSM codecs, see Table 1. Essentially, in a transcoding scenario, when the second codec has a lower bit rate than the first one, second encoding will remove all traces of the first encoding. Therefore, in such situations, the technique is not able to identify the first encoder.

Our last experiment covers transcoding from VoIP to PSTN. Three tests are performed considering transcoding with a-law, PCM and u-law codecs. Table 8 gives the confusion matrix for VoIP to PSTN transcoding with u-law codec. These results are in line with the previous experiments. The overall classification results obtained on the music data set are 97.15%, 99.2% and 96.8%, respectively, for the transcodecs listed above. Speech data set test accuracies are 88.52%, 88.67% and 98.52%. These results can also be seen in Table 9.

From the results of Table 6 it can also be seen that when speech codecs are used for initial coding and transcoding, the initial codec can be identified more accurately on a music data set than a speech data set. This is in line with our observation that codecs optimized for speech coding will leave more traceable artifacts when used in coding of music, thereby making them easier to differentiate. Similar to our earlier observation, Tables 5 and 9 show that when PSTN codecs are used for transcoding, even if the initial codec is a speech codec, music data set will not necessarily yield better results. This is due to indifference of the design of PSTN codecs in coding of different types of audio.

Overall, we observe that the proposed technique performs quite reliably in identifying the first encoder of a doubly-encoded audio as long as the second codec operates on higher bit rates than the first one. Due to this limitation other transcoding scenarios, like PSTN to GSM or PSTN to VoIP are not considered in the experiments.

### *4.3. Comparison*

In the literature there are two other works that took a similar approach in trying to identify the algorithm used in encoding of a data through analysis of statistical properties of the coded data. Both of these works are concerned with different application areas. Ref. [1] is about fast classification of network flows to identify the type of data (e.g., audio, video, text, image, etc.), and Ref. [8] focuses on MP3 coding to differentiate between different implementations of MP3 coding format. As compared to our approach, [1] tries to differentiate between different data types where there is a larger freedom for discrimination, whereas [8] looks at implementation level specifics of an MP3 codec which assumes and utilizes detailed information of the coding format itself. Our approach stands in between [1] and [8] where we constrained ourselves to only an audio data type while remaining oblivious to any details of the coding format and process.

When considered in the context of audio codec identification, the approach given in [8] and the proposed features cannot be applied to our application domain. This is primarily so because introduced features are very specific to MP3 coding and cannot be extended to identification of other formats unless a similar set of features are determined for each coding format. This is exactly the opposite direction we are willing to take in this work. It should also be noted that the basis of our approach hinges on the assumption that there are variations in compression, quality, and degree of redundancy in the way different codecs perform encoding. Among different implementations of a coding format, there will only be minute differences with respect to the above design choices, and this will largely render our approach ineffective.

Our randomness features, however, are inspired by [1], and we have performed several experiments to determine the contribution of chaotic features in discriminating audio codecs. We performed tests considering both single coding and transcoding scenarios. In the single coding scenario, we repeated all the tests by utilizing only [1]'s features, i.e. randomness features. Table 10 shows the results of these tests and average accuracies as compared to those of ours. The experiments are performed on the same data sets using the same window sizes and the same coding parameters. Obtained average accuracies show that both schemes perform with accuracies over 90% and that our scheme can consistently improve the results of [1] by about 2–4%.

Our scheme's superiority becomes more evident in the tests concerning transcoding scenarios. Considering the GSM to PSTN, GSM to VoIP, and VoIP to GSM coding scenarios described in Section 4.2, we performed one test from each category by fixing the

**Table 10**
Comparison of [1] and our scheme in terms of average accuracies for 17-class classification on Music-I and Speeh-I data sets.

| Methods | Accuracy (%) | | | |
|---|---|---|---|---|
| | Music-I<br>(4 KB) | Speech-I<br>(2 KB) | Music-II<br>(4 KB) | Speech-II<br>(4 KB) |
| [1] | 91.87 | 96.72 | 96.49 | 96.46 |
| Our scheme | 95.88 | 97.34 | 97.91 | 97.88 |

**Table 11**
Comparison of [1] and our scheme in terms of transcoding test results obtained on Music-I data set.

| Methods | Accuracy (%) | | |
|---|---|---|---|
| | GSM to PSTN<br>(GSM to<br>a-law test) | GSM to VoIP<br>(GSM to<br>Speex test) | VoIP to PSTN<br>(VoIP to<br>PCM test) |
| [1] | 61.8 | 73.84 | 79.85 |
| Our scheme | 92.28 | 91.20 | 99.20 |

