Power System Technology
ISSN:1000-3673

Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

Secure and Imperceptible Image Steganography Using Dual Encryption with ACO-Guided Pixel Selection

GD Makkar[1], Rohan Verma[2], Sachin Jain[3], Vrinda Tandon[4], Rohit Goyal[5], Apoorva Talwar[6]

[1]School of Engineering & Technology, Shri Guru Ram Rai University, Dehradun, India

[2] Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, India

[3,4]School of Science and Technology, Jigyasa University, Dehradun, India

[5]School Of Engineering and Computing, Dev Bhoomi Uttarakhand University, Dehradun, India

[6]Department of Computer Application, Tula's Institute, India

[1]gdmakkar@gmail.com, [2]rohanvermahnb@gmail.com, [3]sachinjain23jan@gmail.com, [4]vrinda1804tandon@gmail.com, [5]rohitgoyalhzu@gmail.com
[6]apoorvaresearch06@gmail.com

[1]Orcid-0000-0003-4507-3636, [2]Orcid-0000-0002-2140-9329, [3]0009-0009-3656-5269, 0009-[4]0006-1823-048X, [5]0000-0001-7978-4079, [6]Orcid- 0009-0000-6831-2578

Abstract

This paper presents a hybrid image steganography scheme that combines cryptography with intelligent embedding to improve imperceptibility, robustness, and security. The secret payload is first encrypted using AES-GCM for confidentiality and integrity, then further scrambled via a classical Rail Fence cipher. Next, an Ant Colony Optimization (ACO) algorithm intelligently selects an adaptive, non-sequential pixel embedding path, and the encrypted bits are hidden using a round-robin LSB substitution across the image's RGB channels. Experimental results demonstrate that this multi-layer approach achieves high imperceptibility, with PSNR reaching ~74 dB at small payload of 64B and ~56 dB at 5 KB payload, while the Structural Similarity Index (SSIM) consistently stays above 0.999 across all payloads. The method also exhibits strong robustness, with extracted data showing a low BER (~0.25%) under common image processing attacks e.g., noise and compression. Compared to conventional LSB or single-layer steganography techniques, the proposed approach achieves significantly higher image fidelity e.g., ~30 dB PSNR gain over basic LSB at a similar payload and provides security advantages through dual-layer encryption absent in earlier ACO-LSB schemes. These results highlight the benefits of integrating encryption with ACO-based embedding to attain imperceptible, robust, and secure image steganography.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

Keywords: Image steganography, AES-GCM encryption, Rail Fence cipher, Ant Colony Optimization (ACO), LSB substitution

1. Introduction

Steganography, derived from the Greek words "steganos" (covered) and "graphein" (writing), is the art and science of concealing secret information within a non-secret medium, such as digital images, audio, or video, in a manner that remains undetectable [1]. Unlike cryptography, which protects data through encryption, steganography ensures that the communication itself is hidden [2]. Digital images, due to their widespread use and high redundancy, are one of the most common carriers for steganographic applications [3].

Traditional steganographic methods, such as Least Significant Bit (LSB) substitution, modify the least perceptible bits of pixel values to embed data [4]. However, they are susceptible to detection and manipulation. Simply hiding raw text within an image is insufficient for high-security applications. If an attacker detects the hidden message, they can easily extract and read it. Therefore, pre-encrypting the data before embedding significantly enhances security [5]. Encrypting the message adds an extra layer of security, addressing both the confidentiality and integrity aspects of secure communication [6]. However, even with encryption, predictable embedding patterns in static LSB-based methods can be exploited using statistical or visual steganalysis tools [7].

A major weakness of traditional LSB steganography is its sequential embedding pattern, making it detectable through statistical analysis [4]. To address this limitation, intelligent pixel selection strategies are needed to distribute the embedded bits in a less predictable, more natural manner [8]. Inspired by the behavior of real ants foraging for food, ACO enables adaptive and non-linear traversal through image pixels based on pheromone trails and heuristic information such as edge intensity to guide probabilistic decision-making, enabling a dynamic and adaptive traversal of the image's pixel space. This reduces predictability and enhances robustness against detection [1].

In this paper, we propose a hybrid steganographic framework that combines multiple layers of security and adaptability. First, the secret text is encrypted using the AES-GCM (Advanced Encryption Standard - Galois/Counter Mode) algorithm, which ensures both confidentiality and authentication of the message. Next, the encrypted ciphertext undergoes a Rail Fence cipher transformation, a classical transposition technique, to add further obfuscation. The resulting binary message is then embedded into an image using ACO-guided traversal, where the ants probabilistically select embedding positions based on edge gradient information. Finally, LSB embedding is applied in a round-robin manner across the RGB channels of the selected pixels.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

The effectiveness of the proposed method is evaluated using standard performance metrics. Imperceptibility is measured through Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) to ensure the visual quality of the stego images is maintained. Robustness is assessed using Bit Error Rate (BER) under common image processing attacks such as noise addition and compression. The embedding capacity is also analyzed to determine the method's scalability for varying payload sizes.

The rest of this paper is organized as follows: Section 2 reviews related work on steganography and optimization-based embedding. Section 3 presents the proposed methodology in detail, including encryption, Rail Fence obfuscation, ACO traversal, and LSB embedding. Section 4 discusses various algorithms used for encryption, ACO based pixel selection, LSB embedding process and decryption process. Section 5 discusses the experimental setup, performance metrics. and results & analysis. Finally, Section 6 concludes with the security analysis of the proposed methodology.

2. Literature Survey

A clear trend in recent years is the integration of cryptographic encryption with image steganography to achieve dual-layer security. By encrypting the secret payload before hiding it, even if the steganographic concealment is partially compromised, the message remains unintelligible without the decryption key [2]. Modern steganographic systems frequently employ strong ciphers like AES, RSA, or ECC prior to embedding. For instance, Alanzy *et al.* [2] propose a multi-level scheme using dual encryption AES alongside Blowfish to secure the cover image and its payload, with encryption keys themselves embedded as "key images" in the stego image. This hybrid encryption approach yielded high peak signal-to-noise ratio (PSNR) and low mean squared error (MSE) in the stego images, indicating that the added security did not noticeably degrade image quality [2]. In a similar vein, Kadhuma *et al.* [10] combined AES in Galois/Counter Mode (AES-GCM) with LSB image embedding for sensitive data protection; their system achieved imperceptible distortion (PSNR ~70.95 dB) and zero observable correlation between the original plaintext and the encrypted-stego data. Such results underscore that cryptography can be layered with steganography to greatly enhance confidentiality without sacrificing cover image fidelity. Simpler two-layer approaches have also been explored: for example, Gurav *et al.* [11] demonstrated a lightweight scheme in which a secret text is first encrypted with a classical Rail Fence transposition cipher and then hidden in an image via LSB substitution. While classical ciphers like the Rail Fence provide only modest security by modern standards, they add an extra obfuscation layer that, when combined with strong encryption, can further confuse attackers [11]. Combining encryption with image steganography significantly hardens the covert communication channel, making it far more difficult for an adversary to detect the hidden content or decipher it if discovered.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

Another major development in image steganography is the use of metaheuristic optimization algorithms to intelligently select embedding locations. The goal is to maximize payload capacity and security while minimizing distortion to the cover image [6]. A comprehensive review by Gnanalakshmi and Indumathi [6] surveys state-of-the-art optimization-based methods including genetic algorithms (GA), particle swarm optimization (PSO), ant colony optimization (ACO), and concludes that these techniques substantially improve the capacity-versus-imperceptibility trade-off in image steganography. Among these, ACO has gained particular prominence for image-based steganography. Boryczka and Kazana [3] explored ant-based algorithms in both spatial and frequency domains, using ACO to identify optimal regions in an image where secret data can be embedded with minimal visual impact. In their method, an integer wavelet transform was first applied to the cover image for robustness, and then ACO swarms were employed to find the best wavelet coefficients and pixel locations for hiding information. This approach allowed high-capacity embedding with little quality degradation, as evidenced by the ability to hide large payloads while keeping the stego images visually indistinguishable from the originals [3]. In the spatial domain, Jasim and Kurnaz [12] developed an ACO-optimized LSB scheme (ACO-LSB) that adaptively chooses pixel positions for insertion. Their ACO-LSB method achieved up to 30% higher embedding capacity than standard LSB techniques at a given distortion level. Notably, even with the increased payload, the average stego image quality remained high (PSNR $\approx$ 40.5 dB, SSIM $\approx$ 0.98) and the method showed about 20% lower detectability under steganalysis attacks. Such results illustrate the benefit of optimization by steering modifications toward complex or textured regions of the image where changes are less noticeable and algorithms like ACO can hide more data while evading statistical detection.

