Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/110309

Dynamics of Watermark Position in Audio Watermarked Files using Neural Networks

Adamu Abubakar [1,*], *Haruna Chiroma*[2], *Akram Zeki* [1], *Abdullah Khan*[3], *Mueen Uddin*[4] *and Tutut Herawan*[5,6]

[1] Department of Information Systems, International Islamic University Malaysia, 50728 Gombak, Kuala Lumpur, Malaysia.
[2] Artificial Intelligence Department, University of Malaya, 50603 Pantai Valley, Kuala Lumpur, Malaysia.
[3] Software and Multimedia Centre, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
[4] Department of Information System, Faculty of Engineering, Effat University Jeddah, kSA
[5] Department of Information Systems, University of Malaya, 50603 Pantai Valley, Kuala Lumpur, Malaysia.
[6] AMCS Research Center, Yogyakarta, Indonesia.

Received: 2 Jul. 2016, Revised: 9 Mar. 2017, Accepted: 15 Mar. 2017
Published online: 1 May 2017

Abstract: Previous researches on digital audio watermarking has shown that effective techniques ensure inaudibility, reliability, robustness and protection against signal degradation. Crucial to this is the appropriate position of the watermark in the files. There is a risk of perceivable distortion in the audio signal when the watermark is spread in the audio spectrum, which may result in the loss of the watermark. This paper addresses the lack of an optimal position for the watermark when spread spectrum watermarking techniques are used. In an attempt to solve this problem, we model various positions on the audio spectrum for embedding the watermark and use a neural network (feed forward neural network) to predict the best positions for the watermark in the host audio streams. We are able to determine optimal position. The result of the neural network experiment formulated within the spread spectrum watermarking technique enables us to determine the best position for embedding. After embedding, further experimental results on the strength of the watermarking technique utilizing the outcome of the neural network show a high level of robustness against a variety of signal degradations. The contribution of this work is to show that audio signals contain patterns which help determine the most appropriate points at which watermarks should be embedded.

Keywords: feed-forward neural network, spread spectrum, watermark-position

1 Introduction

Digital watermarking provides a promising solution to copyright protection. A watermark that contains information is embedded into the carrier file. The carrier medium tends to be in the form of text, video, audio or image format, while the watermark tends to be in either image or text format [1–7] The watermark can only be extracted by applying specified extracting techniques. From the viewpoint of human perception, watermarks can be categorized as visible, invisible or fragile [1, 2]. Visible watermarks are applicable to image and video files where the embedded watermark remains visible, although it is sometimes transparent. Invisible watermarks remain - as their name implies - hidden in the content and can only be detected by an authorized agency. Fragile watermarks are destroyed in the course of any attempt to manipulate data. Audio watermarking perceptibility falls only within the scope of the human auditory system (HAS) [8–10]. The scope of this paper is limited to the process of digital audio watermarking using spread spectrum techniques because it is the most successful and secure method [11]. Spreading the watermark throughout the spectrum of an audio file ensures greater security against attack. When using the spread spectrum technique, the watermark is inserted into the highest magnitude coefficients of the transformed audio file [12]. Thus, any low bin within those positions would be difficult to detect, intercept or remove. However, when a watermark is spread within the high-frequency spectrum of the carrier file, it can introduce perceivable distortion into the host signal, and can be detected by gradual filtering inspections of the entire carrier file [13–15]. It may also be exposed to signal degradation techniques caused by subsequent

* Corresponding author e-mail: 100adamu@gmail.com

modifications or alterations. There is a high probability that the host or carrier file will be unable to withstand common reverse random processes, and a bit sequence might be re-established which could disrupt or override the watermark spectrum [16–19], resulting in damage or loss of the watermark file [20–23]. This would make it difficult to determine the strength of spread spectrum watermarking. To address this drawback, a modified trained sequence of each segment in the frequency domain for the watermark inserted into the audio stream file is collected. An artificial neural network (ANN) is used to model them in order to determine those positions which provide a good result. It is not the embedding or extracting process of a watermark that is modelled by neural networks, but rather the positions of the sequence of watermark segments in the frequency domain of the carrier file in which the watermark is embedded by the spread spectrum technique. The neural network model is capable of effectively predicting the most suitable position within the spectrum of the carrier file. The commonly used spread spectrum technique of determining watermark positions is a trial and error method which is time consuming and has no justification [24]. Prediction using a neural network is more accurate than conventional statistical, mathematical and econometrics models [25]. The rest of this paper is organized as follow: Section 2 describes techniques used in audio watermarking. Section 3 describes the proposed method. Section 4 describes results following by discussion. Finally, the conclusion of this work is described in Section 5.

2 Audio Watermarking Techniques

Digital watermarking falls into the classification of ‘watermarking by carrier medium ’, which allows imperceptible embedding (in other words, hiding) of digital information into audio content so that this information cannot be removed without damaging the original audio quality [1]. The embedded inaudible information can be retrieved and used to verify the authenticity of the audio content, and identify its owners and recipients; it also serves as an event trigger [1–3]. Audio data exists in signal form and analysis is in both time-domain and frequency-transform domains, therefore the watermarking technique will follow these.
Generally, to embed watermark data into an audio stream requires the transformation of the watermark into a binary format and, to some extent, to an encoded state. Different methods of transformation involve discrete Fourier transform (DFT), discrete cosine transform (DCT) and discrete wavelet transform (DWT) [26–28], If $g(x)$ represents binary watermark data and $g(x) \in -1, +1$ this can then be directly embedded into the audio signal, say $f(x)$ in its time/spatial or transform domain. However, in the time/spatial domain, the audio signal is manipulated directly, whereas in the transform domain, it needs to be changed into a transform representation. If $h(x)$ represents the degree of strength of the watermark as a function of time, spatial or transform domain coefficient, then the embedding procedure will yield a watermarked data $f'(x)$ where

$$f'(x) = f(x) + h(x).g(x) \tag{1}$$

In this case, the watermark $g(x)$ with the technique control function $h(x)$ is embedded directly into the audio signal, and as a result it will be extracted when it is subtracted from the watermarked file. On the other hand, the watermark $g(x)$ and the technique control function $h(x)$ undergo the following procedure:

$$\frac{f(x).[g.h(x)+1]}{f(x)[h(x)+1]} \tag{2}$$

To yield a watermarked $f'(x)$, extraction of the watermark from the watermarked file will follow this procedure:

$$f(x).[g.h(x)+1].f(x)[h(x)+1] \tag{3}$$

In another approach, the watermark can be embedded in a non-linear state, where the procedure involves quantization of the audio host signal as follows, by perturbating the quantized technique control function $h(x)$ through

$$\left\{\left[\frac{f(x)}{f(x)}\right] + \frac{1}{4}.g(x)\right\}.h(x) \tag{4}$$

Thus, the reverse of the embedding leads to extraction; by quantizing the watermarked file with a similar technique control function, the watermark is removed from the watermarked signal.

2.1 Audio Watermarking Based on Least Significant Bit

The least significant bit (LSB) audio-base watermarking technique is the simplest way to embed a watermark into an audio signal [29]. This involves direct substitution of LSB at each audio data sampling point by a watermark coded binary string; the substitution operation hides the bits to substitute the original. This obviously introduces audible noise at a specific point within the content of the audio signal. Any attempt at resampling the content of the audio signal or compression will lead to the loss or destruction of the watermark [30].

