IMPROVED STEGANOGRAPHIC SECURITY BY APPLYING AN IRREGULAR IMAGE SEGMENTATION AND HYBRID ADAPTIVE NEURAL NETWORKS WITH MODIFIED ANT COLONY OPTIMIZATION

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

[1]Department of Computer Science, Philadelphia University, Jordan
[2]Department of Computer Engineering, Warwick University, UK

ABSTRACT

In this paper, a new steganography algorithm has been suggested to enforce the security of data hiding and to increase the amount of payloads. This algorithm is based on four safety layers; the first safety layer has been initiated through compression and an encryption of a confidential message using a set partition in hierarchical trees (SPIHT) and advanced encryption standard (AES) mechanisms respectively. An irregular image segmentation algorithm (IIS) on a cover-image (Ic) has been constructed successfully in the second safety layer, and it is based on the adaptive reallocation segments' edges (ARSE) by applying an adaptive finite-element method (AFEM) to find the numerical solution of the proposed partial differential equation (PDE). An intelligent computing technique using a hybrid adaptive neural network with a modified ant colony optimizer (ANN_MACO) has been proposed in the third safety layer to construct a learning system. This system accepts entry using support vector machine (SVM) to generate input patterns as features of byte attributes and produces new features to modify a cover-image.

The significant innovation of the proposed novel steganography algorithm is applied efficiently on the forth safety layer which is more robust for hiding a large amount of confidential message reach to six bits per pixel (bpp) into color images. The new approach of hiding algorithm works against statistical and visual attacks with high imperceptible of hiding data into stego-images (Is). The experimental results are discussed and compared with the previous steganography algorithms; it demonstrates that the proposed algorithm has a significant improvement on the effect of the security level of steganography by making an arduous task of retrieving embedded confidential message from color images.

KEYWORDS

Image segmentation, steganography, adaptive neural network, ACO, finite elements.

1. INTRODUCTION

In the past years, steganography, which is a technique and science of information hiding, has been matured from restricted applications to comprehensive deployments. The steganographic covers have been also extended from images to almost every multimedia. From an opponent's perspective steganalysis [1], is an art of deterring covert communications while avoiding affecting the innocent ones. Its basic requirement is to determine accurately whether a secret message is hidden in the testing

DOI : 10.5121/ijnsa.2015.7502

medium. It also extracts the hidden message. Steganography and steganalysis are in a hide-and-seek game [1]. They grow with each other. Digital images have a high degree of redundancy in presentations in everyday life, thus appealing for hiding data. As a result, the past decade has seen growing interests in researches on image steganography and image steganalysis [1-4]. To evaluate the performance of categories of steganographic, three common requirements, security, capacity, and imperceptibility, may be used to rate the performance of steganographic techniques. Steganography may suffer from many active or passive attacks. Steganography must be useful in conveying a secret message, the hiding capacity provided by steganography should be as high as possible, and stego-images (Is) should not have severe visual artifacts. Least Significant Bit (LSB) based steganography. LSB based steganography is one of the straight techniques capable of hiding large secret message in a cover-image (Ic) without introducing many detectable biases [5]. It works by replacing the LSBs of randomly selected pixels in the cover-image with the secret message bits, where a secret key may determine the selection of pixels. Stenographic usually takes a learning based approach, which involves a training stage and a testing stage, where a feature extraction step is used in both training and testing stage. Its function is to map an input image from a high-dimensional image space to a low-dimensional feature space. The aim of the training stage is to obtain a trained classifier. Many effective classifiers, such as Fisher Linear Discriminant (FLD), support vector machine (SVM), neural network (NN), etc., can be selected. Decision boundaries are formed by the classifier to separate the feature space into positive regions and negative regions with the help of the feature vectors extracted from the training images.

A rapidly growing of steganalysis algorithms has discussed by many researchers, in particular, Li et al. [6] exploited unbalanced and correlated characteristics of the quantization-index (codeword) distribution, and presented a state-of-the-art steganalysis based on a support vector machine (SVM), which can detect the steganography with precision and recall levels of more than 90%. Therefore, a smaller change in the cover image is less detectable and more secure and resisted the steganalysis [7].

In recent years, some researchers in the data embeddings were using an intelligent algorithm based on soft computing. Such algorithms are used to achieve robust, low cost, optimal and adaptive solutions in data embedding problems. Fuzzy Logic (FL), Rough Sets (RS), Adaptive Neural Networks (ANN), Genetic Algorithms (GA) Support Vector Machine (SVM), Ant Colony, and Practical Swarm Optimizer (PSO) etc. are the various components of soft computing, and each one offers specific attributes [8]. A data embedding scheme by using a well-known GA-AMBTC based on genetic algorithm, block truncation code and modification direction techniques was proposed by Chin-Chen Chang et al. [9] (2009) to embed secret data into compression codes of color images. Yi-Thea Wu and Shih, F.Y [10] (2006) presents an efficient concept of developing a robust steganographic system by artificially counterfeiting statistic features instead of the traditional strategy of avoiding the change of statistic features. This approach is based on genetic algorithm by adjusting gray values of a cover-image while creating the desired statistic features to generate the stego-image that can break the inspection of steganalytic systems. M. Arsalan et.al. [11] developed an intelligent reversible watermarking approach for medical images by using GA to make an optimal tradeoff between imperceptibility and payload through effective selection of threshold. Modified Particle Swarm Optimization algorithm (MPSO) was introduced by (EL-Emam, 2015 [12]) used to improve the quality of stego-image by deriving an optimal change on the lower nibbles of each byte at sego-image. Fan Zhang et al. [13] (2008) proposed a new method of information-embedding capacity bound's analysis that is based on the neural network theories of attractors and attraction basins. Blind detection algorithms, used for digital image steganography were reviewed by Xiangyang Luo et al. [14] (2009); this approach is based on image multi-domain features merging and BP (Back-Propagation) neural network. Weiqi Luo et al. [15] (2010) applied LSB matching revisited image steganography and propose an edge adaptive scheme which can select the embedding regions according to the size of confidential message Φ and the difference between two consecutive pixels in the cover-image.

This paper proposes a new algorithm of data embedding using hybrid adaptive neural networks with an adaptive genetic algorithm based on a new version of adaptive relaxation named uniform adaptive relaxation ANN_MACO. With this algorithm, a large amount of data can be embedded into a color bitmap image with four safety layers.

The rest of the paper is structured as follows: In section 2, the proposed steganography algorithm with four safety layers has been discussed. Phases of the proposed steganography algorithm based on ANN_MACO are appearing in section 3. In section 4, the intelligent technique based on adaptive neural networks and modified ant colony algorithms have been discussed, and the the implementation of the proposed steganography with intelligent techniques is presented in section 5. Results' and discussions are reported in section 6. Finally, section 7 summarizes the algorithm's conclusions.

2. THE SUGGESTED STEGANOGRAPHY ALGORITHM SPECIFICATIONS

The new steganography algorithm has been proposed to hide a large aggregate of secret data using four safety layers, see Fig 3. The first three layers were suggested in a previous work [16]. However, the primary three layers of this work have been matured, and an extra layer is added as fourth safety layer based on adaptive neural networks ANN with meta-heuristic approach using MACO for tight security. It is essential to define the main specifications of the suggested new steganography algorithm:

2.1. Compression and encryption of confidential message

Compression and Encryption functions have been applied on a confidential message (Φ_C) at the sender side; these functions support the first safety layer of the proposed hiding algorithm. The formal definitions of both functions are explained in the following:

Definition 1: Let MC_{SPIHT} is a lossless message compression using a set partition in hierarchical trees (SPIHT) mechanism describe by the map $MC_{SPIHT} : \Phi \times Lim \times Lis \times Lsm \rightarrow \Phi_C$, where (Lim) is the list of insignificant message information, (Lis) the list of insignificant sets, (Lsm) is the list of significant messages, and Φ_C is a compressed confidential message.

Definition 2: Let ME_{AES} is a message encryption using advanced Encryption Standard (AES) mechanism define by the map $ME_{AES} : \Phi_C \times \ell_{\Phi_C} \rightarrow \Phi_{EC}$, where ℓ_{Φ_C} is the length of a compressed confidential message, and Φ_{EC} is a compressed and encrypted confidential message.

2.2. Image segmentation

Image segmentation is shown in Fig. 1 and applied in the second safety layer; it bases on a cipher key κ and the Adaptive Reallocation Segments' Edges (ARSE) as the following definitions.

Definition 3: Let χ_{IIS}^{AFEM} is an irregularly image segmentation function defined by the map $\chi_{IIS}^{AFEM} : \bar{I}_{\zeta}^{HN} \times \eta \times \kappa \times \Gamma_{PDE} \rightarrow \psi \times \bar{I}_{\zeta}^{HN}$

where

AFEM: is an adaptive finite-element method using to find a numerical solution of the proposed partial differential equations (Γ_{PDE}) to produce irregular segments.

η : It is a list of coordinates that represent the initial segments based on a cipher key κ.

ψ : It is a list of coordinates that represent an irregularly segmentation.

$\bar{I}_{\zeta}^{HN}$: It stands for sorted image based on high nibble bytes (HN) of a cover-image with normalization such that, $\bar{I}_{\zeta}^{HN} \in [0,1]$, and using normalization function f_{Norm} defined by the map $f_{Norm} : I_{\zeta}^{HN} \rightarrow \bar{I}_{\zeta}^{HN}$ to generate $\bar{I}_{\zeta}^{HN}$ images.

I_{ζ}^{HN}: It is a sorted image without normalization, which is generated by the sorted map $S : I_{C}^{HN} \rightarrow I_{\zeta}^{HN}$.

Definition 4: Let P is a projection function defines by the map $P : \psi \times \bar{I}_{\zeta}^{HN} \times I_{C}^{HN+LN} \rightarrow \psi \times I_{C}^{HN+LN}$, where the purpose of this function is to get the edges of an irregular segmentations from a sorted image $\bar{I}_{\zeta}^{HN}$ and project them on a cover-image $\psi \times I_{C}^{HN+LN}$.

An irregularly image segmentation shows that, it is safer to bring the input information than uniform segments due to the difficulty of catching the segment's borders by steganalysis.