Recent research has started to fuse cryptographic methods with optimization-driven steganography, yielding hybrid techniques that address multiple aspects of security. Ahmed and Salah [13] introduced a robust image steganography framework that combined ACO with a transform-domain embedding to hide a secret image within a cover image. Their system applied a block-based discrete cosine transform (DCT) to the cover image for improved resistance to image processing attacks, and simultaneously used an ACO algorithm to determine optimal DCT coefficient locations for insertion. Importantly, they incorporated a secret-key encryption stage as well. The overall pipeline included steps for key generation, encryption, ACO-guided embedding, and inverse transformation, embodying a multi-layer defense. This approach not only hides the data effectively but also ensures that an intercepted message would remain protected by encryption and that the stego image can withstand common attacks.

Each component of a steganographic system plays a role in security, and combining them multiplies the overall effectiveness. For example, encryption like AES or even simple ciphers like Rail Fence secures the content of the payload [3], while intelligent embedding via ACO,

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

GA, etc. secures the existence of the payload by minimizing statistical traces [12]. Multi-encryption schemes like using two or more ciphers can further complicate an adversary's job, as noted by Alanzy *et al.* [2], and likewise multi-objective optimization can balance hiding capacity and image fidelity better than any fixed rule. However, relatively few works have integrated *all* these layers in one system. This literature shows examples of two-layer combinations i.e., encryption + LSB hiding. The hybrid method proposed in our work i.e., AES-GCM encryption, Rail Fence transposition, ACO-based pixel traversal, and LSB embedding represents a novel synthesis. By leveraging an authenticated cipher (AES-GCM) for strong confidentiality and integrity of the secret, a classical transposition cipher for an added diffusion layer, and ACO-optimized LSB placement for maximal concealment. The proposed approach aligns with the cutting edge of steganography research. This hybrid ensures robust security against both cryptanalytic and steganalytic attacks. Our approach provides high imperceptibility, increased payload capacity, and resistance to statistical and image-processing-based steganalysis.

3. Proposed Methodology

3.1 System Overview

Our proposed system combines advanced cryptographic algorithms with biologically-inspired optimization techniques to create a robust steganographic method. The system operates in two major phases: first encrypting the message with a hybrid cryptographic approach, and then embedding it within an image using an Ant Colony Optimization (ACO) guided path selection for Least Significant Bit (LSB) embedding. The methodology follows a sequential process flow:

Figure 1 show the system architecture diagram for the proposed methodology: Input Text → Encryption (AES-GCM + Rail Fence) → Binary Conversion → ACO Path Selection → LSB Embedding → Stego Image.

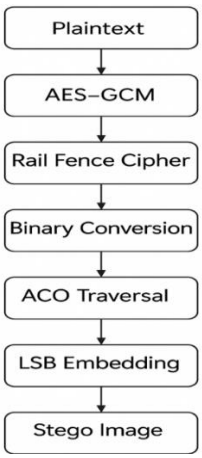

Figure 1: System architecture diagram for the proposed methodology

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

1 **Message Pre-processing**: The original plaintext message is prepared for secure transmission.

2 **Primary Encryption (AES-GCM)**: The message undergoes symmetric encryption using the Advanced Encryption Standard in Galois/Counter Mode.

3 **Secondary Encryption (Rail Fence Cipher)**: The encrypted ciphertext is further transposed using a classical Rail Fence cipher with a dynamically derived key.

4 **Binary Conversion**: The twice-encrypted message is converted to a binary sequence with added integrity verification components.

5 **Cover Image Analysis**: The carrier image is analyzed to identify optimal embedding regions based on edge characteristics.

6 **ACO Path Selection**: A biologically-inspired algorithm determines the traversal path through the image for embedding.

7 **LSB Embedding**: The binary data is embedded into the least significant bits of pixel values along the determined path.

8 **Stego Image Production**: The modified image containing the hidden encrypted message is output as the final stego medium.

The methodology ensures both security through strong encryption and stealth through intelligent embedding that follows the natural contours of the image, making statistical detection more difficult.

3.2 Encryption Phase

3.2.1 AES-GCM Encryption

We employ the Advanced Encryption Standard (AES) in Galois/Counter Mode (GCM) as the primary encryption algorithm due to its:

- Strong confidentiality guarantees through symmetric encryption [14]
- Built-in authentication and integrity verification [14][15]
- Parallel processing capabilities for improved performance. [15]

AES-GCM provides authenticated encryption with associated data (AEAD), enabling us to detect any tampering with the ciphertext. The algorithm uses a 128-bit key and a 12-byte nonce is randomly generated to ensure encryption uniqueness [14]. The nonce ensures that identical messages produce different ciphertexts, preventing pattern analysis. During encryption, GCM simultaneously computes an authentication tag that serves as a message integrity check. The nonce, authentication tag, and ciphertext are concatenated and encoded in Base64 format and produces a ciphertext along with an authentication tag [15].

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

3.2.2 Rail Fence Cipher Enhancement

The Rail Fence Cipher adds a secondary layer of security through transposition rather than substitution. The key (determining the number of "rails" or rows) for the Rail Fence cipher is dynamically derived from the AES ciphertext itself, creating a dependency that makes unauthorized decryption more challenging. The key is derived as:

The derivation process selects hexadecimal characters from the AES-GCM ciphertext and converts them to an integer.

- A random 4-character substring (e.g., "7b4f") is selected from the hex character list. This introduces unpredictability.
- The 4-character hex string is converted to an integer.

  int("7b4f", 16) = 31567

  Modulo: 31567 % 16 = 15

- The integer is mapped to a range between 5 and 20 to produce key for rail fence encryption.
  Final key: 15 + 5 = 20

The AES ciphertext is arranged in a zigzag pattern across multiple rows according to the key. Characters are then read row by row to produce the transposed output. The resulting transposed ciphertext is then converted into a binary stream, where each character is represented using an 8-bit binary format. A checksum is computed by summing the ASCII values of the original plaintext modulo 256, and its 8-bit binary equivalent is appended to the binary stream, followed by a null-byte (00000000) to mark the end of the message.

Even if an attacker could identify that LSB steganography is being used, they would first encounter transposed ciphertext. Breaking the Rail Fence cipher would only reveal another layer of encryption (AES-GCM). The relationship between the two encryption methods creates a system where knowledge of one layer provides minimal information about the other.

3.3 ACO-Based Pixel Traversal

3.3.1 Pheromone and Heuristic Map Initialization

This stage aims to intelligently select pixel positions in the cover image for embedding each bit of the encrypted message. Traditional LSB methods use sequential or static pixel selection, which can be easily analyzed or predicted [6]. To address this, we employ Ant Colony Optimization (ACO), a metaheuristic inspired by the foraging behavior of ants [6] [9].

The pheromone map generation process converts the cover image to grayscale and a Gaussian filter is applied to reduce noise and enhance structural regions. The gradient is

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

computed in both x and y directions to detect edge regions. The pheromone map is derived from edge strength (gradient magnitude of a smoothed grayscale image), normalized to [0,1], while the heuristic map is its inverse (1 - pheromone) [9]. These maps guide ants toward regions of interest (e.g., edges) while avoiding overused paths.

Ants traverse the image based on a probability function that considers both pheromone intensity (τ) and heuristic value (η), balanced by parameters α and β [8]. The probability of choosing a neighbouring pixel (i, j) is given by:

$$P_{ij} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k=allowed}[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}$$

where τ_{ij} is the pheromone level, η_{ij} is the heuristic value, and α and β (both set to 1.0) balance their influence. If an ant encounters a dead-end (all neighbors visited), a breadth-first search (BFS) is triggered to find the next unvisited pixel. This guarantees complete traversal and avoids embedding conflicts or overlaps.

This step is critical in guiding the ants (embedding agents) to traverse meaningful areas in the image, such as edges or high-detail regions, where data embedding is less perceptible [8] [9]. The following procedure is used for the Pheromone Map and initializing heuristic map and we are showing each step for an image.

(A) Grayscale Conversion

Convert the RGB image to grayscale by averaging the color channels:

$$G(x,y) = \frac{I_R(x,y) + I_G(x,y) + I_B(x,y)}{3}$$

Where $I(x,y)$ is an original RGB image and $G(x,y)$ is a Grayscale image computed as the mean of RGB channels. Figure 2(a) and 2(b) shows the original and grayscale image and table 1 represent 5×5 grayscale patch extracted from the center of the image after RGB-to-grayscale conversion.

