Journal of Computers Vol. 35 No. 1, February 2024, pp. 109-129
doi: 10.53106/199115992024023501008

Robust Zero-Watermarking by Circular Features and 1-D NRDPWT Transformation

Hsiu-Chi Tseng[1,2*], King-Chu Hung[1,2]

[1] College of engineering, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan, ROC

[2] Department of Computer and Communication Engineering in the College of engineering, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan, ROC

0415913@nkust.edu.tw, kchung@nkust.edu.tw

Received 18 May 2023; Revised 19 September 2023; Accepted 5 December 2023

Abstract. This paper introduces a secure and robust zero-watermarking framework that leverages the advantages of zero-watermarking, ensuring non-destructive modification of original images and unlimited capacity. The proposed method enables robust watermark embedding while preserving the original image. It employs a novel feature extraction approach using circular areas based on image radius, enhancing feature resilience. Additionally, applying one-dimensional non-recursive discrete periodized wavelet transform (1-D NRDPWT) converts feature values into phi, contributing to enhanced stability and robustness. Enhanced security is achieved through the use of Shuffle and Pseudo-Random Number Generator (PRNG). Experimental results, evaluated using metrics such as Bit Error Rate (BER) and Normalized Correlation (NC), validate the exceptional performance of this watermarking technique. These findings underscore the framework's robustness, security, reliability, and integrity against both general and geometric noise attacks, making it a secure and robust solution for modern digital image copyright protection. In summary, our method offers an effective defense against various noise attacks while ensuring the highest watermark quality without compromising the original image. It is a significant advancement in copyright protection applications.

Keywords: zero-watermarking, one-dimensional non-recursive discrete periodized wavelet transform (1-D NRDPWT), phi, shuffle, Pseudo-Random Number Generator (PRNG), Bit Error Rate (BER), Normalized Correlation (NC)

1 Introduction

In light of the ever-advancing landscape of information technology and the evolving applications of digital media, several pertinent issues emerge. These issues encompass concerns related to distribution, replication, and plagiarism, underscoring the pressing demand for copyright protection, content verification, and data integrity assurance [1, 2]. To address these challenges effectively, watermarking technology emerges as a highly regarded solution. Watermarks can be categorized into two primary types based on their visual distinctiveness: visible and invisible [3, 4]. Visible watermarks are primarily employed to incorporate trademarks or distinctive logos for purposes of identification and advertising [5]. Conversely, invisible watermarks, recognized for their resilience, can be seamlessly integrated into digital content to facilitate content identification, tracking, and verification. Within the domain of invisible watermarking technology, zero-watermarking stands as a widely discussed application, drawing significant attention due to its unique advantages [6]. Notably, zero-watermarking has the potential to preserve the original image and circumvent traditional embedding capacity constraints [7].

The concept and methods of implementing zero-watermarking were initially introduced by Wen et al., [8]. Since then, several embedding algorithms have emerged. For instance, in early research, Liao et al. [9] proposed a neural network-based zero-watermarking technique, discussing two different approaches: one based on spatial domain using variance measurement, and another employing backpropagation neural networks. Additionally, Leng et al. [10] introduced a zero-watermark construction method using techniques like block cutting, PCA decorrelation, chaotic sequence generation, and wavelet transform. Lin et al. [11] proposed an image zero-watermarking scheme based on Generalized Arnold Transform (GAT) with spread spectrum and inverse spread techniques for feature extraction. The above solution was proposed as an early framework for zero-watermark-

* Corresponding Author

ing. However, at this stage of watermark embedding/extraction, the quality of the extracted watermark remains similar to that of other watermarking frameworks, without significant improvement in watermark quality. Nevertheless, it still retains the fundamental advantages of the zero-watermarking framework, preserving the integrity of the original image.

In recent years, zero-watermarking methods have seen continuous advancements and widespread applications. For instance, Xing et al. [12] proposed a method that utilizes Discrete Fourier Transform (DFT) to obtain a transformation coefficient matrix. This matrix is then used in a SIFT-DCT transformation applied to a grayscale host image to select a 32-bit feature sequence. This sequence is employed to distinguish the zero-watermark from the feature sequence of the encrypted watermark image. Huang et al. [13] employs a pre-trained DO-VGG model to extract deep abstract features from medical images and generates the zero-watermark using a perceptual hashing algorithm. Additionally, Liu et al. [14] introduced a method that combines Local Binary Patterns (LBP) with Discrete Cosine Transform (DCT). It extracts low-frequency feature vectors from digital images, performs hash sequence transformation, and binarization to embed the watermark. While these three zero-watermarking methods have improved watermark quality compared to earlier approaches, they share a common limitation. Both early and recent zero-watermarking frameworks have struggled to simultaneously address general noise attacks (such as gaussian or salt & pepper noise) and geometric noise attacks (such as translation or rotation) [15]. When subjected to significant noise attacks, they tend to exhibit noticeable resistance margin effects, reducing their stability and robustness against certain types of noise attacks.

Building upon the literature review in the previous section, we have observed the evolution of both past and recent zero-watermarking technologies. This study introduces a concise and secure watermark embedding framework leveraging the advantages of zero-watermark structures. In this work, we employ circular blocks for feature extraction. These extracted features are transformed into phi using a one-dimensional non-recursive discrete periodized wavelet transform (1-D NRDPWT) to enhance feature stability, robustness, and resistance to noise interference in binary mode. For security, we utilize generated pseudo-random keys and perform multiple shuffling rounds on different images. Experimental data confirms our method's resistance to both general noise attacks and geometric noise attacks, ensuring watermark quality across various attack intensities without compromising integrity. The most significant contribution lies in maintaining watermark quality as attack intensity increases, a crucial advantage for copyright protection.

The remaining sections of this paper are organized as follows: Section 2 presents Preliminaries, providing an explanation of the technical background used in this paper. Section 3 introduces the proposed method, detailing the embedding/extraction algorithms for zero-watermarking, with a focus on the integration of circular block feature extraction and 1-D NRDPWT transformation to achieve zero-watermark embedding and extraction. Section 4 provides experimental results and discussions. Finally, Section 5 summarizes the conclusions drawn from the application of the method proposed in this paper.

2 Preliminaries

2.1 Traditional Wavelet Transform

Wavelet transform is a signal processing technique with the primary purpose of decomposing a signal into multiple wavelet basis functions [16]. Each basis function corresponds to different frequencies and time intervals. These wavelet basis functions possess a localized nature, meaning they have limited duration and varying frequency and amplitude characteristics [17]. They can be irregular or asymmetric, allowing wavelet transform to efficiently capture the local features of non-stationary signals.

Traditional wavelet transform theory is rooted in the multi-scale decomposition theory [18]. The fundamental idea is to recursively partition and smooth the signal to obtain approximate and detailed signals at different scales. Wavelet basis functions are employed for signal analysis at each scale. Traditional wavelet transforms typically utilize orthogonal wavelet basis functions, satisfying orthogonality and completeness criteria, ensuring that the results of wavelet analysis can accurately reconstruct the original signal [19]. These basis functions are generated through recursive division and smoothing operations using low-pass and high-pass filters [20].

In recent years, many new wavelet transform methods have emerged in various signal processing applications, such as image processing, audio processing, and biomedical engineering [21-23]. These methods include non-orthogonal wavelet transforms and continuous wavelet transforms. They have addressed some issues associated with traditional methods, further enhancing the flexibility and efficiency of wavelet analysis in practical applica-

tions. The original signal is decomposed using high-pass and low-pass filters to obtain low-frequency coefficients containing essential information and high-frequency coefficients representing subtle signal variations. By continuously decomposing the low-frequency signal, valuable information can be integrated into the low-frequency coefficients. Simultaneously, less important signals in the high-frequency domain can be identified and represented with fewer data bits. These characteristics make wavelet transform a powerful tool widely used in watermarking and information hiding applications. It allows embedding hidden information into the wavelet coefficients of a signal and enables information retrieval.

2.2 Daubechies' Wavelet Transform

Daubechies' wavelet transform is a widely utilized method in wavelet analysis, initially introduced by the Belgian mathematician Ingrid Daubechies in 1988 [24]. This method enables the decomposition of signals into multiple wavelet bands, each possessing distinct frequency and time resolutions. This capability makes it a fundamental tool in signal processing.

The fundamental theory behind Daubechies' wavelet transform involves the recursive decomposition of a signal into a sequence of detail and approximate signals [25]. This process entails convolving and downsampling the signal through a series of low-pass and high-pass filters. The low-pass filter is responsible for capturing the low-frequency components of the signal, while the high-pass filter captures the high-frequency components. As a result, the sampling rate of the signal is reduced during this process, leading to reduced time resolution in each band.

Using Daubechies' wavelet transform, a signal can be effectively decomposed into multiple bands with varying time and frequency resolutions [26]. Typically, detail bands are employed to capture the high-frequency components of the signal, while approximate bands are used to represent the low-frequency components. This decomposition allows for independent processing of each band, facilitating signal analysis and manipulation.

One of the primary advantages of Daubechies' wavelet transform is its adaptability in terms of time and frequency resolution, making it suitable for handling non-stationary signals. This method has found widespread applications in digital signal processing, particularly in tasks such as image compression and noise reduction. It has become a standard algorithm in various industrial and commercial domains.

