

# Introducing a Novel Approach to Concealing Textual Information Within Digital Images Through the Use of Neural Networks

Mohammad Yasemifar<sup>1</sup>, Sattar Mirzakoochaki<sup>2\*</sup> , and Mohammad Norouzi<sup>3</sup>

**Abstract**--In steganography, a text is placed in a digital image in a secure, imperceptible and retrievable way. The three main methods of digital image steganography are spatial methods, transformation and neural network. Spatial methods change the pixel values of an image to embed information, while transform methods embed information hidden in the frequency of the image. Neural networks are used to perform the hiding process and it is the main part of this research. This research examines the use of LSTM<sup>2</sup> deep neural networks in digital image text steganography. This work extends an existing implementation that uses a two-dimensional LSTM to perform the preparation, hiding, and extraction steps of the steganography process. The proposed method modified the structure of LSTM and used a gain function based on several image similarity measures to maximize the indiscernibility between an overlay and a steganographic image. Genetic algorithm helps in improving the structure of LSTM networks in the textual information within hidden images, with optimizations (number of layers, neurons, evaluations) and selection of appropriate features, increasing the accuracy, improving image quality and preventing overfitting. This method helps to find the optimal architecture for the LSTM network and improves the efficiency of the steganography.

The proposed method demonstrates superior performance based on three evaluation metrics Peak Signal-to-Noise Ratio (PSNR<sup>3</sup>) in decibels, Mean Squared Error (MSE<sup>4</sup>), and accuracy rate in percentage compared to three other benchmark images (lena.png, peppers.png, mandril.png, and monkey.png), achieving values of 93.665275 dB, 0.6945 MSE, and 97.23% accuracy, respectively. The proposed method modified the structure of LSTM and used a gain function

**Index Terms**-- Novel approach - Steganography - Text information - Digital images - Neural networks.

## I. INTRODUCTION

In modern societies, the growing reliance on digital media has elevated the importance of file security, particularly in safeguarding against malicious users, a concern that is especially prevalent in online environments. Cryptography leverages mathematical methods to ensure information security. Essentially, it involves transforming the text of a message or piece of information using a cryptographic key and algorithm, so that only individuals equipped with the appropriate key and algorithm can decipher the original data from the encrypted version. When the exchange of encrypted information poses challenges, it may even become necessary to conceal the existence of the communication itself. In recent years, steganography has gained increasing attention from researchers focused on securing information transmission. Specifically, in the context of image steganography, numerous techniques have been developed with the shared goal of maximizing capacity, security, and resilience. The goal of an encryption system is to achieve three parameters, resistance, capacity and security. Meeting these parameters at the same time in an encryption system is a difficult task. In design and modeling, these parameters are considered as the vertices of a triangle. These three criteria are considered in such a way that a compromise is established between all three items and the optimal design point is determined [1].

However, these three objectives are inherently conflicting, and achieving an optimal balance among them is exceedingly difficult, if not impossible. They are often conceptualized the

1. Department of Electrical Engineering, Faculty of Electrical Engineering, Qazvin branch, Islamic Azad University, Qazvin, Iran.

2. Department of Electrical Engineering, Faculty of Electrical Engineering, University of Mississippi, USA.

3. Department of Electrical Engineering, Faculty of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.

Corresponding author Email: [m\\_kuchaki@iust.ac.ir](mailto:m_kuchaki@iust.ac.ir)

## Cite this article as:

Yasemifar, M., Mirzakoochaki, S. and Norouzi, M., 2024. Introducing a Novel Approach to Concealing Textual Information Within Digital Images Through the Use of Neural Networks. *Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE)*, 4(4), pp. 9-21.

<https://doi.org/10.22075/MSEEE.2025.36371.1194>

vertices of a triangle, where prioritizing one aspect inevitably demands less emphasis on the others. The core objectives resistance, transparency, and capacity are represented as a pyramid in Fig. 1.

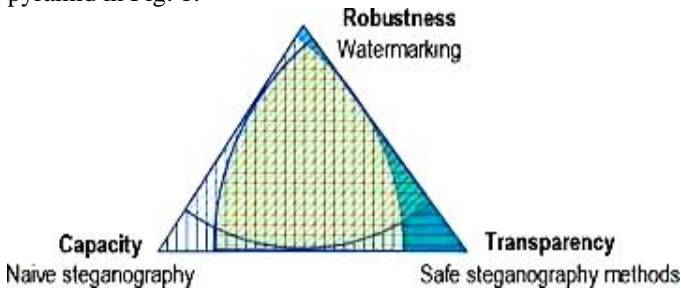


Fig. 1. three goals including resilience, transparency and capacity[1]

The trade-off between capacity, security, and transparency in steganography of textual information within digital images is a fundamental challenge. These three factors are directly related, and improvement in one typically comes at the expense of another. Below, we provide an in-depth analysis of these trade-offs and offer practical solutions for optimal balance:

### 1. Capacity:

**Definition:** Capacity refers to the amount of data that can be hidden in an image. Higher capacity means the ability to hide more information.

**Impact:** Increasing capacity usually involves more changes to the image, which can lead to reduced transparency and security.

### 2. Security:

**Definition:** Security refers to the resistance of the steganographic method against detection and extraction of hidden information. High security means it is more difficult for unauthorized individuals to detect the presence of hidden information and extract it.

**Impact:** Increasing security usually involves reducing capacity or complicating the algorithm, which may reduce transparency.

### 3. Transparency:

**Definition:** Transparency refers to how imperceptible the changes made to the image are due to hiding information. High transparency means the inability to detect the changes made with the naked eye.

**Impact:** Increasing transparency usually involves reducing capacity or security because fewer changes mean less information to hide and greater vulnerability to attacks.

### In-Depth Analysis of Trade-offs:

#### Capacity vs. Transparency:

When we try to increase capacity (i.e., hide more data in the image), more changes are made to the pixels. These changes can be visible and reduce the transparency of the image.

#### Capacity vs. Security:

Increasing capacity often means using more predictable patterns for hiding data, which can reduce security.

Transparency vs. Security:

Efforts to increase transparency (i.e., reduce visible changes) can reduce security because fewer changes mean less information to hide and greater vulnerability to attacks.

Methods that make very few changes to the image may be vulnerable to statistical analysis and machine learning-based attacks.

#### Practical Solutions for Optimal Balance:

##### 1. Using Adaptive Steganography Methods:

These methods adjust the amount of hiding based on the characteristics of the image. For example, in complex and detailed areas of the image, more data can be hidden without significant changes.

##### 2. Employing Frequency Domain Transformations

Transformations such as Discrete Cosine Transform (DCT<sup>5</sup>) and Discrete Wavelet Transform (DWT<sup>6</sup>) can hide information in different frequencies of the image. These methods usually provide better transparency because changes in the frequency domain are less noticeable.

##### 3. Using Encryption Before Steganography

Before hiding data in the image, encrypt it using strong encryption algorithms (such as AES<sup>7</sup>). This ensures that even if an attacker manages to extract the data, they cannot read it.

##### 4. Random Data Distribution:

Instead of using simple and predictable patterns, distribute the data randomly in the image. This makes it more difficult for an attacker to identify patterns and extract information.

##### 5. Using Machine Learning-Based Methods:

Neural networks can be used to learn complex patterns in images and hide data imperceptibly. These methods can create a better balance between capacity, security, and transparency.

##### 6. Combining Multiple Methods (Hybrid Approaches):

Combining several steganographic methods can help improve overall performance. For example, an adaptive method can be used to determine suitable hiding locations, and a frequency-based transformation method can be used to hide the data in those locations.