2.2 Embedding an Audio Signal Echo

An echo-hiding algorithm embeds a watermark into a host audio signal by inserting a small watermark as an

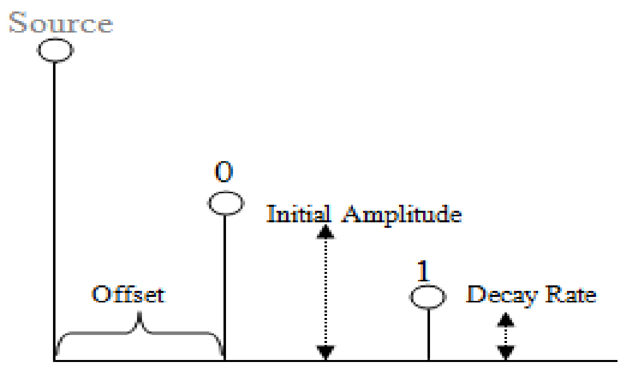

Fig. 1 Echo adjustable parameters.

echo that is perceptible by humans [31]. Since the human ear is used to hearing this slight resonance of sound, adding small amounts of echo within its three parameters (initial amplitude, decay rate and offset) as shown in Figure 1, will not significantly impair the quality of the sound. Since echoes are generated naturally in most sound recording environments, when the original source signal and the echo decrease in the frequency domain, both signals merge to an extent where the offset is so negligible the human ear will not differentiate between them [32]. Unfortunately, the efficiency of this technique in audio watermarking depends on the fact that an echo of less than 1/1000 second is not perceptible by humans because of the weak property of HAS [31]. This means that an increase in the embedded watermark will affect delivery of pure sound quality.

2.3 Phase Coding Audio Watermarking

This technique relies on the phase difference of the audio segment, that is, the short-term phase of the signal over a small time interval [33]. The phase segments are preserved and the watermark will be embedded. When the phase signal $a(t) = cos(2\pi xt)$ is with another phase $a(t) = cos(2\pi xt)$ in an audio stream over a short time, then exits and later gets out of phase by a constant ϕ_0 in the form $a(t) = cos(2\pi xt) + cos(2\pi xt + \phi_0)$, this difference cannot be detected by HAS [33] until the constants are completely out of phase. That is, the degree of phase difference becomes high, as in this form

$$a(t) = cos(2\pi xt) + cos[(2\pi xt + \phi_0 + \phi_t] \tag{5}$$

Thus, the phase differences between segments of audio streams are utilized to embed the watermark. Unfortunately, if a single segment within a phase is removed, compressed or reassembled, it will lead to damage or loss of part of the watermark, which will eventually affect the entire watermark.

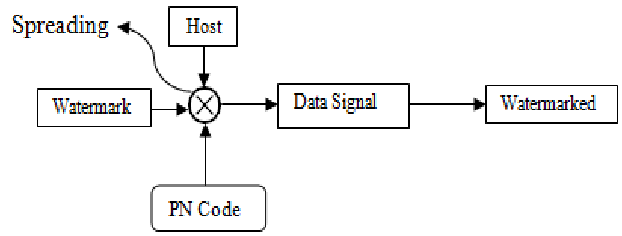

Fig. 2 DSSS encoding method.

2.4 Embedding on Spread Spectrum of Audio Signal

Spread spectrum (SS) refers to the communication signals generated at a specific bandwidth and intentionally spread in a communication system. In the process, a narrow-band signal is transmitted over a large bandwidth signal [31], making it undetectable as it is overlapped by the larger signal. SS is used in watermarking because the watermark to be embedded in the carrier file consists of a low-band signal that is transmitted through the frequency domain of the carrier file in the form of a large-band signal. The watermark is thus spread over multiple frequency bands so that the energy of one bin is very low and undetectable [33]. It becomes more difficult for an unauthorized party to detect and remove the watermark from the host signal.

There are two types of spread spectrum: the frequency-hopping spread spectrum (FHSS) and the direct-sequence spread spectrum (DSSS). In DSSS (see Figure 2), a watermark signal is embedded directly by introducing a pseudo-random noise sequence (PN), which implies that a key is needed to encode and decode the bits [31]. In FHSS, it is customary for the original audio file to be transformed into the frequency domain using the DCT, as follows:

$$W_x(q) = DCT[w_x(n)] \tag{6}$$

where x is the length of the audio signals; hence the PN sequence is used to select randomly from a set of predefined frequencies which are used to control the carrier frequency of the data signal [33, 34]. The carrier frequency ‘hops ’across the spectrum at set intervals of time, and in a pattern determined by the PN sequence [35].

3 Methodology

The approach of this study is closely related to [36] and is presented in Figure 3. The audio carrier file and the image watermark file constituted the payloads. At first, both files were transformed into their frequency domains. Then, they were presented as the input in the artificial neural network

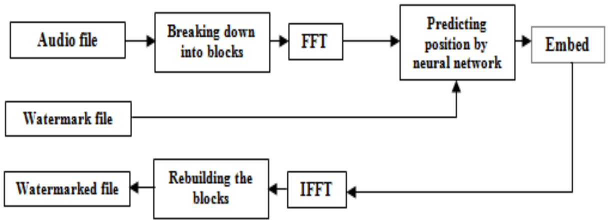

Fig. 3 The proposed watermarking technique.

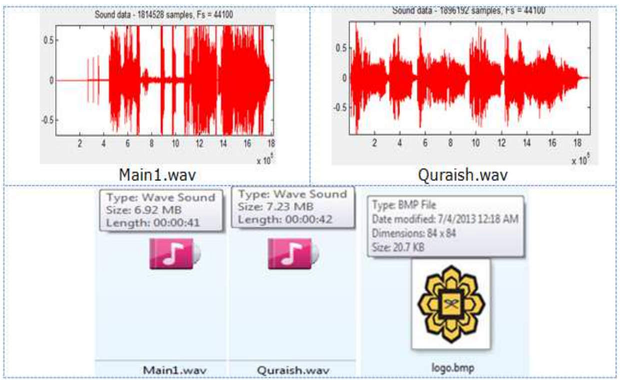

Fig. 4 Two audio carrier files and the watermark image file.

model to determine the best position for the watermark to be embedded into the carrier file. The watermark file was embedded and the sequences of the spectrum were rebuilt and a new watermarked audio file was formed.

3.1 The Host and Watermark Files

In this study, two audio files were chosen as the carrier files (Main1.wav and Quraish.wav) and a single image file was chosen as the watermark file (logo.bmp) as shown in Figure 4. The audio signal of Main1.wav and Quraish.wav had the following parameters: sampling rate 44, 100Hz, resolution 16 bit, hit rate 705 kbit/sec, and mono. The watermark file was a $MM = 8484$ binary image, shown in Figure 4, which was used as the watermark for both audio signals. To achieve a good trade-off among the conflicting requirements of imperceptibility, robustness and payload, the audio block size bb was 1515, the fixed quantization step size $\Delta = 0.5$ and the fixed dither value $d = 1$.