**Table 12**
Comparison of [1] and our scheme in terms of transcoding test results obtained on Speech-I data set.

| Methods | Accuracy (%) | | |
|---|---|---|---|
| | GSM to PSTN<br>(GSM to<br>a-law test) | GSM to VoIP<br>(GSM to<br>G.726 test) | VoIP to PSTN<br>(VoIP to<br>u-law test) |
| [1] | 87.98 | 79.42 | 88.22 |
| Our scheme | 89.78 | 81.30 | 98.52 |

initial codec and the transcodec. Tables 11 and 12 show the accuracies for the tests performed on Music-I and Speech-I data sets, respectively.

Results demonstrate that our scheme may yield improvements up to almost 30% over [1] on Music-I data set and improvements in the range of 2–10% on Speech-I data set. These accuracy results lead to two important conclusions. The first is that chaotic features are less likely to be curtailed by (a higher bit rate) transcoding operation in identifying the initial codec as compared to randomness features. The second is that chaotic features are less dependent on the type of audio content. The larger performance gap on music data set shows that randomness properties of music signals are less resistant to transcoding.

## 5. Discussion and conclusion

A fast and simple method is introduced to identify codec used for encoding of an audio. The method uses a multi-class classification system based on features that characterize randomness and chaotic behavior of encoded data. Two sets of experiments are performed. In the first one, identification among 16 audio codecs is considered. Results show that most codecs can be identified with accuracies higher than 95% and the confusions are due to codecs that have the same or similar encoding technique. In the second set of experiments, encoded audio samples are transcoded with another codec and the technique's ability to identify the first codec is tested. Results in this case show that singly-coded and transcoded audio codecs can be discriminated from each other with an accuracy close to 100%, and when considering double encoding of audio, the codec used prior to transcoding can be identified approximately with 80% accuracy. The only limitation here is that if transcoding is made with a codec that performs severe compression, the technique cannot very reliably distinguish the codec used for the initial encoding. Overall, results show the proposed technique can be successfully used to identify encoder of a singly- or doubly-encoded audio among a number of codecs that utilize

**Table 13**
Confusion matrix for 17-class classification on the Speech-I data set.

| | | | Classified | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PSTN | | | GSM | | | | VoIP | | | | High quality compression | | | | | |
| | | | a-law | u-law | PCM | AMR | AWB | GSM | GSM<br>(WAV) | G.729 | G.726 | iLBC | Speex | AAC | MP3 | OGG | FLAC | WAV | WMA |
| Actual | PSTN | a-law | **100** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | u-law | 6.7 | **93** | 0.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| | | PCM | 0 | 0 | **96** | 0 | 0 | 0 | 0 | 0 | 3.7 | 0 | 0.1 | 0.2 | 0 | 0 | 0 | 0 | 0 |
| | GSM | AMR | 0 | 0 | 0 | **96.7** | 2 | 0 | 0 | 1.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | AWB | 0 | 0 | 0 | 1 | **99** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | GSM | 0 | 0 | 0 | 0 | 0 | **97.5** | 2.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | GSM<br>(WAV) | 0 | 0 | 0 | 0 | 0 | 14.2 | **85.8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | VoIP | G.729 | 0 | 0 | 0 | 2.3 | 0.4 | 0 | 0 | **96.2** | 0 | 0 | 0 | 1.1 | 0 | 0 | 0 | 0 | 0 |
| | | G.726 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **96** | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | iLBC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **99.9** | 0 | 0 | 0.1 | 0 | 0 | 0 | 0 |
| | | Speex | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **100** | 0 | 0 | 0 | 0 | 0 | 0 |
| | High quality compression | AAC | 0 | 0 | 0 | 0.3 | 0 | 0 | 0 | 0 | 0 | 0 | | **99.7** | 0 | 0 | 0 | 0 | 0 |
| | | MP3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **100** | 0 | 0 | 0 | 0 |
| | | OGG | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **100** | 0 | 0 | 0 |
| | | FLAC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **100** | 0 | 0 |
| | | WAV | 0.5 | 2.1 | 0.1 | 0 | 0 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 0 | 0 | 0 | **95** | 0 |
| | | WMA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **100** |

