Int. J. Advance Soft Compu. Appl, Vol. 18, No. 1, March 2026
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Enhanced Data Embedding Using Adaptive Neural Networks with Modified Whale Optimization Algorithm

Nameer N. El. Emam[1], and Kefaya Qaddoum[2]

[1]Department of Computer Science, College of Information Technology, Amman Arab University, Amman, Jordan
e-mail: n.emam@aau.edu.jo
[2]Department of Computer Science and Software Engineering, Concordia University, Canada
e-mail: kefaya.qaddoum@concordia.ca

Abstract

This research paper introduces a new steganography algorithm for embedding large amounts of secret messages within a color image. Five security levels are incorporated to ensure reliable protection. To embed data in a randomized way, the algorithm uses a segmentation technique called New Adaptive Image Segmentation (NAIS). By analyzing the properties of each byte, this method determines the appropriate size of secret data to replace each byte and color in the image. Additionally, the algorithm features a machine-learning component inspired by an Adaptive Neural Network (ANN) combined with a modified version of the Whale Optimization Algorithm (MWOA). The findings show that strong imperceptibility is achieved with the stego-image, even with a large payload, reaching four bits per byte (4-bpb) at specific bytes. When the machine learning model ANN_MWOA is used, the highest PSNR reaches 79.58dB for the Baboon color image with a payload of (16384) bits, while PSNR decreases by 1% when applying ANN_WOA. Moreover, the proposed method outperforms previous approaches by an average of 2%. Additional metrics (MSE, SNR, Euclidean Norm, and others) are used to confirm that the proposed algorithm efficiently embeds hidden data.

Keyword: *Image segmentation, data embedding, payload capacity, neural network, Whale Optimization Algorithm.*

1. Introduction

Organizations continue to have serious concerns about protecting sensitive data, which motivates a lot of information security research. Cryptographic methods have long been employed to ensure the secure and reliable transmission of data [1]. However, because encrypted data is so apparent, malevolent attackers frequently notice it. By subtly incorporating private information into cover media, have created a workable solution to this problem [1] . Unlike cryptography, steganography's embedding property ensures that the hidden information is undetectable to possible eavesdroppers [2]. This exceptional feature makes steganography a vital tool for covert communication, protecting data privacy from unauthorized surveillance and safeguarding intellectual property from

Received 9 October 2025; Accepted 8 December 2025

unauthorized reproduction. Key applications of steganography include secret messaging, fingerprinting, digital signatures, and access control mechanisms [2-4].

Three principal attributes define the effectiveness of image steganography techniques: capacity, imperceptibility, and robustness. These attributes form the "magic triangle" [5-6]. The quantity of information that can be hidden in a cover image (CI) is known as capacity. Robustness measures the resistance to attacks or manipulations by adversaries, and imperceptibility evaluates the stego image's visual quality (SI) through metrics such as the Peak Signal-to-Noise Ratio (PSNR).

This research study proposes a technique that incorporates five security levels to protect hidden information and enhance resistance to steganalysis. We have integrated cryptographic techniques with steganography within a single system to securely transmit data over insecure channels.

1.1 Background

The section provides context for the topic by summarizing existing knowledge, explaining how the proposed study relates to previous research, and justifying the research significance.

1.1.1 Research problem

Due to concerns about the insensitivity and load capacity, it is difficult to include large amounts of secret data. Increased bit density increases detection risk and causes distortion. Traditional LSBs, adaptive LSBs, and neural encoders are designed to maximize load while maintaining visual quality, but when capacity increases, they are often undetectable [2]. Modern ML-based steganalysis can detect embedded traces, necessitating the reduction of statistical fingerprints [3]. Deep models can be accurately decoded but remain at risk of detection and removal. Optimization methods, including WOA variants, target pixel selection or distortion metrics, but do not explicitly optimize for robustness against steganalysis [4]. WOA and similar methods often suffer from early convergence and sensitivity issues, and current research mainly adjusts fitness functions toward visual metrics, while ignoring combined optimization for detectability and robustness.

1.1.2 Theoretical Significance

The theoretical significance of the proposed approach is that integration contributes to theory in several key ways:

- Previous methods optimized embedding parameters using either gradient-based learning for neural steganography or metaheuristics for adaptive LSB. The hybrid model links these, allowing the neural network to adapt content-wise while MWOA searches globally over configurations, avoiding local minima [5].
- It treats data embedding as a multi-objective optimization balancing imperceptibility, payload, and robustness. Unlike fixed-weighted-sum functions, the modified WOA adjusts objectives via neural feedback, approximating Pareto-optimal solutions and demonstrating how global algorithms can work with hybrid machine learning.
- The neural network learns high-level perceptual features of images, such that edges and textures, enabling content-aware embedding. It advances traditional LSB or transform models by adding semantic adaptivity, moving

toward perceptual embedding, and adopting machine learning-guided embedding.

- Improving WOA with adaptive coefficients, randomization, and mutations boosts convergence and exploration–exploitation balance, aiding in solving complex, high-dimensional optimization problems.

1.1.3 Practical Significance

From a practical standpoint, the proposed model addresses pressing limitations in digital data security, authentication, and copyright protection through several impacts:

- The adaptive neural encoder learns distortion-minimized features. At the same time, the modified WOA selects embedding regions optimized for invisibility and resilience, even when traditional methods fail.
- Combining the embedding model with statistical loss functions reduces detectability for secure communication and digital forensics.
- The modified WOA offers faster convergence and lower computational costs than GA or PSO for tuning embedding positions. Neural networks adaptively reduce training time for large images or videos, enabling near real-time operation.
- This hybrid approach extends beyond image steganography to medical data security, multimedia watermarking, IoT security, and blockchain validation, showing versatile applications in secure data transmission.
- The proposed hybrid model ANN_MWOA is adaptable to various cover media and can re-optimize based on changing image statistics, making it robust for cloud and distributed multimedia systems.

1.2 Contributions of the Proposed Work

- The proposed work's main contributions include:
- Modifying the least significant bits (LSBs) of CI's pixels, making this method more effective than traditional techniques by counting bits to hide (Nbplb) at each lower byte based on the surrounding eight high bytes (SEHB).
- Randomly embedding the encoded Smsg with new adaptive image segmentation (NAIS), which uses non-uniform segment sizes.
- Developing a hybrid machine-learning model (ANN_MWO) that combines the Whale Optimization Algorithm (MWOA) with adaptive neural networks to smooth surrounding bytes and defend against WFLogSv and chi-square attacks.
- Assessing the algorithm's performance with various metrics to verify security.

The structure of this manuscript is as follows: Section 2 reviews related work on the proposed steganography. Section 3 describes the data embedding and extraction model and formalizes the key functions it uses. Section 4 details the steps of the embedding and extraction algorithms, including five security levels. Section 5 discusses adaptive neural networks as the foundation of the learning system, along with (MWOA), featuring uniform adaptive relaxation. Section 6 presents the results and includes a discussion. Finally, Section 7 concludes with a summary of the main contributions.

2. Related Work

Image steganography methods are broadly divided into frequency-domain and spatial-domain methods. Techniques in the spatial domain directly embed secret information into image pixel intensities. In contrast, frequency domain techniques transform the CI into frequency coefficients before embedding the data. Least Significant Bit Matching (LSBM) is a simple steganography approach that has been detected under multiple attacks. Imperceptibility (i.e., maintenance of high perceptual image quality) and security are major parameters in steganography. However, most conventional steganography techniques rely to produce stego images that can be transmitted to recipients without detection by potential attackers, thereby ensuring secure communication channels and employs a multi-level randomization technique to embed data within randomly selected cover images, with each byte of the secret image distributed across multiple cover images [7]. In contrast, frequency-domain techniques, such as the Discrete Wavelet Transform (DWT) [5] and Very Large Scale Integration (VLSI) technologies [8], are designed to enhance resistance to these kinds of attacks.

In [1], an innovative image embedding system using the Invertible Rescaling Net (IRN) and the Similarity of Bits Pairs (SBP) was discussed. This approach leverages global statistical features of the CI to guide data embedding, often targeting areas with higher texture variability while avoiding smooth regions. Using edge areas for embedding improves the visual quality of stego images because changes have a less significant impact on the human visual system in edge regions than in smooth areas. Consequently, embedding information in edge locations significantly improves the imperceptibility of stego images; however, the size of the hidden secret data is limited to two levels of security.

As shown in [9], edge detection plays a key role in image steganography, and data is embedded in edge regions to ensure reliability and adaptability and maintain perception quality. The edge is marked by sudden changes in pixel intensity and detected by fuzzy logic such as Sobel, Robert, Laplacian, Prewitt, and Canny filters. Techniques of strange and fuzzy logic are popular in research. The Sobel operator was used to detect edge regions in a CI's single channel (R, G, or B), choosing areas with high intensity gradients. However, their method had limited payload capacity, lacked adaptive bit selection for embedding Smsg at any location within the CI, and offered only two security levels, making it vulnerable to attacks.

In [10], a thorough overview of deep learning-based image steganography methods is provided, emphasizing recent traditional methods and their evolution. They investigated cutting-edge deep learning-based image steganography techniques, enhancing payload capacity and improving imperceptibility. However, the proposed mechanism suffers from computational complexity in both preprocessing the dataset and the training procedure.

RoSteALS was introduced in [11] as a practical steganography technique that leverages pre-trained autoencoders. RoSteALS features a lightweight secret encoder with approximately 300k parameters, making training easier. It achieves perfect secret recovery and comparable image quality across three benchmarks. RoSteALS can be adapted for coverless steganography, with the CI generated from noise or text prompts via a denoising diffusion process. While RoSteALS delivers excellent results in controlled conditions, its scalability to high-resolution images and real-world applications remains to be explored. Additionally, it is necessary to enhance its resilience against adaptive steganalysis methods, especially in dynamic environments.

The use of implicit neuronal representation (INR)in a deep cross-modal steganography framework allows Smsg to be embedded in various formats, as shown in [12]. This framework uses INRs to represent confidential data and supports a number of options and solutions. Experiments have demonstrated the flexibility of this approach and the ability to handle various data formats.
In [13], a deep network steganography technique for secretly communicating deep neural network (DNN) models was introduced. This technique, unlike conventional steganography, embeds the learning task of a secret deep neural network (DNN) model as an ordinary learning task within a stego DNN model. The stego model is created by embedding interference filters into key locations within the secret deep neural network (DNN) model using a gradient-based filter insertion scheme. However, this technique includes two security layers that are easy to breach, allowing access to the Smsg.

3. The Proposed Embedding/Extraction Algorithm with Five Levels of Security

The workflow diagram Fig. 1a illustrates the process sequence in the proposed data concealing and extraction framework.

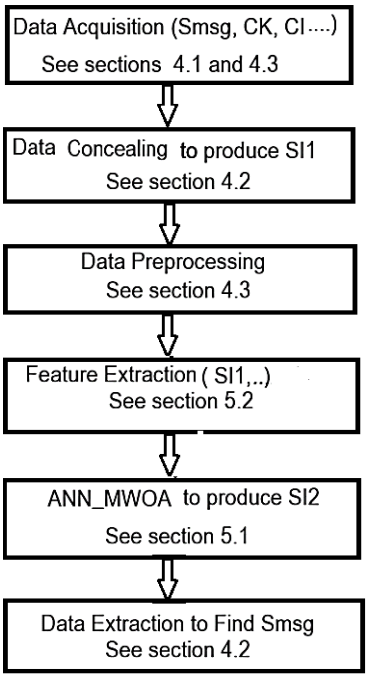

Fig. 1-a: The general workflow of the proposed study

Each block in the workflow diagram represents a key stage, and arrows indicate data flow among algorithms. Fig.1a shows how components from Sections 4 and 5 integrate. The first stage collects input data: sensitive information, a cover image, and a cipher key (CK). The second stage embeds secret data into the cover image (CI) to create SI1, the initial stego-image. The third stage involves organizing and cleaning the data for consistency before applying machine learning or optimization. The fourth stage extracts features from SI1 and from parameters such as NbpIb. These features are combined with an Adaptive Neural Network (ANN) and the Modified Whale Optimization Algorithm (MWOA) in the fifth stage to enhance metrics such as imperceptibility and capacity,

thereby optimizing SI2. The final stage uses SI2 and keys to recover the hidden message Smsg.

The proposed embedding algorithm aims to securely embed a large Smsg while effectively resisting statistical and visual attacks. It is organized into five distinct security levels, which are illustrated in Fig. 1b for clarity.

