entropy

MDPI

Article

Comparative Study of Three Steganographic Methods Using a Chaotic System and Their Universal Steganalysis Based on Three Feature Vectors

Dalia Battikh [1], Safwan El Assad [2,*], Thang Manh Hoang [3], Bassem Bakhache [1], Olivier Deforges [4] and Mohamad Khalil [1]

1 LASTRE Laboratory, Lebanese University, 210 Tripoli, Lebanon
2 Institut d'Electronique et des Télécommunications de Rennes (IETR), UMR CNRS 6164, Université de Nantes—Polytech Nantes, Rue Christian Pauc CS 50609, CEDEX 3, 44306 Nantes, France
3 School of Electronics and Telecommunications, Hanoi University of Science and Technology, 1 Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam
4 INSA de Rennes, CNRS, IETR, CEDEX 7, 35708 Rennes, France
* Correspondence: safwan.elassad@univ-nantes.fr

Received: 27 June 2019; Accepted: 24 July 2019; Published: 30 July 2019

Abstract: In this paper, we firstly study the security enhancement of three steganographic methods by using a proposed chaotic system. The first method, namely the Enhanced Edge Adaptive Image Steganography Based on LSB Matching Revisited (EEALSBMR), is present in the spatial domain. The two other methods, the Enhanced Discrete Cosine Transform (EDCT) and Enhanced Discrete Wavelet transform (EDWT), are present in the frequency domain. The chaotic system is extremely robust and consists of a strong chaotic generator and a 2-D Cat map. Its main role is to secure the content of a message in case a message is detected. Secondly, three blind steganalysis methods, based on multi-resolution wavelet decomposition, are used to detect whether an embedded message is hidden in the tested image (stego image) or not (cover image). The steganalysis approach is based on the hypothesis that message-embedding schemes leave statistical evidence or structure in images that can be exploited for detection. The simulation results show that the Support Vector Machine (SVM) classifier and the Fisher Linear Discriminant (FLD) cannot distinguish between cover and stego images if the message size is smaller than 20% in the EEALSBMR steganographic method and if the message size is smaller than 15% in the EDCT steganographic method. However, SVM and FLD can distinguish between cover and stego images with reasonable accuracy in the EDWT steganographic method, irrespective of the message size.

Keywords: steganography; chaotic system; steganalysis; wavelet; feature vector; SVM; FLD

1. Introduction

Steganography is an increasingly important security domain; it aims to hide a message (secret information) in digital cover media without causing perceptual degradation (in this study, we use images as cover media). It should be noted that many steganographic methods have been proposed in the spatial and frequency domains. In the spatial domain, pixels are directly used to hide secret messages; these techniques are normally easy to implement and have a high capacity. However, they are not generally robust against statistical attacks [1,2]. In the transform domain, coefficients of frequency transforms, such as DCT (Discrete Cosine Transform), FFT (Fast Fourier Transform), and DWT (Discrete Wavelet Transform), are used to hide secret data. Generally, these techniques are complex, but they are more robust against steganalysis (to noise and to image processing).

The main steganographic methods in the spatial domain [3–17] are LSB-based (Low Significant Bit). Recently, entropy has also been extensively used to support data-hiding algorithms [18–20]. The LSB methods entail replacing the least significant bit of pixels with a bit of the secret data. Among these methods, the EALSBMR method [3] is an edge adaptive scheme with respect to the message size and can embed data according to the difference between two consecutive pixels in the cover image. To the best of our knowledge, we conclude that this method is the best (good PNSR, high embedding capacity, and especially adaptive), but it suffers from low security in terms of message detection. For this reason, we have enhanced its security.

Frequency domain steganography, as a watermarking domain [21–29], is widely based on the DCT and DWT transforms. The DCT usually transforms an image representation into a frequency representation by grouping pixels into 8×8 pixel blocks and transforming each block, using the DCT transform, into 64 DCT coefficients. A message is then embedded into the DCT coefficients. The Forward Discrete Wavelet Transform is, in general, suitable for identifying areas in the cover image where a secret message can be effectively embedded due to excellent space-frequency localization properties. In particular, these properties allow exploiting the masking effect of a human visual system so that if a DWT coefficient is modified, it modifies only the region that corresponds to that coefficient. The Haar wavelet is the simplest possible wavelet that can achieve the DWT.

However, the aforementioned steganographic methods are not secure in terms of message detection. To protect the content of messages, chaos can be used. Indeed, chaotic sequences play an important role in information hiding and in security domains, such as cryptography, steganography, and watermarking, because of their properties such as sensitivity to initial conditions and parameters of the system, ergodicity, uniformity, and pseudo-randomness. Steganography generally leaves traces that can be detected in stego images. This can allow an adversary, using steganalysis techniques, to divulge a hiding secret message. There are two types of opponents: passive and active. A passive adversary only examines communication to detect whether communication contains hidden messages. In this case, the content of the communication is not modified by the rival. An active adversary can intentionally cause disruption, distortion, or destruction of communication, even in the absence of evidence of secret communication. The main steganographic methods have been designed for cases of passive adversary. In general, there are two kinds of steganalysis: specific and universal. Specific steganalysis is designed to attack a specific steganography algorithm. This type of specific steganalysis can generally produce more accurate results, but it fails to produce satisfactory results if the inserted secret messages are in the form of a modified algorithm. Universal steganalysis, on the other hand, can be regarded as a universal technique to detect various types of steganography. Moreover, it can be used to detect new steganographic techniques where specific steganalysis does not yet exist. In other words, universal steganalysis is an irreplaceable tool for detection if the integration algorithm is unknown or secret.

In this paper, we first integrate an efficient chaotic system into the three steganographic methods mentioned above to make them more secure. The chaotic system quasi-chaotically chooses pixel positions in the cover image where the bits of the secret message will be embedded. Thus, the inserted bits of the secret message becomes secure against message bits recovery attacks because their position is unknown.

Second, we study and apply three universal steganalysis methods to the aforementioned chaos-based steganographic methods. The first steganalysis method, developed by Farid [30], uses higher-order statistics of high-frequency wavelet sub-bands and their prediction errors to form the feature vectors. In the second steganalysis method, as formulated by Shi et al. [31], the statistical moments of the characteristic functions of the prediction-error image, the test image, and their wavelet sub-bands are selected as the feature vectors. The third steganalysis method, introduced by Wang et al. [32], uses the features that are extracted from both the empirical probability density function (PDF) moments and the normalized absolute characteristic function (CF). For the three steganalysis

algorithms, we applied FLD analysis and the SVM method with the RBF kernel as classifiers between cover images and stego images.

The paper has been organized as follows: In Section 2, we describe the proposed chaotic system. In Section 3, we present the three enhanced steganographic algorithms. In Section 4, we illustrate the experimental results and analyze the enhanced algorithms. In Section 5, we develop, in detail, the steganalysis techniques for the previous algorithms. In Section 6, we report the results of the steganalysis, and in the last section, we conclude our work.

2. Description of the Proposed Chaotic System

This system is made of a perturbed chaotic generator and a 2-D cat map. The chaotic generator supplies the dynamic keys K_p for the process of provides the position of the new random pixel (see Figure 1). The chaotic system allows inserting a message both in a secretive and uniform manner [33–40].

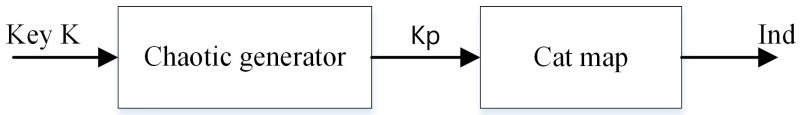


Figure 1. Proposed chaotic generator.

The generator of discrete chaotic sequences exhibits orbits with very large lengths. It is based on two connected non-linear digital IIR filters (cells). The discrete PWLCM and SKEW TENT maps (non-linear functions) are used. A linear feedback shift register (m-LFSR) is then used to disturb each cell (Figure 2). The disturbing technique is associated with the cascading technique, which allows controlling and increasing the length of the orbits that are produced. The minimum orbit length of the generator output is calculated using Equation (1):

$$o_{min} = lcm\left\{\Delta_1 \times \left(2^{k_1} - 1\right), \Delta_2 \times \left(2^{k_2} - 1\right)\right\} \tag{1}$$

In the above equation, lcm is the least common multiple, $k_1 = 23$ and $k_2 = 21$ are the degrees of the LFSR's primitive polynomials, and Δ_1 and Δ_2 are the lengths s_1 and s_2 of outputs cells, respectively, without disturbance. The equations of the chaotic generators are formulated as follows:

$$\begin{aligned} &s_i(n) = NLF_i\left\{u_i(n-1), p_i\right\}, i = 1,2 \\ &u_i(n-1) = mod\left\{s_i(n-1) \times c_{i,1} + s_i(n-2) \times c_{i,2}, 2^N\right\}, i = 1,2 \\ &s(n) = s_1(n) + s_2(n) \end{aligned} \tag{2}$$

The two previously mentioned functions, PWLCM map and Skew map, are defined according to the following relations:

$$\begin{aligned} s_1(n) &= NLF_1\left\{u_1(n-1), p_1\right\} \\ &= \begin{cases} \left\lfloor 2^N \times \frac{u_1(n-1)}{p_1} \right\rfloor & if \quad 0 \le u_1(n-1) < p_1 \\ \left\lfloor 2^N \times \frac{2^N - u_1(n-1)}{2^N - p_1} \right\rfloor & if \quad p_1 \le u_1(n-1) < 2^{N-1} \\ NLF_1\left[2^N - u_1(n-1)\right] & otherwise \end{cases} \end{aligned} \tag{3}$$

$$\begin{aligned} s_2(n) &= NLF_2\left[u_2(n-2), p_2\right] \\ &= \begin{cases} \left\lfloor 2^N \times \frac{u_2(n-1)}{p_1} \right\rfloor & if \quad 0 \le u_2(n-1) < p_2 \\ \left\lfloor 2^N \times \frac{2^N - u_2(n-1)}{2^N - p_2} \right\rfloor + 1 & if \quad p_2 \le u_2(n-1) < 2^N \end{cases} \end{aligned} \tag{4}$$

The control parameter p_1 is used for the PWLCM map and ranges from 1 to $2^{N-1}-1$, and p_2 is the control parameter that is used for the Skew map and ranges from 1 to 2^N-1. $N = 32$ is the word length used for simulations. The size of the secret key K, formed by all initial conditions and parameters of the chaotic generator, is ($6 \times 32 + 5 \times 32 + 31 + 23 + 21$) = 427 bits. It is large enough to resist a brute-force attack.

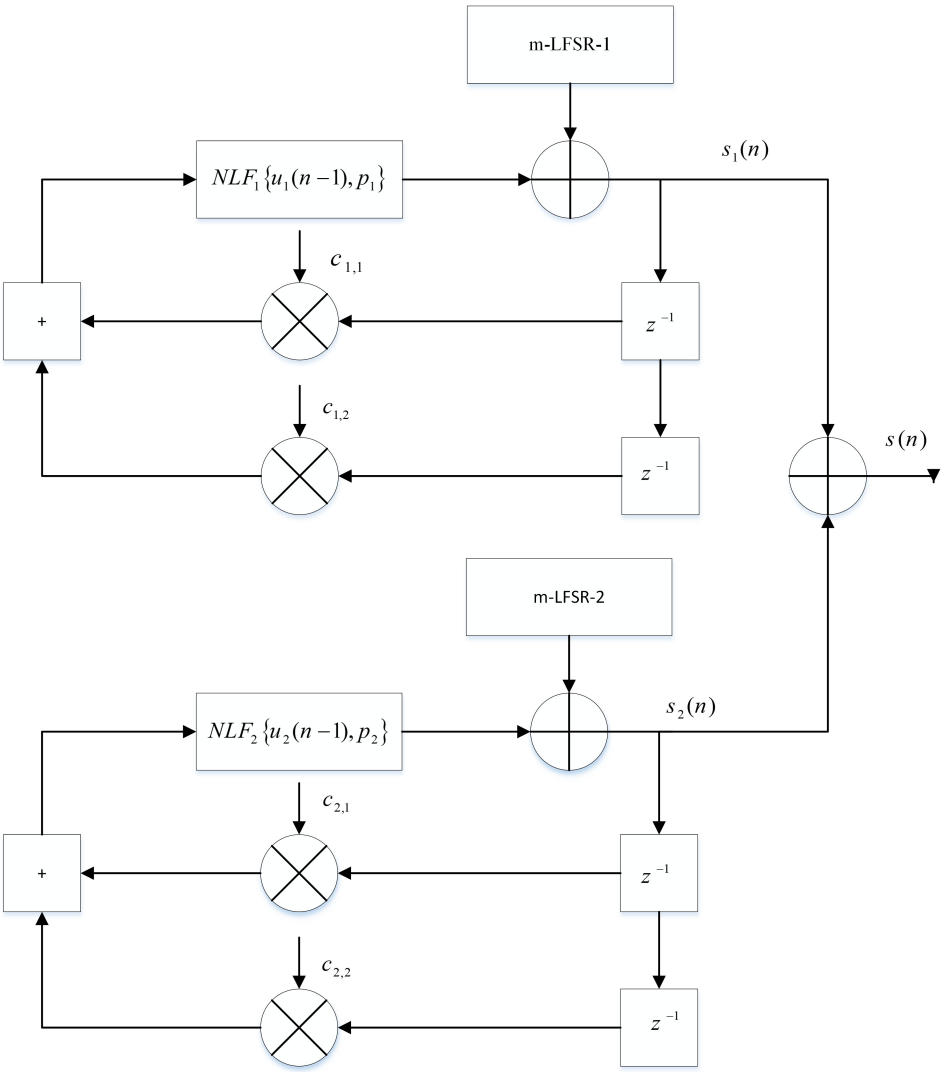


Figure 2. Chaotic generator.

