Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 10 No. 5 (2026) 1423 - 1434 e-ISSN: 2580-0760

Deterministic Texture-Adaptive Steganography for IoT Edge: Achieving Optimal Rate-Distortion With O(N) Complexity

Saepudin Nirwan[1], Naufal Fachrudin Nirwan[2], Agus Perdana Windarto[3]
[1]Informatics, School of Information Technology, University of Logistics and Business International, Bandung, Indonesia,
[2]Data Science Study Program, Faculty of Informatics, Telkom University, Bandung, Indonesia,
[3]Master's Program, Informatics Study Program, STIKOM Tunas Bangsa, Pematangsiantar, North Sumatra, Indonesia

*nirwansaepudin@ulbi.ac.id

Abstract

The rapid expansion of Internet of Things (IoT) edge systems has increased the demand for image steganography methods that provide strong security, high visual quality, and low computational overhead under resource-constrained environments. Existing adaptive steganography techniques, such as HUGO, WOW, and S-UNIWARD, achieve high embedding performance but generally rely on computationally intensive optimization procedures that limit their applicability to real-time IoT edge devices. This study addresses this limitation by proposing a Deterministic Texture-Adaptive Steganography framework based on a Texture Complexity Analyzer (TCA) integrated with Syndrome-Trellis Codes (STC). The TCA module generates a deterministic pixel-wise distortion cost map using second-order directional filtering, while the STC encoder performs minimum-distortion message embedding through trellis-based optimization with linear computational complexity. The proposed framework was evaluated on 40,000 images from the LIRMM CroppedBOSSbase dataset using five-fold stratified cross-validation. Experimental results demonstrate a mean PSNR of 59.63 dB, an SSIM of 0.9996, and a detection error of 0.50 against SR-Net steganalysis, indicating high visual imperceptibility and strong statistical security. The framework also achieved an average execution time of 80 ms per 256 × 256 image with an estimated energy consumption of 0.28 J, substantially reducing computational cost compared with iterative baseline methods. These findings demonstrate that the proposed deterministic framework provides an efficient balance between security, imperceptibility, and computational efficiency, making it well suited for practical deployment in resource-constrained IoT edge environments where predictable execution time and energy efficiency are essential.

Keywords: deterministic steganography; IoT edge computing; image steganography; Syndrome-Trellis codes; texture-adaptive embedding

How to Cite: S. Nirwan, N. F. . Nirwan, and A. P. . Windarto, "Deterministic Texture-Adaptive Steganography for IoT Edge: Achieving Optimal Rate-Distortion With O(N) Complexity", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 10, no. 5, pp. 1423–1434, Oct. 2026, doi: 10.29207/resti.v10i5.7582.
Permalink/DOI: https://doi.org/10.29207/resti.v10i5.7582

Received: February 13, 2026
Accepted: September 5, 2026
Available Online: October 13, 2026

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

Over the past decade, the proliferation of Internet of Things (IoT) technology has fundamentally reshaped the architecture of modern computing, accelerating a transition from centralized, server-based processing toward distributed edge-based frameworks [1-5]. Heterogeneous IoT devices, including embedded industrial sensors, medical telemetry nodes, smart surveillance cameras, and autonomous vehicle platforms, now operate at the network edge to perform real-time data acquisition, local preprocessing, and secure transmission. This paradigm reduces round-trip latency, alleviates core network bandwidth pressure, and extends system availability in environments where continuous cloud connectivity cannot be guaranteed [6-8].

The distinction between IoT edge environments and conventional computing platforms lies in fundamental resource asymmetry rather than operational scale alone. Desktop and server-grade systems operate with multi-gigahertz processors, gigabytes of addressable memory, and effectively unconstrained power supplies, whereas IoT edge devices are bound by microcontrollers clocked at tens to hundreds of megahertz, kilobyte-scale memory, and finite battery or harvested energy [9-13]. A security algorithm requiring a few hundred

milliseconds on a desktop may consume several joules and seconds on a battery-powered sensor node, making it incompatible with real-time throughput and energy budgets [11-13]. This resource asymmetry is especially consequential for visual content, which constitutes an especially sensitive and high-volume data category in IoT networks [14-16]. Applications such as telemedicine, industrial inspection, perimeter surveillance, and facial recognition depend on continuous image transmission over channels that are often public or shared across untrusted infrastructure, exposing visual payloads to eavesdropping, interception, and forensic analysis [17-19]. The convergence of sensitive content and inadequately protected channels therefore creates a security exposure that must be addressed well below the resource ceiling of edge hardware.

