Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9074771, 11 pages
https://doi.org/10.1155/2022/9074771

Research Article

Lightweight AAC Audio Steganalysis Model Based on ResNeXt

Zhongyuan Wei and Kaixi Wang

College of Computer Science and Technology, Qingdao University, Qingdao 266071, China

Correspondence should be addressed to Kaixi Wang; kxwang@qdu.edu.cn

Received 7 March 2022; Revised 18 March 2022; Accepted 16 April 2022; Published 20 May 2022

Academic Editor: Deepak Kumar Jain

Traditional AAC (Advanced Audio Coding) audio steganalysis methods rely on manual feature extraction, which results in low detection accuracy and low efficiency. Nowadays, the new steganalysis model based on neural network is very attractive, but its scale is large and its detection accuracy needs further improvement. Aiming at the above problems, this paper proposes a lightweight AAC audio general steganalysis model based on ResNeXt network. Firstly, the residual signal of QMDCT (Quantized Modified Discrete Cosine Transform) coefficients is calculated through a fixed convolution layer composed of multiple sets of high-pass filters. Then, based on the original structure of ResNeXt network, two ResNeXt blocks are designed to form a residual learning module, by which the steganalysis features in the QMDCT coefficients are further extracted. Finally, the classification module consisting of the fully connected layer and the Softmax layer is designed to obtain the classification result. The experimental results show that the model detection accuracy can reach more than 94% under all relative embedding rates when it operates on both the steganography algorithm based on the small value area of the QMDCT coefficient and the steganography algorithm based on the Huffman code sign bit. For the algorithm based on Huffman codeword mapping, even with the relative embedding rate of 0.1, the detection accuracy of the model can reach 85.5%, which is obviously better than the existing steganalysis schemes. Compared with other steganalysis schemes based on neural network, the model in this paper has fewer parameters, and reduces the scale by more than 40%, which is more lightweight and more efficient.

1. Overview

Digital steganography is a technology of hiding a secret into digital carriers, which is often used for covert communication. Common carriers include images, audios, texts, and so on. However, steganography is easy to be maliciously used by illegal organizations or individuals. The opponent of steganography, namely steganalysis, has been widely studied to detect whether secret messages are hidden in digital media.

With the rapid development of the network and multimedia technologies, a large number of compressed audios have been widely spread on the Internet. AAC and MP3 are two most popular audio compression standards. Compared with MP3 with the same bit rate, AAC has obvious advantages in encoding quality and compressed volume. In many compressed audio applications, the AAC audio compression standard gradually replaces the MP3 audio compression standard. The wide spread of AAC audio compression standard has spawned a variety of steganography algorithms for AAC compressed audio.

Reference [1] (after this referred to as the MIN method) performed steganographic embedding by modifying the quantized coefficients in the small value region of the MDCT (modified discrete cosine transform) coefficients. Reference [2] (after this referred to as the SIGN method) realized the embedding of secret information by modifying the sign bits of the quantization coefficients in the MDCT coefficients that are less than a certain threshold. Reference [3] used matrix coding to modify the least significant bit of the Huffman coding escape sequences to realize the embedding of secret information. Reference [4] (after this referred to as the HCM method) classified Huffman codewords and embeds secret information by replacing the corresponding Huffman codewords. Reference [5] realized the embedding of secret information based on the modified sections of the Huffman coding of AAC. Reference [6] proposed an adaptive AAC steganography scheme based on distortion minimization model. Reference [7] proposed a secure AAC steganography scheme based on multiview statistical distortion. Although the above steganography algorithms have different embedding domains of secret information, they embed secret messages by modifying the compression parameters of AAC audio, which will eventually lead to different degrees of modification of QMDCT coefficient of AAC Audio. Therefore, QMDCT coefficients contain the steganalysis features.

There are few steganalysis methods about AAC at this stage. In order to detect the steganography algorithm in the AAC Huffman coding domain, reference [8] extracted the Markov transition probability of the adjacent scale factor band's codebook as the steganalysis feature and used the calibration technology to improve the detection accuracy. In order to detect the steganography algorithm in the AAC modified discrete cosine transform domain, reference [9] proposed to design steganalysis features by extracting multiorder differences between interframe and intraframe of MDCT coefficients. When the relative embedding rate is greater than 50%, good steganalysis performance is achieved. Reference [10] proposed a steganalysis method to detect AAC steganography algorithms by extracting statistical features of QMDCT coefficients. Because MP3 and AAC have a similar theoretical basis, the steganalysis method for MP3 has a certain reference value for AAC audio steganalysis. Reference [11] extracted the difference between the quantization step sizes of adjacent frames of MP3 as the steganalysis features. Reference [12] proposed to extract the statistical distribution difference of the number of bits in the audio bit pool before and after recompression as the steganalysis features. Reference [13] proposed an MP3 steganalysis method by deriving a combination of features from quantized MDCT coefficients, which include frequency-based subband moment statistical features, accumulative Markov transition features, and accumulative neighboring joint density features on second-order derivatives. Reference [14] proposed an effective MP3 steganalysis algorithm by extracting joint point-wise and block-wise correlations of quantified modified discrete cosine coefficient matrix. However, all of the above solutions are based on manual feature extraction, with poor performance and poor generality, and cannot meet the needs of the current stage.

With the successful application of deep learning technology in various fields, neural networks are gradually applied to the detection of audio steganography algorithms. Reference [15] applied DBN (Deep Belief Network) to speech steganalysis task for the first time, which improved the detection accuracy. However, this method did not take advantage of neural network in adaptive feature extraction. Reference [16] proposed a convolutional neural network to detect ±1 LSB (least significant bit) steganography in the temporal domain, which achieved better results than steganalysis algorithms based on handcrafted features. Based on the theory that the steganalysis modification of different embedded domains of compressed audio will change the decoded audio signal, reference [17] proposed a general steganalysis scheme suitable for detecting different embedded domains of AAC and MP3. However, for the complex AAC steganalysis algorithm, the detection performance of this scheme needs to be further improved. In order to detect the MP3 steganography algorithm in the Huffman coding domain, reference [18] proposed a steganalysis method based on convolutional neural network. The experimental results show that the CNN-based scheme performs better than the handcrafted steganalysis features. Although this method can be used to detect AAC audio steganography algorithm, the detection performance is still poor. The number of model parameters is large, which is not conducive to deployment to the production environment.