3.2 Predicting Embedding Positions by Artificial Neural Network

Predicting the best positions to embed watermarks in an audio signal aims to maximize the robustness of the SS technique against attacks and make it easier to detect embedded bits in case of loss of synchronization. Embedding and extraction of the watermark (logo.bmp) on host audio files (Main1.wav and Quraish.wav) (see Figure 4) using the SS technique were carried out first. This was done several times and at each iteration the corresponding watermark bits and host file bits were recorded. The degrees of robustness against attacks were also recorded. These are used as our dataset for the neural network experiment. The audio file signal to be watermarked is in $z \in R^N$ which is modelled as a random vector where z_i is the total length of the audio signal containing a set of values that are independently and equally distributed with standard deviation ϕ_z , such that

$$\frac{1}{2}(z_[i] \sim N(0,\phi)) \tag{7}$$

Where z is segmented into non-overlapping segments i with segment n. The i^{th} segment z_i contains n number of samples represented as follows:

$$z_i = \left\{z_{i,1}, z_{i,2}, \ldots, z_{i,n}\right\} \tag{8}$$

where $z_{i,j}$ represents the j^{th} number of i^{th}. Thus, the magnitude of the audio spectrum traverses each segment or sub-band of a frame of the audio signal. The segments of the host audio signal are subsequently divided into 131 frames, each containing 1024 samples. Each frame is further segmented into 32 sub-bands, each sub-band containing 32 samples using the following equation:

$$S(i) = \frac{1}{n}\sum_{n}^{j=1} |z_{i,j}|, i = 1, 2\ldots|, Q \tag{9}$$

Where Q is the total number of segments. For each segment, DCT is performed to obtain both the mean value and the maximal peaks. This marks the end of the pre-processing step. A pilot test was carried out for the binary of those segments where the position for embedding will be predicted by using the well-trained feed forward neural network, for which it returns the desired position. This is for each audio sub-segment within the finite length of the input in

$$Z = \left\{z_i | S(i) \geq X, i = 1, 2, \ldots, Q\right\} \tag{10}$$

where Z represents the collections of sound data that are spread within the line that indicates the fitness of the entry. The maximum entry is obtained by X when each frame consists of 32 sub-bands, each sub-band containing 32 samples. If s' represents a single segment, and is divided into s'_1 and s'_2 with ρ_1 and ρ_2 sample, then the synchronization code and watermark are embedded into s'_1 and s'_2 respectively, while masking the threshold for the sub-band to establish a synchronization code sequence and initialise the sample starting position. The gathered watermark bit, host audio bit and watermarked bits are

finally prepared for ANN experiment. Conventionally, this is a mathematical representation of biological neural systems which are able to process vast streams of data quickly [19]. Using neural networks to determine a suitable position for the watermark in the audio host file involves creating networks made up of simple 'artificial neurons 'which process information. They function much like the human brain, by learning from past experience and detecting novel ways to solve a problem [37]. Such a network is connected with input, hidden and output layers, and neurons distributed in the layers. The input and output neurons are determined by independent and dependent variables of the problem to be solved.
The data in our study were normalized to improve prediction accuracy [38] in a range of -1 to 1. Using equation 11

$$x' = \frac{0.9-0.1}{x_{max}-x_{min}}x + \left(0.9 - \frac{0.9-0.1}{x_{max}-x_{min}}x_{m}ax\right) \quad (11)$$

where x signifies the initial training data, x' the normalized value, and x_{max} and x_{min} the maximum and minimum values of the same input node in all the training samples. The data was divided into 70%, 15% and 15% for training, validation and testing, respectively, according to convention in Beale et al. [39]. The watermark signal sequences generated from equation 3 were mapped to the overall segments of the audio files which were fed onto the multilayer perceptron neural network during the training phase.
The NN used in this research comprised input, hidden and output layers as typically represented in the structure shown in Figure 5. NN can consist of several hidden layers. According to established theory, as described in [40], a single hidden layer is sufficient to approximate any function with arbitrary accuracy. Determining the number of hidden neurons, however, has no ideal framework [19]. In this study, the commonly used trial and error technique is applied [41] to determine the optimal number of hidden neurons, which was found to be ten. Input and output layer neurons corresponded to independent and prediction horizons, respectively [42]. The number of neurons in the input layer corresponded to the embedded image, and the output of 1 corresponded to a one-step prediction of embedding positions. In Figure 5 (the figure of NN with inputs from $x1,...,x10,h1,h2...hn$ where $n=10$, according to the equation and output y), the neurons distributed in the NN structure layers operated by summing its weighted input vectors and transferring the computed results through a non-linear transfer function [43]. Training NN is a non-linear, unconstrained minimization problem that minimizes the weights of a NN by iteratively modifying them to minimize the MSE between predicted and target values for all output neurons over all input vectors [44]. Several learning algorithms have already been discussed in previous studies, such as conjugate gradient descent algorithms, quick propagation, resilient back propagation and scaled conjugate gradients.

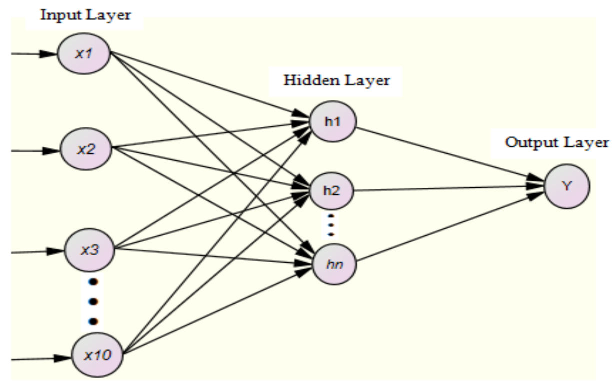

Fig. 5 Model of a multilayer perceptron neural network with one hidden layer.

This experiment used *Levenberg − Marquardt*, as empirical evidence by [42] suggests that it performs better than other fast learning algorithms in terms of prediction.
In the present study, a sigmoid transfer function is used to transfer the embedded image from input neurons to hidden neurons (see Table 1). At the output layer, a linear activation transfer function is used as recommended by [42], since the sigmoid is at the hidden layer and using a non-linear transfer function may restrict the output values to a limited range [45]. In addition, in the majority of previous studies, a sigmoid activation function was employed [40]. At each hidden layer neuron, output is computed using equation (14); output at the output neuron is also computed using equation (14).

$$f(x) = (1+exp(-x))^{-}1 \quad (12)$$

$$x_j = (1+exp(-x))^{-}1\left(\sum_{i=1}^{n=10} x_i w_{ji} + w_{j0}\right) \quad (13)$$

$$f(x) = X \quad (14)$$

$$y = \sum_{j=1}^{m} X_j w_{kj} + w_{k0} \quad (15)$$

According to the equations, x_i represents the embedded image which is the input variable w_{ji}, and w_{jk} represents weights connecting the input and hidden neurons, thus forming the connection between the hidden and the output neurons. The threshold for bias of ith and kth neurons is w_{j0} and w_{k0}, respectively. The numbers of neurons for the layers are i, j and k. In equation (15), y is compared with the target output for training datasets, and the computed output differences are summed together to generate error function (Er). Er is computed using equation (16):

$$Er = 0.5(\text{Predicted embedding position-embeddd position})^2 \quad (16)$$

Table 1 Summary of the nn model architectural configurations

Parameters	Configurations
Hidden neurons	10
Hidden layers	1
Input neurons	10
Input layer	1
Output layer	1
Learning algorithms	LM
Transfer function at hidden layer	Sigmoid
Transfer function at output layer	Linear

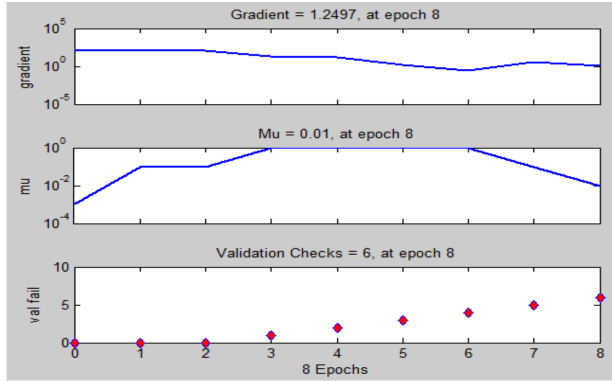

Fig. 6 Training state.