#### Practical Solution for Optimal Balance:

Balancing capacity, security, and transparency in steganography of textual information within digital images requires the use of intelligent and hybrid methods. By selecting appropriate algorithms, encrypting data, and distributing information randomly, an optimal balance can be achieved that provides adequate capacity, high security, and preserves image transparency. Also, continuous evaluation and adjustment of algorithm parameters based on the specific needs of the application can help improve overall performance [2].

The most famous and common method of hiding data in files is to use graphic images as hidden locations. By combining the two, confidentiality and security of confidential information can be greatly improved [3]. For steganography operations in digital images, several methods have been invented, including working in the space or frequency domain [4], and each has its own advantages and disadvantages and has a special application [5].

<sup>5</sup> Discrete Cosine Transform

<sup>6</sup> Discrete Wavelet Transform

<sup>7</sup> Advanced Encryption Standard

**DWT (Discrete Wavelet Transform) Method:**

DWT helps decompose the image into different frequency levels, allowing information to be distributed across various regions of the image. This method can help preserve important image features, as the hidden information is embedded in different frequency levels.

Images steganographed using DWT generally have better visual quality because this method is better at preserving image details [6].

**DCT (Discrete Cosine Transform) Method:**

DCT is commonly used in image compression and can be effective in reducing data size. This method can also embed information in lower frequencies, which are less noticeable to the human eye. Although DCT can effectively hide information, it may sometimes reduce image quality, especially if a large amount of information is embedded in lower frequencies [7].

**Methods to Improve Visual Quality:**

1. Adjusting Steganography Parameters: Fine-tuning the parameters related to the amount and type of hidden information can help preserve image quality.

2. Using Error Correction Algorithms: Ensuring that the hidden information is correctly extracted and does not negatively impact image quality.

3. Intelligent Compression: Employing compression techniques that maintain image quality after steganography.

4. Post-Steganography Processing: Applying filters or image processing techniques to enhance the visual quality of the final image.

5. Hybrid Methods: Combining DWT and DCT to leverage the advantages of both methods and mitigate their drawbacks.

LSB embedding is one of the most common steganography methods in images, and several attacks have been proposed, some of which are accurate and some of which are less accurate (Mohammadi, 2014). Today, research in this field is focused on increasing capacity, increasing quality, and reducing the suspiciousness of the output image. (Chun, 2013)

**Challenges and Vulnerabilities of the LSB Method:****1. Vulnerability to Statistical Attacks:**

Statistical attacks analyze the statistical distribution of pixels in the image, looking for abnormal patterns caused by data hiding. The LSB method, because it changes the least significant bits, can alter the statistical distribution of pixels and create identifiable patterns.

**Types of Statistical Attacks:**

Histogram-based attacks: Changes in the pixel distribution are identified by examining the image histogram.

Pair-wise attacks: Abnormal patterns are identified by examining the parity of the LSB bits.

Regular Singular (RS) attacks: Changes in the pixel distribution are identified using RS matrices.

**Solutions:**

□ Using adaptive steganography methods that adjust the amount of hiding based on image features.

□ Distributing data randomly in the image instead of using

simple and predictable patterns.

□ Using encryption before steganography to reduce identifiable patterns in the hidden data.

**2. Vulnerability to Visual Attacks:**

If a large amount of data is hidden using the LSB method, changes in the pixels can be visually noticeable as noise. This noise is usually more noticeable in smooth and low-detail areas of the image.

**Solutions:**

□ Reducing the amount of hidden data.

□ Using adaptive steganography methods that adjust the amount of hiding based on image features.

□ Using frequency domain transformations (such as DCT and DWT), where changes in the frequency domain are less noticeable.

**3. Vulnerability to Filtering Attacks:**

Applying various filters (such as mean and Gaussian filters) can reduce the noise caused by LSB<sup>8</sup> hiding and corrupt the hidden data.

**Solutions:**

□ Using steganography methods that are more resistant to filters.

□ Hiding data in high-frequency DCT coefficients, which are less affected by filters.

□ Using Error Correction Codes to increase resistance to data corruption.

**4. Vulnerability to Machine Learning-based Attacks:**

Neural networks and other machine learning algorithms can be used to learn complex patterns in images and identify the presence of hidden data. These attacks are usually performed by training a model on a set of images containing hidden data and images without hidden data.

**Solutions:**

□ Using steganography methods that are more resistant to machine learning.

□ Using Generative Adversarial Networks (GANs<sup>9</sup>) to create images that contain hidden data while still appearing natural.

□ Using strong encryption before steganography to reduce learnable patterns.

**5. Vulnerability to Signal Analysis Attacks:**

These attacks analyze the signal characteristics of the image (such as the frequency spectrum), looking for changes caused by data hiding.

**Solutions:**

□ Using frequency domain transformations to hide data at different frequencies of the image.

□ Using adaptive steganography methods that adjust the amount of hiding based on the signal characteristics of the image.

**6. Capacity Limitations:**

The LSB method usually has a limited capacity, and a large amount of data cannot be hidden using it. Increasing capacity usually comes at the cost of reduced transparency and security.

**Solutions:**

□ Using steganography methods with higher capacity.

<sup>8</sup> Least Significant bit

<sup>9</sup> Generative Adversarial Networks

- Using data compression before steganography.

## II. LITERATURE REVIEW

The LSB method is a simple and widely used method for steganography, but it has several vulnerabilities against various attacks. To increase the security of this method, appropriate strategies (such as encryption, adaptive methods, random distribution, and frequency domain transformations) should be used. Also, continuously evaluating and testing the steganography method against various attacks can help identify and address vulnerabilities [8].

Digital steganography is a process by which a signal specified by the user is hidden in other signals, for example, digital content can be electronic documents, images, audio, video, etc. There are various techniques to protect the original digital data and to prevent unauthorized copying and duplication or manipulation. Steganography provides a basic solution to the problem of image aggregation and its authenticity and validity. This solution is a type of data hiding technique, so that it provides another way to maintain security for digital image data. Unlike the use of a special encryption algorithm to protect and secure secret data from unauthorized access, the goal of steganography is to embed or embed secret data in pre-selected meaningful images, of course, without making any visually perceptible changes to continue the cyber attackers' ignorance of the existence of a security secret in the information data, these images are called cover images. Generally, a steganographic message may be a photo, video or audio file. A message may be created by using algorithms such as invisible ink between the lines of secure documents, ensuring information security: a topic that is considered important in the modern world of image transfers over networks, on which information is then embedded [8].

After reviewing the previous methods, simulation of the results based on the presented model will be done. This evaluation will include analysis of system performance compared to previous methods, examination of strengths and weaknesses of simulation and analysis of obtained data to determine the success rate of the new model.

In this research, the theoretical foundations of digital image processing, hidden graphics and algorithms will be investigated. Then the proposed method will be presented. Finally, simulation, hidden text writing inside digital images, evaluation and conclusion will be done [9].

### THEORETICAL FOUNDATIONS OF IMAGE PROCESSING

An image can be defined as a two-dimensional function  $f(x,y)$  where  $x$  and  $y$  are spatial coordinates on the screen and the range of  $f$  at each pair of coordinates  $(x,y)$  is called the brightness or gray level of the image at that point. When  $x$  and  $y$  and the brightness values of  $f$  are all finite quantities, then the image is called a digital image. The subject of digital image processing refers to the processing of digital images by a digital computer. A digital image is made up of a finite number of elements, each with a specific location and value. These elements are called picture elements, picture elements, and pixels. Pixels are the common term for the elements of a digital image [10].