To sum up, the neural network-based audio steganalysis method achieves better performance than traditional steganalysis methods based on manual feature extraction. However, there are still many problems that remain unsolved. The classification accuracy is not high enough when the relative embedding rate is low. The existing neural network models are relatively complex and large in scale and cannot achieve the purpose of high efficiency. How to take advantage of the feature extraction of neural network, improve the detection accuracy of steganalysis model, and improve the performance of neural network is still a very challenging problem.

This paper proposes a lightweight general steganalysis method named LARXNet based on ResNeXt to detect AAC steganography algorithms. The main contributions of our work are summarized as follows: (1) in order to extract the available features introduced by the AAC audio steganography algorithm, the QMDCT coefficient matrix is adopted as the input of the neural network. Almost all AAC audio steganography methods lead to changes in the QMDCT coefficients of AAC audio, so the QMDCT coefficients contain the noise information generated by the steganography algorithms. (2) We use a set of high-pass filters to preprocess the input data and suppress the adverse effect of the audio signal itself on steganalysis. Then, the residual data of QMDCT coefficients and the original data are concatenated in depth to make full use of the input data. (3) In order to more effectively extract the classification features of steganography schemes, referring to the design experience of existing steganalysis methods, we designed two basic ResNeXt blocks based on ResNeXt [19] and built a lightweight deep residual network. The detailed parameters of the network model are adjusted through experiments. LARXNet does not suffer from gradient vanishing and network degradation problems and can automatically extract weak steganalysis features. The experimental results show that LARXNet greatly improves the detection performance of three classic AAC audio steganography algorithms compared with the existing audio steganalysis methods.

The rest of this paper is organized as follows. The relevant knowledge is introduced in Section 2. The scheme of this paper is presented in Section 3. And Section 4 shows the experimental setup and results. Finally, the conclusions are drawn in Section 5.

2. Relevant Knowledge

2.1. AAC Audio Encoding. AAC is a perceptual compression coding standard based on a psychoacoustic model, and its basic working principle is shown in Figure 1.

The AAC coding process is mainly divided into four steps: MDCT filtering, spectral processing, coefficient quantization, and noiseless coding. MDCT is a critical step in the encoding process. AAC coding adopts the method of time-domain aliasing, and each frame contains 1024 time-domain sampling points, which together with the 1024 sampling points of the previous frame form 2048 time-domain signals. And cooperate with the psychoacoustic model to confirm the corresponding window type, perform MDCT filtering according to the window, and obtain 1024 spectral coefficients. After time-domain noise shaping, joint stereo coding module prediction, and other frequency-domain processing, the obtained spectral coefficients are quantized. The quantized coefficients are then subjected to Huffman noiseless coding. Finally, the bitstream is packaged to get standard AAC audio.

2.2. AAC Steganography Algorithm. A variety of AAC steganography algorithms have emerged in recent years. Although these steganography algorithms are embedded in different stages and positions during AAC encoding, the basic idea behind these steganography schemes still fine-tunes the compression parameters to hide secret information. This section will introduce steganography schemes for three different embedding methods.

2.2.1. MIN Algorithm. The small value area is the area where the QMDCT coefficients are {-1, 0, 1}, which are basically concentrated in the middle- and high-frequency parts. If it is closer to the high frequency, the quantized coefficient magnitude is basically {-1, 0, 1}. The quantization error in this part is relatively large, and the bits used in the encoding process are relatively small. Therefore, small changes to the small value region have little effect on the audio quality. The QMDCT coefficients in the small value area are generally coded by the No. 1 and No. 2 codebooks of the Huffman codebook. In the coding process, the index is obtained by taking every four quantized coefficients as a group. The specific calculation formula is

$$\begin{aligned} \text{index} = 27 * \text{quant}[i] + 9 * \text{quant}[i+1] \\ + 3 * \text{quant}[i+2] + \text{quant}[i+3] + 40, \end{aligned} \tag{1}$$

where i is the serial number of a group of spectral coefficients. Then, according to the index of the corresponding Huffman code table, the corresponding codeword is searched for Huffman encoding. The MIN algorithm performs information hiding by modifying the last of a set of quantized coefficients. The modification rules used are as follows:

(1) When quant $[i+3]=0$, if the embedded bit is 0, the quantization coefficient quant $[i+3]$ remains unchanged. If the embedded bit is 1, then the quantization coefficients quant $[i+3]$ are random ±1

(2) When quant $[i+3]=1$, if the embedded bit is 0, quant $[i+3]$ is changed to 0. If the embedded bit is 1, it remains unchanged

(3) When quant $[i+3]=-1$, if the embedded bit is 0, then quant $[i+3]$ is modified to 0. If the embedded bit is 1, then quant $[i+3]$ remains unchanged

2.2.2. SIGN Algorithm. The AAC encoding process uses multiple Huffman codebooks. AAC coding adopts the Huffman codebook with the optimal number of coding bits for coding. The codebooks 3, 4, 7, 8, 9, 10, and 11 are all Huffman codebooks with signed bits. The sign bits of the nonzero quantized coefficients are appended to the corresponding Huffman codeword in order. Among them, each Huffman codeword in the codebooks 3 and 4 encode four quantization coefficients, and the data stream format is shown in Figure 2.

Each Huffman codeword in the codebooks 7, 8, 9, and 10 encode two quantization coefficients, and the data stream format is shown in Figure 3.

Each Huffman codeword in the codebook 11 encode two quantized coefficients and uses escape sequences to represent quantized coefficients greater than 16. The data stream format is shown in Figure 4.

As shown in the figure, Huffman_code represents the Huffman codeword, representing the frequency coefficients w, x, y, and z. Sign_w, Sign_x, and Sign_y, and Sign_z are the corresponding sign bits of the nonzero frequency coefficients. There may be 0~4 sign bits, each of which is 1 bit. A sign bit of 0 indicates that the coefficient is negative, and a sign bit of 1 indicates that the coefficient is positive.

