DOUBLE COMPRESSED AMR AUDIO DETECTION USING LINEAR PREDICTION COEFFICIENTS AND SUPPORT VECTOR MACHINE

JOSÉ FABRIZIO P. SAMPAIO[1,2] AND FRANCISCO ASSIS O. NASCIMENTO[2]

[1]*Brazilian Federal Police – National Institute of Criminalistics*
SPO Lote 7, Setores Complementares, Brasília-DF, 70610-902
E-mail: fabrizio.jfps@dpf.gov.br

[2]*University of Brasilia – Electrical Engineering Department*
Faculdade de Tecnologia, Campus Darcy Ribeiro, Braília-DF, 70910-900
E-mail: assis@ene.unb.br

Abstract— The adaptive multirate codec (AMR) standard usage has been intensive in mobile networks as well as a speech storage format. Due to its high availability, many AMR audio files take on a forensic evidence condition, which implies the need to demonstrate they are authentic. Detecting AMR double compression means that, in the multimedia forensics context, the file is not an original one and a tampering likely happened, since it is always necessary to decode and encode again to change utterance meaning. In this paper, we show a new method to detect AMR double compression based on compressed-domain linear prediction (LP) coefficient extraction, statistical feature computation from LP coefficients and SVM application. The experiments demonstrate the proposed method can discriminate double compressed AMR files from single compressed files with satisfactory accuracy, either using mixed first compression bitrate sets, or fixed first compression bitrate ones. Using the TIMIT corpus, the average accuracy reached 93.66 %, which is a very satisfactory result.

Keywords— AMR codec, double compression, SVM, forensic authentication.

Resumo— O uso do codec padrão AMR (multitaxa adaptativo) tem se mostrado muito intenso nas redes móveis e também como formato de armazenamento de sinal de voz. Devido à alta disponibilidade desse codec, muitos arquivos de áudio nesse formato têm assumido a condição de prova forense, o que implica a necessidade de se demonstrar que são autênticos. Detectar a dupla compressão AMR significa dizer, no contexto da multimídia forense, que o arquivo não é original e que provavelmente houve edição do conteúdo, uma vez que sempre é necessário decodificar e codificar novamente para editar. Neste artigo, nós apresentamos um novo método para detectar a dupla compressão AMR baseado na extração de coeficientes de predição linear no domínio da compressão, cálculo de características estatísticas desses coeficientes e no uso de máquina de vetor suporte (SVM). Os experimentos demonstram que o método proposto é capaz de discriminar arquivos AMR com dupla compressão com boa taxa de acerto, tanto com conjuntos com primeira taxa de bits de compressão misturada, quanto com primeira taxa de bits fixa. Utilizando o corpus de referência TIMIT, a taxa de acerto média atingida foi de 93,66 %, que é um resultado bastante satisfatório.

Palavras-chave— codec AMR, dupla compressão, SVM, autenticação forense.

1 Introduction

Since the advent of digital audio, the task of authenticating speech recordings has become a complex and an audio generation dependent problem. It is actually easy to tamper with digital audio and, in contrast, speech evidence must be authentic to be admitted in court. Such a fact requires a myriad of techniques from forensic examiners to report digital audio authentication, like the passive ones, which take part of the so called multimedia forensics field. Passive techniques use only information extracted from the audio file instead of ancillary authentication bitstream, like watermarking.

Double compression detection is a part of the more general compression history identification issue. The related works show that the previous compression traces are essential to determine authenticity and they can be disclosed by processing the encoded or decoded signal to extract statistics or some encoder specialty. By processing encoded MP3, fake-quality MP3 can be enlightened (Yang, 2009) because it has fewer MDCT coefficients of small values than unaltered MP3 audio. MP3 double compression detection was first accomplished by applying an SVM to statistical features extracted from MDCT coefficients (Liu, 2010). Since MP3 is a widely used codec, detecting MP3 double compression is an important problem of compression history for forensic authentication purposes. The analysis of chaotic features extracted from encoded audio can give valuable information about compression history for different encoders (Hiçsönmez, 2011), while the decoded audio can be used to detect the speech codec with very low misidentification (Jenner, 2012). By using SVM to analyze statistical features of MDCT coefficients, the compression history for MP3 and WMA files can be traced as well as bitrate transcoding (Luo, 2012). Specifically about double compression with different codecs, like AAC, MP3 and AMR, an SVM-based method can identify this procedure and the respective bitrates before and after, with satisfactory accuracy (Hiçsönmez, 2013). If the audio specimen is an uncompressed WAV file, the compression

history can be identified, including codec and bitrate determination (Luo, 2014a).