(B) Apply Gaussian Smoothing

Use a Gaussian filter $\mathcal{G}_{\sigma}$ with standard deviation σ =1.0 to reduce noise:

$$Smooth(x,y) = \mathcal{G}_{\sigma} * G(x,y)$$

Where $Smooth(x,y)$ is a smoothed grayscale image after applying Gaussian filter. Figure 2(c) shows the Gaussian smoothing image. This table 2 shows a representative 5×5 patch extracted from the center of the image after applying a Gaussian filter (σ = 1.0) to the grayscale image. The Gaussian smoothing reduces high-frequency noise and softens abrupt intensity transitions. As a result, the pixel values in the smoothed patch are more moderated

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

compared to those in the original grayscale image. For example, while the original grayscale patch exhibited a wide range from very low values (e.g., 4) to high values (e.g., 244), the smoothed patch shows a more uniform distribution, with values ranging approximately from 24 to 186. This more gradual change in intensities aids in robust gradient computation later in the process, ensuring that the pheromone map derived from the gradients accurately reflects the underlying edge structure with reduced influence from noise.

(C) Compute Gradient Magnitude

Image gradients are computed in both directions:

$$\nabla_x(x,y) = \frac{\partial Smooth(x,y)}{\partial x}, \nabla_y(x,y) = \frac{\partial Smooth(x,y)}{\partial y}$$

Then computed edge strength is used as initial pheromone values.

$$E(x,y) = \sqrt{\nabla_x(x,y)^2 + \nabla_x(x,y)^2 + \varepsilon}$$

Where $\varepsilon = 10^{-10}$ is used to ensure stability. Figure 2(d) shows the gradient magnitude image. Table 3 displays a representative 5×5 patch extracted from the gradient magnitude image, which is computed using Sobel filters applied to the smoothed grayscale image. The displayed values, ranging from approximately 32 to 468, indicate the degree of intensity change across adjacent pixels. Regions such as the middle rows exhibit high gradient magnitudes (e.g., values exceeding 400), indicating strong intensity transitions often corresponding to edges or textured areas. In contrast, the top-left corner shows lower gradient values (around 32 to 58), suggesting smoother intensity variations in these regions. The considerable range within the patch reflects the heterogeneous nature of the image's local features. This variability is critical as it influences the subsequent pheromone map initialization, where higher gradients receive higher pheromone values.

(D) Normalize Edge Strength for Pheromone Map

Normalize edge strength to [0, 1]:

$$P(x,y) = \frac{E(x,y)}{max(E)}$$

This becomes the initial pheromone map $\tau(x,y)$. Figure 2(e) shows image obtained after mapping the pheromone. The table 4 presents a 5×5 patch extracted from the normalized pheromone map, which is derived from the gradient magnitude of the smoothed grayscale image. Where $E(x,y)$ is the gradient magnitude and $max(E)$ is the maximum gradient value within the image. This normalization confines the values to a range between 0 and 1, with higher values indicating regions of strong intensity transitions (i.e., edges). The upper-left portion of the patch shows relatively low pheromone values (e.g., 0.0546, 0.0988), suggesting

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

a smoother region with minor intensity variations. The central region exhibits higher pheromone values (e.g., 0.4159, 0.6139, up to 0.7803), indicative of strong edges or significant local contrast. These areas are crucial for guiding the ant-based optimization since they represent structural details where embedding modifications are less perceptible.

The range of values—from approximately 0.05 in smoother areas to values close to 0.78 in edge-rich regions—demonstrates the heterogeneous nature of the image. This variability is essential as it allows the algorithm to balance between exploitation (favoring prominent edges) and exploration (allowing traversal through smoother areas).

(E) Heuristic Map

The heuristic map $\eta(x, y)$ is computed as:

$$\eta(x, y) = 1 - \tau(x, y)$$

So, higher edge strength (more visible) → high pheromone, low heuristic. Figure 2(f) shows the heuristic map image. The table 5 shows a representative 5×5 patch extracted from the heuristic map, which is computed as the complement of the pheromone map. Because the heuristic map is the inverse of the pheromone map, it assigns higher values to regions with relatively low edge strength and lower values to regions with strong edges. This balance encourages exploration of smoother areas where subtle changes are less likely to be visually detected, while still allowing the algorithm to consider salient image features.

In some parts of the patch, the values are very high (e.g., 0.9454, 0.9012), indicating regions with low gradient magnitudes—these areas are likely to be smooth and contain fewer edges. Other areas in the patch display lower heuristic values (e.g., 0.2197, 0.2311), particularly where the corresponding pheromone values were higher due to pronounced edges or texture. The overall distribution, with values ranging from approximately 0.22 to 0.95, reflects the heterogeneity of the image. This variability is crucial for the Ant Colony Optimization process, as it supports a balanced trade-off between exploring new areas (high heuristic values) and exploiting known structural details (lower heuristic values).

Table 6 shows the summary of the pixel values for Grayscale, Smoothed, Gradient, Pheromone, and Heuristic.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

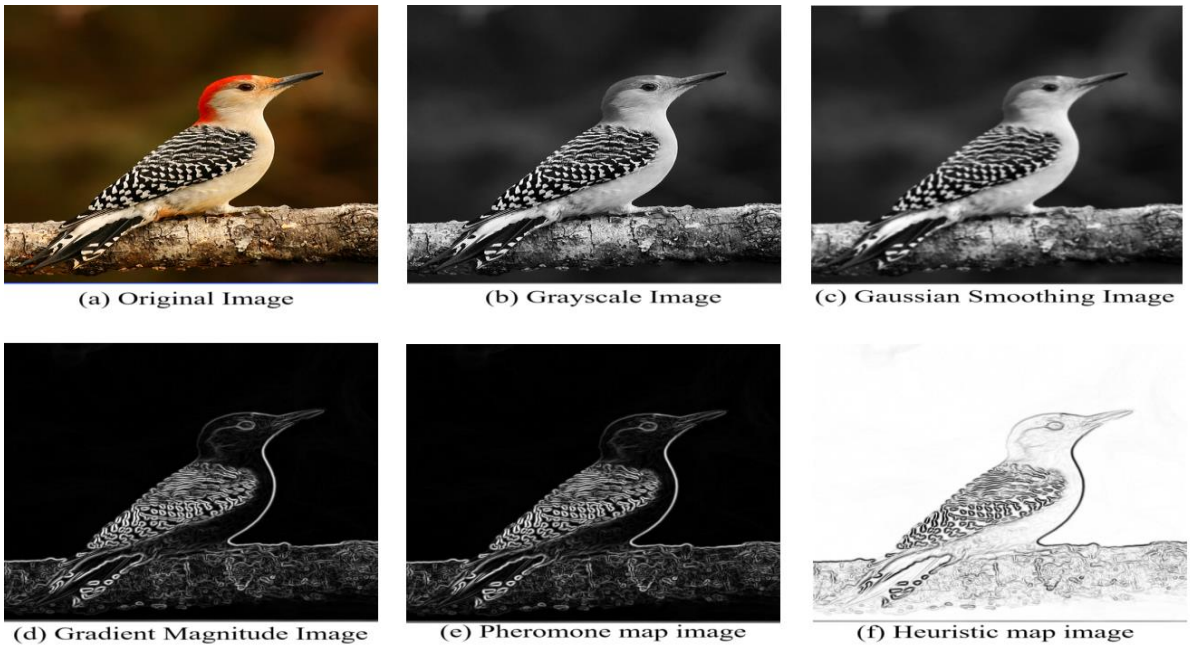

Figure 2: Pheromone and heuristic map initialization process: (a) Original image, (b) grayscale conversion, (c) Gaussian smoothing, (d) gradient magnitude image, (e) pheromone map, and (f) heuristic map.

Table1: Grayscale patch extracted from the center of the image after RGB-to-grayscale conversion

```
Grayscale Patch:
[[244.33333333 234.66666667 227.33333333 233.66666667 166.66666667]
 [198.66666667 155.66666667 173.66666667 120.33333333  86.66666667]
 [ 70.          42.33333333  43.          22.          21.        ]
 [  4.           9.66666667   4.33333333   7.33333333  43.66666667]
 [ 12.33333333  12.66666667   8.          18.33333333 172.66666667]]
```

Table2: Patch extracted from the center of the image after applying a Gaussian filter to the grayscale image

```
Smoothed Patch:
[[186.4147958  186.53156226 183.57813913 172.64355642 146.72862648]
 [149.73046266 146.39406325 136.99387324 118.9854927   91.78606633]
 [ 79.99791319  76.12107968  68.94046528  61.25176467  55.1071717 ]
 [ 31.50871592  31.85528089  33.45868716  44.74499107  65.25236569]
 [ 24.43973958  34.90754564  48.85501324  75.2823705  110.12170977]]
```

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

Table3: Patch extracted from the gradient magnitude image

```
Gradient Patch:
[[ 32.11594264  58.08270984 127.86954192 244.5428046  360.90482597]
 [428.23191128 443.62569763 458.75452235 451.64855028 398.02841397]
 [468.30518094 452.03346533 398.31605004 281.56289954 117.68631386]
 [214.29512825 159.00998328  84.65626256 127.64555125 233.00392423]
 [116.90722374 204.47336743 302.83187986 383.35147439 379.60618379]]
```