Description of the Cat Map Used

The permutation process is based on the modified Cat map and is calculated in a very efficient manner using the equation below [37]:

$$\begin{bmatrix} M_{cn} \\ M_{ln} \end{bmatrix} = mod\left\{ \begin{pmatrix} 1 & u \\ v & 1+uv \end{pmatrix} \times \begin{pmatrix} M_l \\ M_c \end{pmatrix} + \begin{bmatrix} r_l + r_c \\ r_c \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix} \right\} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (5)$$

In the above equation, (M_l, M_c) and (M_{ln}, M_{cn}) are the original and permuted square matrices of size (M, M), from which we calculate the Ind matrix as follows:

$$M_l = \begin{pmatrix} 1 & 1 & . & . & 1 \\ 2 & 2 & . & . & 2 \\ . & & . & & . \\ . & & . & & . \\ M & M & . & . & M \end{pmatrix}; M_c = \begin{pmatrix} 1 & 2 & . & . & M \\ 1 & 2 & . & . & M \\ . & & . & & . \\ . & & . & & . \\ 1 & 2 & . & . & M \end{pmatrix}$$

$$Ind = (M_{ln} - 1) + (M_{cn} - 1) \times M + 1$$

The dynamic key K_p is structured as follows:

$$K_p = \left[k_{p_1}, k_{p_2}, ..., k_{p_r}\right]$$

$$k_{p_i} = \{u_i, v_i, rl_i, rc_i\}; i = 1, 2, ..., r$$

In the above equations, $0 \leq u_i, v_i, rl_i, rc_i \leq M - 1$ are the parameters of the Cat map and r is the number of rounds.

3. Enhanced Steganographic Algorithms

In this section, we describe three enhanced steganographic algorithms by using an efficient chaotic system.

3.1. Enhanced EALSBMR (EEALSBMR)

Below, we present the insertion procedure and the extraction procedure of the proposed enhancement of the EALSBMR method (EEALSBMR) [41].

3.1.1. Insertion Procedure

The flow diagram of the embedding scheme can be found in Figure 3.

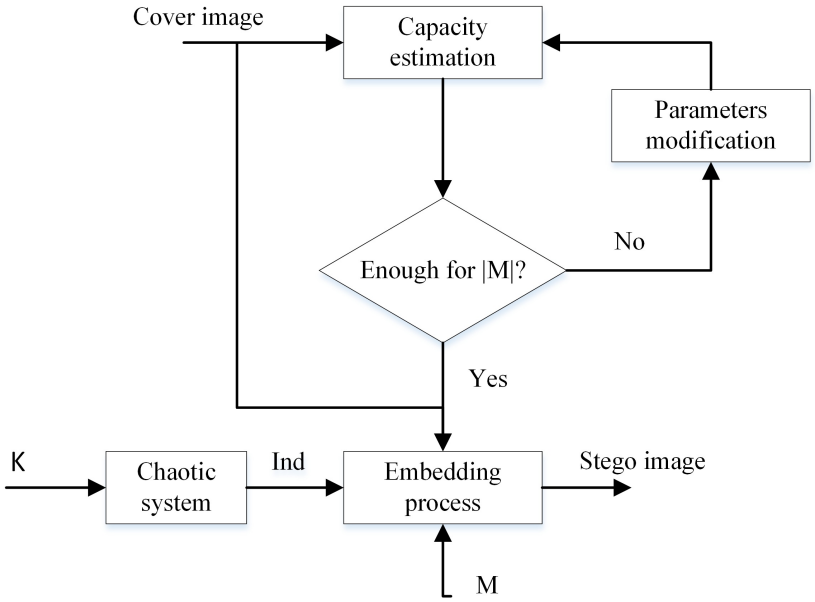


Figure 3. EEALSBMR insertion procedure.

The detailed embedding steps for this algorithm have been explained as follows:

Step 1: Capacity estimation

- To estimate the insertion capacity, we arrange the cover image into a 1D vector V, and we divide its content into non-overlapping embedding units (blocks) consisting of two consecutive pixels (p_i, p_{i+1}). Following this, we calculate the difference between the pixels of each block, and we increase by one the content of the vector-difference VD of 31 elements $t \in \{1, 2, 3, ..., 31\}$, in which each element contains $|EU(t)|$ number of blocks where $EU(t)$ is a set of pixel pairs whose absolute differences are greater than or equal to t, as shown below:

$$EU(t) = \{(p_i, p_{i+1}) \,||\, |p_i - p_{i+1}| \geq t, \forall (p_i, p_{i+1}) \in V\} \tag{6}$$

- For a given secret message M of size $|M|$ bits, the threshold T used in the embedding process is determined by the following expression and pseudo-code (Algorithm 1):

$$T = argmax_t \{2 * |EU(t)| \geq |M|\} \tag{7}$$

Algorithm 1 Pseudo-code determining the value of the threshold T

```
procedure
    number_pixels = 0;
    for t = 31:-1:1 do
        number_pixels = number_pixels + VD(t);
        if (2*number_pixels >= |M|) then
            T = t;
            break;
        end if;
    end for;
end procedure
```

Step 2: Embedding process

- The embedding process is achieved as follows: we divide the cover image into two sub-images; one includes the odd columns, and the other includes the even columns.
- Following this, the chaotic system chooses a pixel position (Ind) from the odd sub-image; the second pixel position of the corresponding block must have the same Ind in the even image. If a pair of pixel units (p_i, p_{i+1}) satisfies Equation (8), then a 2 bit-message can be hidden (one bit by pixel); otherwise, the chaotic system chooses another Ind.

$$(|p_i - p_{i+1}| \geq T, \forall (p_i, p_{i+1}) \in V) \tag{8}$$

- For each unit (p_i, p_{i+1}), we perform data-hiding based on the following four cases [42]:

Case 1: if $LSB(p_i) = m_i$ and $f(p_i, p_{i+1}) = m_{i+1} \rightarrow (p_i^{'}, p_{i+1}^{'}) = (p_i, p_{i+1})$
Case 2: if $LSB(p_i) = m_i$ and $f(p_i, p_{i+1}) \neq m_{i+1} \rightarrow (p_i^{'}, p_{i+1}^{'}) = (p_i, p_{i+1} + r)$
Case 3: if $LSB(p_i) \neq m_i$ and $f(p_i - 1, p_{i+1}) = m_{i+1} \rightarrow (p_i^{'}, p_{i+1}^{'}) = (p_i - 1, p_{i+1})$
Case 4: if $LSB(p_i) \neq m_i$ and $f(p_i - 1, p_{i+1}) \neq m_{i+1} \rightarrow (p_i^{'}, p_{i+1}^{'}) = (p_i + 1, p_{i+1})$

In the above equations, m_i and m_{i+1} are the ith and $(i+1)$th secret bits of the message to be embedded; r is a random value belonging to $\{-1, 1\}$, and $(p_i^{'}, p_{i+1}^{'})$ denotes the pixel pair after data-hiding. The function f is defined as follows:

$$f(a, b) = LSB(\left\lfloor \frac{a}{2} \right\rfloor + b) \tag{9}$$

- Readjustment if necessary: After hiding, $(p_i^{'}, p_{i+1}^{'})$ may be out of range [0, 255] or the new difference value $\left|p_i^{'} - p_{i+1}^{'}\right|$ may be less than the threshold T. In these cases, we need to readjust $p_i^{'}$ and $p_{i+1}^{'}$, and the new readjusted values, $p_i^{''}$ and $p_{i+1}^{''}$, are calculated as follows [3]:

$$(\, p_i^{''}, p_{i+1}^{''}) = argmin_{(e_1, e_2)} \left\{ \left|e_1 - p_i^{'}\right| + \left|e_2 - p_{i+1}^{'}\right| \right\} \tag{10}$$

with :

$$\begin{cases} e_1 = p_i^{'} + 4k_1 \\ e_2 = p_{i+1}^{'} + 2k_2 \end{cases} k_1, k_2 \in \mathbb{Z} \tag{11}$$

k_1, k_2 are two arbitrary numbers from $\mathbb{Z}$; when:

$$0 \leq e_1, e_1 \leq 255 \quad \text{and} \quad |e_1 - e_2| \geq T \tag{12}$$

then :

$$\begin{aligned} p_i^{''} &= e_1 \\ p_{i+1}^{''} &= e_2 \end{aligned} \tag{13}$$

The sequence follows as such for each new block position.

- Finally, we embed the parameter T of the stego image into the first five pixels or the last five pixels, for example.

3.1.2. Extraction Procedure

- Extract the parameter T from the stego image.
- Divide the stego image into two sub-images; one includes the odd columns, and the other includes the even columns.
- Generate a pseudo-chaotic position (using the same secret key K), as done in the insertion procedure, to obtain the same order of pixel unit position as the odd sub-image. The second pixel block has the same Ind in the even image.
- Verify if $\left|p_i^s - p_{i+1}^s\right| \geq T$ and then extract the two secret bits of M (m_i, m_{i+1}) as follows:

$$m_i = LSB(p_i^s); \quad m_{i+1} = f(p_i^s, p_{i+1}^s) \tag{14}$$

with : $p_i^s = p_i^{'} \quad$ or $\quad p_i^{''}$
Otherwise, the chaotic system chooses another pseudo-chaotic position. The sequence follows as such for each unit position until all messages have been extracted.

- Example of insertion:
The cover image is this image of "peppers" as in Figure 4:

Figure 4. "Peppers" as cover image.

The embedded message appears as follows in 40 × 40 pixels as shown in Figure 5:

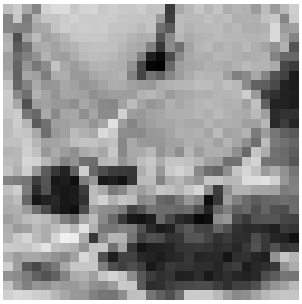


Figure 5. "Bike" is as embedded message.

The corresponding sequence of the bits message has been given as follows:

$$M = 1000100010001100100011000110011100100111101001011$$

$$1110101100011010101110100000011010001011001 0...$$

The length of the binary message is 13,120 bits.

Capacity estimation produces the threshold $T = 12$

Suppose that the pseudo-chaotic positions of a block to embed the two bits message $m_1 = 1$ and $m_2 = 0$ are (354, 375) and (354, 376) that correspond to the 141 and 129 gray values (see Figure 6).

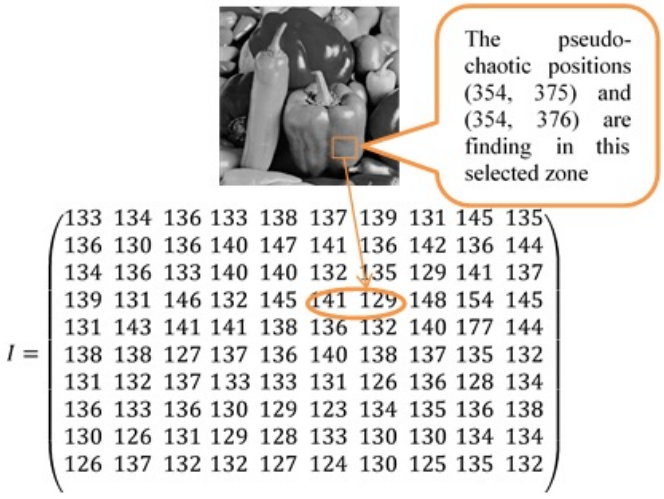


Figure 6. Pseudo-chaotic block selection and its corresponding gray value.

Hiding the message bits:

$$LSB(141) = 1 = m_1 = 1$$

$$f(p_1, p_2) = LSB(\left\lfloor \frac{p_1}{2} \right\rfloor + p_2) = LSB(70 + 129) = 1 \neq m_2$$

We are in Case 2:

$$LSB(p_i) = m_i; f(p_i, p_{i+1}) \neq m_{i+1}$$

Therefore, the new pixel values are as follows:

$$(p_1^{'}, p_2^{'}) = (p_1, p_2 + r) = (141, 130) \qquad \text{with} \qquad r = 1$$

The difference between the new pixel values is:

$$d^{'} = \left| p_1^{'} - p_2^{'} \right| = |141 - 130| = 11 < T$$

Then we need to adjust the new pixel values:
We test the values $-50 < k_1 < 50$ and $-50 < k_2 < 50$ until we obtain the smallest difference between the initial values $p_1^{'}$ and $p_2^{'}$ and the corresponding obtained values e_1 and e_2 by using Equations (12) and (13). In our example, we find $k_1 = 0$ and $k_2 = -1$ and then: $p_1^{''} = 141$, $p_2^{''} = 128$.

- Extraction of the bits message in the previous insertion example:
  The extraction is performed using the following equation:

$$m_1 = LSB(p_1^{''}) = LSB(141) = 1$$

$$m_2 = f(p_1^{''}, p_2^{''}) = LSB(\left\lfloor \frac{p_1^{''}}{2} \right\rfloor + p_2^{''})) = LSB(70 + 128) = LSB(198) = 0$$

3.2. Enhanced DCT Steganographic Method (EDCT)

The DCT transforms a signal or image from the spatial domain into the frequency domain [43,44]. A DCT expresses a sequence of finitely many data points in terms of a sum of cosine functions, oscillating at different frequencies. The 2D DCT is calculated as follows:

$$DCT_{i,j} = \alpha_i \alpha_j \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} C_{mn} \cos \frac{\pi(2m+1)i}{2M} \cos \frac{\pi(2n+1)j}{2N} \tag{15}$$

where:

$$\alpha_i = \begin{cases} \frac{1}{\sqrt{M}} & i = 0 \\ \sqrt{\frac{2}{M}} & 0 \leq i \leq M-1 \end{cases} \qquad \alpha_j = \begin{cases} \frac{1}{\sqrt{N}} & i = 0 \\ \sqrt{\frac{2}{N}} & 0 \leq i \leq N-1 \end{cases}$$

The block diagram of the proposed enhanced steganographic-based DCT transform has been shown in Figure 7.

