AUDIO WATERMARKING BASED ON EMPIRICAL MODE DECOMPOSITION AND BEAT DETECTION

Marius Teleşpan and Björn W. Schuller

Department of Computing, Imperial College London, U. K.
bjoern.schuller@imperial.ac.uk

ABSTRACT

In the recent years a large number of methods have been proposed in order to reliably embed information into audio files. Despite their increased robustness against attacks, they tend to have a lot of redundancy due to a large number of bits used for majority voting due to disability to correctly select regions that are unlikely to be distorted by attacks. To overcome this, we propose a robust method for audio watermarking where Empirical Mode Decomposition and beat detection are used for detecting the locations for embedding the watermark. In order to find the embedding locations, we use a simplified psychoacoustic model to split the input into audible frequency bands and two phase comb filtering on those bands to find the beat metrical structure. Then, at each embedding location, we take several frames and decompose them into Intrinsic Mode Functions. In an extensive test, we show promising results on a selection of songs spanning over three musical genres.

Index Terms— audio watermarking, beat tracking, empirical mode decomposition, intrinsic mode function, beat detection

1. INTRODUCTION

Digital Audio Watermarking has received great interest over the last years due to the lack of methods to prove ownership over audio content. A good audio watermarking technique should satisfy two core functionalities: the watermark should be imperceptible by the Human Auditory System (HAS) and robust against attacks. Attacks can be split into two categories: Deliberate or malicious attacks and unintended or non-malicious attacks [1]. Malicious attacks are created by 'pirates' in order to distort the audio watermark so that there will be no trace of the distribution of the audio. These attacks are done, e. g., by cropping the audio, by executing geometric alteration or simply by applying some other form of alteration. Non-malicious attacks are the ones that are executed by mistake when the holder of the audio does some form of processing over the audio such as compression, filtering, time stretching, etc. It should be observed that, these methods can be used also for intentional malicious attacks. As a consequence, a watermark should be prone against such simple 'attacks' that can occur (even) by mistake. Looking at the state of the art in methods for audio watermarking, one already finds algorithms providing substantial robustness and efficiency. For example, the algorithm presented in [2] manages to include the watermark in the audio signal while reaching a considerable data payload, high resiliency against attacks and at the same time the watermark remains inaudible. However, looking at Fig. 1 we can see that the last Intrinsic Mode Function (IMF) of a signal is quite altered by compression (in this case 128 kbs mp3 compression), which inspired us to use a lower IMF in this work. Another interesting idea is the use of the floor function in the embedding of the watermark. On a negative number, this will not have a similar behaviour as on a positive number leading to potential inconsistency. Considering the threshold used, this could easily produce values for the samples that are smaller than the minimum sample value allowed. There are further very mature methods for audio watermarking, such as [3], which uses spread-spectrum. Yet, its data payload is rather limited. Further promising approaches to audio watermarking include usage of the wavelets domain [4, 5], or phase coding and related techniques [1, 6, 7]. Overall, a popular classification of watermarks is as follows: algorithms operating in the time domain, transform domain, compressed domain, and combined domain audio watermarking.

In this paper we propose a novel robust method in the time domain using Empirical Mode Decomposition (EMD) which has been used, e. g., in [8]. EMD is a method for analysing non-stationary signals in a totally adaptive way [9]. The technique breaks a signal down into IMFs: These are nearly orthogonal functions with zero-mean. The decomposition has a finite number of modes depending entirely on the data. After decomposing a signal, it can be easily reconstructed from the IMFs, by adding them up as in (1), where $S(t)$ is the signal, IMF_i is the i-th IMF function, and r represents the residual.

$$S(t) = \sum_{i=1}^{n} IMF_i(t) + r(t). \tag{1}$$

An important characteristic of this decomposition is that the number of maxima and minima decreases with the order of each mode and that the high frequency modes (lower order IMFs) keep the same maxima and minima even after being attacked. We can observe this in Fig. 1 where in a) we see the difference between an original signal and the signal after compression and in b) we can see the minima and maxima of the first IMF of the signal before and after being compressed. In Fig. 1 we can observe that, the extreme points are entirely distorted for the second and third (last) IMFs.