The AMR codec compression history identification is better explored in terms of double compression detection. The first known specific method to detect AMR double compression applied features extracted from decompressed audio to an SVM classifier using TIMIT database to compute accuracies (Shen, 2012). Such results were surpassed by deep learning technique which also used decompressed AMR audio and reached higher accuracies (Luo, 2014b). The state-of-the-art was reached by using a stacked autoencoder neural network to detect AMR double compressed audio also using the decompressed AMR version (Luo, 2017). Such a technique made possible accuracies about 98% with TIMIT database.

In this work, a new passive method based on compressed-domain features is proposed to detect AMR double compression via SVM. The method uses encoded audio instead of decoded speech to extract linear prediction (LP) coefficients and compute their statistics. Such compressed-domain approach has already been applied to AMR codec for speaker recognition task, but not for double compression detection. To the best of our knowledge, our method is the first to use LP coefficient extraction from encoded speech to detect double compressed AMR audio.

This article is organized in five sections. In Section 2, we provide useful information upon AMR algorithm and analyze double compression effects over bitstream quantized LP coefficients. Section 3 gives details about the methodology for LP coefficient extraction and feature computation via SVM. We explain how to extract LP coefficients from AMR files, how to generate statistical compressed-domain features and how to select the SVM models. In Section 4 we explain experiment set up and reveal the results comparing them to previous works. We reach conclusions in Section 5 and also state future work to improve our method.

2 AMR Codec and Double Compression

The detection of double compression is a true indicative of forgery in AMR digital audio. Like in MP3 encoder, whenever the AMR audio is found to be double compressed, a time-domain conversion happened with or without a bitrate transcoding. We can face AMR double compression detection inspired by MP3 double compression detection, but, instead of MDCT statistics, we should pursue LP coefficient statistics.

When it comes to mobile communication, the AMR codec is a widely used compression standard for speech in 3G and 4G networks and in voice massage apps (3GPP, 2017). Not less important, AMR is a file format and file extension designed to store encoded speech, which can be recorded and played by almost all mobile phone worldwide. The AMR codec considered in this work was engineered for commuted circuit mobile networks, transmitting 3,200 Hz narrowband speech with sampling rate 8 kHz. As the name implies, AMR codec can encode in eight modes (bitrates) depending on channel conditions. The bitrates are 12.2, 10.2, 7.95, 7.4, 6.7, 5.9, 5.15 and 4.75 kbits/s, which corresponds to modes MR122, MR102, MR79, MR74, MR67, MR59, MR515 and MR475 respectively. In this paper, we focus on AMR file format with constant bitrate (one mode) identified by file header reading.

AMR encoder algorithm MR-ACELP (Multi-Rate Algebraic Code Excited Linear Prediction) is based on a 10th order LP filter with coefficients a_i, i=1...10, applied to each of 4 subframes of a 160 sample frame. These LP coefficients are converted to line spectrum pairs (LSP) and line spectrum frequencies (LSF) by means of a nonlinear operation. When speech is encoded, the AMR bitstream consists of quantized parameters whose bit allocation depends on AMR mode. If we inspect such parameters, we find out that the quantized LSF subvectors are the only ones directly related to LP coefficients. Double compression operation might affect LSF subvectors statistics and point out double compressed files. The statistics in the histogram shown in Figure 1 reveal that the differences occur, but they are too subtle to be promptly employed to build features. As we can

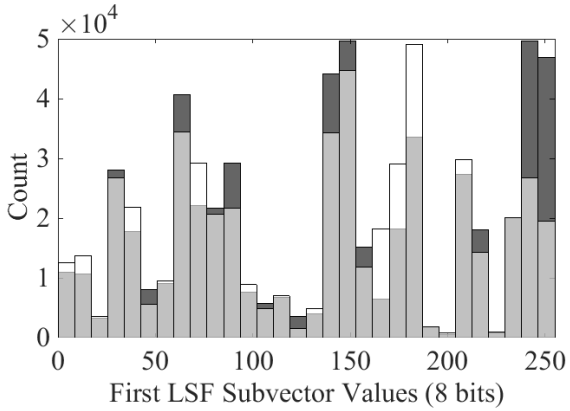

Figure 1. Histograms of single compressed file 1st LSF subvectors (black bars) and of double compressed file 1st LSF subvectors (white bars) extracted from encoded AMR over 6300 single and 6300 double compressed AMR files at MR475 mode. The gray bars indicate overlapped histograms.

see, the single compressed file 1st LSF subvector histogram (black bars) is not much different from double compressed file 1st LSF subvector histogram (white bars), noting that the gray bars indicate overlapped histograms.