In this way, the sign bit of the quantized coefficient can be replaced with a binary secret message for steganography. In order to minimize the distortion caused by modifying the sign bits of the coefficients, a threshold α is set. Then, the sign bits of the quantized coefficients between -α and α are modified to embed the secret message, and the undesired quantized coefficients are skipped. This will not modify any other parameters and ensure the good imperceptibility of the algorithm.

2.2.3. HCM Algorithm. The Huffman codeword itself has no redundant space that can be used for steganography, and the bits of the Huffman codeword cannot be directly modified for information embedding, but the Huffman codeword can be replaced with another approximate Huffman codeword to embed the secret message.

The AAC audio compression standard has strict regulations. According to the characteristics of the Huffman codebook, arbitrarily modifying or replacing the Huffman codeword may cause the code stream structure of AAC audio to be confused; then, it cannot be decoded normally. Huffman codewords are also known as VLC codewords. The VLC codewords in the two codeword spaces for codeword mapping must satisfy the following conditions:

$$\text{Code\_len}(vlc_i) = \text{Code\_len}(vlc_j), \tag{2}$$

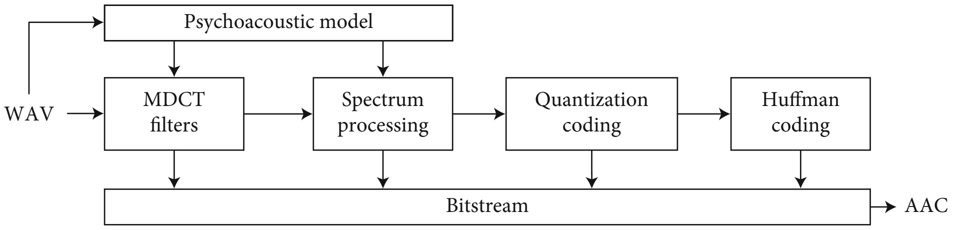


Figure 1: AAC audio coding procedure.

...	Huffman_code	Sign_w	Sign_x	Sign_y	Sign_z	...

Figure 2: Bitstream structure of codebooks 3 and 4.

...	Huffman_code	Sign_x	Sign_y	...

Figure 3: Bitstream structure of codebooks 7, 8, 9, and 10.

$$\mathrm{Sign}(vlc_i) = \mathrm{Sign}\left(vlc_j\right), \tag{3}$$

$$\left(w_i - w_j\right) + \left(x_i - x_j\right) + \left(y_i - y_j\right) + \left(z_i - z_j\right), \tag{4}$$

where $\mathrm{Code\_len}(vlc_i)$ is the codeword length of the codeword vlc_i. $\mathrm{Sign}(vlc_i)$ is each sign bit of the codeword vlc_i. w_i, x_i, y_i, and z_i are the quantization coefficients represented by the codeword vlc_i. i and j are the numbers of the codewords.

The VLC codewords satisfying Equations (1), (2), and (3) are divided into two types of codeword spaces, which are defined as V_0 and V_1, respectively.

$$\begin{aligned} V_0 &= \{v_0, v_1, \cdots, v_i, \cdots, v_I\}, \\ V_1 &= \left\{v_0, v_1, \cdots, v_j, \cdots, v_I\right\}, \end{aligned} \tag{5}$$

where v_i and v_j represent the VLC codewords in the codeword spaces V_0 and V_1, respectively. $i = 0, 1, \cdots, I$ is the codeword number in the codeword space V_0. $j = 0, 1, \cdots, I$ is the codeword number in the codeword space V_0. $I + 1$ represents the length of codeword space. The remaining codewords form the codeword space V_2. The codeword spaces V_0 and V_1 are used to represent binary information "0" and "1," respectively. Therefore, the HCM algorithm can be described as follows: if the secret information is "0," the VLC codeword should belong to the codeword space V_0. If the secret information is "1," the VLC codeword should belong to the codeword space V_1. If the current codeword does not meet the requirements, it needs to be replaced with a codeword corresponding to another codeword space. When the codeword belongs to the codeword space V_2, the current codeword is skipped and the embedding operation is not performed.

2.3. ResNeXt Network. The ResNeXt [19] network is an improved network of the ResNet [20]. The ResNeXt combines the ideas of ResNet identity mapping and residual learning with the idea of segmentation, transformation, and merging of the Inception structure in GoogleNet [21]. The structure of the original ResNeXt block is shown in Figure 5(a). "$\mathrm{Conv}(N, 1 \times 1, p)$" means that the kernel size of the convolutional layer is 1×1, the input channels are N, and the output channels are p. It adopts ResNet's strategy of easily expanding repeated layers, dividing high-dimensional convolutions into multiple groups of low-dimensional convolutions, and each group of low-dimensional convolutions uses the same topology. Then, the features of all grouped low-dimensional convolutions are combined to obtain the final convolution result. The number of divided groups is called cardinality. Reference [19] shows that increasing the cardinality can achieve better results than increasing the depth and width of the network.

As shown in Figure 5(b), the original ResNeXt block can be expressed based on grouped convolution [22] more succinctly. All 1×1 convolutional layers can be replaced with a single wider 1×1 convolutional layer. The grouped convolution can divide the input channel into multiple branches, which can perfectly implement the segmentation operation required by the ResNeXt block. The parentheses are the number of input channels, the size of the convolution kernel, and the number of convolution kernels, and c is the number of groups of grouped convolutions. Since the equivalent form based on grouped convolution is concise and easy to implement, our network model is constructed using the equivalent form.

3. Scheme of This Paper

In order to apply the ResNeXt residual network to AAC audio steganalysis work, this paper designs the corresponding network structure according to the relevant characteristics of audio steganalysis. The overall structure and design details of the steganalysis model in this paper will be introduced in detail below.

3.1. The Overall Structure. The network structure is shown in Figure 6. The first part is a high-pass filter and a merge operation, which combines the high-pass filtered features and the original features into new features in depth, as the input of the first convolutional layer. The second part consists of a normal convolutional layer and a residual learning module containing three sets of ResNeXt modules. The size of the convolution kernel of the first convolution layer is 3×3, the number of convolution kernels is 64, and the step size is 1. The residual learning module includes three groups,

...	Huffman_code	Sign_x	Sign_y	Escape_sequence	Escape_sequence	...