The conventional response to data security in networked systems relies on cryptographic protocols to protect message confidentiality. Cryptography is, in principle, effective at ensuring that intercepted content cannot be read without the appropriate decryption key. However, it carries a structural limitation of particular consequence in adversarial IoT deployments, namely its complete inability to conceal the existence of communication itself. Encrypted traffic is visually and statistically distinguishable from plaintext traffic, making it a recognizable target on monitored networks. In contexts where transmissions are subject to traffic analysis, deep packet inspection, or metadata correlation, all of which are techniques routinely employed in state-level network surveillance and targeted cyber-attack campaigns, the mere presence of ciphertext is sufficient to attract adversarial attention [20-22]. An attacker unable to break encryption may nonetheless infer communicating parties, transmission frequency, payload volume, and behavioral patterns solely from traffic metadata. In high-sensitivity IoT deployments, including covert sensing systems, medical data relay networks, and industrial monitoring infrastructure, this residual information leakage constitutes a non-trivial security threat that cryptographic methods alone are insufficient to address.

Digital steganography addresses this limitation by embedding secret information within an innocuous carrier medium, most commonly a digital image, in a manner designed to be indistinguishable from unmodified content to both visual inspection and automated statistical analysis [23-27]. Unlike cryptography, steganography does not merely protect the content of a message; it conceals the very fact of communication, rendering it invisible to passive monitoring systems and traffic pattern analysis [25], [28-30]. Image steganography has attracted sustained and growing research interest over the past two decades, driven by the dominance of visual data in modern communication ecosystems and by the rich perceptual redundancy inherent in natural images, which provides exploitable space for covert embedding without apparent degradation [31-34]. These properties make image steganography a strategically important complement to cryptography in environments where both content confidentiality and communicative undetectability are simultaneously required.

A substantial body of research has yielded increasingly sophisticated image steganography algorithms, progressing from rudimentary least-significant-bit (LSB) substitution to advanced methods grounded in distortion minimization theory, probabilistic cost modeling, and deep neural architectures. Methods such as HUGO [35], WOW [36], S-UNIWARD [37], and MiPOD [38] have demonstrated strong performance in terms of embedding security and rate-distortion efficiency under standard evaluation conditions. However, these methods share a characteristic that systematically undermines their suitability for IoT edge deployment: they rely on computationally intensive procedures whose complexity profiles render them impractical on embedded microprocessors operating under strict execution time and energy budgets [9-13]. In particular, HUGO [35] employs high-dimensional feature residuals computed over multi-scale filter banks, in the spirit of rich-model steganalysis features [39], while S-UNIWARD [37] requires repeated wavelet-domain decompositions across three directional subbands; both procedures are inconsistent with the real-time throughput demands of edge hardware. For the majority of IoT platforms, deploying such algorithms within live image transmission pipelines is not merely suboptimal; it is operationally infeasible [11-13].

Despite these theoretical advantages, most texture-aware steganographic methods continue to employ stochastic or probabilistic embedding mechanisms, a characteristic evident in methods such as HUGO [35] and WOW [36]. Such approaches involve complex distortion cost function computation, probability mass assignment over pixel modification states, and non-deterministic sampling, all of which collectively compound computational overhead and introduce execution variability. In IoT edge systems, where reproducibility, deterministic latency, and predictable energy consumption are essential operational requirements, this non-deterministic behavior is not merely inefficient; it is architecturally incompatible with the deployment constraints of real-time embedded systems.

A further challenge that pervades the steganography literature concerns the fundamental rate-distortion trade-off: increasing embedding payload necessarily raises pixel modification density, elevating both perceptible distortion and statistical detectability risk. Achieving an optimal balance between these competing objectives typically requires numerical optimization or global search procedures whose complexity commonly exceeds O(N), where N denotes the total number of image pixels. On IoT edge hardware, even modest superlinear complexity manifests as disproportionate increases in execution time and energy consumption.

To illustrate this concretely: for a battery-powered edge device transmitting images at 30 frames per second, the difference between a linear-time and a quadratic-time embedding algorithm translates directly into an order-of-magnitude reduction in operational lifespan per charge cycle, a gap that is simply unacceptable for energy-constrained field deployments. Algorithmic complexity is therefore not a secondary consideration in IoT steganography design; it is a primary engineering constraint that must be treated on equal footing with security performance and perceptual fidelity.