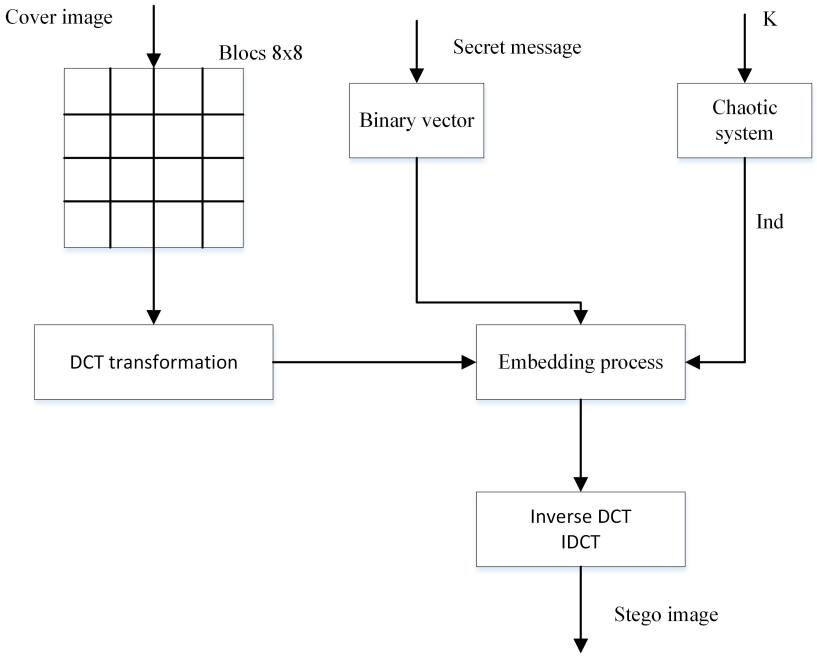


Figure 7. Diagram of the enhanced steganographic-based DCT transform.

3.2.1. Insertion Procedure

The embedding process consists of the following steps:

- Read the cover image and the secret message.
- Convert the secret message into a 1-D binary vector.
- Divide the cover image into 8×8 blocks. Then apply the 2D DCT transformation to each block (from left to right, top to bottom).
- Use the same chaotic system to generate a pseudo-chaotic Ind.
- Replace the LSB of each located DCT coefficient with the one bit of the secret message to hide.
- Apply the 2D Inverse DCT transform to produce the stego image.

3.2.2. Extraction Procedure

The extraction procedure consists of the following steps:

- Read the stego image.
- Divide the stego image into 8×8 blocks and then apply the 2D DCT to each block.
- Use the same chaotic system to generate pseudo-chaotic Ind.
- Extract the LSB of each pseudo-located coefficient.
- Construct the secret image.

3.3. Enhanced DWT Steganographic Method (EDWT)

The embedded secret image in the lower frequency sub-band (A) is generally more robust than the other sub-bands, but it significantly decreases the visual quality of the image, as normally, most of the image energy is stored in this sub-band. In contrast, the edges and textures of the image and the human eye are not generally sensitive to changes in the high-frequency sub-band (D); this allows secret information to be embedded without being perceived by the human eye. However, the sub-band (D) is not robust against active attacks (filtering, compression, etc.). The compromise adopted by many DWT-based algorithms to achieve accepted performance of imperceptibility and robustness enables embedding the secret image in the middle-frequency sub-bands (H) or (V). In the block diagram of the proposed steganographic EDWT method shown in Figure 8, we embed the secret image in the

sub-band (H) of the cover image (the size of the secret message must, at most, be equal to the size of the sub-band (H) of the cover image).

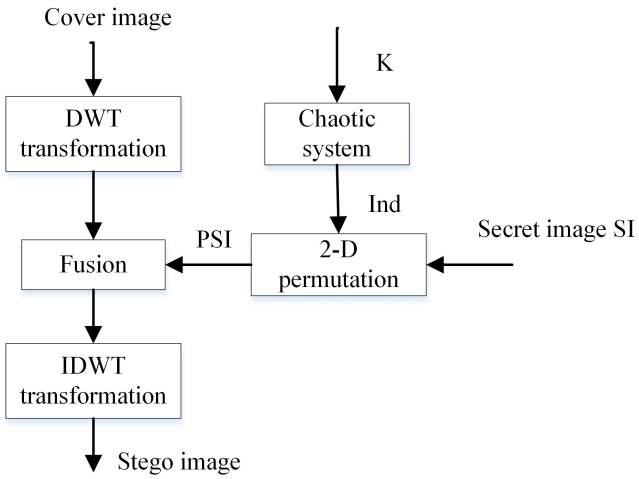


Figure 8. Diagram of the EDWT algorithm.

3.3.1. Insertion Procedure

The embedding process consists of the following steps:

- Read the cover image and the secret image.
- Transform the cover image into one level of decomposition using Haar Wavelet.
- Permute the secret image in a pseudo-chaotic manner.
- Fuse the DWT coefficients (H) of the cover image and the permuted secret image PSI as follows [45]:

$$\begin{aligned} X^{'} &= \alpha X + \beta \times PSI \\ \alpha + \beta &= 1; \quad \alpha \gg \beta \end{aligned} \tag{16}$$

  In the above equations, $X^{'}$ is the modified DWT coefficient (H); X is the original DWT coefficient (H). α and β are the embedding strength factors; they are chosen such that the resulting stego image has a large $PSNR$. In our experiments, we tested some values of β, and the best value was found to be approximately 0.01.
- Apply Inverse Discrete Wavelet Transform (IDWT) to produce the stego image in the spatial domain.

3.3.2. Extraction Procedure

The extraction procedure involves the following steps:

- Read the stego image.
- Transform the stego image into one level of decomposition using Haar Wavelet.
- Apply inverse fusion transform to extract the permuted secret image as follows:

$$PSI = (X^{'} - \alpha X)/\beta \tag{17}$$

  The extraction procedure is not blind, as we need the cover image to extract the permuted secret message.
- Apply the inverse permutation procedure using the same chaotic system to obtain the secret image.

4. Experimental Results and Analysis

In the experiments, we first create the stego images by using the implemented steganographic methods that were applied on the standard gray level cover images "Lena", "Peppers", "Baboon" in 512×512 pixels and using "Boat" as a secret message with different sizes (embedding rates, ranging from 5% to 40%). The six criteria used to evaluate the qualities of the stego images have been listed as follows: Peak Signal-to-Noise Ratio ($PSNR$) [46], Image Fidelity (IF), structural similarity ($SSIM$), the entropy (E), the redundancy (R), and the image redundancy (IR). They can be represented by the following equations:

$$PSNR = 10 \times \log_{10}\left(\frac{Max \quad p_c^2(i,j)}{\frac{1}{M \times N}\left(\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[p_c(i,j) - p_s(i,j)]^2\right)}\right) \tag{18}$$

$$IF = 1 - \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[p_c(i,j)]^2)}{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[p_c(i,j) - p_s(i,j)]^2\right)} \tag{19}$$

$$SSIM = \frac{(2\mu_c\mu_s - 1)(2cov_{cs} + c_2)}{(\mu_c^2 + \mu_s^2 + c_1)(\sigma_c^2 + \sigma_s^2 + c_2)} \tag{20}$$

In the above equations, $p_c(i,j)$ and $p_s(i,j)$ are the pixel value of the ith row and jth column of the cover and stego image; M and N are the width and height of the considered cover image.

μ_c, μ_s are the average of the cover and stego images; σ_c^2, σ_s^2 are the variance of the cover and stego images; μ_{cs} is the co-variance of the cover-stego; $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ are two variables that are used to stabilize the division with a weak denominator; L is the dynamic range of the pixel values, and k_1, k_2 are two much smaller constants compared to 1. We considered $k_1 = k_2 = 0.05$.

The higher the $PSNR$, IF, and $SSIM$, the better the quality of the stego image. $PSNR$ values falling below 40 dB indicate a fairly low quality. Therefore, a high-quality stego should strive to be above 40 dB.

Additionally, we used three other parameters to estimate the qualities of the stego images. These parameters have been listed as follows:

- The Entropy E, given by the following relation:

$$E = -\sum_{0}^{2^L-1} p(P_i) log_2(p(P_i)) \tag{21}$$

  L is already defined. $p(P_i)$ is the probability of the pixel value P_i.
- The Redundancy R is usually represented by the following formula:

$$R = \frac{E_{max} - E}{E} \tag{22}$$

  Here, $E_{max} = 8$. However, this relationship is problematic because the value of the minimal entropy is not known. For that, Tasnime [47] proposed using the following relationship, which seems to be more precise:

$$IR = \frac{\sum_{i=1}^{L} \left|R_i - R_{opt}\right|}{R_{opt}(2^L - 1) + (S - R_{opt})} \tag{23}$$

  Called Image Redundancy (IR) with:

  - S being the size of the image under test;
  - R_i being the number of occurrences of each pixel value;
  - R_{opt} being the optimal number of occurrences that each pixel value should have to get a non-redundant image.

In the following section, we present and compare the performance of the three implemented steganographic methods.

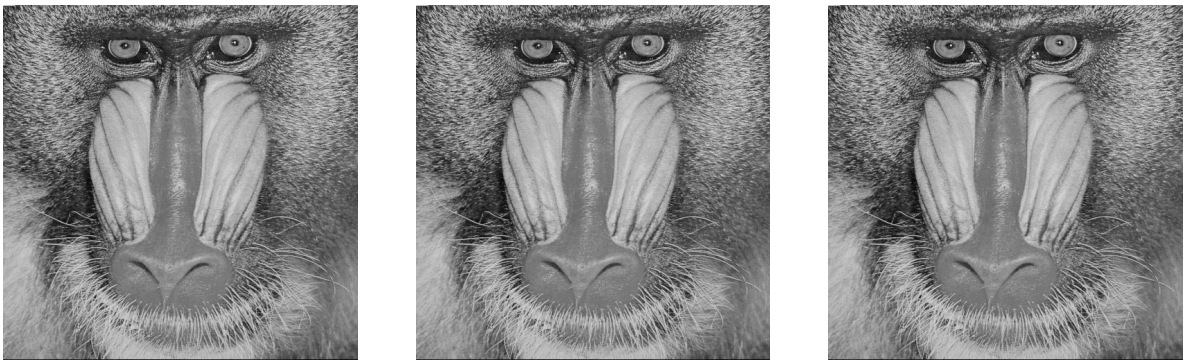
4.1. Enhanced EALSBMR

The results obtained from the parameters $PSNR$, IF, and $SSIM$ for the algorithm have been presented in Table 1; their values indicate the high quality of the stego images, even with a high embedding rate of 40%. We observe that the $PSNR$, IF, and $SSIM$ values decrease, as expected, when the size of the secret message increases.

Table 1. $PSNR$, IF, and $SSIM$ values for the EEALSBMR method.

Embedding Rate	Cover Image	*PSNR*	*IF*	*SSIM*
5%	Baboon	68.3810	0.9999	0.9999
	Lena	68.1847	0.9999	0.9999
	Peppers	67.7160	0.9999	0.9999
10%	Baboon	65.5986	0.9999	0.9999
	Lena	65.2821	0.9999	0.9999
	Peppers	64.7763	0.9999	0.9999
20%	Baboon	62.3551	0.9999	0.9999
	Lena	62.3559	0.9999	0.9996
	Peppers	61.7066	0.9999	0.9995
30%	Baboon	60.6902	0.9998	0.9999
	Lena	60.5630	0.9998	0.9990
	Peppers	59.9585	0.9998	0.9992
40%	Baboon	59.4245	0.9997	0.9999
	Lena	59.2608	0.9997	0.9985
	Peppers	58.6662	0.9997	0.9988

In Figure 9a–c, we show the "Baboon" cover image and the corresponding stego images for 5% and 40% embedding rates, respectively. The visual quality obtained from the "Baboon" stego images is very high because visually, it is impossible to discriminate between the cover and stego images.



(**a**) (**b**) (**c**)

Figure 9. (**a**) Cover image, (**b**) Stego image with embedding rate of 5%, (**c**) Stego image with embedding rate of 40%.

Just to fix the ideas, using the Lina image as the cover, and to obtain approximately identical capacity, we globally compared the obtained $PSNR$ of the EEALSBMP method with that obtained by the following methods: [4–6,17]. We observed that only the method proposed by Borislav et al. [17] produces a better $PSNR$ than the EEALSBMP method. However, this method cannot be adapted.

4.2. Enhanced DCT Steganographic Method

The results obtained from this method, as presented in Table 2, indicate the high quality of the stego images, even with a high embedding rate. Additionally, even the visual quality obtained is very high, as shown in Figure 10.

Table 2. *PSNR*, *IF*, and *SSIM* values for the EDCT method.

