Detecting Double Compressed AMR-format Audio Recordings

Yifeng Shen [1], *Jia Jia* [1], *Lianhong Cai* [1]

[1]Key Laboratory of Pervasive Computing, Ministry of Education
Tsinghua National Laboratory for Information Science and Technology（TNList）
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
yf-chen09@mails.tsinghua.edu.cn, jjia@mail.tsinghua.edu.cn, clh-dcs@tsinghua.edu.cn

Abstract

The Adaptive Multi-Rate (AMR) audio codec is now widely used as the default file format for various mobile phones to store spoken audio recordings. Meanwhile, more and more people start to use their mobile phones to record sounds for convenience, which results in rapid growth of the amount of digital audio recordings as evidences occur in court, including AMR format audios. It is critical to authenticating the integrity of AMR audios when they appear as evidences. AMR audio forgery manipulations generally uncompressed an AMR file, tamper with the file in PCM domain, and then re-compressed the tampered audio back into AMR format, which come the double AMR compression. In this paper, we propose a method on the detection of double AMR compression by discriminating double-compressed AMR audio recordings from single-compressed ones. Binary classification algorithm is applied for the discrimination. Statistical features related on frequency energy distribution and frequency components correlation are extracted and a support vector machine is applied to execute the classification. Experiment results show our method is promising on this binary classification task.

Index Terms: double AMR compression, digital audio forensic, SVM

1. Introduction

The Adaptive Multi-Rate (AMR) audio codec was adopted as the standard speech codec by 3GPP [1] in 1999 and is now widely used in GSM and UMTS. AMR is also a file format to store spoken audios using the AMR codec, which is now used by plenty of modern cell phones as the default audio storage format. People can record voices via their mobile phones and store in AMR audios anytime, anywhere. It follows the rapid growth of the amount of AMR-format audio recordings as evidence occurs in court, which is of extreme importance to authenticating the integrity of AMR audio recordings.

However, as an AMR format audio does not save speech waveform straightly – it encodes speech waveform into LPC coefficients and excitation codebook parameters, saves them after quantization – it's hard to tamper with an AMR format audio directly. A probably way to manipulate an AMR format audio is first uncompressing the AMR audio into raw PCM (Pulse-Code Modulation) waveform domain, applying insertion, deletion, substitution or other manipulations on the PCM waveform, then compressing back to AMR format, as the procedure shown in Figure 1. It comes double compression in AMR audios. Manipulations done throughout the double AMR compression procedure may be unperceivable and undetectable via current digital audio forensic detecting technologies. Although double AMR compression does not prove a malicious tampering, it could be a trace for checking the integrity of AMR audio recordings.

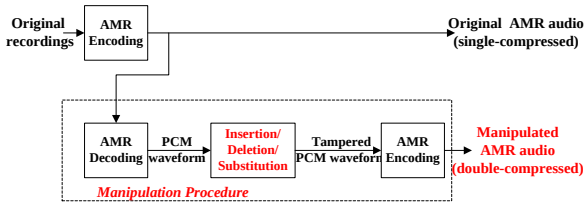

Figure 1: *Procedure of AMR audio manipulation.*

Research on digital audio forensic is just on its way. Tampering with uncompressed digital audios like PCM format waveforms may be detected by using the Electrical Network Frequency criterion [2][3][4][5], as well as high order spectral analysis [6], and method presented by A.J. Cooper via butt-spliced edits detection [7]. As to manipulation upon compressed format digital audios, the widely used MP3 format is studied in-depth, and there have been literatures published about the detection of double MP3 compression [8][9][10][11]. However, variety of other audio compressed formats has not been involved, including the detection on AMR compression.

The key in detecting double AMR compression is to discriminating double-compressed (D-compressed) AMR audios from single-compressed (S-compressed) ones. In this paper, we propose a method using support vector machine with statistical features extracted from AMR audios for the two-class classification. The framework of our task is shown in Figure 2.