where the most widely used MSE serves as an error measure [19]. The lower the value of MSE, the better is the result. A value of 0 MSE means perfect prediction.
In the simulated experiment, a multilayer perceptron is selected to contain input, hidden and output nodes, with 5, 3 and 1 nodes respectively. The segmented audio signals are categorized into two parts, one part being used for multilayer perceptron training and the other part for testing to improve prediction accuracy and prevent the hidden layer neurons from saturating [46].
At the hidden and output layers, the sigmoid function $f(x)$ and linear function were used as recommended in [42]. The network model was run 15 times (see Table 2) to confirm the result. The training, as stated earlier,

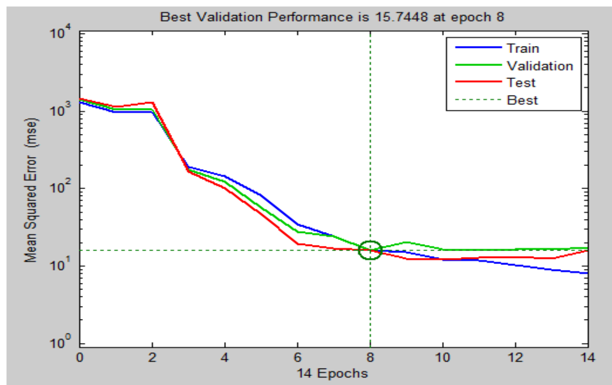

Fig. 7 Validation check.

Table 2 Descriptive statistic of the neural network experiment

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Iteration	15	8	19	11.6	3.41844	11.686
Training	15	54.56	289.17	132.706	75.91107	5762.49
Validation	15	60.4	434.14	155.6765	123.64368	15287.759
Testing	15	86.47	443.78	195.1967	120.7473	14579.911
Time Complexity	15	0.01	0.06	0.0247	0.01642	0

involves loading the inputs to the network which the network then runs, and is then adjusted according to its error. The 15 experimental runs which were carried out resulted in a minimum output of 54.56 and a standard deviation of 75.9, less than the mean which indicated an appropriate sequence of training. The validation measured network generalization and halted training when the generalization stopped improving. The mean value arrived at after the 15 experimental runs was greater than the standard deviation of 123.64368. The testing had no effect on training and thus provided an independent measure of network performance during and after training. The total testing sum of the network model resulted in 2927.95, which constituted the highest variable. Optimal architecture of the NN was found to have 10 input neurons, 20 hidden neurons and one single output neuron, since the model was designed to predict the most suitable position for embedding the watermark in an audio file. The extracted normalized data was partitioned according to the ratio $80\%, 10\%, 10\%$ representing training, validation and testing datasets, respectively, although there was no fixed ideal framework for the data partitioning ratio. The chosen partition ratio was adopted from [19]. MSE was the chosen performance indicator, and the neural network was trained with *Levenberg – Marquardt* learning algorithms. The training state is depicted in Figure 6; optimal validation was obtained at epoch 8, as shown in Figure 7, and all-datasets fit is seen in Figure 8. The predicted watermark positions comply with the data payload, which is the number of bits that can be embedded into the audio signal within a unit of time and is measured in bps (bits per second). It is determined by the length of the host audio signal and the watermark size given by $P = \frac{M}{L}$ where P is the data payload, M is the size of the watermark and L is the length of the audio signal.

3.3 Embedding the Watermark

Following the NN experiment for a specific dataset for this study, the binary of the watermark for which good results

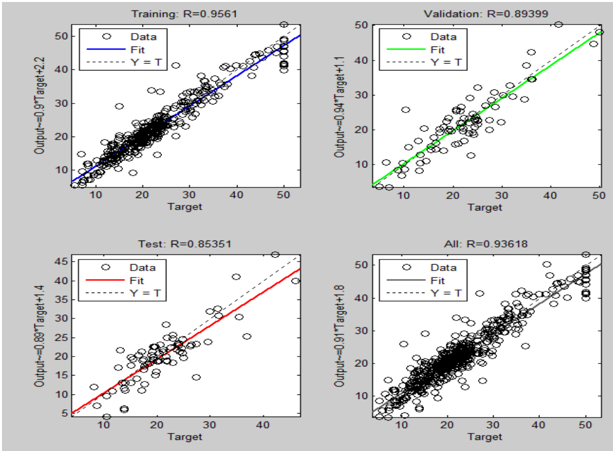

Fig. 8 Training, validation, testing and all-datasets fit.

were obtained is pre-processed. The watermark image file is transformed from W to WI , where

$$W_1 = \left\{ s_1(i,j), 0 \le i < M, 0 \le j < N \right\} \tag{17}$$

Its one-dimensional sequence of ones and zeros is as follows:

$$W_2 = \Big\{ s_2(q) = w(i,j), 0 \le i < M, 0 \le j < N,$$
$$q = i \times N_j, w_2(q) \in 0,1 \Big\}$$

Each bit in the segment s'_n of the watermark data is then mapped into the preceding audio segment $s'_n(q)(q = 0,1,\ldots,M \times N - 1)$ selected by $\frac{L_2}{(M \times N)}$ samples in the sequence, according to the following equations: $W_3 = \left\{ s_3(q) = w - 2 \times s_2(q), q = 0,1,\ldots,M \times N - 1, w_3(q) \in -1,1 \right\}$ In each of the preceding segments s'_n , they are transformed into DCT, giving rise to the DCT coefficients of $s'_2(q)^P, h'_2(q)^P, h'_2(q)^{P-1}, \ldots, h'_2(q)^1$. This means that $s'_2(q)^P$ is the common sample signal and $h'_2(q), h'_2(q)^{P-1}, \ldots, h'_2(q)^1$ indicates the complete depiction of the signals in the following state:

$$h'_2(q)^P = \Big\{ s'_2(q)(u)^P, q = 0,1,\ldots,M \times N - 1,$$
$$0 \le u < \frac{L}{M \times N \times 2^P} \Big\} \tag{18}$$

While a quantizing value $s'_2(q)$ maintains the state, the watermarked sequence of data $s_1(q)$ is then embedded into the audio segments, in the following phase:

$$s'_n(q)(u)^P = \begin{cases} s'_n(q)(u)^P - s'_n(q)(u)^P + k \;\; if \;\; w_n(q) = 1, \\ \\ s'_n(q)(u)^P - s'_n(q)(u)^P - k \;\; if \;\; w_n(q) = -1, \end{cases}$$

$$\left(q = 0,1,\ldots,M \times N - 1, 0 \le u < \frac{L}{(M \times N \times n^P)} \right)$$

Thus, the embedding follows the equation below:

$$s'_2(q)(u)^P = \begin{cases} s'_2(q)(u)^P - s'_2(q)(u)^P + k \;\; if \;\; w_3(k) = 1, \\ \\ s'_2(q)(u)^P - s'_2(q)(u)^P - k \;\; if \;\; w_3(k) = -1, \end{cases}$$

$$\left(q = 0,1,\ldots,M \times N - 1, 0 \le u < \frac{L_2}{(M \times N \times 2^P)} \right)$$