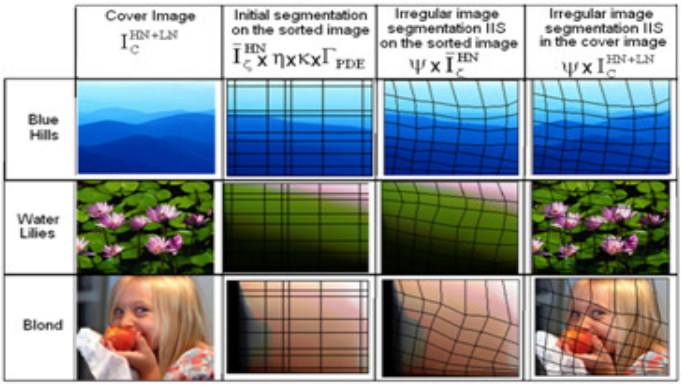

Figure 1. Using χ_{IIS}^{AFEM} on colour images

2.3. An intelligent technique

The modification of a cover-image $\Psi \times I_{C}^{LN+HN}$ has been reached using an intelligent technique based on adaptive neural network with a modified ant colony optimizer (ANN_MACO). The main concept of the proposed intelligent technique is to modify a cover-image according to the form of Φ_{EC} this is appeared at t this rd safety layer.

Definition 5: Let the map $L_{ANN\_MACO} : F \rightarrow F'$ is the learning function bases on ANN_MACO, where F is a feature of byte attributes based on three parameters, the third and fourth bits at the low nibble in a cover-image $I_{C}^{LN_{2,3}}$, two bits from secret message $\Phi_{EC_{pair}}$, and two bits from cipher key κ_{pair}. These features are selected using support vector machine (SVM) defines by the map $SVM : \Psi \times I_{C}^{LN+HN} \times \Phi_{EC} \times \kappa \rightarrow F$ see Eq.(1), such that,

$$F = \left\{ \Psi \times I_C^{LN_{2,3}}, \Phi_{EC_{pair}}, \kappa_{pair} \right\} \quad (1)$$

The new features of bytes attributes F' include a set of bits that are used by hiding algorithm ($\hbar_{pair}$) to place at the first and the second least significant bits from each byte $I_C^{LN_{0,1}}$ (pair of bits from low nibble) see Eq.(2a).

$$F' = \bigcup_{\forall \text{ byte}} \overline{\kappa}_{pair}^{byte} \quad (2a)$$

where κ_{pair} is a pair of two bits selected from cipher key κ and $\overline{\kappa}_{pair}^{byte}$ is defined according to the proposed formula, see Eq.(2b):

$$\overline{\kappa}_{pair} \leftarrow \left(\left(I_C^{LN_{2,3}} \oplus \kappa_{pair} \right) \oplus \Phi_{EC_{pair}} \right) \oplus \kappa_{pair} \quad (2b)$$

where $\Phi_{EC_{pair}}$ is a sequence of two bits from encrypted and compressed secret message.

2.4. Data hiding

The hiding algorithm is used at the fourth safety layer by accepting a modified cover-image and produces a stego-image $\Psi \times I_S^{LN+HN}$. We suggested new idea of image steganography according to the following definition.

Definition 6: Let $\hbar_{pair}$ is the proposed hiding function based on two least significant bits, and it defines by the map $\hbar_{pair} : I_{\tilde{C}}^{LN+HN} \times \Psi \times \Phi_{EC} \times F' \rightarrow I_S^{LN+HN}$, see Fig. 2.

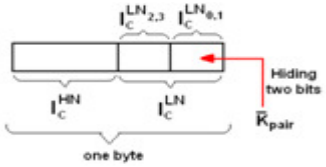

Figure 2. Hiding process using two least significant bits

2.5. Compression of stego-image

Lossless image compression using SPIHT algorithm is implemented on stego-image to avoid sending huge file size.

Definition7: A lossless image compression function (IC_{SPIHT}) is defined by the map $IC_{SPIHT} : I_S^{LN+HN} \times Lip \times Lis \times Lsp \rightarrow I_S^C$ using SPIHT algorithm, where (Lip, Lis, and Lsp) are defined as in (**Definition 1**), and I_S^C is the compression of a stego-image.

2.6. Decompression of stego-image

Lossless image decompression using SPIHT algorithm is implemented on I_S^C to avoid receiving a huge file size.

Definition8: A lossless image decompression function (ID_{SPIHT}) is defined by the map $ID_{SPIHT} : I_S^C \times Lip \times Lis \times Lsp \rightarrow \Psi \times I_S^{LN+HN}$ using SPIHT algorithm.

2.7. Data extracting

Data extraction algorithm is used at the receiver side; it accepts stego-image and produce secret message Φ.

Definition9: Let E_{pair} is the extracted function to produce two bits from each byte, and it defines by the map $E_{pair} : \Psi \times I_S^{LN+HN} \rightarrow \Phi_{EC}$ such that $\Phi_{EC_{pair}}$ is calculated according to the following mathematical formula, see Eq. (3):

$$\Phi_{EC_{pair}} \leftarrow \overline{\kappa}_{pair} \oplus I_C^{LN_{2,3}} \tag{3}$$

2.8. Decompression and decryption of a secret message

Decompression and decryption functions on compressed and encrypted secret messages ($M\Phi_{EC}$) are applied at the receiver side of the proposed system. The formal definitions of both functions are defined in the following:

Definition10: Let MD_{SPIHT} is a lossless message decompression using a set partition in hierarchical trees (SPIHT) mechanism describe by the map $MD_{SPIHT} : \Phi_{EC} \times Lim \times Lis \times Lsm \rightarrow \Phi_E$, where Φ_E is an encrypted confidential message.

Definition11: Let MDE_{AES} is a message decryption function using advanced encryption standard (AES) mechanism approach, and it defines by the map $MDE_{AES} : \Phi_E \times \ell_{M\Phi_E} \times \kappa \rightarrow \Phi$, where ℓ_{Φ_E} is the length of a decrypted confidential message.

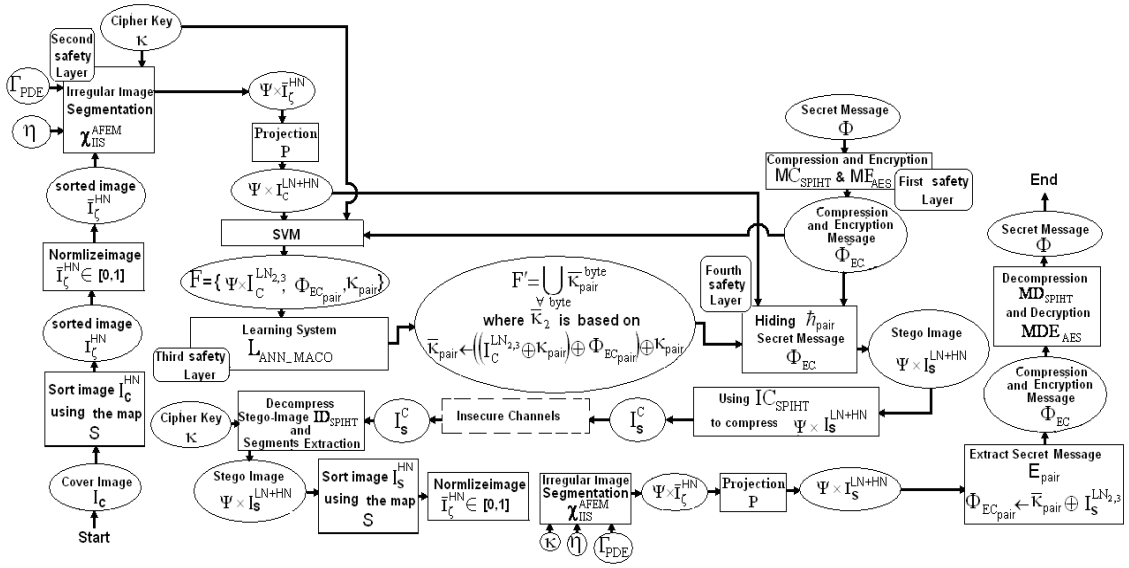

Figure 3. Steganography with four safety layers

3. THE SUGGESTED STEGANOGRAPHY ALGORITHM

The present steganography algorithm has two phases (data embedding at the sender side and data extracting at the receiver side). These phases have been constructed and implemented to reduce the chances of statistical detection and provide robustness against a variety of image manipulation attacks. After embedding data, stego-image is produced, which does not have any distortion artifacts. Moreover, the new steganography algorithm must not sacrifice an embedding capacity in order to decrease the perceptible of data embedding.

3.1. General design of hiding algorithm

The first phase is used to hide Φ_{EC} into I_C according to the following general steps:

Step 1: Input I_C, κ, Γ_{PDE}, η and Φ;

Step 2*:* Apply ME_{AES} and MC_{SPIHT} to encrypted and compressed Φ to produce Φ_{EC};

Step 3: Perform S on the HN of I_C to Construct sorted image I_ζ^{HN};

Step 4: Normalize I_ζ^{HN} to produce $\bar{I}_\zeta^{HN} \in [0,1]$;

Step 5*:* Perform χ_{IIS}^{AFEM} to define an irregular image segmentation Ψ on sorted image $\bar{I}_\zeta^{HN}$ to produce $\Psi \times \bar{I}_\zeta^{HN}$**;**

Step 6: Apply a projection function **P** to generate cover-image with an irregular segments' boundaries, $\Psi \times I_C^{LN+HN}$**;**

Step 7: Apply SVM to extract features F of byte attributes which includes a cover-image $\Psi \times I_C^{LN_{2,3}}$, a secret message Φ_{EC}, and cipher key κ;

Step 8: Implement learning system L_{ANN_MACO} to modify bytes' attributes and produce new features F'; *// see definition 5.*

Step 9*:* Generate stego-image Ψ, I_S^{LN+HN} by using hiding function $\hbar_{pair}$ of secret message Φ_{EC} on a modified cover-image $\Psi \times I_C^{LN_{2,3}}$. This process is done by replacing two bits from each low byte nibble of $\Psi \times I_C^{LN}$; // *see section 3.2.*

Step 10*:* Apply IC_{SPIHT} to find compressed stego-image I_S^C.