2. PROPOSED WATERMARKING ALGORITHM

The idea of our proposed algorithm is to embed the watermark at locations that are less likely to be attacked by techniques such as 'mp3' or similar compression. This leads to content-based audio watermarking [10, 11, 12]. These parts of the songs usually coincide with the beat locations that contain low frequencies which mask other sounds. Considering beat detection in the context of watermarking has been considered in [13], albeit in other ways. Our algorithm will first identify according beat positions and then take

The research leading to these results has received funding from the European Union's Horizon 2020 Programme through the RIA #645378 (ARIA-VALUSPA).

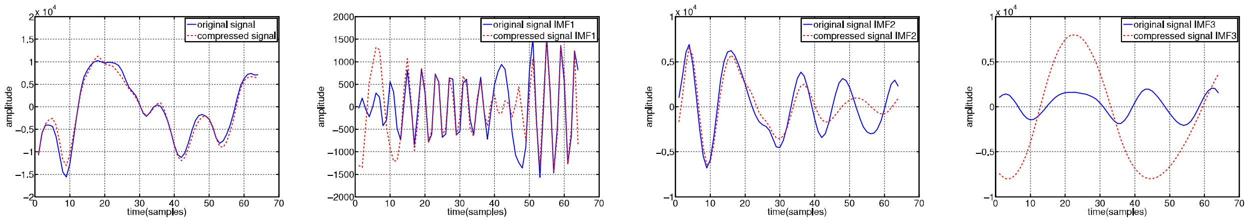

Fig. 1. Distortion created by 128 kbs mp3 compression over 64 samples frame from *The Saturdays – Gentleman* illustrated over the samples' amplitudes and over the first to third (last) IMF of the EMD (left to right).

20 frames of 64 samples at each beat location. The number of 64 samples was chosen, since they have 15–20 extreme points. We will use these extreme points of each frame for majority voting for one bit. We take 20 frames because we will have 1 280 (=20·64) samples at each beat location. Considering 44.1 kHz PCM WAV files, this number of samples represents just under 30 ms and it is less than half a length of a drum beat. So even if there is a slight error of an automatic detector of the beats, we can still make sure we will remain in the beat region with the watermark embedding. Afterwards, we take each frame, analyse it by EMD, and embed the watermark at the extreme position of the first IMF. For embedding the watermark, we use a Quantisation Index Modulation (QIM) technique [14] since it has good robustness against attacks and it can be used blindly to detect the watermark. In order to decode the watermark, we will use EMD as well and we will threshold the extreme points to see whether they represent a '1' or a '0'. Before arriving at the method presented here for including the watermark in the audio described in section 2.2, a number of variations have been tested. An initially promising one among these was to include the watermark at the extreme points of the first IMF by modifying the sample of the signal which was an extreme point for the first IMF. By lowering or increasing its value, depending on whether the sample was a minimum or a maximum, we were ensuring to keep the same extreme points. Even though the watermark had good resiliency agains attacks, this method was significantly altering the wave form of the signal. Therefore, the audio quality of the watermarked signal was not very good and the watermark was highly perceptible. Our final approach is based on this, but it modifies all the samples from the frame in order to keep a similar wave form for the signal and to keep the distortion of the audio to a minimum possible.

2.1. Finding the beat locations

The algorithm for finding the beat locations requires tempo independent information about the song's rhythmic structure. For this, it is necessary to find the song's beat tempo reliably. The approach used relies solemnly on finding multiple tempos in a song and comparing how well they resonate with the song. There are mainly three different approaches for tempo detection: using correlation methods, detecting note onsets and then finding the most common inter-onset interval (IOI), and a multiple resonator approach – usually with comb filters. Our tempo extraction falls into the third category, as we require a larger tempo search range, which implies computing more comb filters. Thus, the method relies merely on finding a base tempo called Tatum, and analysing how well integer multiples of this Tatum resonate with a large part of the song. The Tatum thereby corresponds to a tempo of at least the beat tempo or higher. In a later stage of the algorithm, after the beat tempo is known, it is possible to find the correct phase of the beats, by looking at the filter output and tracking the phase over the whole song to sort out errors. A more explicit description can be found in [15], [16], and [17].