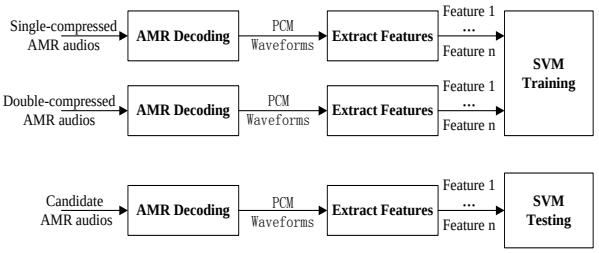

Figure 2: *Framework of detecting D-compressed AMR audio recordings.*

The rest of this paper is organized as follows. Section 2 briefly introduces the procedures of AMR encoding and decoding. In Section 3, statistical features related on frequency energy distribution and frequency components correlation for classification are presented. Experimental evaluation of the proposed features is given in Section 4. In the end, Section 5 discusses the conclusions and future work.

2. Introducing to AMR Codec

The AMR codec is designed to meet the requirement of more efficient and intelligent usage on source and channel coding rate. It has two versions: a narrow-band version and a

wide-band version. The narrow-band version AMR codec supports eight different source codec bit-rate modes: 4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.20 and 12.20kbit/s. It could switch between different bit-rate modes dynamically to select the best mode depending on the channel condition and capacity requirement. Audios used for AMR encoding must have the sampling frequency 8000Hz/13-bit, with each frame 160 samples (20 ms) length. In this paper, we are mainly concerned on the narrow-band version AMR codec.

AMR codec applies the Algebraic Codebook Excitation Linear Prediction (ACELP) technology, which is based upon code-excited linear prediction (CELP) model. The procedure of AMR encoding can be split into three parts: Linear Predictive Coding (LPC) analysis, which extracts 10 order LPC coefficients from the every 20ms length's speech clip, transfers to LSF coefficients, and then do quantization on those coefficients; pitch search, including open-loop and closed-loop pitch search, is to find the pitch delay and gain; and algebraic codebook search, which is processed to generate the codebook index and gain, as well as do quantization on the codebook gain.

The decoding procedure of AMR is reverse to its encoding. Decode the parameters (LPC coefficients, codebook vector and gain) from the AMR bit stream, synthesis the speech via source-excitation filter, pass post-filtering and post-processing to get the final reconstructed speech.

The AMR encoding and decoding procedure both have 80Hz high-pass filters, while the encoding process has another 3600Hz low-pass filter. So when uncompressing an S-compressed AMR audio to PCM waveform, the waveform passes two high-pass filters and one low-pass filter; the corresponding filter numbers are four and two when uncompressing a D-compressed AMR audio.

3. Features

AMR codec applies Linear Prediction analysis on each frame, when decoding to PCM domain, the formants of voices in the codec shifts [11]. Moreover, low-pass filters and high-pass filters also change the energy distribution in frequency domain. Statistical features related on frequency energy distribution and frequency components correlation are extracted. AMR audios are first uncompressed into PCM format waveform, say $x(n)$, and then features are calculated.

3.1. Average Subband Frequency Energy Ratio

Subband frequency energy ratio is used by researchers in audio coding, speech/music classification [12], and audio classification [13] , so on. Here we use it for our task.

$x(n)$ is sub-framed into M 160-point-length frames with step length 80-point. After windowing, STFT (Short-term Fourier Transform) is applied on each frame with FFT length 512, resulting in $X_i(k)$. The left half of each $X_i(k)$ is separated into 32 bands, and 32 averaged subband frequency energy ratios (SBR) are calculated as follows:

$$SBR(j) = \sum_{i=1}^{M} \sum_{k=8j-7}^{8j} X_i(k) \Big/ \sum_{i=1}^{M} \sum_{k=1}^{256} X_i(k), (j = 1...32) \quad (1)$$