This analysis shows that we better extract the unquantized LP coefficients to build features, since quantization may change LP coefficient double compression discrimination power. As we show in Section 3, a useful approach is to partly decode AMR files to extract unquantized LP and LSP coefficients and use them to compute compressed-domains features.

3 Methodology for LP Coefficient Extraction and Feature Computation using SVM

Instead of processing decoded AMR audio to detect double compression, we propose a method to compute features based on compressed AMR, i.e., we discriminate double compressed AMR by using compressed-domain features. In fact, compressed-domain feature computation is a quite common approach. For instance, it is used to classify sound in MPEG-1 bitstream (Pfeiffer, 2003) and to implement an automatic speaker recognition system based on AMR compressed-domain features (Petracca, 2005). The aforementioned works used basic statistics over LSF indexes extracted directly from AMR bitstream, which encourages us to formulate a compressed-domain feature algorithm focused on discriminating double compressed AMR audio. In the block diagram depicted in Figure 2, we present the proposed method which is based on an SVM neural network to classify AMR audio.

Considering a speech corpus with N uncompressed WAV files, we double compress each of them at a second bitrate BR2 using the double compression module, generating $2N$ audio files (N single compressed S-AMR and N double compressed D-AMR) for each of the eight AMR modes. Inspired by the methodology proposed by Shen *et al* (Shen, 2012), we assume two kinds of AMR file sets: the S_{B1B2}, whose first compression bitrate BR1 is the same for all N double compressed files, and S_{BmB2}, whose first compression bitrate B_m may assume all the eight possible AMR bitrates, that is to say, about $N/8$ files are first compressed at each AMR bitrate. Therefore, we generate 8 sets of type S_{BmB2}, one for each BR2, and 64 sets of type S_{B1B2}, because BR1 and BR2 may be set to eight possible bitrates.

As long as a given AMR set is computed, we are able to extract compressed-domain LP and LSP coefficients of each file without the need to decode it to waveform. Such coefficients are essential to compute 572 statistical features, i.e., a given AMR file corresponds to a 572-dimension vector. From this point on, we compute a feature matrix with $2N$ rows and 572 columns but, on account of LP or LSP coefficients distributions, some columns may carry no useful information because they have null or constant values. This situation leads to the deletion of some features depending on AMR bitrate and kind of set, resulting in a number of current features (NCF) such that NCF $\leq$ 572. We then compute a $2N$ x NCF feature matrix and reach a new experiment starting

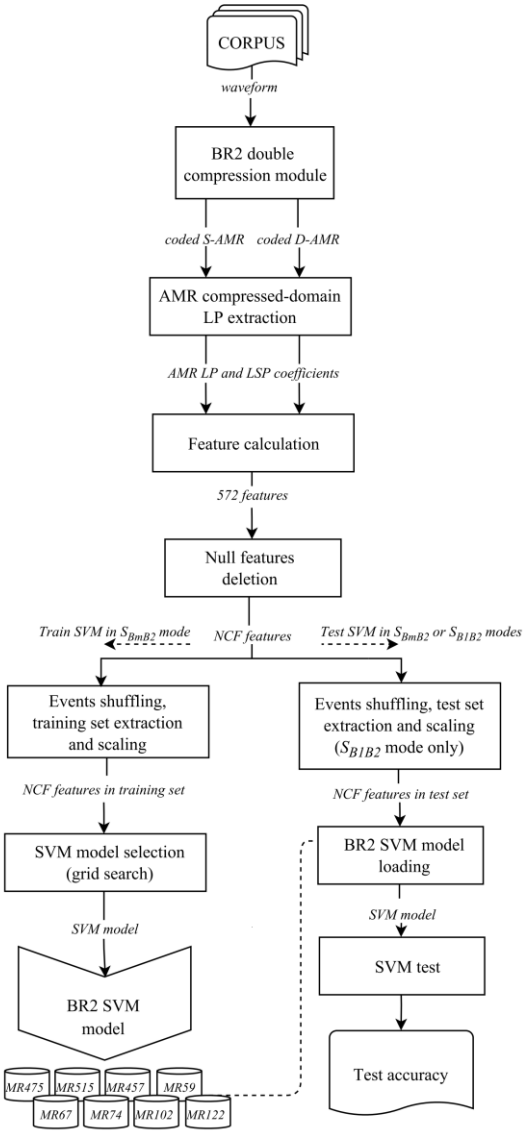

Figure 2. Block diagram of the proposed method.