Step 11: Send I_S^C to insecure channel.

End.

The second phase is used to extract data from bitmap image at the receiver side in conformity with the following steps:

Step 1: Apply ID_{SPIHT} on I_S^C to generate I_S^{LN+HN};

Step 2: Perform S on a I_S^{HN} to Construct sorted image I_ζ^{HN};

Step 3 Normalize I_ζ^{HN} to produce $\bar{I}_\zeta^{HN} \in [0,1]$;

Step 4: Apply χ_{IIS}^{AFEM} on $\bar{I}_\zeta^{HN}$ to find segments' boundaries of stego-image $\Psi \times \bar{I}_\zeta^{HN}$;// *see section. 3.2;*

Step 5: Apply a projection function **P** to generate stego-image with an irregular segments' boundaries, $\Psi \times I_S^{LN+HN}$**;**

Step 6*:* Scanning all bytes from each color and then apply E_{pair} function to extract two bits from each byte;

Step 7*:* Gathering all extracted bits to produce Φ_{EC};

Step 8: Apply MD_{SPIHT} and MDE_{AES}on Φ_{EC} to find a confidential message Φ;

End.

3.2. New image segmentation (χ_{IIS}^{AFEM}) function.

An Irregular image segmentation function χ_{IIS}^{AFEM} has been applied to improve steganographic security; this function is based on (ARSE) to reallocate segments' edges; where the segments' edges have been calculated by solving the suggested two-dimensional partial differential equation PDE on a sorted image I_ζ, which is created from cover-image. The proposed algorithm has been summarized in the following steps:

Step 1: Input cover-image I_C and input cipher key κ;

Step 2: Create a sorted image I_ζ from a cover-image I_C by sorting color of each column in ascending order using the sorting map $S : I_C \times \kappa \rightarrow I_\zeta$;

Step 3: Using κ to construct polynomial function Poly(r_i) , see Eq. (4a). This function has been applied to find set of pixels $\{pixcel(r_1, c_1), pixcel(r_2, c_2), ..., pixcel(r_m, c_m)\}$ and using a set of pixels to define segments' boundaries.

$$c_i = \sum_{j=1}^{m} a_j r_i^j \quad , \quad \forall\, i = 1,..., \quad m \tag{4a}$$

where the coefficients $a_1, a_2, \ldots, a_m$ have been extracted from κ and equal to the and decimal value of κ's symbols, see Eq. (4b):

$$a_i = \mathrm{Dec}(\kappa_i), \forall i = 1,...,m \tag{4b}$$

Moreover, the constant (m) represents the length of a cipher key κ , while the product $(m \times m)$ represents a number of segments in the image, see Fig. 4. The concatenation $(\|_{i=1}^{m})$ of the ASCII code for the coefficients a_i , $\forall i = 10 \ldots m$ is closed to κ , see Eq.(5).

$$\|_{i=1}^{m} \mathrm{ASCII}(a_i) = \kappa, \qquad \text{and} \qquad m = |\kappa| \tag{5}$$

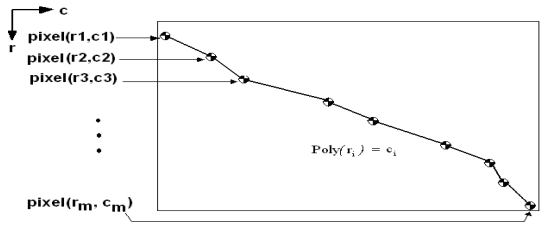

Figure 4. Applying polynomial to set the boundary of the initial segmentation

Step4: Normalize pixels' values of a sorted image I_{ζ}^{HN} to produce $\bar{I}_{\zeta}^{HN}$ which includes pixels in the interval [0, 1] ; *// see Eq (6):*

$$I_{\zeta} = a + (b - a) I'_{\zeta} \ , \ \text{where} \ \ a = 0 \ \text{and} \ \ b = 1 \tag{6}$$

Step 5: Construct the initial segments by using the boundary of the initial segmentation at the **step 3** on the normalized image; *// See Fig. 5;*

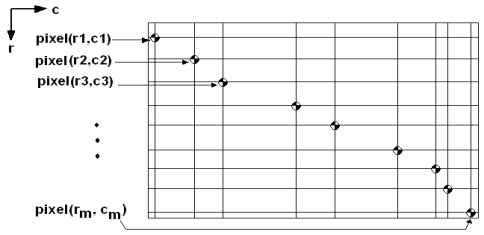

Figure 5. Initial segmentation using selected pixels

Step 6: Apply χ_{IIS}^{AFEM} using Adaptive Finite Element Method AFEM on the proposed PDE Eqs (8, 9). This method is used to construct pattern by moving edges of segments by solving Γ_{PDE} with specific number of iteration equal to Total Fig. 6; *// see Eq.(7)* .

$$\mathrm{Total} = \sum_{j=1}^{m} \mathrm{Dec}\ (\kappa_j) \tag{7}$$

where, $\mathrm{Dec}(\kappa_j)$ represents the decimal value of the j^{th} character at the κ.

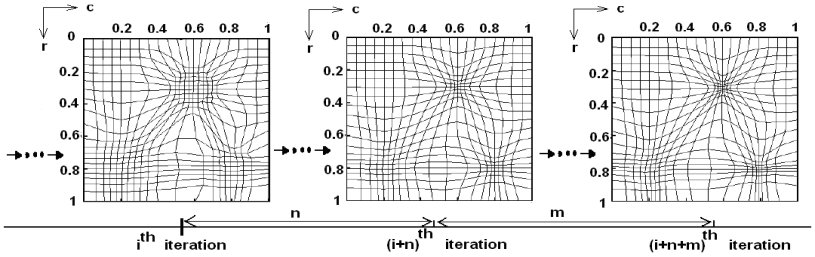

Figure 6. Segments' edges of the sorted image through the steps of iterations

Step 6.1: Set the initial two-dimensional coordinates (R, C*)* for each segment of the sorted image $\bar{I}_{\zeta}^{HN}$ and using the proposed Γ_{PDE} model, see Eqs(8, 9) to govern the moving edge points *(*r, c) of image's segments;

$$\nabla^2 C - (I'_r C_c - I'_c C_r) = 0. \quad (8)$$

$$\nabla^2 R - (I'_r R_c - I'_c R_r) = 0 \quad (9)$$

where C and R are a two-dimensional coordinates for the row and the column directions respectively. The first derivative I'_r and I'_c are equal to $\frac{\partial \bar{I}_{\zeta}^{HN}}{\partial r}$ and $\frac{\partial \bar{I}_{\zeta}^{HN}}{\partial c}$ respectively, while C_c, C_r, R_c and R_r are the first derivative of coordinates, which are equal to $\frac{\partial C}{\partial c}$, $\frac{\partial C}{\partial r}$, $\frac{\partial R}{\partial c}$, and $\frac{\partial R}{\partial r}$ respectively. The proposed mathematical model is based on second-order partial differential equation (PDE) defined in the Eqs. (8, 9), these equations have been constructed using two terms, the diffusion and nonlinear convection terms.

Step6.2 Apply the numerical method using AFEM to solve Eqs (8-9) numerically to find the segment's edges $\Psi \times \bar{I}_{\zeta}^{HN}$;

Step6.3 Projection P the segments' edges $\Psi \times \bar{I}_{\zeta}^{HN}$ on the cover-image to produce $\Psi \times I_C^{LN+HN}$; *// see Fig. 7;*

End.

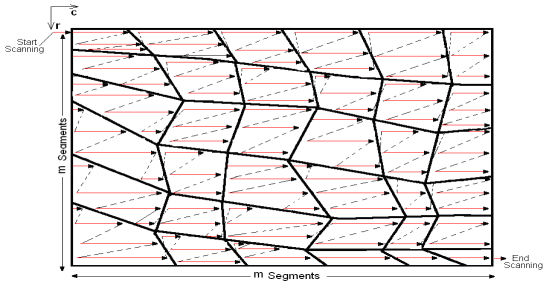

Figure 7. Scanning pixels on the adaptive image's segments

Using irregular segments to hide secret message Φ randomly instead of sequentially and this approach is playing the basic role to reduce the probability of detecting secret message Φ into $\left(\frac{1}{\sigma^2(s)+m^2}\right)$, where $\sigma^2(s)$ is the variance of segments' sizes.

3.2.1. Numerical solution using AFEM

AFEM is applied to find the numerical formulas defining in Eqs. (8-9). However, these formulas are always subject to evaluation with regards to the satisfactory security. The numerical solution of IIS has been reached by solvating both Eqs (8-9) simultaneously with a specific number of iteration using cipher key κ . Consequently, it becomes necessary to modify FEM to reduce the time and memory requirements.