Table 4: Patch extracted from the normalized pheromone map

```
Pheromone Patch:
[[0.05462702 0.09879472 0.21749734 0.41595058 0.61387442]
 [0.72839319 0.75457697 0.78031006 0.76822329 0.67701911]
 [0.79655508 0.76887801 0.67750836 0.47891923 0.20017637]
 [0.36450136 0.27046511 0.14399451 0.21711635 0.39632374]
 [0.19885119 0.34779521 0.51509631 0.65205464 0.64568415]]
```

Table 5: Patch extracted from the heuristic map

```
Heuristic Patch:
[[0.94537298 0.90120528 0.78250266 0.58404942 0.38612558]
 [0.27160681 0.24542303 0.21968994 0.23177671 0.32298089]
 [0.20344492 0.23112199 0.32249164 0.52108077 0.79982363]
 [0.63549864 0.72953489 0.85600549 0.78288365 0.60367626]
 [0.80114881 0.65220479 0.48490369 0.34794536 0.35431585]]
```

Table 6: Summary of the pixel values for Grayscale, Smoothed, Gradient, Pheromone, and Heuristic

```
Pixel Value Summary:
        Step       Min         Max       Mean     StdDev
0  Grayscale  0.000000  255.000000  60.367998  55.090068
1   Smoothed  2.345656  240.534037  60.367998  50.738803
2   Gradient  0.005838  587.913118  45.463918  87.092626
3  Pheromone  0.000010    1.000000   0.077331   0.148139
4  Heuristic  0.000000    0.999990   0.922669   0.148139
```

3.3.2 Probabilistic Pixel Selection by Ants

The core of our ACO-based steganography approach lies in how "virtual ants" select pixels for embedding through a probabilistic decision-making process that mimics natural ant foraging behavior [3] [16]. This selection process is governed by two key information sources: pheromone levels and heuristic information [16].

When selecting the next pixel for the embedding path, the ant located at position(i, j) evaluates all candidate neighbor pixels according to a probability distribution based on

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

Dorigo's ACO formulation [17]. Each "ant" at pixel (i, j) chooses its next move to a neighboring pixel (k, l) according to a probability that balances two factors:

- **Pheromone intensity** $\tau(k, l)$, which encodes the desirability of previously successful paths and here proportional to edge strength, and
- **Heuristic information** $\eta(k, l)$, which encourages exploration of less-visited or smoother regions [16].

The transition probability from (i, j) to (k, l) is given by:

$$P_{(i,j)\to(k,l)} = \frac{[\tau(k,l)]^{\alpha}\,[\eta(k,l)]^{\beta}}{\Sigma_{(u,v)\in\mathcal{N}(i,j)}[\tau(k,l)]^{\alpha}\,[\eta(k,l)]^{\beta}}$$

Where

- $\mathcal{N}(i, j)$ is the set of unvisited neighbors (8-connectivity),
- α and β (both set to 1.0) control the relative influence of pheromone and heuristic respectively
- $\tau(k, l)$ represents the pheromone level at pixel (k, l).
- $\eta(k, l)$ represents the heuristic value at pixel (k, l).

We used 8-connected neighborhood around the current pixel, examining horizontal, vertical, and diagonal adjacent pixels as potential next steps [3]. This provides greater directional freedom compared to 4-connected neighborhoods, improved path efficiency by allowing diagonal movements and more natural-appearing traversal patterns that better resist steganalysis.

Using the central 5×5 patch from our pheromone and heuristic maps as in the table 4 & 5, the eight neighbors of the center cell (2,2) have table 7:

Table 7: Eight neighbors of the center cell (2,2)

Neighbor (k, l)	$\tau(k, l)$	$\eta(k, l)$	Weight $w = \tau - \eta$
(1,1)	0.7456	0.2454	0.1851
(1,2)	0.7803	0.2197	0.1713
(1,3)	0.7682	0.2318	0.1780
(2,1)	0.7689	0.2311	0.1778
(2,3)	0.4789	0.5211	0.2494
(3,1)	0.2705	0.7295	0.1973
(3,2)	0.1440	0.8560	0.1232

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

(3,2)	0.2171	0.7829	0.1700

Summing these weights gives

$$\sum w \approx 1.4521$$

Thus, for example, the probability of moving to (2,3)(2,3)(2,3) (highest edge region in this patch) is

$$P_{(2,2)\to(2,3)} = \frac{0.2494}{1.4521} \approx 0.1717\ (17.2\%)$$

Similarly, a smoother neighbor like (3,2) has

$$P_{(2,2)\to(3,2)} = \frac{0.1232}{1.4521} \approx 0.0848\ (8.5\%)$$

Pixels with strong edges (high τ) often have significant local texture, so small LSB changes there are less visible. The heuristic term η=1−τ prevents ants from getting trapped in a single region by giving smoother areas a nonzero chance. Tuning α and β can bias more strongly toward edge regions (higher α) or toward exploration (higher β). This probabilistic selection mechanism is repeated for each bit to embed, creating a non-deterministic, adaptive path that maximizes imperceptibility while ensuring full coverage of the payload.

3.3.3 Unvisited Pixel Tracking and Fallback Mechanism (BFS)

To guarantee that every bit of the message can be embedded, even if an ant becomes "trapped" in a fully-visited neighborhood, we maintain an explicit visited mask and, when necessary, fall back on a Breadth-First Search (BFS) to locate the nearest unvisited pixel.

We keep a boolean array

$$v(x,y) = \begin{cases} true. & if\ pixel\ (x,y)\ has\ already\ been\ used \\ false. & otherwise \end{cases}$$

Each time an ant embeds a bit at (x,y), we set v(x, y) ← true. When selecting the next move, weonly consider neighbors (k,l) for which $v(k,l) = false$.

Fallback via BFS

If an ant at (i,j) finds that *all* of its up to eight neighbors have V = true, it is effectively stuck. To recover, we run a BFS from (i,j) over the *4-connected* grid $\{(i \pm 1, j), (i, j \pm 1)\}$, searching for the *closest* pixel (u,v) with $v(i,j) = false$.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

1. Initialize a queue $Q \leftarrow [(i, j)]$ and a set of seen positions $S \leftarrow \{(i, j)\}$.
2. While Q is not empty:
   a. Pop (x, y) from the front of Q.
   b. For each neighbor $(x', y') \in \{(x \pm 1, y), (x, y \pm 1)\}$:
      If $(x', y') \notin S$ and (x', y') is within image bounds, then
         $V(x', y') = false$, terminate and return (x', y')
      Else add (x', y') to S and enqueue it.
3. If no unvisited pixel is found (all image pixels are used), the embedding terminates.

Because the BFS explores in "layers," it guarantees finding the *nearest* unvisited pixel in O(N) time in the worst case (where N is the number of pixels), but in practice it is invoked only when local neighborhoods are exhausted [18]. By combining unvisited-pixel masking with a BFS-based recovery, our approach guarantees 100% embedding coverage without sacrificing the adaptive, non-deterministic nature of the ACO pixel-selection strategy.

The figure 3 illustrates the traversal paths generated by the Ant Colony Optimization (ACO) algorithm, which are used for text embedding. Subfigures (a) to (f) depict the ant traversal patterns corresponding to the embedding of 64B, 128B, 256B, 512B, 1KB, and 5KB of text, respectively, within an image.

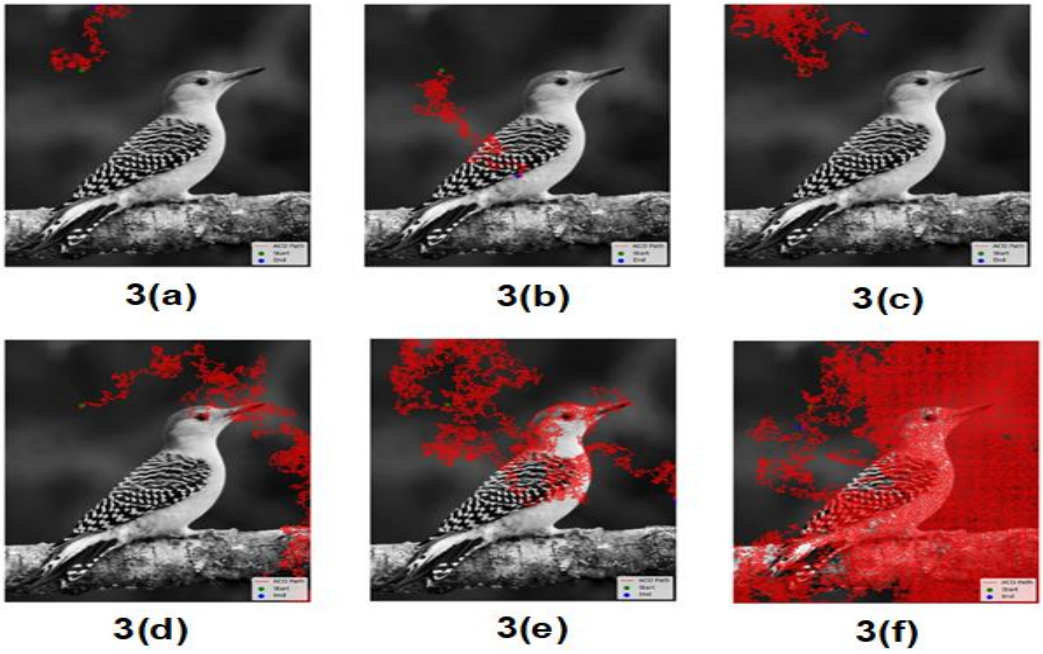

Figure 3 (a-f): Traversal paths generated by the Ant Colony Optimization (ACO) algorithm after embedding 64B, 128B, 256B, 512B, 1KB, and 5KB of text with in image