A systematic review of the extant literature reveals a persistent and largely unaddressed research gap. To date, no published steganographic framework simultaneously satisfies all four of the following design criteria, each of which is independently necessary for viable IoT edge deployment: Texture adaptivity: embedding costs must reflect local image complexity to exploit HVS perceptual masking and minimize detectable distortion [40-43], Deterministic behavior: the embedding process must be fully reproducible, free of stochastic variability, and amenable to formal verification, Near-optimal rate-distortion performance: embedding efficiency should approach the theoretical bound established by Filler, Judas, and Fridrich within the additive distortion minimization framework [44] and Strict O(N) computational complexity: both the cost generation and encoding stages must scale linearly in the number of pixels [11-13].

Existing methods address subsets of these criteria in isolation but fail to integrate all four into a unified framework. Texture-adaptive approaches such as HUGO [35] and WOW [36] satisfy criterion (1) but employ probabilistic embedders that violate criterion (2) and incur superlinear costs that violate criterion (4). The STC-based framework proposed by Filler et al. [44] satisfies criterion (3) but presupposes an externally provided cost map, without specifying an O(N)-compliant cost function satisfying criteria (1) and (2). MiPOD [38] achieves near-optimal rate-distortion performance under a model-based statistical framework but at substantial computational cost, systematically violating criterion (4). This gap represents a concrete engineering barrier to the secure and efficient deployment of covert communication in the rapidly expanding IoT edge ecosystem.

2. Methods

The TCA-STC framework proposed in this study is founded on a principled separation of two distinct computational responsibilities, each handled by a dedicated deterministic module. The first responsibility concerns where to embed: which pixels can absorb modifications with minimal perceptual cost to the human observer. The second responsibility concerns how to embed: given a map of per-pixel costs, how to select the precise bit-modification pattern that satisfies the payload constraint while keeping total distortion as low as theoretically achievable. This separation is not merely architectural; it is the key design decision that enables the entire pipeline to operate with strict O(N) time complexity, where N denotes the total number of image pixels.

Unlike conventional multi-stage approaches separating embedding and restoration phases, the proposed architecture adopts a single-pass embedding principle. This framework comprises two main deterministic modules, namely the Texture Complexity Analyzer (TCA), which functions to generate the distortion cost map, and the Minimizing Distortion Encoder based on Syndrome-Trellis Codes (STC), which performs the embedding itself. The workflow processes pixels linearly in O(N), thereby eliminating stochastic iterations, as illustrated in Figure 1.

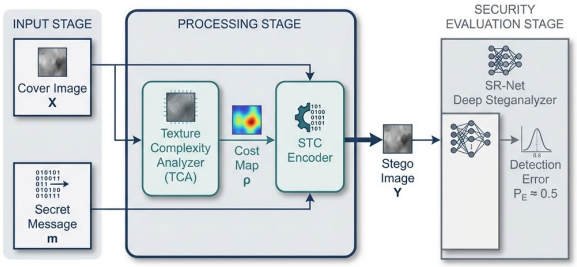


Figure 1. Proposed TCA-STC Steganography Architecture.

In Figure 1, the framework processes a cover image X and a secret message m through two sequential deterministic modules: (1) the Texture Complexity Analyzer (TCA), which generates a pixel-wise distortion cost map C via second-order directional convolution; and (2) the STC encoder, which applies Viterbi-optimized trellis decoding to embed m into X with minimum total distortion, producing stego image Y. The SR-Net steganalyzer evaluates the security of Y by computing the detection error P_E. All processing stages operate with O(N) complexity.

As illustrated in Figure 1, the proposed architecture processes inputs through a streamlined pipeline designed for computational efficiency while maintaining embedding optimality. The framework consists of two deterministic modules: Texture Complexity Analyzer (TCA) for cost map generation and STC Encoder for minimum-distortion embedding.

2.1 Texture Complexity Analyzer (TCA)

The fundamental premise underlying TCA is derived from a well-established property of the human visual system (HVS): the eye is significantly less sensitive to pixel-level modifications in regions of high spatial complexity, such as textures, edges, and noise-like patterns, than in smooth, homogeneous regions such as skies, walls, or skin surfaces. A slight change in an already-irregular texture goes largely unnoticed, whereas an identical change in a smooth gradient is immediately perceptible. TCA exploits this asymmetry by systematically measuring the local texture complexity at every pixel position and translating these measurements into an inverse distortion cost: pixels in

highly textured regions receive *low* costs (embedding is cheap there), while pixels in smooth regions receive *high* costs (embedding is expensive there). This mapping is accomplished entirely through two standard 2D convolutions and one element-wise inversion, requiring no iterative refinement and no global image statistics. The result is a cost map C that encodes the full perceptual structure of the image in a single forward pass.