point, i.e., from this point onwards we can extract different training and test sets by shuffling the feature matrix, thus reaching different models and verifying different accuracies. After scaling training and test sets, we use the training set for SVM model selection by grid search based on maximum cross-validation accuracies. Once concluded the training, we store 8 AMR bitrate models for testing both S_{BmB2} and S_{B1B2} sets. We show the method details in the following subsections.

3.1 Extraction of LP Coefficients

We extract the unquantized LP coefficients by using a modified AMR decoder instead of collecting quantized LSF subvectors from bitstream, enlightening double compression clues in LP coefficients. Starting from the unaltered 3GPP source code of AMR decoder, we deliberately added few lines to extract the LP (*A_t[]* variable) and LSP (*lsp[]* variable) coefficients after decoding a given file, i.e., we partly decode AMR files to generate two raw binary files which contain the coefficients. At the end of

AMR decoding, we store the binary files, whose sizes depend on AMR file durations.

3.2 Statistical Feature Computation

We create statistical features using extracted LP and LSP coefficients in order to discriminate double compressed from single compressed AMR files. Such features are simple, but we intentionally designed a large number of them because double compression requires an in-depth description to be detected. Table I shows the definition of all the used statistical measures where we introduce the normalized first digit frequencies as the LP and LSP coefficient first digit histogram frequencies. When it comes to MP3 encoder, we know that first digit probabilities of single compressed MDCT coefficients obey Benford's law while double compressed MDCT coefficients disobey (Yang, 2010). This finding inspired us to verify first digit probability behavior of LP and LSP coefficients, leading to the conclusion that neither single compressed nor double compressed coefficients obey Benford's law. We verify, however, that single compressed AMR LP coefficient first digit probabilities are different from double compressed ones, i. e., they are useful to build features do detect double compression.

Table 1. Definition of Statistical Measures

Measure	Definition
$\bar{x}$	sample mean
σ^2	variance
σ	standard deviation
Mo	mode
kurt	kurtosis
γ_1	skewness
$\tilde{x}$	median
max	maximum of samples
min	minimum of samples
CV	coefficient of variation
$\bar{x}_{geom}$	geometric mean
$\bar{x}_{harm}$	harmonic mean
meanabs	mean of absolute elements
meansqr	mean of squared elements
ADev	mean absolute deviation
MAD	median absolute deviation
trimmean	mean excluding outliers trimmed at 5%
$m_y(x)$	normalized first digit frequency of y, $x=1...9$

Each LP coefficient a_i, each LSP coefficient q_i, i=1...10, and all ten LP ($a_1, a_2, \ldots, a_{10}$ together) and ten LSP ($q_1, q_2, \ldots, q_{10}$ together) coefficients may be considered for computing the statistics in Table 1. By doing so, we can calculate features for 10 LP and 10 LSP coefficients individually plus one set of ten LP and one set of LSP coefficients, totalizing 22 possible parameters. Considering there are 26 measures in Table 1 (note that there are 9 digits to calculate the normalized first digit frequency), we can compute $22 \times 26 = 572$ possible features to compose the feature matrix columns.

3.3 Feature Deletion and Scaling

We verify that the feature matrix needs to be processed after its computation because some features carry no useful information. This observation is related to numerical properties of LP and LSP coefficients, like the absence of some first digits which results in zero frequency or the presence of sparse non-null values in the feature. We adopt the following criteria to purge features: either the feature has zero standard deviation in single compressed half or in double compressed half, or the number of zero elements exceeds an empirical tolerance (we use 99%). By deleting such features, we compute a new feature matrix with a lower number of columns which corresponds to the NCF for a given AMR bitrate.