In wavelet transformation, Daubechies wavelets, including D4 and D6, are commonly employed for signal processing [27]. The primary distinction between using D4 and D6 lies in their filter coefficients, which result in different frequency responses and reconstruction performance. D4 is a wavelet basis function of length 4 (n = 4), with both its low-pass and high-pass filters having a length of 4. On the other hand, D6 is a wavelet basis function of length 6 (n = 6). These differing filter lengths lead to variations in frequency responses and reconstruction performance between D4 and D6. D6 exhibits a flatter frequency response and better preservation of low-frequency signals, making it more suitable for processing low-frequency signals when compared to D4. The coefficients of the low-pass filter vector h and high-pass filter vector g can be determined using the definitions in formulas (1). Table 1 presents the coefficients of the D6 filter.

$$h(k)=\frac{1}{4\sqrt{2}}\left[\left(1+\sqrt{3}\right)\left(1+2\cos\left(\frac{2\pi k}{6}\right)\right)+\left(3+\sqrt{3}\right)\left(1+2\cos\left(\frac{4\pi k}{6}\right)\right)\right],\ g\left(n-k\right)=\left(-1\right)^{k}a_{k} \tag{1}$$

Table 1. The coefficients of the D6 filter

Coiflet filter k	$h(k)$	$g(k)$
0	0.3326705529	0.0352262918
1	0.8068915093	0.0854412738
2	0.4598775021	-0.1350110200
3	-0.1350110200	-0.4598775021
4	-0.0854412738	0.8068915093
5	0.0352262918	-0.3326705529
6	0	0
7	0	0
(ignore)		
N-1	0	0

2.3 1-Dimensional Non-Recursive Discrete Periodized Wavelet Transform (1-D NRDPWT)

Daubechies wavelet transform in wavelet analysis is regarded as a polynomial encoder, where wavelet functions are represented by polynomial coefficients. Each Daubechies wavelet is represented as a polynomial, and these polynomial coefficients remain constant across different signals. This characteristic has led to the widespread application of Daubechies wavelets in signal processing.

In Discrete Wavelet Transform (DWT), signals are decomposed into wavelet functions of various scales. These wavelet basis functions are obtained by scaling and shifting Daubechies wavelets, using a recursive process for signal decomposition. However, when dealing with periodic signals, simplifications can be achieved by exploiting their periodic nature, eliminating the need for recursive processing. This approach is known as NRDPWT, which is based on Daubechies wavelet transform and is particularly suitable for analyzing periodic signals [28].

A key advantage of NRDPWT is its ability to simultaneously compute all frequency band coefficients without the need for iterative signal decomposition. This addresses some of the limitations of traditional recursive wavelet transforms while maintaining the efficiency and accuracy of wavelet transformation. This feature makes NRDPWT a powerful tool, especially when dealing with periodic signals. Fig. 1 illustrates the schematic diagram of the coefficient decomposition obtained when the length of the original signal is assumed to be N_j =8, j = 3.

$$H=\begin{bmatrix} h_0 & 0 & 0 & \cdots & h_3 \\ h_1 & 0 & 0 & \cdots & h_4 \\ h_2 & h_0 & 0 & \cdots & 0 \\ h_3 & h_1 & 0 & \cdots & 0 \\ 0 & h_2 & h_0 & \ddots & 0 \\ 0 & h_3 & h_1 & \ddots & 0 \\ 0 & 0 & h_2 & \ddots & h_0 \\ 0 & 0 & h_3 & \cdots & h_1 \end{bmatrix}, G=\begin{bmatrix} g_0 & 0 & 0 & \cdots & g_3 \\ g_1 & 0 & 0 & \cdots & g_4 \\ g_2 & g_0 & 0 & \cdots & 0 \\ g_3 & g_1 & 0 & \cdots & 0 \\ 0 & g_2 & g_0 & \ddots & 0 \\ 0 & g_3 & g_1 & \ddots & 0 \\ 0 & 0 & g_2 & \ddots & g_0 \\ 0 & 0 & g_3 & \cdots & g_1 \end{bmatrix}. \tag{2}$$

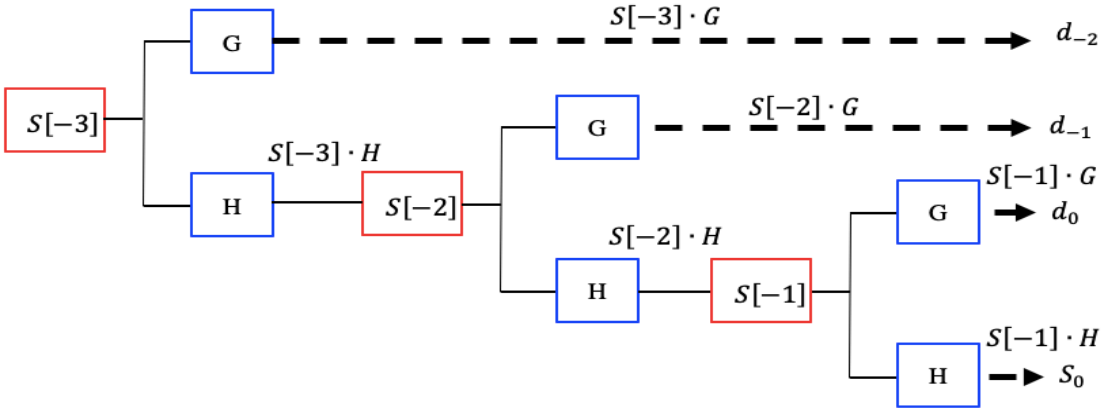


Fig. 1. Decomposition process of filter coefficients in non-recursive wavelet transform

The d_{-2} level is obtained by applying the high-pass filter coefficient matrix G to the original signal, while d_{-1} is obtained by applying both the low-pass filter coefficient matrix H and the high-pass filter coefficient matrix G. This process continues recursively. Let A_j be an N × 2^{-j} matrix, which is composed of filter coefficient combinations from a set of 2^{-j} row vectors in $H^{-j-J-1}G$. Then, the row vector coefficients of H^{-j} are used to represent B0, and finally, all the coefficients are combined to form a 1-D NRDPWT, which is transformed into matrix A [29], as shown in the following formula (3):

$$A = [B_0, A_0, A_{-1}, A_{-2}, A_{-3}, ..., A_{J+1}]. \quad (3)$$

Then, Let $P_j = [p_{j0}, \ldots, p_{j(N-1)}]^T$ denotes an $N \times 1$ normalized column vector with each element p_{jn} of A_j. Vectors P_j for $J < j \leq 0$ are inherent with the following properties.

2.4 Fisher-Yates Algorithm

When the need arises to perform a random shuffle of an array or list, one of the go-to shuffling algorithms is the Fisher-Yates algorithm [30]. It finds widespread application in various domains, including random shuffling, encryption, and simulations. What sets the Fisher-Yates algorithm apart from other random permutation algorithms is its remarkable time complexity and randomness performance, ensuring that each permutation has an equal probability. It stands as a straightforward yet highly effective algorithm for generating random permutations.

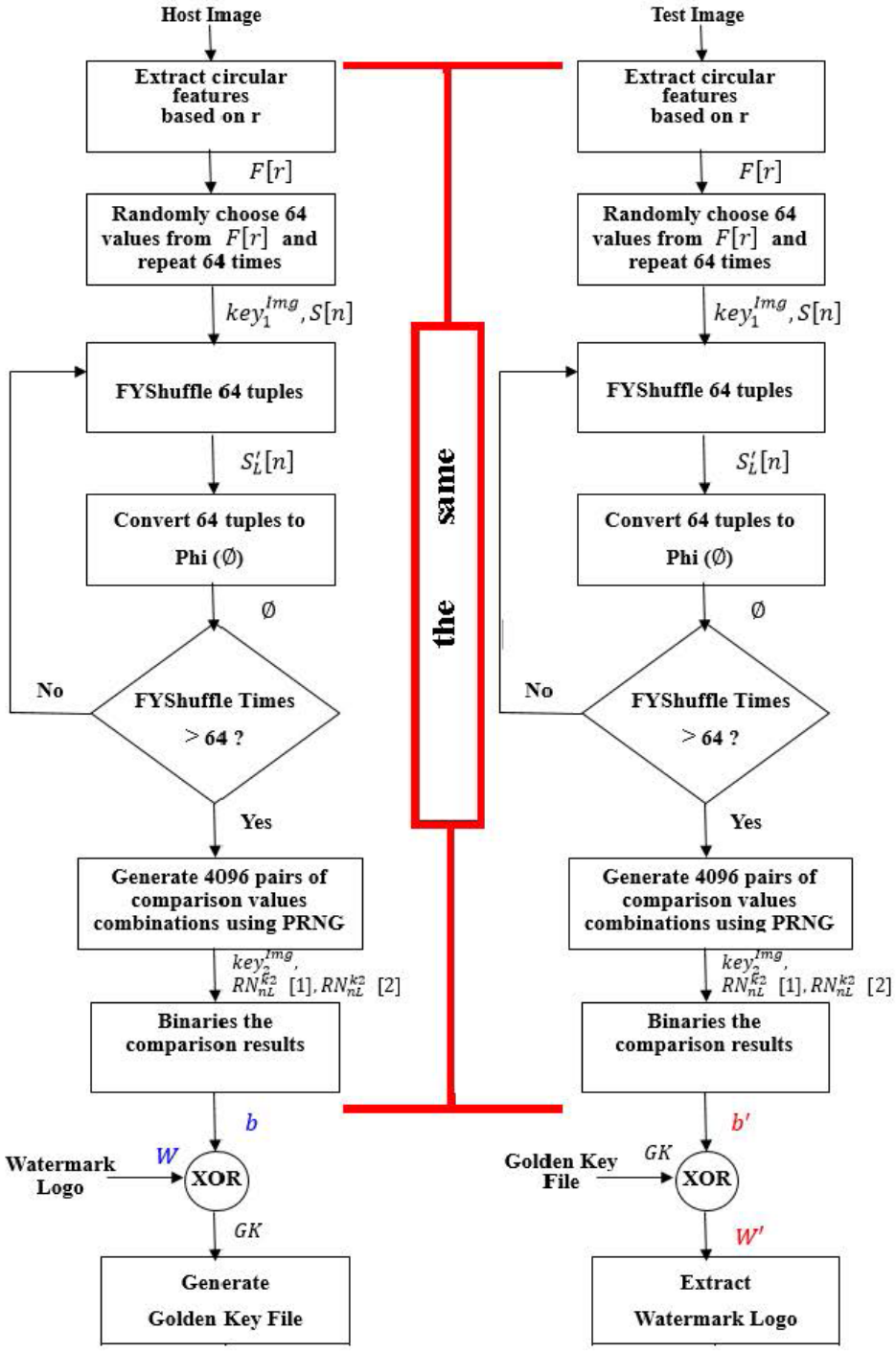
In essence, the Fisher-Yates shuffle algorithm operates as a highly efficient and equitable method for random sorting. The fundamental concept behind the Fisher-Yates algorithm involves traversing the array from the end to the beginning. For the current element under consideration during traversal, a random number j is generated, where 0 <= j <= i represents the current position being traversed. Subsequently, the current element is swapped with the element at index j. This process continues by advancing the current processing position by one step until the first element of the array is processed. After n swaps, a randomized array permutation is achieved. The algorithm is presented below:

```
program FYShuffle (NumberArray)
    {Assuming n is a Length of NumberArray }
    for i from n-1 down to 1 do
        j = random integer with 0 <=j <=i
        swap NumberArray[i] with NumberArray[j]
    end
end.
```