AFEM with modified Newton's method is used to find the variation-vectors δR and δC and from Eq. (8), considering the C coordinated and using the weighted residual method, we get:

$$\sum_{k=1}^{AD} \iint_{\Omega} N_k \left(\nabla^2 C - (I'_r C_c - I'_c C_r)\right) d\Omega = 0. \tag{10}$$

where, N_k is a weighted function based on an adaptive degree (AD) of Lagrange polynomial using a variation of colors in one segment [8], and the degree of polynomial is calculated according to the new approach using Eq. 11.

$$AD_i = 4 + \text{int}\left(\sigma^2(s_i)\right) \quad \forall i = 1,..., m^2 \tag{11}$$

where s_i is the i^{th} segment and $\text{int}\left(\sigma^2(s_i)\right) \in [0, \infty]$

Using Green's theorem on Eq.(10), the following is obtained:

$$\sum_{k=1}^{AD} \iint_{\Omega} \left(N_{kc} C_c^{n+1} + N_{kr} C_r^{n+1} + N_k (I'_r C_c^{n+1} - I'_c C_r^{n+1})\right) d\Omega = 0 \tag{12}$$

Let us define the following variations:

$$\delta C = C^{n+1} - C^n, \quad \delta C_c = C_c^{n+1} - C_c^n, \delta C_r = C_r^{n+1} - C_r^n \tag{13}$$

Using "Eq.(13)" in "Eq.(12)" and then simplifying, we get:

$$\begin{aligned} \sum_{sk} \iint_{\Omega} & \left[\left(N_{kc} \delta C_c + N_{kr} \delta C_r \right) + N_k \left(I'_r \delta C_c - I'_c \delta C_r\right)\right] d\Omega \\ & = -\sum_{sk} \iint_{\Gamma} \left[\left(N_{kc} + N_k I'_r\right) + \left(N_{kr} - N_k I'_c\right) C_r^n \right] d\Gamma . \end{aligned} \tag{14}$$

where s is the number of segments at the sorted image I_ζ. Now let us define the following approximations:

$$C_c = \sum_{i=1}^{4} N_{ic} \delta C_i, \ \delta C_r = \sum_{i=1}^{4} N_{ir} \delta C_i \ I'_c = \sum_{i=1}^{4} N_{ic} I'_i \ \ I'_r = \sum_{i=1}^{4} N_{ir} I'_i \tag{15}$$

where $I'_i \in \bar{I}_\zeta^{HN}$. Using iso-parametric segments to construct regular segments from irregular segments by using a normal coordinates (ξ, η) [8], and applying Gauss's quadrature on "Eq.(14)" for all segments in the sorted image $\bar{I}_\zeta^{HN}$ to produce the following terms:

$$\sum_{\text{all}} \omega^1 |J^1| \left[\left[(N_{k\xi}\xi_c + N_{k\eta}\eta_c)^1 (N_{i\xi}\xi_c + N_{i\eta}\eta_c)^1 + (N_k\xi_r + N_{k\eta}\eta_r)^1 (N_{i\xi}\xi_r + N_{i\eta}\eta_r)^1 \right] \right] = A'_{ki} . \quad (16)$$

$$\sum_{\text{all}} \omega^1 |J^1| \left[N_k \left[\sum_{j=1}^{4} (N_{j\xi}\xi_r + N_{j\eta}\eta_r)^1 I'_j (N_{i\xi}\xi_c + N_{i\eta}\eta_c)^1 - \sum_{j=1}^{4} (N_{j\xi}\xi_c + N_{j\eta}\eta_c)^1 P_j (N_{i\xi}\xi_r + N_{i\eta}\eta_r)^1 \right] \right] = A''_{ij} . \quad (17)$$

$$\sum_{\text{all}} \omega^1 |J^1| \left[\left\{ (N_{k\xi}\xi_c + N_{k\eta}\eta_c)^1 + N_k - \left[\sum_{j=1}^{4} (N_{j\xi}\xi_r + N_{j\eta}\eta_r)^1 I'_j \right] \right\} \sum_{i} (N_{i\xi}\xi_c + N_{i\eta}\eta_c)^1 \right] = A'''_{kji} . \quad (18)$$

$$\sum_{\text{all}} \left[(N_{k\xi}\xi_r + N_{k\eta}\eta_r)^1 - N_k \sum_{j} (N_{j\xi}\xi_c + N_{j\eta}\eta_c) I'_j \sum_{i=1}^{4} (N_{i\xi}\xi_r + N_{i\eta}\eta_r) \right] = \hat{A}_{kji} . \quad (19)$$

where, ω^1 is a weighting factor for integral approximation and $|J^1|$ is the determinant of the Jacobian matrix [8]. "Eqs. (16-19)" are used to build the following system of linear equations.

$$(A' + A'')\Delta C = (A''' + \hat{A}) C . \quad (20)$$

Multiply "Eq.(20)" by $(A+B)^{-1}$ to get:

$$\Delta C = (A' + A'')^{-1} (A''' + \hat{A}) C . \quad (21)$$

Now calculate the vector C^{new}.

$$C^{new} = C^{old} + \Delta C . \quad (22)$$

The same processes on "Eq. (9)" with respect to Y coordinate are used to obtain the R^{new} vector Eq. (23):

$$R^{new} = R^{old} + \Delta R . \quad (23)$$

4. THE INTELLIGENT TECHNIQUE USING ANN_MACO ARCHITECTURE

We proposed the intelligent technique; based on hybrid adaptive neural networks with a modified ant colony optimizer (ANN_MACO), see Fig. 8. In this work, ANN and MACO represented the third safety layer; this layer is introduced to support and the enhanced steganography algorithms by constructing an excellent imperceptible of I_S and working effectively against statistical and visual attacks. The proposed intelligent technique ANN_MACO includes (n-p-m) Perceptron layers' architecture; it has (n) neurons in input layer, (p) neurons in the hidden layer and (m) neurons in the output layer with full connections.

The solid arrow in Fig. 8 shows two kinds of transitions; one of them is many-to-one while the other is one to many transitions among Perceptron layers, whereas dotted arrow refers to one-to-one transition, and the dashed arrow shows the sending action to adjust a process. Back-propagation algorithm with hybrid ANN_MACO algorithm is applied through three stages: the feed forward of the input training pattern, the back-propagation of the associated error, and the adjustment of the weights. In addition, the adaptive smoothing error ASE is introduced effectively to speedup training processes [8]. Extra difficulties are added to work against statistical and visual attacks if new features of cover-image are used before hiding process.

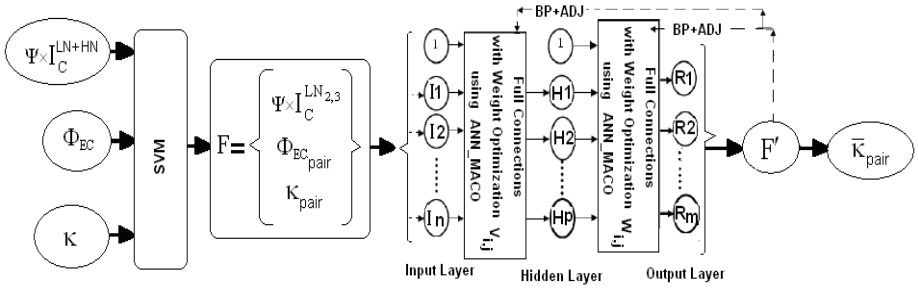

Figure 8. Learning system L_{ANN_MACO} using ANN_MACO architecture and SVM

4.1. Adaptive neural networks ANN with modified ant colony optimizations MACO approach

The adaptive neural networks ANN has been trained using back-propagation algorithm using three layers; these layers are: The input layer that includes n-neurons I_i ($\forall$ i =1,…,n), where this layer accepts feature's attributes F from cover-image $\left\langle \Psi \times I_C^{LN} \right\rangle$, secret message Φ_{EC} and Nbpb=2 then broadcasts them to the hidden layer. The hidden layer includes p-neurons H_j ($\forall$ j=1,…,p) , where this layer accepts weights V_{ij} by using the activation function $f(.)$, See Eq. (24).

$$H_j = f\left(\sum_{i=0}^{n} I_i V_{ij}\right) = \frac{1 - e^{-\sum_{i=0}^{n} I_i V_{ij}}}{1 + e^{-\sum_{i=0}^{n} I_i V_{ij}}}, \quad I_0 = 1, \quad \forall j = 1,..., p \tag{24}$$

Each hidden neuron computes its activation function $f(h)$ and sends its signal H_j to the output layer that includes m-neurons R_k ($\forall$k=1,…,m), where this layer accepts weights W_{jk}, where h is the activation function parameter of the hidden layer.

Each output neuron R_k computes its activation function $f(r)$ to form the response of the neural network as in Eq. (25), where r is the activation function parameter of the output layer.

$$R_k = f\left(\sum_{j=0}^{p} H_j W_{jk}\right) = \frac{1 - e^{-\sum_{j=0}^{p} H_j W_{jk}}}{1 + e^{-\sum_{j=0}^{p} H_j W_{jk}}}, \quad H_0 = 1, \forall k = 1,..., n \tag{25}$$

The activation function $f(.)$ applied in the training system is bipolar sigmoid defined in the range [-1, +1], see Eq. (26).

$$f(\gamma) = \frac{1 - e^{-\gamma}}{1 + e^{-\gamma}} \tag{26}$$

where γ is the activation function parameter and the first-order derivative of $f(.)$ is defined in Eq. (27).

$$\frac{d(f(\gamma))}{d\gamma} = \frac{1 - f^2(\gamma)}{2} \tag{27}$$

During the training, the set of output neurons represents the new features attributes F' of cover-image.

The new probabilities are used to check matching with probabilities t_k of cover-image as in Eq. (28), using δ_k ($\forall$ k=1… n) to compute the distribution error of neurons R_k at the output layer.

$$\delta_k = (t_k - R_k)\frac{d\left(f\left(\sum_{j=0}^{p} H_j W_{jk}\right)\right)}{dr} = (t_k - R_k)\left(\frac{1 - f^2\left(\sum_{j=0}^{p} H_j W_{jk}\right)}{2}\right) = (t_k - R_k)\left(\frac{1 - \left(\frac{1 - e^{-\sum_{j=0}^{p} H_j W_{jk}}}{1 + e^{-\sum_{j=0}^{p} H_j W_{jk}}}\right)^2}{2}\right), \forall k = 1,..., n \quad (28)$$

where β is damping parameter at the interval [0, 1] and in the same manner, the distribution error factor δ_j ($\forall$j=1,…, p) has been computed for each hidden neuron H_j. The adjustment of the weight W_{jk} is defined in Eq. (29) and it is based on both the distribution error factor δ_k and the activation of the hidden neuron h_j.

$$\Delta W_{jk}^{new} = \beta \alpha_{jk}^{new} \delta_k H_j + (1 - \beta).\Delta W_{jk}^{old} \quad (29)$$

The error factors are defined as in Eqs. (30). The distribution error of neurons R_k ($\forall$k=1,…,p) at the hidden neuron.