We initially assemble the feature matrix with first half accounting for single compressed AMR files and the second half for the double compressed files. The training matrix is thus extracted from a fraction of first half of feature matrix (e.g., 70%) and the same for second half, while the test matrix is extracted from the remaining events. The way the events are ordered defines one experiment to compute, i.e., whenever we shuffle events we extract different training and test matrices for different experiments, observing that we have to shuffle the single compressed half and the double compressed half in the same way so as to guarantee in training and test matrices the single and the double compressed versions of a given event.

After defining training and test matrices, we make scaling procedures according to the well-known min-max algorithm, keeping feature values between -1 and $+1$, described by Equation 1:

$$y = -1 + 2\frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x_{min} is the minimum value of the feature and x_{max} is its maximum value.

3.4 SVM Model Selection

Once computing the scaled training and test matrices, an SVM model for each BR2 has to be chosen. For each experiment, we perform a grid search to find optimized penalty parameter C and gamma SVM parameters based on n-fold cross-validation maximum accuracy, since we use the RBF kernel for SVM computation, according to the methodology suggested in (Chang, 2011) and due to lower performance observed using other kernels. Starting from a fixed grid for C and gamma, we perform a loose search and two fine searches by splitting the neighborhoods of previous maximum accuracy grid points. We only proceed with the model selection for S_{BmB2} sets and we use such models to compute the S_{B1B2} sets, jointly with the experiment training matrix (used only for scaling) and the permutation used for training matrix extraction (this is necessary to prevent using the same events in training for testing).

4 Experiments

We compute a series of three complete experiments to confirm method effectiveness. We give detailed information about implementation in the following subsections and discuss the performance in contrast with published works.

4.1 Experimental Configuration

We consider the methodology proposed by Shen *et al.* (Shen, 2012) and the original TIMIT speech corpus (Garofolo, 1993) in all experiments. The TIMIT corpus consists of 6,300 uncompressed files which contain ten sentences spoken by native USA speakers and last between 915 ms and 7,888 ms. For AMR double compression detection algorithm, we filter and resample files at 8 kHz with 16 bits sample size. We compute eight AMR models for each experiment and use them to test eight S_{BmB2} sets and 64 S_{B1B2} sets. Starting from the TIMIT corpus, we generate 6,300 single compressed AMR files and 6,300 double compressed AMR files, observing that BR1 is assigned to one of the eight possible AMR bitrates for S_{BmB2} sets. We split the TIMIT corpus in approximately eight fractions so as to guarantee BR1 assumes all the AMR modes. After double compression procedure and feature computation, the feature matrix has 12,600 rows and 572 columns (or NCF columns after feature deletion).

At the beginning of each experiment and before assembling the training and test matrices, we shuffle the first and second halves of feature matrix in the same way. We set 70% for training ratio so as to extract 4,410 rows of single compressed features and 4,410 of double compressed ones, making 8,820 row training matrix and 3,780 row test matrix. We define a label vector for training matrix with 4,410 rows labeled $+1$ representing single compressed features, while 4,410 rows labeled -1 representing double compressed features. The label vector for test matrix is defined the same way with 1,890 labels $+1$ and 1,890 labels -1. Before proceeding to test, the test matrix and its labels are randomly shuffled in the same way.

We scale the training and test matrices using the min-max algorithm depicted in Equation 1. After scaling the training matrix, the used parameters x_{min} and x_{max} of each feature are stored to scale the test matrix in order to avoid bad scaling and SVM performance degradation.

The SVM algorithm used in this paper is the same described in the package LibSVM (Chang, 2011) which can be downloaded from (LIBSVM, 2018).

4.2 Parameter Selection for SVM

We use the grid search method to find SVM parameters for each experiment. This procedure searches for C and gamma from a 5-fold maximum cross-validation accuracy criterion. Starting from a 12 x 12 grid of power-of-2 values for C and gamma, we proceed with a loose search, find the pair that gives the highest accuracy, make a new 12 x 12 grid in the neighborhood of such a pair and make a fine search, repeating this process once again to make a second fine search. At the end, we find the best C and gamma that maximizes the cross-validation accuracy for a given training matrix and use them do train SVM.