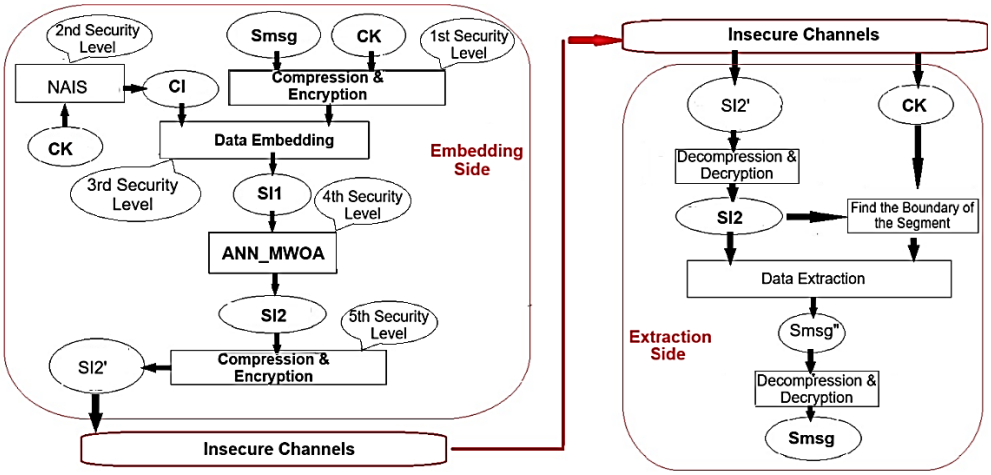

Fig. 1-b: The proposed embedding/extraction architecture with five security levels

At the first and fifth levels, well-established techniques are used to compress and encrypt both the Smsg and the CI [14]. These basic levels form the foundation of the algorithm, providing robust security. The middle three levels introduce significant modifications to improve the overall steganography process. In the second and third levels, advanced image segmentation and embedding techniques are implemented. These approaches strategically allow for the random embedding of the Smsg within the CI, moving away from traditional sequential methods. This randomness enhances security by making it harder for potential attackers to detect hidden information. At the fourth security level, the proposed machine learning model serves as an effective strategy to cut down the number of training iterations. Here, a metaheuristic algorithm based on adaptive neural networks (ANNs) is combined with an optimization algorithm (MWOA). This combination allows the algorithm to precisely adjust pixel values, making visual detection of the Stego image very difficult. This method not only maintains the visual quality of the CI ,but also ensures that the embedded Smsg is protected from various analysis attacks. The fifth level involves applying compression and encryption to the Stego image.

It is crucial to outline the key stipulations of the proposed new data-embedding algorithm as follows:

Definition 3.1 In the first and the fifth levels of security, we applied lossless data compression $C^{x,y}_{SPIHT}: x \rightarrow y$ and decompression $DC^{y,x}_{SPIHT}: y \rightarrow x$ functions that are based on SPIHT (Set Partitioning in Hierarchical Trees) algorithm. These functions are applied explicitly to a Smsg, as well as a stego image SI. The variable x signifies the original form of Smsg before undergoing the compression process, or it may represent a secondary variant of the stego image prior to any compression and subsequent smoothing, denoted as $SI2'$. On the other hand, variable y represents the transformation state of the Smsg after compression or the set of updated SI bytes after compression and smoothing. The main purpose of using these compression and decompression functions

is to reduce the file size and make them more efficient for transmission and storage. This approach not only facilitates more effective information exchange but also enhances security in line with the objectives of the first and fifth levels of the security protocol.

Definition 3.2: At the first and fifth levels of security, we applied data encryption and decryption. Let the mapping function define the process of encryption, denoted as $E_{AES}^{x,y}: x \rightarrow y$, while the corresponding inverse mapping function defines the decryption process, represented as, where these two functions are inspired by the principles of the Advanced Encryption Standard (AES) technique [8]. To ensure data transmission security, the AES algorithm is used for encryption and decryption. The variable x represents plaintext data, before encryption, while y represents the resulting ciphertext after encryption. These algorithms form the first security layer in the proposed data embedding and abstraction method, safeguarding sensitive information and maintaining its integrity. This dual function ensures confidentiality and enables secure data retrieval.

Definition 3.3 In the second level of security, we applied a new image segmentation function defined in the map $\Psi_{NAIS}^{x,y}: CI \times ck' \rightarrow y$ that is predicated upon two stages of new adaptive image segmentation (NAIS). In this context, y represents the information about the edges of the newly formed segments. The NAIS algorithm, as outlined in Section 4.1, has been designed to provide the second level of security (refer to Fig. 2). This particular level is intended to complicate the detection of segment edges by a steganalysis, thereby affording greater protection for the transmission of a confidential message compared to a single-level non-uniform segmentation [15].

Definition 3.4 In the third level of security, we define the function $H_{SEHB}^{x,y,z}$: $x \times Nbplb \times z \rightarrow y$ as a novel data embedding function using byte characteristics related to CI, to ascertain the number of bits of the lower nibble at the current byte (Nbplb) as stated in section (4.2). Here, x represents the CI, y represents the stego image (SI1) before data smoothing, which embeds the data, and z represents another variable. The $H_{SEHB}^{x,y,z}$, function provides an additional layer of security, as illustrated in Fig. 1. It specifies the embedded information using Nbplb at each byte to minimize noise in SI1. Byte characteristics were evaluated by comparing the current byte's high nibble with the eight surrounding high nibbles (SEHB).

Definition 3.5 In the fourth level of security, we applied the modified machine learning function in mapping $\xi_{ANN\_MWOA}^{SI1^{FB},SI2^{FB'}} : x \rightarrow y$; defined on a hybrid machine learning based on ANN_MWOA. The variable (x) signifies the free bits (FB) of the initial image (SI-1) from the embedding algorithm. At the same time (y) indicates the new free bits (FB') of the subsequent (SI-2), created after the learning system smooths the FB. This function ($\xi_{ANN\_MWOA}^{SI1^{FB},SI2^{FB'}}$) supports the fourth security level and generates SI-2, optimizing it for insecure channels against visual and statistical attacks [15].

Definition 3.6 Let the mapping function ($Ext_{SEHB}^{x,y} : x \times Nbplb \rightarrow y$) signify a modified abstraction mechanism to extract an encrypted and compacted Smsg presented by the script y. In contrast, the script x denotes SI2. This function focuses on evaluating the attributes of each byte in SI2 to facilitate the extraction of the encrypted message.

4. The Proposed Steganography Algorithm

The proposed algorithm includes three phases: adaptive segmentation, embedding Smsg, and extraction of Smsg according to the following:

4.1 New adaptive image segmentation algorithm (NAIS):

At the second level of security, the new adaptive image segmentation algorithm (NAIS) was improved by adding random data, which further complicated edge detection through the non-uniform segmentation [15]. NAIS uses irregular separation through diagonal technology. The steps for the algorithm are outlined as follows:

Algorithm 1: NAIS algorithm

Step 1: Assume the cipher-key (ck) length is represented by L. //see equation (1).

$$L = |ck| \quad (1)$$

Step 2: Estimate the segment sizes in the vertical (v) and horizontal (h) directions using equations (2-3).

$$\Gamma_i^v = \left\lceil \frac{\Im(ck_i) \times h^{CI}}{\sum_{m=1}^{L} \Im(ck_m)} \right\rceil \quad \forall i = 1 \dots L \quad (2)$$

$$\Gamma_i^h = \left\lceil \frac{\Im(ck_i) \times w^{CI}}{\sum_{m=1}^{L} \Im(ck_m)} \right\rceil \quad \forall i = 1 \dots L \quad (3)$$

Where $\Im(ck_i)$ signifies the numerical value of the $(\mathrm{i}^{\mathrm{th}})$ character in the ck, where (h^{CI}) and (w^{CI}) signify the height and the width of the CI, respectively.

Step 3: Call Segment_Edges(.) function // (*see algorithm 2), to find the non-uniform segments' edges of the* $X_{\square}^{edge}$, $Y_{\square}^{edge}$ *at CI,*

Step 4: Apply row-wise scanning on non-uniform segments at CI.

Step 4-1: Insert a right or left diagonal to each segment according to the following conditions; see Fig. 2, side (b).

If $\Im(ck_i)$ is an odd number, then

Add the left diagonal to segment (i);

Else

Add the right diagonal to segment (i);

Step 5: Store x, y coordinates of each edge at each segment S, $e_j^s(x, y), \forall j = 1, \dots, Nedge_s$,

where $Nedge_s$ is the count of edges at the S^{th}segment.

Step 6: End.

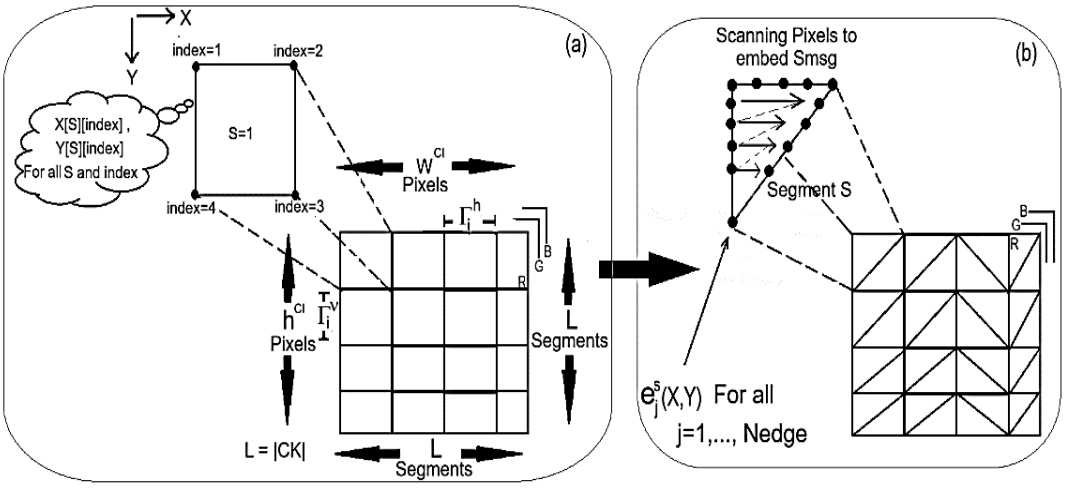

Fig. 2: New adaptive image segmentation NAIS

Algorithm 2: Segment Edges Function

Function Segment_Edges(**Image** C, int $X_{S,k}^{edge}$, int $Y_{S,2k}^{edge}$, int Γ_i^h, int Γ_i^v) { *// see Fig. 2*

Step 1: Set zero to the first elements of both x and y vectors:

$x_0 = 0;\ y_0 = 0;$

Step 2: Set the initial values to another element of the two vectors x and y:

While(i< L+1) {

$x_i = x_{i-1} + \Gamma_{i-1}^h$;

$y_i = y_{i-1} + \Gamma_{i-1}^v$;

} *// end of while i*

Step 3: Apply nested for to find $X_{S,k}^{edge}$ and $Y_{S,2k}^{edge}$ $\forall\ k = 1, \dots, 4$

S:=0;

While(j< L)

While(i< L) do { // find segments' edges

$X_{S,1}^{edge} = x_i$; $X_{S,2}^{edge} = x_{i+1}$;

$X_{S,3}^{edge} = x_{i+1}$; $X_{S,4}^{edge} = x_i$;

$Y_{S,1}^{edge} = y_j$; $Y_{S,2}^{edge} = y_j$;

$Y_{S,3}^{edge} = y_{j+1}$; $Y_{S,4}^{edge} = y_{j+1}$;

S= S + 1;

} *// end while i & while j*

Step 4: End. } *// end of function **Segment** Edges*

4.2 New data embedding and extraction phases

The new embedding algorithm comprises two phases: the embedding phase ($\xi_{ANN\_MWOA}^{SI1^{FB},SI2^{FB\prime}}$) is defined in Algorithm 3, and the abstraction phase $Ext_{SEHB}^{x,y}$ is defined in Algorithm 4. The embedding phase utilizes the proposed machine learning ANN_MWOA to achieve high undetectability in security applications.

Algorithm 3: The Embedding Phase ($\xi_{ANN\_MWOA}^{SI1^{FB},SI2^{FB\prime}}$)

Step 1: Input Smsg, ck, CI;

Step 2: Apply $E_{AES}^{ck,ck\prime}$; *// see definition 3.2*

Step 3: Apply $C_{SPIHT}^{Smsg,Smsg\prime}$; *// see definition 3.1*

Step 4: Apply $E_{AES}^{Smsg\prime,Smsg\prime\prime}$; *// see definition 3.2*

Step 5: Apply $\Psi_{NAIS}^{CI,ck\prime}$; *// see definition 3.3*

Step 6: Execute $H_{SEHB}^{CI,SI1,Smsg}$; *// see definition 3.4*

Step 7: Execute $\xi_{ANN\_MWOA}^{SI1^{FB},SI2^{FB\prime}}$; *// see definition 3.5*

Step 8: Apply $C_{SPIHT}^{SI2,SI2\prime}$; *// see definition 3.1*

Step 9: End.