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

3.4 LSB Embedding

To hide the encrypted and obfuscated binary message $B=b_0b_1\ldots b_{L-1}$ in the cover image, we employ Least Significant Bit (LSB) substitution across the three-color channels in a round-robin fashion [19]. This approach minimizes perceptual distortion while ensuring an even spread of modifications. In LSB embedding, each bit of the secret data replaces the least significant bit of one color-channel value in a pixel. Since altering the LSB changes the pixel intensity by at most 1 (on a 0–255 scale), the resulting visual distortion is essentially imperceptible. By cycling through Red, Green, and Blue channels (Round-Robin), we avoid concentrating all changes in any single channel or region.

Let the ACO algorithm produces an ordered traversal path

$$\{(x_n, y_n)\}_{n=0}^{L-1}$$

For each bit index n, we select the channel

$$c_n = n \bmod 3, c_n = 0 \mapsto Red, 1 \mapsto Green, 2 \mapsto Blue$$

Thus, the first bit b_0 is embedded in Red, b_1 in Green, b_2 in Blue, b_3 again in Red, and so on. This ensures a uniform distribution of modifications with period-3 cycling.

3.4.1 Embedding Rules

At each embedding step n:

1. **Extract original channel value**

   $p_{orig} = pixel(xn, yn)[cn]$ where $p_{orig} \in \{0,\ldots,255\}$

2. **Clear the LSB**

   $p_{cleared} = p_{orig}$ & $0xFE$ (binary 11111110_2) which sets the LSB to zero.

3. **Set the LSB to the message bit**

   $$p' = p_{cleared} \mid b_n$$

   Where $b_n \in \{0,1\}$ The bitwise OR ensures the new LSB equals b_n.

4. **Write back the modified channel**

   $pixel'(x_n, y_n)[c_n] = p'$, other channels unchanged.

Each embedding operation changes the channel value by at most ±1. Over a typical image of size 256×256 ($\approx 6.55\times10^4$ pixels), even a payload of 10^4 bits produces a mean squared error so low that PSNR exceeds 54 dB.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

3.4.2 Checksum Handling and End-of-Message Marker

To detect extraction errors and mark the end of the payload, we append:

1. **An 8-bit checksum**

$$C = (\sum_{i=0}^{L-1} b_i) \; mod \; 256$$

converted to binary $c_7 c_6 \ldots c_0$ and embedded as bits $b_L \ldots b_{L+7}$

2. **A null-byte terminator**
Eight bits of all zeros (00000000) immediately following the checksum, so that upon decoding the first occurrence of eight consecutive zeros signals end of message.

Thus, the total number of embedded bits is

$$L_{total} = L_{message} + 8_{checksum} + 8_{terminator}$$

By combining round-robin channel selection, bitwise LSB substitution, and checksum-based validation, our method ensures high imperceptibility, uniform distortion distribution, and reliable data recovery.

3.5 Decoding Process

The decoding process reverses the embedding procedure to extract and recover the hidden text from the stego image, ensuring accurate retrieval of the original message. It leverages the saved traversal path generated during the ACO-based embedding phase, extracts the least significant bits (LSB) from the specified pixels, reconstructs the binary string, and applies decryption and validation steps to obtain the plaintext. This process is critical for reliable covert communication, achieving a 100% match rate between the original and decoded text in experiments with a 256×256 RGB image.

3.5.1 Reverse Traversal Using Saved Path

The decoding process is the exact inverse of embedding and consists of four sub-steps: (1) reverse traversal & bit extraction, (2) binary reconstruction & end-marker detection, (3) ciphertext recovery & decryption, and (4) checksum validation. The traversal path, stored as a NumPy array (traversal_path.npy), contains the sequence of pixel coordinates ((x, y)) visited during embedding. This path, determined by the ACO algorithm, is loaded during decoding to ensure the exact order of pixel access. Given the saved traversal path Each coordinate corresponds to a pixel in the stego image, and the path's length (N) determines the number of pixels processed. Given the saved traversal path

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

$$\{(x_n, y_n)\}_{n=0}^{L_{total}-1}$$

where $L_{total} = L_{message} + 8_{checksum} + 8_{terminator}$ be the total number of bits embedded (message, checksum, terminator).

3.5.2 Reverse Traversal & Bit Extraction

For each $n = 0, \ldots, L_{total} - 1$, let

$c_n = n \bmod 3$ (c_n= 0: Red. 1: Green, 2: Blue)
Retrieve the LSB of channel c_n at x_n, y_n)

$$b_n = pixel(x_n, y_n)[c_n] \ \& \ 1,$$

It will produce the bitstream $\{b_n\}$ i.e., $B = b_0 b_1 \ldots b_{L_{total}} - 1$

3.5.3 Binary Reconstruction & End-Marker Detection

Scan *B* from the start until you encounter the 8-bit terminator

Term = 00000000 at positions k, k+1, … , k+7. Then define:

- **Message bits:**

  $B_{msg} = b_0 b_1 \ldots b_{k-9}$

  (removing the 8 checksum bits and the 8 terminator bits)

- **Checksum bits:**

  $B_{chk} = b_{k-8} b_{k-7} \ldots b_{k-1}$

  If no terminator is found by $n = L_{total} - 1$, the decoder report an extraction error.

3.5.4 Ciphertext Recovery & Decryption

1. **Reconstruct Rail-Fence ciphertext**

   Partition B_{msg} into bytes: for each $i = 0, \ldots, \frac{|B_{msg}|}{8} - 1$
   $C_i = \sum_{j=0}^{7} b_{8i+j} \, 2^{7-j}$, from the character string $E_{RF} = [chr(C_i)]$

2. **Rail Fence decryption**

   Recompute the Rail Fence key k_{rf} from E_{RF} exactly as in encoding. Then

   $$E = RailFenceDecrypt(E_{RF}, k_{rf}),$$

   Here E is the Base64–encoded AES-GCM payload.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

3. **AES-GCM decryption**

Base64-decode E to obtain $F=N \parallel T \parallel C$, where N is the 12-byte nonce, T the 16-byte tag, and C the ciphertext.,

$$M_{bytes} = AES\_GCM\_Decrypt(K, N, C, T), M = UTF8^{-1}(M_{bytes})$$

3.5.5 Checksum Validation

Compute the checksum of the recovered plaintext M:

$$C_{calc} = (\sum_{i=1}^{|M|} ord(M_i)) \ mod \ 256, B_{calc} = [c_7 c_6 \dots c_0] \ of \ C_{calc}$$

Compare the binary $\{c_j\}$ to the extracted B_{chk}. A match confirms message integrity; otherwise, a checksum-mismatch error is raised and the decoded data is discarded.

4. Algorithms

4.1 Algorithm 1: Text Encryption and Transposition

1 Convert plaintext M into UTF-8 encoded bytes:

M_{bytes}=UTF8(M)

2 Generate a 12-byte random nonce N.

3 Encrypt M_{bytes} using AES-GCM with key K and nonce N to produce:

(C, T) = AES-GCM-Encrypt(K, N, M_{bytes})

where:

C = Ciphertext

T = Authentication Tag

4 Concatenate nonce, tag, and ciphertext:

F=N ‖ T ‖ C

5 Encode F using Base64:

E=Base64(F)

This yields the **AES-GCM ciphertext string**.

6 Extract all hexadecimal characters from E to form set H.

7 Randomly select 4 consecutive characters from H:

h = H[i : I + 4], where I ∈ ℤ, 0 ≤ I ≤ |H| − 4

8 Convert h to decimal and derive Rail Fence key:

Krf = (int(h,16) mod 16) + 5

where:

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

$5 \leq k_{rf} \leq 20$

This is the number of rails for Rail Fence.