Figure 4: Bitstream structure of codebook 11.

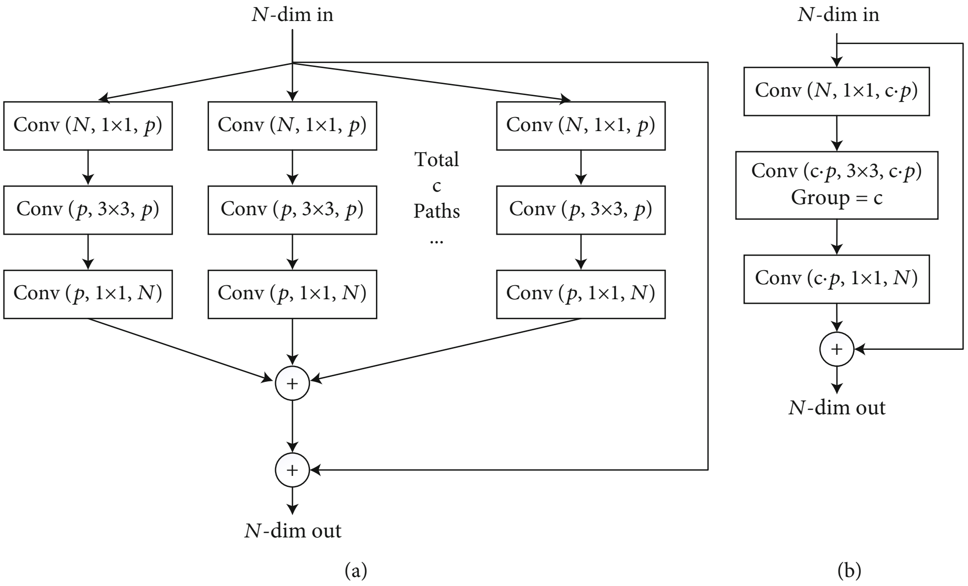


Figure 5: Original architecture of ResNeXt block.

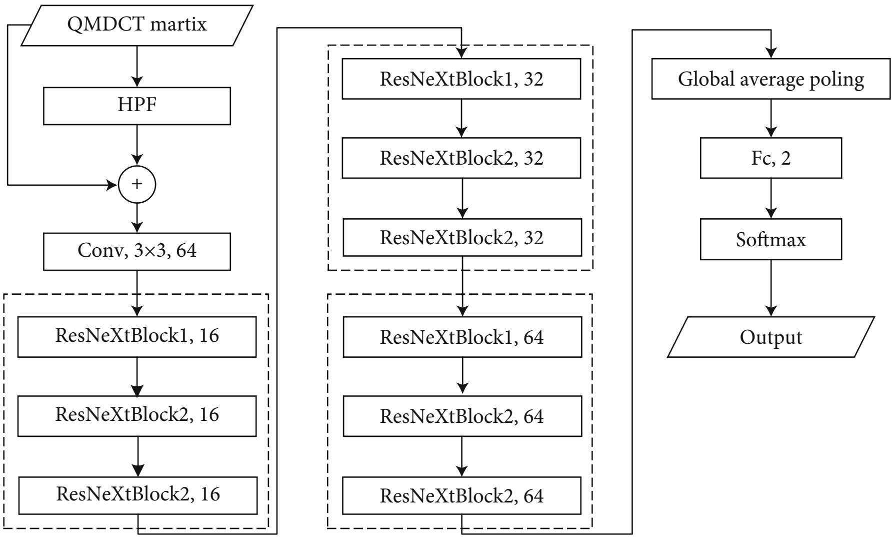


Figure 6: Overall structure of the proposed model.

and each group includes one ResNeXtBlock1 module and two ResNeXtBlock2 modules, which are arranged in order. The number after the module name is the number of convolution kernels for group convolution in the module. The number of group convolution kernels for the three groups is 16, 32, and 64, respectively. The third part is the classification module, which includes a global average pooling and a fully connected layer. Global average pooling converts the feature map of the residual learning module into a 128-dimensional feature vector and feeds it to the fully connected layer. Finally, the Softmax classifier is used to obtain the corresponding classification probability as the final steganalysis result.

3.2. QMDCT and Preprocessing Modules. The network uses QMDCT coefficients as input data. The QMDCT coefficients play a key role in the AAC coding process. The time-domain audio signal is converted into frequency-domain MDCT coefficients through an MDCT filter and then quantized to obtain QMDCT coefficients. Finally, the QMDCT coefficients are noise-free encoded by Huffman coding and packed into audio data stream together with other compression parameters. Huffman coding is a lossless coding, so QMDCT coefficients and Huffman coding correspond to each other. Regardless of the MIN, SIGN, or HCM algorithm, the modifications caused by the embedding process are reflected in the QMDCT coefficients. Previous studies have shown that the statistical features of the QMDCT matrix are effective for MP3 steganography. MP3 and AAC have similar coding principles. The idea of steganalysis for MP3 can be used for reference to AAC steganalysis. In addition, reference [9] proposes an AAC steganalysis method based on QMDCT coefficients, which shows that QMDCT is effective in AAC steganalysis.

AAC stereo contains 1024 coefficients per channel per frame. 25 frames of AAC audio are used as a basic analysis unit to form a 50×1024 QMDCT coefficient matrix. For a better description, the coefficient matrix is expressed as

$$\begin{bmatrix} Q_{1,1} & \cdots & Q_{1,j} & \cdots & Q_{1,1024} \\ \vdots & \ddots & \vdots & & \vdots \\ Q_{i,1} & \cdots & Q_{i,j} & \cdots & Q_{i,1024} \\ \vdots & & \vdots & \ddots & \vdots \\ Q_{50,1} & \cdots & Q_{50,j} & \cdots & Q_{50,1024} \end{bmatrix}, \tag{6}$$

where $Q_{i,j}$ represents the QMDCT coefficient. $i \in \{1, 2, 3, \cdots ,50\}$ is the index of the channel. $j \in \{1, 2, 3, \cdots, 1024\}$ is the index of the QMDCT coefficients in one channel.