Algorithm 4: The Extraction Phase $Ext_{SEHB}^{x,y}$

Step 1: Read ck, SI2'

Step 2: Execute $E_{AES}^{ck,ck\prime}$; *// see definition 3.2*

Step 3 Execute $DC_{SPIHT}^{SI2\prime,SI2}$; *// see definition 3.1*

Step 4: Execute $\Psi_{TLANUS}^{SI2,ck\prime}$; *// see definition 3.3*

Step 5: Execute $\mathrm{Ext}_{\mathrm{SEHB}}^{x,y}$; // *see definition 3.6*
Step 6: Execute $\mathrm{DC}_{\mathrm{AES}}^{\mathrm{Smsg}',\mathrm{Smsg}'}$;
Step 7: Execute $\mathrm{DC}_{\mathrm{SPIHT}}^{\mathrm{Smsg}',\mathrm{Smsg}}$;
Step 8: End.

4.3 Evaluation of the byte characteristics

A byte's characteristic evaluation is defined in ($\mathrm{Nbplb}_{i,j}$). It depends on how many secret bits should be in each color segment c to replace the current byte's particular low nibble. $\mathrm{Nbplb}_{i,j}$is calculated by Algorithm 5, using the surrounding eight bytes based on high nibbles ($\mathrm{SEHB}_{i,j}^{c}$) related to the present byte ($\mathrm{b}_{i,j}^{c}$) (see Fig. 3).

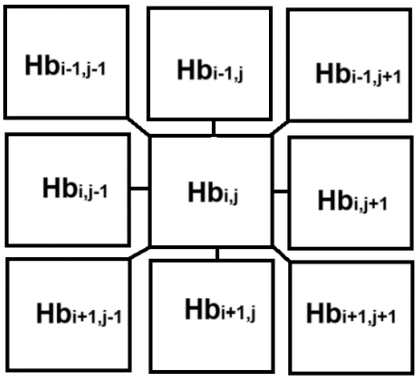

Fig. 3: The surrounding eight higher bits $\mathrm{SEHB}_{i,j}^{c}$ of $\mathrm{Hb}_{i,j}^{c}$

Algorithm 5: Calculate $\mathrm{Nbplb}_{i,j}$

Step1: Input color image I^C with the size $(n \times n)$, where c={R, G, B}; // *Three color components*
Step 2: Find the variance ($\sigma_{\mathrm{I}^c}^2$) for each color component of Image I^c as in equation(4):

$$\sigma_{\mathrm{I}^c}^2 = \frac{1}{n \times n} \sum_{s=1}^{n} \sum_{r=1}^{n} \left(\mathrm{Hb}_{r,s} - \mu_{\mathrm{I}^c}\right)^2 \quad (4)$$

and μ_{I^c} is the mean of the image I^c at the color c, such that:

$$\mu_{\mathrm{I}^c} = \frac{1}{n \times n} \sum_{s=1}^{n} \sum_{r=1}^{n} \mathrm{Hb}_{r,s} \quad (5)$$

Step 3: Find the variance of the pixel (i, j) concerning the surrounding eight high bytes ($\sigma_{\mathrm{SEHB}_{I,j}^{c}}^2$) for each color component as in equation (6): // *see Fig. 3*

$$\sigma_{\mathrm{SEHB}_{i,j}^{c}}^2 = \frac{1}{3 \times 3} \sum_{s=j-1}^{j+1} \sum_{r=i-1}^{i+1} \left(\mathrm{Hb}_{r,s} - \mu_{i,j}\right)^2 \quad (6)$$

and $\mu_{i,j}$ is the mean of the image I^C at the color c, such that:

$$\mu_{\mathrm{SEHB}_{i,j}^{c}} = \frac{1}{3 \times 3} \sum_{s=j-1}^{j+1} \sum_{r=i-1}^{i+1} \mathrm{Hb}_{r,s} \quad (7)$$

Step4: **Compute** the threshold (τ) according to equation (8)

$$\tau = \frac{\sigma_{\mathrm{SEHB}_{i,j}^{c}}^2}{\sigma_{\mathrm{I}^c}^2} \quad (8)$$

Step 5: If ($\tau > 1$) then $\mathrm{Nbplb}_{i,j} = 4$
else if ($\tau > 0.5$) then $\mathrm{Nbplb}_{i,j} = 2$
else $\mathrm{Nbplb}_{i,j} = 1$
Step 6: End.

This study makes the Smsg undetectable when Nbplb is accurately calculated according to Algorithm 5. In Fig. 4, we consider a (3×3) window selected from the color image. We demonstrate the process of identifying the number of secret bits of Nbplb that need to be substituted in the current byte's low nibble. This process is implemented on the blue component (c=3).

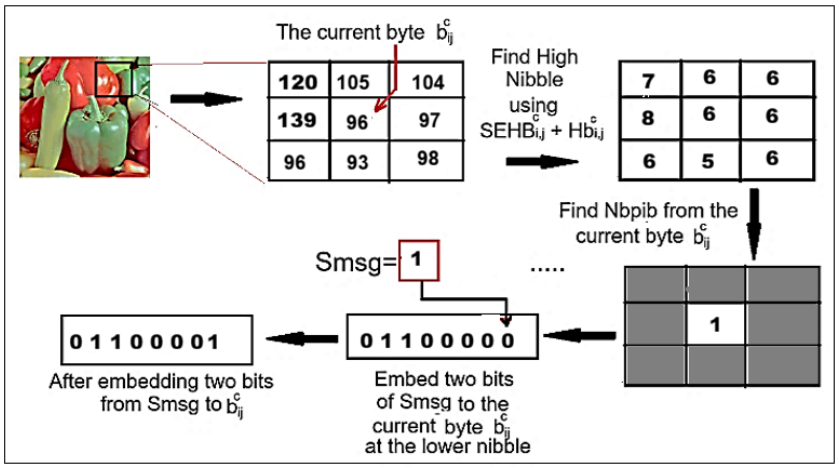

Fig. 4: The steps of embedding secret data into a color image

5. The Proposed Machine Learning System

This We proposed a new model, named (ξANN_MWOASI1FB, SI2FB´), which is based on ANN_MWOA to minimize changes in both CI and the modified stego image SI2 [16].

5.1 Applied ANN_MWOA algorithm

The choice of an Adaptive Neural Network (ANN) is motivated by its ability to model complex, nonlinear relationships while dynamically adjusting its internal parameters to optimize performance [17]. Traditional NN with fixed architectures often slow down convergence and risk being stuck in local minima. The adaptive design offers greater flexibility, enabling the network to modify its learning behavior based on feedback from the optimization process. This adaptability is further enhanced by integrating the Modified Whale Optimization Algorithm (MWOA), which enables efficient global search and parameter tuning. Consequently, the proposed adaptive ANN demonstrates faster convergence, improved stability, and better embedding accuracy compared to conventional neural models, as detailed in Section 6.

The learning model, ANN_MWOA, utilizes adjustments on the FB to generate a new FB'. This approach is designed to enhance security levels, accelerate training, and reduce the likelihood of undetectable changes (refer to sections 4.2-4.3). The proposed three-layer adaptive neural network (ANN) features a Perceptron architecture with an (n'-p-n) configuration, consisting of n' neurons in the first layer, p neurons in the second layer, and n output neurons in the third layer, along with a fully connected layer (see Fig. 5).

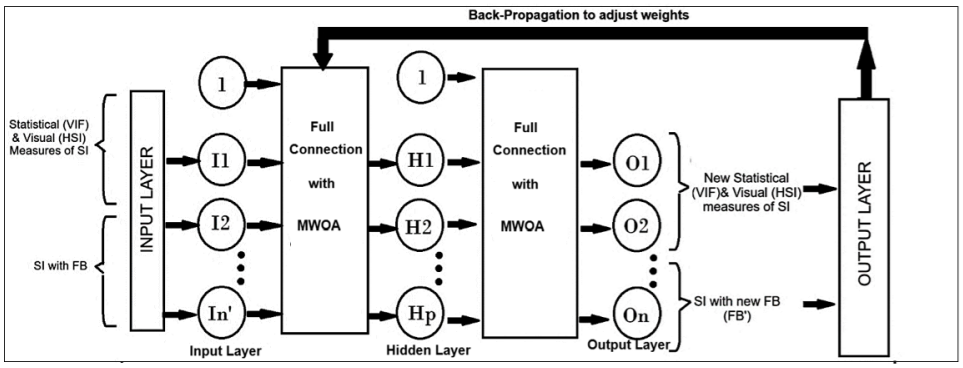

Fig. 5: ANN_MWOA with three layers $(\xi_{\text{ANN\_MWOA}}^{\text{SI1}^{\text{FB}},\text{SI2}^{\text{FB}\prime}})$

Fig. 5 illustrates two types of processing levels: many-to-one and one-to-many. The black arrow shows one-to-one transitions, while the green arrow indicates the adjustment route. The back-propagation technique with adaptive smoothing error (BPASE) employs the optimization approach MWOA and a neural network for weights adjustment (γ and ν) through three layers and to accelerate training [1].

5.1.1 Adaptive Neural Networks (ANN) training algorithm

Algorithm 6: ANN_MWOA training side

Step 1: Let us use three layers of neural networks:

The first layer $I_s \ \forall s = 1, \dots, \acute{n}$, *// where (n') is the number of neurons, and the input signals are collected of {FB, HSI, and VIF } values of SI1.*

The second layer $H_t \ \forall t = 1, \dots, p$, *// where (p) is the number of neurons to connect between input and output layers, See equation (9).*

$$\mathrm{H_t} = \Phi(\mathrm{h_t}) \tag{9}$$

The third layer $O_z \ \forall z = 1, \dots, n$,

// where (n) is the number of neurons, and the output signals are collected of {FB', new VIF, and new HIS} values of the new stego image SI2, and Φ (.) is a bipolar activation function, see equation(10).

$$\Phi(\mathrm{x}) = \frac{2}{1+\exp(-x)} - 1 \tag{10}$$

// and the first derivative of equation (10) is defined in equation(11).

$$\frac{d(\Phi(\mathrm{x}))}{dx} = \frac{1}{2}(1 + \Phi(x))(1 - \Phi(x)) \tag{11}$$

// where (h_t) is the activation function's parameter defined in equation (12).

$$h_t = \gamma_{0,t} + \sum_{s=1}^{n'} I_s \gamma_{s,t} \tag{12}$$

// and the weight $\gamma_{s,t}$ is working from the first layer's neurons Is to the second layer's neurons H_t)

Step 2: For(t=1; to t< p+1; t++) { *//check all second layer's neurons H_t;*

Calculate the activation value $\Phi(h_t)$ and send it to the third layer's neurons (Oz), see equation (13).

$$\mathrm{O_z} = \Phi(\mathrm{o_z}) \ \ \forall \mathrm{z} = 1, \dots, \mathrm{n} \tag{13}$$

//where (o_z) is defined in equation (14).

$$\mathrm{o_z} = \nu_{0\mathrm{z}} + \sum_{\mathrm{t}=1}^{\mathrm{n}'} \mathrm{H_t}\nu_{\mathrm{t,z}} \tag{14}$$

Step 3: For(z=1; to z< n+1; z++) { *//check third layer's neurons O_z :*

Step 3.1 Calculate correlated errors (δz);

// this error between (O_z) and target results (t,z) as seen in equation (15), where the outputs are {FB', new VIF, new HSI} values.

$$\delta_{\mathrm{z}} = (\mathrm{t_z} - \mathrm{O_z})\frac{\mathrm{d}(\Phi(\mathrm{o_z}))}{\mathrm{d\,o}} \ , \forall \mathrm{z} \tag{15}$$

Step 4: Call Algorithm 7 to apply MWOA to find $\nu_{t,z}$

Step 5: Update $\nu_{t,z}$;

// update the weights from the second layer's neurons H_t to the third layer's neurons (Oz) by using equation(16);

$$\Delta v_{t,z}^{new} = \beta \alpha_{t,z}^{new} \delta_z H_t + (1-\beta)\Delta v_{t,z}^{old} \quad (16)$$

// where $\beta \in [0,1]$ is the relaxation factor, to improve the speed of training during the training process, the adaptive learning rate (α) is applied (see equation(17)).

$$\alpha_{t,z}^{new} = \begin{cases} \alpha_{t,z}^{old} + \lambda & \text{if } \Delta v_{t,z}^{new} \Delta v_{t,z}^{old} > 0 \\ (1-\chi)\alpha_{t,z}^{old} & \text{if } \Delta v_{t,z}^{new} \Delta v_{t,z}^{old} < 0 \\ \alpha_{t,z}^{old} & \text{otherwise} \end{cases} \quad (17)$$

// where the value of parameters (λ and χ) has been selected to be equal to 0.018 and 0.85, respectively.