### Steganography in Images

In image-based steganography, one image is embedded within another so that the concealed image remains undetectable in the cover image. The primary challenge lies in ensuring that the edges or elements of the hidden image referred to as the stego do not become visible in the cover image. Among the various techniques available, the Least Significant Bit (LSB) algorithm is one of the most commonly used methods for achieving effective and covert embedding in images [11].

### Applications of Steganography

Steganography finds use in various contexts, including thwarting the unauthorized distribution of illegal content and protecting sensitive organizational data. For example, a stock exchange might encode confidential messages in text using steganographic techniques to securely transmit them between sender and recipient. Similarly, an organization might embed coded information within an image so that only the intended recipient can interpret its hidden content. This makes steganography a valuable tool for secure communication and encrypted data sharing [11].

### The Significance of Steganography

As advancements in information technology and multimedia continue, safeguarding intellectual property has become increasingly critical. Steganography plays a vital role by embedding identifying information covertly into multimedia products, ensuring ownership rights are protected. The two core principles of steganography are its difficulty which refers to the complexity of separating hidden data from its carrier and its invisibility. There is often a balance between these two aspects: as the complexity of the steganographic method increases, its subtlety may decrease and vice versa. At its core, steganography is both an art and a science, focusing on embedding information within a host medium. Its importance is growing alongside advancements in digital communication technologies. Unlike cryptography, where security focuses on preventing message detection altogether, watermarking emphasizes resistance to alterations due to its unique use cases. Each of these fields cryptography, watermarking, and steganography serves distinct purposes and applications, catering to diverse requirements for secure communication, content protection, and data integrity [11].

Fig. 2 shows Security systems are divided into two parts: hiding and encryption. Information hiding is done by two methods: watermarking and steganography. Digital images, video, text and audio host confidential information in steganography.

### Steganography is broadly categorized into three main types:

- **Full steganography:** This involves no use of a steganographic key. The method operates under the assumption that no intermediaries between the source and the destination are aware of the communication.
- **Secret key steganography:** In this method, a steganographic key is shared between parties prior to the exchange. However, it is particularly vulnerable to interception.
- **Public key steganography:** This approach uses a

combination of public and private keys to secure communication effectively.

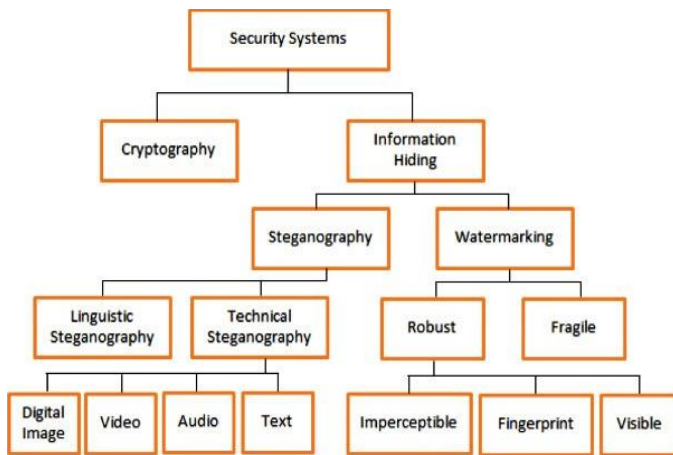


Fig. 2. Different areas of information hiding [11]

Additionally, steganographic techniques are classified into six primary methods:

- **Substitution methods:** These work by replacing redundant parts of the cover medium with the hidden message, often utilizing spatial domain techniques for embedding the data.

- **Domain transformation methods:** Secret information is embedded in the signal's transmission space using frequency domain techniques.

- **Spread spectrum methods:** These apply principles from spread spectrum communication to hide the information.

- **Statistical methods:** These involve manipulating statistical properties of the cover medium for encryption and employ hypothesis testing during the extraction process.

- **Derivative methods:** Here, information is extracted by analyzing and measuring changes in the signal derivative of the original cover during decoding.

- **Cover generation methods:** These create a path for concealing information while simultaneously hiding any traceable links associated with the cover medium.

Steganography in the digital image

Hiding information in the image can be classified according to several methods. The classification can be based on the overlay image used (2D or 3D images), the type of target application, or the reversible or irreversible recovery process, the nature of the embedding process, the spatial domain or transformation, and adaptive cryptography.

### III. PROPOSED METHOD

According to the research background, an optimal method for textual information steganography in an image is to be performed. The approach of this research is that an image is entered into the program and a separate text is entered. The LSB method deals with the initial steganography and then a combined genetic algorithm method based on the chaotic Rasler mapping is presented to improve steganography in the image and a deep LSTM neural network will be used for continuous training of this operation.

**The data placement method is shown in Fig. 3.**

Step 1) Separating the RGB components from the cover image and selecting the appropriate components, namely R for red, G for green and B for blue for data placement.

Step 2) converting the input message into binary bits with the help of converting the value into ASCII.

Step 3) Evaluate the threshold value according to the size of the secret message. Threshold value is the difference in brightness between two pixels that is greater than or equal to the threshold value.

Step 4) Selection of the region according to the threshold value to generate replacement pairs. In this section, storage of encrypted pairs as secret keys is also done.

Step 5) Roussler's chaotic mapping algorithm based on genetic algorithm to place hidden message bits in the cover image.

Step 6) Rewriting the changes in the overlay image at the time of data placement.

Step 7) combining the RGB components of the cover image and storing the image as a stego image.

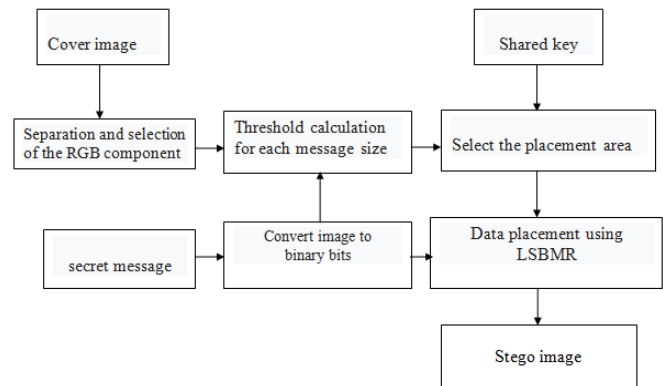


Fig. 3. Data placement method.

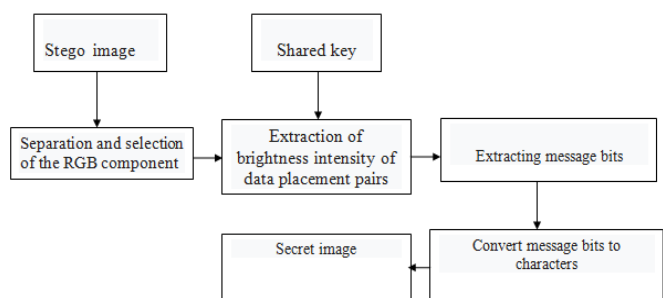


Fig. 4. extracting Data method.

**The data tracking method is shown in Fig. 4.**

Step 1) Separation of RGB components from the cover image and choosing the appropriate components, namely R for red, G for green and B for blue for data extraction.

Second step) extracting the brightness intensity of encrypted pairs using the sharing key.

Third step) extracting the message bits from the brightness intensity calculated in the second step using LSTM deep neural network.