3.2. Average Low-Frequency Subband Energy Ratio

Low-frequency subband energy ratio is to describe the characteristics of frequency intensity distribution of the low-band (from 0Hz-100Hz). Using the same procedure in Section 3.2.1 with a different FFT length 2048, we get $X_i'(k)$. Range from 0Hz to 100Hz in frequency domain of each frame is separated into 25 bands, and 25 averaged low-frequency subband energy ratios (LSBR) are calculated as follows:

$$LSBR(j) = \frac{\sum_{i=1}^{M} X_j'(k)}{\sum_{i=1}^{M} \sum_{k=1}^{25} X_i'(k)}, (j = 1, ..., 25) \quad (2)$$

3.3. Bispectrum features

Bispectrum is used to detect the presence of third-order correlations on frequency components, as well as to detect the non-linear transformation of one digital audio [6]. The definition of bispectrum is:

$$B(\omega_i, \omega_j) = \frac{\sum_{k=1}^{M} X_k(\omega_i) X_k(\omega_j) X_k^*(\omega_i + \omega_j)}{\sqrt{\sum_{k=1}^{M} \left| X_k(\omega_i) X_k(\omega_j) \right|^2 \cdot \sum_{k=1}^{M} \left| X_k^*(\omega_i + \omega_j) \right|^2}} \quad (3)$$

The frame length is 160 and FFT length is 128 to get $X_k(\omega_i)$.

Four statistical features are extracted from the bispectrum results. The mean and standard variance of $B(\omega_i, \omega_j)$ are calculated as:

$$bic_{mean} = \frac{\sum_{i=1}^{128} \sum_{j=1}^{128} \left| B(\omega_i, \omega_j) \right|}{128 * 128} \quad (4)$$

$$bic_{std} = \frac{1}{128} \sqrt{\sum_{i=1}^{128} \sum_{j=1}^{128} (\left| B(\omega_i, \omega_j) \right| - bic_{mean})^2} \quad (5)$$

The means and standard variances of phases of $B(\omega_i, \omega_j)$ are calculated as:

$$phs_{mean} = \frac{\sum_{i=1}^{128} \sum_{j=1}^{128} phase(B(\omega_i, \omega_j))}{128 * 128} \quad (6)$$

$$phs_{std} = \frac{1}{128} \sqrt{\sum_{i=1}^{128} \sum_{j=1}^{128} (phase(B(\omega_i, \omega_j)) - phs_{mean})^2} \quad (7)$$

3.4. Long-term LPC

Linear prediction spectrum is another way to describe the contour of frequency energy distribution. 10-order linear prediction is applied on the whole waveform $x(n)$ and 10 coefficients $a(i)(1 \le i \le 10)$ are calculated by minimum the sum of the squares of errors E below:

$$E = \sum_{n} (x(n) - \hat{x}(n))^2 \quad (8)$$

$$where: \hat{x}(n) = -a(1)x(n-1) - ... - a(10)x(n-10)$$

4. Evaluation and results

In this section, we present the experimental evaluation and results on the classification between D-compressed AMR audios and S-compressed ones. We use the 3GPP official AMR-NB codec [14] to do the compressing/decompressing procedures in the experiment.

4.1. Data Set

Two data sets are used for training and testing: (1) the very famous TIMIT [15] data set, which contains 6300 audio clips recorded by different people from all around the world, with each clip length 5s-10s; (2) a small data set, containing 55 12.20kbit/s AMR audio recordings, is recorded by an Android mobile phone under real environment. The small data set is used for testing and each clip is within length 5s-15s.

4.2. Experiments

4.2.1. Training and testing on TIMIT

Pre-processing is executed on TIMIT to meet the requirement of AMR codec. Each audio clip in TIMIT is transferred from 16KHz/16bit mono NIST format to 8KHz/16bit mono PCM format using SoX [16].

To each transferred PCM format audio clip, we compress it into bitrate 4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.20 and 12.20kbit/s AMR format audios respectively, which are called as S-compressed AMR audios. Then we de-compress the 8 AMR audios, and re-compress back into AMR format with those 8 different bitrates, which are called as D-compressed ones. Thus we get 8 S-compressed AMR audios and 64 D-compressed AMR audios from one original PCM audio clip. Repeating the above process to all 6300 audio clips, finally we get 6300*72=453600 AMR format audio clips for the experiment, in which 50400 are S-compressed and 403200 are D-compressed.