Step 6: For(t=1; to t< p+1; t++) { *//check errors of the second layer's neurons H_t;*

Compute δt error ∀t; ; // see equations. (18, 19).

$$\delta_{t'} = \sum_{z=1}^{m} \delta_z v_{t,z} \quad (18)$$

$$\delta_t = \delta_{t'} \times \left(\frac{d(\Phi(h_t))}{dh}\right) \quad (19)$$

Step 7: Call Algorithm 7 to apply MWOA to find $\gamma_{s,t}$

Step 8: Update ($\gamma_{s,t}$) (from the first layer's neurons Ii to the second layer's neurons H_t) using equation(20),

$$\Delta \gamma_{t,z}^{new} = \beta \alpha_{t,z}^{new} \delta_t I_s + (1-\beta)\Delta \gamma_{t,z}^{old} \quad (20)$$

// where β=0.15 is selected damping parameter,

Step 9: Update $v^{new}$and γ^{new}using equations (21, 22).

$$v_{t,z}^{new} = v_{t,z}^{old} + \Delta v_{t,z} \quad (21)$$

$$\gamma_{t,z}^{new} = \gamma_{t,z}^{old} + \Delta \gamma_{t,z} \quad (22)$$

Step 10: Repeat the above process many times until the accuracy is satisfactory.

Step 11: End.

5.1.2 Modified Scheme of the Optimization Algorithm (MWOA)

The mathematical model proposed is stimulated by a humpback's bubble-net feeding technique [16, 18]. This section explains a new intelligence optimization technique using the MWOA algorithm (see Fig. 6).

Humpback whales can effectively locate their prey and encircle them. While the exact location of the optimal solution in the search space remains unexplored beforehand, the MWOA algorithm assumes that the existing top nominee solution is either the target prey or near the optimum. Once the best search agent is identified, the other search agents adjust their locations to move to it. The following equation (23) illustrates this behavior.

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \times \left|\vec{C} \times \vec{X}_p(t) - \vec{X}(t)\right| \quad (23)$$

Where (t) refers to the current iteration, ($\vec{A}$ and $\vec{C}$) are two vectors, vector represents the location of the prey, and (X) is the location value of the whale [19].

The following two vectors ($\vec{A}$ and $\vec{C}$) are evaluated in the equations (24-25):

$$\vec{A} = 2 \times \vec{a} \times \overrightarrow{r1} - \vec{a} \quad (24)$$

$$\vec{C} = 2 \times \overrightarrow{r2} \quad (25)$$

Where ($\vec{a}$) components are linearly decreased from 2 to 0 throughout iterations, and ($\overrightarrow{r1}$, $\overrightarrow{r2}$) are random vectors in [0, 1].

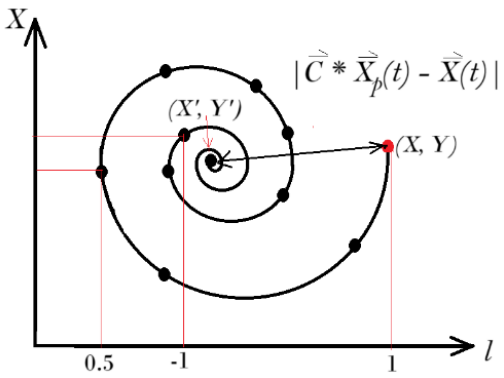

Fig. 6: A bubble-net hunting attack to show the spiral updating position

Bubble-net attacking method (exploitation phase)

The algorithm calculates the distance between the whale at location (X, Y) and its prey at (X', Y'). Then, a spiral equation is created to mimic the helix-shaped movement of humpback whales between these two locations [16], see equation (26).

$$\vec{X}(t+1) = \left|\vec{X}'(t) - \vec{X}(t)\right| e^{b \times r} \cos(2\pi r) + \vec{X}'(t) \tag{26}$$

Where X is the space of the i-th whale from the prey (which point is the top solution gathered up to this point), (b) represents a static number for defining the shape of the logarithmic spiral, where (r) is a random number in [-1,1].

At the same time, humpback whales swim in a spiral pattern, moving in a decreasing circle around their prey. To simulate this concurrent behavior, we assume a 50% chance of selecting either the shrinking encircling mechanism or the spiral model to revise the whales' locations throughout the optimization process [18]. The arithmetic model is described in equation (27):

$$\vec{X}(t+1) = \begin{cases} \vec{X}'(t) - \vec{A} \times \left|\vec{X}'(t) - \vec{X}(t)\right| & p < 0.5 \\ \left|\vec{X}'(t) - \vec{X}(t)\right| e^{br} \cos(2\pi r) + \vec{X}'(t) & p \geq 0.5 \end{cases} \tag{27}$$

The MWOA is proposed to smooth the whale locations as in equations (28-30).

$$\vec{X}(t+1) = p \times \vec{X}(t+1) + (1-p) \times \vec{X}(t) \tag{28}$$

Where p is in the interval [0,1], and it is calculated in equation (29):

$$p = \begin{cases} 0.3 & \left|\overrightarrow{X}(t+1) - \vec{X}(t)\right| > \left|\overrightarrow{X}(t) - \vec{X}(t-1)\right| \\ 0.7 & \left|\overrightarrow{X}(t+1) - \vec{X}(t)\right| \leq \left|\overrightarrow{X}(t) - \vec{X}(t-1)\right| \end{cases} \tag{29}$$

The bubble is enabled arbitrarily, allowing them to hunt prey at their discretion. Furthermore, the p-value falls within {0.3, 0.7}, which is determined by numerous experiments.

Search for prey (exploration phase)

During the exploration phase, humpback whales search for prey by moving randomly about each other. To facilitate this, we use a value $\vec{A}$ that is either greater than one or less than -1 to encourage the agent to travel away from a reference whale. In the exploration phase, we update the location of the exploring agent based on a randomly chosen search agent rather than the best search agent. This approach, combined with $A >$

1, emphasizes exploration and enables the MWOA algorithm to conduct a global search [16]. The mathematical model is as follows:

$$\vec{X}(t+1) = p \times (\overrightarrow{X_{rand}} - \vec{A} \times \left|\vec{C} \times \vec{X}_{rand} - \vec{X}(t)\right|) + (1-p) \times \vec{X}(t)) \quad (30)$$

where $\vec{X}_{rand}$ is a random location vector (a random whale).

The parameters used by MWOA

The parameters used by MWOA are defined as follows:

- Parameter $\vec{a}$ (in $\vec{A}$ vector) is defined in equation (24), where its range starts at 2 and linearly decreases to 0 across iterations. The effect on $\vec{A}$ is defined at the start: $\vec{A} \in [-2,2]$ and at the end: $\vec{A} \to 0$ (converges to exploitation mode) [16].
- Parameter $\vec{C}$ is defined in equation (25). Its range is in the interval [0,2] since $\overrightarrow{r2} \in [0,1]$. This parameter scales the difference between prey and whale positions, affecting encircling/exploration dynamics.
- The spiral movement parameters are defined in equation (26), where b is a constant that defines spiral tightness. The common choice: b=1, and the possible range: $0.5 \leq b \leq 2$. In addition, r is a random number in [−1,1] to control the spiral direction and pitch.
- The smoothing probability p is defined in equations (28–29), and its interval is {0.3, 0.7}. The smoothing condition is determined by equation (29).
- Parameter p used in equations (27, 30) is in the range, $p \in [0,1]$ acts as a random choice parameter to switch between encircling and spiral updating. The exploration condition happens when $|A|>1$ to ensure whales move away from a reference agent (global search). The exploitation occurs when $|A| \leq 1$; this makes whales move towards the best solution (local search).
- The random position Xrand defined in equation (30) is a uniformly random whale position vector within the search space bounds of the optimization problem. The final parameter ranges are shown in Table 1.

Table 1: Range of the parameters in the optimization algorithm (MWOA)

Parameter	Symbol	Range	Purpose
Linear coefficient	$\vec{a}$	[2 → 0]	Controls balance between exploration & exploitation
Position coefficient	$\vec{A}$	[-2$\vec{a}$, 2$\vec{a}$]	Determines whether to explore
Distance scaling	$\vec{C}$	[0, 2]	Adjusts distance weighting
Spiral tightness	b	[0.5, 2] (usually 1)	Shapes a logarithmic spiral
Spiral rotation	r	[-1, 1]	Defines spiral direction and length
Smoothing probability	p	{0.3, 0.7} or [0, 1]	Smooths position updates & controls movement mode
Random whale position	Xrand	Search space bounds	Ensures diverse exploration

Steps of the MWOA algorithm to update the location

Algorithm 7: MWOA algorithm

Step1. Initialize Solution:

- **Begin** with an initial solution.

- **Calculate** the fitness function (FitnessFunc) for each whale to check if it is near the bubble-net.

$$\text{FitnessFunc} = \alpha \times \text{MSE} + (1-\alpha) \times \frac{\text{Nf}-\text{Sf}}{\text{Nf}} \quad (31)$$

// *Where* $\alpha \in [0,1]$, *MSE is mean square error,* Sf *is the length of the selected feature, and* Nf *is the total number of features, see [20].*

Step2. Update the Location:

- **Use** equation (26) to update the location, which may vary based on the values of (p and A).

- **Step 2.1: Implement** spiral movement by updating the current location using equation (23).

- **Step 2.2: Transition** between exploration and exploitation phases by updating the current location with either equation (26) for exploration or equation (30) for exploitation.

Step3. Check Constraints:

- **Evaluate** the search space constraints and update the best solution if a better one is discovered, see equations (28-30);

Step4. Repeat Process:

- **Continue** looping from Step 2 until convergence is achieved in the following condition:

$$\left|\vec{X}(t+1) - \vec{X}(t)\right| < 10^{-5} \quad (32)$$

Step5. End the process.

5.2 Utilizing Visual and Statistical Measures in the ANN_MWOA Model

We identified and selected two models that effectively tackle visual and statistical attacks. During ANN_MWOA training,t he Salient Information (SI) is adjusted until the Stygenographic Message (Smsg) is optimized (seeFigure7).This study will use the "high, saturated, and intense" method(HIS) to address visual attacks and evaluate statistical attacks using the reliability of visual information (VIF).This approach is intended to improve the robustness of the model against various attack vectors.

5.2.1 Checking the visual attack using the HSI color conversion model

The HSI color conversion model is derived from the Red, Green, and Blue colors using equation (33-35). When an embedding algorithm slightly alters the pixel values, the HSI model can reflect this change. Therefore, the learning model ANN_MWOA preserves the SI pixel values to avoid color variations in the HSI model, ensuring that the difference between CI and SI is negligible. The HSI model begins by normalizing values to [0, 1].

$$H = \text{Cos}^{-1}\left(\frac{R-\frac{G}{2}-\frac{B}{2}}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right) \quad (33)$$

$$S = \frac{R+G+B-3\min(R,G,B)}{R+G+B} \quad (34)$$

$$I = \frac{R+G+B}{3} \quad (35)$$

5.2.2 Applying the VIF metric to check the statistical attacks

Human visual system (HVS), Image distortion (ID), and Gaussian scale mixture (GSM), are used to measure Visual Information Fidelity (VIF). This metric is derived from two measures of mutual information: the first is between the input to the distortion channel and the output of the HVS channels in the Structural Information (SI) [12] and [21]. The training algorithm calculates VIF using Algorithm 8, as discussed in [22].

Algorithm 8: The statistical and visual attack models
Step 1: Divide the SI and the CI into many sub-bands, and divide
Step 2: Divide sub-bands into Blocks. //*where the number of blocks is* $|ck|$
Step 3: Calculate GSM, ID, and HVS to regulate the VIF.
Step 3-1: Perform GSM on blocks at CI
Step3-1-1: compute C_{ji} // *where* C_{ji} *– the* i^{th} *block at the* j^{th} *sub-band in the CI.*

$$C_{ji} = S_{ji} \times U_{ji} \quad (36)$$

// *where* S_{ji} *is a positive scale and* U_{ji} *is a Gaussian vector*
Step 3-2: Perform ID on blocks at SI
Step 3-2-1: compute D_{ji};// where D_{ji} is the ith block at the jth sub-band in the SI

$$D_{ji} = G_{ji} \times \varphi_{ji} \quad (37)$$

// *where Gji – scalar gain field and φji– Gaussian noise*
Step 3-3: Perform HVS ;
Step 3-3-1: Assume μ and μ' are the noise in CI
Step 3-3-2 Find E_{ji} // *where* E_{ji} *is a cognitive output for the ith block at the jth sub-band of the CI,*

$$E_{ji} = S_{ji} + 2\mu \quad (38)$$

Step 3-3-3: Find F_{ji} // *where* F_{ji} *is a cognitive output for the ith block at the jth sub-band of the SI,*

$$F_{ji} = G_{ji} \times C_{ji} + \varphi_{ji} + \acute{\mu} \quad (39)$$

Step 4: Estimate the image superiority value
Step 4-1: Perform I(Cji, Eji); //*the mutual information for the CI, see equation (40).*

$$I(C_{ji}, E_{ji}) = H(C_{ji} + \mu) - H(\mu) \quad (40)$$

// where H(.) is the entropy
Step 4-2: Find I (Cji, Fji); // *the mutual information for the SI that is retrieved from the output of HVS. see equation (41).*

$$I(C_{ji}, F_{ji}) = H(G_{ji} \times C_{ji} + \varphi_{ji} + \acute{\mu}) \quad (41)$$