3 Proposed Method

Our proposed method introduces an innovative watermark embedding scheme that capitalizes on the architectural characteristics of zero-watermarking. In this approach, circular blocks within the host image are employed for extracting the image's feature code. Subsequently, we apply the 1-D NRDPWT technique to enhance the stability of feature code extraction.

Additionally, this paper incorporates Fisher-Yates shuffling technology and a pseudo-random number generator (PRNG) enhancement method to bolster the security of the scheme. Under the zero-watermarking architecture, both the embedding and extraction of the watermark entail similar steps in the implementation process, as illustrated in Fig. 2. Notably, the flowchart demonstrates that no inverse transformation is required for embedding and extraction within the zero-watermarking architecture. The experimental data presented in this paper highlights the method's commendable performance in terms of security and stability.



(a) Embedding watermark architecture (b) Extracting watermark architecture

Fig. 2. Propose the architecture for embedding/extracting watermark

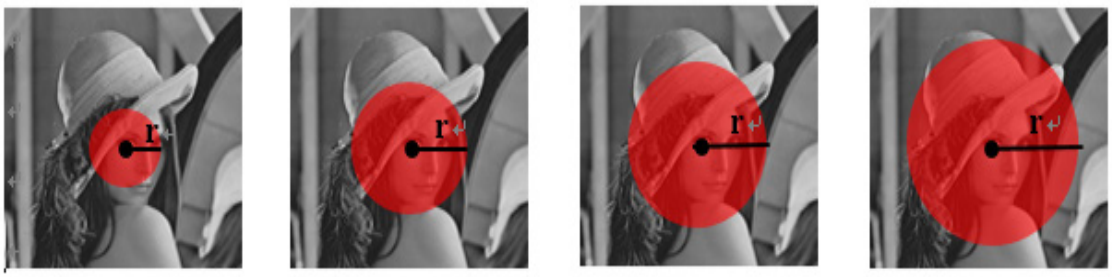
3.1 Embedding Watermark

Extract Image Features. The host image is defined as a square image of size $M \times M$, with its center located at ($\frac{M}{2}$, $\frac{M}{2}$). Starting from this central position, a circular region with radius $r$is extracted from the image in an outward direction. For each pixel position (i, j) in the image, the distance $d_{i,j}$ from the pixel to the center of the circle is calculated. The average pixel value $F[r]$of the circular region within the extracted range is then calculated using the following formula. This formula is used to extract a range of $r$features from 0 to $\frac{M}{2}$, as illustrated in Fig. 3.

$$d_{i,j} = \sqrt{\left(i - \frac{M}{2}\right)^2 + \left(j - \frac{M}{2}\right)^2},$$

$$F[r] = \frac{1}{\pi r^2} \sum_{i=0}^{r} \sum_{j=0}^{r} Img(i, j),\ if\ \left(d_{i,j} \le r\right)\ and\ r \in \left\{0, 1, 2, ..., \frac{M}{2}\right\}. \quad (4)$$

Fig. 3 provides illustrative examples of circular regions selected using different radii, denoted as (a) to (d). Each corresponds to a distinct radius defining the circular regions. To extract features from the host image, we employ a straightforward yet highly effective method that utilizes circular regions with different radii. By computing the mean pixel value within these regions, we can capture valuable information about the image content across various scales. This approach has demonstrated its efficacy in numerous image processing applications.



(a) Let r = 5 (b) Let r = 55 (c) Let r = 105 (d) Let r = 155

Fig. 3. Average pixel coverage within circular area of radius r

Image Features to 1-D NRDPWT Transform. In accordance with different digital images, distinct random keys, denoted as key_1^{Img}, are assigned. Utilizing PRNG techniques, $key1$ generates 64 random numerical sets, RN^{k1}, ranging between 0 and $\frac{M}{2}$ (the value domain of r). Based on the random RN^{k1} values for each image, a collection of random feature values, $F[RN_1^{k1}]$ to $F[RN_{64}^{k1}]$, is extracted from the feature set $F[RN^{k1}]$, forming set $S[n]$. This extraction process iterates Formula (5) 64 times, where n in $S[n]$ ranges from 1 to 64. For instance, $S[1]$ consists of 64 random $F[RN^{k1}]$, values. With each iteration, a new set of RN^{k1} values is generated to produce the corresponding $S[n]$ values.

$$
\begin{aligned}
&RN^{k1}=\left[key_1^{Img},\ random\left(\frac{M}{2}\right)\right],\\
&RN^{k1}=\left[RN_1^{k1}\ RN_2^{k1}\ RN_3^{k1}\ RN_4^{k1}\ RN_5^{k1}\ RN_6^{k1},\ \ldots,\ \ldots\ \ldots\ RN_{64}^{k1}\right],\\
&\text{where, } k1=key_1^{Img},\\
&S[n]=\left[F\left[RN_1^{k1}\right],F\left[RN_2^{k1}\right],F\left[RN_3^{k1}\right],F\left[RN_4^{k1}\right],F\left[RN_{3\ 5}^{k1}\right],F\left[RN_6^{k1}\right],\ldots,\ldots,\ldots,F\left[RN_{64}^{k1}\right]\right],\\
&\text{where } n\in\{1,2,3,4,\ldots,64\}.
\end{aligned}
\tag{5}
$$

To enhance the security and strengthen the robustness of the acquired signal $S[n]$, we employ a 1-D NRDPWT transformation on the signal $S[n]$. Using the Fisher-Yates shuffle algorithm, we shuffle the values of $S[n]$ L times to extract L of ϕ values for each $S[n]$. Here, L represents the a-th iteration of the Fisher-Yates algorithm used for shuffling $S[n]$, assuming a equals 64.

Each $S'_L[n]$ value set undergoes inner product operations, denoted as $P_j|_{j=0}$, with the 64 vectors generated by the dot $P_j|_{j=0}$. Subsequently, we obtain phi (ϕ) values, denoted as ϕ, using the inverse cosine transform, as illustrated in (6).

$$
\begin{aligned}
&S'_L[n]=FYShuffle(S[n])_L,\\
&\varnothing[n\times L]=\cos^{-1}\left(S'_L[n]\cdot P_j|_{j=0}\right),\\
&\text{where } L\in\{1,2,3,4,...,64\},\ n\in\{1,2,3,4,...,64\}
\end{aligned}
\tag{6}
$$

This formula is applied to each $S[n]$ sequentially, ranging from n = 1 to 64, resulting in 64 iterations of computations as per (6). For instance, after iterations of the Fisher-Yates shuffle algorithm for $S[1]$, the formula yields a of ϕ values. In the case where n = 64 and L = 64, this process generates $n \times L = 64 \times 64 = 4096$ of $\phi[n \times L]$values. These 64 values represent the dimensions of the binary watermark, interrelated and coordinated with one another.