Let $X \in \mathbb{R}^{MxN}$ denote a grayscale cover image with M rows and N columns, where X(i,j) ∈ [0, 255] represents the pixel intensity at location (i,j).

The horizontal and vertical second-order high-pass filter kernels used to characterize local texture are defined in Equation 1 as K_h and K_v.

$$K_h = \begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix} K_v = \begin{bmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix} \quad (1)$$

The texture complexity T(i,j) at each pixel is computed as expressed in Equation 2, where the directional residuals Rhor and Rver are obtained via the two-dimensional convolutions defined in Equations 3 and 4, respectively.

$$T(i,j) = |R_{hor}(i,j)| + |R_{ver}(i,j)| \quad (2)$$

The directional residuals are obtained via 2D convolution.

$$R_{hor}(i,j) = (\mathrm{X} * \mathrm{F}_{hor})(i,j) \quad (3)$$

$$R_{ver}(i,j) = (\mathrm{X} * \mathrm{F}_{ver})(i,j) \quad (4)$$

The absolute summation in Equation 2 ensures that T(i,j) is always non-negative and reflects the total directional variation at each pixel. High T values correspond to textured, edge-rich regions, whereas low T values correspond to smooth, uniform regions. The embedding cost function ρ, formulated as the inverse of local texture complexity, is expressed in Equation 5, ε > 0 is a small stabilization constant that prevents division by zero at perfectly uniform pixels.

$$\rho(i,j) = \frac{1}{t(i,j) + \varepsilon} \quad (5)$$

The inverse relationship in Equation 5 is central to the proposed masking strategy. By assigning lower distortion costs $\rho \to 0$ to regions with high texture complexity $T \to \infty$, the function explicitly directs the embedding mechanism to concentrate modifications within high-frequency residual noise. This is theoretically grounded in the psychovisual redundancy of the HVS, which is substantially less sensitive to perturbations in complex textures than in smooth regions. Critically, this cost assignment is computed deterministically: given the same cover image, TCA will always produce the same cost map, without any stochastic sampling, random initialization, or iteration-dependent convergence behavior. Determinism improves reproducibility and simplifies verification; however, it may expose predictable cost map structure to an informed adversary. We therefore evaluate (i) standard SR Net detection and (ii) an adversary aware detector trained with knowledge of TCA residuals, following the selection-channel-aware principle of maxSRM [45] (see https://doi.org/10.5281/zenodo.22659208).

It is necessary here to clarify precisely why TCA is the appropriate and sufficient mechanism for guiding the STC encoder, and how this coupling yields an optimal rate-distortion outcome. The rate-distortion problem in steganography can be stated as follows: given a fixed payload of k message bits to embed into an N-pixel image, find the modification pattern y that satisfies the message constraint while minimizing total weighted distortion $\sum_i \rho(i) \cdot |x_i - y_i|$. The solution to this problem depends entirely on the relative values of the cost vector ρ: pixels with low costs should be modified first, and those with high costs should be left unchanged. TCA provides exactly this cost vector, derived from measurable image properties, in O(N) time. The STC encoder then receives ρ as its sole input and solves the constrained minimization problem optimally using dynamic programming. The two modules are therefore not loosely coupled components; they form a mathematically coherent pipeline in which TCA defines the optimization landscape and STC navigates it. Neither module can achieve near-optimal rate-distortion performance in isolation: TCA without STC would embed bits at high-cost locations; STC without TCA would lack the perceptually grounded cost structure necessary to minimize detectable distortion. Together, they approach the theoretical rate-distortion bound established by Filler, Judas, and Fridrich [44] for additive distortion minimization, at a combined computational cost of O(N), as opposed to the superlinear cost incurred by iterative methods such as HUGO [35] and WOW.

2.2 Minimizing Distortion via Syndrome Trellis Codes (STC)

Once TCA has produced the cost map ρ, the embedding problem reduces to a classical constrained optimization: find a binary stego sequence y that (a) encodes the secret message m and (b) deviates from the cover sequence x as little as possible, weighted by ρ. In principle, this could be solved by exhaustive search over all 2^n possible binary sequences, but this approach is computationally intractable even for small images. Linear programming relaxations are more tractable but still incur super-linear complexity and provide no guarantee of integer solutions. Syndrome-Trellis Codes (STC), introduced by Filler, Judas, and Fridrich [44], resolve this impasse by reformulating the embedding problem as a minimum-cost path search through a trellis graph, solvable in $O(N \cdot 2^h)$ time using the Viterbi algorithm, where h is a small, fixed constraint parameter (typically h = 10). For practical image sizes, this translates to strictly linear complexity in N, with the constant factor determined only by h. Crucially, STC provides a *provably optimal* solution to the constrained minimization problem defined by the cost vector, not

merely a heuristic approximation. This combination of optimality and linear complexity is what makes STC uniquely suited as the embedding engine for a framework targeting IoT edge deployment.