Step 4-3: Perform VIF(see Equation(42));

$$VIF = \frac{\sum_{j\in Sub-band}\sum_{i\in block} I(C_{ji},F_{ji})}{\sum_{j\in Sub-band}\sum_{i\in block} I(C_{ji},E_{ji})} \quad (42)$$

Step5: End;

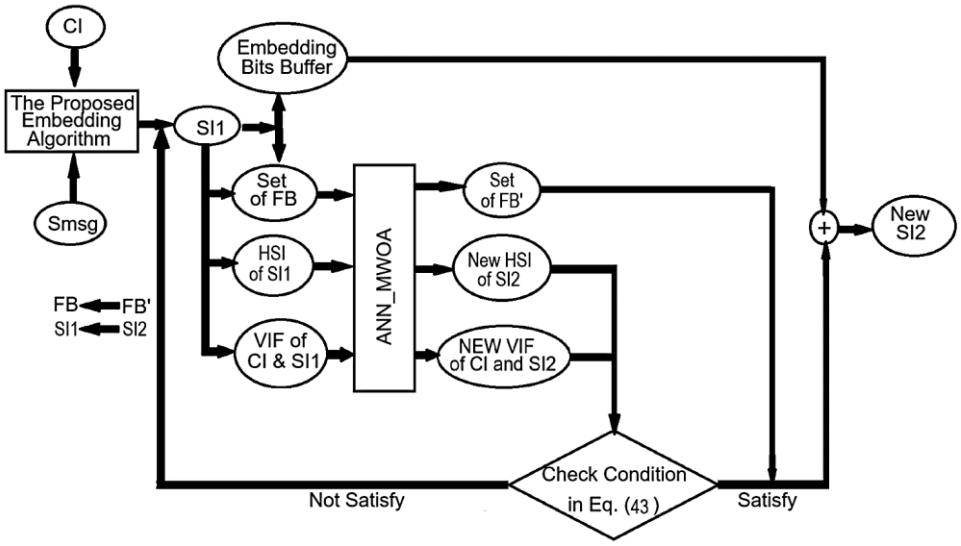

Fig. 7: The training system model ($\xi_{\mathbf{ANN\_MWO}}^{\mathbf{SI1^{FB},SI2^{FB\prime}}}$)

5.3 Combine the Data Embedding Algorithm with ANN_MWOA

The combination of Algorithms (1-3 and 5-8) is explained in Algorithm 9 according to the following steps:

Algorithm 9: Combine Algorithms
Let the three types of relative error be:
Hue, (REH,)
saturation, (RES,)
intensity, (REI)
Let HCI and HSI be the hues of CI and SI, respectively.
Let SCI and SSI represent the saturations of CI and SI, respectively.
Step 1: Input Smsg and CI;
Step 2: Apply NAIS; // *Calling Algorithms (1 & 2),*
Step 3: Apply embedding a Smsg; to generate SI1 // Calling *Algorithm* (3& 5),
Step 4: Calculate HSI and VIF of SI1 // *Calling Algorithm (8)*
Step 5: Identify the set of free bits, FBs, and their location from SI1 ;
Step 6: Perform training on a set of FB to predict a new set of FB'.
Step 6.1: Apply ANN_MWOA //*Calling Algorithm (6);*
Step 6.2: Apply MWOA; //Calling *Algorithm* (7);
Step 6.3: Generate SI2 with new free bits FB';
Step 6.4: Compute HSI and VIF of SI2// *Calling Algorithm (8);*
Step 7: Check the given condition in equation (43) for each pixel, // *if the relative error REs, REh, and REI, are in the range of the conditions, then build the new SI1 by adding FB' to the stego bits' buffer and go to Step 8;*

Otherwise, adjust the weight values (ADJ) as in the Equations. (21, 22) (See Fig. 7), update the set of FB by the set of FB', then go to **Step 5**.

$$\left.\begin{array}{c} \text{VIF} > 0.85 \\ \text{RE}_{\text{H}} = \frac{\text{H}_{\text{CI}} - \text{H}_{\text{SI}}}{\text{H}_{\text{SI}}} < \pi/18 \\ \text{RE}_{\text{S}} = \frac{\text{S}_{\text{CI}} - \text{S}_{\text{SI}}}{\text{S}_{\text{SI}}} < 10^{-6} \\ \text{RE}_{\text{I}} = \frac{\text{I}_{\text{CI}} - \text{I}_{\text{SI}}}{\text{I}_{\text{SI}}} < 10^{-6} \end{array}\right) \quad (43)$$

Step 8: End.

5.4 Implementation of the data embedding algorithm with ANN_MWOA

The embedding mechanism uses the surrounding eight bytes (SELB) in conjunction with the current byte in a context indicator (CI), as illustrated in Fig. 8. We need to explain, step-by-step, how to embed the message (Smsg) using a hybrid machine learning ANN_MWOA to make Smsg imperceptible.

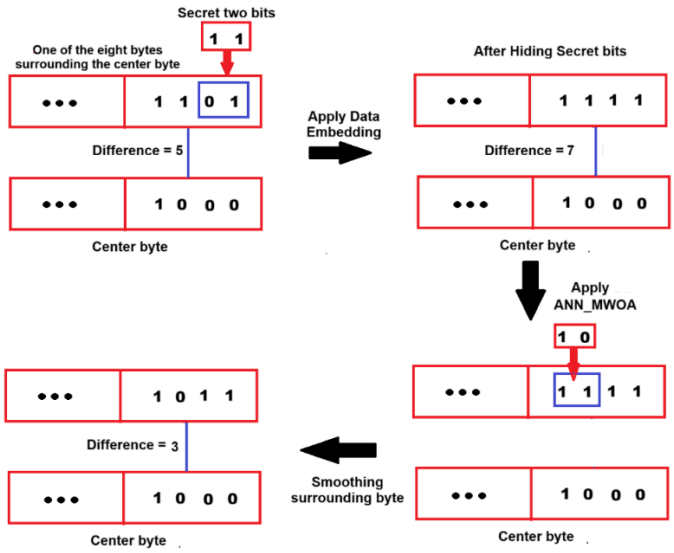

Fig. 8: Embedding Smsg into CI to produce SI

Fig. 8 illustrates eight lower nibbles of bytes surrounding the current byte (SELB). SELB is utilized to determine the difference between neighboring bytes. The first step of Fig. 8 demonstrates the proposed embedding algorithm, which embeds the Smsg into CI. At the same time, the second step applies the ANN_MWOA algorithm to dampen the pixel value difference (PVD) between cover and stego images. This approach aims to achieve pixel consistency and reliability, and to enhance resilience against attackers.

6. Results and Discussions

The presented data embedding technique uses a hybrid model, ANN_MWOA, to embed Smsg within color images while incorporating a novel embedding algorithm to guard against visual and statistical attackers.

To evaluate its effectiveness, the technique was trained using 1040 color images and tested on 260 color images. These images are from the UCID v2 dataset. The parameter range used in the proposed ANN_MWOA is shown in Table 2.

Table 2: The range of parameters used in the proposed ANN_MWOA

Parameters	Value/Condition
Batch size	64
Learning rate	1e-5
Epochs	205 (maximum)
Noise dimension	100
Hidden dimension	64
Loss function	Binary Cross Entropy
Optimizer (Discriminator)	MWOA
MWOA - Population Size	30
MWOA - Max Iterations	Until the satisfied condition of convergence
Smoothing probability p	p∈ [0.3, 0.7]

Preprocessing steps are applied on images at the dataset by getting free bits (after the embedding process) from SI1, getting HIS and VIF information of SI1, and combining them as an input signal for each SI1 image.

Hyperparameter tuning for neural networks is conducted as follows:

- Using adaptive smoothing error (BPASE)
- Using the optimization approach MWOA and a neural network for weights adjustment (γ and ν) across three layers during training to speed up the process.
- Applying an adaptive learning rate (α) (see equation (17)).
- The Modified WOA has been proposed to smooth whale locations during the training phase, see equations (28-30).
- The number of epochs is used to enhance training accuracy, along with the optimizer type.

In this section, we display the results for six images selected from the datasets, shown in Fig. 9a-9b, to compare them with their analysis. The UCID v2 and Kodak-PCD0992 color image datasets are combined to form a dataset spanning sizes from 150×150 to 1080×1024 pixels, enabling a thorough assessment of data embedding effectiveness and payload capacity. Image colors influence bits per byte within a 3×3 window (8-neighbor bytes around the current byte), and the variance of the individual cover for each color component was computed to determine data embedding capacity [23-24].

Fig. 9a: A sample of bitmap images sourced from the UCID v2 database

Fig. 9b: A sample of bitmap images sourced from the Kodak-PCD0992 database

6.1 Evaluation metrics and comparisons with the previous works

A comprehensive test is performed on the image dataset, comparing the proposed approach's results with those of other techniques under the same conditions. Table 3 shows the results of three metrics: Mean Squared Error (MSE), Signal-to-Noise Ratio (SNR), and Peak Signal-to-Noise Ratio (PSNR), as defined in equations (44-46).

$$MSE \frac{1}{h \times w} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \left(CI_{i,j} \quad SI_{i,j}\right)^2 \tag{44}$$

$$\text{SNR} = \frac{\mu}{\sigma} \tag{45}$$

where SNR denotes the signal-to-noise ratio, it is the ratio between the mean (μ) and the standard deviation (σ).

$$\text{PSNR} = 10 \log_{10} \left(\frac{MAX^2}{MSE}\right) \tag{46}$$

where MAX is the maximum pixel value at SI.

The proposed method is compared with the existing techniques in [25-26], as well as the conventional Least Significant Bit (LSB) method. Evaluation metrics included Peak Signal-to-Noise Ratio (PSNR), Signal-to-Noise Ratio (SNR), and Mean Squared Error (MSE). The results show that the proposed techniques outperform the alternatives, as summarized in Table 3.

Table 3: Comparison between the proposed algorithm and the previous works using three metrics (MSE, SNR, and PSNR)

Image Name $512 \times 512 \times 3$	Payload Capacity (bits)	Traditional LSB		[25]		[26]			The Proposed Algorithm (ANN_MWO)		
		SNR (dB)	PSNR (dB)	SNR (dB)	PSNR (dB)	MSE (dB)	SNR (dB)	PSNR (dB)	MSE (dB)	SNR (dB)	PSNR (dB)
People1	3162480	25.36	27.94	43.96	44.14	0.58	48.78	50.44	0.049	55.13	61.22
Tulips	3162480	30.62	34.99	48.63	53.70	0.06	55.40	59.76	0.046	59.12	61.42
Lenna	3162480	24.11	29.25	44.41	46.39	0.19	50.14	55.28	0.061	58.45	60.25
Onion	3162480	17.22	24.22	33.37	38.47	1.85	38.44	45.45	0.047	53.11	61.34

Specifically, the PSNR results show that when using a large payload capacity, the proposed methods outperform previous studies by approximately 17.08 dB, 10.78 dB, and 33.28 dB for the People1 image; by about 7.72 dB, 1.66 dB, and 26.43 dB for the Tulips image; by approximately 13.86 dB, 4.97 dB, and 31.0 dB for the Lenna image; and by around 22.87 dB, 15.89 dB, and 37.12 dB for the Onion image.

Across all test cases, the proposed ANN_MWO achieves substantially lower MSE values. The previous work [26] records MSE values ranging from 0.06dB to 1.85dB, whereas the proposed algorithm reduces MSE intensely to the range from 0.046dB to 0.061dB.

Similarly, the proposed SNR results, also using a large payload capacity, exceed those of previous studies by approximately 11.17dB, 6.35dB, and 29.77dB for the People1 image; by about 10.49dB, 3.72dB, and 28.5dB for the Tulips image; by approximately 14.04dB, 8.31dB, and 34.34dB for the Lenna image; and by around 19.74dB, 14.67dB, and 35.89dB for the Onion image.

The results for the large payload capacity demonstrate improvements compared to those of Ahmed and El-Emam (2021). The errors were reduced by approximately 0.17 for the People1 image, about 0.04 for the Tulips image, roughly 0.14 for the Lenna image, and around 1.18 for the Onion image.

6.2 Comparison of the proposed results with previous studies and traditional WOA.

The impact of payload capacity on PSNR is shown in Table 4. In addition, it shows the performance of previously documented techniques from studies [22], [25], and [26], as well as the proposed approach, which utilizes the traditional Whale optimization algorithm (ANN_WOA). Additionally, this table compares these techniques with the proposed ANN_MWOA method. The comparison is performed using four colored bitmap test images, each measuring $512 \times 512 \times 3$ pixels, sourced from the UCID v2 database.