$$\delta_j = \sum_{k=1}^{m} \delta_k W_{jk}\left(\frac{d\left(f(\sum_{i=0}^{n} I_i V_{ij})\right)}{dh}\right) = \sum_{k=1}^{m} \delta_k W_{jk}\left(\frac{1 - \left(f(\sum_{i=0}^{n} I_i V_{ij})\right)^2}{2}\right) = \sum_{k=1}^{m} \delta_k W_{jk}\left(\frac{1 - \left(\frac{1 - e^{-\sum_{i=0}^{n} I_i V_{ij}}}{1 + e^{-\sum_{i=0}^{n} I_i V_{ij}}}\right)^2}{2}\right), \forall j = 1,..., p \quad (30)$$

The adjustment to the weight V_{ij} from the input neuron I_i to hidden neuron H_j is based on the factor δ_j and the activation of the input neuron as in Eq. (31).

$$\Delta V_{ij}^{new} = \beta\, \alpha_{ij}^{new} \delta_j I_i + (1 - \beta)\Delta V_{ij}^{old} \quad (31)$$

where β at Eqs(29, 31) is the damping parameter in the interval $\beta \in [0,1]$, in this work, we select $\beta = 0.1$. Update the value of weight functions using Eqs. (32, 33) which are based on the new optimization approach $f_{MACO}(.)$.

$$W_{jk}^{new} = f_{MACO}\left(W_{jk}^{old} + \Delta W_{jk}\right) \quad (32)$$

$$V_{ij}^{new} = f_{MACO}\left(V_{ij}^{old} + \Delta V_{ij}\right) \quad (33)$$

and using adaptive learning rate [8] to improve the speed of training by changing the rate of learning α during a training process, see Eq. (34).

$$\alpha_{jk}^{new} = \begin{cases} \alpha_{jk}^{old} + \lambda & \text{if} \quad \Delta W_{jk}^{new} \Delta W_{jk}^{old} > 0 \\ (1 - \varepsilon)\alpha_{jk}^{old} & \text{if} \quad \Delta W_{jk}^{new} \Delta W_{jk}^{old} < 0 \\ \alpha_{jk}^{old} & \text{otherwise} \end{cases} \quad (34)$$

The suitable values of parameters λ and ε have been predicted. These values are equal to **0.016 and 0.82** respectively. The training processes in the proposed algorithm are repeated many times and

update the old values of weights, which are represented by two-dimensional arrays V and W The repetition of training is reached when the following condition is satisfied Eq. (35):

$$\underset{\forall k}{\text{Max}} \left| t_k - R_k \right|^2 < 10^{-6} \tag{35}$$

Finally, we introduce Adaptive Smoothing Error (ASE) to speed up training processes, see [8],[17].

MACO algorithm represented by the function $f_{MACO}(.)$ it is used to solve and wide variety of problems. It has been proven to be of notable usefulness in solving optimization problems of all kinds. The probability distribution of the trial parameters has been applied using new approach based on damping factor (ξ), where $\xi \in [0,1]$, see Eq.(37). The basic idea behind MACO is that an initial population of candidate states of size (n) is chosen at random, and each state is evaluated according to the function of the optimization.

Main Algorithm of $f_{MACO}(.)$

Step1: input the size of the population equal to n ; // *where n represents the number of neurons in the input layer of ANN.*

Step2: foreach (Iteration BP) *//using this iteration for Back-propagation* (*BP*) *algorithm*

Step2-2*:* foreach (Pattern P) // *using to move from pattern to next pattern of SVM*

Step 2-2-1*:* Create randomly an initial population of individual Weights $V_i , W_i , \forall i = 1,\ldots,n$;

Step 2-2-2*:* Apply training on the pattern (P) through the following steps;

Step 2-2-2-1 foreach (Iteration AC) *// using MACO for smoothing Weights* V_i , W_i .

Step 2-2-2-1-1 foreach (Colony v*) // each colony v represents the weight* V_i .

Step2-2-2-1-1-1 Call **Sub-Algorithm1**; // *Apply ACO on the colony v.*

Step2-2-2-1-1-2 Set $V_i = \tau_{ij}$; *// save pheromone trails* τ_{ij} *to the weight* V_i

End foreach (v).

Step 2-2-2-1-2 foreach (Colony v) *// each colony v represents the weight* W_i .

Step 2-2-2-1-2-1: Call **Sub-Algorithm1**; // *Apply ACO on the colony v.*

Step2-2-2-1-2-2 : Set $W_i = \tau_{ij}$; *// save pheromone trails* τ_{ij} *to the weight* W_i .

End// *foreach (v)*; **End** *//foreach (AC)*; **End** *//foreach Pattern (P);* **End** *//foreach Iteration (BP).*

Step3 Call **Sub-Algorithm2**;

End Main algorithm .

Sub-algorithm1 // *Apply ACO on the colony* **v**.

Step1: Define all nodes and all arcs between nodes in the Colony **v***;*

Step2: Let **Na** is the number of ants using in the colony v*;*

Step3: Set the initial value to all pheromone trails τ_{ij} ;

Step4: Compute the priori desirability η_{ij} ; // η_{ij} *is the attractiveness by using heuristic information Eq. 36.*

$$\eta_{ij} = \frac{1}{d_{ij}} \quad \forall i, j \tag{36}$$

where d_{ij} is the distance between i and j.

Step5: foreach time **(t) //** *t=2,3,..*

Step5-1: foreach Ant (a) until all Ants have completed a solution (from starting node to the goal node.

Step5-1-1: foreach $arc(i, j) \notin tabu(a)$ in the Colony **v** *// tabu(a) is the set of infeasible arcs.*

Step5-1-1-1: compute the probability P_{ij} in Eq. (37);

$$P_{ij}=\begin{cases}\dfrac{\xi\times\left(\tau_{ij}^{\alpha}(t)\times\eta_{ij}^{\beta}(t)\right)+(1-\xi)\times\left(\tau_{ij}^{\alpha}(t-1)\times\eta_{ij}^{\beta}(t-1)\right)}{\sum_{\forall ik\notin tabu(a)}\left(\xi\times\left(\tau_{ik}^{\alpha}(t)\times\eta_{ik}^{\beta}(t)\right)+(1-\xi)\times\left(\tau_{ik}^{\alpha}(t-1)\times\eta_{ik}^{\beta}(t-1)\right)\right)} & arc(i,j)\notin tabu(a)\\ 0 & arc(i,j)\in tabu(a)\end{cases} \quad (37)$$

where α and β are two parameters to control the influence of pheromone trails τ_{ij} and priori desirability η_{ij} respectively and $\xi\in[0,1]$ is a damping parameter, and the values of $\tau_{ij}^{\alpha}(0)$,$\eta_{ij}^{\beta}(0)$, $\tau_{ij}^{\alpha}(1)$,$\eta_{ij}^{\beta}(1)$ are selected randomly.

Step5-1-1-2: Selected the next node using the probabilistic decision of the Ant (a) move from the node (i) to the node (j)**;** *//see Fig. 9*

$$j=\arg\max_{\ell\in N_i^a}(P_{i\ell}) \quad (38)$$

where N_i^a is the set of remaining nodes to be visited by the a^{th} Ant located at node i.

Step5-1-1-3: Append the chosen infeasible move(s) of the a^{th} ant to the set tabu(a).

Step5-1-1-4: Find the amount of trail $\Delta\tau_{ij}^{a}$ on each arc (i,j) chosen by Ant (a):

$$\Delta\tau_{ij}^{a}=\begin{cases}\dfrac{Q}{L_a} & \text{if Ant (a) uses arc (i, j) in its tour}\\ 0 & \text{otherwise}\end{cases} \quad (39)$$

where L_a is the length of the trail tour length by Ant (a) and Q is the constant parameter related to the quantity of trail laid by ants as trail evaporation.

Step5-1-1-5: Update the pheromone trails τ_{ij} $\forall$i,j using under relaxation based on evaporation coefficient ρ , see Eq. (40),

$$\tau_{ij}(t)=\rho\,\tau_{ij}(t-1)+(1-\rho)\Delta\tau_{ij} \quad (40)$$

where

$$\Delta\tau_{ij}=\sum_{a=1}^{m}\Delta\tau_{ij}^{a} \quad ,m\le Na \quad (41)$$

where m- is the current number of ants using in this step $\rho\in(0,1)$.

End //*foreach arc(i,j)*; **End //***foreach Ant (a);* **End //***foreach time (t).*

Step 6 return pheromone trails τ_{ij} ;

End Sub-algorithm1.

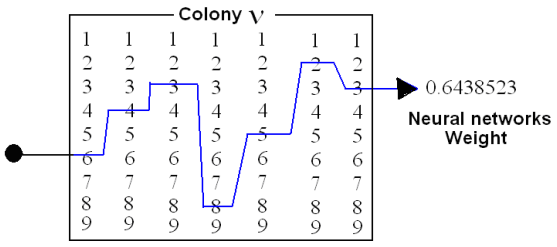

Figure 9. A solution path is a vector coded as seven significant decimal digits searching by ants to adjust Neural networks weights (W or V) for colony v.

Sub-Algorithm2 // *to implement Adaptive Smoothing Error ASE.*
Step1: Apply ASE through the following steps:

Step 1-1*:* For each pattern, find the sum of neuron's errors.
Step 1-2*:* select pattern's index that has the maximum of a sum of neuron's errors Eq. (42).

$$\underset{\forall P}{\text{Max}} \left| \sum_{\forall \text{ neurons}} (\text{Error}) \right| , \; P = 1,..., NP \tag{42}$$

where (NP) is a number of patterns.

Step 1-3: Go to **step 2-2-2** (in the Main algorithm) and set the variable (P) equal to the pattern's index at the **Step 1-2** in the **Sub-Algorithm2**.
End Sub-Algorithm2.

5. Implementation Of The Proposed Steganography Algorithm With Intelligent Technique.

Assume we have 9 bytes from Lina cover-image, the secret message $\Phi_{EC} = 000111001101011110$, and the cipher key $\kappa = 101111010011$ shown in Fig. 10. We should explain step by step how to hide a pair of secret bits $\Phi_{EC_{Pair}}$ at each byte using the proposed hiding algorithm with learning system based on the ANN-MACO. Assume that the κ_{pair} is the first two bits from κ. The pack of optional parameters of MACO have been obtained through several tests is as follows: $\alpha = 1, \beta = 3, \rho = 0.1, Q = 100$. Figure 10 shows that small difference between cover and stego sections has been obtained when hiding two bits on the least significant bits carried out using learning technique L_{ANN_MACO}; whereas hiding secret bits directly without using learning technique is incompetent due to large difference between cover and stego sections.