Following some series of embedding and applying attacks, it was realized that the scheme needed improvement in the embedding, through the trained positions from the audio and watermark inputs for the NN model. There follows a sequence of treating each segment of the audio separately. The watermark is then represented by the sequence w which is a vector pseudo-randomly generated in $w \in \pm 1^N$. Each elemen w_i is usually called a '$chip'$. Watermark chips are generated in such a way that they are independent of the original recording z. The marked signal a is created by $a = z + \beta_w$ where β signifies the watermark amplitude.

$$Z_2^0(x)(y)^B = \begin{cases} z_i X - \Delta(z_2^0(x)(y)x_i, S)^B \times v((z_i) + z_i|z_2^0|) + \\ X, \;\; if \;\; wi(k) = 1 \\ \\ z_i X - \Delta(z_2^0(x)(y)x_i, S)^B \times v((z_i) + z_i|z_2^0|) + \\ X, \;\; if \;\; wi(k) = -1 \end{cases}$$

$$k = (0,1,\ldots M \times N - 1; 0 \le t < q|M \times N \times 2^B)$$

$$Z_2^0(x)(y)^b = \begin{cases} 1 \;\; if \;\; wi(k) = 1 z_i X - \Delta(z_2^0(x)(y)x_i, S)^b \times v((z_i) \\ +z_i|z_2^0|) \ge S/2 \;\; for \;\; i = 1,2,\ldots X \\ \\ 0 \;\; if \;\; wi(k) = 1 - z_i X - \Delta(z_2^0(x)(y)x_i, S)^B \times v((z_i) \\ +z_i|z_2^0|) \ge S/2 \;\; for \;\; i = 1,2,\ldots X \end{cases}$$

$$k = (0,1,\ldots M \times N - 1; 0 \le t < q|M \times N \times 2^b; b = B, B-1,\ldots,1)$$

where $Z_2^0(x)(y)^B$ constitutes the original segments and $Z_2^0(x)(y)^b$ the segments into which the watermark is to be embedded. If the watermark bit is x, then shift x' as shown in equation 14, otherwise if $x = -1$, place the bit. The first approach is applied in Figure 9 (Quraish.wav) and the watermarked version is shown in Figure 11 with PNSR 45.06dB. The second embedding is applied to Main1.wav, as shown in Figure 10, and the corresponding watermarked version is presented in Figure 12.

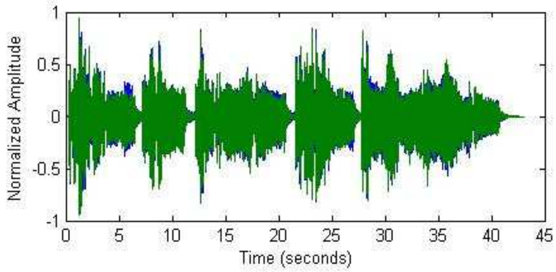

Fig. 9 Original digital audio (Quraish.wav).

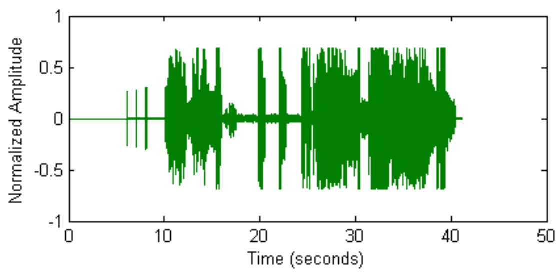

Fig. 10 Original digital audio1 (Main1.wav).

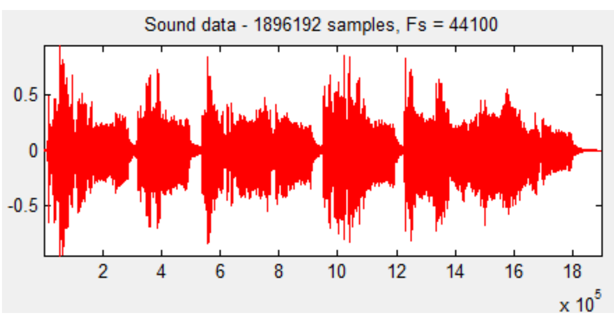

Fig. 11 Watermarked Main1.wav (PNSR=45.06dB)

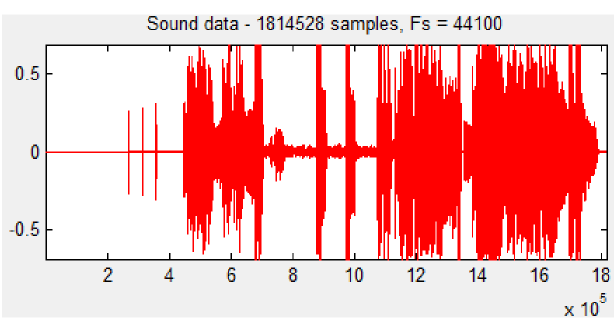

Fig. 12 Watermarked Quraish.wav (PNSR=45.06dB).

3.4 Detecting and Extraction of the Watermark

Synchronization code detection refers to the process of checking the synchronization code, it is commonly called resynchronization as it is the reverse of synchronizing audio data segments to detect any anomaly. The watermarked audio signal W' is segmented into S_i where $i = 1,2,\ldots,M\times M$ of size $w\times w$ and MM is the number of bits in the watermark image. Thus, we first establish the start position by representing $x\cdot y$ as the normalized inner product of vectors x and y,i.e. $x\cdot y \equiv N^{-1}\sigma_i x_i y_i$ with $x^2 \equiv x\cdot x$. In the case where P is introduced as $p^2 = 1$, inverse DCT is performed on each segment $s^*(q)$, and we proceed as follows: $s^*(q)^P, h^*(q)^{P-1},\ldots,h^*(q)^1$ Thus, a watermark (let us call it P) is detected by correlating a given signal vector q with P:

$$O(q,p) = q\cdot p = E[q\cdot p] + N'\left(0,\frac{a_z}{\sqrt{N}}\right). \qquad (19)$$

and subsequently extracted from each of the preceding segments in the following:

$$w'_n(q) = \begin{cases} 1 \;\; if \;\; s^*(q)^P > 0, \\ \\ -1 \;\; if \;\; s^*(q)^P \leq 0 \end{cases}$$

$$(q = 0,1,\ldots,M\times N-1)$$

In the second case, the synchronization code can be extracted by

$$Z_2^0(x)(y)^b = \begin{cases} 1 \;\; if \;\; wi(k) == 1z_iX - \Delta(z_2^0(x)(y)x_i,S)^b \\ \times v((z_i)+z_i|z_2^0|) \geq S/2 \;\; for \;\; i=1,2,\ldots X \\ \\ 0 \;\; if \;\; wi(k) == 1 - z_iX - \Delta(z_2^0(x)(y)x_i,S)^B \\ \times v((z_i)+z_i|z_2^0|) \geq S/2 \;\; for \;\; i=1,2,\ldots X \end{cases}$$

4 Evaluation of the Technique

To ensure that the technique used does not alter the host file, peak signal to noise ratio (PSNR) is used. PSNR has been used to evaluate the quality of watermarked audio after embedding the watermark and it is represented by $PNSR(X,X') = 10log10\frac{X_{peak}^2}{\lambda_j^2}$ where λ_j^2 is defined as

$$\lambda_j^2 = \left(\frac{1}{Z}\right)\sum_{i=1}^{Z}(y(i) - y'(i))^2$$

Where Z is the length of the host audio, $y(i)$ is the magnitude of the audio X at time i. Similarly, $y'(i)$ denotes the magnitude of watermarked audio X' at time i. X_{peak}^2 denotes the squared peak value of the host audio. The higher PSNR means that the watermarked audio is more like the original. There is no standard value for PSNR; however, the larger the PSNR, the better the audio quality will be. Some researchers considered the acceptable quality of watermarked audio when the PSNR was greater than 30 [47, 48], while some asked for 34 dB [49, 50] and others asked for an exaggerated value as high as 38 dB for an acceptable image quality [51]. Thus, all the PSNR values are above 40 dB.