Embedding Rate	Cover Image	*PSNR*	*IF*	*SSIM*
5%	Baboon	71.2372	0.9999	0.9999
	Lena	71.1769	0.9999	0.9999
	Peppers	70.4866	0.9999	0.9999
10%	Baboon	64.8846	0.9999	0.9999
	Lena	64.9487	0.9999	0.9998
	Peppers	64.1426	0.9999	0.9998
20%	Baboon	59.6895	0.9997	0.9999
	Lena	59.6225	0.9997	0.9992
	Peppers	58.9535	0.9997	0.9993
30%	Baboon	57.4212	0.9995	0.9998
	Lena	57.3421	0.9995	0.9989
	Peppers	56.7406	0.9995	0.9988
40%	Baboon	56.3421	0.9994	0.9997
	Lena	56.2265	0.79994	0.9987
	Peppers	55.4876	0.9994	0.9985

(**a**) (**b**) (**c**)

Figure 10. (**a**) Cover image, (**b**) Stego image with embedding rate of 5%, (**c**) Stego image with embedding rate of 40%.

4.3. Enhanced DWT Steganographic Method

Table 3 presents the results obtained from the EDWT algorithm, which indicate that the steganographic algorithm exhibits good performance. Furthermore, no visual trace can be found in the resulting stego images, as shown in Figure 11a–c.

Table 3. *PSNR*, *IF*, and *SSIM* values for EDWT method.

Embedding Rate	Cover Image	*PSNR*	*IF*	*SSIM*
5%	Baboon	59.1876	0.9999	0.9999
	Lena	58.7673	0.9997	0.9999
	Peppers	58.1699	0.9997	0.9999
10%	Baboon	56.2224	0.9997	0.9999
	Lena	55.8085	0.9994	0.9999
	Peppers	55.2086	0.9993	0.9999
20%	Baboon	53.3463	0.9988	0.9999
	Lena	52.8205	0.9988	0.9999
	Peppers	52.2269	0.9987	0.9999
30%	Baboon	52.0465	0.9984	0.9999
	Lena	51.6471	0.9984	0.9999
	Peppers	51.0509	0.9983	0.9999
40%	Baboon	51.3450	0.9982	0.9999
	Lena	50.9536	0.9981	0.9999
	Peppers	50.3417	0.9980	0.9999

(**a**) (**b**) (**c**)

Figure 11. (**a**) Cover image, (**b**) Stego image with embedding rate of 5%, (**c**) Stego image with embedding rate of 40%.

4.4. Performance Comparison of the Three Steganographic Methods

Tables 1–3 of *PSNR*, *IF*, and *SSIM* of the three methods show that the EEALSBMR and EDCT methods, in comparison with the EDWT method, ensure better quality of the stego images at different embedding rates. There is approximately a 10-dB difference in *PSNR*s at a 5% embedding rate and a 5 to 8 dB difference in *PSNR*s at a 40% embedding rate.

4.5. Performance Using Parameters E, R and IR

The results obtained from parameters *E*, *R*, and *IR* for the three algorithms on the stego images with different embedding rates have been presented in Tables 4–6. As we can see, these values, given in Table 7, are too close to the values obtained over the original images. This is consistent with the previous results obtained from the parameters *PSNR*, *IF*, and *SSIM* regarding the high quality of the stego images.

Table 4. E, R, and IR for the EEALSBMR method.

Embedding Rate	Cover Image	E	R	IR
5%	Baboon	7.3586	0.0802	0.3805
	Lena	7.4455	0.0693	0.3261
	Peppers	7.5715	0.0536	0.2975
10%	Baboon	7.3586	0.0802	0.3805
	Lena	7.4456	0.0693	0.3261
	Peppers	7.5715	0.0535	0.2976
20%	Baboon	7.3585	0.0802	0.3805
	Lena	7.4457	0.0693	0.3261
	Peppers	7.5717	0.0535	0.2977
30%	Baboon	7.3584	0.0802	0.3805
	Lena	7.4457	0.0693	0.3261
	Peppers	7.5718	0.0535	0.2975
40%	Baboon	7.3578	0.0803	0.3806
	Lena	7.4454	0,0693	0.3260
	Peppers	7.5722	0.0535	0.2973

Table 5. E, R, and IR values for the EDCT method.

Embedding Rate	Cover Image	E	R	IR
5%	Baboon	7.3585	0.0802	0.3804
	Lena	7.4456	0.0693	0.3261
	Peppers	7.5716	0.0536	0.2976
10%	Baboon	7.3585	0.0802	0.3805
	Lena	7.4456	0.0693	0.3262
	Peppers	7.5717	0.0535	0.2976
20%	Baboon	7.3585	0.0802	0.3804
	Lena	7.4457	0.0693	0.3263
	Peppers	7.5725	0.0534	0.2973
30%	Baboon	7.3584	0.0802	0.3802
	Lena	7.4459	0.0693	0.3261
	Peppers	7.5730	0.0534	0.2969
40%	Baboon	7.3578	0.0803	0.3806
	Lena	7.4462	0,0692	0.3257
	Peppers	7.5734	0.0533	0.2973

Table 6. E, R, and IR values for EDWT method.

Embedding Rate	Cover Image	E	R	IR
5%	Baboon	7.3581	0.0802	0.3805
	Lena	7.4455	0.0693	0.3261
	Peppers	7.5715	0.0536	0.2975
10%	Baboon	7.3580	0.0802	0.3806
	Lena	7.4456	0.0693	0.3261
	Peppers	7.5717	0.0535	0.2974
20%	Baboon	7.3580	0.0802	0.3806
	Lena	7.4456	0.0693	0.3261
	Peppers	7.5718	0.0535	0.2975
30%	Baboon	7.3580	0.0802	0.3805
	Lena	7.4456	0.0693	0.3261
	Peppers	7.5718	0.0535	0.2974
40%	Baboon	7.3580	0.0803	0.3806
	Lena	7.4457	0,0693	0.3261
	Peppers	7.5721	0.0533	0.2973

Table 7. E, R, and IR values for the cover images.

Cover Image	E	R	IR
Baboon	7.3585	0.0802	0.3805
Lena	7.4455	0.0693	0.3261
Peppers	7.5715	0.0536	0.2976

5. Universal Steganalysis

A good steganographic method should be imperceptible not only to human vision systems but also to computer analysis. Steganalysis is the art and science that detects whether a given image has a message hidden in it [1,48]. The extensive range of natural images and the wide range of data embedding algorithms make steganalysis a difficult task. In this work, we consider universal steganalysis to be based on statistical analysis.

Universal (blind) steganalysis attempts to detect hidden information without any knowledge about the steganographic algorithm. The idea is to extract the features of cover images and the features of stego images and then use them as the feature vectors that are used by a supervised classifier (SVM, FLD, neural networks…) to distinguish whether the image under test is a stego image. This procedure is illustrated in Figure 12. The left side of the flowchart displays the different steps of the learning process while the right side illustrates the different steps of the testing process.

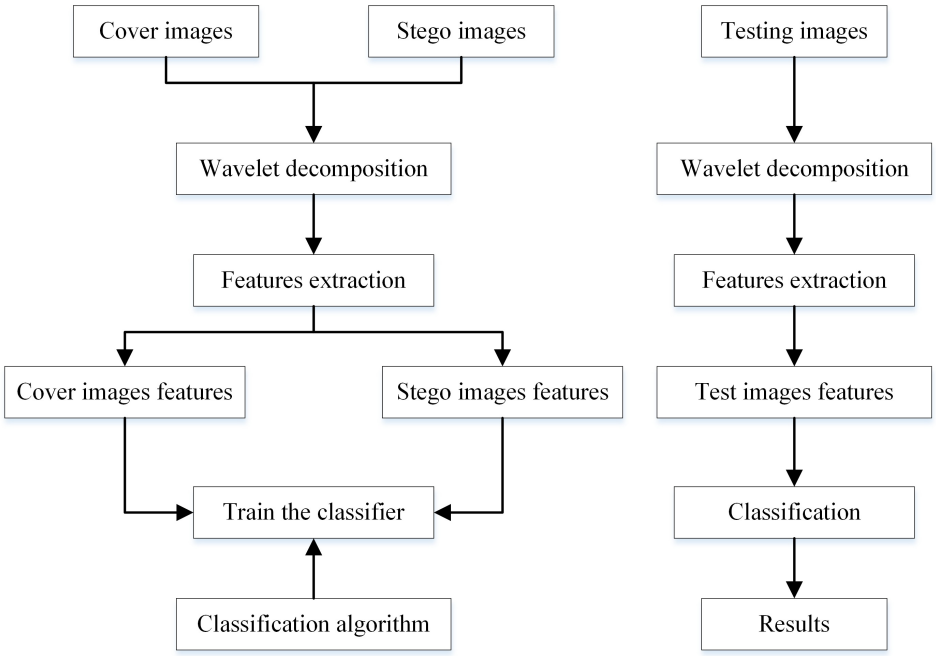


Figure 12. Flowchart of the blind steganalysis process.

5.1. Multi-Resolution Wavelet Decomposition

The DWT, which uses a sub-bands coding algorithm, is found to quickly compute the Wavelet Transform. Furthermore, it is easy to implement and reduces the computation time and the number of resources required. The DWT analyses the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and into detailed information. The decomposition of the signal into different frequencies is achieved by applying separable low-pass $\hat{g}(n)$ and high-pass $\hat{h}(n)$ filters along the image axes. The DWT computes the approximation coefficients matrix A and details coefficients matrices H, V, and D (horizontal, vertical, and diagonal, respectively) of the input matrix X, as illustrated in Figure 13.

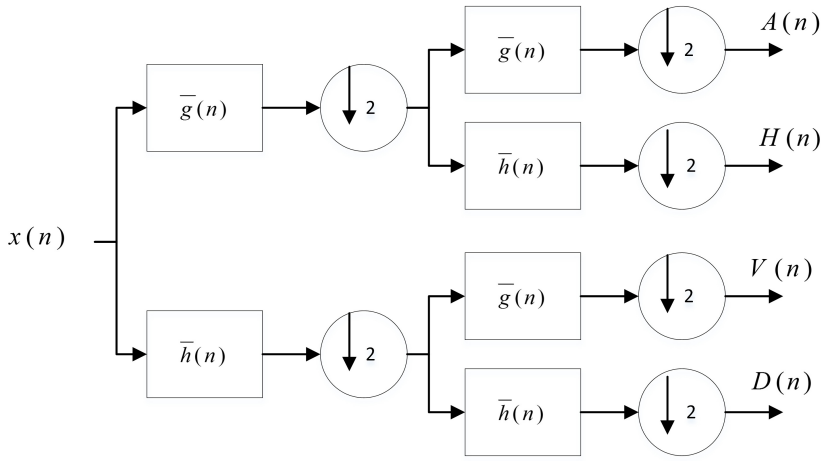


Figure 13. Multi-resolution wavelet decomposition.

5.2. Feature Vector Extraction

As the amount of image data is enormous, it is not feasible to directly use the complete image data for analysis. Therefore, for steganalysis, it is useful to extract a certain amount of useful data features that represent the image instead of the image itself. The addition of a message to a cover image may not affect the visual appearance of the image, but it will affect some statistics. The features required for steganalysis should be able to detect these minor statistical disorders that are created during the data-hiding process.

Three feature-extraction techniques are used in this paper to detect the presence of a secret message; these methods calculate the statistical properties of the images by employing multi-resolution wavelet decomposition.

5.2.1. Method 1: Feature Vectors Extracted from the Empirical Moments of the PDF-Based Multi-Resolution Coefficients and Their Prediction Error

The multi-resolution wavelet decomposition employed here is based on separable quadrature mirror filters (QMFs). This decomposition splits the frequency space into multiple scales and orientations. This is accomplished by applying separable low-pass and high-pass filters along the image axes, generating a vertical, horizontal, diagonal, and low-pass sub-band. The horizontal, vertical, and diagonal sub-bands at scale m = 1, 2, ..., n are denoted as H_m , V_m and D_m.

In our work, the first set of features is extracted from the statistics over coefficients S_m (x,y) of each sub-band and for levels (scales) m = 1 and n = 3. These characteristics represent the following: mean μ, variance σ^2, skewness ξ, and kurtosis κ. They can be represented as follows:

$$\begin{aligned}
\mu &= \frac{1}{N_x N_y} \sum_{x,y} S_m(x,y) \\
\sigma^2 &= \frac{1}{N_x N_y} \sum_{x,y} (S_m(x,y) - \mu)^2 \\
\xi &= \frac{1}{N_x N_y \sigma^3} \sum_{x,y} (S_m(x,y) - \mu)^3 \\
\kappa &= \frac{1}{N_x N_y \sigma^4} \sum_{x,y} (S_m(x,y) - \mu)^4 - 3
\end{aligned} \tag{24}$$

From Equation (24), we can build the first feature vector Z_s of $N_m \times N_{bd} \times n = 4 \times 3 \times 3 = 36$ elements, where N_m, N_{bd}, and n are the number of moments, sub-bands, and scales. The feature vector Z_s is represented as follows:

$$Z_s = [Z_1, Z_2, Z_3]$$

where:

$$Z_1 = [\mu_{H_1}, \mu_{V_1}, \mu_{D_1} | \sigma_{H_1}, \sigma_{V_1}, \sigma_{D_1} | \xi_{H_1}, \xi_{V_1}, \xi_{D_1} | \kappa_{H_1}, \kappa_{V_1}, \kappa_{D_1}]$$

$$Z_2 = [\mu_{H_2}, \mu_{V_2}, \mu_{D_2} | \sigma_{H_2}, \sigma_{V_2}, \sigma_{D_2} | \xi_{H_2}, \xi_{V_2}, \xi_{D_2} | \kappa_{H_2}, \kappa_{V_2}, \kappa_{D_2}]$$

$$Z_3 = [\mu_{H_3}, \mu_{V_3}, \mu_{D_3} | \sigma_{H_3}, \sigma_{V_3}, \sigma_{D_3} | \xi_{H_3}, \xi_{V_3}, \xi_{D_3} | \kappa_{H_3}, \kappa_{V_3}, \kappa_{D_3}]$$