Table 4: Performance comparison between the proposed algorithm and four embedding algorithms using the PSNR metric

Images $512 \times 512 \times 3$	Payload Capacity (bits) $\times 10^4$	PSNR (dB) of [22]	PSNR (dB) [25]	PSNR (dB) of [26]	PSNR (dB) of the Algorithm ANN_WOA	PSNR (dB) of the proposed algorithm ANN_MWOA
Baboon	2	65.6	59.2	63.3	65.7	67.5
	2.8	64.2	56.1	57.8	65.7	66.7
	3.6	63.0	54.9	56.1	64.1	65.5
	4.4	62.1	53.2	55.4	63.2	64.2
	5.6	61.0	53.4	53.6	61.1	63.1
Peppers	2	67.3	61.2	64.9	68.2	69.5
	4	64.9	59.1	59.2	65.3	67.2
	6	63.4	55.3	57.5	63.5	65.6
	8	62.2	53.7	56.7	63.6	64.2
	10.5	61.2	53.7	54.9	62.1	63.7

In Table 4, the PSNR metric, which has a minimum payload capacity of 20,000 bits, outperforms the earlier algorithms by approximately 8.3 dB, 4.2 dB, 1.9 dB, and 1.8 dB, respectively, when tested on the Baboon image. For the Peppers image, it surpasses the other algorithms by about 8.3 dB, 4.6 dB, 2.2 dB, and 1.3 dB, respectively. In contrast, the PSNR results of the proposed algorithm with a maximum payload capacity of 56,000 bits for the Baboon image and 105,000 bits for the Peppers image exceed those of the other algorithms by approximately 9.7dB, 9.5dB, 2.1dB, and 2dB, respectively. The improvement is around 10dB, 8.8dB, 2.5dB, and 1.6 dB, respectively, for the Peppers image.

Table 5: PSNR comparison between the proposed algorithm and previous work in [27]

Images $\mathbf{512 \times 512 \times 3}$	Payload (bits)	PSNR (simple LSB method)	PSNR (dB) (DDV method) using PSO [27]	PSNR (dB) (Hufman + DDV) using PSO [27]	PSNR (dB) Proposed embedding technique using ANN_AWOA
Lenna	16384	62.2	71.32	78.09	79.23
	32768	61.22	67.63	73.25	76.15
	49152	55.76	63.16	68.94	74.69
	65536	50.82	55.91	61.41	68.31
Baboon	16384	62.3	71.31	78.08	79.58
	32768	61.25	67.56	73.28	76.15
	49152	55.65	63.12	68.89	72.95
	65536	50.77	55.90	61.43	68.3

Table 5 presents the performance of previously developed techniques from a study by [27] that used the PSO optimization algorithm. The proposed approach employs new embedding techniques based on ANN_MWOA. Additionally, this table compares these techniques using two colored bitmap test images, each measuring 512 x 512 pixels, sourced from the UCID v2 database. It appears that the average PSNR of two images using the proposed approach is better than the DDV method by about 10 dB and better than the Huffman with DDV method by about 4 dB.

Table 6: MSE comparison between the proposed algorithm and the previous work [27]

Images $\mathbf{512 \times 512 \times 3}$	Payload (bits)	MSE (dB) (simple LSB method)	MSE (dB) (DDV method) using PSO [27]	MSE (dB) (Huffman + DDV) using PSO, [27]	MSE(dB) of the proposed Algorithm
Lenna	16384	0.3328	0.2123	0.1012	0.00077
	32768	0.6453	0.5023	0.2011	0.00157
	49152	0.9332	0.8323	0.6011	0.00220
	65536	1.3228	0.9881	0.8021	0.00959
Baboon	16384	0.6451	0.4312	0.2212	0.00071
	32768	0.9881	0.7244	0.5231	0.00157
	49152	1.1123	0.8212	0.7511	0.00329
	65536	0.6451	0.4312	0.2212	0.00961

Table 6 compares the MSE of the proposed embedding technique using ANN_MWOA with previous results from [27] using the PSO optimization Algorithm. The proposed

approach utilizes two color images and compares them with the last technique. It appears that the proposed technique, using the average of the MSE of two images, is lower than the LSB method about 0.8244dB, lower that DDV method by about 0.614 (dB) and lower than the (Huffman with DDV) method by about 0.42dB.

6.3 Compare the proposed and previous works on a set of images

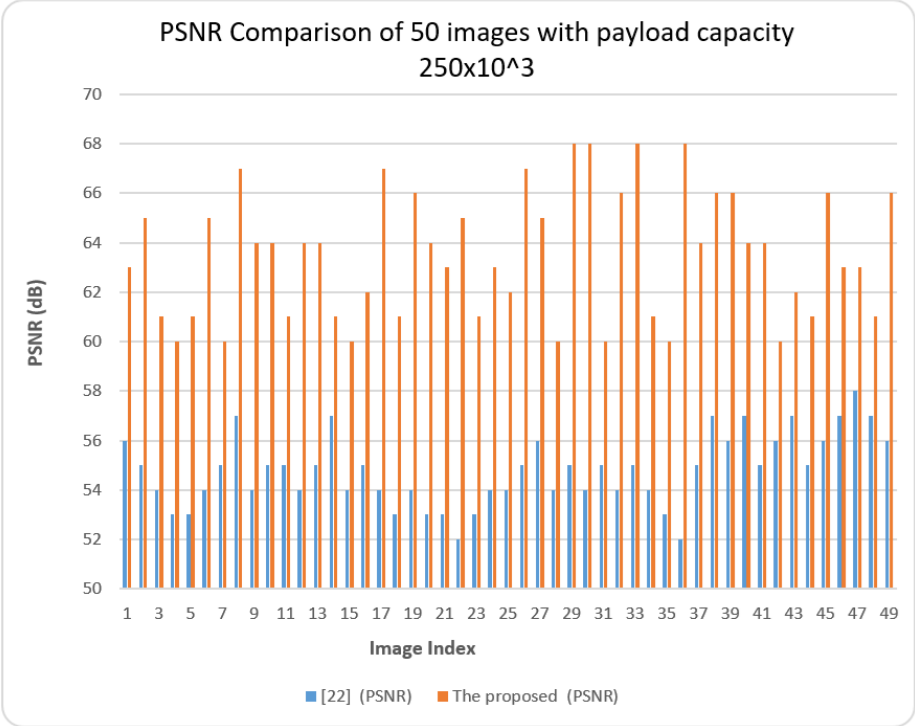

Fig. 10: PSNR comparison of 50 images with a payload capacity of 250×10^3

The results shown in Figure 10 compare the Peak Signal-to-Noise Ratio (PSNR) of 50 test images embedded with a payload capacity of (250 ×10^3) bits using two methods: the approach from [22] and the proposed approach. PSNR is a common metric for assessing image quality, where higher values indicate better preservation of visual fidelity after data embedding. The proposed approach consistently outperforms the reference approach across all images. While the PSNR values for [22] range from 52 dB to 56 dB, the proposed approach achieves notably higher values, mostly between 62 dB and 68 dB, with several images reaching the maximum of 68 dB. This improvement of approximately 10–12 dB is significant since even a 1 dB increase in PSNR can lead to a noticeable enhancement in image quality. Results show that the proposed method significantly improves image quality at the same payload capacity, confirming its robustness and efficiency compared to current techniques.

6.4 Measurement of the Euclidean Norm

The Euclidean norm, as defined in equation (41), was applied to color images with a size of 512x512 in three color channels (R, G, and B) to evaluate the distance (d) between the CI and the SI.

$$d = ((R_{CI} - R_{SI})^2 + (G_{CI} - G_{SI})^2 + (B_{CI} - B_{SI})^2)^{\frac{1}{2}} \quad (47)$$

The proposed algorithm achieves the shortest distance. Results show a maximum difference of 330 with a 40% payload for Pepper's image, and a minimum difference of 9 with a 10% payload for Lenna's image. See Table 7 for details.

Table 7: Assessment of the effects of the Euclidean norm vs the payload capacity on the proposed algorithm and previous works

Images $\mathbf{512 \times 512 \times 3}$	Payload Capacity	Euclidean norm [25]	Euclidean norm [26]	Euclidean norm [22]	Euclidean norm of the Algorithm **ANN_WOA**	Euclidean norm of the proposed Algorithm **ANN_MWA**
Lenna	10%	250	200	270	194	185
	30%	600	570	450	433	410
	40%	780	750	550	544	520
Peppers	10%	200	150	250	145	120
	30%	700	630	430	417	410
	40%	800	780	500	494	470

6.5 Ablation Study on the Proposed Optimization Algorithm

To assess the impact of modifications to the Whale Optimization Algorithm (WOA), we compare the performance of the standard ANN_WOA with that of the modified ANN_MWOA. As shown in Table 4, the modified variant consistently yields higher PSNR values for both the Baboon and Peppers images. This suggests that the changes implemented in the MWOA improve reconstruction quality, demonstrating the effectiveness of the proposed optimization enhancements.

Moreover, to evaluate the effectiveness of the proposed ANN_MWOA algorithm compared to the base ANN_WOA, we conducted an ablation study using the Euclidean norm as a distortion metric. Table 7 shows that ANN_MWOA consistently achieves lower Euclidean norms across various payload capacities and for different cover images. For instance, at 40% payload on the Lenna image, the Euclidean norm decreased from 544 (ANN_WOA) to 520 (ANN_MWOA), indicating reduced distortion. This confirms that enhancements in the MWOA component improve robustness and preserve image quality, even at higher embedding capacities.

6.6 Preventing a WFLogSv Attack

The proposed data embedding algorithm aims to embed a Smsg within a color CI while ensuring the SI appears normal and does not raise suspicion. This study evaluates the SI against the WFLogSv attacker using the Receiver Operating Characteristic (ROC) curve, calculating the probability of detection (1−PMD) and the likelihood of false alarms (PFA).

$$P_{MD} = \frac{NSI_{CI}}{N_{SI}} \quad (48)$$

$$P_{FA} = \frac{NCI_{SI}}{NCI} \quad (49)$$

Where NCI_{SI} is the number of CIs documented as SIs, NCI is the total number of CIs, NSI_{CI} is the number of SIs documented as CIs, and NSI is the total number of SIs.

The ROC curve presents P_{FA} and $(1 - P_{MD})$ as horizontal and vertical axes, respectively. A Data embedding technique is said to be secure from attackers if the following condition is satisfied:

$$|P_{FA} - 1 + P_{MD}| = \varepsilon, \ \varepsilon \to 0 \quad (50)$$

Perfect detection is achieved when AUC equals 0, while perfect security is achieved when AUC equals 0.5.

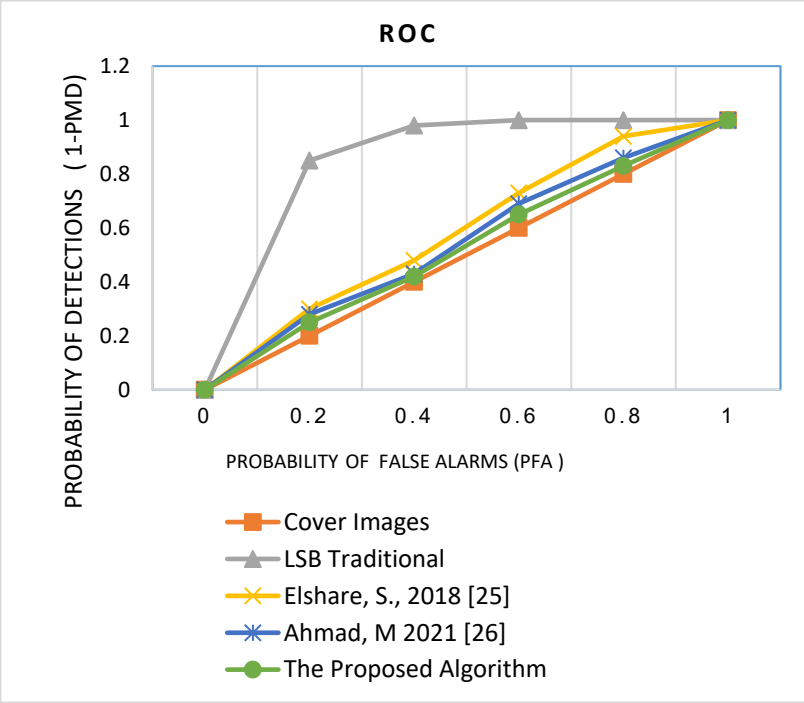

Fig. 11: ROC curves of WFLogSv against the proposed hiding technique and traditional LSB at 40% payload capacity.

Fig. 11 compares study performance on the WFLogSv attack at 40% payload capacity. The findings show that the WFLogSv attack has a high detection rate with traditional Least Significant Bit (LSB) techniques. However, when the proposed approach is used, the detection rate drops significantly, with a maximum likelihood of detecting a Smsg of only 7.5%. Previous techniques in [25-26] achieved detection rates of about 22.5%, 13%, and 91.5%, respectively.