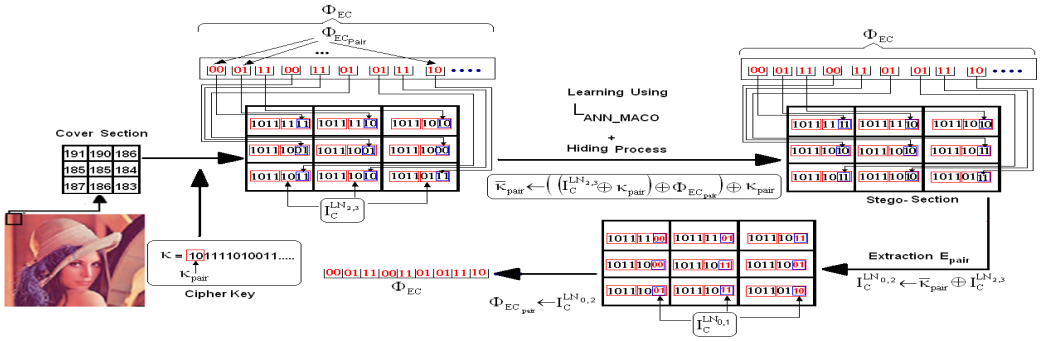

Figure 10. Steps to hide secret message

6. EXPERIMENTAL RESULTS AND DISCUSSIONS

New steganography algorithm is running efficiently to hide a large amount of Φ into a cover-image. The payload capacity reached to 25% of the I_C size; moreover, ANN_MACO has been introduced successfully to work against statistical and visual attacks and to modify the stego-image to become imperceptible to the human eye. More than 500 color images are used in this work to achieve training on the proposed system ANN_MACO and to perform comparisons between the proposed scheme and previous works. The results are discussed as follows:

6.1. The amount of payloads vs. image distortion:

Using the peak signal to noise ratio (PSNR dB) and structural similarity (SSIM) measurements to make sure the image quality after hiding data. PSNR has been calculated using Eq.(43).

$$PSNR = 10 \times \log_{10} \frac{\max^2}{MSE} \tag{43}$$

where max is the maximum pixel value, and MSE represents the average of mean square errors for RGB colors shown in Eq. (44)

$$MSE = \frac{MSE_R + MSE_G + MSE_B}{3} \tag{44}$$

and the MSE_R , MSE_G , MSE_B are the mean square of the three colors and is computed by using the following Eq.(45):

$$MSE_c = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left(C_{ij}^c - S_{ij}^c \right)^2 \quad ; c \in \{R, G, B\} \tag{45}$$

where (m x n) is the size of image and C^c, S^c are two bytes at the location (i,j) in the specific color c from the cover and stego images respectively. Five testing color images (512 x 512) have been used, namely "Baboon", "F16", "Lena", "Peppers and "Tiffany" shown in Fig. 9.

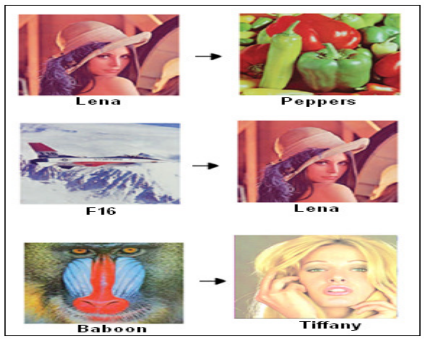

Figure 11. Stego-images and their corresponding extracted secret images

PSNR for each color plane (R, G, B) has been computed on three stego-images separately, and three secret images have been extracted from stego-images see Fig. 11. The result of the proposed algorithm based on ANN_MACO is compared with the El-Emam (2013) algorithm [17], it appears obviously that the quality of stego-image using the proposed scheme is working superior than the previous work, and it obtains better performance than the algorithm in [17] for all colors with an excellent imperceptibility see Table 1.

Table 1. PSNR(dB) results of Is on a color plane between El-Emam (2013) algorithm [17], Al-Shatanawi, (2015) [18], and the proposed algorithm

Stego-images 512x512	Using SPIHT compression on Secret image	PSNR(dB) of I_s on R-plane				PSNR(dB) of I_s on G- plane				PSNR(dB) of I_s on B- plane				PSNR(dB) of I_s (the average of three color Planes)			
		Ref. [17]	Ref. [18]	The proposed algorithm with/without ANN_MACO		Ref. [17]	Ref. [18]	The proposed algorithm with/without ANN_MACO		Ref. [17]	Ref. [18]	The proposed algorithm with/without ANN_MACO		Ref. [17]	Ref. [18]	The proposed algorithm with/without ANN_MACO	
				With-out	With			With-out	With			With-out	With			With-out	With
Lena	Peppers	47.24	42.97	39.01	48.21	47.65	42.94	39.42	48.14	46.22	43.05	39.98	47.22	47.03	42.96	39.47	47.85
F16	Lena	45.48	44.88	37.45	47.66	45.35	44.95	37.12	47.54	47.94	45.14	39.71	48.67	46.25	44.99	38.09	47.95
Baboon	Tiffany	47.44	42.45	39.21	47.88	46.22	42.4	37.98	47.14	45.41	42.57	37.17	46.68	46.35	42.47	38.12	47.23

The experimental results reported in Table 1 explained that the proposed algorithm with ANN_MACO adjusts an image visual quality significantly. Results with ANN_MACO improve the quality without ANN_MACO and the earlier work (El-Emam, 2013, [17]), and (Al-Shatanawi, 2015 [18]) respectively. Moreover, results are showing that PSNR of Lena's compression in F16's stego-image has a best quality with ANN_ MACO, but it is better than [17] with 1.7 dB, and better than [18] with 2.96 dB, whereas, Tiffany's compression in Baboon stego-image has a worst quality with ANN_ MACO, but it is better than [17] with 0.88 dB, and better than [18] with 4.76 dB.

Table 2 shows the comparison between the experimental results of the proposed hiding algorithm with/without ANN_MACO and the algorithm in [17]. The comparison is based on PSNR (dB) to demonstrate the visual quality after embedding Smsg, where Smsg is the largest size of the random bit stream generated randomly by using a random number generator.

Table 2 confirms that the quality of stego-image using the proposed algorithm is preserved and better than the algorithm in [17] for all colors. Where the best improvement was using the proposed algorithm with ANN_MACO for Lena's image over ref [17] where the PSNR improvement was 0.98, where Tiffany's image was improved with 0.16 PSNR when used with ANN_MACO proposed algorithm.

Table 2. PSNR(dB) results of Stego-image on a color plane between El-Emam (2013), [17] and the proposed algorithm for the payload capacity equal to 25%

Stego-image (512x512) includes random secret bits	PSNR(dB) of I_s on R-plane			PSNR(dB) of I_s on G-plane with maximum payload			PSNR(dB) of I_s on B-plane with maximum payload		
	Ref. [17]	The proposed algorithm with/without ANN-MACO		Ref. [17]	The proposed algorithm with/without ANN_MACO		Ref. [17]	The proposed algorithm with/without ANN_MACO	
		Without	With		Without	With		Without	With
Lena	47.24	40.21	48.22	46.22	38.68	47.14	46.22	40.68	47.24
F16	47.44	41.23	47.67	47.65	41.47	48.54	47.94	41.78	48.65
Baboon	45.48	39.65	47.23	45.35	38.32	47.14	45.40	38.16	46.78
Peppers	46.20	37.96	48.12	46.37	38.13	47.44	45.92	37.68	47.15
Tiffany	49.51	41.27	49.55	46.43	38.19	47.16	46.60	38.36	47.32

The SSIM algorithm [17] is used to measure the similarity between two identical images. In this work, this metric is introduced using Eq (46):

$$SIMM(I_c, I_s) = \frac{\left(2\mu_{I_c}\mu_{I_s} + \left((2^{24}-1)*0.01\right)^2\right)\left(2\sigma_{I_cI_s} + \left((2^{24}-1)*0.03\right)^2\right)}{\left(\mu_{I_c}^2 + \mu_{I_s}^2 + \left((2^{24}-1)*0.01\right)^2\right)\left(\sigma_{I_c}^2 + \sigma_{I_s}^2 + \left((2^{24}-1)*0.03\right)^2\right)} \quad (46)$$

where μ_{Ic} and μ_{Is} are a mean of cover and stego images respectively, whereas σ_{IcIs} is a covariance of cover and stego images, and σ^2_{Ic} , σ^2_{Is} are the variance of cover and stego images respectively. Table 3 reported the comparative visual quality of the stego-images by using four payload capacities (10%, 15%, and 25%). The quality of stego-images is measured by using PSNR (dB) and SSIM metrics to show the performance of the proposed algorithm over typical existing references [17, 19, and 20]. In this study, 400 images have been selected by size (384x512); all these images are converted to the grayscale images.

Table 3. The average values of PSNR (dB), and SSIM of various Stego-images generated by different Steganographic algorithms

Payload Capacity	Reference [19] average values of 8 images		Reference [20] average values of 300 images		Reference [17] average values of 400 images		The proposed algorithm with ANN_MACO average values of 500 images	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
10%	51.74	NA	50.8	1	64.11	0.9999	69.32	1
15%	44.72	NA	45.5	0.9997	63.50	0.9998	66.76	0.9999
25%	40.83	NA	NA	NA	58.17	0.9996	62.63	0.9998

It seems that the proposed algorithm is working efficiently, and the proposed ANN_MACO has outperformed algorithms in [17,19, and 20]. The Table 3 shows that PSNR for 10% payload increased significantly from 51.74 in [19], 50.8 and 64.11 in [19] and [20] up to 69.32 dB using the proposed ANN_MACO, where the greatest improvement of 22.04 dB with ANN_MACO when performed with payload capacity of 15%. On the other hand, Table 3 shows a similarity SSIM using ANN_MACO relatively better than the algorithms referenced in [17, 19, and 20].