The second set of statistics is based on the prediction errors of coefficients $S_m(x,y)$ of an optimal linear predictor. The sub-band coefficients are correlated with their spatial, orientation, and scale neighbors. Several prediction techniques of coefficients $S^p_{H_m}(x,y)$, $S^p_{V_m}(x,y)$, and $S^p_{D_m}(x,y)$ ($m = 1, 2, 3$) may be used. In this work, we used a linear predictor, specifically the one proposed by Farid in [30], as shown below:

$$\begin{aligned} S^p_{H_m}(x,y) = {} & w_1 S_{H_m}(x-1,y) + w_2 S_{H_m}(x+1,y) + w_3 S_{H_m}(x,y-1) + \\ & w_4 S_{H_m}(x,y+1) + w_5 S_{H_{m+1}}(\frac{x}{2},\frac{y}{2}) + w_6 S_{D_m}(x,y) + \\ & w_7 S_{D_{m+1}}(\frac{x}{2},\frac{y}{2}) \end{aligned} \tag{25}$$

$$\begin{aligned} S^p_{V_m}(x,y) = {} & w_1 S_{V_m}(x-1,y) + w_2 S_{V_m}(x+1,y) + w_3 S_{V_m}(x,y-1) + \\ & w_4 S_{V_m}(x,y+1) + w_5 S_{V_{m+1}}(\frac{x}{2},\frac{y}{2}) + w_6 S_{D_m}(x,y) + \\ & w_7 S_{D_{m+1}}(\frac{x}{2},\frac{y}{2}) \end{aligned} \tag{26}$$

$$\begin{aligned} S^p_{D_m}(x,y) = {} & w_1 S_{D_m}(x-1,y) + w_2 S_{D_m}(x+1,y) + w_3 S_{D_m}(x,y-1) + \\ & w_4 S_{D_m}(x,y+1) + w_5 S_{D_{m+1}}(\frac{x}{2},\frac{y}{2}) + w_6 S_{H_m}(x,y) + \\ & w_7 S_{V_{m+1}}(\frac{x}{2},\frac{y}{2}) \end{aligned} \tag{27}$$

For more clarity, in Figure 14, we provide the block diagram for the prediction of coefficient $S^p_{V_1}(x,y)$.

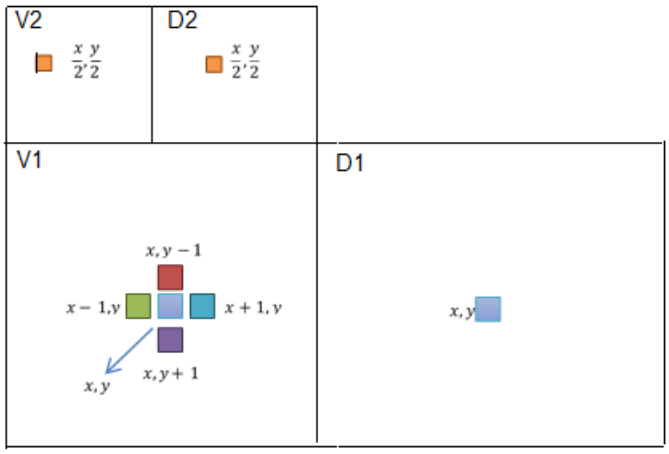


Figure 14. Block diagram for the prediction of coefficient $S^p_{V_1}(x,y)$.

The parameters w_i (scalar weighting values) of the error prediction coefficients of each sub-band for a given level m are adjusted to minimize the prediction error by minimizing the quadratic error function, as shown below:

$$E(w) = [S_m - Qw]^2 \tag{28}$$

The columns of the matrix Q contain the neighboring coefficient magnitudes, as specified in Equations (25)–(27). The quadratic error function is minimized analytically as follows:

$$\frac{\mathrm{d}E(w)}{\mathrm{d}w} = 2Q^T(S_m - Qw) = 0 \tag{29}$$

Then, we obtain:

$$w_{opt} = (Q^tQ)^{-1}Q^tS_m \tag{30}$$

For the optimal predictor, we use the log error given by the following equation to predict error coefficients of each sub-band for a given level m:

$$\epsilon_m^p = \log_2 S_m - \log_2(|Qw_{opt}|) \tag{31}$$

By using Equation (31), additional statistics are collected, namely the mean, variance, skewness, and kurtosis (see Equation (24)). The feature vector Z_ϵ^p is similar to Z_s; it is represented as follows:

$$Z_\epsilon^p = [Z_{1\epsilon}^p, Z_{2\epsilon}^p, Z_{3\epsilon}^p]$$

where:

$$Z_{1\epsilon}^p = [\mu_{\epsilon_{H_1}}^p, \mu_{\epsilon_{V_1}}^p, \mu_{\epsilon_{D_1}}^p | \sigma_{\epsilon_{H_1}}^p, \sigma_{\epsilon_{V_1}}^p, \sigma_{\epsilon_{D_1}}^p | \xi_{\epsilon_{H_1}}^p, \xi_{\epsilon_{V_1}}^p, \xi_{\epsilon_{D_1}}^p | \kappa_{\epsilon_{H_1}}^p, \kappa_{\epsilon_{V_1}}^p, \kappa_{\epsilon_{D_1}}^p]$$

$$Z_{2\epsilon}^p = [\mu_{\epsilon_{H_2}}^p, \mu_{\epsilon_{V_2}}^p, \mu_{\epsilon_{D_2}}^p | \sigma_{\epsilon_{H_2}}^p, \sigma_{\epsilon_{V_2}}^p, \sigma_{\epsilon_{D_2}}^p | \xi_{\epsilon_{H_2}}^p, \xi_{\epsilon_{V_2}}^p, \xi_{\epsilon_{D_2}}^p | \kappa_{\epsilon_{H_2}}^p, \kappa_{\epsilon_{V_2}}^p, \kappa_{\epsilon_{D_2}}^p]$$

$$Z_{3\epsilon}^p = [\mu_{\epsilon_{H_3}}^p, \mu_{\epsilon_{V_3}}^p, \mu_{\epsilon_{D_3}}^p | \sigma_{\epsilon_{H_3}}^p, \sigma_{\epsilon_{V_3}}^p, \sigma_{\epsilon_{D_3}}^p | \xi_{\epsilon_{H_3}}^p, \xi_{\epsilon_{V_3}}^p, \xi_{\epsilon_{D_3}}^p | \kappa_{\epsilon_{H_3}}^p, \kappa_{\epsilon_{V_3}}^p, \kappa_{\epsilon_{D_3}}^p]$$

Finally, the feature vector that will be used for the learning classifier is represented by $Z = [Z_s | Z_\epsilon^p]$. It contains 72 components.

5.2.2. Method 2: Feature Vectors Extracted from Empirical Moments of CF-Based Multi-Resolution

The first set of feature vectors Z_s is extracted based on the CF and the wavelet decomposition, as proposed by Shi et al. [31]. The statistical moments of the characteristic function $\phi(k)$ of order $n =$ 1 to 3 are represented for each sub-band (A_m, H_m, V_m, D_m) at different levels m = 1, 2, and 3 of the wavelet decomposition as follows:

$$M_{S_m}^n = \frac{\sum_{k=1}^{\frac{N}{2}} |\phi(k)| \times k^n}{\sum_{k=1}^{\frac{N}{2}} |\phi(k)|} \tag{32}$$

$$\phi(k) = \sum_{i=1}^{N} h(i) \exp\left\{\frac{j2\pi ik}{K}\right\} \quad 1 \le k \le K \tag{33}$$

is a component of the characteristic function at frequency k, calculated from the histogram of the sub-band S_m, and N is the total number of points of the histogram. Equation (32) allows us to build the first feature vector Z_m of size $12 \times 3 = 36$ components and 3 moments of the initial image. The feature vectors Z_m have been listed as follows:

$$\begin{aligned} Z_s = \; & [M_I^1, M_I^2, M_I^3 | M_{A_1}^1, M_{A_1}^2, M_{A_1}^3 | M_{H_1}^1, M_{H_1}^2, M_{H_1}^3 | M_{V_1}^1, M_{V_1}^2, M_{V_1}^3 \\ & | M_{D_1}^1, M_{D_1}^2, M_{D_1}^3 | M_{A_2}^1, M_{A_2}^2, M_{A_2}^3 | M_{H_2}^1, M_{H_2}^2, M_{H_2}^3 | M_{V_2}^1, M_{V_2}^2, M_{V_2}^3 \\ & | M_{D_2}^1, M_{D_2}^2, M_{D_2}^3 | M_{A_3}^1, M_{A_3}^2, M_{A_3}^3 | M_{H_3}^1, M_{H_3}^2, M_{H_3}^3 | M_{V_3}^1, M_{V_3}^2, M_{V_3}^3 \\ & | M_{D_3}^1, M_{D_3}^2, M_{D_3}^3] \end{aligned}$$

In the above equation, M_I^1, M_I^2, M_I^3 are the moments of the initial image.

The second category of features is calculated from the moments of prediction-error image and its wavelet decomposition.

Prediction-error image:

In steganalysis, we only care about the distortion caused by data-hiding. This type of distortion may be rather weak and, hence, covered by other types of noises, including those caused due to the peculiar feature of the image itself. To make the steganalysis more effective, it is necessary to keep the noise of the dissimulation and eliminate most of the other noises. For this purpose, we calculate the moments of characteristic functions of order $n = 1, 3$ of the predicted error image and of its wavelet decomposition at the various levels $m = 1, 2$, and 3 (see Equation (32)). The prediction-error image is obtained by subtracting the predicted image (in which each predicted pixel grayscale value in the cover image uses its neighboring pixels' grayscale values (see Equation (34))) from the cover image. Such features make the steganalysis more efficient because the hidden data is usually unrelated to the cover media. The prediction pixel is expressed as follows:

$$\hat{x} = \begin{cases} max(a,b) & c \leq min(a,b) \\ min(a,b) & c \geq max(a,b) \\ a+b-c & otherwise \end{cases} \tag{34}$$

In the above equation, a, b, c are the context of the pixel x under consideration; $\hat{x}$ is the prediction value of x. The location of a, b, c can be illustrated as in Figure 15.

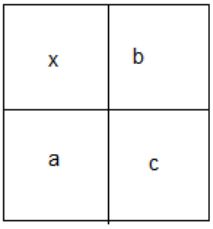


Figure 15. Prediction context of a pixel x.

The feature vector Z_ϵ^p is represented as follows:

$$\begin{aligned} Z_\epsilon^p = \; & [M_{\epsilon_1}^{p_1}, M_{\epsilon_1}^{p_2}, M_{\epsilon_1}^{p_3} | M_{A_1}^1, M_{A_1}^2, M_{A_1}^3 | M_{H_1}^1, M_{H_1}^2, M_{H_1}^3 | M_{V_1}^1, M_{V_1}^2, M_{V_1}^3 \\ & | M_{D_1}^1, M_{D_1}^2, M_{D_1}^3 | M_{A_2}^1, M_{A_2}^2, M_{A_2}^3 | M_{H_2}^1, M_{H_2}^2, M_{H_2}^3 | M_{V_2}^1, M_{V_2}^2, M_{V_2}^3 \\ & | M_{D_2}^1, M_{D_2}^2, M_{D_2}^3 | M_{A_3}^1, M_{A_3}^2, M_{A_3}^3 | M_{H_3}^1, M_{H_3}^2, M_{H_3}^3 | M_{V_3}^1, M_{V_3}^2, M_{V_3}^3 \\ & | M_{D_3}^1, M_{D_3}^2, M_{D_3}^3] \end{aligned}$$

In the above equation, $M_{A_1}^1, M_{A_1}^2, M_{A_1}^3$ are the 1st, 2nd , and 3rd order moments of the corresponding CFs, from the sub-band A_1 of the 1^{st} level decomposition on the error image.

Finally, the feature vector that will be used for learning classification is $Z = [Z_s | Z_\epsilon^p]$, containing 78 components.

5.2.3. Method 3: Feature Vector Extracted from Empirical Moments Based on the FC and the PDF of Image Prediction Error and Its Different Sub-Bands of the Multi-Resolution Decomposition

The first characteristic vector Z_s combines two types of normalized moments: moments based on the function density of probability and moments based on the characteristic function of various sub-bands of the multi-resolution decomposition at three levels of the gray image. We use the expression of Wang and Moulin [32] to calculate the moments of order $n = 1$ to 6 of the initial

image and its sub-band (A_m, H_m, V_m, D_m) of the three-level ($m = 1$ to 3) wavelet decomposition, as shown below:

$$M^n_{S_m} = \frac{\sum_{k=1}^{\frac{N}{2}} |\phi(k)| \times \sin^n(\frac{\pi k}{K})}{\sum_{k=1}^{\frac{N}{2}} |\phi(k)|} \tag{35}$$

$$\phi(k) = \sum_{i=1}^{N} h(i) \exp\left\{\frac{j2\pi ik}{K}\right\} \quad 1 \leq k \leq K \tag{36}$$

is a component of the characteristic function at frequency k, estimated from the histogram. Equation (35) already allows having a feature vector of 6 × 1 + 6 × (4 × 3) = 78 components. Also, to improve the performance of the learning system, we calculate the moments of the sub-bands $A_2^{'}$, $H_2^{'}$, $V_2^{'}$, $D_2^{'}$ obtained from the decomposition of the diagonal sub-band D_1. Therefore, the total size of the vector Z_s is 78 + (6 × 4) = 102 components.

$$\begin{aligned} Z_s = \quad & [M^i_I | M^i_{A_1} | M^i_{H_1} | M^i_{V_1} | M^i_{D_1} | M^i_{A_2} | M^i_{H_2} | M^i_{V_2} | M^i_{D_2} | M^i_{A_3} | M^i_{H_3} | M^i_{V_3} | M^i_{D_3} \\ & | M^i_{A_2'} | M^i_{H_2'} | M^i_{V_2'} | M^i_{D_2'}], \quad i = 1, 2, ..., 6 \end{aligned}$$

For example, $M^i_I = [M^1_I, M^2_I, M^3_I, M^4_I, M^5_I, M^6_I]$ are the first six order moments of the original image.