Denote A_{b1} as the set of S-compressed AMR audios compressed in bitrate b1, which has 6300 audios; A_{b2-b1} as the set of D-compressed AMR audios that is first compressed in bitrate b2, and then re-compressed in bitrate b1, which also has 6300 audios. So to each pair of bitrates b1 and b2, we have the corresponding set A_{b1} and A_{b2-b1}. Combine these two sets into one set and denote it as $S_{b2,b1}$. As there are 8 different bitrates, we get 64 sets totally. Then to each set $S_{b2,b1}$, we do following steps:

Step 1: De-compress the 12,600 AMR audios of the set to PCM format respectively, and then extract the 4 types of features proposed above: 32 dimensions averaged subband frequency energy rate $SBR(j)$, 25 dimensions averaged low-frequency subband energy rate $LSBR(j)$, 4 dimensions bispectrum features (bic_{mean}, bic_{std}, phs_{mean} and phs_{std}), and 10 dimension long-term LPC coefficients $a(j)$. Each AMR audio in the set is now reflect to a 73-dimension feature vector. So we get a feature set with 12600 73-dimension feature vectors.

Step 2: Apply the welcome machine-learning tool LibSVM package [17] on the feature set to perform the classification experiment. The standard RBF Kernel is chosen. 70% of the feature set is randomly chosen for training and the remaining 30% for testing. The classification experiment is repeated 20 times and average testing accuracy is obtained. The testing accuracy is consisted of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) and calculated as:

$$\beta \cdot \frac{TP}{TP+FN} + (1-\beta) \cdot \frac{TN}{TN+FP} \quad (9)$$

Without loss of generality, β is set to 0.5.

Step 3: Randomly selecting one-eighth audios of each A_{b2-b1} set over all 8 sets, which are consisting of AMR audios with 2nd-compression-bitrate b1, combined with audios of A_{b1}, we get a new set S_{b1} with 6300 S-compressed AMR audios and 6300 D-compressed audios. Extract features from the AMR audios as in Step 1, we get a feature set with 12600 73-dimension feature vectors. SVM is applied on the feature set, as in Step 2. 70% of the feature set is randomly chosen for training and the remaining 30% for testing. The classification is repeated 20 times and average precision, average recall and average testing accuracy (defined above) are obtained. Precision is defined as $TP/(TP+FP)$, while recall is defined as $TP/(TP+TN)$.

4.2.2. Testing on the real data set

The 55 12.20kbit/s AMR audio recordings in the small data set are recorded under real environment, and are used to test the performance of the features proposed above. Each audio clip is first de-compressed into PCM waveform, and then re-compressed back to AMR format with 8 different bitrates. So there are 8 D-compressed AMR audios corresponding to an original S-compressed AMR audio. Then to each bitrate b1, we use the 12600 AMR audios of set S_{b1} to train a model and test on the real data set (110 AMR audio clips in 12.20kbit/s, 55 otherwise). The procedure of testing on a candidate AMR audio is shown in Figure 3.

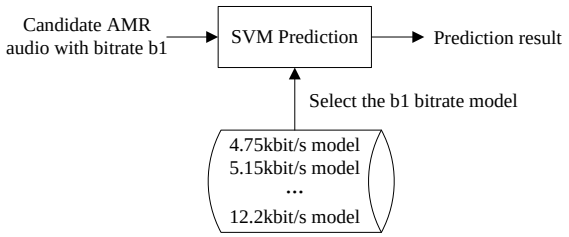

Figure 3: *Procedure of Detecting a AMR audio clip*

4.3. Results

4.3.1. Results on subsets $S_{b2,b1}$

Table 1 lists results of the average testing accuracy over twenty experiments on each subset. Each element in the first row "Bitrate1" denotes the bitrates of S-compressed AMR audios, as well as the 2nd-time-compressing bitrates of D-compressed AMR audios. Elements in the first column "Bitrate2" then denote the 1st-time-compressing bitrates of D-compressed AMR audios.