6.2. Difference between neighboring pixels

The difference values of the horizontal neighboring pair for both cover and stego images are computed using the formula in Eq. (47):

$$d^c_{i,j} = P^c_{i,j} - P^c_{i,j+1}, \quad d^s_{i,j} = P^s_{i,j} - P^s_{i,j+1}, \quad \forall i, j \quad (47)$$

where P^c_{ij}, P^s_{ij} are two pixels values at the location (i,j) of cover and stego images respectively. Comparisons of two differences d^c_{ij} and d^s_{ij} using four images are reported in Figs 12(a-d).

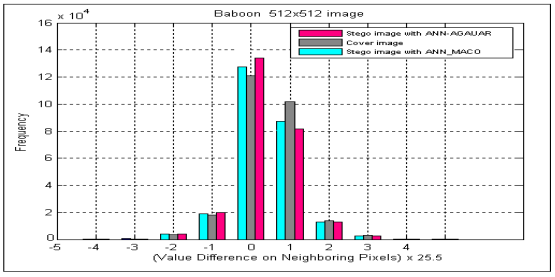

Figure 12a. Value difference on neighboring pixels for Baboon cover and stego images

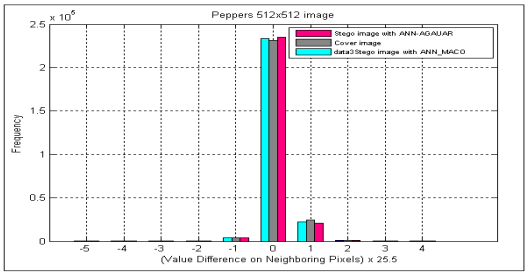

Figure 12b. Value difference on neighboring pixels for Peppers cover and stego images

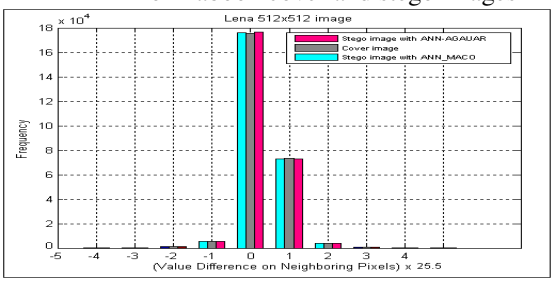

Figure 12c. Value difference on neighboring pixels for Lena cover and stego images

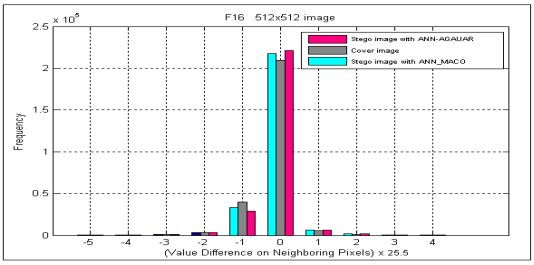

Figure 12d. Value difference on neighboring pixels for F16 cover and stego images

The results show that distances of four images are calculated individually; it seems that the smallest norm is reached when the proposed algorithm using ANN_MACO is implemented. Moreover, we observed that the greatest difference is at the image Baboon with pay- load percentage equal 37% when the earliest work (El-Emam, 2013) [17] is used, while with the proposed algorithm with/without using ANN_MACO, we can reduce the difference by approximately 24%.

6.3. Working against visual attack

Two kinds of testing are implemented, the first one bases on the set of the closest colors (one corresponding to the same pixel) using Euclidean norm Eq. (48) to find the distance between the cover-image and stego-image. Experimental testing of the Euclidean norm has been implemented on two algorithms (ANN_AGAUAR algorithm [17] and the proposed algorithm ANN_MACO), see Figs. 13a-13e.

$$d = \sqrt{(R_c - R_s)^2 + (G_c - G_s)^2 + (B_c - B_s)^2} \tag{48}$$

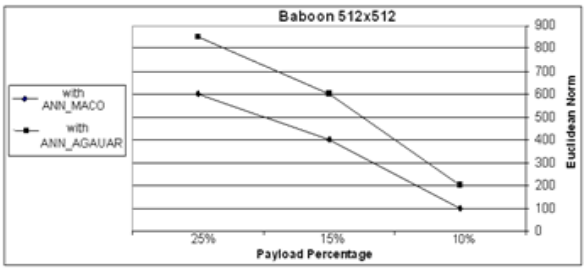

Figure 13a. Euclidean Norm Testing of Baboon color image

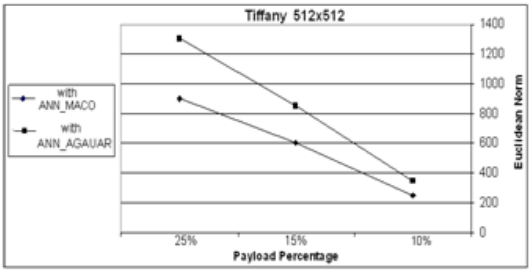

Figure 13b. Euclidean Norm Testing of Tiffany color image

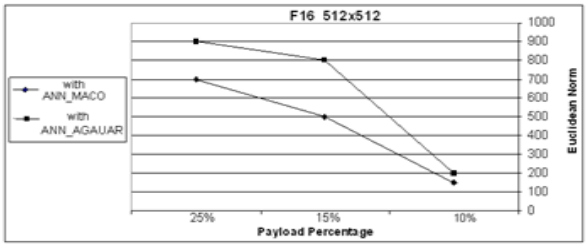

Figure 13c. Euclidean Norm Testing of F16 color image

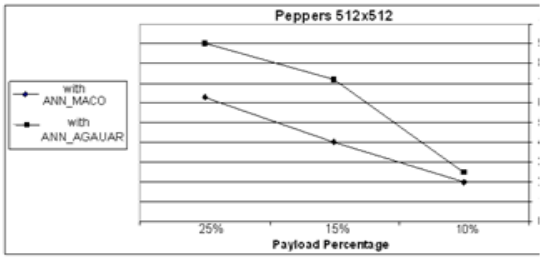

Figure 13d. Euclidean Norm Testing of Peppers color image

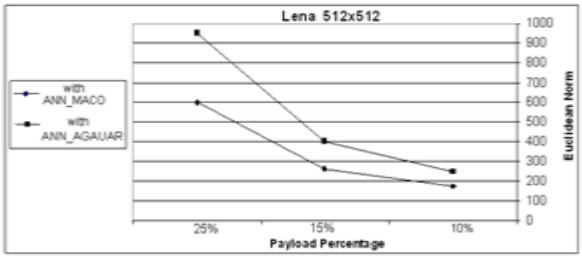

Figure 13e. Euclidean Norm Testing of Lena color image

The distances of five images are calculated individually; it appears that the minimum norm has been reached when the proposed algorithm using ANN_MACO is implemented. In addition; it is clearly that Tiffany's image has the least distance while Peppers's image has the greatest distance among other images. Results justify that the proposed algorithm ANN_MACO has demonstrated a clear improvement with closer Euclidean distance, which means that the stego-images are closer to the cover-images.

6.4. Working against statistical attack

The performance of the proposed steganography algorithm to hide secret message in the color image at the spatial-domain has been evaluated and tested against statistical attacks using modified WFLogSv attacker [21]. The experimental results have been implemented on 500 color images to check imperceptible level and compared with two hiding algorithms, standard LSB and modified LSB, (see [21]).

We apply "Receiver Operating Characteristic" (ROC) curve, see Figs. 14(a-b), which are based on two parameters, the probability of false alarms (P_{FA}) and the probability of detections ($1 - P_{MD}$), see Eq.(49).

$$P_{FA} = \frac{NCI\ (I_S)}{NCI},\ P_{MD} = \frac{NSI\ (I_C)}{NSI},\ P_E = \min \frac{1}{2}\left(P_{FA} + P_{MD}\right) \qquad (49)$$

where

NCI(Is) is the number of cover-images that recognized as stego-image, NCI is the total number of cover-images,

NSI(Ic) is the number of stego-image recognized as cover-images, NSI is the total number of stego-images,

and $P_{FA}, P_{MD} \in [0,1]$.

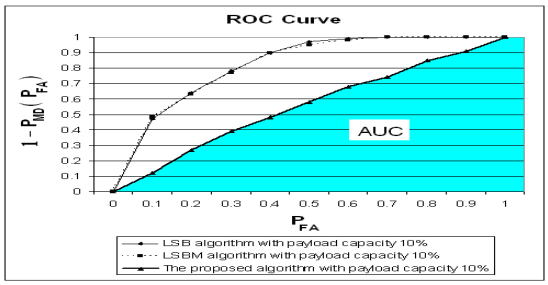

Figure 14a. ROC curves of Modified WFLogSv against LSB. LSBM and the proposed hiding algorithms with 10% payloads capacity

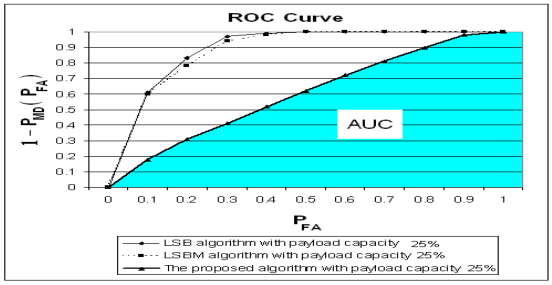

Figure 14b. ROC curves of Modified WFLogSv against LSB. LSBM and the proposed hiding algorithms with 25% payloads capacity.

It appears that (P_{FA}) is plotted on the horizontal axis while ($1 - P_{MD}$) is plotted on the vertical axis. The perfect security of the hiding algorithm has been reached when the area under a curve AUC equal to 0.5, while the perfect detection of steganalyzer is reached when AUC is equal to 1, see [21].

Results confirm that the proposed embedding algorithm with ANN_ MACO produces high imperceptible and working against modified WFLogSv attacker for different payload's capacities. Moreover, the security level of the present steganography for the payload capacity 10% is better than LSB and LSBM by approximately 50%, 49% respectively, while the security level of the present steganography for the payload capacity 25 % is better than LSB and LSBM by approximately 54 %, 52% respectively.