4.3 Method Performance for Discriminating Single/Double Compressed AMR files

We compute each experiment starting from SVM models for each BR2 and then use them to test S_{BmB2} and S_{B1B2} sets. The test accuracy is defined as:

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \tag{2}$$

where TP is the total true positive cases (label -1 and predicted -1), TN is the total true negative cases (label 1 and predicted 1), FP is the total false positive cases (label 1 and predicted -1), FN is the total false negative cases (label -1 and predicted 1). We also calculate the double compressed file detection rate TP[%] and the single compressed file detection rate TN[%] as follows:

$$TP[\%] = \frac{TP}{TP + FN} \tag{3}$$

$$TN[\%] = \frac{TN}{TN + FP} \tag{4}$$

Table 2 shows the average results for S_{BmB2} sets. We can observe that the proposed method is effective in discriminating double compressed AMR files from single compressed ones. The average accuracy over all bitrates is 93.66%, the average detection rate of double compressed AMR files (TP) is 93.44% and the average detection rate of single compressed AMR files (TN) is 93.87%.

Table 2. Average Test Accuracies for BR2 SVM Models (S_{BmB2} sets with TIMIT Speech Corpus, bitrates in kbits/s)

BR2	Acc [%]	TP[%]	TN[%]
4.75	92.62	92.63	92.61
5.15	92.53	92.49	92.57
5.9	94.03	94.06	94.00
6.7	93.80	93.74	93.86
7.4	93.72	92.82	94.62
7.95	94.00	93.81	94.18
10.2	93.99	93.70	94.28
12.2	94.58	94.32	94.85

We also compute for each experiment all the 64 accuracies for S_{B1B2} sets using the eight SVM models employed in S_{BmB2} accuracy computations. Such results are useful to see the model performance with

Table 3. Average Test Accuracies for BR2 SVM Models (S_{B1B2} sets with TIMIT Speech Corpus, bitrates in kbits/s)

	BR2							
BR1	4.75	5.15	5.9	6.7	7.4	7.95	10.2	12.2
4.75	95.57	95.46	96.87	96.70	97.08	**96.93**	**96.94**	**97.29**
5.15	95.54	95.57	96.78	96.68	97.07	96.97	**96.92**	**97.31**
5.9	94.43	94.40	95.43	95.48	95.73	95.70	95.48	**96.36**
6.7	94.25	94.27	95.15	95.24	95.51	95.48	95.23	96.21
7.4	93.91	93.93	95.05	95.33	95.33	95.29	95.08	96.00
7.95	93.97	93.94	95.08	95.11	95.30	95.13	95.36	95.92
10.2	86.66	86.38	87.57	87.76	87.28	87.95	87.87	90.10
12.2	87.94	87.91	89.44	89.71	89.03	89.44	89.42	87.63

constant BR1 in the training and test sets. We can see the results in Table 3 as average accuracies for all experiments while we also observe the models are effective to discriminate double compressed AMR files for S_{B1B2} sets with average accuracy 93.78%. Like in Shen *et. al* (Shen, 2012), Table 3 shows that the accuracies increase (accuracies in bold at upper right corner) if an up-transcoding operations is done (BR1 < BR2) and, in contrast, they decrease (accuracies at lower left corner) if a down-transcoding operation takes place (BR1 > BR2). Such behavior is expected because the down-transcoding operation loses useful information for double compressed AMR detection.

4.4 Comparison with State-of-the-Art

Our literature research identifies three methods to detect double compressed AMR audio, but all of them use uncompressed audio. Table 4 compares such techniques with the proposed method and we can affirm that, except for stacked autoencoder neural network (Luo, 2017), it outperforms all of them in all bitrates.

Table 4. Comparative Performance Evaluation for S_{BmB2} sets with TIMIT Speech Corpus (bitrates in kbits/s).

	Average Accuracies [%]			
	Methods			
AMR Birate	Shen, 2012	Luo, 2014	Luo, 2017	Proposed Method
4.75	79.76	91.14	98.78	**92.62**
5.15	83.73	91.32	98.88	**92.53**
5.90	87.26	91.18	98.74	**94.03**
6.70	86.17	91.23	98.77	**93.80**
7.40	80.14	91.39	98.95	**93.72**
7.95	82.36	91.28	98.87	**94.00**
10.2	85.31	91.27	98.84	**93.99**
12.2	88.16	91.33	98.82	**94.58**

5 Conclusions and Future Work

Whenever an AMR audio file is classified as double compressed, its authenticity becomes a question raised because an original AMR file should be single compressed. Double compressed AMR detection is a complex multimedia forensics problem which solution is still in progress.

In this paper, we propose a novel SVM-based algorithm for double compressed AMR detection which offers a higher performance than deep learning techniques. Our method innovates because we use LP-based compressed-domain features instead of decompressed audio to detect AMR double compressed files.