Bitrate2 (kbit/s)	Bitrate1 (kbit/s) 4.75	5.15	5.90	6.70	7.40	7.95	10.2	12.2
4.75	88.68	89.44	92.23	92.39	93.37	92.90	95.08	95.76
5.15	88.11	88.84	92.13	92.49	93.35	92.62	95.00	95.72
5.90	85.98	86.81	88.32	88.30	90.10	89.73	92.65	93.91
6.70	84.61	85.62	87.73	87.49	89.02	88.31	92.33	93.05
7.40	82.74	83.71	85.94	84.37	83.52	85.29	86.65	87.98
7.95	81.66	82.84	85.06	84.12	84.00	83.92	88.00	89.05
10.2	77.11	77.98	78.68	78.16	77.57	78.51	79.19	80.41
12.2	89.18	88.89	88.58	87.05	83.14	84.44	79.22	78.99

Table 1. *Average Testing Accuracy (%) over twenty classification experiments for all subsets* $S_{b2,b1}$

In table 1, the testing result 93.37%, corresponding to 1st-time-compressing in 4.75kbit/s and 2nd-time-compressing in 7.4kbit/s, is the accuracy on the classification between the 7.4kbit/s S-compressed AMR audios and the 7.4kbit/s D-compressed AMR audios that were actually trans-coded from the 4.75kbit/s ones. The testing result 77.98%, corresponding to 1st-time-compressing in 10.2kbit/s and 2nd-time-compressing in 5.15kbit/s, is the accuracy on the classification between the 5.15kbit/s S-compressed AMR audios and the 5.15kbit/s D-compressed AMR audios that were actually trans-coded from 10.2kbit/s, and so on.

The experimental results show high accuracy in the upper-right corner of table 1, which are higher than 92%. While in the lower-left corner of table 1, the accuracy are much lower, under 80%. This means D-compressed AMR audios which are trans-coded from low bitrates to high bitrates are more discriminative than those trans-coded from high bitrates to low. Except for bitrate 12.2 which may be caused by the distinctive AMR encoding procedure in 12.2kbit/s comparing to other bitrates, in which procedure each frame is applied with LP analysis twice and 20 LPC coefficients are calculated and stored.

4.3.2. Results on sets S_{b1}

Bitrates (kbit/s)	Average Precision (%)	Average Recall (%)	Average Accuracy (%)
4.75	78.84	88.63	79.76
5.15	82.42	88.12	83.73
5.90	86.15	88.80	87.26
6.70	85.42	87.22	86.17
7.40	79.31	88.80	80.14
7.95	81.48	88.58	82.36
10.2	84.60	88.75	85.31
12.2	87.59	88.89	88.16

Table 2. *Average precision, recall and accuracy over twenty classification experiments for sets* S_{b1}

Table 2 lists results of the average precision, recall and accuracy over twenty classification experiments on each set S_{b1}. For instance, to bitrate b1=5.90kbit/s, the testing average precision of S_{b1} is 86.15%, the average recall is 88.80% and the average accuracy is 87.26%.

Results show that the value of average recall is higher than average precision to each bitrate, which means the trained model are more likely to judge a candidate AMR audio as a D-compressed audio rather than an S-compressed audio. Moreover, all values of average precision are above 78%, especially with bitrates 5.90, 6.70, 12.2 above 85%, which indicates good discrimination between D-compressed AMR audios and S-compressed ones.

4.3.3. Results on the real data set

Bitrates (kbit/s)	Correct/Total	Precision
4.75	48/55	87.27%
5.15	41/55	74.55%
5.90	44/55	80.00%
6.70	40/55	72.73%
7.40	41/55	74.55%
7.95	43/55	78.18%
10.2	47/55	85.45%
12.2	79/110	71.82%

Table 3. *Precision on the real data set.*

Table 3 shows the precisions of applying specific bitrate models on real AMR audio recordings recorded by mobile phones with the same bitrate. For example, 87.27% in the 2nd row 3rd column stands for the precision of applying the 4.75kbit/s model on the real 4.75kbit/s AMR audio recordings, which mean 48 of the 55 D-compressed AMR audios are detected as D-compressed ones, while the rest 7 are wrongly detected as S-compressed.