The AAC audio encoding process performs lossy compression of audio time-domain signals. Although the lossy compression process will cause a little loss to the original signal, in a short time, the MDCT coefficients can still show continuity in the time and frequency domains, that is, the correlation between frames within a frame. The preprocessing module used in reference [17] has a good effect, which can enhance the correlation between frames, suppress the influence of carrier content on steganalysis features, and improve the accuracy of steganalysis. This paper follows the high-pass filter (HPF) used in reference [17] to extract more steganalysis features.

3.3. Residual Learning Module. Referring to the design experience of audio steganalysis network, this paper designs two kinds of ResNeXt blocks based on the original ResNeXt block structure, as shown in Figure 7.

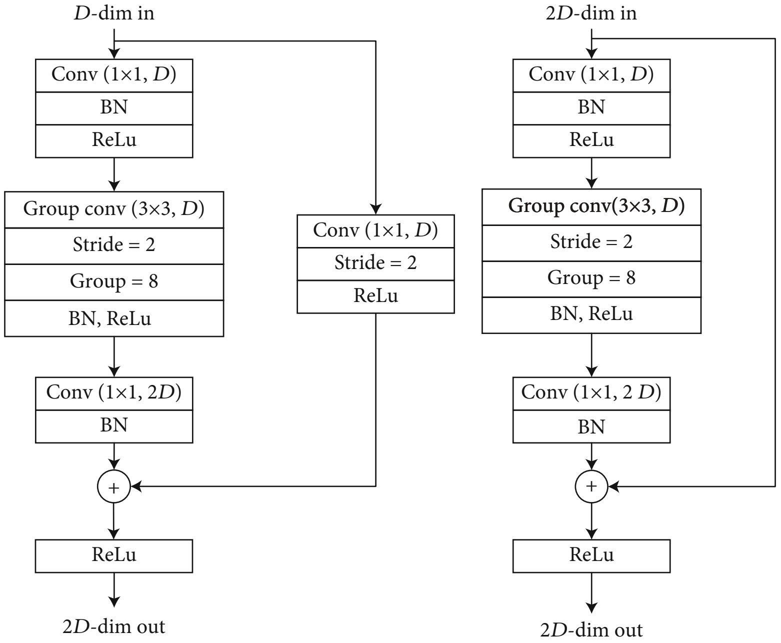
Considering the scale and actual performance of the network, this paper sets the cardinality of the ResNeXt block to 8. Each ResNeXt block consists of three convolutional layers and a shortcut connection. The three convolutional layers are 1×1 convolution, 3×3 group convolution, and 1×1 convolution in sequence. Except for the last convolutional layer, the other convolutional layers are followed by a BN (batch normalization) layer, and the ReLU activation function is used to increase the nonlinearity of network. The output of the last convolution layer is added to the input characteristics passed by the shortcut and activated using the ReLU activation function. The group convolution stride of the ResNeXtBlock1 module is 2, which can reduce the size of the feature map to half the size of the input feature map and replace the pooling layer. Steganalysis feature loss caused by pooling layers can be reduced. However, the input and output feature map dimensions of the ResNeXtBlock1 module are inconsistent, so the shortcut connection needs to be mapped using 1×1 convolution to ensure that the input and output dimensions are consistent. The experiments also test the performance of target shortcut mapping using 3×3 convolutions. Taken together, using 1×1 convolution as target shortcut mapping has the best performance. The group convolution stride of the ResNeXtBlock2 module is 1, and the input and output feature maps have the same dimensions, and the input and output of the module can be directly added.

4. Experimental Results and Analysis

4.1. Experimental Setup. There is no publicly available AAC audio dataset at this stage. Reference [18] provides a large WAV audio dataset with a sampling frequency of 44.1 KHz and a length of 10 s. 15,000 WAV audios are selected and encoded into M4A files with a bitrate of 128 Kbps by AAC audio encoder. Then, three steganography algorithms, MIN [1], SIGN [2], and HCM [4], are implemented to generate the stego-AAC audio files. For the three steganography algorithms, the random secret information is embedded in the audio files during the encoding process at the relative embedding rate of 0.1, 0.2, 0.3, 0.5, and 1.0, so that we get 15000 cover-stego pairs. Relative embedding rate is the percentage of embedded message length to the maximum embedded message length. Among them, 9000 cover-stego pairs are used as training set, 3000 cover-stego pairs are used as validation set, and the remaining 3000 cover-stego pairs are used as test set.

The model in this paper is built using the TensorFlow deep learning framework. The experimental software environment is Windows 10 64-bit operating system, and the hardware environment is Intel (R) Xeon (R) W-2133 CPU, NVIDIA GTX 2080Ti 11 GB GPU, and 32 GB memory. Adam optimizer with $\beta 1 = 0.9$, $\beta 2 = 0.999$, and $\epsilon = 10^{-7}$ was used to update the network parameters. The network is trained with an initial learning rate of 0.001. We adopted the learning rate strategy that the validation loss as a monitor was employed in the training stage, and if the validation loss does not decrease within 5 epochs, the current learning rate will be reduced with a decay rate of 0.9. The batch size during the training stage is set to 32 (16 cover-stego pairs). The weights of all convolutional layers and fully connected layers are initialized via the Xavier uniform initializer [23]. And the initialization of biases is zero. The parameters of the model with the best performance on validation accuracy are saved every epoch. The average detection accuracy on the test set is used as the experimental evaluation metric.

4.2. The Influence of Model Structure on Detection Accuracy. Different network structures will affect the final detection



(a) ResNeXtBlock1

(b) ResNeXtBlock2

Figure 7: Structure of ResNeXtBlock1 and ResNeXtBlock2.

accuracy. In order to obtain the optimal model structure, a lot of experiments have been done with different network structures for the HCM steganography algorithm with a relative embedding rate of 0.3. In order to reduce the randomness of the experiments, 10 experiments are performed for each different network structure. The average detection accuracy of 10 experiments was taken as the final detection result. Table 1 shows the modified network structure parameters and the corresponding average detection accuracy. Among them, model #1 is the network structure proposed in this paper. The average detection accuracy of LARXNet is 95.13%, which can achieve the best steganalysis performance and is significantly better than other network structures. The average detection accuracy of the #2 model is 93.56%. It can be seen that the high-pass filter layer is beneficial to capture better steganographic feature information and improve the detection accuracy of the model. The average detection accuracy of the #3 model is 94.25%. A set of modules has been added, and a large number of parameters have been added, but the result is not better, it should be that the model has overfitted. Both the #4 model and the #5 model have adjusted the cardinality of ResNeXt module. The number of parameters of the ResNeXt module is related to the cardinality; the larger the cardinality, the smaller the number of parameters. According to the experimental results, considering the scale of the lightweight model in this paper, setting the cardinality of the ResNeXt module to 8 can obtain the best network performance. The average detection accuracy of the #6 model is 94.10%. It is proved that using 1×1 convolution on the target shortcut is better than 3×3 convolution, and using 3×3 convolution will nearly double the parameters and increase the training time of the network. Compared with other variant models, LARXNet has the highest detection accuracy, which is the most effective.