Convert To Binarization Pattern. Building upon the previous steps, the 1-D NRDPWT transformation is applied to the signal $S[n]$ based on its feature values. This transformation yields $n \times L$ of ϕ elements, referred to as $\phi[n \times L]$. Subsequently, a different random $k2$, key_2^{Img}, is used for each image. A PRNG is employed to generate pairs of two random numbers, such as $RN_1^{k2}[1]$ and $RN_1^{k2}[2]$, resulting in $n \times L$ sets of random numbers RN_{nL}^{k2}, with values ranging from 1 to $n \times L$. If ($\phi[RN_1^{k2}[1]] > \phi[RN_1^{k2}[2]]$), the binary pattern value is set to 1; otherwise, it is set to 0. Finally, a binary pattern b with a length of $n \times L$ is generated using the following formula:

$$
\begin{aligned}
&\begin{cases} RN_{nL}^{k2}[1]=\left[key_2^{Img},\ random(1,n\times L)\right]\\ RN_{nL}^{k2}[2]=\left[key_2^{Img},\ random(1,n\times L)\right]\end{cases},\\
&\text{where } k2=key_2^{Img},\ nL\in\{1,2,3,4,\cdots,\cdots,\cdots,4096\},\\
&b=\begin{cases}1, & if\left(\varnothing\left[RN_{nL}^{k2}[1]\right]>\varnothing\left[RN_{nL}^{k2}[2]\right]\right)\\ 0, & otherwise\end{cases}.
\end{aligned}
\tag{7}
$$

Generate Golden Key. In the final step, the embedding of the binary watermark logo is completed. An XOR operation is performed between the binary pattern b with a length of $n \times L$, and the target binary watermark logo W, as indicated in (8). This operation generates the watermark embedding key GK, marking the completion of the entire watermark embedding process.

$$
GK=b\oplus W. \tag{8}
$$

3.2 Extracting Watermark

In the zero-watermarking framework, the process of extracting the watermark does not employ a reversible extraction method. Instead, the extraction of the zero-watermark closely resembles the embedding process, as described in the algorithmic formulas (5) to (7). The architecture depicting the watermark embedding and extraction processes is illustrated in Fig. 2, with repeated steps marked. This illustrates the identical computational steps during watermark extraction. Therefore, we omit the repetition of the same algorithmic process during watermark extraction. However, in the final step of watermark extraction, it is necessary to perform the XOR ($\oplus$) operation between the watermark key *GK* generated by (8) and the extracted binary pattern *b'* to obtain the extracted watermark *W'*, as shown in (9):

$$W' = b' \oplus GK \,. \tag{9}$$

It is important to note that in this formula. During the watermark extraction process, the binary pattern is referred to as *b'*, which is distinct from the binary pattern b used in the watermark embedding process. This distinction arises because, during watermark extraction, the image used for extracting the watermark may have been tampered with or subjected to noise attacks. Therefore, the extracted binary pattern may be influenced and may differ from the original *b*.

4 Experimental Analysis and Results

This paper's experiment utilizes common grayscale images that are widely used as benchmarks. These images are of size 512512 and include Airplane, Baboon, Barbara, Cameraman, Fishing boat, Gold hill, Lena, and Peppers, as depicted in Fig. 4(a) to Fig. 4(h). Additionally, a binary image of size 6464 is employed as the binary watermark logo, which is illustrated in Fig. 5.

In the following, we will design 12 different attacks with varying intensities for the zero-watermark framework proposed in this paper, as listed in Table 2. We selected eight images for experimentation, as shown in Fig. 4, and individually subjected each image to all 12 attacks for watermark embedding and extraction tests. The Peak Signal-to-Noise Ratio (*PSNR*) values in Table 2 represent the average *PSNR* after applying each noise attack to the eight images. Subsequently, we will organize and compare the experimental and analytical results obtained for each image under different attack types and intensities.

(a) Airplane (b) Baboon (c) Barbara (d) Cameraman

(e) Fishing boat (f) Gold hill (g) Lena (h) Peppers

Fig. 4. Our host images

Fig. 5. Binary watermark logo

Table 2. Description symbol of the original ant it's 12 types of noise attacks

Types of noise attacks	Symbol	PSNR(db)
Original Image	N0	-
Gaussian noise (σ = 0.01)	N1	28.16742
Salt & pepper (σ = 0.15)	N2	11.07210
Poisson noise (σ = 0.01)	N3	28.86735
JPEG compress (Q=90%)	N4	38.06961
JPEG compress (Q=50%)	N5	32.17544
JPEG compress (Q=10%)	N6	27.62172
Left rotate (angle=1°)	N7	19.35058
Left rotate (angle=50°)	N8	9.46347
Left rotate (angle=90°)	N9	10.73379
Right rotate (angle=1°)	N10	19.32716
Right rotate (angle=50°)	N11	9.458048
Right rotate (angle=90°)	N12	10.73379

In this section, we will not only conduct a comprehensive analysis and comparison of the experiments conducted with our designed framework but also extend our analysis to compare the results with other recent methods that utilize zero-watermarking, such as Xing et al. [12], Huang et al. [13], and Liu et al. [14]. This comparative analysis aims to provide a more objective and impartial perspective on the effectiveness of our proposed method in the context of the latest approaches in the field of zero-watermarking.

4.1 Experimental Analysis

In the watermark embedding method proposed in this paper, there is a crucial relationship between the extraction of image feature codes and digital images. Furthermore, signal transformation plays a vital role in the strong embedding/extraction of watermarks within this process. To ensure the watermark's effectiveness, reliability, integrity, and robustness against various types of noise attacks, we will analyze these two closely related sets of data in this section. Through this analysis, we can verify and confirm the method's stability against different types of noise interferences.

Based on Fig. 4, we conducted experiments with 12 different types of noise attacks on each original image. The types of noise attacks employed and the corresponding average PSNR evaluation data can be found in Table 2. The PSNR evaluation method is detailed in Formula (10). In this section, our focus is on analyzing the extracted features from eight experimental images and comparing the presentation of these features between the original images and the 12 attack images with noise interference. Through this analysis, our primary goal is to assess the stability of the features obtained from the original images and various noise-affected attack images under different conditions.

$$MSE = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} \left(I_{x,y} - I'_{x,y} \right)^2 ,$$

$$PSNR = 10 \times log_{10} \left(\frac{Max^2}{MSE} \right). \quad (10)$$

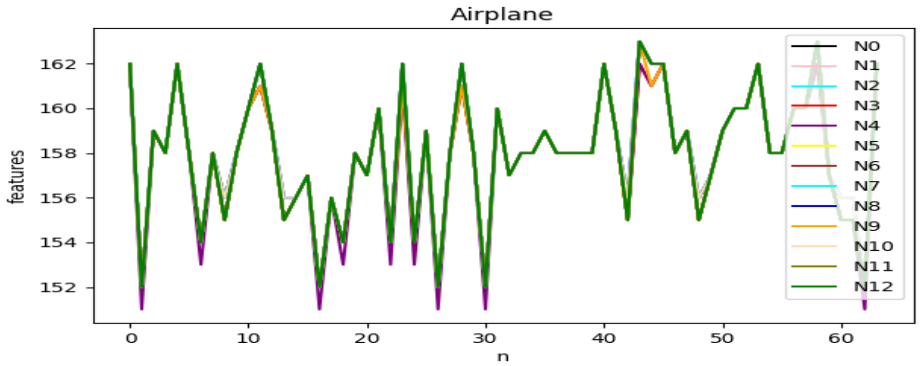
MSE is used to measure the difference between two images, typically an original image (I) and a processed image (I'). Here, M and N represent the width and height of the images, respectively. $I_{x,y}$ denotes the pixel value in the original reference image, while $I'_{x,y}$ represents the pixel value in the processed test image. A smaller *MSE* value indicates that the processed image is closer to the reference image, indicating less difference.

PSNR serves as an indicator of image quality and is often used to assess the information loss during processes like image compression or watermark embedding. In the formula, *Max* represents the maximum pixel value in the image, typically 255 (for 8-bit grayscale images). *MSE* is the mean squared error value. *PSNR* is measured in decibels (dB), and a higher *PSNR* value corresponds to better image quality [31].

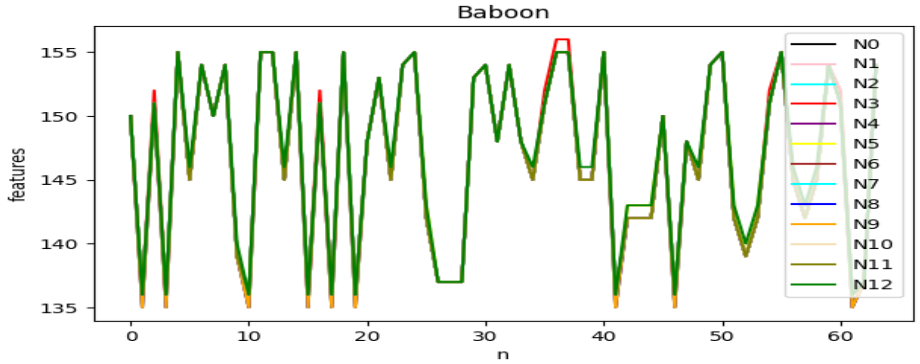
Data Analysis of Image Features. Using the algorithm proposed in this paper for extracting image features, we conducted a comprehensive experimentation involving a set of eight distinct graphs. Each graph was subjected to various attacks by introducing twelve different types of noise. By extracting the eigenvalues and employing a shuffling technique, we using PRNG randomly selected 64 features from the resulting dataset. We conducted a comprehensive analysis using the first set of 64 randomly selected values extracted from these eight images, as illustrated in Fig. 6. Each symbol in the figure, from N0 to N12, corresponds to different noise attack identifiers specified in Table 2. Our objective was to assess the resistance of these images to various noise attacks and whether there were deviations from their original feature values.

By comparing the N0 data of these eight images with the corresponding feature values of the original host image, we observed subtle variations in the impact of different attacks, as evident in the chart. What's remarkable is that these variations persist even under the influence of 12 different attacks (N1 to N12).