The second category of characteristics consists of the first six moments of the prediction error, which is $\epsilon^p_m = \log_2 S_m - \log_2(|Qw_{opt}|)$ of coefficients of each sub-band for a given level m, as shown below:

$$m^n_{\epsilon^p_m} = \frac{1}{N} \sum_{i=1}^{N} (\epsilon^p_m)^n \quad n = 1, 2...., 6 \tag{37}$$

The vector of the second category is defined by Z^p_ϵ, as shown below:

$$Z^p_\epsilon = [m^i_{\epsilon H_m} | m^i_{\epsilon V_m} | m^i_{\epsilon D_m}]$$

for each

$$m = \{1, 2, 3\}; i = 1, 2, ..., 6$$

The size of Z^p_ϵ is 3 x 6 x 3 = 54 components.

Finally, the feature vector to be used for classification by learning is $Z = [Z_s | Z^p_\epsilon]$. It has 156 components.

5.3. Classification

The last stage of the learning and test process of the universal steganalysis is classification (see Figure 12). Its objective is to group the images into two classes, class of the cover images and class of the stego images, according to their feature vectors. We adopt the Fisher linear discriminator (FLD) and the support vector machine (SVM) for training and testing.

5.3.1. FLD Classifier

Below, we reformulate the FLD classifier for our application and apply it to two classes. Let $\mathbf{Z} = \{Z_1, Z_2, ..., Z_N\}$ be a set of feature vectors, each with nd dimensions. Among these vectors, N_1 vectors are $\mathbf{Z_c}$ feature vectors labeled 1, indicating cover images. N_2 vectors are $\mathbf{Z_s}$ labeled 2, indicating stego images, with $N = N_1 + N_2$. We want to form all projection values $(Z_p) = \{Z_{p_1}, Z_{p_2}, ..., Z_{p_N}\}$ of dimension N through linear combinations of feature vectors $\mathbf{Z}_p$ as follows:

$$\mathbf{Z_p} = \mathbf{W^t Z} \tag{38}$$

In the above equation, $\mathbf{W}$ is an orientation vector of dimension nd.

In our study, the feature vector **Z** is projected into a space of two classes. This projection tends to maximize the distance between the projected class means (M_{cp}, M_{sp}) while minimizing projected class scatters S_{cp}, S_{sp}.

- Learning process
  The learning process involves optimizing the following expression:

$$J(W) = \frac{|M_{cp} - M_{sp}|^2}{S_{cp} + S_{sp}} \tag{39}$$

where:

$$M_{cp} = \frac{1}{N_1} \sum_{Z_p \in Z_{cp}} Z_p = \frac{1}{N_1} \sum_{Z \in Z_c} W^t Z = W^t M_c \tag{40}$$

is the mean feature vector of cover class after projection, and

$$M_c = \frac{1}{N_1} \sum_{Z \in Z_c} Z \tag{41}$$

is the mean feature vector of cover class of dimension nd.
The mean feature vector of stego class after projection is represented as follows:

$$M_{sp} = \frac{1}{N_2} \sum_{Z_p \in Z_{sp}} Z_p = \frac{1}{N_2} \sum_{Z \in Z_s} W^t Z = W^t M_s \tag{42}$$

where:

$$M_s = \frac{1}{N_2} \sum_{Z \in Z_s} Z \tag{43}$$

is the mean feature vector of a stego class of dimension nd.
The scatter matrix of the cover class after projection has been shown as follows:

$$S_{cp} = \sum_{Z_p \in Z_{cp}} (Z_p - M_{cp})^2 = \sum_{Z \in Z_c} (W^t Z - W^t M_c)^2 = \sum_{Z \in Z_c} W^t (Z - M_c)(Z - M_c)^t W = W^t S_c W \tag{44}$$

where:

$$S_c = (Z - M_c)(Z - M_c)^t \tag{45}$$

is the scatter matrix (of dimension $nd \times nd$) of a cover class.
The scatter matrix of the projected samples of a stego class has been shown as follows:

$$S_{sp} = \sum_{Z_p \in Z_{sp}} (Z_p - M_{sp})^2 = \sum_{Z \in Z_s} (W^t Z - W^t M_s)^2 = \sum_{Z \in Z_s} W^t (Z - M_s)(Z - M_s)^t W = W^t S_s W \tag{46}$$

where:

$$S_s = (Z - M_s)(Z - M_s)^t \tag{47}$$

is a scatter matrix (of dimension $nd \times nd$) for the samples in the original feature space of a stego class.
The within-class scatter matrix after projection is defined as follows:

$$S_{cp} + S_{sp} = W^t (S_c + S_s) W = W^t S_w W \tag{48}$$

where:

$$S_w = S_c + S_s \tag{49}$$

The difference between the projected means is expressed as follows:

$$(M_{cp} - M_{sp})^2 = (W^t M_c - W^t M_s)^2 = W^t (M_c - M_s)(M_c - M_s)^t W = W^t S_B W \tag{50}$$

where:

$$S_B = (M_c - M_s)(M_c - M_s)^t \tag{51}$$

We can finally express the Fisher criterion (Equation (39)) in terms of S_B and S_W as follows:

$$J(W) = \frac{W^t S_B W}{W^t S_w W} \tag{52}$$

The solution of Equation (52) is given by [49].

$$W_{opt} = S_w^{-1}(M_c - M_s) \tag{53}$$

- Testing process
  The testing process (classification step) is conducted as follows:
  Let Z be the matrix containing the feature vectors of covers and stegos.
  The projection of Z on the orientation vector W_{opt} gives all projected values Z_p.

$$Z_p(j) = \sum_{i=1}^{9} W_{opt}(i) \times Z(i,j) + bj = 1, 2, ..., N \tag{54}$$

  b is a threshold of discrimination between both classes, and it can be fixed to a value that is halfway between both averages projected on the cover and stego.

$$b = 0.5 \times (M_{cp} + M_{sp}) \tag{55}$$

  with:

$$M_{cp} = W_{opt}^t \times M_c$$
$$M_{sp} = W_{opt}^t \times M_s$$

  In the above equations, W_{opt}^t is the transposed of W_{opt}.
  The result $Z_p(j)$, $j = 1, ..., N$ determines the cover or stego class of every test image. Indeed, if $Z_p(j) \geq 0$, then the image under test is cover; otherwise, it is stego.

5.3.2. SVM Classifier

According to numerous recent studies, the SVM classification method is better than the other data classification algorithms in terms of classification accuracy [50]. SVM performs classification by creating a hyper-plan that separates the data into two categories in the most optimal way.

Let $(Z_i, y_i)_{(1 \leq i \leq N)}$ be a set of training examples, each example $Z_i \in \mathbb{R}^{nd}$, nd being the dimension of the input space; it belongs to a class labeled as $y_i \in \{-1, 1\}$. SVM classification constructs a hyper-plan $W^T Z + b = 0$, which best separates the data through a minimizing process, as shown below:

$$\begin{aligned} &\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\zeta_i \\ subject \quad &to : y_i(wZ_i + b) \geq 1 - \zeta_i \end{aligned} \tag{56}$$

Variables ζ_i are called slack variables, and they measure the error made at point (Z_i, y_i).
Parameter C can be viewed as a way to control overfitting.

$\zeta_i \geq 0$ and $C > 0$ is the trade-off between regularization and constraint violation.

Problems related to quadratic optimization are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist to aid in solving them. Solutions involve constructing a dual problem where a Lagrange multiplier α_i is associated with every constraint in the primary problem, as shown below:

$$\begin{aligned} L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j Z_i^T Z_j \\ subject \quad to : \sum_i \alpha_i y_i = 0 \\ 0 \leq \alpha_i y_i \leq C \end{aligned} \tag{57}$$

α_i or Lagrange multipliers are also known as support values.

The linear classifier presented previously is very limited. In most case, classes not only overlap, but the genuine separation functions are non-linear hyper-surfaces. The motivation for such an extension is that an SVM that can create a non-linear decision hyper-surface will be able to non-linearly classify separable data.

The idea is that the input space can always be mapped on to a higher dimensional feature space where the training set is separable.

The linear classifier relies on the dot product between vectors $K(Z_i, Z_j) = Z_i^T Z_j$. If every data point is mapped on to a high-dimensional space via some transformation $\Phi : Z \rightarrow \varphi(Z)$, the dot product becomes:$K(Z_i, Z_j) = \varphi(Z_i)^T \varphi(Z_j)$. Then in the dual formulation, we maximize the following:

$$\begin{aligned} L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(Z_i, Z_j) \\ subject \quad to : \sum_i \alpha_i y_i = 0 \\ 0 \leq \alpha_i y_i \leq C \end{aligned} \tag{58}$$

Subsequently, the decision function turns into the following:

$$f(x) = sgn(\sum_{i=n}^{m} \alpha_i y_i K(Z_i, Z) + b) \tag{59}$$

It should be noted that the dual formulation only requires access to the kernel function and not the features $\Phi(.)$, allowing one to solve the formulation in very high-dimensional feature spaces efficiently. This is also called the kernel trick.

There are many kernel functions in SVM. Therefore, determining how to select a good kernel function is also a research issue. However, for general purposes, there are some popular kernel functions [50,51], which have been listed as follows:

- Linear Kernel:
$$K(Z_i, Z_j) = Z_i^T Z_j \tag{60}$$
- Polynomial Kernel:
$$K(Z_i, Z_j) = (\gamma Z_i^T Z_j + r)^d \quad \gamma > 0 \tag{61}$$
- RBF Kernel:
$$K(Z_i, Z_j) = \exp(-\gamma \left\| Z_i - Z_j \right\|^2) \quad \gamma > 0 \tag{62}$$
- Sigmoid Kernel:
$$K(Z_i, Z_j) = \tanh(\gamma Z_i^T Z_j + r) \tag{63}$$

Here, γ, r, and d are kernel parameters.

In our work, we used the RBF kernel function.

6. Experimental Results of Steganalysis

In this section, we present some experimental results that were obtained from the studied steganalysis system that was applied to the enhanced steganographic methods in the spatial and frequency domain. For this purpose, the image dataset UCID [52,53] is used, which includes 1338 uncompressed color images, and all the images were converted to grayscale before conducting the experiments.

In our experiments, we first created the stego images using the following steganographic methods: Enhanced EALSBMR (EEALSBMR), Enhanced DCT steganography (EDCT), and Enhanced DWT steganography (EDWT). We used these methods with different embedding rates of 5%, 10%, and 20%. Following this, we extracted the image features using the three feature-extraction techniques described above (Farid, Shi, and Moulin techniques) for both the cover and stego images. Finally, we employed the classifiers FLD and SVM to classify the images as either containing a hidden message or not. The evaluation of the classification (binary classification) and the steganalysis (also indirectly the efficiency of insertion methods) is performed by calculating the following parameters: sensibility, specificity, and precision of the confusion matrix and the Kappa coefficient (see Table 8 and Equation (64))

$$Kappa = \frac{P_0 - P_a}{1 - P_a} \tag{64}$$

with:

$$P_0 = TP + TN; P_a = (TP + FP) \times (TP + FN) + (FN + TN) \times (FP + TN) \tag{65}$$

In the above equation, P_0 is the total agreement probability (related to the accuracy), and P_a is the agreement probability that arises out of chance.

Here is one possible interpretation of Kappa values:

- Poor agreement = Less than 0.20
- Fair agreement = 0.20 to 0.40
- Moderate agreement = 0.40 to 0.60
- Good agreement = 0.60 to 0.80
- Very good agreement = 0.80 to 1.00

Table 8. Confusion matrix.

		H0: Stego Image	**H1: Cover Image**	
Test outcome	Test outcome positive	True Positive TP	False Positive FP	Positive predictive value (PPV), or Precision $Pr = \frac{TP}{TP+FP}$
	Test outcome negative	False Negative FN	True Negative TN	Negative predictive value (NPV) $NPV = \frac{TN}{TN+FN}$
		True positive rate (TPR), or, Sensitivity (Se), $Se = \frac{TP}{TP+FN}$	True negative rate (TNR), or Specificity(Sp), $Sp = \frac{TN}{TN+FP}$	Accuracy (Ac), $Ac = \frac{TP+TN}{TP+FN+FP+TN}$

6.1. Classification Results Applied to the Steganographic Method EEALSBMR

In Tables 9–14, we present the classification results (steganalysis) based on the classifiers FLD and SVM and the features of Farid, Shi, and Moulin for the EEALSBMR insertion method with different

insertion rates of 5%, 10%, and 20%. The results show that steganalysis is not effective for all insertion rates. Indeed, the values Se, Sp, and Pr vary around 50%, so these values are not informative values and do not give any idea about the nature of the data. The value of the Kappa coefficient (lower than 0.2) confirms these results. The EEALSBMR steganographic method is robust against statistical steganalysis techniques.