4.3. Experimental Comparative Analysis. For the three steganography algorithms MIN, SIGN, and HCM, four existing audio steganalysis methods are compared with LARXNet under different relative embedding rates, including two traditional audio steganalysis methods based on manual feature extraction (MDI2 [9] and JPBC [14]) and two neural network-based audio steganalysis models (Spec-ResNet [17] and WASDN [18]). The comparison results with methods based on manual feature extraction are shown in Table 2. The average detection accuracy of 10 experiments was taken as the evaluation index. The comparison results with the steganalysis model based on neural network are shown in Table 3. In addition to selecting the average detection accuracy as the evaluation index, the number of model parameters and model size are also compared to measure the scale of the modulus. In order to clearly show the change of detection accuracy, Figures 8–10 are drawn at the same time as a visual display of the data in Tables 2 and 3.

From the data in Table 2, the detection difficulty of our model for the three steganography algorithms is ranked as HCM, SIGN, and MIN from difficult to easy. Due to the relatively large modification of the audio carrier by the MIN algorithm, the high-frequency noise generated by steganography is relatively strong. It is easier to detect by steganalysis tools. For the three steganography algorithms, with the improvement of the relative embedding rate, the modification of original covers will grow up and the detection accuracy of various steganalysis schemes also will increase. The detection accuracy of LARXNet for the MIN algorithm reaches more than 98%. The detection performance of traditional audio steganalysis methods depends on the selection and extraction of features, which requires researchers to be proficient in the characteristics of steganography algorithms. The designed features are highly targeted, and it is not easy

Table 1: Detection accuracies of different network structures.

Number	Network structure parameters	Accuracy (%)
#1	Proposed model—LARXNet	95.13
#2	Remove the high-pass filter layer	93.56
#3	A set of ResNeXt modules are added	94.25
#4	The cardinality of ResNeXt module is adjusted to 16	93.85
#5	The cardinality of ResNeXt module is adjusted to 4	93.82
#6	ResNeXtBlock1 shortcut replaced with 3 × 3 convolution kernel	94.10

Table 2: Comparison of detection results of traditional steganalysis methods.

Steganalysis method	Steganography algorithm	Accuracy (%)				
		0.1	0.2	0.3	0.5	1.0
Our model—LARXNet	MIN	98.65	99.05	99.12	99.34	99.90
	SIGN	94.27	96.65	98.54	99.12	99.87
	HCM	85.50	91.65	95.13	96.85	99.55
MDI2	MIN	56.12	64.35	75.12	86.43	97.66
	SIGN	54.83	62.47	68.45	71.55	74.48
	HCM	57.84	63.23	69.20	78.57	92.60
JPBC	MIN	63.25	74.52	81.45	92.38	99.92
	SIGN	55.81	57.42	61.42	65.07	76.08
	HCM	62.79	73.02	80.05	91.69	99.35

Table 3: Comparison of detection results of neural network steganalysis methods.

Steganalysis model	Steganography algorithm	Accuracy (%)					Parameters (K)	Model size (MB)
		0.1	0.2	0.3	0.5	1.0		
Our model—LARXNet	MIN	98.65	99.05	99.12	99.34	99.90	102	1.49
	SIGN	94.27	96.65	98.54	99.12	99.87		
	HCM	85.50	91.65	95.13	96.85	99.55		
Spec-ResNet	MIN	94.31	95.72	96.43	98.81	99.85	206	2.49
	SIGN	92.12	93.29	94.79	98.48	99.82		
	HCM	72.50	81.53	88.03	93.27	98.35		
WASDN	MIN	93.57	94.52	95.96	98.70	99.61	73000	692.27
	SIGN	90.24	92.19	93.25	98.35	99.57		
	HCM	69.32	73.47	81.10	85.32	92.95		

to extract comprehensive features. Therefore, for the JPEC steganalysis algorithm and the MDI2 steganalysis algorithm, the detection performance of the HCM algorithm is significantly higher than that of the SIGN algorithm. The possible reason is that two traditional steganalysis algorithms cannot accurately extract the noise characteristics of the SIGN algorithm. Our model based on the latest ResNext network can automatically extract the steganographic noise features in the audio carrier and exert the powerful feature extraction and feature fusion capabilities of the neural network, which is very suitable for steganalysis tasks. Therefore, our model can still achieve an excellent classification effect when the relative embedding rate is low, with an average increase of more than 30%. Even for the most difficult HCM algorithm to detect, when the relative embedding rate is 0.1, the detection accuracy of our model can reach 85.5%. For the three steganography algorithms, the detection accuracy of our model is higher than that of traditional steganalysis methods based on handcrafted features. Overall, LARXNet effectively detects existing AAC steganography algorithms at different relative embedding rates.