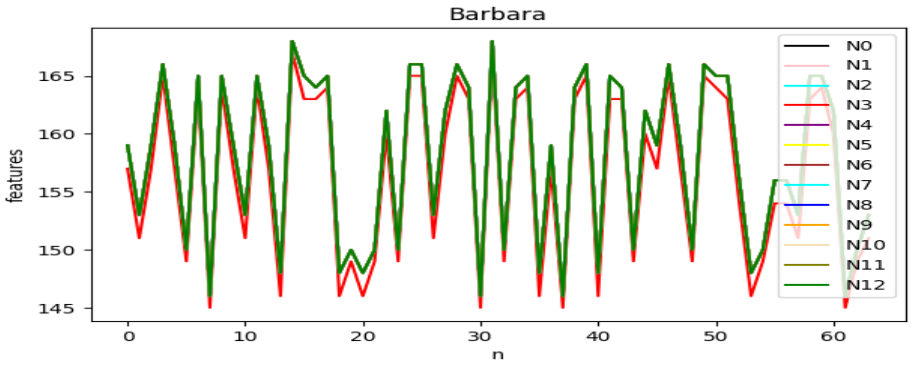
However, it is worth noting the stability in the waveform shape of the image features presented in our analysis. Despite noise interference, the numerical ranges within the images still closely approximate those of the original image. The contrast variation refers to a situation where one value falls within an extremely high range, while another value falls within an extremely low range. This remarkable consistency underscores the robustness of the extracted features, indicating their resilience even in the face of various noise attacks.



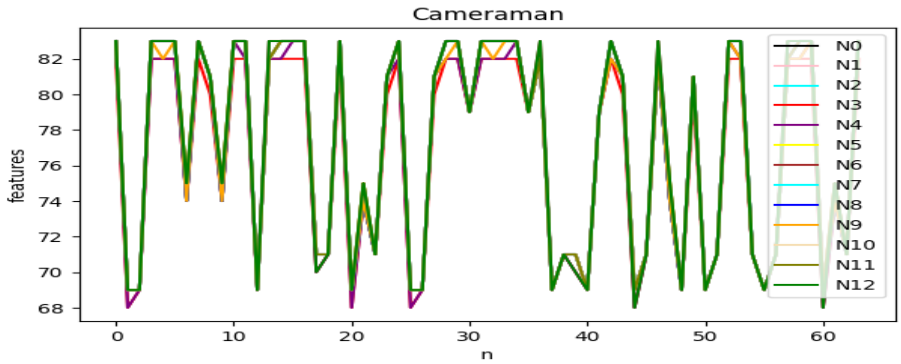
(a) Airplane



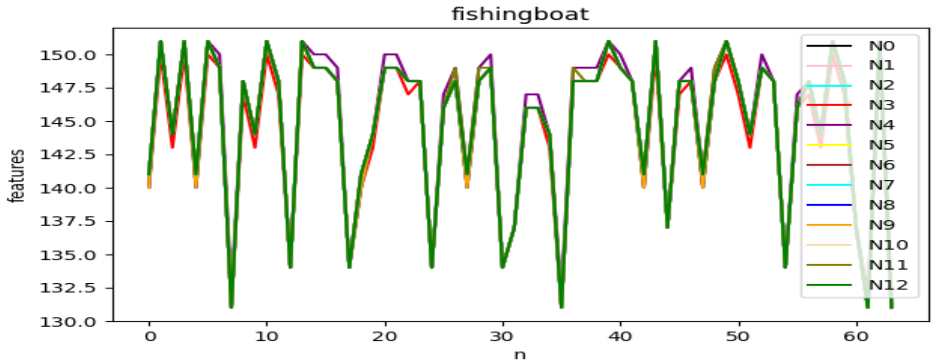
(b) Baboon



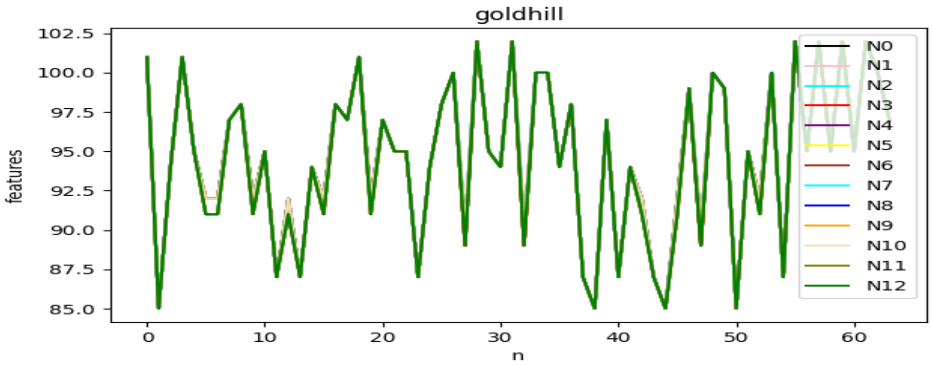
(c) Barbara



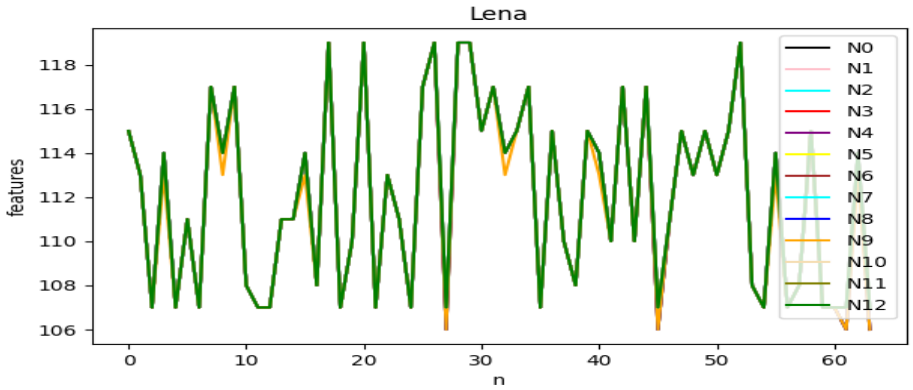
(d) Cameraman



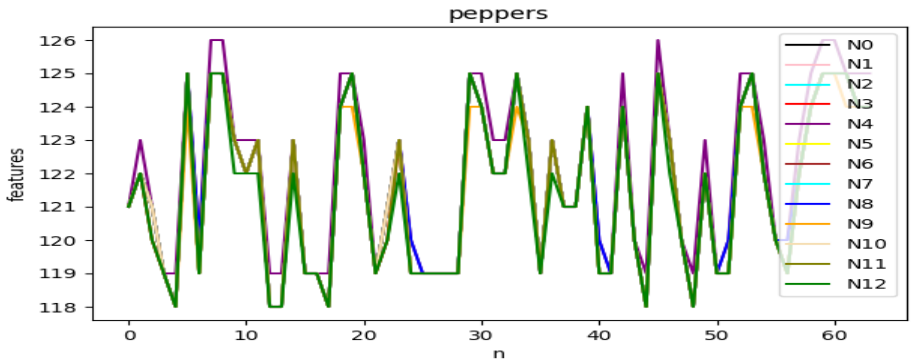
(e) Fishing boat



(f) Gold hill



(g) Lena



(h) Peppers

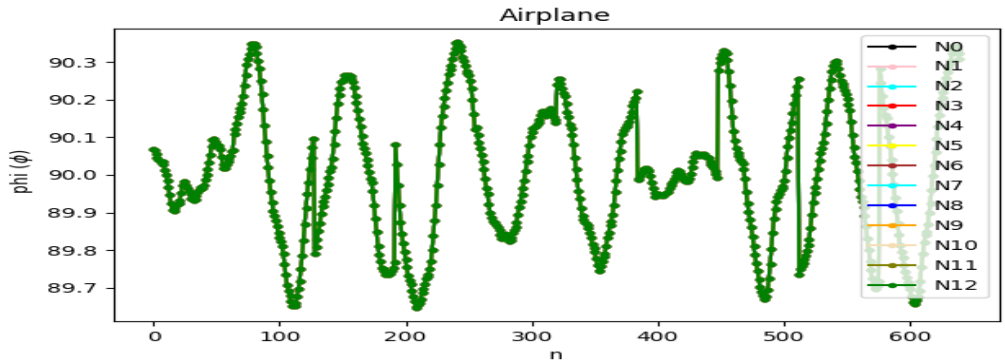
Fig. 6. Analysis and comparison of the first Fisher-Yates shuffle features for each host image

The results in Fig. 6 how that fluctuations are minimally impacted, whether subjected to general noise attacks or geometric noise attacks. Achieving stability under both types of attacks simultaneously has posed a challenge in many previous research approaches. In this phase, we effectively mitigated geometric noise attacks through circular block extraction. Additionally, we bolstered stability against general noise attacks through extensive value sampling.

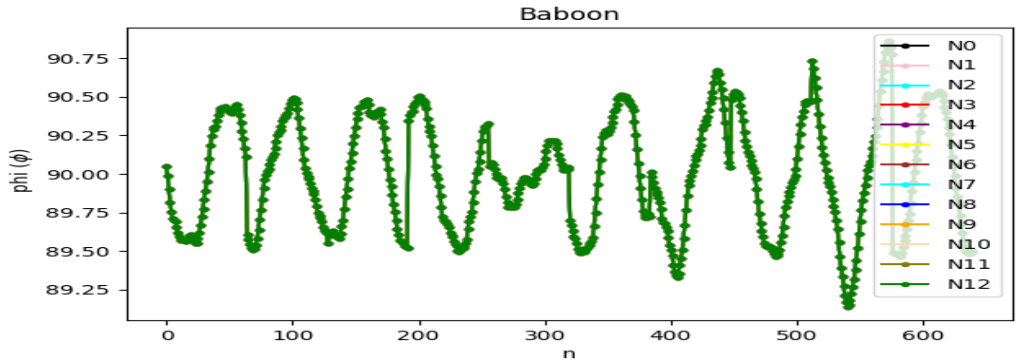
Data Analysis of Phi (ϕ). Building upon the previous phase focused on stability, we applied formula (6), a transformation formula introduced in this study, to convert the extracted feature values into phi values. To conduct a more comprehensive and precise analysis, we expanded our examination. In the earlier stage, we extracted 64 feature values, but now we have extended our analysis to encompass ten sets of 64 randomly chosen feature values, all subject to transformation into phi values. This comprehensive approach entails analyzing a total of 640 phi values, ensuring a more effective, comprehensive, and representative evaluation.