6.7 Assess the Detection Error (P_E)

The detection error probability (P_E) in equation (45) exhibits performance comparable to that of current steganography techniques. The error P_E ranges from 0 to 0.5, where 0 is for perfect detection while 0.5 is for perfect security, as shown in Fig. 12. Detection error is estimated as a function of payload capacity in bits per pixel (bpp) to determine the area under the curve (AUC). Results are compared with those of previous studies in [25-26].

$$P_E = \min\left(\frac{1}{2}(P_{FA} + P_{MD})\right) \quad (51)$$

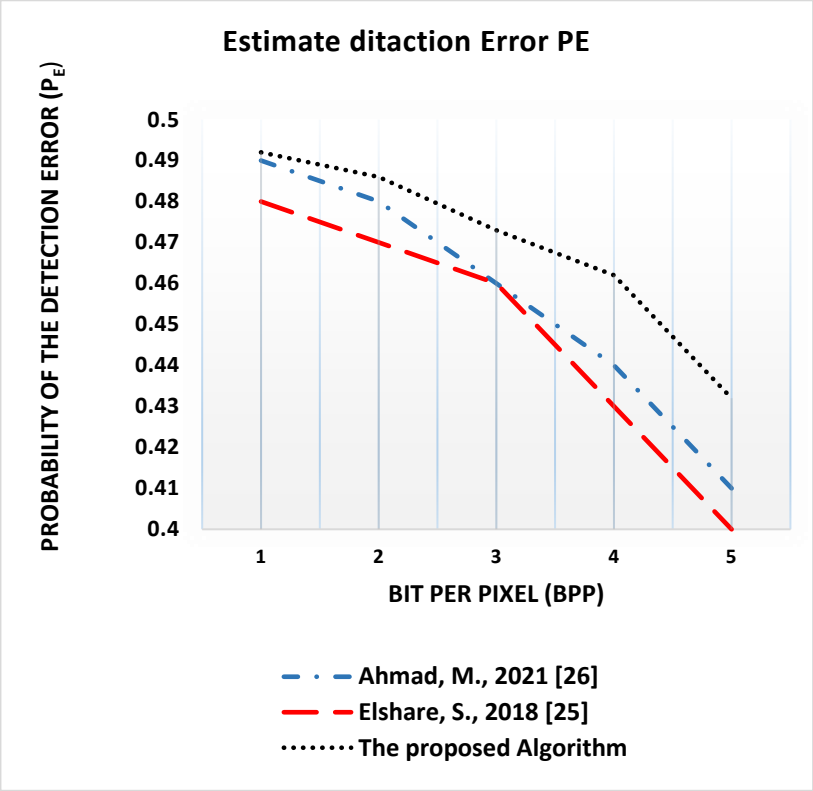

Fig. 12: The probability of the detection error (PE)

The presented embedding algorithm achieves an average bits per pixel (bpp) of 0.469, outperforming in [25] by 4.2%. and [26] by 2.6%, and it also attained a security rate of 98.4% at 0.05 bpp, with a minimum of 86.4% at 0.4 bpp.

6.8 Preventing Chi-square ($\chi 2$) Attack

The embedding algorithm aims to embed Smsg within a color image without revealing that the resulting stego image contains hidden information. This paper evaluates the expected (Ei) and observed (Oi) frequencies of stego pixels with the Chi-square statistic, as shown in equation (52) [22].

$$\chi^2_{k-1} = \sum_{i=0}^{k-1} \frac{(O_i - E_i)^2}{E_i} \tag{52}$$

In this context, (k) represents the number of pairs in the stego image, (k-1) indicates the degree of freedom, and (Ei) denotes the expected frequency of the $\{P^c_{2i}, P^c_{2i+1}\}$ pair; refer to equation (53).

$$E_i = \frac{1}{2} \underset{c\in\{R,G,B\}}{fr}\{P^c_{2i}, P^c_{2i+1}\}, \forall i = 0, \ldots, k-1 \tag{53}$$

In the palette colors for pixels $\{\{P^c_0, P^c_1, P^c_2, \ldots, P^c_k\}$, the colors $\{P^c_{2i}, P^c_{2i+1}\}$ correspond to each other. Additionally, the frequency observed at the color (c) is represented in equation (54).

$$Oi. = fr(C_i) \forall i = 0, \ldots, k-1 \tag{54}$$

The probability ($Pr_{\chi^2,k-1}$) based on the Chi-square value (χ^2) with (k-1) degrees of freedom is calculated using Equation (55).

$$\Pr_{\chi^2,k-1} = \left(2^{\frac{k-1}{2}}\,\Gamma\left(\frac{k-1}{2}\right)\right)^{-1} \int_{\chi^2}^{\infty} (t)^{\frac{k-1}{2}-1} e^{-\frac{t}{2}}\, dt \tag{55}$$

where Gamma $\Gamma(.)$ is the simplification of the factorial function equation (56).

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt \tag{56}$$

A Chi-square attack is measured on SI that includes data embedding with disordered messages. Comparisons between two cover images (Baboon and Peppers) and their stego images reveal that the Pr value drops to zero at block sizes of 25 or 50, depending on the image color. This leads to an embedding message size (Smsg) of about 25% to 50% of the image size for different message types. Consequently, steganalysis cannot detect Sm because the Pr values are highly similar. [22].

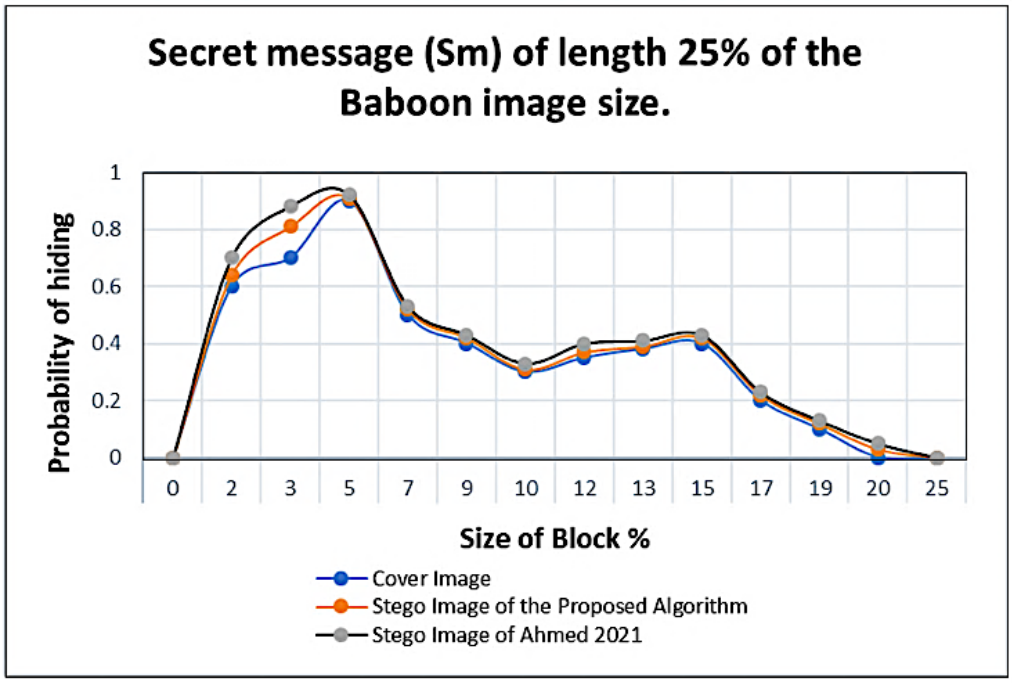

Fig. 13: The probability of embedding an Smsg that represents 25% the size of the Baboon image

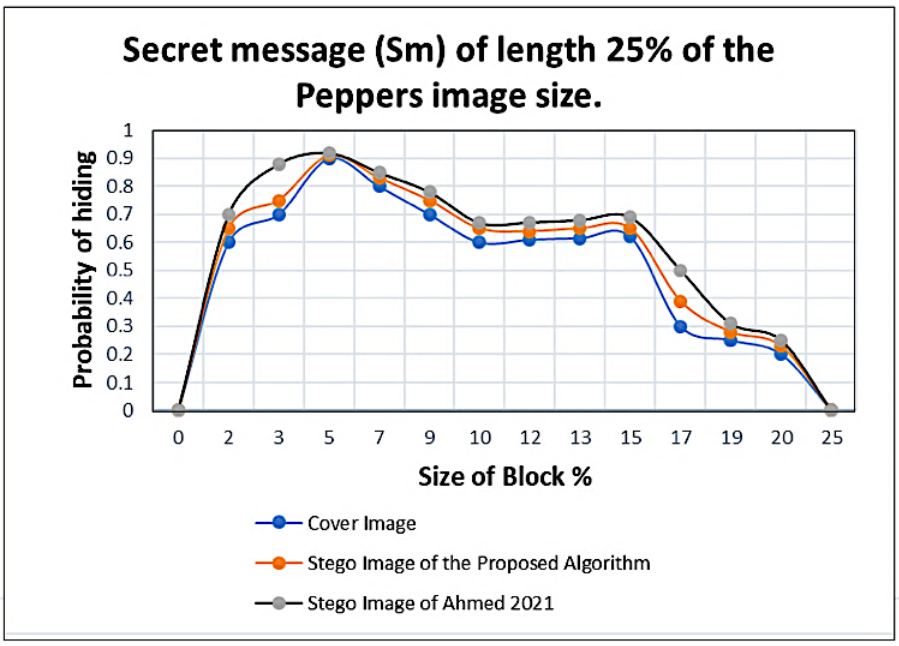

Fig. 14: The probability of embedding an Smsg that represents 25% the size of the Pepper's image

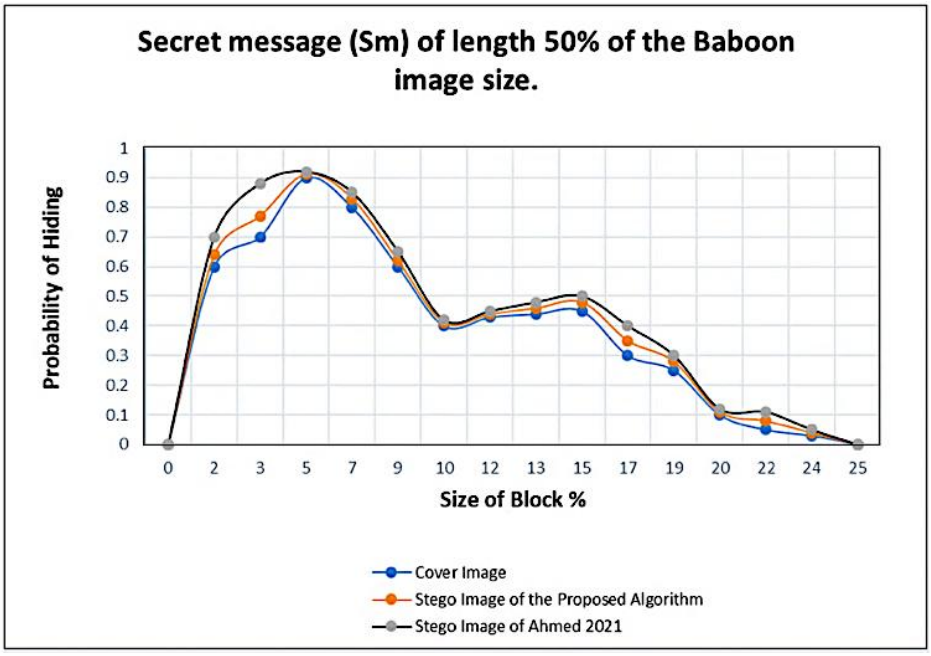

Fig. 15: The probability of embedding an Smsg that represents 50% the size of the Baboon image

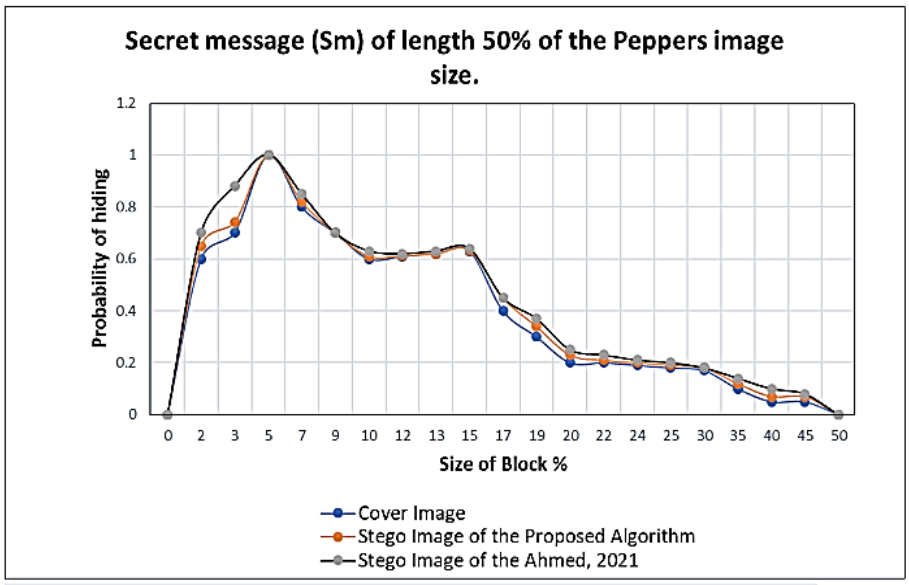

Fig. 16: The probability of embedding an SMS that represents 50% the size of Pepper's image

Figures 13-16 show the proposed algorithm analyzing average embedding probabilities for the CI and SI. It can embed Smsg that is 25% or 50% of the image size. The differences in embedding probabilities were approximately 0.0435 and 0.0233 for Baboon images, and 0.0164 for Peppers images. Additionally, when the Smsg size is 25% or 50% of the image size, the proposed algorithm has a lower success rate for embedding the Smsg than Ahmed et al. [26]. The differences are about 4.4% and 2.6% for the Baboon image, and 8.5% and 1.98% for the Pepper image.