**Table 14**
Confusion matrix for 17-class classification on the Music-I data set.

| | | | Classified | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PSTN | | | GSM | | | | VoIP | | | | High quality compression | | | | | |
| | | | a-law | u-law | PCM | AMR | AWB | GSM | GSM (WAV) | G.729 | G.726 | iLBC | Speex | AAC | MP3 | OGG | FLAC | WAV | WMA |
| Actual | PSTN | a-law | **93.6** | 4.6 | 1 | 0 | 0 | 0 | 0 | 0 | 0.2 | 0 | 0 | 0 | 0 | 0 | 0 | 0.6 | 0 |
| | | u-law | 3 | **96** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | PCM | 0 | 0.4 | **99.6** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | GSM | AMR | 0 | 0 | 0 | **99.4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.6 | 0 | 0 |
| | | AWB | 0 | 0 | 0 | 1.6 | **96.4** | 0 | 0 | 0.2 | 0 | 0 | 0.4 | 0 | 0 | 0 | 1.4 | 0 | 0 |
| | | GSM | 0 | 0 | 0 | 0 | 0 | **92** | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | GSM (WAV) | 0 | 0 | 0 | 0.2 | 0 | 1 | **96.6** | 0.8 | 0 | 0 | 0.4 | 0 | 0 | 0 | 1 | 0 | 0 |
| | VoIP | G.729 | 0 | 0 | 0 | 0 | 0 | 0 | 0.8 | **96.8** | 0 | 0 | 0 | 0 | 0 | 0 | 2.4 | 0 | 0 |
| | | G.726 | 0.2 | 0 | 0.2 | 0 | 0 | 0 | 0 | 0 | **99.6** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | iLBC | 0 | 0 | 0 | 0 | 0 | 0 | 0.2 | 0 | 0 | **98.6** | | 1 | 0 | 0.2 | 0 | 0 | 0 |
| | | Speex | 0 | 0 | 0 | 0 | 0 | 0 | 0.2 | 0.2 | 0 | 0 | **99** | 0.2 | 0 | 0.2 | 0.2 | 0 | 0 |
| | High quality compression | AAC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.2 | **97.4** | 1.4 | 0.4 | 0.6 | 0 | 0 |
| | | MP3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.2 | 2 | **88.4** | 6.4 | 3 | 0 | 0 |
| | | OGG | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.8 | 0.2 | 7.8 | **91** | 0 | 0 | 0.2 |
| | | FLAC | 0 | 0 | 0 | 0 | 0 | 0 | 0.2 | 0.6 | 0 | 0 | 0 | 0 | 2 | 0 | **97** | 0 | 0.2 |
| | | WAV | 0 | 0 | 1.6 | 0 | 0 | 0 | 0 | 0 | 0.6 | 0 | 0 | 0 | 0 | 0 | 0 | **97.8** | 0 |
| | | WMA | 0 | 0 | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6.8 | 1.6 | 0 | 0 | **90.8** |

different encoding techniques even if they encode at the same bit rate or among codecs that use similar or same encoding technique but at reasonably different coding bit rates.

## Acknowledgment

This material is based upon work supported by the Scientific and Technological Research Council of Turkey under research grant number 110E049.

## References

[1] K. Shanmugasundaram, M. Kharrazi, N. Memon, Nabs: A system for detecting resource abuses via characterization of flow content type, in: 20th Annual Computer Security Applications Conference, 2004, pp. 316–325.

[2] V.A. Balasubramaniyan, A. Poonawalla, M. Ahamad, M.T. Hunter, P. Traynor, Pindr0p: using single-ended audio features to determine call provenance, in: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, IL, USA, October 4–8, 2010, pp. 109–120.

[3] R. Yang, Y.-Q. Shi, J. Huang, Defeating fake-quality mp3, in: Proceedings of the 11th ACM Workshop on Multimedia and Security, 2009, pp. 117–124.

[4] D. Alley, Automatic identification of voice band telephony coding schemes using neural networks, Electron. Lett. 29 (13) (1993) 1156–1157.

[5] K. Scholz, L. Leutelt, U. Heute, Speech-codec detection by spectral harmonic-plus-noise decomposition, in: Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, vol. 2, 2004, pp. 2295–2299.

[6] F. Jenner, A. Kwasinski, Highly accurate non-intrusive speech forensics for codec identifications from observed decoded signals, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 1737–1740.

[7] S. Hicsonmez, H.T. Sencar, I. Avcibas, Audio codec identification through payload sampling, in: IEEE International Workshop on Information Forensics and Security, (WIFS), 2011, pp. 1–6.

[8] R. Böhme, A. Westfeld, Statistical characterisation of mp3 encoders for steganalysis, in: Proceedings of the 2004 Workshop on Multimedia and Security, 2004, pp. 25–34.