Fig. 7 illustrates the analysis and comparison of these 640 phi values across the eight images. The symbols N0 to N12 used in this analysis phi correspond to the specified noise attack identifiers detailed in Table 2.

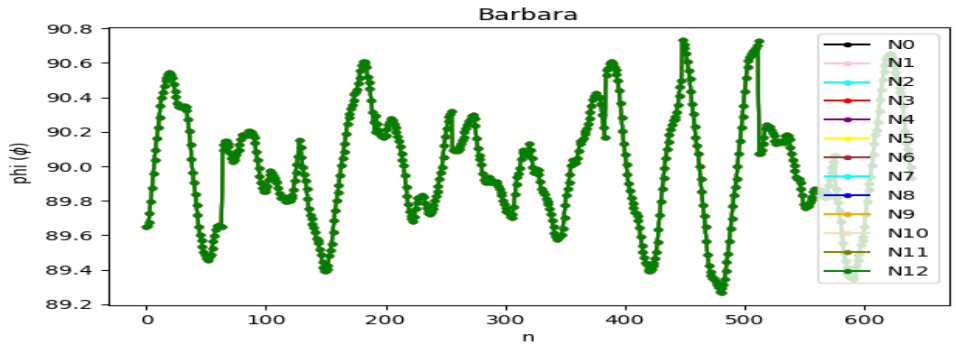
The analysis results presented in Fig. 7 unmistakably indicate minimal variations between the attacked images (N1 to N12) and the host image (N0) data. A straightforward observation of the color distribution in Fig. 6 and Fig. 7 underscores the remarkable stability exhibited in this analysis phase. In particular, we enlarged the image in Fig. 7(h) to compare it with Fig. 6(h). This is because Fig. 6(h) is the most affected by noise attacks among all the analyses, showing more significant variations. However, in Fig. 7(h), we can observe that the analysis results exhibit a level of stability almost identical to Fig. 7(a) to Fig. 7(g). In essence, compared to the feature values in Fig. 6, this performance, achieved through the transformation into phi values via 1-D NRDPWT, showcases even stronger robustness, thereby underscoring its potential to enhance watermark embedding quality.



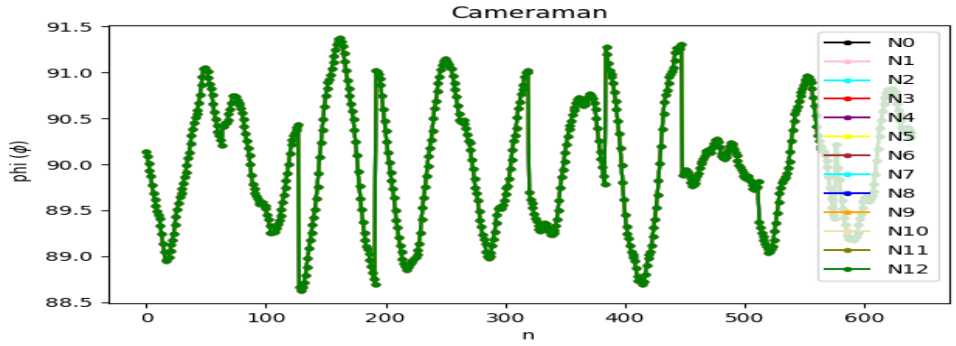
(a) Airplane



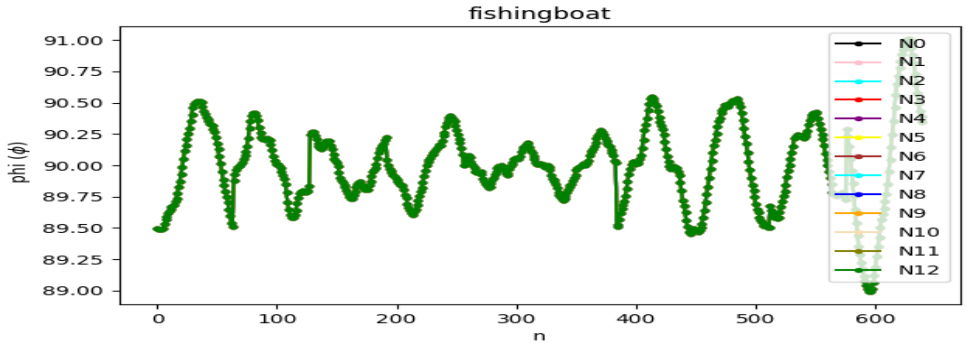
(b) Baboon



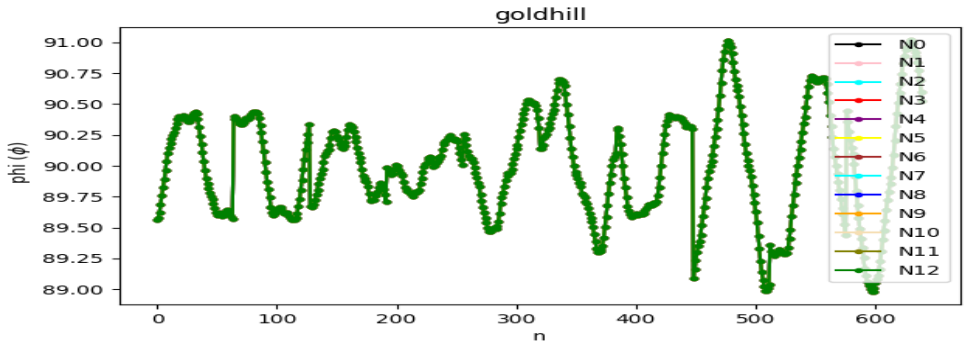
(c) Barbara



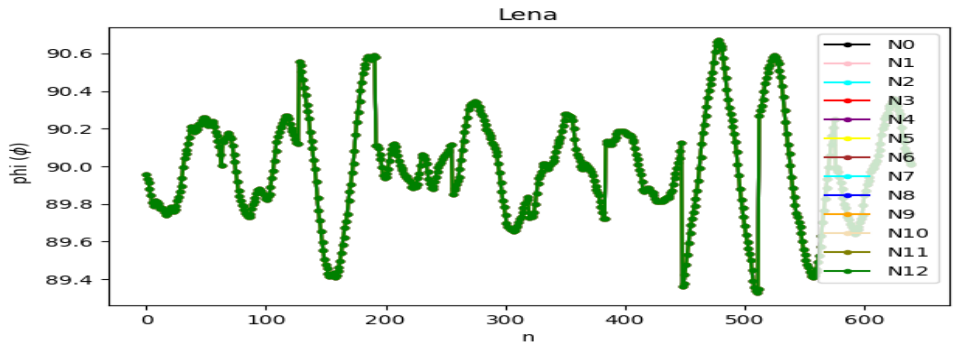
(d) Cameraman



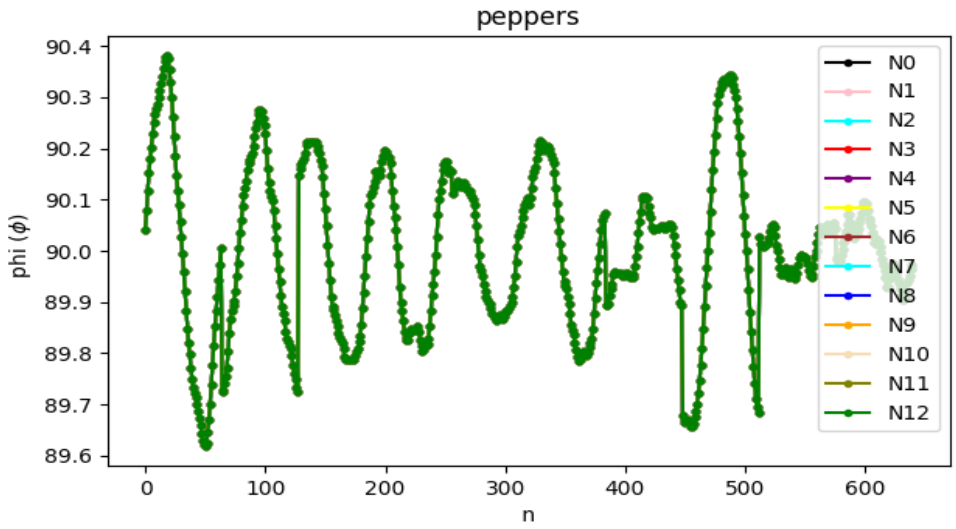
(e) Fishing boat



(f) Gold hill



(g) Lena



(h) Peppers

Fig. 7. Analysis and comparison of the first transform to phi for each host image

In Fig. 7, we can clearly observe the stability under both general and geometric attacks. These stable and straightforward waveforms demonstrate a significant improvement over the instability observed in Fig. 6. This also showcases the excellent data performance and handling of zero-watermark feature values.

4.2 Experimental Results

In this section, we employed objective evaluation criteria, namely the Bit Error Rate (BER) and Normalized Correlation (NC) to assess the effectiveness and feasibility of watermark embedding and extraction within the proposed zero-watermarking framework. The evaluation formulas for BER and NC are represented by formula (11) and (12), respectively. To validate the efficacy of our approach, a series of experiments were conducted using a set of eight 512×512 images obtained from Fig. 4. Each image underwent the embedding a size of 64×64 of watermark logo as illustrated in Fig. 5.