6.9 Calculating the VIF

MWOA and other metaheuristics can be used to optimize encoding, enhancement, or embedding strategies to maximize VIF and thereby improve perceptual video quality [18].

Table 8 explains how to assess image quality and similarity using the Visual Information Fidelity (VIF) metric, which is defined in equation (33). Three 256 x 256 pixel standard color images, Lenna, Pappers, and Baboon, were used in this study. Payload capacities of 10%, 20%, and 30% were used to test the proposed embedding procedure. Results indicate that the algorithm surpasses the reference in [22] by about 2%, 3%, and 4%.

Table 8: Comparison between the proposed algorithm and the previous work [22] using the VIF metric

Color stego image 256 × 256	Payload capacity %	[22]	Proposed Algorithm
Lenna	10%	0.96	0.98
Pappers	20%	0.92	0.95
Baboon	30%	0.90	0.94

6.10 Computing loss (MSE) for every Epoch

This paper applies Mean Squared Error (MSE) calculation across epochs during model training to monitor performance over time. The objectives of this study are: to track how

well a model is learning, to detect overfitting or underfitting, and to tune hyperparameters using validation loss.

Fig. 17: Computing loss vs Epoch using WOA and MWOA for both training and testing

Figure 17 shows the loss over epochs for different optimization algorithms. The loss function for both training and testing with ANN_MWOA decreases steadily, especially during the first 5 to 55 epochs. The results suggest that the model improves during training, and the training and testing lines stay close together. After 155 epochs, both lines level off around 0.4, indicating the model has reached its optimal performance. At epoch 205, the MSE for both training and testing drops to 0.04. Additionally, we found that the proposed system with the ANN_MWOA model performs about 27.7% better than the traditional ANN_WOA model during training and roughly 29.8% better during testing.

7. Conclusion and Future Works

This paper presents a new machine learning model (ANN-MWOA) that utilizes a novel data-embedding algorithm. This sophisticated algorithm enables embedding a large amount of Smsg into color images and enhancing the security of the sensitive data.

Five layers of protection have been put in place to strengthen the embedding process; each is tailored to effectively thwart both statistical and visual attacks.

Moreover, in the embedding algorithm, we have developed a state-of-the-art image segmentation algorithm called New Adaptive Image Segmentation (NAIS). This algorithm enables the random embedding of Smsg throughout the image, further complicating detection efforts. Our approach also incorporates a novel assessment method for byte characteristics to determine the optimal number of bits to embed per lower byte (Nbplb). This assessment is achieved by utilizing the high nibbles extracted

from the eight neighboring high bytes (designated as STHB) adjacent to the current byte being processed.

The proposed work applied an adaptive neural network-based machine learning component, strengthened by a modified Whale Optimization embedding rate. Employing uniform adaptive relaxation after introducing the embedding technique accelerates training and enables us to achieve a high embedding rate without compromising the undetectability of the resulting SI. The innovative MWOA plays a pivotal role in identifying optimal particle locations to find the weights of the adaptive neural network, thereby enhancing its performance. This process employs a smoothing approach that fine-tunes the whale's location, improving accuracy and efficiency.

Our experimental results validate the effectiveness of the proposed embedding algorithm, demonstrating a significant increase in the data embedding rate while also enhancing the imperceptibility of the embedded messages. The implementation of the five layers of protective measures has proven essential to maintaining high performance. Moreover, the comparative analysis reveals that the differences between the CI pixels and their corresponding SI pixels are minimal when using the ANN_MWOA technique. This subtlety significantly reduces the likelihood that attackers will perceive any alterations, ensuring that the hidden data remains secure and undetectable.

Future work on this approach includes extending, improving, or applying it in new contexts by integrating the proposed algorithm with other optimization techniques. This involves combining the modified Whale Optimization Algorithm (WOA) with additional metaheuristics or hybrid methods, such as Genetic Algorithms and Particle Swarm Optimization, to improve convergence speed and accuracy. It is also important to examine scalability and the use of embedding methods on large-scale datasets, such as social media or IoT sensor data, to evaluate performance in real-world big data scenarios. Furthermore, investigating robustness to adversarial attacks and noisy data is a growing trend in embedding techniques to improve data security and privacy-preserving machine learning [28].

References

[1] Ping, P., Li, Q., Guo, B., Xu, F., Sibanda, F., & Mao, Y. (2023). Image embedding using an invertible neural network and the similarity of bit pairs. *Applied Soft Computing*, 151, 111180.https://doi.org/10.1016/j.asoc.2023.111180

[2] Abd-El-Atty, B. (2023). A robust medical image steganography approach based on particle swarm optimization algorithm and quantum walks. *Neural Computing & Applications*, 35, 773–785 . https://doi.org/10.1007/s00521-022-07830-0

[3] Wang, S., Asif, M., Shahzad, M. F., & Ashfaq, M. (2024). Data privacy and cybersecurity challenges in the digital transformation of the banking sector. *Computers & Security*, 147, 104051. https://doi.org/10.1016/j.cose.2024.104051

[4] Younis, Y.M., Mstafa, R.J., & AL-Dohuki, S. (2025). AttenHideNet: A novel deep learning-based image steganography method using a lightweight U-net with soft attention. *Applied Soft Computing*. 182, 113583. DOI: 10.1016/j.asoc.2025.113583

[5] Hu, K., Wang, M., Ma, X., Chen, J., Wang, X., & Wang, X. (2024). Learning-based image steganography and watermarking: A survey. *Expert Systems with Application*, 249, 123715. https://doi.org/10.1016/j.eswa.2024.123715

[6] Shmueli, R., Mishra, D., Shmueli, T., & Hadar, O. (2024). A novel technique for image steganography based on maximum energy seam. Multimedia Tools and Applications, 83(28), 70907–70920. https://doi.org/10.1007/s11042-024-18476-6

[7] Al-Laham, M. K., El-Emam, N., & Qaddoum, K. (2025). Data hiding and extraction using pseudo-random generation and cover image replication. *International Journal of Data and Network Science*, *9*(4), 993–1010. https://doi.org/10.5267/j.ijdns.2024.10.005

[8] Lawrence, J., Budagam, D. K., Mukilan, P., Veerappan, K., Chandrasekhar, E., and Barathi, K. (2025). Enhanced Security Protocol for VLSI Systems: Modified AES Algorithm for Robust Data Transmission, 2025 *3rd International Conference on Integrated Circuits and Communication Systems (ICICACS)*, Raichur, India, 2025, pp. 1-5. doi: 10.1109/ICICACS65178.2025.10968738.

[9] Alam, S. T., Hassan, M. M., Bhuiyan, T. (2025). An Enhanced Image Steganographic Algorithm Integrating 1-Bit LSB-Based XOR Operations in the YCbCr Color Model with Canny Edge Detection, *2025 8th International Conference on Information and Computer Technologies (ICICT)*, Hawaii-Hilo, HI, USA, 2025, pp. 467-472.

[10] Rahman, S., Uddin, J., Zakarya, M., Hussain, H., Khan, A. A., Ahmed, A., & Haleem, M. (2023). A comprehensive study of digital image steganographic techniques. *IEEE Access*, 11, 6770–6791. https://doi.org/10.1109/access.2023.3237393

[11] Bui, T., Agarwal, S., Yu, N., & Collomosse, J. (2023). RoSteALS: Robust Steganography using Autoencoder Latent Space. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 933–942. https://doi.org/10.1109/cvprw59228.2023.00100

[12] Han, G., Lee, D., Hur, J., Choi, J., & Kim, J. (2023). Deep Cross-Modal steganography using neural representations. 2022 IEEE *International Conference on Image Processing (ICIP)*, 1205–1209. https://doi.org/10.1109/icip49359.2023.10222113

[13] Gan, Q. W., Yau, W., Gan, Y., Salam, I., Guo, S., Chang, C., Wu, Y., & Zhou, L. (2023). DSteganoM: Deep steganography for motion capture data. *Expert Systems with Applications*, 238, 121955. https://doi.org/10.1016/j.eswa.2023.121955

[14] Adi El-Dalahmeh, A., Alia, M., (2025). Enhancing VANET Security with Lattice-Based Cryptography and Dynamic Pseudonym Updates. *International Journal of Advances in Soft Computing and its Application*, 17(2), 232-246 https://doi.org/10.15849/IJASCA.250730.12

[15] Yin, L., & Li, Y. (2024). Data-Segmentation verification and a target generative adversarial network: EEG-Based Emotion Recognition. *Journal of Computational and Cognitive Engineering*, 3(4), 421–433. https://doi.org/10.47852/bonviewjcce42022571

[16] Kumar, A. (2024). Genetic whale optimisation algorithm for solving travelling salesman problem. *International Journal of Artificial Intelligence and Soft Computing*, 8(2), 109–128. https://doi.org/10.1504/ijaisc.2024.139608

[17] Cheng, L., Wang, Z., Jiang, F., & Li, J. (2020). Adaptive neural network control of nonlinear systems with unknown dynamics. *Advances in Space Research*, 67(3), 1114–1123. https://doi.org/10.1016/j.asr.2020.10.052

[18] Hussain, A., Sanaa Mohsin S., Alsaedi, H., Tashtoush, M., (2025). Parameter Estimation in Linear Rate Process: Modified Maximum Likelihood Estimation Using Particle Swarm Optimization, Least Square Estimation, and Simulation Methods . *International Journal of Advances in Soft Computing and its Application*, 17(2), 354-377 https://doi.org/10.15849/IJASCA.250730.19

[19] Zan, J. (2022). Research on robot path perception and optimization Technology based on whale Optimization Algorithm. *Journal of Computational and Cognitive Engineering*, 1(4), 201–208. https://doi.org/10.47852/bonviewjcce597820205514

[20] Shi, C., Wang, B., Chen, J., Zhong, R., Guo, S., Sun, P., & Ma, Z. (2022). Bending force of hot-rolled strip based on improved whale optimization algorithm and twinning support vector machine. *Metals*, 12(10), 1589. https://doi.org/10.3390/met12101589

[21] Chen, F., Fu, H., Yu, H., & Chu, Y. (2023). Using HVS Dual-Pathway and Contrast Sensitivity to Blindly Assess Image Quality. *Sensors*, *23*(10), 4974. https://doi.org/10.3390/s23104974

[22] Hardan, H., Alawneh, A., & El-Emam, N. N. (2022). New deep data embedding and extraction algorithm using multi-channel with multi-level to improve data security and payload capacity. *PeerJ Computer Science*, 8, e1115. https://doi.org/10.7717/peerj-cs.1115

[23] Xiong, Y., & Shen, Y. (2023). A High-Capacity adaptive image steganography algorithm based on Three-Shell matrices. Proceedings of the 2021 5th *International Conference on Electronic Information Technology and Computer Engineering*, 297–302. https://doi.org/10.1145/3650400.3650448

[24] Bandyopadhyay, S., Mukherjee, S., Mukhopadhyay, S., & Sarkar, S. (2024). Parallel BFS through pennant data structure with reducer hyper-object based data embedding for 3D mesh images. *Security and Privacy*, 7(5). https://doi.org/10.1002/spy2.390

[25] Elshare, S., El.Emam, N. (2022). Modified multi-level steganography to enhance data security. *International Journal of Communication Networks and Information Security (IJCNIS)*, 10(3). https://doi.org/10.17762/ijcnis.v10i3.3614

[26] Ahmad, M., El-Emam, N., Al-Azawi, A. (2022). Improved Deep Embedding/Extraction algorithm to enhance the payload capacity and security level of hidden information. *International Journal of Communication Networks and Information Security (IJCNIS)*, 13(3). https://doi.org/10.17762/ijcnis.v13i3.4759

[27] Ali, A. S., Alsamaraee, S., & Hussein, A. A. (2024). Optimize image steganography based on distinction disparity value and HMPSO to ensure

confidentiality and integrity. *Journal of Computer Networks and Communications,* 2024(1). https://doi.org/10.1155/2024/2516567

[28] Hawamdeh, M. A. Alsalahat, A., Jaradat, G., Al Shugran, M, Ayasrah, R. (2024). Kidney Disease Prediction using Elitist-Ant System Algorithm with Multilayer Perceptron Neural Network. *International Journal of Advances in Soft Computing and Its Applications*, *16*(2). https://doi.org/10.15849/ijasca.240730.11