We believe the proposed method is a promising technique to detect double compressed AMR audio, since it is capable of getting around time domain transients. The LP and LSP coefficients can translate, in a compact way, some spectral differences between single and double compressed AMR audio, since we achieve a higher performance than deep learning algorithms.

For future work, we intend to improve the proposed method by introducing some modifications. First, we can extract from AMR encoded files more information, such as pitch lags. Second, a feature selection algorithm can be used because there must be an optimum number of features which increases detection accuracy. And third, at last, we should investigate the features so as to understand its statistical behavior which could affect SVM performance.

Acknowledgement

This work was supported in part by the Brazilian Federal Police.

References

3GPP (2017). AMR Codec. [online] Available at: http://www.3gpp.org/ftp/Specs/archive/26_series/26.104/ [Accessed 10 Dec. 2017].

Chang, CC. and Lin, CJ. (2011). LIBSVM: a Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology, Vol. 2, Issue 3, Article No. 27.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., Dahlgren, N. L. and Zue, V. (1993). TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. [online] Available at:

https://catalog.ldc.upenn.edu/topten [Accessed 10 Dec. 2017].

Hiçsönmez, S., Sencar, H. T. and Avcibas, I. (2011). Audio Codec Identification through Payload Sampling. In: Int. Workshop Inf. Forensics Secur., Iguacu Falls, Brazil, pp. 1–6.

Hiçsönmez, S., Uzon, E. and Sencar, H. T. (2013). Methods for Identifying Traces of Compression in Audio. In: Proc. 1st Int. Conf. Commun., Signal Process., Appl., Sharjah, United Arab Emirates, pp. 1–6.

Jenner, F. and Kwasinski, A. (2012). Highly Accurate Non-intrusive Speech Forensics for Codec Identifications from Observed Decoded Signals. In: IEEE Int. Conf. Acoust., Speech Signal Process., Kyoto, Japan, pp. 1737–1740.

LIBSVM version 3.2.2. (2018). [online] Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm [Accessed 10 jan. 2018].

Liu, Q., Sung, A. H. and Qiao, M. (2010). Detection of Double MP3 Compression. Cognit. Comput., Vol.2, No. 4; pp. 291–296.

Luo, D., Luo, W., Yang, R. and Huang, J. (2012). Compression History Identification for Digital Audio Signal. In: IEEE Int. Conf. Acoust., Speech Signal Process., Kyoto, Japan, pp. 1732-1736.

Luo, D., Luo, W., Yang, R. and Huang, J. (2014a). Identifying Compression History of Wave Audio and its Applications. ACM Trans. Multimedia Comput., Commun., Appl., Vol.10, No. 3; Article. 30, pp. 30-1–30-19.

Luo, D., Yang, R., and Huang, J. (2014b). Detecting Double Compressed AMR Audio Using Deep Learning. In: Proc. Int. Conf. Acoust., Speech, Signal Process. Appl., Florence, Italy, pp. 2669–2673.

Luo, D., Yang, R., Lin, B. and Huang, J. (2017). Detection of Double Compressed AMR Audio Using Stacked Autoencoder. IEEE Trans. Inf. Forensics Security, Vol.12, No. 2, pp. 432–444.

Pfeiffer, S. and Vincent, T. (2003). Survey of Compressed Domain Audio Features and Their Expressiveness. In: Proc. SPIE Conf. Electronic Imaging, San Francisco, USA, pp. 133-147.

Petracca, H., Servetti, A. and De Martin, J. C. (2005). Low-complexity Automatic Speaker Recognition in the Compressed GSM AMR Domain. In: Proc. IEEE Int. Conf. Multimedia and Expo (ICME), Netherlands, pp. 1-4.

Shen, Y., Lia, J. and Cai, L. (2012). Detecting Double Compressed AMR-format Audio Recordings. In: Proc. 10th Phonetics Conf. China (PCC), China, pp. 1–5.

Yang, R., Shi, Q. and Huang, J. (2009). Defeating Fake-quality MP3. In: ACM Workshop Multimedia Secur., Princeton, USA, pp.117–124.

Yang, R., Shi, Y. and Huang, J. (2010). Detecting Double Compression of Audio Signal. In: Proc. SPIE Conference on Media Forensics and Security II, vol. 7541, USA, pp. 1-10.