To ensure robustness against attacks, normalized correlation (NC) is used to evaluate the correlation between the extracted and the original watermark and is given by

$$NC(W,W') = \frac{\Sigma_{i=1}^{M}\Sigma_{j=1}^{M} W(i,j)W'(i,j)}{\sqrt{\Sigma_{i=1}^{M}\Sigma_{j=1}^{M} W^2(i,j)}\sqrt{\Sigma_{i=1}^{M}\Sigma_{j=1}^{M} W'^2(i,j)}} \tag{20}$$

In the above, W and W' are the original and the extracted watermarks, respectively, while i, j are indices in the binary watermark image. If NC (W,W') is close to 1, then the correlation between W and W' is very high, while close to zero means a very low correlation. Another measure for calculating the robustness of the watermarking algorithm is the bit error rate (BER) represented by:

$$BER = \frac{\Sigma_{i=1}^{M}\Sigma_{j=1}^{M} W(i,j)W'(i,j)}{\sqrt{\Sigma_{i=1}^{M}\Sigma_{j=1}^{M} W^2(i,j)}\sqrt{M \times M}} \tag{21}$$

Utilizing the above equations, some common audio signal processing attacks are performed to assess the robustness of the technique. The following are the various attacks used:

1) MP3 compression 64 kbps and 32 kbps (MP3C): MPEG-1 layer-3 compression is applied. The watermarked audio signal is compressed at a bit rate of 64 kbps and 32 kbps, respectively, and then decompressed back to WAVE format.

2) Echo addition (Echo): An echo signal with a delay of 98 ms and a decay of 41% was added to the watermarked audio signal.

3) Re-quantization (Re-quan): This process involves re-quantization of a 16-bit watermarked audio signal to 8-bit and back to 16-bit.

4) Cropping (Re-ass): The removal of segments of 500 samples (5 100) from the watermarked audio signal at five positions, subsequently replaced by segments of the watermarked audio signal attacked with low-pass filtering and additive white Gaussian noise.

5) Jittering (Pitch): This involves an evenly performed form of random cropping. One sample out of every 5,000 (10,000) samples in our jittering experiment is removed.

6) Additive noise (AddN): White noise with 10% of the power of the audio signal is added, until the resulting signal has an SNR of 20 dB.

7) Low-pass filtering (Low): The low-pass filter with cut-off frequency of 11,025 Hz is applied to the watermarked audio signal.

8) Pitch shifting: This is applied by shifting one degree higher and one degree lower.

The attacks performed on the watermarked files and the extracted watermarks, along with the measures of NC and BER values, are summarized in Table 3. The entire NC above 0.7 and all the BER values are below 2%. The extracted watermark visual presentation is similar to the original watermark, showing the strength of the technique used. The values NC and BER obtained are from the following attacks: MP3 compression 64 kbps, echo addition, re-quantization, cropping, jittering, MP3 64 kbps, additive noise and low-pass filtering; they were compared with some of the similar published measures acquired through different techniques (see Tables 4 and 5). This research's choice of getting best position for embedding achieves high embedding capacity and low BER against attacks when compared with previous research.

Table 3 Extracted watermark with PSNR, NC AND BER

	MP3C	Echo	Re-quan	Re-ass	Pitch	AddN	Low	
PSNR	44.1149	43.0243	27.0331	47.9121	27.0911	47.1632	27.4217	46.0019
NCC	1	1	0.9001	1	0.9901	1	0.7927	1
BER	0	0	1	0	1	0	1	0
Extracted watermark								

Table 4 Comparisons of previous findings with different techniques in terms of NC

Algo	MP3C	Echo	Re-quan	Re-ass	Pitch	AddN	Low
[54]	1	1	1	1	–	1	–
[63]	0.9013	0.9411	1	0.9156	0.9200	–	1
[57]	1	1	0.9013	1	0.9013	0.9013	1
[68]	1	1	–	–	1	1	0.9317
[56]	1	0.9013	1	–	–	1	1
[66]	1	0.9453	1	0.9435	–	0.9876	–
[53]	0.9595	1	0.9001	1	0.9123	1	–
[61]	–	1	0.9317	0.8664	0.9541	–	0.8643
[55]	1	1	1	0.9938	–	0.9456	1

Table 5 Comparisons of previous findings with different techniques in terms of BER (%)

Algo	MP3C	Echo	Re-quan	Re-ass	Pitch	AddN	Low
[60]	0.1994	0.9013	0	0.9013	0.9013	0.0195	0.5019
[52]	-	0	0.4435	0	0	0.9432	0.8324
[62]	0.9435	0.9143	0.8234	-	-	0	0.8911
[59]	-	0	0.9231	0.8765	-	0.9013	0
[64]	0	0.7843	-	0.0769	0	-	0.0957
[65]	0.8659	0	-	0	0.1994	0	0
[58]	-	0.9317	0	-	0.0190	0	-
[67]	0	-	0.0789	0.9013	0.9013	0.0817	0.0899
[24]	0.83(32 bps)	-	0	0.0000	-	-	-
[25]	0	-	0	0	-	-	-

5 Conclusions

A watermark that is carrying no extra bits, and which will be inserted in a different file format (host), should not mean that it is going to be a threat to the host, even though there are many techniques that ensure it remains intact yet still alters with the host conditions. There is a watermark detector which checks for the presence of the watermark inside the host file to ensure it is where it is supposed to be. Nevertheless, there is a possibility that the host will not withstand hosting the watermark, which will result in the damage or loss of the watermark file, or it can introduce perceptible distortion into the host signal. The problem which this paper attempts to address is the question of which position is best for watermarking inside a host file when spread spectrum watermarking techniques are used. To solve this problem, we used a neural network (feed-forward neural network) and designed a model that was trained to predict the best positions for the watermark in the host audio files. The model gave good predictions, and the result was formulated within the spread spectrum technique and used for embedding. Upon evaluation by experimental analysis and applying some attacks, the technique showed a good performance and was strongly robust to common signal-processing operations. We have compared the performance of our approach with other, recent audio watermarking algorithms. Overall, our technique has high embedding capacity and achieves low BER against attacks, MP3 compression 64 kbps, echo addition, re-quantization, cropping, jittering, pitch shifting, additive noise, and low-pass filtering. The unique characteristic of the method proposed in this study lies in its utilization of the positions predicted by neural network.

Acknowledgement

This research is funded under the Fundamental Research Grant Scheme (FRGS) by the Ministry of Higher Education Malaysia The work of Tutut Herawan and Haruna Chiroma is supported by University of Malaya High Impact Research Grant no vote UM.C/625/HIR/MOHE/SC/13/2 from Ministry of Higher Education Malaysia.

References

[1] Abubakar, A. I., Zeki, A. M., Chiroma, H., Muaz, S. A., Sari, E. N., & Herawan, T. Spread Spectrum Audio Watermarking Using Vector Space Projections. In *Advances in Intelligent Informatics*, 297-307. (2015).

[2] Baiying Lei, Feng Zhou, Ee-Leng Tan, Dong Ni, Haijun Lei, Siping Chen, Tianfu Wang, *Optimal and secure audio watermarking scheme based on self-adaptive particle swarm optimization and quaternion wavelet transform*, Signal Processing, 113, 80-94, (2015)

[3] Isac, B., & Santhi, V. (2011). A study on digital image and video watermarking schemes using neural networks. International Journal of Computer Applications, 12, 1-6.