First, the image data and a text are read and its structure is



converted into a numerical array, because the builders of images are numerical arrays. With these arrays, vectors can be created faster and more easily. Because this research presents a model based on LSB, then the structure is in the form of a vector that requires a numerical array. In the second stage, windowing and segmentation of images are performed and in the next stage, the numerical array or texts are converted into an ASCII number, which creates LSB vectors. Unicode is one of the standards used for encoding characters in text. This standard is compatible with most encoding standards such as ISO/IEC 10646. In the Unicode standard, each character in the text is assigned a number or code. Each of these numbers, as a number or code, is a code point. When referring to text in hexadecimal, they start with the prefix "U+". Each character also has a unique name, that is, the English character "A" with decimal code 65 is represented in the Unicode system as "U+0041". These Unicode names have a correspondence for the same characters in the ISO/IEC10646 standard. The Unicode standard groups characters in blocks using scripts. A script is a system of related characters. The standard keeps the characters in a resource set. When the characters of a language are in a specific order, i.e. alphabetical order, the Unicode standard also specifies them in the code area in a way that uses the same order. The blocks are of different sizes, for example, the Sri Lankan block has 265 code points, but the CJK block contains thousands of code points. Each language is arranged in a block in alphabetical order. The coding starts at "U+0000" and continues with the ASCII characters, followed by Greek, Sri Lankan, Hebrew, Arabic, Hindi and other languages, followed by punctuation marks and symbols, and so on. A range of code points are reserved for private use, followed by a range of compatibility characters. These characters were created to accommodate older versions. A large range of code points on the BMP and two very large ranges at the supplementary level are reserved as private use areas that have no public meaning and are for the use of users or programs. A set of page layout programs may use them as control codes for placing text on the page. Encoding standards not only identify a numeric value, or code point, for each character, but also specify how the value is represented in bits. These standards define three types of encoding forms that allow the same data to be sent in three ways. Byte, word, and double word-oriented formats, in other words, 8, 16, and 32 bits are assumed for each code unit. All three use the same common code set and can be converted to each other without loss of data [12].

The three forms are as follows:

- UTF-8: Common for HTML and similar protocols. Provides a way to convert all characters to a single byte value, similar to ASCII bytes.

- UTF-16: Common in environments that require a balance between efficient use of characters and economical use of memory. It is compact and all characters used are in a single 16-bit code unit, while the remaining characters are accessed through pairs of 16-bit code units.

- UTF-32: Useful where memory space is not a concern, but fixed code unit access width is desirable. Each Unicode character is encoded in a single 32-bit code.

All encoding forms require a maximum of 4 bytes or 32 bits of data for each character. Due to the similarity of the Persian language to Arabic, Persian characters are in blocks corresponding to the Arabic language. The Arabic block starts at 0600 and continues to 06FF. The first "06" that appears in Arabic or Persian characters indicates that this character belongs to the Arabic block and the next two digits are the code of the character itself. For example, the letter "M" with the code. In the Unicode standard, which almost all software follows today for encoding digital texts, there are many control characters that are used for special purposes and are not normally present in the text [12]. The presented steganography method is performed in several stages:

Converting the cipher message to its binary equivalent: First, the cipher message must be converted to binary to be hidden within the original text. In this method, the ASCII equivalent of each character is converted to an 8-bit binary set and then each eight bits are two's complement, finally the binary string of the message is reversed and grouped into bit pairs and provided to the function. Table I describes the different states of each bit pair, which is conventionally marked with one of the Unicode lengthless control characters.

TABLE I  
Grouping of Cipher Message Bytes and Conventional Notation [12]

Unicode character code	Character shape	Two-bit classification	Character name
U+200B	-	00	ZWSP
U+200C	-	01	ZWJ
U+200D	-	10	ZWNJ
U+180E	-	11	MVS
U+202A OR U+202B	-	FOM	LRE OR RLE

Insertion and marking of the message: Upon receiving the binary code generated by the Rasler chaos function, each eight-bit segment is divided into two bytes. For every pair of bits, a corresponding hidden token is assigned based on the classifications outlined in Table I. The character scanner proceeds to compare characters sequentially until it reaches a designated index position, at which point two tokens are inserted into these positions according to the classification table. This process is repeated systematically for all bit pairs within the message. To maintain the integrity of the encrypted file, six specific characters used for marking are inserted in an invisible manner, ensuring they remain undetectable. This approach preserves the original transparency of the file. To enhance resistance against potential tampering or attacks, a mechanism has been implemented to allow message recovery even if portions of the text are edited by the user. Within this mechanism, the message undergoes multiple encryption passes from the beginning to the end of the file. For verification, if the message cannot be correctly detected during scanning, the algorithm continues to examine the host file thoroughly to locate the encrypted message. To delineate the number of

encryption iterations, a control character (with no Unicode length) is inserted before the encryption process. Depending on the language's text direction left-to-right (LRE) or right-to-left (RLE) the relevant hex code (U+202A or U+202B) is embedded. Following this, the encrypted message and its corresponding length are appended to enable precise message recognition during decryption. Discovery and extraction of the encrypted message: At this stage, the stego file, serving as the carrier of the encrypted data, is provided as input and scanned character by character [12]. The text scanner identifies character bits by evaluating the positions of bit-pair type indicators in accordance with Table I. Once all marking characters are detected and removed, the algorithm verifies whether a control character (LRE or RLE) is present at the start of the password. It then separates the message length from its tail, enabling accurate retrieval of individual encrypted characters. The process concludes with a final accuracy check to confirm successful message extraction [12].

In text files, arrow keys, tab spaces, and trailing spaces at the end of lines which are often invisible to users in most text editors can be used as carriers for hidden data. The location of indicators for watermarking in digital signals is shown in Table II.

This concealed information doesn't necessarily have to be textual; it can encompass other types of data as well. For instance, an image can be embedded within another image, or a signal can be hidden inside a different signal. Additionally, steganography is not confined to predefined techniques; individuals are free to devise their own methods for concealing information. In the case of audio signals, hidden data is embedded into the lower-order bits of the carrier signal, along with the reference signal into which the information is to be concealed. This particular approach represents a straightforward example of signal-based steganography [12].

Adaptability of the proposed method to new technologies and emerging security challenges:

**Deep learning:** Deep learning is constantly evolving and can easily adapt to new data and new types of cryptography. This flexibility allows them to effectively respond to technological changes and new security needs.

**Transfer Learning:** Using transfer learning techniques, neural networks can be trained for specific encryption tasks, even if the training data is limited.

**Generative models:** The use of generative models such as GAN (Generative Adversarial Networks) can help generate new high-quality images in which information is hidden.

**Learning and updating:** Neural networks can continuously learn and adapt to new data and techniques. This capability allows them to adapt to new technologies such as the Internet of Things (IoT) and 5G.

**Chaotic Genetic Algorithms:** Due to their stochastic and non-linear characteristics, these algorithms can easily adapt to changing environmental conditions and new input data. This flexibility helps them to withstand technological changes and

new security needs.

**Generating complex data:** Using chaotic genetic algorithms can help generate complex and unpredictable patterns that make it more difficult to identify hidden information.

Given that there are many methods for securely sending information on cyberspace platforms today, the use of steganography methods can be of great help in sending and receiving data. In addition, steganography can be sent in such a way that only the sender and receiver have the ability to extract information, and on the other hand, the ability to change the original data is not easily possible. If the components required for the steganography algorithm in this method are expressed as follows, the steganography operation can be shown symbolically as follows:

- Host or cover file (C) (can be an image file).
- Cipher message, which can be plain text, encrypted text, or any other type of data. (M)
- Stego function ( $F(s)$ ) and its inverse ( $F(s)^{-1}$ ). (CMM function)
- Steganography position tracker  $Com(Wi)$ , including the ASCII code of the index positions in the text (the tracker scans the index positions based on the ASCII code range).
- An optional secret key or password that may be used to hide or retrieve the message.