Table 1. Pseudocode for embedding the watermark

```
method embed(int beatLocation) {
  // we use 20 frames
  for $i$ = 1 to 20 {
    // get i-th frame from beatLocation
    frame = getFrame(beatLocation, i);
    // decompose the signal via EMD
    emd = emdDecomposition(frame);
    // get first IMF
    imf1 = getIMF(end, 1);
    // 64 samples in each frame
    for $j = 1$ to 64 {
      if(extremumPoint(imf1, j)) {
        // calculate new extremum value by (1)
        newEx = NewEx(frame[$j$], Threshold);
        // calculate the difference by (2)
        difference = newEx - frame[$i$];
        // update the same sign neighbours (3)
        $l$ = $j$;
        while(sign(frame[$l$]) == sign(frame[$j$]) && $l > 0$) {
          frame[$l$]$+=$difference; $l--$;}
        $r = j + 1$;
        while(sign(frame[$l$]) == sign(frame($j$) && $r < 63$) {
          frame[$r$]$+=$difference; $r++$;}}}}}
```

2.2. Embedding the watermarks at a specific beat location

As mentioned, at each beat location we will embed a watermark in each of the 20 frames. So in each frame (64 samples) we will embed the same watermark at each extremum position in order to later use this for majority voting. This will enable us to include around 40 – 50 bits per second in a song, depending on the tempo of the song. Accordingly, the algorithm is sketched as:

1. At each beat location get 20 frames
2. Foreach frame compute EMD and get the first IMF
3. At each extremum of the first IMF calculate the following:

$$S^* = \begin{cases} rnd(S(t)/T) * T + sgn(3 * T/4) & \text{if } w_i \equiv 1 \\ rnd(S(t)/T) * T + sgn(T/4) & \text{if } w_i \equiv 0. \end{cases} \tag{2}$$

$$Difference = S^* - S(t), \tag{3}$$

where $S(t)$ is the original signal, and at sample t there is an extreme point in the first IMF, S^* is the value of the extreme

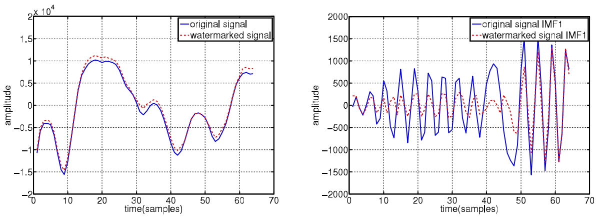

Fig. 2. Comparison of the amplitude of the original and 0-watermarked signal (left) and the amplitude of the original and 0-watermarked first IMF (right) using a 64 sample frame from *Disclosure feat. Eliza Doolittle – You & Me*.

point the watermarked signal should have, T is a threshold specified in order to allow for the balance between inaudibility and reliability, *rnd* represents the floor function if the point is a maximum point for the first IMF, or ceil if the point is a minimum point. *sgn* is the sign function, that is '+' if the point is a maximum and '-' if the point is a minimum, and w_i is the watermark for the particular frame.

4. for sample t and all neighbours of sample t that have the same sign apply the following
$$S(p)^* = S(p) + Difference. \qquad (4)$$

The last step is needed to ensure that the audio frame keeps a similar audio wave as the original signal. In Fig. 2 we can see the difference between the original and the watermark frame from *Disclosure feat. Eliza Doolittle – You & Me*. We see that, even at this scale there is a small difference between the original and the watermarked signal. However, considering that the values of the audio samples can range between -32 768 and 32 767, the actual difference between the original signal and the watermarked one is very small. In order to allow for a better understanding of the watermarking embedding technique, the pseudocode is shown in Listing 1.

2.3. Watermark Extraction

In order to extract the watermark, we first have to reliable identify the beat positions. After beat detection, we identify the frames, analyse them using EMD, and threshold them in order to identify the watermark. Accordingly, one executes as follows:

1. identify the beat location
2. foreach beat location select the 20 frames
3. foreach frame, find all extremum points and calculate D(t):
$$D(t) = \begin{cases} S(t) - \lceil S(t)/T \rceil * T & \text{if } minimum \\ S(t) - \lfloor S(t)/T \rfloor * T & \text{if } maximum, \end{cases} \qquad (5)$$
where $S(t)$ is the original signal, at sample t there is an extreme point in first IMF, and T is the same threshold used in the embedding for the particular frame.
4. foreach frame use $D(t)$ to calculate w_i, the watermark at an extremum, which will vote for the watermark bit of the frame
$$w_i = \begin{cases} 1 & \text{if } maximum \text{ and } D(t) > T/2 \\ 1 & \text{if } minimum \text{ and } D(t) < -T/2 \\ 0 & \text{if } maximum \text{ and } D(t) < T/2 \\ 0 & \text{if } minimum \text{ and } D(t) > -T/2, \end{cases} \qquad (6)$$
where w_i is the watermark of an extremum of the frame.