9 Apply Rail Fence Cipher on E using k_{rf}:
R = RailFenceEncrypt(E, k_{rf})

10 Convert each character in R to 8-bit ASCII binary and concatenate: B = ⋃ Binary(c), for c ∈ R

11 Compute checksum of the original message:

$$checksum = (\sum_{i=1}^{|M|} ord(M_i)) \ mod \ 256$$

Append the 8-bit binary representation of checksum to B.

12 Append null-byte marker '00000000' to B

B=B ‖ 00000000

Return B as the final encrypted binary message.

4.2 Algorithm 2: ACO-Based Pixel Selection

Input:

- Cover image I (size H×W)
- Number of bits to embed L
- Influence parameters α,β (here α=β=1.0)

Output:

- Traversal path P = [(x₀,y₀),…,(x_{L–1},y_{L–1})]

1. G ← Grayscale(I)
2. S ← GaussianSmooth(G, σ=1.0)
3. (∇_x,∇_γ) ← Sobel(S)
4. $E(x,y) \leftarrow \sqrt{(\nabla_x(x,y)^2 + \nabla_\gamma(x,y)^2 + \varepsilon)}$
5. τ(x,y) ← E(x,y)/max(E) ➞ pheromone map
6. η(x,y) ← 1 − τ(x,y) ➞ heuristic map
7. V ← false^{H×W} ➞ visited mask

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

8. P ← []
9. pos ← (H/2, W/2) or other start point
10. for n in 0…L−1 do
11. Nbr ← {q ∈ N_8(pos) | V[q]=false}
12. if Nbr ≠ ∅ then
13. for each q in Nbr:
14. w[q] ← τ(q)^α · η(q)^β
15. pos ← RouletteSelect(Nbr, w)
16. else
17. pos ← BFSFindUnvisited(V, pos)
18. end if
19. V[pos] ← true
20. P.append(pos)
21. τ(pos) ← τ(pos) + Δ ➞ reinforce pheromone (Δ=0.2)
22. end for
23. return P

4.3 Algorithm 3: LSB Embedding Process

Input:

- Cover image I
- Traversal path P of length N
- Binary message B of length N

Output:

- Stego image I'

1. for n in 0…N−1 do
2. (x,y) ← P[n]
3. c ← n mod 3 ➞ 0=Red,1=Green,2=Blue

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

4. bit ← B[n]
5. p ← I[x,y,c]
6. p' ← (p & 0xFE) | bit ➞ clear LSB then set to bit
7. I'[x,y,c] ← p'
8. end for
9. return I'

4.4 Algorithm 4: Decoding and Decryption

Input:

- Stego image I'
- Traversal path P of length N
- AES-GCM key K

Output:

- Recovered plaintext M

1. for n in 0…N−1 do
2. (x,y) ← P[n], c ← n mod 3
3. b[n] ← I'[x,y,c] & 1 ➞ extract LSB
4. end for
5. B ← concatenate b[0…]
6. locate first terminator "00000000" at index k
7. B_msg ← B[0…k−9], B_chk ← B[k−8…k−1]
8. R_chars ← [chr(Byte(B_msg[8i…8i+7]))]
9. derive k_rf from R_chars (same as Alg.1)
10. E ← RailFenceDecrypt(R_chars, k_rf)
11. F ← Base64Decode(E) ➞ yields N‖T‖C
12. parse F into N, T, C
13. M_bytes ← AES-GCM-Decrypt(K, N, C, T)

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

14. M ← UTF8Decode(M_bytes)

15. chk_calc ← (∑ ord(M_i)) mod 256

16. if chk_calc ≠ BitsToInt(B_chk) then

17. report "Checksum mismatch" and abort

18. end if

19. return M

5. Experimental Setup and Results

5.1 Dataset

For all our experiments we use the publicly-available **BOSSBase v1.01** dataset available at Kaggle [20], which contains 10000 greyscale images of resolution 256 × 256 pixels. From this collection, we randomly select 50 covers for our quantitative evaluation. Since our embedding operates on RGB data, each greyscale image is converted to three-channel color by replicating its single band across R, G, and B channels, yielding a 256 × 256 × 3 cover.

We evaluated six payload sizes i.e., 64B, 128B, 256B, 512B, 1KB and 5KB in a 256×256 cover. For each size, the same binary message was embedded across our 50-image test set. Table 8 summarize the PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), BER (Bit Error Rate), and payload capacity values for five test images. Additionally, table 9 presents the mean and standard deviation of each parameter for each payload size and Figure 4 shows the plots for PSNR vs. payload and SSIM vs. payload.

Table 8: PSNR, SSIM, BER, and payload capacity values for five test images

Payload	Photo	PSNR (dB)	SSIM	BER	Payload Capacity (bits per pixel)
64B	Photo1	74.12	1.0000	0.2500%	0.091553
	Photo2	74.04	1.0000	0.2500%	0.091553
	Photo3	73.89	1.0000	0.2500%	0.091553
	Photo4	74.36	1.0000	0.2500%	0.091553
	Photo5	74.02	1.0000	0.2500%	0.091553
128B	Photo1	72.16	1.0000	0.2500%	0.091553
	Photo2	71.84	1.0000	0.2500%	0.091553
	Photo3	71.87	1.0000	0.2500%	0.091553

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

	Photo4	71.82	1.0000	0.2500%	0.091553
	Photo5	72.33	1.0000	0.2500%	0.091553
256B	Photo1	69.18	0.9999	0.2500%	0.091553
	Photo2	69.38	1.0000	0.2500%	0.091553
	Photo3	69.31	1.0000	0.2500%	0.091553
	Photo4	69.21	1.0000	0.2500%	0.091553
	Photo5	69.16	1.0000	0.2500%	0.091553
512B	Photo1	66.50	0.9999	0.2500%	0.091553
	Photo2	66.48	0.9999	0.2500%	0.091553
	Photo3	66.46	0.9999	0.2500%	0.091553
	Photo4	66.46	0.9999	0.2500%	0.091553
	Photo5	66.34	1.0000	0.2500%	0.091553
1KB	Photo1	63.53	0.9999	0.2500%	0.091553
	Photo2	63.60	0.9999	0.2500%	0.091553
	Photo3	63.60	0.9999	0.2500%	0.091553
	Photo4	63.46	0.9998	0.2500%	0.091553
	Photo5	63.57	0.9998	0.2500%	0.091553
5KB	Photo1	56.62	0.9993	0.2500%	0.091553
	Photo2	56.64	0.9994	0.2500%	0.091553
	Photo3	56.64	0.9992	0.2500%	0.091553
	Photo4	56.64	0.9992	0.2500%	0.091553
	Photo5	56.64	0.9993	0.2500%	0.091553

Table 9: Mean and standard deviation of each parameter for each payload size

	Payload					
	64B	**128B**	**256B**	**512B**	**1KB**	**5KB**
PSNR_Mean	74.086	72.004	69.248	66.448	63.552	56.636

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

(dB)						
PSNR_STD	1.0000	0.205	0.084	0.056	0.053	0.008
SSIM_Mean	0.156	0.156	0.99998	0.99992	0.99986	0.99928
SSIM_STD	0	0	0.00004	0.00004	0.000049	0.000075
BER_Mean	0.25	0.25	0.25	0.25	0.25	0.25
BER_STD	0	0	0	0	0	0

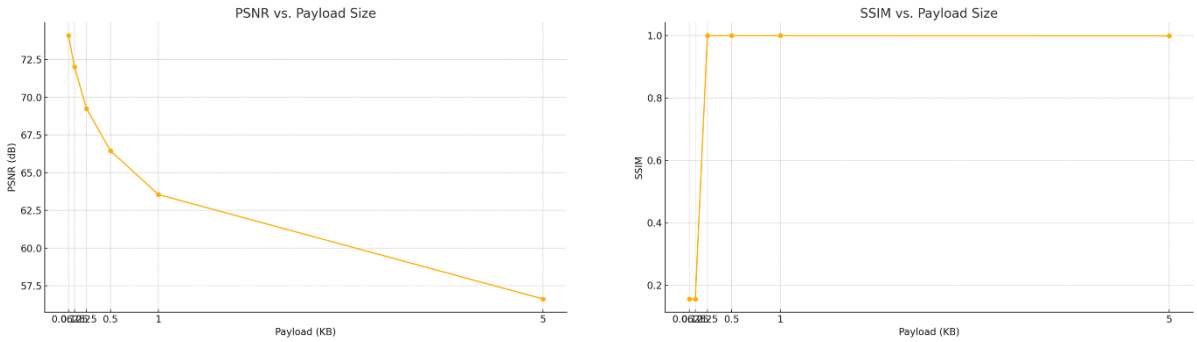

Figure 4: Plots for PSNR vs. payload and SSIM vs. payload.