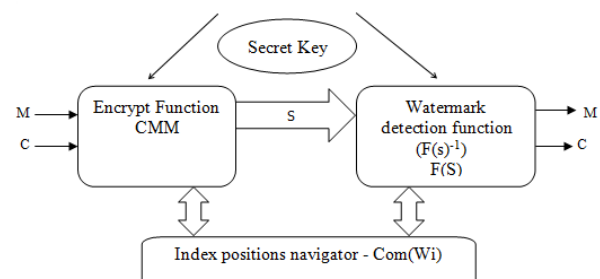


Fig. 5. Outline of the message watermarking mechanism in Signal [12].

In the stego process, the host file and the message to be concealed are provided, along with an optional encryption key. After applying the stego technique to the host file, the resulting stego file (S) is produced. This process is illustrated in Fig. 5. The encryption mechanism within the stego system utilizes the chaotic Rössler function in conjunction with a genetic algorithm. This combination allows for multiple functional variations and significantly enhances message security by transforming the original message into an encrypted format. Unlike many conventional methods that embed only a single digit during each stego operation, this approach embeds two digits per algorithm run, thereby increasing the method's capacity. Further details about the components of this method are outlined below.

TABLE II

Positions of Indicators for Watermarking in Digital Signals

ASCII code	Condition for checking the position	Index position in the text
33-47, 58-63, 91-95, etc. ASCII code of special characters in all languages	Before and after punctuation characters	Special punctuation characters such as ".,.} {} {(_.....,,\$*"
Checking the ASCII code 13	After the Enter character, the start of each paragraph and the spacing between the lines of the paragraphs are stored.	The beginning of each paragraph and blank lines between paragraphs

Now, the LSB method based on the chaotic Rössler function is improved using the genetic algorithm. The LSB algorithm is used for gray-level images in the cover image. The algorithm uses two pixels of the cover image as the insertion part to hide the secret message. Of the two pixels, the first pixel  $x_i$  is used to hide the secret message of bit  $m_i$  and the binary relationship between the pixels with the value  $(x_i, x_{i+1})$  is used to hide another message in bit  $m_{i+1}$ . The relationship between the two pixels is calculated with the help of the smoothing function, which is in the form of relation (1).

$$f(x_i, x_{i+1}) = LSB(floor(\frac{x_i}{2}) + x_i + 1) \quad (1)$$

The chaotic Rössler function with the help of genetic algorithm eliminates the non-contradiction feature introduced by the LSB approach. This makes it robust against steganizer attacks that exploit the asymmetry feature of the stego image. As a schematic addition and subtraction, the expected number of changes is reduced from 0.5 in LSB to 0.375 for the chaotic Rössler function with the help of genetic algorithm for the embedding capacity. Hence, the statistical detection of image steganography is low [13]. The chaotic Rössler function algorithm with the help of genetic algorithm leads to embedding data in a pixel pair using the four states in relation (2).

$$\begin{aligned}
 \text{case1: } & LSB(x_i) = m_i \& f(x_i, x_{i+1}) = m_{i+1} \quad , \quad (x'_i, x'_{i+1}) = (x_i, x_{i+1}) \\
 \text{case2: } & LSB(x_i) = m_i \& f(x_i, x_{i+1}) \neq m_{i+1} \quad , \quad (x'_i, x'_{i+1}) = (x_i, x_{i+1} \pm 1) \\
 \text{case3: } & LSB(x_i) \neq m_i \& f(x_i, x_{i+1}) = m_{i+1} \quad , \quad (x'_i, x'_{i+1}) = (x_i - 1, x_{i+1}) \\
 \text{case4: } & LSB(x_i) \neq m_i \& f(x_i, x_{i+1}) \neq m_{i+1} \quad , \quad (x'_i, x'_{i+1}) \neq (x_i, x_{i+1}) \quad (2)
 \end{aligned}$$

According to the equation (2),  $m_i, m_{i+1}$  are the message bits  $x_i$  and  $x_{i+1}$  are a pair of pixels before data embedding,  $x'_i$  and  $x'_{i+1}$  are a pair of pixels after data embedding. Embedding cannot be provided for pure pixels with small values or large permissible values. The message embedding rate is discovered by the process of generating a quasi-random sequence [13].

### Features of Rossler Chaos Map:

1- Choosing the initial conditions and programs: By choosing the exact initial and preliminary conditions, it is possible to produce keys that even the smallest change in the choices will lead to completely different results.

2- Changing the phase of the carrier data: by using the chaotic mapping outputs, subtle changes are made in the carrier data of the image samples, which are practically imperceptible, but contain the original information.

3- High security: Due to the chaotic nature, it is very difficult to analyze and break the algorithms based on Rossler mapping. In addition to information security, these methods help to reduce the risks of identifying and revealing hidden data in public media.

The proposed framework for adaptive edge steganography leverages a threshold value and the chaotic Rössler function algorithm in combination with a genetic algorithm [14]. Here, the threshold value represents the difference in brightness between two consecutive pixels, which serves as the foundational placement units for the algorithm. The chaotic Rössler function algorithm, paired with the genetic algorithm, utilizes these pixel pairs as encryption inputs to embed message bits. To enable the automated functionality of this approach across subsequent images, the improved LSB method incorporates the chaotic Rössler mapping algorithm alongside a deep LSTM neural network configured as a probabilistic structure, referred to as P-LSTM. This novel variation of LSTM, introduced in this research, diverges from the standard LSTM described in Chapter 2 by using the state  $c^*$  rather than  $h^*$  to regulate the forget, input, and output gates. Fig. 6. illustrates the internal connectivity of a P-LSTM unit.

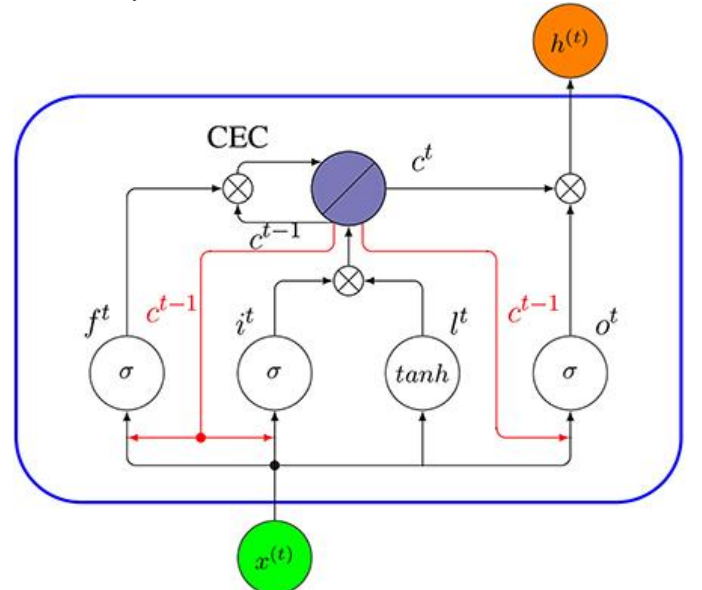


Fig. 6. P-LSTM structure of this research.