Table 2. ODG scale

Impairment description	**ODG**
Imperceptible	0
Perceptible, but not annoying	-1
Slightly annoying	-2
Annoying	-3
Very annoying	-4

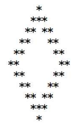

Fig. 3. 11 x 11 bit watermark as embedded in our experiments (00000100000000001110000...)

3. PERFORMANCE ANALYSIS

The performances of any audio watermarking technique should be analysed based on the imperceptibility of the audio watermark, on the capability of the watermark to resist against attacks, and on the data payload. In this sense, we will measure the performance against attacks by using Bit Error Rate (BER) [18] and Normalised cross-Correlation (NC) as defined in (7) and (8). A low BER indicates an accurate detection mechanism for the watermark, and a high NC (the maximum being 1) shows a high similarity between the embedded and extracted watermark.

$$BER(W, W') = \frac{\sum_{i=0}^{N} W(i) \bigoplus W'(i)}{N} \qquad (7)$$

$$NC(W, W') = \frac{\sum_{i=0}^{N} W(i) * W'(i)}{\sqrt{\sum_{i=0}^{N} W(i)^2} * \sqrt{\sum_{i=0}^{N} W'(i)^2}}. \qquad (8)$$

In (7) and (8), W and W' represent the watermark embedded in the audio and the watermark extracted from the audio, and N is the length of the watermark. For measuring the audio quality of the watermarked audio signal, a range of options such as PEAQ are used [19]. We decided for two popular and relevant measures: Signal to Noise Ratio (SNR) and Objective Difference Grade (ODG). SNR is defined as the ratio of signal power over the noise power and is measured in dB. The International Federation of the Photographic Industry (IFPI) recommends for a good audio watermarking technique an SNR greater than 20 dB. The ODG is a subjective quality assessment of the audio by human perception. They grade the imperceptibility of the watermark based on the scale defined in Table 2. Data payload measures how much information can be embedded in the audio in a second. This measure will vary for the proposed method depending on the tempo, and it will be compared to other watermarking techniques in the following section.

4. RESULTS

We will now consider the performance of our novel watermarking technique against a number of typical attacks. The watermark embedded in the audio files is a 121 bit sequence that is an 11 x 11 representation of Fig. 3. Since we do not only measure the resistance against attacks, but also audio quality and data payload, we had to

Table 3. Average BER(%) against common attacks for all, and the Pop, EDM, and Jazz songs

Genre	ALL	POP	EDM	JAZZ	ALL	POP	EDM	JAZZ
	SNR				ODG			
Original	27.5	28.0	28.2	26.3	-0.35	-0.18	-0.27	-0.60
	BER(%)				NC			
128 kbs MP3	3.8	2.5	3.5	5.4	0.92	0.95	0.93	0.89
96 kbs MP3	9.8	7.4	12.5	9.6	0.79	0.85	0.75	0.76
Resampling	7.5	5.0	11.1	6.4	0.85	0.90	0.79	0.86
Adding WGN	6.8	7.3	4.3	8.9	0.85	0.83	0.91	0.80

choose a threshold (1 000) that keeps a balance between all these factors. Having in mind that we use 44.1 kHz 16 bit PCM WAV audio files and that the samples can range between -32768 to 32767, altering the high frequencies on average only by +/-500 will not induce high audio corruption. In Table 3 one can find in the Audio Quality line the SNR ratio and the ODG value for Pop, Jazz, and Electronic Dance Music (EDM). Few efforts are so far made as in [20] to provide standardised song sets for comparison. Our results resemble averages over ten recent songs per considered genre[1]. From the results, the technique performs well on samples that have the tempo given by a drum beat or a snare drum. Having an ODG value between *Imperceptible* and *Perceptible but not annoying*, our technique passes the perceptibility test with good results. Note that, the technique embeds the watermark in the higher frequencies at beat locations. Therefore, it will enhance most of the high frequencies at these positions; and, if there is a predominant sound only with low frequencies, such as a drum beat intro for an EDM song, the watermark will be perceptible in this section, but usually not in an annoying way. However, as soon as the song 'drops in', and other instruments appear as well, the watermark becomes imperceptible. When measuring the ODG for each song, we took a measurement for each 10 seconds of the song and averaged that over the whole song. Otherwise, if voting straight for the whole song, the ODG will be close to 0 for all our tests. The high values of SNR are obtained as we only slightly modify the audio file. We use a threshold such that we maintain a good quality of the audio file. Therefore, we will not introduce significant noise. When measuring the SNR value, we took into account only such frames that were watermarked. If we were considering all frames from the audio file, the values of the SNR would have been significantly higher, but it would not show the actual difference between the original signal and altered signal.