First, let's introduce the formula for *BER* as follows:

$$BER = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N} e_{x,y}}{MxN} . \quad (11)$$

Where W and W' represent the values of the original watermark data and the extracted watermark data, respectively, at the position (x, y). The calculation involves the length of the binary watermark data, denoted as $M \times N$. $e_{x,y}$ represents an error occurring at position (x, y) in the watermark W and W', summing up $e_{x,y}$ and dividing it by the size of the watermark gives the measured watermark bit error rate. The BER value is used to measure the dissimilarity between the extracted and original watermark data, ranging from 0 to 1. A smaller BER value indicates a closer resemblance between the extracted watermark data and the original watermark data.

We will compare NC with other recent zero-watermarking methods. The evaluation method for *NC* is described by the following formula [32]:

$$NC = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left(W_{x,y}\right)\left(W'_{x,y}\right)}{\sqrt{\sum_{x=1}^{M}\sum_{y=1}^{N}(W_{x,y})^2 \times \sum_{x=1}^{M}\sum_{y=1}^{N}(W'_{x,y})^2}} . \quad (12)$$

Where W and W' denote the values of the original watermark data and the extracted watermark data respectively, at the specific position (x, y). To evaluate the similarity, we compute the ratio of the total number of matching bits. The resulting *NC* value falls within the range of 0 to 1, with a higher value indicating a stronger resemblance between the two datasets.

Table 3. Comparison of BER experimental results under 12 types of noise attacks

Attacks		Airplane	Baboon	Barbara	Cameraman	Fishing boat	Gold hill	Lena	Peppers
N1	PSNR	28.126	28.143	28.171	28.287	28.133	28.134	28.122	28.220
	BER	0.	0.0014	0.0019	0.0036	0.	0.	0.	0.
N2	PSNR	10.975	11.438	10.690	10.919	11.376	11.175	11.137	10.861
	BER	0.0039	0.0056	0.0046	0.0080	0.0034	0.0083	0.0058	**0.0097**
N3	PSNR	27.571	30.499	23.377	29.615	29.736	27.750	31.275	31.113
	BER	0.0026	0.	0.0043	0.0058	0.	0.	0.	0.
N4	PSNR	37.332	36.517	39.089	36.713	39.016	37.791	41.434	36.660
	BER	0.	0.	0.0021	0.0034	0.	0.	0.	0.
N5	PSNR	34.302	26.870	30.943	31.605	33.487	32.718	36.457	31.016
	BER	0.0002	0.	0.0021	0.0037	0.	0.	0.	0.
N6	PSNR	29.293	22.525	24.386	29.301	28.133	28.316	30.820	28.197
	BER	0.	0.0014	0.0019	0.0053	0.	0.	0.	0.0026
N7	PSNR	8.3186	10.380	8.1641	9.1991	9.6926	10.212	10.955	8.7544
	BER	0.	0.	0.0021	0.	0.	0.	0.	0.
N8	PSNR	11.817	12.633	8.0253	9.8846	11.731	10.640	11.336	9.8004
	BER	0.	0.0014	0.0034	0.0048	0.	0.	0.	0.0026
N9	PSNR	11.817	12.633	8.0253	9.8846	11.731	10.640	11.336	9.8004
	BER	0.0012	0.0029	0.0034	0.0057	0.	0.	0.	0.0034
N10	PSNR	19.083	16.640	16.271	20.622	19.641	20.698	21.626	20.032
	BER	0.	0.	0.0021	0.	0.	0.	0.	0.
N11	PSNR	8.3345	10.317	8.1849	9.1752	9.6778	10.202	11.003	8.7685
	BER	0.0002	0.0019	0.0024	0.0031	0.	0.	0.	0.
N12	PSNR	11.817	12.633	8.0253	9.8846	11.731	10.640	11.336	9.8004
	BER	0.0002	0.	0.0026	0.0031	0.0004	0.	0.	0.

Our Experimental Result. Table 3 displays experimental results for the eight images from Fig. 4, each subjected to 12 different noise attacks (N1 to N12). The table includes PSNR values, which indicate the PSNR of each image after a specific noise attack compared to its original state. We also assessed the extracted watermarks using BER method, comparing the results to the original binary watermark (Fig. 5).

In our experiments, we conducted a total of $8 \times 12 = 96$ watermark embedding and extraction tests. The highest *BER* measurement, indicating the most significant impact, was observed for the image, Peppers, under the N2 attack (salt & pepper noise), with a *BER* measurement of 0.0097. Remarkably, this still represents an influence of less than 1%. Notably, when examining the entire set of 96 measurements, more than half of the *BER* values were equal to or very close to 0. Lower *BER* values, closer to zero, indicate resistance to noise interference and the ability to maintain a complete and robust watermark. In this experiment, we observed that our method exhibited consistent robustness under both general attacks (N1 to N6) and geometric attacks (N7 to N12), demonstrating its resilience across both types of attacks.

Next, our proposed method will be compared to the methods proposed by Xing et al. [12], Huang et al. [13], and Liu et al. [14]. All of methods are under the same zero-watermark framework but with different embedding algorithms, we will utilize NC as the evaluation metric and comparing. We compare experimental data based on average watermark NC values extracted from each image under different noise attacks in our experimental setup. This data is used as the benchmark for our analysis.

Comparing [12] on the NC Metric. Table 4 presents the results of comparisons with the zero-watermarking framework proposed by Xing et al. [12]. Our method achieves an NC value close to 1 under Gaussian attacks with $\sigma = 0.005$. Moreover, under Gaussian attacks with $\sigma = 0.025$, there is a significant difference between the two methods (0.9948-0.8084=0.1864). Across various noise attack intensities in this experiment, our method consistently exhibits superior performance, with NC values consistently surpassing this threshold. Notably, our method maintains robustness against both general and geometric attacks without bias.

Table 4. Compare [12] the average NC value under different attacks

Attacks intensity	Method in [12]	Proposed
Gaussian ($\sigma = 0.005$)	0.9254	0.9999
Gaussian ($\sigma = 0.015$)	0.8492	0.9952
Gaussian ($\sigma = 0.025$)	0.8084	0.9948
Salt&Pepper ($\sigma = 0.01$)	0.9494	0.9971
Salt&Pepper ($\sigma = 0.05$)	0.8614	0.9950
Salt&Pepper ($\sigma = 0.12$)	0.8186	0.9930
Median filtering (3x3)	0.9870	0.9974
Median filtering (7x7)	0.9750	0.9965
Median filtering (9x9)	0.9706	0.9957
Average filtering (3x3)	0.9790	0.9996
Average filtering (7x7)	0.9614	0.9995
Average filtering (9x9)	0.9548	0.9995
Rotation Clockwise (angle=5°)	0.8480	0.9998
Rotation Clockwise (angle=10°)	0.8300	0.9996
Rotation Counter Clockwise (angle=5°)	0.8820	0.9993
Rotation Counter Clockwise (angle=10°)	0.8530	0.9988

Comparing [13] on the NC Metric. Table 5 presents the results of comparisons with the zero-watermarking framework proposed by Huang et al. [13]. The compared methods consistently achieve NC values of at least 0.9657 or higher under various attacks. In contrast, our proposed method consistently achieves NC values of at least 0.9957 or higher. The difference between the two methods is most pronounced in the lowest data point, with a difference of 0.03 (0.9957-0.9657). Across various noise attack intensities in this experiment, our method maintains the stability of NC values starting with 0.99. In contrast, the compared method exhibits instability within NC values starting with 0.96, 0.97, 0.98, and 0.99. Overall, our proposed method demonstrates stability and robustness compared to this method.

Table 5. Compare [13] the average NC value under different attacks

Attacks intensity	Method in [13]	Proposed
Gaussian (σ = 0.05)	0.9828	0.9959
Gaussian (σ = 0.25)	0.9657	0.9963
Gaussian (σ = 0.5)	0.9688	0.9963
JPEG Compress (Q = 2%)	0.9812	0.9957
JPEG Compress (Q = 10%)	0.9952	0.9980
JPEG Compress (Q = 30%)	0.9998	0.9999
Median filtering (3x3)	0.9928	0.9974
Median filtering (5x5)	0.9789	0.9959
Median filtering (7x7)	0.9673	0.9965
Rotation Clockwise (angle=5%)	0.9890	0.9990
Rotation Clockwise (angle=20%)	0.9704	0.9979
Rotation Clockwise (angle=35%)	0.9734	0.9975

Comparing [14] on the NC Metric. Table 6 presents the results of comparisons with the zero-watermarking framework proposed by Liu et al. [14]. The compared methods were tested under predominantly geometric attacks. It can be observed in the table that, under the significant Downshift attack (move=20%), our data, while not maintaining a performance above 0.99, shows a considerable difference of 0.4814 (0.9786-0.4800) compared to the method being compared. Although the compared method achieves excellent NC values of 1 under Scaling attack (factor=1.5) and Downshift attack (move=2%), the data for this method is unstable, ranging between 0.4 and 1.0, for other attack scenarios. In contrast, our proposed method consistently demonstrates stability and robustness with values ranging from 0.97 to 0.99, all starting with 0.9 or higher.