[9] O. Kocal, E. Yuruklu, I. Avcibas, Chaotic-type features for speech steganalysis, IEEE Trans. Inform. Forensics Secur. 3 (4) (2008) 651–661.

[10] H. Abarbanel, Analysis of Observed Chaotic Data, Springer, 1996.

[11] B.F. Takens, Dynamical Systems and Turbulence, Lecture Notes in Math., vol. 898, Springer, 1981.

[12] C.R. Hilborn, Chaos and Nonlinear Dynamics, Oxford University Press, 2000.

[13] H.K.R. Hegger, T. Schreiber, Practical implementation of nonlinear time series methods: The TISEAN package, Chaos 9 (1999) 413.

[14] A. Rukhin, J. Soto, J. Nechvatal, E. Barker, S. Leigh, M. Levenson, D. Banks, A. Heckert, J. Dray, S. Vo, A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST special publication 800-22, 2001.

[15] M. Bosi, R.E. Goldberg, Introduction to Digital Audio Coding and Standards, Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[16] J. Bradbury, Linear predictive coding, Online PDF, http://my.fit.edu/vKepuska/ece5525/lpc_paper.pdf, 2000.

[17] M. Schroeder, B. Atal, Code-excited linear prediction(celp): High-quality speech at very low bit rates, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 10, ICASSP'85, 1985, pp. 937–940.

[18] W.B. Kleijn, K.K. Paliwal (Eds.), Speech Coding and Synthesis, 1995.

[19] D. Câmpeanu, A. Câmpeanu, Peaq – An objective method to assess the perceptual quality of audio compressed files, in: Proceedings of the International Symposium on System Theory, SINTES 12, 2005, pp. 487–492.

[20] ITU-T recommendation G.711, Pulse code modulation (PCM) of voice frequencies, 1988.

[21] E. Ekudden, R. Hagen, I. Johansson, J. Svedberg, The adaptive multi-rate speech coder, in: Proceedings of IEEE Workshop on Speech Coding, 1999, pp. 117–119.

[22] ITU-T Recommendation G.722.2, Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB), Tech. rep., International Telecommunication Union, 2003.

[23] ETSI, Digital cellular telecommunications system (phase 2+), full rate speech, transcoding (gsm 06.10 version 7.0.0 release 1998), Tech. rep.

[24] L. Juan, L. Biqin, F. Qiuliang, An 8-kb/s conjugate-structure algebraic celp (cs-acelp) speech coding, in: Fourth International Conference on Signal Processing Proceedings, vol. 2, ICSP'98, 1998, pp. 1729–1732.

[25] I.T. Union, G.726: 40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM), Series G: Transmission systems and media, digital systems and networks; General aspects of digital transmission systems, Terminal equipments.

[26] S. Andersen, A. Duric, H. Astrom, R. Hagen, W. Kleijn, J. Linden, Internet low bit rate codec (iLBC), 2004.

[27] Speex: A free codec for free speech, http://www.speex.org/.

[28] K. Brandenburg, Mp3 and aac explained, in: Audio Engineering Society 17th International Conference: High-Quality Audio Coding, 1999.

[29] Voxforge speech corpus, http://voxforge.org/.

[30] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27.

[31] D. Zongker, A. Jain, Algorithms for feature selection: An evaluation, in: Proceedings of the 13th International Conference on Pattern Recognition, vol. 2, 1996, pp. 18–22.

**Samet Hicsonmez** received his B.Sc. degree in electronics engineering from Istanbul Technical University in 2009. He is currently an M.Sc. student in computer engineering at TOBB Economics and Technology University and working in The Scientific and Technological Research Council of Turkey, Ankara. His research interests lie in audio signal & image processing and embedded systems.

**Husrev Taha Sencar** is an Assistant Professor in the Computer Engineering Department at the TOBB University of Economics and Technology, Ankara, Turkey. His research interests include digital forensics and security problems of multimedia systems. He obtained the B.Sc. and M.Sc. degrees in Electrical and Electronics engineering, respectively, from Middle East Technical University and Baskent University of Ankara, Turkey. He received Ph.D. degree in electrical engineering from the New Jersey Institute of Technology, Newark, in 2004.

**İsmail Avcıbaş** received the B.Sc. and M.Sc. degrees in electronics engineering from Uludağ University, Turkey, in 1992 and 1994, and the Ph.D. degree in electrical and electronics engineering from Boğaziçi University, Turkey, in 2001. He is currently with the Electrical and Electronics Engineering Department, Turgut Özal University, Ankara, Turkey, as a professor. His research interests include data compression, signal processing, multimedia communications and forensics.