The key difference between P-LSTM and standard LSTM is that the gate  $f^t$  and its input gate and output gate do not use  $h^{t-1}$  as input. Instead, these gates use the cell state  $c^{t-1}$ . To understand the basic idea behind P-LSTM, assume that the output gate  $o^{t-1}$  in a traditional LSTM network is closed. Therefore, according to the LSTM equation, the network output



$h^{t-1}$  will be zero at time  $t - 1$ , and at the next time step  $t$ , the tuning mechanism of all three gates depends only on the network input  $x^{t-1}$ . Therefore, the historical information is completely lost. A P-LSTM avoids this problem by using the cell state instead of the output  $h$  to control the gates. Equations (3) to (8) formally describe a P-LSTM [15].

$$i^t = \sigma(W^{ix}x^t + U^{ic}c^{t-1} + b^i) \quad (3)$$

$$l^t = \tanh(W^{lx}x^t + b^l) \quad (4)$$

$$f^t = \sigma(W^{ox}x^t + U^{fc}c^{t-1} + b^f) \quad (5)$$

$$o^t = \sigma(W^{ox}x^t + U^{oc}c^{t-1} + b^o) \quad (6)$$

$$c^t = f^t \cdot c^{t-1} + i^t \cdot l^t \quad (7)$$

$$h^t = o^t \cdot c^t \quad (8)$$

The classification structure with P-LSTM is as shown in Fig. 7.

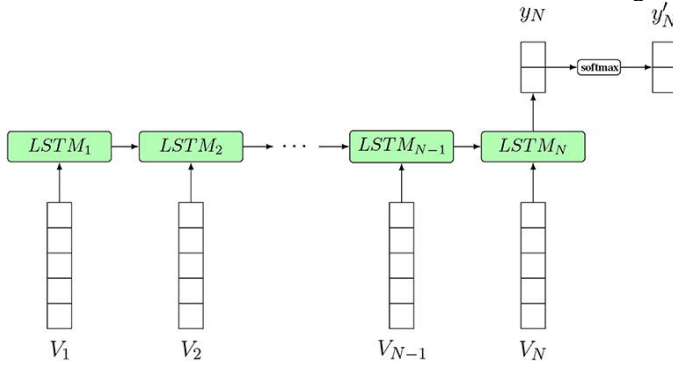


Fig. 7. Classification structure with P-LSTM.

In Fig. 7.,  $N$  represents the length of the input sequence, corresponding to the number of cover images and texts used for steganography. The sequence from  $V_1$  to  $V_N$  consists of embedded vectors, each representing the cover images provided as input to the model at various time steps. The output  $y'_N$  denotes the final outcome of text steganography embedded within the images, while  $y_N$  indicates the positioning of the texts within the cover images for the steganography process [15].

The used LSTM (Long Short-Term Memory) neural networks are RNN (Recurrent Neural Networks).

Recurrent neural networks are designed for sequential data processing. Textual data is inherently ordinal (i.e. word order is important) and LSTM is an advanced type of RNN designed to solve the "vanishing gradient" problem in traditional RNNs. This property makes LSTM very suitable for long-term dependencies in text [15].

In this study, several evaluation criteria were used, including mean square error, peak signal-to-noise ratio, signal-to-noise ratio, and accuracy criteria [16].

Firstly, every aspect of the system is thoroughly investigated, and the results derived from the research based on each evaluation criterion are documented. In this project, the mean square error (MSE) is employed as the primary metric for evaluating the system's performance. The goal of this evaluation is to ensure that the proposed approach outperforms other existing methods. It is important to note that the mean

square error typically includes a training phase, which is not considered in this study. The training phase is relevant when neural network architectures are employed; in such cases, MSE serves as both a loss function during training and an evaluation metric thereafter. During each iteration, the forward-feedback process continues until the A matrix and output values are computed using the least squares error method. It should be emphasized that all training data must consistently be applied, with initial parameters remaining fixed throughout the process. Subsequently, these fixed parameters are refined using gradient descent optimization. When the minimum MSE value is achieved, it often signifies reduced variance in erosion, with the result usually expressed as a percentage. This percentage is used to measure factors such as accuracy, sensitivity, and feature rate in the data evaluation. Since the mean square error value typically ranges from below 5 to fractions between 0.1 and 0.9, these smaller values indicate optimal performance [16]. Changes in the MSE metric can directly influence other evaluation criteria and significantly impact their results. As a result, MSE affects evaluations of accuracy, sensitivity, and feature rate by producing precise percentage figures. Additionally, it contributes to calculations involving decibel values for metrics such as peak signal-to-noise ratio (PSNR) and signal-to-noise ratio (SNR<sup>10</sup>). PSNR is another evaluation criterion applied in this project [16]. It measures the ratio between the maximum possible signal power and the noise power, expressed in decibels, and is calculated using Equation(9).

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \quad (9)$$

Where  $MAX_I^2$  is the maximum possible signal value.

#### SIGNAL-TO-NOISE RATIO

It is essential to establish a criterion to assess the ratio of useful signal to noise in the constructed project. A value below 12 signifies a severe issue with noise in the images. Values exceeding 20 indicate a satisfactory condition, while those above 30 represent an optimal level [16]. This assertion can be supported by referencing scientific literature and readily available explanations on numerous websites.

The higher this index, the better, as it signifies a more useful signal within the image. The signal-to-noise ratio is determined by the ratio of signal power to noise power, as described in equation (10).

$$SNR = \frac{P_{signal}}{P_{noise}} \quad (10)$$

Where  $P$  is the average power of the signal. Because most signals have a dynamic range, they are defined as logarithmic decibels, which is given by equation (11) for a power signal and equation (12) for a noise signal [16].

$$P_{signal,dB} = 10 \log_{10}(P_{signal}) \quad (11)$$

$$P_{noise,dB} = 10 \log_{10}(P_{noise}) \quad (12)$$

<sup>10</sup> signal-to-noise ratio

## ACCURACY CRITERIA

Classification rate or accuracy [16] is a criterion expressed in percentage and its relationship is as follows (13).

$$\text{Classification Rate} = 100 \times \frac{TP+TN}{TN+TP+FN+FP} \quad (13)$$

In equation (13), TP is true positive , TN is negative positive , FP is false positive , and FN is false negative . Equation (14) shows the sensitivity, which is expressed in percentage.

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (14)$$

Equation (15) shows the characteristics of the data , expressed in percentages.

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (15)$$

## IV. RESULTS AND DISCUSSION

During the simulation phase, standard images like Lena, along with other sample images, are utilized to identify the most suitable method for text steganography within images. A critical aspect of the process is that all selected images are in PNG format and utilize the RGB color model. The input image used is illustrated in Fig. 8.



Fig. 8. input image.

After reading the image, the color channels R,G, and B are separated. A text, such as "Nima," is then designated to be placed within the overlay image, with the objective of later extraction. The message size is calculated assuming that each character requires 8 bits. All characters in the text are represented by their respective ASCII codes, which are subsequently converted into binary. These binary values are arranged in a dedicated column to determine the size of the binarized message. Each character array is also converted into a numeric array. The red(R) color channel is selected to embed

the text within the image. During this process, a counter keeps track of the number of bits embedded and halts once all the bits are successfully embedded. Upon completing the embedding of the full message, the system exits the outer loop. The markers for the green(G) and blue(B) channels are also accounted for during this operation. Next, image traversal is performed using a chaotic genetic algorithm, which checks whether there are any remaining bits left to embed.

At this stage, the Least Significant Bit (LSB) method is used to locate the least significant bit of the current pixel, while a secondary LSB method is employed to embed two bits into the green channel. After one row is filled, the algorithm moves to the next row. Similarly, one bit is embedded in the blue channel and three in the green channel. The pattern alternates, embedding two bits in both the blue and green channels, then two more in the blue channel and three in the green channel, and finally three additional bits in the blue channel. Once this embedding process is completed, the RGB color channels are reassembled.

A neural network is then applied, to automate text processing in subsequent images. This automated procedure is optimized using the Rasler Chaos Mapping algorithm based on a genetic algorithm, which enhances the LSB method. To achieve this, a deep Long Short-Term Memory (LSTM) neural network with a probabilistic structure referred to as P-LSTM is utilized. The output showing how the text is concealed within the image is presented in Fig. 8.