Line *128 kbs* of Table 3 shows how the audio watermarking technique proposed performs against 128 kbs mp3 compression attacks. 128 kbs is the current (lower) mp3 de-facto standard used in the distribution of (music) audio. For all our tests we give the BER and the NC for our novel watermark.

Line *96 kbs* of Table 3 gives the BER and NC for a 96 kbs mp3 compression attack. Since this attack uses a higher compression rate than the 128 kbs one, one expects a (small) increase in the average BER. In the line *Resampling*, we measured the BER and NC against resampling attacks. In this attack, a 44.1 kHz signal is re-sampled at 20.5 kHz and then back to 44.1 kHz. Finally, we measured resilience against adding White Gaussian Noise (WGN) for an SNR of 30 dB as given in line *AWGN*.

To give a better idea on how well the watermarking technique performs across different genres against the above mentioned attacks, Table 3 also gives the average BER and NC for all attacks across genres.

From Table 3, we see fairly good results, but to set these into relation, techniques such in [2] and [21] can be considered. In these papers, lower BERs are reported for the same attacks, albeit considering only Pop or piano pieces. Data payload for our proposed watermarking techniques is 40 – 50 bits/s depending on the tempo of the song. This is similar to related techniques, such as in [4] which has a payload of 45.9, or the one described in [2] which has a payload of 46.9 – 50.3 bits/s. Mainly the technique presented in [21] achieves highly competitive results at 128 bits/s. Our payload can be obtained from ensuring that the locations used for embedding are not often altered by attacks.

5. CONCLUSION

We proposed a novel method for audio watermarking based on EMD. We used Neuronal Networks in order to find locations with high energy (beat locations) and then embedded 20 bits at each of these locations. From the experiments made, we observe good results against the de-facto internet standard of 128 kbs mp3 compression. Having an average BER of 2.5 % for Pop Music, 3.5 % for EDM, and 5.4 % for Jazz music, we find pronounced genre effects. This might be owed to the fact that, the beat detector is (data-)trained for a specific genre and thus might not identify reliably the beat positions of other genres. Therefore, it seems reasonable to train it individually for each particular type of music when used for embedding and extracting the beat locations. We measured good results against further common attacks. The average BER against resampling was 7.5 %, and the average BER against adding WGN was 7.7%.

Despite this Audio Watermarking technique maintaining fairly good performance against attacks, there are a number of improvements from which it could benefit. The usage of the beat detector is very effective in choosing the locations for embedding, but it might be a bottleneck for the extraction process. In order to reliably extract the same watermark, we need to reliably identify the same samples for extraction. If the audio is attacked using time scale modification, such as tempo variation, the beat detector might occasionally be inefficient in detecting the correct sample for extraction. Likewise, it seems interesting to measure whether better results can be achieved using a synchronisation code [2] at the embedding locations before the watermark. Accordingly, one can be certain to search at the correct positions for extraction. This would increase the complexity for the watermark extraction, and lower the data payload, but might increase the resilience against attacks such as high pass filtering and time stretching. Finally, to increase data payload, more general onset positions could be identified by onset detection as in [22].

[1] For comparison, the list of these is found at www.openaudio.eu.

6. REFERENCES

[1] R. F. Olanrewaju and O. Khalifa, "Digital audio watermarking; techniques and applications," in *Proc. International Conference on Computer and Communication Engineering (ICCCE 2012)*, Kuala Lumpur, Malaysia, July 2012, pp. 830–835.

[2] K. Khaldi and A. O. Boudraa, "Audio watermarking via emd," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 3, pp. 675–680, March 2013.

[3] D. Kirovski and H. Malvar, "Robust spread-spectrum audio watermarking," in *Proc. ICASSP*, Salt Lake City, Utah, 2001, pp. 1345–1348, IEEE.