[4] Yu-Cheng Fan & Yu-Yao Hsu, Novel Fragile Watermarking Scheme using an Artificial Neural Network for Image Authentication, J. Appl. Math 9, 2681-2689, (2015)

[5] F. A., Petitcolas, Ross J. Anderson, G. K. Markus, Information Hiding-A Survey. Proceedings of the IEEE, special issue on protection of multimedia content, 87:1062-1078, July 1999.

[6] M. Barni, F. Bartolini, Watermarking Systems Engineering: Enabling Digital Assets Security and other Applications. Marcel Dekker Inc, 2004.

[7] J. J. Garcia-Hernandez, R. Parra-Michel, C. Feregrino-Uribe, R. Cumplido, High payload data-hiding in audio signals based on a modified OFDM approach, J. Expert Syst Appl, 40 (2013), 3055-3064.

[8] Shi Liu, M.H. Bryan,, J.T. Sheridan, Digital image watermarking spread-space spread-spectrum technique based on Double Random Phase Encoding, J. Opt Commun, 300 (2013) 162-177.

[9] R. Bansal, P. Sehgal, P. Bedi, Minutiae Extraction from Fingerprint Images- a Review, IJCSI, 8 (2011) 74-12.

[10] B. Lei, I. Song, S.A. Rahman, Robust and secure watermarking scheme for breath sound, J. Syst Software, 86 (2013) 1638-1649.

[11] W. Zeng, R. Hu, H. Ai. Audio Steganalysis of spread spectrum information hiding based on statistical moment and distance metric, J. Multimed Tools Appl, 55 (2011) 525-556.

[12] Myasar Mundher, Dzulkifli Muhamad, Amjad Rehman, Tanzila Saba, Firdous Kausar, Digital Watermarking for Images Security using Discrete Slantlet Transform, J. Appl. Math 8, 2823-2830

[13] Hwai-Tsu Hu, Ling-Yuan Hsu, *Robust, transparent and high-capacity audio watermarking in DCT domain, Signal Processing, 109, 226-235, 2015*

[14] Loukhaoukha, K., Refaey, A., Zebbiche, K., & Nabti, M. (2015). On the Security of Robust Image Watermarking Algorithm based on Discrete Wavelet Transform, Discrete Cosine Transform and Singular Value Decomposition. *Appl. Math, 9*, 1159-1166.

[15] I.J. Cox, ; NEC Res. Inst., Princeton, NJ, USA; K. Joe, F.T. Leighton, T. Shamoon. Secure spread spectrum watermarking for multimedia. IEEE T Image Process, 6 1673 - 1687.

[16] L. Cao, C. Men, Y. Gao. A recursive embedding algorithm towards lossless 2D vector map watermarking. J. Digit Signal Process. 23 (2013) 912–918.

[17] A.M. Zeki, A. A. Ibrahim, A. A. Manaf. Steganographic software: analysis and implementation, Inter Jour of comp and comm 6 (2012).

[18] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding. IBM Systems Journal, 35 (1996) 313 – 336.

[19] G.B. Zhang, B.P. Eddy, M.Y. Hu, Forecasting with artificial neural networks: The state of the art, Int J Forecasting 14(1998) 35–62.

[20] Pranab Kumar Dhar, Tetsuya Shimamura, *Blind SVD-based audio watermarking using entropy and log-polar transformation, Journal of Information Security and Applications, Volume 20, February 2015, Pages 74-83.*

[21] Ji-Hong Chen, Wen-Yuan Chen, Chin-Hsing Chen Identification Recovery Scheme using Quick Response

(QR) Code and Watermarking Technique, J. Appl. Math, 8, 585-596, (2014)

[22] Xiao-Chen Yuan, Chi-Man Pun, C.L. Philip Chen, *Robust Mel-Frequency Cepstral coefficients feature detection and dual-tree complex wavelet transform for digital audio watermarking, Information Sciences, Volume 298, 20 March 2015, Pages 159-179*

[23] Baiying Lei, Ing Yann Soon, Perception-based audio watermarking scheme in the compressed bitstream, AEU - International Journal of Electronics and Communications, Volume 69, Issue 1, January 2015, Pages 188-197

[24] Ren Shuai, Lei Jingxiang, Zhang Tao, Duan Zongtao Fast Watermarking of Traffic Images Secure Transmission in Effective Representation Mode, J. Appl. Math 8, 2565-2569, (2014),

[25] A.B. Kumar, K. Kiran, U.S.A. Murty, C.H. Venkateswarlu, Classification, identification of mosquito species using artificial neural networks Biology and Chemistry, 32 (2008) 442-447.

[26] P. Karthigaikumar, K.J. Kirubavathy, K. Baskaran, FPGA based audio watermarking—Covert communication, J. Microelectron, 42 (2011) 778-784.

[27] S. Chen, H. Huang, C. Chen, K. T. Seng, S. Tu. Adaptive audio watermarking via the optimization point of view on the wavelet-based entropy, J. Digit Signal Process, 23 (2013) 971-980.

[28] A.M. Zeki, A.A. Ibrahim, A A. Manaf, S. M. Abdullah, Comparative Study of Different Steganographic Techniques 11th WSEAS International Conference on Applied Computer Science (ACS '11). 48- 52. WSEAS Press. Penang, Malaysia, October 3-5, 2011.

[29] T. Liang, W. Bo, L. Zhen, Z. Mingtian, An Audio Information Hiding Algorithm with High Capacity Which Based on Chaotic and Wavelet Transform, ACTA Electronica Sinica, 38 (2010) 1812-1824.

[30] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding. IBM Systems Journal, 35 (1996) 313 – 336.

[31] H. Peng, B. Li, X. Luo, J. Wang, Z. Zhang, A learning-based audio watermarking scheme using kernel Fisher discriminant analysis, J. Digit Signal Process, 23 (2013) 382-389.

[32] J.D. Gordy, L.T. Bruton, Performance evaluation of digital audio watermarking algorithms. In Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, Michigan, USA, August 8-11, 2000. 1, pp. 456 - 459.

[33] Y. Jiang, Y. Zhang, W. Pei, K. Wang, Adaptive spread transform QIM watermarking algorithm based on improved perceptual models, Aeu-Int J Electron C, 67 (2013) 690-696.

[34] N. Cvejic, T. Seppanen, Spread spectrum audio watermarking using frequency hopping and attack characterization, Signal Process, 84 (2004) 207– 213.

[35] M. Khashei, M. Bijari, Hybridization of the probabilistic neural networks with feed-forward neural networks for forecasting. ENG Appl Artif Intel, 25 (2012) 1277–1288.

[36] K. Hornick, M. Stinchcombe, H. White, Multilayer feed forward networks are universal approximators. Neural Netw 2 (1989) 359–366.

[37] R. Fletcher, Practical Methods of Optimization, 2nd edition, John Wiley, Chichester, 1987,

[38] M. A. Akhaee, N.K. Kalantari, F. Marvasti, Robust audio and speech watermarking using Gaussian and Laplacian modeling, Signal Process, 90 (2010) 2487–2497

[39] F.S. Wong, Time series forecasting using backpropagation neural networks. Neurocomputing, 2 (1991) 147–159.

[40] B.Y.Lei, F.Ing, Y. Soon, Z. Zhon, H.J. Li, A.Lei, A robust audio watermarking scheme based on lifting wavelet transform and singular value decomposition, Signal Processing 92 (2012) 1985–2001.