From the data in Table 3, it can be seen that the number of parameters of the model in this paper is the least, and the size of the model is the smallest. Compared with the optimal Spec-ResNet, the model size is reduced by more than 40%. Because the WASDN model does not use the global average

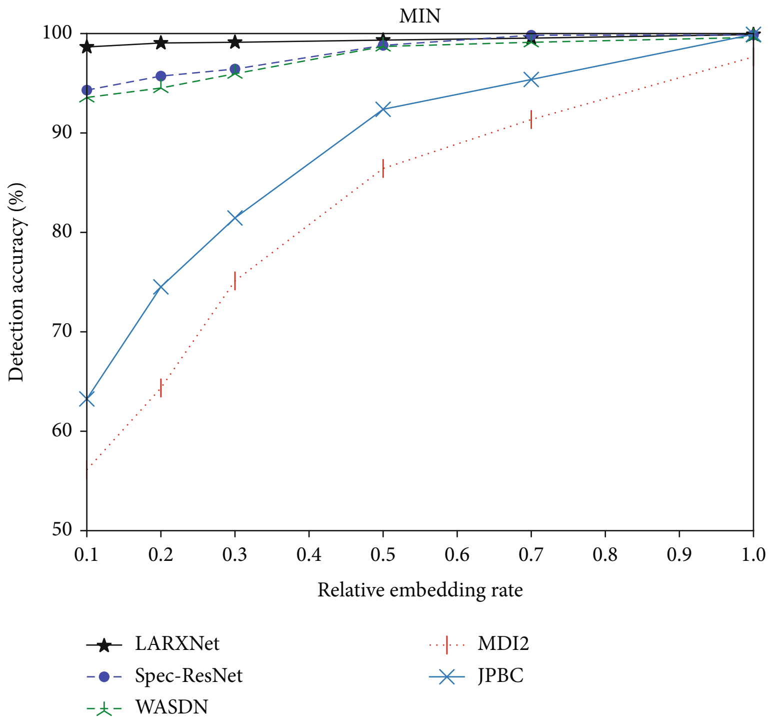


Figure 8: Detection accuracy of MIN algorithm.

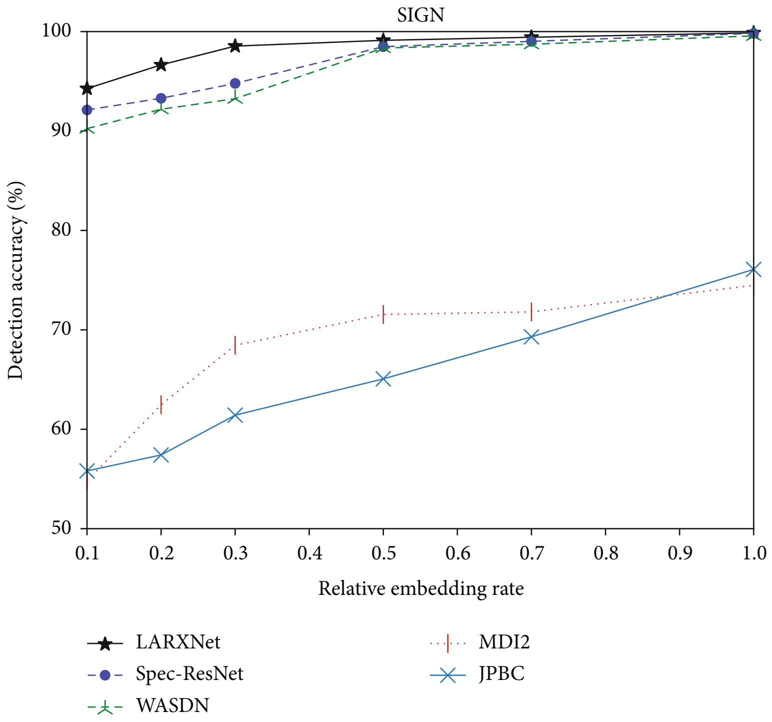


Figure 9: Detection accuracy of SIGN algorithm.

pooling layer, all fully connected layers need to use a large number of parameters, accounting for the vast majority of the total number of parameters. So the number of parameters of the WASDN model reaches 73 million. For the MIN algorithm and the SIGN algorithm, the three steganalysis models can achieve more than 90% detection accuracy at a low relative embedding rate of 0.1. At a relative embedding rate of 1.0, the accuracy of all three models is close to 100%. This is because the MIN algorithm and the SIGN algorithm modify the audio carrier to a large extent, and the noise generated during the embedding process is stronger. At the same time, the steganalysis method based on

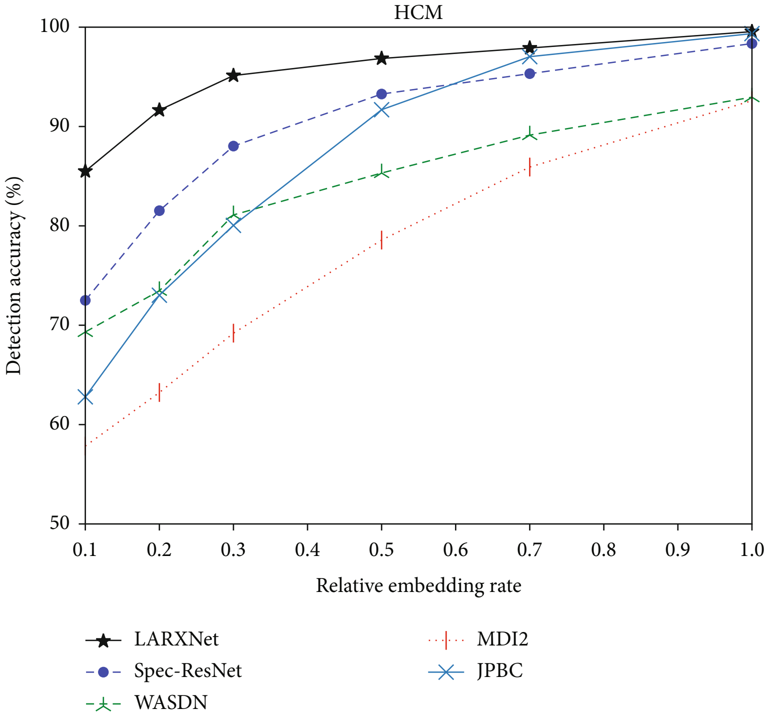


Figure 10: Detection accuracy of HCM algorithm.

neural network has powerful feature extraction ability and can effectively extract the feature information of MIN algorithm and SIGN algorithm. For the HCM algorithm that is more difficult to detect, our model has obvious advantages. Compared with Spec-ResNet model, the accuracy is improved by 13 percentage points at the embedding rate of 0.1. In general, the neural network-based steganalysis method does not require the manual design of complex features and can automatically extract more comprehensive steganalysis features with better detection performance than traditional steganalysis methods. Our model combines the advantages of ResNet and GoogleNet, which not only improves the detection performance of the model but also reduces the number of parameters of the model. Compared with WASDN, our model adopts a residual structure, which can alleviate the problems of network overfitting and network degradation and speed up model training. Compared with Spec-ResNet and WASDN, our model adopts grouped convolution and uses the strategy of *split-transform-merge*, which can extract more diverse and comprehensive steganalysis features. At the same time, this structure can reduce network parameters and network scale. Therefore, our model can achieve optimal results. Thanks to the powerful feature extraction capability of ResNeXt, the model in this paper has the best detection results under various relative embedding rates and is more suitable for complex steganalysis environments.