6.5. Performance of MACO

In this section, we explain the performance of the proposed learning system based on MACO that has been used to improve the quality of stego-image. Therefore, to check the performance of MACO, the best results of the Multiple Traveling Salesman Problem (MTSPs) have been calculated to find the shortest minimum cycle using the proposed MACO and compared these results with the best results of the previous works based on NMACO, classical ACO [22], and MACO [23]. These results have been illustrated in Table 4, which contains six instances of standard MTSPs for an acceptable number of nodes whose sizes are between 76 and 1002. These instances belong to TSP problems of TSPLIB including Pr76, Pr152, Pr226, Pr299, Pr439 and Pr1002, (see [24]). For each instance, the number of nodes (N), the number of salesmen (NS), and the max number of nodes that a salesman can visit (Max-N) has been applied. The proposed MACO has been capable to find better solution than the others techniques. Table 4 demonstrates that the standard deviation between the optimal solution in [24] and the best solutions of the proposed MACO for six standard MTSPs is 69583.82129, whereas the standard deviation between the optimal solutions in [24] and the best solutions the of NMACO, classical ACO [22], and MACO [23]are 97421.98788, 99128.30381 and 97978.17381 respectively.

The results in Table 4 confirm that the proposed MACO algorithm is better than NMACO, classical ACO [22], and MACO [23] by approximately 32 %, 35 % and 37 % respectively.

Table 4. Comparison between the proposed MACO algorithm, , NMACO, classical ACO [22], and MACO algorithms [23].

Instance	N	NS	Max-N	Optimal solution [24]	Best solutions using NMACO [22]	Best solutions using MACO [23]	Best solutions using classical ACO	Best Solutions of the proposed MACO
Pr76	76	5	20	108159	157413	178597	158425	128456
Pr152	152	5	40	73682	127083	130953	127993	95700
Pr226	226	5	50	80369	167239	167646	168631	136036
Pr299	299	5	70	48191	81261	82106	82871	63816
Pr439	439	5	100	107217	160298	161955	164941	146985
Pr1002	1002	5	220	259045	379042	382198	384901	282731

7. CONCLUSIONS

This paper proposed new steganography algorithm to enforce the security of data hiding and to increase the amount of payloads using four safety layers. The main contributions of this paper are: Proposed four safety layers to perform compression and encryption of a confidential message using a set partition in hierarchical trees (SPIHT) and advanced encryption standard (AES) mechanisms. An irregular image segmentation algorithm (IIS) on a cover-image has been constructed successfully in the second safety layer, and it is based on the adaptive reallocation segments' edges (ARSE) by applying an adaptive finite-element method (AFEM) to find the numerical solution of a proposed partial differential equation (PDE). The Proposed new intelligent computing technique using a hybrid adaptive neural network with a modified ant colony optimizer (ANN_MACO), to construct a learning system, which speeds up training process, and to achieve a more robust technique for hiding confidential messages into color images with an excellent imperceptible data in stego-images.

ACKNOWLEDGEMENT

The authors would like to thank Prof. R. H. Al-Rabeh from Cambridge University for his support and help with this research. This support is gratefully acknowledged.

REFERENCES

[1] Johnson, N. & Jajodia, S, (1998) “Steganalysis of images created using current steganography software”, Proc. of the Second International Workshop on Information Hiding, vol. 1525 , pp 273- 273, Springer. DOI: 10.1007/3-540-49380-8_19

[2] Wang, H. & Wang, S., (2004) “Cyber warfare: Steganography vs. steganalysis”, Communications of the ACM, Vol. 47, No. 10, pp 76-82.DOI:10.1145/1022594.1022597

[3] Provos, N. & Honeyman, P., (2003) “Hide and seek: An introduction to steganography”, IEEE Security and Privacy, Vol. 1, No. 3, pp 32-44. DOI:10.1109/MSECP.2003.1203220

[4] Chandramouli, R., Kharrazi, M. & Memon, N., (2004) “Image steganography and steganalysis concepts and practice ”, Proc. of IWDW’03, Vol. 2939, pp 35-49. DOI:10.1007/978-3-540-24624-4_3

[5] Bender, W., Gruhl, D., Morimoto, N. & Lu, A., (1996) ”Techniques for data hiding”, IBM System Journal, Vol. 35, No. 3, pp 313-336. DOI:10.1147/sj.353.0313

[6] Li, S., Tao, H. & Huang, Y., (2012) ”Detection of QIM steganography in G.723.1 bit stream based on quantization index sequence analy¬sis”, J. Zhejiang Univ. Sci. C, Vol.13, No. 8, pp 624–634. DOI:10.1631/jzus.C1100374

[7] Böhme, R., (2010) Advanced Statistical Steganalysis, Springer publisher.

[8] El. Emam, N. & Abdul Shaheed, R., (2008) “Computing an Adaptive Mesh in Fluid Problems using Neural Network and Genetic Algorithm with Adaptive Relaxation”, International Journal on Artificial Intelligence Tools. Vol. 17, No. 6, pp 1089-1108. DOI: 10.1142/S021821300800431X

[9] Chang,C., Chen,Y. & Lin, C., (2009) “A data embedding scheme for color images based on genetic algorithm and absolute moment block truncation coding”, Soft Compute, Vol. 13, No. 4, pp 321-331. DOI: 10.1007/s00500-008-0332-x

[10] Yi-Ta, W. & Shih, F., (2006) "Genetic algorithm based methodology for breaking the steganalytic systems" , IEEE Transaction on system, Man, and Cybernetica Part B: Cybernetics, Vol. 36, No. 1, pp 24 -31. DOI: 10.1109/TSMCB.2005.852474

[11] Arsalan, M., Malik, S. & Khan, A., (2012) "Intelligent reversible watermarking in integer wavelet domain for medical images", Journal of Systems and Software, Vol. 85, No. 4, pp 883-894. DOI: 10.1016/j.jss.2011.11.005.

[12] El. Emam, N., (2015) "New data-hiding algorithm based adaptive neural networks with modified particle swarm optimization", Computers & Security, Vol. 55, pp 21–45. DOI:10.1016/j.cose.2015.06.012

[13] Zhang, F., Pan, Z., Cao, K., Zheng, F. & Wa, F., (2008) "The upper and lower bounds of the information-hiding capacity of digital images", Information Sciences, Vol. 178, pp 2950–2959. DOI:10.1016/j.ins.2008.03.011

[14] Luo, X., Wang, D., Hu, W. & Liu, F., (2009) "Blind detection for image steganography: a system framework and implementation", International Journal of Innovative Computing, Information and Control. Vol. 5, No. 2, pp 433-442.

[15] Luo, W., Huang, F. & Huang, J., (2010) "Edge Adaptive Image Steganography Based On LSB Matching Revisited", IEEE Transactions On Information Forensics And Security, Vol. 5, No. 2, pp 201 – 214. DOI:10.1109/TIFS.2010.2041812

[16] EL-Emam, N., (2007) "Hiding a Large Amount of Data with High Security Using Steganography Algorithm", Journal of Computer Science. Vol. 3, No.4, pp 223-232. DOI:10.3844/jcssp.2007.223.232

[17] El-Emam, N. & AL-Zubidy, R., (2013) "New steganography algorithm to conceal a large amount of secret message using hybrid adaptive neural networks with modified adaptive genetic algorithm", The Journal of Systems and Software, Vol. 86, No. 6, pp 1465-1481. DOI:10.1016/j.jss.2012.12.006

[18] Al-Shatanawi, O., El. Emam, N., (2015) "A new image steganography algorithm based on MLSB method with random pixels selection", International Journal of Network Security & Its Applications , Vol. 7, No 2, pp 37-53.DOI : 10.5121/ijnsa.2015.7203

[19] Hong,W., Chen, T. & Luo, C., (2012) " Data embedding using pixel value differencing and diamond encoding with multiple-base notational system" , The Journal of Systems and Software , Vol. 85, pp 1166- 1175. DOI:10.1016/j.jss.2011.12.045

[20] S. Geetha, V. Kabilan, S.P. Chockalingam, & N. Kamaraj Varying, (2011) "Radix numeral system based adaptive image steganography", Information Processing Letters, Vol. 111 , pp 792–797. DOI:10.1016/j.ipl.2011.05.013

[21] Shojaei-Hashemi, A., Soltanian-Zadeh, H., Ghaemmagham S. & Kamarei M., (2011) "Universal image steganalysis against spatialdomain steganography based on energy distribution of singular values", Proceeding of 7th International Conference on Information Technology and Applications (ICITA 2011), pp 179–83.

[22] Yousefikhoshbakht, M., Didehvar, F. & Rahmati, F., (2013) "Modification of the Ant Colony Optimization for Solving the Multiple Traveling, Salesman Problem", Romanian Journal of Information Science and Technology, Vol. 6, No. 1, pp 65–80.

[23] Junjie, P. & Dingwei, W., (2006) "An ant colony optimization algorithm for multiple traveling salesman Problem", ICICIC '06: Proceedings of the First International Conference on Innovative Computing, Information and Control, pp 210–213.DOI: 10.1109/ICICIC.2006.40

[24] Ruprecht-Karls-University Heidelberg. Tsplib network optimization problems, 2008. http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.

AUTHORS

Nameer N. EL-Emam: He completed his PhD with honor at Basra University in 1997. He works as an assistant professor in the Computer Science Department at Basra University. In 1998, he joins the department of Computer Science, Philadelphia University, as an assistance professor. Now he is an associated professor at the same university, and he works as a chair of computer science department and the deputy dean of the faculty of Information Technology, Philadelphia University. His research interest includes Computer Simulation with intelligent system, Parallel Algorithms, and Soft computing using Neural Network, GA, ACO, and PSO for many kinds of applications like Image Processing, Sound Processing, Fluid Flow, and Computer Security (Seteganography).

Kefaya Qaddoum has obtained her first degree in computer science and information technology from Philadelphia university, as well as the master degree, did her PhD at Warwick University, UK in Artificial Intelligence. Worked as Lecturer at Warwick university for two years, worked for Bahrain university for one year and finally worked for Prince sultan university in Saudi Arabia. she conducted and published research papers covering AI methods, and Data mining.