[41] T.Y. Pan, R.Y. Wang, State space neural networks for short term rainfall–runoff forecasting, J.Hydrol, 297 (2004) 34–50.

[42] A.Z. Taher, Fast neural network learning algorithms for medical applications, Neural Computing and Applications .DOI 10.1007/s00521-012-1026-y, 2012.

[43] Sedki, D. Ouazar, E. El Mazoudi, Evolving neural network using real coded genetic algorithm for daily rainfall–runoff forecasting. Expert Syst Appl 36 (2009) 4523–4527.

[44] R. Fletcher, Practical Methods of Optimization, 2nd edition, John Wiley, Chichester, 1987,

[45] M.H. Beale, M.T. Hagan, H.B. Demuth, Neural network toolboxTM 7 user's guide. The MathWorks, Inc. 2011, Natick.

[46] O. Kaynar, I. Yilmaz, F. Demirkoparan, Forecasting of natural gas consumption with neural networks and neuro fuzzy system. J. Energy Educ Sci Tech 26 (2011) 221 – 238.

[47] N.I. Wu, M.S. Hwang, Data Hiding: Current Status and Key Issues, Inter Jour of Network Security, 4 (2007) 1–9.

[48] J. Bennour, J.L. Dugelay, F. Matta, Watermarking Attack: BOWS contest. Proceedings of SPIE. Feb, 2011.

[49] W.N. Cheung, Digital image watermarking in spatial and transform domains. TENCON 2000 Proceedings. 3 (2000) 374-378.

[50] J.J. Eggers, J.K. Su, B. Girod, Robustness of a Blind Image Watermarking Scheme. International Conference on Image Processing (ICIP 2000).

[51] C. Hosinger, M. Rabbani, Data embedding using phase dispersion, International Conference on Information Technology: Coding and Computing (ITCC2000).

[52] Hussain, A novel approach of audio watermarking based on image-box transformation, Math Comput Model, 57 (2013) 963-969.

[53] J. Joachim, J. Eggers, G. Bernd, Robustness of a blind image watermarking scheme, ICIP 2000, Special Session on WM. Sep. 10–13. Canada.

[54] X. Wang, P. Wang, P. Zhang, S. Xu, H.Yang, A norm-space, adaptive, and blind audio watermarking algorithm by discrete wavelet transform. Signal Process, 93 (2013) 913-922.

[55] J. Bennour, J.l. Dugelay F. Matta, Watermarking Attack: BOWS contest. Proceedings of SPIE. Feb 2007.

[56] N. Chen, H. Xiao, Perceptual audio hashing algorithm based on Zernike moment and maximum-likelihood watermark detection, J. Digit Signal Process, 23 (2013) 1216-1227.

[57] S.D. Lin, C.-C. Huang, J.-H. Lin, A hybrid audio watermarking technique in cepstrum domain, ICIC Express Lett. 4 (5A) (2010) 1597–1602.

[58] H. Peng, J. Wang, Optimal audio watermarking scheme using genetic optimization, Ann. Telecommun. 66 (5–6) (2011) 307–318.

[59] K. Kondo, K. Nakagawa, A digital watermark for stereo audio signals using variable inter-channel delay in high-frequency bands and its evaluation, Int. J. Innov. Comput., Inf. Control 6 (3B) (2010) 1209–1220.

[60] D.M.L. Ballesteros, J.M.A. Moreno, Highly transparent steganography model of speech signals using Efficient Wavelet Masking, Expert Syst Appl, 39 (2012) 9141-9149.

[61] D. Megas, J. Serra-Ruiz, M. Fallahpour, Efficient self-synchronised blind audio watermarking system based on time domain and FFT amplitude modification, Signal Process, 90 (2010) 3078-3092.
[62] Baritha Begum, M. ,Venkataramani, Y., LSB based audio steganography based on text compression, Procedia Engineering, 30 (2012) 702-710.
[63] B. Lei, I. Y. Soon, F. Zhou, Z.Li, H. Lei, A robust audio watermarking scheme based on lifting wavelet transform and singular value decomposition, Signal Process, 92 (2012) 1985-2001.
[64] H. Hu, W. Chen, A dual cepstrum-based watermarking scheme with self-synchronization, Signal Process, 92 (2012) 1109-1116.
[65] S. Xiang, H. J. Kim, J. Huang, Audio watermarking robust against time-scale modification and MP3 compression, Signal Process, 88 (2008) 2372–2387.
[66] Wang, X.H. Ma, X.P. Cong, F.L. Yin, An audio watermarking scheme with neural network, in: J. Wang, X. Liao, Z. Yi (Eds.), ISNN2005, in: LNCS, vol. 3497, Springer, Heidelberg, 2005, pp. 795–800.
[67] X.-J. Xu, H. Peng, C.-Y. He, DWT-based audio watermarking using support vector regression and sub sampling, in: F. Masulli, S. Mitra, G. Pasi (Eds.), WILF2007, in: LNAI, vol. 4578, Springer, Heidelberg, Berlin, 2007, pp. 136–144.
[68] S.Q. Wu, J.W. Huang, Y.Q. Shi, Efficiently self-synchronized audio watermarking for assured audio data transmission. IEEE Trans Broadcast, 51 (2005) 69–76.

Adamu Abubakar is currently an Assistant Professor at the International Islamic University of Malaysia, Kuala Lumpur. His academic qualifications were obtained from Bayero University, Kano, Nigeria, for Bachelor and Post-graduate Diploma and Master degrees, and the International Islamic University Malaysia for PhD degree. His research area of interest is navigation and Information security. He is now working on 3D Mobile Navigation Aids and Watermarking.

Haruna Chiroma is a PhD candidate in the Department of Artificial Intelligence, University of Malaya, and a Lecturer at the Federal College of Education (Technical), Gombe, Nigeria. He received his BTech and MSc in Computer Science from Abubakar Tafawa Balewa University, Bauchi, Nigeria and Bayero University, Kano, Nigeria, respectively. His main research interest is on metaheuristic algorithms.

Akram Zeki is an Associate Professor in the Information Systems Department at the International Islamic University Malaysia. He received his PhD in Digital Watermarking from the University Technology Malaysia. His current research interest is on watermarking.

Abullah Khan is currently working as Assistant Professor in Depertment of Computer and Information Technology University of Agriculture Peshawer Pakistan. He complete his PhD from Faculty of Computer Sciences and Information Technology (FSKTM), University Tun Hussein Onn Malaysia (UTHM). He is currently applying different hybrid techniques to improve the performance accuracy in Back-Propagation Neural Networks under the supervision of Assoc. Prof. Dr. Nazri Mohd. Nawi. He obtained his MS in Computer Science from the University of Science and Technology Bannu in KPK, 2008. His research interests are in the fields of data mining, swarm optimization, and hybrid neural networks etc.

Mueen Uddin is Assistant Professor at Faculty of Computer Systems and Software Engineering, University Malaysia Pahang (UMP). He received his PhD from Universiti Teknologi Malaysia UTM in 2013. His research interests include green IT, energy efficient data centers, green metrics, virtualization and cloud computing. He received his BS and MS degrees in Computer Science from Isra University Pakistan with specialization in information networks.

Tutut Herawan received a PhD degree in computer science in 2010 from Universiti Tun Hussein Onn Malaysia. He is currently a Senior Lecturer in the Department of Information Systems, University of Malaya. His research area includes rough and soft set theory, DMKDD, and decision support in information systems.