5. Conclusion

This paper proposes a lightweight network model based on the ResNeXt network to detect AAC steganography algorithms. Firstly, the data is preprocessed with a high-pass filter; then, our model is carefully designed based on the ResNeXt model, and our model is fine-tuned experimentally. Finally, we design comparative experiments to verify the effectiveness of LARXNet. The experimental results show that our model can be used to detect a variety of AAC steganography algorithms with different relative embedding rates. Although the relative embedding rate is relatively low, LARXNet still has excellent detection performance. For the HCM algorithm with a relative embedding rate of 0.1, the detection accuracy of LARXNet is 85.5%, which is 13 percentage points higher than Spec-ResNet. Our model significantly outperforms existing audio steganalysis schemes under the same experimental conditions. Compared with the neural network-based audio steganalysis method, LARXNet requires fewer parameters and storage space, reduces the model size by more than 40%, and has better detection accuracy, which can be applied to more scenarios, such as mobile devices. However, new adaptive AAC steganography algorithms have recently appeared, and there is no targeted steganalysis scheme yet. Our model also needs to be further improved to detect adaptive steganography methods. In the future, we will continue to optimize the network structure according to the adaptive steganography algorithms' characteristics and improve our model's detection performance for the adaptive steganography algorithms.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Qingdao Shinan District Science and Technology Plan Project Public Domain Science and Technology Support Project (No. 2022-2-025-XX).

References

[1] Y. J. Wang, L. Guo, and C. P. Wang, "Steganography method for advanced audio coding," *Journal of Chinese Computer Systems*, vol. 32, no. 7, pp. 1465–1468, 2011.

[2] J. Zhu, R. Wang, and D. Yan, "The sign bits of Huffman codeword-based steganography for AAC audio," in *2010 International Conference on Multimedia Technology*, pp. 1–4, Ningbo, China, 2010.

[3] Y. Wang, L. Guo, Y. Wei, and C. Wang, "A steganography method for AAC audio based on escape sequences," in *2010 International Conference on Multimedia Information Networking and Security*, pp. 841–845, Nanjing, China, 2010.

[4] Z. Jie, *The research on information hiding in MPEG-2/4 advanced audio coding, [Ph.D. Thesis]*, Ningbo University, Ningbo, 2012.

[5] J. Zhu, R. D. Wang, J. Li, and D. Q. Yan, "A Huffman coding section-based steganography for AAC audio," *Information Technology Journal*, vol. 10, no. 10, pp. 1983–1988, 2011.

[6] Z. Zhang, X. Yi, and X. Zhao, "An AAC steganography scheme for adaptive embedding with distortion minimization model," *Multimedia Tools and Applications*, vol. 79, no. 37-38, pp. 27777–27790, 2020.

[7] Y. Ren, S. Cai, and L. Wang, "Secure AAC steganography scheme based on multi-view statistical distortion (SofMvD)," *Journal of Information Security and Applications*, vol. 59, article 102863, 2021.

[8] Y. Ren, Q. Xiong, and L. Wang, "Steganalysis of AAC using calibrated Markov model of adjacent codebook," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2139–2143, Shanghai, China, 2016.

[9] Y. Ren, Q. Xiong, and L. Wang, "A steganalysis scheme for AAC audio based on MDCT difference between intra and inter frame," in *International Workshop on Digital Watermarking*, pp. 217–231, Magdeburg, Germany, 2017.

[10] Y. J. Wang, P. Yang, and W. W. Jiang, "A steganalysis method of AAC audio based on statistical features of MDCT quantized coefficients," *Journal of Hefei University of Technology (Natural Science)*, vol. 38, no. 10, pp. 1348–1352, 2015.

[11] D. Yan, R. Wang, X. Yu, and J. Zhu, "Steganalysis for MP3Stego using differential statistics of quantization step," *Digital Signal Processing*, vol. 23, no. 4, pp. 1181–1185, 2013.

[12] D. Yan and R. Wang, "Detection of MP3Stego exploiting recompression calibration-based feature," *Multimedia Tools and Applications*, vol. 72, no. 1, pp. 865–878, 2014.

[13] M. Qiao, A. H. Sung, and Q. Liu, "MP3 audio steganalysis," *Information Sciences*, vol. 231, pp. 123–134, 2013.

[14] Y. Wang, X. Yi, and X. Zhao, "MP3 steganalysis based on joint point-wise and block-wise correlations," *Information Sciences*, vol. 512, pp. 1118–1133, 2020.

[15] C. Paulin, S. A. Selouani, and E. Hervet, "Audio steganalysis using deep belief networks," *International Journal of Speech Technology*, vol. 19, no. 3, pp. 585–591, 2016.

[16] B. Chen, W. Luo, and H. Li, "Audio steganalysis with convolutional neural network," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, pp. 85–90, Philadelphia, PA, USA, 2017.

[17] Y. Ren, D. Liu, Q. Xiong, J. Fu, and L. Wang, "Spec-ResNet: a general audio steganalysis scheme based on deep residual network of spectrogram," 2019, https://arxiv.org/abs/1901.06838.

[18] Y. Wang, K. Yang, X. Yi, X. Zhao, and Z. Xu, "CNN-based steganalysis of MP3 steganography in the entropy code domain," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pp. 55–65, Innsbruck, Austria, 2018.

[19] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition.*, pp. 1492–1500, Honolulu, Hawaii, 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, America, 2016.

[21] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, Boston, America, 2015.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1–9, 2012.

[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 2010.