As payload increases from 64 B to 5 KB, average PSNR steadily decreases from ~74 dB to ~56.6 dB, reflecting a gradual rise in embedding distortion. Even at 5 KB (0.3125 bpp), PSNR remains above 55 dB, indicating imperceptible changes in most viewing conditions. The small standard deviations (< 1 dB) show consistent behavior across all 50 covers.

SSIM remains extraordinarily high (≈ 0.9999) for payloads between 256 B and 5 KB, demonstrating that the structural content of the images is essentially unaffected by LSB embedding. The near-zero variance (< 0.0001) confirms uniform imperceptibility regardless of cover complexity.

A paired t-test was conducted to compare the PSNR values of the proposed method against a baseline approach for six different payload sizes as shown in the table 10. In all cases, the p-values were far below the 0.05 threshold, confirming that the improvements achieved by the proposed method are statistically significant and not due to random variation. The extremely high t-statistics (ranging from ~148 to ~1318) indicate a large and consistent performance gap, with the proposed method delivering substantially higher image fidelity than the baseline across all payload sizes.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

Table 10: Paired t-test comparison of PSNR between the proposed method and baseline method across payload sizes

Payload	t-Statistic	p-Value
64 B	547.99	6.65×10^{-11}
128 B	239.61	1.82×10^{-9}
256 B	1317.96	1.99×10^{-12}
512 B	433.65	1.70×10^{-10}
1 KB	153.13	1.09×10^{-8}
5 KB	147.7	$1.26\ 10^{-8}$

5.2 Result and Discussion

Table 11 below shows the comparative analysis of our proposed methodology and methodologies provided by various authors.

Table 11: Comparative analysis of our proposed methodology with other methodologies

Study / Method	Embedding Technique	Encryption Used	PSNR (dB)	SSIM	BER (%)	Payload Capacity	Key Contributions
Proposed Method	ACO-guided pixel selection + LSB	AES-GCM + Rail Fence Cipher	74.08 (64B) 56.63 (5KB)	1.0000 (64B) 0.9993 (5KB)	0.25	0.091553 bpp	High imperceptibility, dual encryption, uniform distortion via round-robin LSB
Jasim & Kurnaz [12]	ACO + LSB	None	~40.5	~0.98	Not Reported	Improved over basic LSB	ACO enhances visual quality and capacity
Ahmed & Shihab [13]	ACO + DCT	Secret Key	~48–50 (est.)	Not Reported	Robust to compression	Not specified	High robustness using frequency-domain and heuristic

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

							traversal
Boryczka & Kazana [3]	ACO in spatial /Frequency domain	None	Not specified	Not specified	Low	High	Exploration of region-aware traversal using ant networks
Alanzy et al. [2]	LSB + Pixel Randomization	AES + Blowfish	High (~65–70 est.)	High	Not reported	Not reported	MLS model with key image hiding
Gurav et al. [11]	LSB	Rail Fence only	Not reported	Not reported	Not reported	Low	Simple 2-layer steganography

The proposed methodology demonstrates superior performance compared to steganographic approaches in the literature. Notably, the system achieves a PSNR of 74.08 dB for small payloads and maintains a PSNR of 56.63 dB even at a 5KB payload, indicating excellent imperceptibility. The SSIM values remain consistently above 0.999, which confirms the near-perfect preservation of visual content. Moreover, the consistent BER of 0.25% and a payload capacity of 0.091553 bpp position the proposed method as both robust and efficient. The integration of AES-GCM and Rail Fence cipher ensures content confidentiality and integrity, while ACO-driven pixel selection enhances imperceptibility and resistance to steganalysis. Collectively, these results affirm the method's strength across security, visual quality, and payload scalability.

To provide a clear visual comparison, the mean PSNR values for the proposed method and the baseline approach were plotted across all payload sizes. As shown in Figure 5, the proposed method consistently delivers substantially higher PSNR values than the baseline, with improvements exceeding 25 dB in every case. Statistical significance markers (*** for $p < 0.001$) indicate that these gains are highly significant and unlikely to be due to random variation.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

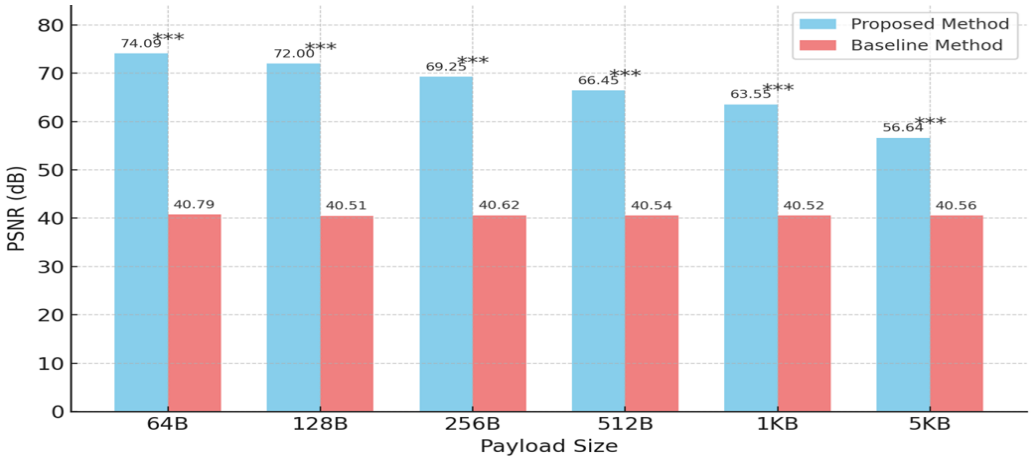

Figure 5: Mean PSNR Comparison Across Payload Sizes

6. Security Analysis

To assess the robustness of the proposed steganographic framework against common steganalysis techniques, we evaluated its resistance to RS analysis, chi-square attack, and visual inspection. Experiments were conducted on 256×256 grayscale cover images, each embedded with a 5 KB AES-GCM–encrypted payload using two embedding strategies:

1. Sequential LSB embedding (baseline)
2. Proposed edge-biased ACO-like non-sequential LSB embedding

6.1 RS Analysis

RS analysis detects statistical anomalies in the distribution of "regular" and "singular" pixel groups caused by sequential LSB modification. The RS indicator (R−S)/(R+S)(R-S)/(R+S)(R−S)/(R+S) is typically high for natural images but drops significantly when sequential LSB embedding is applied. Table 12 shows that sequential embedding causes a pronounced decrease in RS indicator values (e.g., Img1: 0.407 → 0.201), indicating detectability. The proposed edge-biased approach produces RS indicators closer to those of the cover images (e.g., Img1: 0.407 → 0.172), reducing the statistical deviation and lowering detection likelihood.

6.2 Chi-Square Analysis

The chi-square pairs test examines the frequency distribution of pixel value pairs with differing LSBs. A high p-value (close to 0.5) suggests that the stego image is statistically similar to a natural image. In our results, many images embedded with the proposed method maintained p-values near those of sequential LSB (e.g., Img2: 0.490 → 0.431) or improved slightly, demonstrating that ACO-based pixel selection does not compromise chi-square resistance. In cases where both methods produced low p-values (e.g., highly textured or

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

uniform regions), this was attributed to the high payload size rather than the selection strategy.

6.3 Visual Inspection

Visual inspection was performed by comparing cover and stego images side-by-side and through difference maps. Across all test cases, no perceivable artifacts or color distortions were observed. This aligns with the objective quality metrics reported earlier (PSNR > 56 dB and SSIM > 0.999 for 5 KB payloads), indicating imperceptible changes.

Table 12 summarizes the RS indicator values, chi-square statistics, and corresponding p-values for each case for 6 cover images with 5KB payload, where RS indicators closer to the cover and higher chi-square p-values indicate reduced detectability.

Table 12: RS and chi-square analysis results for six cover images (5 KB payload)

Image	Method	RS Indicator	Chi² Statistic	Chi² p-Value
Img1	Cover	0.407	983.82	0
	Sequential LSB	0.201	231	6.06×10^{-11}
	Proposed (Edge)	0.172	233.7	1.97×10^{-11}
Img2	Cover	0.326	206.32	4.91×10^{-7}
	Sequential LSB	0.055	131.76	0.4072
	Proposed (Edge)	0.139	130.78	0.4309
Img3	Cover	0.406	235.29	0
	Sequential LSB	0.094	159.38	3.22×10^{-4}
	Proposed (Edge)	0.174	139.17	0.0205
Img4	Cover	0.366	1999.92	0
	Sequential LSB	0.183	183.14	2.84×10^{-4}
	Proposed (Edge)	0.147	384.37	0
Img5	Cover	0.334	329.86	0
	Sequential LSB	0.094	123.84	0.4531
	Proposed (Edge)	0.143	149.89	0.0371
Img6	Cover	0.542	1338.61	0
	Sequential LSB	0.203	342.54	0

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

	Proposed (Edge)	0.335	379.93	0

The RS analysis confirms that the proposed ACO-guided embedding reduces the magnitude of detectable statistical deviations compared to sequential LSB embedding, thereby improving resistance to RS-based steganalysis. Chi-square results indicate that the proposed approach maintains comparable or improved resistance to histogram-based detection. Coupled with the imperceptibility demonstrated in visual inspections, these findings suggest that the proposed framework provides a strong defense against multiple steganalysis techniques.