Table 9. FLD classification evaluation of EEALSBMR algorithm using Farid features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2744	0.2714	Pr = 0.5027
H1	0.2256	0.2286	NPV = 0.5033
	Se = 0.5487	Sp = 0.4572	Ex = 0.5030
		$Kappa$ = 0.0060	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2690	0.2645	Pr = 0.5042
H1	0.2310	0.2355	NPV = 0.5048
	Se = 0.5380	Sp = 0.4710	Ex = 0.5045
		$Kappa$ = 0.0090	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2745	0.2459	Pr = 0.5275
H1	0.2255	0.2541	NPV = 0.5298
	Se = 0.5490	Sp = 0.5082	Ex = 0.5286
		$Kappa$ = 0.0572	

Table 10. FLD classification evaluation of EEALSBMR algorithm using Shi features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2612	0.2405	Pr = 0.5207
H1	0.2387	0.2595	NPV = 0.5208
	Se = 0.5225	Sp = 0.5190	Ex = 0.5208
		$Kappa$ = 0.0415	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2504	0.2448	Pr = 0.5057
H1	0.2496	0.2552	NPV = 0.5056
	Se = 0.5008	Sp = 0.5105	Ex = 0.5056
		$Kappa$ = 0.0112	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3191	0.1946	Pr = 0.6212
H1	0.1809	0.3054	NPV = 0.6280
	Se = 0.6382	Sp = 0.6108	Ex = 0.6245
		$Kappa$ = 0.2490	

Table 11. FLD classification evaluation of EEALSBMR algorithm using Moulin features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2489	0.2476	$Pr = 0.5013$
H1	0.2511	0.2524	$NPV = 0.5012$
	$Se = 0.4977$	$Sp = 0.5048$	$Ex = 0.5012$
		$Kappa = 0.0025$	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2559	0.2299	$Pr = 0.5268$
H1	0.2441	0.2701	$NPV = 0.5253$
	$Se = 0.5117$	$Sp = 0.5403$	$Ex = 0.5260$
		$Kappa = 0.0520$	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2990	0.1985	$Pr = 0.6010$
H1	0.2010	0.3015	$NPV = 0.6000$
	$Se = 0.5980$	$Sp = 0.6030$	$Ex = 0.6005$
		$Kappa = 0.2010$	

Table 12. SVM classification evaluation of EEALSBMR algorithm using Farid features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3438	0.3431	$Pr = 0.5005$
H1	0.1562	0.1569	$NPV = 0.5011$
	$Se = 0.6876$	$Sp = 0.3137$	$Ac = 0.6870$
		$Kappa = 0.0013$	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4006	0.3977	$Pr = 0.5018$
H1	0.0994	0.1023	$NPV = 0.5071$
	$Se = 0.8011$	$Sp = 0.2046$	$Ac = 0.5029$
		$Kappa = 0.0057$	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3251	0.3199	$Pr = 0.5041$
H1	0.1749	0.1801	$NPV = 0.5074$
	$Se = 0.6503$	$Sp = 0.3602$	$Ac = 0.5052$
		$Kappa = 0.0105$	

Table 13. SVM classification evaluation of EEALSBMR algorithm using Shi features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2220	0.2188	$Pr = 0.5037$
H1	0.2780	0.2812	$NPV = 0.5029$
	$Se = 0.4440$	$Sp = 0.5625$	$Ac = 0.5032$
		$Kappa = 0.0065$	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2189	0.2161	$Pr = 0.5032$
H1	0.2811	0.2839	$NPV = 0.5024$
	$Se = 0.4377$	$Sp = 0.5678$	$Ac = 0.5028$
		$Kappa = 0.0055$	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2282	0.1999	$Pr = 0.5330$
H1	0.2718	0.3001	$NPV = 0.5247$
	$Se = 0.4564$	$Sp = 0.6002$	$Ac = 0.5283$
		$Kappa = 0.0566$	

Table 14. SVM classification evaluation of EEALSBMR algorithm using Moulin features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2275	0.2264	Pr = 0.5013
H1	0.2725	0.2736	NPV = 0.5010
	Se = 0.4550	Sp = 0.5472	Ac = 0.5011
		$Kappa$ = 0.0023	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2412	0.2380	Pr = 0.5034
H1	0.2588	0.2620	NPV = 0.5031
	Se = 0.4825	Sp = 0.5240	Ac = 0.5032
		$Kappa$ = 0.0065	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2922	0.2684	Pr = 0.5212
H1	0.2078	0.2316	NPV = 0.5271
	Se = 0.5844	Sp = 0.4632	Ac = 0.5238
		$Kappa$ = 0.0476	

6.2. Classification Results Applied to the Steganographic Method EDCT

The classification results (steganalysis) provided in Tables 15–20 for the EDCT insertion method show that with the FLD classifier, when the insertion rate is equal to or higher than 20%, steganalysis is very effective with Shi features and Moulin features, but it is less effective with Farid features. With the SVM classifier, except in the case of Shi features, when an insertion rate of 20% is applied, the results obtained are quite similar to those obtained from the EEALSBMR algorithm and, therefore, steganalysis is not effective. It should be noted that the FLD classifier is more effective for a feature vector of a high dimension than the SVM classifier.

Table 15. FLD classification evaluation of EDCT algorithm using Farid features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2524	0.2454	Pr = 0.5070
H1	0.2476	0.2546	NPV = 0.5069
	Se = 0.5048	Sp = 0.5091	Ac = 0.5070
		$Kappa$ = 0.0139	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2617	0.2238	Pr = 0.5390
H1	0.2383	0.2762	NPV = 0.5368
	Se = 0.5234	Sp = 0.5524	Ac = 0.5379
		$Kappa$ = 0.0758	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3104	0.1719	Pr = 0.6436
H1	0.1896	0.3281	NPV = 0.6337
	Se = 0.6208	Sp = 0.6562	Ac = 0.6385
		$Kappa$ = 0.2770	

Table 16. FLD classification evaluation of EDCT algorithm using Shi features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2548	0.2343	Pr = 0.5209
H1	0.2452	0.2657	NPV = 0.5200
	Se = 0.5095	Sp = 0.5314	Ac = 0.5205
		$Kappa$ = 0.0410	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3242	0.1893	Pr = 0.6313
H1	0.1758	0.3107	NPV = 0.6386
	Se = 0.6484	Sp = 0.6213	Ac = 0.6349
		$Kappa$ = 0.2697	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4409	0.0635	Pr = 0.8741
H1	0.0591	0.4365	NPV = 0.8807
	Se = 0.8817	Sp = 0.8730	Ac = 0.8773
		$Kappa$ = 0.7547	

Table 17. FLD classification evaluation of EDCT algorithm using Moulin features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2611	0.2499	Pr = 0.5110
H1	0.2389	0.2501	NPV = 0.5115
	Se = 0.5223	Sp = 0.5002	Ac = 0.5112
		$Kappa$ = 0.0225	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.2780	0.2136	Pr = 0.5655
H1	0.2220	0.2864	NPV = 0.5633
	Se = 0.5560	Sp = 0.5728	Ac = 0.5644
		$Kappa$ = 0.1288	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3739	0.1243	Pr = 0.7505
H1	0.1261	0.3757	NPV = 0.7487
	Se = 0.7478	Sp = 0.7514	Ac = 0.7496
		$Kappa$ = 0.4992	

Table 18. SVM classification evaluation of EDCT algorithm using Farid features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.0653	0.0591	Pr = 0.5249
H1	0.4347	0.4409	NPV = 0.5035
	Se = 0.1307	Sp = 0.8817	Ac = 0.5062
		$Kappa$ = 0.0124	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.0848	0.0644	Pr = 0.5683
H1	0.4152	0.4356	NPV = 0.5120
	Se = 0.1695	Sp = 0.8712	Ac = 0.5204
		$Kappa$ = 0.0408	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.1734	0.0843	Pr = 0.6729
H1	0.3266	0.4157	NPV = 0.5600
	Se = 0.3469	Sp = 0.8314	Ac = 0.5891
		$Kappa$ = 0.1783	

Table 19. SVM classification evaluation of EDCT algorithm using Shi features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3156	0.3138	Pr = 0.5014
H1	0.1844	0.1862	NPV = 0.5024
	Se = 0.6312	Sp = 0.3724	Ac = 0.5018
		$Kappa$ = 0.0036	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3572	0.3266	Pr = 0.5224
H1	0.1428	0.1734	NPV = 0.5485
	Se = 0.7145	Sp = 0.3469	Ac = 0.5307
		$Kappa$ = 0.0613	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4217	0.2220	Pr = 0.6551
H1	0.0783	0.2780	NPV = 0.7803
	Se = 0.8434	Sp = 0.5560	Ac = 0.6997
		$Kappa$ = 0.3994	

Table 20. SVM classification evaluation of EDCT algorithm using Moulin features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3053	0.3020	Pr = 0.5027
H1	0.1947	0.1980	NPV = 0.5042
	Se = 0.6107	Sp = 0.3960	Ac = 0.5033
		$Kappa$ = 0.0067	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3021	0.2924	Pr = 0.5082
H1	0.1979	0.2076	NPV = 0.5120
	Se = 0.6042	Sp = 0.4152	Ac = 0.5097
		$Kappa$ = 0.0194	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3264	0.2427	Pr = 0.5736
H1	0.1736	0.2573	NPV = 0.5971
	Se = 0.6528	Sp = 0.5147	Ac = 0.5837
		$Kappa$ = 0.1674	

6.3. Classification Results Applied to the Steganographic Method EDWT

With respect to the EDWT method, the results are provided in Tables 21–26. These results obtained with the classifiers FLD and SVM indicate that the values of the parameters Se, Sp, Pr, Ac, and $Kappa$ are high for all insertion rates and feature vectors (Farid, Shi, and Moulin). These results can easily inform us about the presence of hidden information; therefore, steganalysis can be concluded to be very effective. As a result, the insertion method is not robust. It should be noted that steganalysis is very effective here because both the steganagraphic method and feature vectors are based on multi-resolution wavelet decomposition.

Table 21. FLD classification evaluation of EDWT algorithm using Farid features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4786	0.0150	Pr = 0.9695
H1	0.0214	0.4850	NPV = 0.9577
	Se = 0.9571	Sp = 0.9699	Ac = 0.9635
		$Kappa$ = 0.9270	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4941	0.0056	Pr = 0.9888
H1	0.0059	0.4944	NPV = 0.9882
	Se = 0.9882	Sp = 0.9888	Ac = 0.9885
		$Kappa$ = 0.9770	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4993	0.0005	Pr = 0.9990
H1	0.0007	0.4995	NPV = 0.9987
	Se = 0.9987	Sp = 0.9990	Ac = 0.9989
		$Kappa$ = 0.9977	

Table 22. FLD classification evaluation of EDWT algorithm using Shi features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4048	0.0470	Pr = 0.8961
H1	0.0952	0.4530	NPV = 0.8263
	Se = 0.8095	Sp = 0.9061	Ac = 0.8578
		$Kappa$ = 0.7156	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4536	0.0311	Pr = 0.9358
H1	0.0464	0.4689	NPV = 0.9100
	Se = 0.9072	Sp = 0.9377	Ac = 0.9225
		$Kappa$ = 0.8450	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4753	0.0232	Pr = 0.9534
H1	0.0247	0.4768	NPV = 0.9508
	Se = 0.9507	Sp = 0.9535	Ac = 0.9521
		$Kappa$ = 0.9042	

Table 23. FLD classification evaluation of EDWT algorithm using Moulin features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3946	0.0650	Pr = 0.8587
H1	0.1054	0.4350	NPV = 0.8049
	Se = 0.7891	Sp = 0.8701	Ac = 0.8296
		$Kappa$ = 0.6592	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4394	0.0387	Pr = 0.9191
H1	0.0606	0.4613	NPV = 0.8839
	Se = 0.8789	Sp = 0.9227	Ac = 0.9008
		$Kappa$ = 0.8015	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4603	0.0321	Pr = 0.9348
H1	0.0397	0.4679	NPV = 0.9218
	Se = 0.9206	Sp = 0.9358	Ac = 0.9282
		$Kappa$ = 0.8564	

Table 24. SVM classification evaluation of EDWT algorithm using Farid features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4770	0.0230	Pr = 0.9541
H1	0.0230	0.4770	NPV = 0.9541
	Se = 0.9541	Sp = 0.9541	Ac = 0.9541
		$Kappa$ = 0.9082	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4893	0.0058	Pr = 0.9883
H1	0.0107	0.4942	NPV = 0.9789
	Se = 0.9787	Sp = 0.9884	Ac = 0.9835
		$Kappa$ = 0.9670	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4984	0.0084	Pr = 0.9835
H1	0.0016	0.4916	NPV = 0.9967
	Se = 0.9968	Sp = 0.9832	Ac = 0.9900
		$Kappa$ = 0.9800	

Table 25. SVM classification evaluation of EDWT algorithm using Shi features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3366	0.1658	Pr = 0.6700
H1	0.1634	0.3342	NPV = 0.6716
	Se = 0.6731	Sp = 0.6684	Ac = 0.6708
		$Kappa$ = 0.3415	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4107	0.1371	Pr = 0.7497
H1	0.0893	0.3629	NPV = 0.8024
	Se = 0.8213	Sp = 0.7257	Ac = 0.7735
		$Kappa$ = 0.5470	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4605	0.1175	Pr = 0.7967
H1	0.0395	0.3825	NPV = 0.9063
	Se = 0.9210	Sp = 0.7650	Ac = 0.8430
		$Kappa$ = 0.6859	

Table 26. SVM classification evaluation of EDWT algorithm using Moulin features.