Fig. 9. the output result on the image with the text hidden in it.

The shape of the cover and the output remains visually identical; however, the size of the image can reveal the modifications applied. The input image in Fig. 8. has a size of 463 KB, while the image containing steganographic text in Fig. 9. is slightly larger at 465 KB. Fig. 10. illustrates the histogram display derived from the chaotic genetic algorithm with the Rasler map, whereas Fig. 11. presents the histogram display generated using the PLSTM neural network.

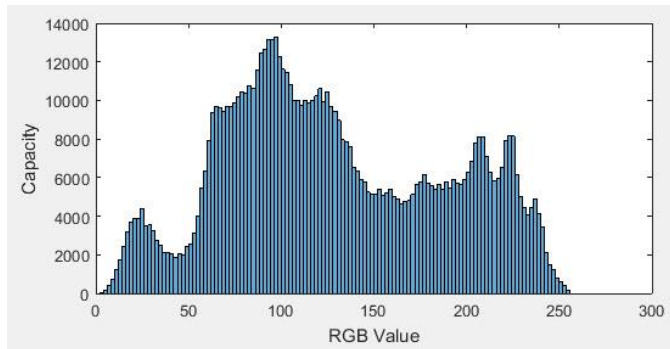


Fig. 10. Histogram display when using the chaotic genetic algorithm using the Rössler map.

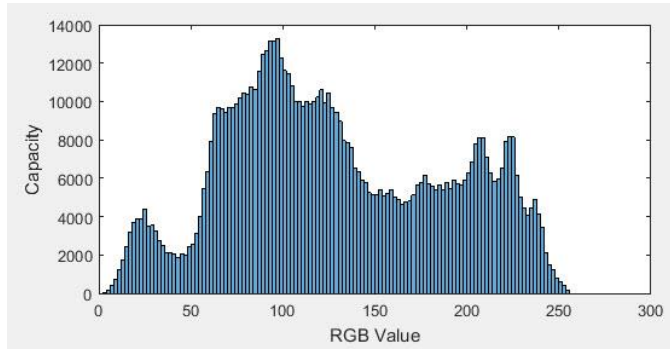


Fig. 11. shows the histogram when using the P-LSTM neural network.

To extract text from an image, the color channels red(R), green(G), and blue(B) are first separated. The process involves reading an image file, such as lena.png, which contains the embedded text. The text data within the image is extracted by isolating and processing its binary information. Each binary number is stored in a separate column, following which the overall size of the binarized message is calculated. Subsequently, the character arrays representing the text are converted into numeric arrays for further manipulation. In this method, a specific color channel, such as the red (R) channel, is used to embed and retrieve the hidden text. A counter is employed to track the number of bits being extracted throughout the process. To identify and isolate the text within the image, a hybrid technique is applied that involves traversing the image. If additional bits are detected during traversal, the text extraction process continues until completion. A key step in this procedure is examining the least significant bit (LSB) of each pixel. Additionally, exploration of the second LSB is performed, where in two bits are taken from the green channel and one bit from the blue channel. These LSBs are stored in a collection referred to as Extracted\_bits. Once extracted, the binary data is converted into its corresponding ASCII value using powers of two. To organize this data, the bits are arranged into an 8-column table, where each row represents a set of bits corresponding to a single character within the cipher text. Finally, these extracted bits are translated back into readable characters, revealing the hidden message. In this particular case, the result of the process decodes to the text "Nima". The histogram display of the extracted text image using the proposed method is shown in Fig. 12.

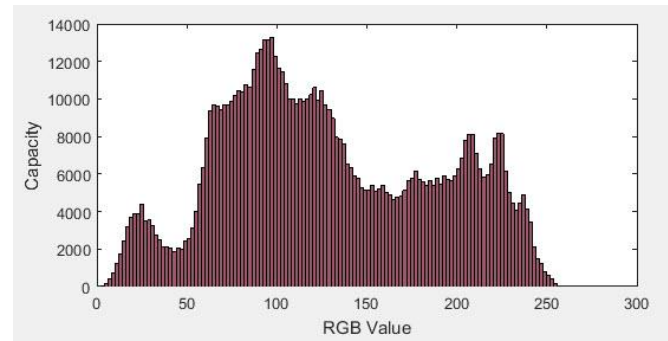


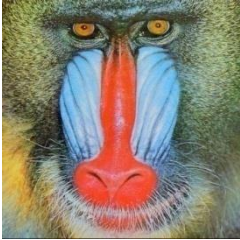



Fig. 12. histogram display during text extraction using the proposed method.

The results of evaluating the proposed method on several different images by storing the text "Nima" in each are shown in Table III.

TABLE III  
Evaluation Results of the Proposed Method

Accuracy rate in percentage	MSE (Mean Squared Error) Rate	PSNR rate in decibels (dB)	Reference image
%97.50	0.394	92.1738dB	lena.png 
%97.84	0.178	95.6261dB	peppers.png 
%96.42	0.190	92.3265dB	mandril.png 
%97.16	0.228	94.5347dB	monkey.png 
%97.23	0.6945	93.66527dB	Average results

The results were analyzed by averaging each evaluation criterion, demonstrating superior performance compared to previous methods, as shown in Table IV.

TABLE IV  
Comparison of Results with Previous Methods

Accuracy rate in percentage	MSE Rate(Mean Squared Error)	PSNR rate in decibels (dB)	Reference
%97.23	0.6945	93.665275dB	Suggested method
%95.72	0.6971	92.16dB	Lingamallu Naga Srinivasu, and Vijayaraghavan Veeramani, 2022
%96.68	0.6837	93.10dB	A. Yousefian Darani, et al, 2023
%96.73	0.6942	92.78dB	Keshav Kaushik, and Akashdeep Bhardwaj, 2021

## V. CONCLUSION

In conclusion, information security has gained critical importance with the rapid advancement of multimedia technology. Steganographic techniques revolve around two essential components: the cover medium (image, e.g.) and the hidden data (text, images, audio, or video). In traditional methods, images are preferred as cover media due to their complex textures and edges, which make detection more challenging once data embedding is carried out. Presently, many steganographic methods embed relatively small amounts of hidden data to evade detection. These techniques typically fall under the broader umbrella of spatial domain and transform domain approaches. Spatial domain methods, such as LSB (Least Significant Bit), WOW (Wavelet Obtained Weights), HUGO (Highly Undetectable Steganography), and SUNIWARD (Universal Wavelet Relative Distortion), directly alter pixel values in the source image. However, such modifications often degrade the visual quality of the image, which may raise suspicion during analysis. On the other hand, transform domain methods like DWT (Discrete Wavelet Transform), DCT (Discrete Cosine Transform), and IWT (Integer Wavelet Transform) operate on subbands of images to address perceptual quality concerns. Nonetheless, adjusting subbands is a complex process that must be tailored for specific cases; otherwise, visible artifacts may appear in the final output.

In order to confirm the validity of the results, various images and different words and texts have been used. The images and text mentioned are examples.

Among the advantages of the presented method compared to previous similar methods, it is possible to improve the result of PSNR and MSE, as well as sufficient and appropriate accuracy, finding the best embedding place, suitable, fast, effective, efficient and flexible embedding capacity, availability, existence of encoder and decoder system, High execution speed in about 10 seconds in the system with 7-core processor with cache speed of 6 and 16 MB of memory, and finally the reduction of computational complexity.

This is the time to run the entire encryption and decryption algorithm and extract all the parameters of the algorithm. The execution time of the encryption algorithm is less than 5 seconds, and the time to recover and reveal the image is less than 5 seconds.