[4] V. K. Bhat, I. Sengupta, and A. Das, "An adaptive audio watermarking based on the singular value decomposition in the wavelet domain," *Digital Signal Processing*, vol. 2010, no. 20, pp. 1547–1558, 2010.

[5] S. Wu, J. Huang, D. Huang, and Y. Q. Shi, "Efficiently self-synchronized audio watermarking for assured audio data transmission," *IEEE Transactions on Broadcasting*, vol. 51, no. 1, pp. 69–76, March 2005.

[6] G. B. Khatri and D. S. Chaudhari, "Digital audio watermarking applications and techniques," *International Journal of Electronics and Communication Engeneering & Technology (IJECET)*, vol. 4, pp. 109–115, March – April 2013.

[7] J. S. Pan, H. C. Huang, and L. C. Jain, *Intelligent Watermarking Techniques*, Innovative Intelligence. World Scientific, 2004.

[8] Z. Fu, P. Zhang, W. Huang, L. Wang, S. Emmanuel, and G. Chen, "Empirical mode decomposition based blind audio watermarking," *Multimedia Tools and Applications*, pp. 1–22, 2014.

[9] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of The Royal Society A Mathematical Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[10] D. Cui, L. Qirui, G. Yu, and X. Jianbin, "Content-based audio watermarking method to resist de-synchronization attacks," in *Information and Network Security, ICINS 2014-2014 International Conference on*. IET, 2014, pp. 28–32.

[11] C. Xu, N. C. Maddage, X. Shao, and Q. Tian, "Content-adaptive digital music watermarking based on music structure analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 3, no. 1, pp. 1, 2007.

[12] W. Li and X. Xue, "Audio watermarking based on music content analysis: robust against time scale modification," in *Digital Watermarking*, pp. 289–300. Springer, 2004.

[13] D. Kirovski and H. Attias, "Audio watermark robustness to desynchronization via beat detection," in *Information Hiding*. Springer, 2003, pp. 160–176.

[14] B. Chen and G. W. Wornell, "Quantization index modulation methods for digital watermarking and information embedding of multimedia," *Journal of VLSI Signal Processing Systems*, vol. 8, no. 1, pp. 46–59, Feb. 2006.

[15] F. Eyben, B. Schuller, S. Reiter, and G. Rigoll, "Wearable assistance for the ballroom-dance hobbyist – holistic rhythm analysis and dance-style classification," in *Proc. IEEE International Conference on Multimedia and Expo 2007*, Beijing, China, 2007, pp. 92–95, IEEE.

[16] G. Ferroni, E. Marchi, F. Eyben, S. Squartini, and B. Schuller, "Onset detection exploiting wavelet transform with bidirectional long short-term memory neural networks," in *Proc. Annual Meeting of the MIREX 2013 community as part of the 14th International Conference on Music Information Retrieval, ISMIR, Curitiba, Brazil,*, 11 2013, no pagination.

[17] B. Schuller, F. Eyben, and G. Rigoll, "Fast and robust meter and tempo recognition for the automatic discrimination of ballroom dance styles," in *Proc. ICASSP*, Honolulu, Hawaii, 2007, vol. I, pp. 217–220.

[18] A. G. Acevedo, "Audio watermarking: properties, techniques and evaluation," *Information Security and Ethics: Concepts, Methodologies, Tools, and Applications*, vol. 6, pp. 23–61, 2007.

[19] S. Nakaya and S. Wada, "Audio watermarking based on vector quantization index modulation using audio fingerprint," *Electronics and Communications in Japan*, vol. 98, no. 3, pp. 14–23, 2015.

[20] Y. Lin and W. H. Abdulla, "Audio watermarking techniques," in *Audio Watermark*, pp. 51–94. Springer, 2015.

[21] W. Li, X. Xue, and P. Lu, "Localized audio watermarking technique robust against time-scale modification," *IEEE Transactions on Multimedia*, vol. 8, no. 1, February 2006.

[22] E. Marchi, G. Ferroni, F. Eyben, S. Squartini, and B. Schuller, "Audio Onset Detection: A Wavelet Packet Based Approach with Recurrent Neural Networks," in *Proceedings 2014 International Joint Conference on Neural Networks (IJCNN) as part of the IEEE World Congress on Computational Intelligence (IEEE WCCI)*, Beijing, China, July 2014, IEEE, pp. 3585–3591.