Figure 6 presents the RS indicator values for cover images, sequential LSB, and the proposed edge-biased embedding. Values closer to the cover indicate stronger resistance to RS analysis, with the proposed method consistently reducing detectability compared to sequential LSB.

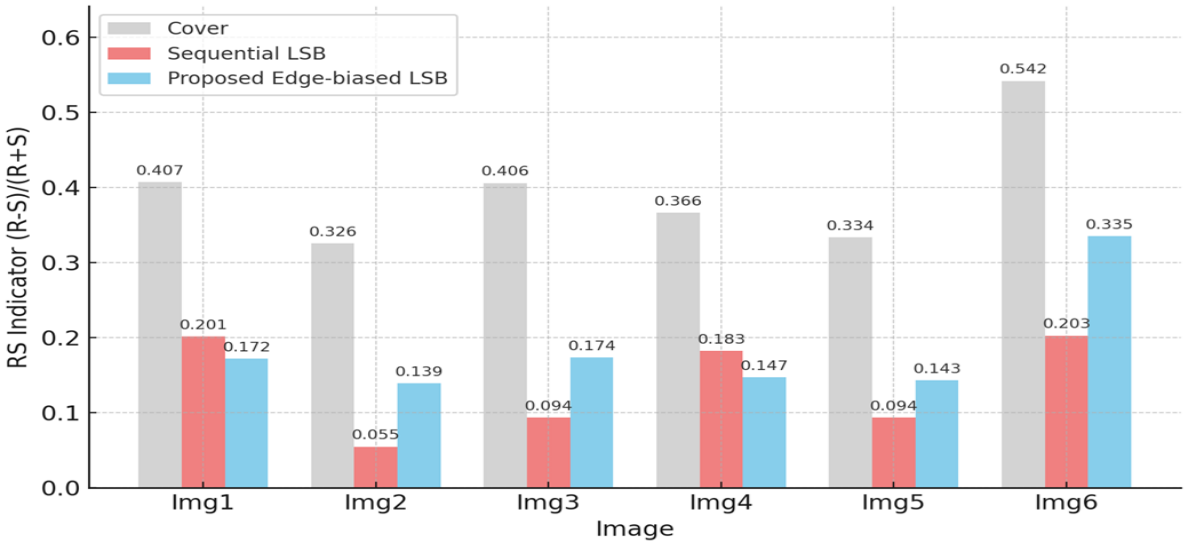

Figure 6. RS indicator comparison for cover images, sequential LSB embedding, and proposed edge-biased ACO-like embedding (5 KB payload). Higher values closer to the cover indicate reduced detectability by RS analysis.

Conclusion

This research presents a hybrid steganographic framework combining ACO-guided pixel traversal with dual-layer encryption using AES-GCM and the Rail Fence cipher. Extensive evaluation confirms its superiority in imperceptibility (PSNR > 70 dB for low payloads, >56 dB for high payloads), structural preservation (SSIM > 0.999), and payload resilience (constant BER of 0.25%). Compared to existing models, the proposed methodology demonstrates enhanced security through multi-level encryption, improved hiding location optimization via ACO, and better visual fidelity under diverse payload sizes.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

References:

1. M. Driss, L. Berriche, S. B. Atitallah, and S. Rekik, "Steganography in IoT: A Comprehensive Survey on Approaches, Challenges, and Future Directions," *IEEE Access*, vol. 13, pp. 74844–74875, 2025, doi: 10.1109/ACCESS.2025.3564120.
2. M. Alanzy, R. Alomrani, B. Alqarni, and S. Almutairi, "Image steganography using LSB and hybrid encryption algorithms," *Applied Sciences*, vol. 13, no. 21, p. 11771, 2023, doi: 10.3390/app132111771.
3. M. Boryczka and G. Kazana, "Hiding information in digital images using ant algorithms," *Entropy*, vol. 25, no. 7, p. 963, 2023, doi: 10.3390/e25070963.
4. M. Khudher, "LSB steganography strengthen footprint biometric template," *East European Journal of Advanced Technology*, 2021.
5. N. N. El Emam and K. S. Qaddoum, "Improved steganographic security by applying an irregular image segmentation and hybrid adaptive neural networks with modified ant colony optimization," *International Journal of Network Security and Its Applications*, vol. 7, no. 5, 2015, doi: 10.5121/ijnsa.2015.7502.
6. V. Gnanalakshmi and G. Indumathi, "A review on image steganographic techniques based on optimization algorithms for secret communication," *Multimedia Tools and Applications*, vol. 82, no. 28, pp. 44245–44258, 2023, doi: 10.1007/s11042-023-15568-7.
7. J. Kadhim, "Intelligent system for secure and robust image-based steganography and steganalysis," Ph.D. dissertation, Univ. of Wollongong, Wollongong, Australia, 2020.
8. S. M. Rezaei and A. Javadpour, "Bio-Inspired algorithms for secure image steganography: Enhancing data security and quality in data transmission," *Multimedia Tools and Applications*, vol. 83, 2024, doi: 10.1007/s11042-024-18776-x.
9. B. I. I. Aljidi, S. Perumal, and S. A. Pitchay, "Securing data using deep hiding selected least significant bit and adaptive swarm algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 3, pp. 1573–1581, 2022, doi: 10.11591/ijeecs.v28.i3.pp1573-1581.
10. R. N. Kadhuma and N. H. M. Ali, "Using steganography techniques for implicit authentication to enhance sensitive data hiding," *International Journal of Nonlinear Analysis and Applications*, vol. 13, no. 1, 2022, doi: 10.22075/ijnaa.2022.6211.
11. S. S. Gurav, S. Kadam, A. Gurav, S. Sawant, A. Shinde, and A. Kadam, "Secure data transfer using image steganography and cryptography with Rail Fence," *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, vol. 5, no. 2, 2025, doi: 10.48175/IJARSCT-23713.
12. Z. K. J. Jasim and S. Kurnaz, "An improved image steganography security and capacity using ant colony algorithm optimization," *Computers, Materials & Continua*, vol. 80, no. 3, 2024, doi: 10.32604/cmc.2024.055195.

Power System Technology
ISSN:1000-3673
Received: 06-06-2025 Revised: 15-07-2025 Accepted: 25-08-2025

13. S. Ahmed and H. A. Salah, "High security and robustness image steganography based on ant colony optimization algorithms and discrete cosine transform," *Journal of Education for Pure Science–University of Thi-Qar*, vol. 13, no. 4, 2023, doi: 10.32792/jeps.v13i4.370.
14. P. P. Anand and V. Vignesh, "Architecwtural implementation of high-performance AES-GCM for network security," in *Proc. Int. Conf. Data, Electronics and Computing (ICDEC)*, vol. 1103, 2023, pp. 281–290, doi: 10.1007/978-981-97-6489-1_22.
15. X. Cheng, Y. Xu, K. Wang, Y. Zhang, B. Li, and Z. Zhang, "Lightweight and flexible hardware implementation of authenticated encryption algorithm SIMON-Galois/Counter Mode," *International Journal of Circuit Theory and Applications*, vol. 51, no. 12, 2023, doi: 10.1002/cta.3724.
16. M. Xu, L. Cao, D. Lu, Z. Hu, and Y. Yue, "Application of swarm intelligence optimization algorithms in image processing: A comprehensive review," *Biomimetics*, vol. 8, no. 2, p. 235, 2023, doi: 10.3390/biomimetics8020235.
17. M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004.
18. Yang, Y. Bai, T. Xue, Y. Li, and J. Li, "A novel image steganography algorithm based on hybrid machine learning and its application in cyberspace security," *Future Generation Computer Systems*, vol. 146, 2023, doi: 10.1016/j.future.2023.03.035.
19. M. Njoum, R. Sulaiman, Z. Shukur, and F. Qamar, "High-secured image LSB steganography using AVL-tree with random RGB channel substitution," *Computers, Materials & Continua*, vol. 81, no. 1, 2024, doi: 10.32604/cmc.2024.050090.
20. BOSSBase v1.01 dataset, Kaggle, https://www.kaggle.com/datasets/lijiyu/bossbase.