5%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.3707	0.1108	Pr = 0.7699
H1	0.1293	0.3892	NPV = 0.7506
	Se = 0.7413	Sp = 0.7785	Ac = 0.7599
		$Kappa$ = 0.5198	
10%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4332	0.0725	Pr = 0.8567
H1	0.0668	0.4275	NPV = 0.8649
	Se = 0.8665	Sp = 0.8550	Ac = 0.8608
		$Kappa$ = 0.7215	
20%	**H0: Stego Images**	**H1: Cover Images**	
H0	0.4672	0.0724	Pr = 0.8659
H1	0.0668	0.4276	NPV = 0.9288
	Se = 0.9345	Sp = 0.8552	Ac = 0.8949
		$Kappa$ = 0.7897	

6.4. Discussion

The enhanced adaptive LSB methods of steganography in the spatial domain (EEALSBMR) and frequency domain (EDCT and EDWT) provide stego images with a good visual quality up to an embedding rate of 40%: the *PSNR* is over 50 dB, and the distortion is not visible to the naked eye. Security of the message contents, in case detected by an opponent, is ensured by using the chaotic system. On the other hand, we applied a universal steganalysis method that can work well with all known and unknown steganography algorithms. Universal steganalysis methods exploit the changes in certain inherent features of the cover images when a message is embedded. The accuracy of the classification (discrimination between two classes: cover and stego) of the system greatly relies on several factors, such as the choice of the right characteristic vectors, the classifier, and its parameters.

7. Conclusions

In this work, we first improved the structure and security of three steganagraphic methods that are studied in the spatial and frequency domain by integrating them with a robust proposed chaotic system. Following this, we built a statistical steganalysis system to evaluate the robustness of the three enhanced steganographic methods. In this system, we selected three different feature vectors, namely higher-order statistics of high-frequency wavelet sub-bands and their prediction errors, statistical moments of the characteristic functions of the prediction-error image, the test image, and their wavelet sub-bands, and both empirical PDF moments and the normalized absolute CF. After this, we applied two types of classifiers, namely FLD and SVM, with the RBF kernel.

Extensive experimental work has demonstrated that the proposed steganalysis system based on the multi-dimensional feature vectors can detect hidden messages using the EDWT steganographic method, irrespective of the message size. However, it cannot distinguish between cover and stego images using the EEALSBMR steganographic and EDCT methods if the message size is smaller than 20% and 15%, respectively.

Author Contributions: Funding acquisition, T.M.H.; Supervision, B.B., O.D. and M.K.; Writing—original draft preparation, D.B.; Writing—review & editing, S.E.A., T.M.H.

Funding: This work is supported by the National Foundation for Science and Technology Development (NAFOSTED) of Vietnam through the grant number 102.04-2018.06.

Acknowledgments: The authors thank the anonymous reviewers for useful comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xia, Z.; Wang, X.; Sun, X.; Liu, Q.; Xiong, N. Steganalysis of LSB matching using differences between nonadjacent pixels. *Multimed. Tools Appl.* **2016**, *75*, 1947–1962. [CrossRef]
2. Mohammadi, F.G.; Abadeh, M.S. Image steganalysis using a bee colony based feature selection algorithm. *Eng. Appl. Artif. Intell.* **2014**, *31*, 35–43. [CrossRef]
3. Luo, W.; Huang, F.; Huang, J. Edge Adaptive Image Steganography Based on LSB Matching Revisited. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 201–214.
4. Chan, C.K.; Cheng, L. Hiding data in images by simple LSB substitution. *Pattern Recognit.* **2004**, *37*, 469–474. [CrossRef]
5. Wu, H.C.; Wu, N.I.; Tsai, C.S.; Hwang, M.S. Image steganographic scheme based on pixel-value differencing and LSB replacement methods. *IEE Proc.-Vis. Image Signal Process.* **2005**, *152*, 611–615. [CrossRef]
6. Jung, K.; Ha, K.; Yoo, K. Image Data Hiding Method Based on Multi-Pixel Differencing and LSB Substitution Methods. In Proceedings of the 2008 International Conference on Convergence and Hybrid Information Technology, Daejeon, Korea, 28–30 August 2008; pp. 355–358. [CrossRef]
7. Huang, Q.; Ouyang, W. Protect fragile regions in steganography LSB embedding. In Proceedings of the 2010 Third International Symposium on Knowledge Acquisition and Modeling, Wuhan, China, 20–21 October 2010; pp. 175–178.

8. Xi, L.; Ping, X.; Zhang, T. Improved LSB matching steganography resisting histogram attacks. In Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China, 9–11 July 2010; Volume 1, pp. 203–206.
9. Swain, G.; Lenka, S.K. Steganography using two sided, three sided, and four sided side match methods. *CSI Trans. ICT* **2013**, *1*, 127–133. [CrossRef]
10. Islam, S.; Modi, M.R.; Gupta, P. Edge-based image steganography. *EURASIP J. Inf. Secur.* **2014**, *2014*, 1–14. [CrossRef]
11. Mungmode, S.; Sedamkar, R.; Kulkarni, N. A Modified High Frequency Adaptive Security Approach using Steganography for Region Selection based on Threshold Value. *Procedia Comput. Sci.* **2016**, *79*, 912–921. [CrossRef]
12. Akhter, F. A Novel Approach for Image Steganography in Spatial Domain. *arXiv* **2015**, arXiv:1506.03681.
13. Iranpour, M.; Rahmati, M. An efficient steganographic framework based on dynamic blocking and genetic algorithm. *Multimed. Tools Appl.* **2015**, *74*, 11429–11450. [CrossRef]
14. Kumar, R.; Chand, S. A reversible high capacity data hiding scheme using pixel value adjusting feature. *Multimed. Tools Appl.* **2016**, *75*, 241–259. [CrossRef]
15. Muhammad, K.; Ahmad, J.; Farman, H.; Jan, Z. A new image steganographic technique using pattern based bits shuffling and magic LSB for grayscale images. *arXiv* **2016**, arXiv:1601.01386.
16. Kordov, K.; Stoyanov, B. Least Significant Bit Steganography using Hitzl-Zele Chaotic Map. *Int. J. Electron. Telecommun.* **2017**, *63*, 417–422. [CrossRef]
17. Stoyanov, B.P.; Zhelezov, S.K.; Kordov, K.M. Least significant bit image steganography algorithm based on chaotic rotation equations. *C. R. L'Academie Bulgare Sci.* **2016**, *69*, 845–850.
18. Taleby Ahvanooey, M.; Li, Q.; Hou, J.; Rajput, A.R.; Chen, Y. Modern Text Hiding, Text Steganalysis, and Applications: A Comparative Analysis. *Entropy* **2019**, *21*, 355. [CrossRef]
19. Sadat, E.S.; Faez, K.; Saffari Pour, M. Entropy-Based Video Steganalysis of Motion Vectors. *Entropy* **2018**, *20*, 244. [CrossRef]
20. Yu, C.; Li, X.; Chen, X.; Li, J. An Adaptive and Secure Holographic Image Watermarking Scheme. *Entropy* **2019**, *21*, 460. [CrossRef]
21. Hashad, A.; Madani, A.S.; Wahdan, A.E.M.A. A robust steganography technique using discrete cosine transform insertion. In Proceedings of the 2005 International Conference on Information and Communication Technology, Cairo, Egypt, 5–6 December 2005; pp. 255–264.
22. Fard, A.M.; Akbarzadeh-T, M.R.; Varasteh-A, F. A new genetic algorithm approach for secure JPEG steganography. In Proceedings of the 2006 IEEE International Conference on Engineering of Intelligent Systems, Islamabad, Pakistan, 22–23 April 2006; pp. 1–6.
23. McKeon, R.T. Strange Fourier steganography in movies. In Proceedings of the 2007 IEEE International Conference on Electro/Information Technology, Chicago, IL, USA, 17–20 May 2007; pp. 178–182.
24. Abdelwahab, A.; Hassaan, L. A discrete wavelet transform based technique for image data hiding. In Proceedings of the 2008 National Radio Science Conference, Tanta, Egypt, 18–20 March 2008; pp. 1–9.
25. Singh, I.; Khullar, S.; Laroiya, D.S. DFT based image enhancement and steganography. *Int. J. Comput. Sci. Commun. Eng.* **2013**, *2*, 5–7.
26. Samata, R.; Parghi, N.; Vekariya, D. An Enhanced Image Steganography Technique using DCT, Jsteg and Data Mining Bayesian Classification Algorithm. *Int. J. Sci. Technol. Eng. (IJSTE)* **2015**, *2*, 9–13.
27. Karri, S.; Sur, A. Steganographic algorithm based on randomization of DCT kernel. *Multimed. Tools Appl.* **2015**, *74*, 9207–9230. [CrossRef]
28. Pan, J.S.; Li, W.; Yang, C.S.; Yan, L.J. Image steganography based on subsampling and compressive sensing. *Multimed. Tools Appl.* **2015**, *74*, 9191–9205. [CrossRef]
29. Ali, M.; Ahn, C.W.; Siarry, P. Differential evolution algorithm for the selection of optimal scaling factors in image watermarking. *Eng. Appl. Artif. Intell.* **2014**, *31*, 15–26. [CrossRef]
30. Farid, H. Detecting hidden messages using higher-order statistical models. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; Volume 2.
31. Shi, Y.Q.; Zou, D.; Chen, W.; Chen, C. Image steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network. In Proceedings of the 2005 IEEE International Conference on Multimedia and Expo, Amsterdam, The Netherlands, 6 July 2005; p. 4.

32. Wang, Y.; Moulin, P. Optimized Feature Extraction for Learning-Based Image Steganalysis. *IEEE Trans. Inf. Forensics Secur.* **2007**, *2*, 31–45. [CrossRef]
33. Abutaha, M. Real-Time and Portable Chaos-Based Crypto-Compression Systems for Efficient Embedded Architectures. Ph.D. Thesis, University of Nantes, Nantes, France, 2017.
34. Abu Taha, M.; El Assad, S.; Queudet, A.; Deforges, O. Design and efficient implementation of a chaos-based stream cipher. *Int. J. Internet Technol. Secur. Trans.* **2017**, *7*, 89–114. [CrossRef]
35. El Assad, S. Chaos based information hiding and security. In Proceedings of the 2012 International Conference for Internet Technology and Secured Transactions, London, UK, 10–12 December 2012; pp. 67–72.
36. Song, C.Y.; Qiao, Y.L.; Zhang, X.Z. An image encryption scheme based on new spatiotemporal chaos. *Opt.-Int. J. Light Electron Opt.* **2013**, *124*, 3329–3334. [CrossRef]
37. Tataru, R.L.; Battikh, D.; Assad, S.E.; Noura, H.; Déforges, O. Enhanced adaptive data hiding in spatial LSB domain by using chaotic sequences. In Proceedings of the 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Piraeus, Greece, 18–20 July 2012; pp. 85–88.
38. Assad, S.E.; Noura, H. Generator of Chaotic Sequences and Corresponding Generating System. Patent No. WO2011121218A1, 28 March 2011.
39. Farajallah, M.; El Assad, S.; Deforges, O. Fast and secure chaos-based cryptosystem for images. *Int. J. Bifurc. Chaos* **2015**. [CrossRef]
40. El Assad, S.; Farajallah, M. A new chaos-based image encryption system. *Signal Proc. Image Commun.* **2015**. [CrossRef]
41. Battikh, D.; El Assad, S.; Bakhache, B.; Déforges, O.; Khalil, M. Enhancement of two spatial steganography algorithms by using a chaotic system: Comparative analysis. In Proceedings of the 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), London, UK, 9–12 December 2013; pp. 20–25.
42. Mielikainen, J. LSB matching revisited. *IEEE Signal Process. Lett.* **2006**, *13*, 285–287. [CrossRef]
43. Habib, M.; Bakhache, B.; Battikh, D.; El Assad, S. Enhancement using chaos of a Steganography method in DCT domain. In Proceedings of the 2015 Fifth International Conference on Digital Information and Communication Technology and its Applications (DICTAP), Beirut, Lebanon, 29 April–1 May 2015; pp. 204–209.
44. Danti, A.; Acharya, P. Randomized embedding scheme based on dct coefficients for image steganography. *IJCA Spec. Issue Recent Trends Image Process. Pattern Recognit* **2010**, *2*, 97–103.
45. Boora, M.; Gambhir, M. Arnold Transform Based Steganography. *Int. J. Soft Comput. Eng. (IJSCE)* **2013**, *3*, 136–140.
46. Walia, E.; Jain, P.; Navdeep, N. An analysis of LSB & DCT based steganography. *Glob. J. Comput. Sci. Technol.* **2010**, *10*, 4–8.
47. Omrani, T. Conception et Cryptanalyse des CryptosystèMes LéGers Pour l'IoT. Ph.D. Thesis, El Manar University, Tunis, Tunisia, 2019
48. Song, X.; Liu, F.; Luo, X.; Lu, J.; Zhang, Y. Steganalysis of perturbed quantization steganography based on the enhanced histogram features. *Multimed. Tools Appl.* **2015**, *74*, 11045–11071. [CrossRef]
49. Lee, C.K. Infrared Face Recognition. 2004. Available online: https://apps.dtic.mil/dtic/tr/fulltext/u2/a424713.pdf (accessed on 26 July 2019).
50. Vapnik, V.N. *Statistical Learning Theory*; Adaptive and Learning Systems for Signal Processing, Communications, and Control; Wiley: Hoboken, NJ, USA, 1998.
51. Vapnik, V.N. An overview of statistical learning theory. *Neural Netw. IEEE Trans.* **1999**, *10*, 988–999. [CrossRef] [PubMed]
52. Schaefer, G.; Stich, M. UCID: An uncompressed color image database. In Proceedings of the Storage and Retrieval Methods and Applications for Multimedia 2004, San Jose, CA, USA, 18–22 January 2004; pp. 472–480.
53. Battikh, D.; El Assad, S.; Deforges, O.; Bakhache, B.; Khalil, M. *Stéganographie Basée Chaos Pour Assurer la Sécurité de L'information*; Presses Académiques Francophones: Sarrebruck, France, 2015. (In French)