Experimental results show the results are slightly worse than listed in table 2, but still achieve acceptable performance with all precisions of the 8 bitrates above 71%, which can be used as a trace for further detecting on the candidate D-compressed AMR audios.

5. Conclusions and future work

In this paper, we have proposed a method on detecting D-compressed AMR audio recordings. We discriminate D-compressed AMR audios from S-compressed by extracting frequency energy distribution and frequency components correlation related features from the uncompressed PCM waveforms, and then using a support vector machine to perform the classification. Experimental results have shown promising effectiveness of our method in discrimination between the D-compressed AMR audio recordings and S-compressed ones.

However, the performance is not so promising when applying the training model on real data. It's still a long way to meet the requirement of authentication and integrity on AMR audio recordings. More discriminated features extracted from AMR codec procedure should be investigated, to meet the requirement of detecting on D-compressed AMR audio recordings from the real world.

6. Acknowledgment

This work is supported by National Natural, and Science Foundation of China (61003094, 90920302). And this work is supported by Tsinghua - Tencent Joint Laboratory for Internet Innovation Technology.

7. References

[1] 3GPP: 3rd Generation Partnership Project. http://www.3gpp.org/

[2] C. Grigoras, "Digital Audio Recording Analysis: The Electric Network Frequency (ENF) Criterion", The International Journal of Speech Language and the Law, vol. 12, no. 1, pp. 63-76,2005.

[3] C. Grigoras, "Applications of ENF Analysis in Forensic Authentication of Digital Audio and Video Recordings", J. Audio Eng Soc, vol. 57, No.9, pp. 643-661, (2009 Sep).

[4] A. J. Cooper, "The Electric Network Frequency (ENF) as an aid to Authenticating Forensic Digital Audio Recordings – An automated Approach" AES 33rd International Conference Audio Forensics Theory and Practice, Denver, CO, USA, (2008, Jun).

[5] D. P. Nicolalde, J. A. Apolinario Jr, "Evaluating Digital Audio Authenticity with Spectral Distances and Phase Change", IEEE ICASSP, pp. 1417-1420, (2009).

[6] H.Farid. Detecting Digital Forgeries Using Bispectral Analysis. MIT AI Memo AIM-1657, MIT, (1999).

[7] A.J Cooper. Detecting butt-spliced edits in forensic digital audio recordings. AES 39th International Conference, Hillerød, Denmark, 2010 June 17–19.

[8] Yang R, Qu Z, Huang J. Detecting digital audio forgeries by checking frame offsets. MM&Sec 2008: 21–26; (2008).

[9] Yang R, Shi YQ, Huang J. Defeating fake-quality MP3. MM&Sec 2009: 117–124; (2009).

[10] Q Liu, AH Sung, M Qiao. Detection of Double MP3 Compression. Cognitive Computation 2010:291-296;(2010).

[11] Yang R, Qu Z, Huang J. Detecting double compression of audio signals. Proc. SPIE 7541, 75410K (2010).

[12] Classification of speech and music using sub-band energy. Patents Application Publication May 5,2005.

[13] L Lu, HJ Zhang and Stan Z. Li. Content-based audio classification and segmentation by using support vector machines. Multimedia Systems, 2003.

[14] 3GPP TS26.073 ANSI-C code for the Adaptive Multi Rate (AMR) speech codec. http://www.3gpp.org/ftp/Specs/html-info/26073.htm

[15] TIMIT Acoustic-Phonetic Continuous Speech Corpus. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1.

[16] SoX: Sound eXchange Project. http://sox.sourceforge.net/

[17] C.-W. Hsu, C-C. Chang and C. J. Lin, LIBSVM: a library for support vector machines, 2001.

[18] B.J.Guillemin, C.I.Watson. Impact of the GSM AMR Speech Codec on Formant Information Important to Forensic Speaker Identification. ASSTA, 2006.