In our experiments and comparisons with recent zero-watermarking methods. Our proposed watermark extraction framework consistently shows exceptional stability, reliability, and robustness. It performs well under both general noise attacks and geometric noise attacks. We assessed its performance using both the BER method and the NC method to measure experimental data. These evaluations offer compelling evidence that supports the effectiveness and applicability of our advanced zero-watermarking approach. In conclusion, our experimental results strongly affirm the superiority of our proposed framework.

Table 6. Compare [14] the average NC value under different attacks

Attacks intensity	Method in [14]	Proposed
Rotation Clockwise (angle=3°)	0.8750	0.9998
Rotation Clockwise (angle=5°)	0.8750	0.9998
Rotation Clockwise (angle=9°)	0.8200	0.9997
Rotation Clockwise (angle=11°)	0.7800	0.9992
Scaling attack (factor=0.5)	0.8050	0.9981
Scaling attack (factor=0.7)	0.9350	0.9971
Scaling attack (factor=1.5)	1.	0.9979
Scaling attack (factor=2)	0.9800	0.9981
Downshift attack (move=2%)	1.	0.9950
Downshift attack (move=6%)	0.9650	0.9897
Downshift attack (move=10%)	0.7450	0.9856
Downshift attack (move=20%)	0.4800	0.9786

5 Conclusion

Our proposed method is built upon recent advances in zero-watermarking frameworks widely used for watermark embedding. These frameworks offer the advantage of unrestricted watermark capacity and non-destructive effects on the original image.

Building on these foundational advantages, we introduce a novel feature extraction method based on circular area extraction. This method leverages radius-related ranges to extract robust features from the original image.

Additionally, we employ a 1-D NRDPWT to transform these features into phi, further enhancing their stability. Through rigorous experiments involving various levels of noise attacks, encompassing different types and strengths of noise, our method consistently demonstrates excellent and robust performance in terms of BER and NC measures. This stability is maintained under both general noise and geometric noise attacks. Comparative analysis with recent methods reinforces the exceptional stability, reliability, and robustness of our watermark embedding. Addressing security concerns, we employ variations in the Fisher-Yates Shuffle and the use of PRNG, enhancing the security of our approach.

In summary, our proposed zero-watermarking method offers (a) unlimited capacity, (b) preservation of original image integrity, (c) strong key management and reliability, (d) stability, and (e) security.

Our future research aims to enhance the scheme's resistance to combined attacks and explore new technologies to bolster its defense mechanisms. Furthermore, we plan to extend the application of this scheme to the field of information hiding, opening up new possibilities for secure data protection.

6 Acknowledgement

The authors express their gratitude for the guidance provided by anonymous reviewers of this article.

References

[1] U. Khadam, M.M. Iqbal, M. Alruily, M.A. Al Ghamdi, M. Ramzan, S.H. Almotiri, Text Data Security and Privacy in the Internet of Things: Threats, Challenges, and Future Directions, Wireless Communications and Mobile Computing 2020(2020) 1–15.
[2] P.C. van Oorschot, S.W. Smith, The Internet of Things: Security Challenges, IEEE Security & Privacy 17(5)(2019) 7–9.
[3] M. Begum, M.S. Uddin, Digital Image Watermarking Techniques: A Review, Information 11(2)(2020) 110.
[4] D.K. Mahto, A.K. Singh, A survey of color image watermarking: State-of-the-art and research directions, Computers & Electrical Engineering 93(2021) 107255.
[5] A.A. Mohammed, D.A. Salih, A.M. Saeed, M.Q. Kheder, An imperceptible semi-blind image watermarking scheme in DWT-SVD domain using a zigzag embedding technique, Multimedia Tools and Applications 79(43–44)(2020) 32095–32118.
[6] H. Cao, H. Xiang, X. Li, M. Liu, Y. Sangbong, W. Fang, A zero-watermarking algorithm based on DWT and chaotic modulation, In Independent Component Analyses, Wavelets, Unsupervised Smart Sensors, and Neural Networks IV 6247(2006) 420–428.
[7] Z. Dai, F. Hong, G. Cui, M. Fu, Watermarking text document based on statistic property of part of speech string, JOURNAL-CHINA INSTITUTE OF COMMUNICATIONS 28(4)(2007) 108.
[8] Q. Wen, T.-F. Sun, S.-X. Wang, Concept and application of zerowatermark, Acta electronica sinica 31(2)(2003) 214–216.
[9] X. Liao, Neural-network-based zero-watermark scheme for digital images, Optical Engineering 45(9)(2006) 097006.
[10] X. Leng, J. Xiao, Y. Wang, A Robust Image ZeroWatermarking Algorithm Based on DWT and PCA, in: Proc. First International Conference Communcations and Information Processing, ICCP Part II, 2012.
[11] S. Lin, X. Jiucheng, Z. Xingxing, D. Wan, T. Yun, A novel Generalized Arnold Transform-based Zero-Watermarking Scheme, Applied Mathematics and Information Sciences 9(4)(2015) 2023–2035.
[12] S.M. Xing, Y.L. Tong, J.G. Liang, A Zero-Watermark Hybrid Algorithm for Remote Sensing Images Based on DCT and DFT, Journal of physics 1952(2)(2021) 022049–022049.
[13] W. Liu, J. Li, C. Shao, J. Ma, M. Huang, U.A. Bhatti, Robust zero watermarking algorithm for medical images using local binary pattern and discrete cosine transform, in: Proc. international conference on artificial intelligence and security , 2022.
[14] T. Huang, J. Xu, S. Tu, B. Han, Robust zero-watermarking scheme based on a depthwise overparameterized VGG network in healthcare information security, Biomedical Signal Processing and Control 81(2023) 104478.
[15] A. Mudhafar, A. Rusul, K. Nidhal, El Abbadi, Noise in Digital Image Processing: A Review Study, in: Proc. 2022 3rd Information Technology To Enhance e-learning and Other Application (IT-ELA) (2022) 79–84.
[16] C.P. Dautov, M.S. Mehmet, Wavelet transform and signal denoising using Wavelet method, in: Proc. 2018 26th Signal Processing and Communications Applications Conference (SIU) (2018) 1–4.
[17] I.L. Cascio, Wavelet analysis and denoising: New tools for economists, Queen Mary Uniiversity of London (600)(2007) 1–7.
[18] W. Li, Wavelets for Electrocardiogram: Overview and Taxonomy, IEEE Access 7(2019) 25627–25649.
[19] X. Ge, G. De Stefano, M.Y. Hussaini, O.V. Vasilyev, Wavelet-Based Adaptive Eddy-Resolving Methods for Modeling

and Simulation of Complex Wall-Bounded Compressible Turbulent Flows, Fluids 6(9)(20121) 331.
[20] G. Baldazzi, E. Sulas, E. Brungiu, M. Urru, R. Tumbarello, L. Raffo, D. Pani, Wavelet-based post-processing methods for the enhancement of non-invasive fetal ECG, in: Proc. 2019 Computing in Cardiology (CinC) (2019) 1–4.
[21] W. Wang, G. Zhang, L. Yang, V.S. Balaji, V. Elamaran, N. Arunkumar, Revisiting signal processing with spectrogram analysis on EEG, ECG and speech signals, Future Generation Computer Systems 98(2019) 227–232.
[22] A. Gómez-Echavarría, J.P. Ugarte, C. Tobón, The fractional Fourier transform as a biomedical signal and image processing tool: A review, Biocybernetics and Biomedical Engineering 40(3)(2020) 1081–1093.
[23] S.M. Pourhashemi, M. Mosleh, Y. Erfani, A novel audio watermarking scheme using ensemble-based watermark detector and discrete wavelet transform, Neural Computing and Applications 33(11)(2020) 6161–6181.
[24] C.J. Lin, H. Chuang, C.W. Hsu, C.S. Chen, Pneumatic artificial muscle actuated robot for lower limb rehabilitation triggered by electromyography signals using discrete wavelet transformation and support vector machines, Sens. Mater 29(2017) 1625–1636.
[25] S. Kamatchi, M. Sundararajan, Diagnosing Sinusitis using Fractional B-spline Wavelet with Near Infrared Spectroscopy, Biomedical and Pharmacology Journal 10(1)(2017) 95–103.
[26] A. Silik, M. Noori, W.A. Altabey, R. Ghiasi, Selecting optimum levels of wavelet multi-resolution analysis for time-varying signals in structural health monitoring, Structural Control and Health Monitoring 28(8)(2021) e2762.
[27] K. Fukumori, H.T. Nguyen, N. Yoshida, T. Tanaka, Fully data-driven convolutional filters with deep learning models for epileptic spike detection. in: Proc. 2019 IEEE international conference on acoustics, speech and signal processing (ICASSP) (2019) 2772–2776.
[28] S. Chandra, A. Sharma, G.K. Singh, A comparative analysis of performance of several wavelet based ECG data compression methodologies, Irbm 42(4)(2021) 227–244.
[29] R.C. Lee, K.C. Hung, New Modified SPIHT Algorithm for Data Compression System, Journal of Medical and Biological Engineering 39(1)(2018) 18–26.
[30] T.K. Hazra, G. Ghosh, S. Kumar, S. Dutta, A.K. Chakraborty, File encryption using fisher-yates shuffle, in: Proc. 2015 International Conference and Workshop on Computing and Communication (IEMCON) (2015).
[31] F.S. Tahir, A.A. Abdulrahman, The effectiveness of the Hermite wavelet discrete filter technique in modify a convolutional neural network for person identification, The Indonesian Journal of Electrical Engineering and Computer Science (IJEECS) 31(2023) 290–298.
[32] R. Sinhal, I.A. Ansari, Machine learning based multipurpose medical image watermarking, Neural Computing and Applications (2023) 1–22.