In terms of complexity, the larger the key size of an encryption system, the less the possibility of possible attacks. Larger keys add more complexity to the system. In the designed system, according to the use of LSB algorithm and chaotic genetic algorithm, a relatively favorable situation is established in the stealth system.

In this study, three evaluation metrics PSNR (Peak Signal-to-Noise Ratio) in decibels, MSE (Mean Square Error), and accuracy in percentage were analyzed using four sample images: lena.png, peppers.png, mandril.png, and monkey.png. The results demonstrated average values of 93.665 dB for PSNR, 0.6945 for MSE, and 97.23% for accuracy. These findings showed superior performance compared to three reference methods. Overall, steganography continues to evolve with new algorithms and techniques that balance invisibility, security, and robustness while adapting to modern-day challenges in secure communication across multimedia channels.

The evaluation of the proposed algorithm by examining the peak signal-to-noise ratio (PSNR) and the results of the error indices (MSE) compared to similar tasks shows the good performance quality and the improved robustness of the algorithm. Despite the hidden opinion, the better the noise, the less chance of error and the less change between the original image and the rendered image.

In terms of capacity, due to the use of 2LSB algorithm, a better capacity has been created than other similar methods. Due to the use of chaotic genetic algorithm, optimal embedding is also done. According to the generated capacity and the length of the password message, it is possible to perform encryption operations several times. LSB cryptography methods provide good capacity for cryptographic operations.

## VI. PRESENTING FUTURE SOLUTIONS

Future solutions in the field of image watermarking can leverage a variety of advanced methods, including machine learning, fuzzy logic, and mapping techniques. Neural networks represent a highly versatile set of tools within the machine learning family. They can function in various capacities, such as trainers, evaluators, compressors, or edge detectors. For future applications in watermarking, specialized neural networks may be explored, including Hopfield neural networks, cellular neural networks, probabilistic neural networks, radial basis function (RBF) neural networks, Grossberg neural networks, error back propagation neural networks, positive-time back



propagation networks, forward perturbation networks, counter-propagation networks, and other models often rooted in recurrent neural network algorithms. Evolutionary algorithms and swarm intelligence methodologies also offer great potential for image watermarking due to their high speed, robustness, and adaptability. While each has unique advantages and potential trade-offs, they collectively highlight the range of opportunities and ideas for future exploration.

The combination of genetic algorithm and crowd intelligence: the combination of these two algorithms can lead to an increase in the efficiency and power of the encryption system. For example, the genetic algorithm can be used to initially search the space and find potential areas for embedding, and then use crowd intelligence to accurately optimize the parameters in those areas. This combination leads to increasing the speed, accuracy and security of encryption.

Challenges and opportunities: Although the combination of genetic algorithm and crowd intelligence in cryptography has many advantages, there are also challenges. One of the challenges is the computational complexity of these algorithms. Finding a balance between security, the amount of embedded information and the amount of visible change in the image is another challenge. However, research in this field is progressing rapidly and there are many opportunities to develop more secure and efficient encryption methods using these algorithms. Future research can focus on improving computation speed, providing new hybrid algorithms, and developing attack-resistant methods. This powerful combination can lead to encryption systems with higher power and security and open new fields in information security and data protection.

Decision trees are used to categorize and select appropriate locations in the image for hidden information. Neural networks can be used for complex patterns in images and detecting points that have the least impact on visual quality. For example, a neural network can be trained to identify points where small changes are less discernible to humans in an image.

Evolutionary algorithms (such as genetic algorithm) can be used to optimize the decision tree parameters to select the best points for hiding. These algorithms can automatically create decision trees that are able to select points with the lowest noise and the highest hiding capacity.

Evolutionary algorithms can be used to optimize the structure and weights of neural networks to improve their performance in cryptography. Neural networks can learn complex patterns in images, and evolutionary algorithms can tune these networks to perform best at hiding information.

Using mappings means converting textual information into another format (such as a specific code or pattern) that is easier to hide in an image. These maps can be optimized with the help of decision trees, neural networks or evolutionary algorithms so that the information is placed in the image in a way that causes the least visual changes.

In a more sophisticated way, all these techniques can be combined. For example, a hybrid algorithm can use decision trees to select suitable points, neural networks to learn image patterns, evolutionary algorithms to optimize parameters, and maps to transform data.

The use of other image processing methods such as morphology-based segmentation can also be used as a new idea in steganography by combining the previously mentioned methods. Also, methods combining decision trees with neural networks or with evolutionary algorithms and mappings, or a combination of any of these, can be considered a new method in the work. Of course, it should be noted that the combined modes may provide high speed and flexibility, but the execution time and computational complexity may increase exponentially.

## REFERENCES

- [1] P. N. Shingote, A. Bhujbal, P. M. Syed, Advanced Security Using Cryptography and LSB Matching Steganography, *IJCER*, 3(2), 52-55, 2014
- [2] F. G. Mohammadi, M. S. Abadeh, Image steganalysis using a bee colony based feature selection algorithm., *Engineering Applications of Artificial Intelligence*, 2014
- [3] Z. V. Patel, S. A. Gadhiya, A Survey Paper on Steganography and Cryptography, *International Multidisciplinary Research Journal (RHIMRJ)*, Volume-2, Issue-5, 2015
- [4] Y. Wang, X. Liu, Z. Gao, "Chaos-Based Image Steganography," *IEEE Internet Things J.*, vol. 10, no. 2, pp. 1478–1487, 2023.
- [5] M. Juneja, P. S. Sandhu, Improved LSB based Steganography Techniques for Color Images in Spatial Domain. *IJ Network Security*, 16(4), 366376, 2014
- [6] Y. Wang et al., "Chaos-Based Image Steganography," *IEEE IoT J.*, 2023.
- [7] R. C. Gonzalez, R. E. Woods, *Digital Image Processing* translator: Ainullah Jafarnejad Qomi, publisher: Computer Science Publications, 4th edition, 2021
- [8] H. Sajedi and Sh. R. Yaghoubi, "Review on Steganographic Methods in Texts", 2018.
- [9] D. Gomez, R. Martinez, J. Lopez, "LSTM-Assisted Data Hiding in Color Images," *IEEE Access*, vol. 10, pp. 109123–109135, 2022.
- [10] S. M. Hosseini, A. Nemati, Investigation of image-based information encryption methods with the approach of increasing the security of information and communications against multiple attacks, *Specialized Scientific Quarterly of New Technologies in Electrical and Computer Engineering*, Volume 3, Number 4, 2022
- [11] T. Nguyen, P. Tran, H. Huynh, "Deep LSTM-Based Image Steganography," *IEEE Access*, vol. 9, pp. 87632–87645, 2021.
- [12] N. Patel, M. Patel, K. Joshi, "Robust Steganographic System Using Autoencoders," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 220–233, 2022.
- [13] A. Sabhany, A. Ahmed, H. Ali, R. Mokhtar, "Digital audio steganography: Systematic review," classification, and analysis of the current state of the art. *Computer Science Review*, Volume 38, November 2020
- [14] D. Wenbo, Z. Mingyuan, Y. Wen, P. Matjaž, W. Dapeng, "The networked evolutionary algorithm: A network science perspective", *Applied Mathematics and Computation*, Volume 338, Pages 33-43, 1 December 2018
- [15] E. Asadzadeh, M. Asadzadeh, M. Asadbek Genetic Algorithm, the First Publications, 2018
- [16] A. R. Mirqadari and A. Jolfaei, "A new scheme for image encryption using chaotic mappings," *Modern Defense Sciences and Technologies (Non-Agent Defense Sciences and Technologies)*, vol. 2, no. 2 (series 4), pp. 111-124, 2019,