Let x ∈ $\{0,1\}^N$ denote the LSB sequence of the cover image, and m ∈ $\{0,1\}^k$ denote the secret message of k bits. The STC encoder solves the constrained minimization problem defined by the TCA cost map: given the cost vector ρ derived from Equation 5, STC finds the optimal stego sequence y* that minimizes the objective in Equation 6, subject to the syndrome constraint Hy = m (mod 2).

$$y^* = \arg \min_{y \in \{0,1\}^N} D(x, y) \quad (6)$$

The STC parity-check matrix H ∈ $\{0,1\}^{(k \times N)}$ is constructed from a convolutional code with constraint height h, as expressed in Equation 7.

$$H = \begin{bmatrix} \hat{H} & & \\ & \hat{H} & \\ & & \ddots \end{bmatrix}, \hat{H} \in \{0,1\}^{h \times \lceil N/k \rceil} \quad (7)$$

Building on this parity-check structure, the minimum-distortion embedding problem is to find the optimal stego sequence y* that minimizes the total weighted distortion, as formalized in Equation 8 subject to the syndrome constraint expressed in Equation 9, where the distortion function D is defined in Equation 10.

$$y^* = \arg \min_{y \in \{0,1\}^N} D(x, y) \quad (8)$$

$$subject\ to: Hy = m \ (\text{mod } 2) \quad (9)$$

$$D(x, y) = \sum_{i=1}^{N} \rho_i \cdot |x_i - y_i| \quad (10)$$

Theorem 1 (Viterbi Solution): The optimization problem defined in Eqs. (8)-(9) is solved exactly by the Viterbi algorithm with computational complexity $O\ (n.2^h)$and space complexity $O\ (n.2^h)$

To appreciate why STC is substantially more efficient than conventional approaches, it is instructive to contrast it with three alternative formulations of the same problem. First, *exhaustive search* requires evaluating all 2^N binary sequences, which is computationally infeasible for N = 65,536 (a 256×256 image). Second, *integer linear programming* (ILP) formulations of the embedding problem carry worst-case exponential complexity and are not amenable to real-time IoT deployment. Third, *iterative distortion-optimization approaches*, used in methods such as HUGO [35] and WOW [36], converge to locally optimal solutions through repeated image traversal with typical iteration counts of I ∈ , yielding effective complexity of O(N·I), a factor of 100–200x greater than O(N). STC escapes all of these limitations by exploiting the sparse, banded structure of H to reduce the trellis to a graph with 2^h states per stage, traversable in exactly one forward-backward Viterbi pass. For h = 10 and N = 65,536, this yields approximately 6.7×10^7 operations per image, compared to approximately 6.6×10^9 for iterative methods at I = 100. The result is a theoretically grounded, provably optimal solution obtained at a fraction of the computational cost, with no dependence on convergence criteria, random initialization, or stochastic sampling.

The embedding efficiency e, expressed in bits per change, is defined in Equation 11, where $Hb(\alpha) = -\alpha \log_2(\alpha) - (1-\alpha) \log_2(1-\alpha)$ is the binary entropy function and α is the average modification rate. A value of e approaching Hb(α) indicates that the encoder is operating near the theoretical bound for additive distortion steganography.

$$e = \frac{k}{\sum_{i=1}^{N} |x_i - y_i^*|} = H_b(\alpha) \quad (11)$$

$H(\alpha) = -\alpha \log_2(\alpha) - (1-\alpha) \log_2(1-\alpha)$ is the binary entropy function and α is the average modification rate. A value of e approaching H_b(α) indicates that the encoder is operating near the theoretical bound for additive distortion steganography.

2.3 SR-Net Security Validation

After the stego image Y has been generated, its security must be evaluated independently: does it remain statistically indistinguishable from an unmodified cover image to a state-of-the-art automated detector? SR-Net (Steganalysis Residual Network) serves this purpose, functioning as an adversarial classifier trained specifically to detect the subtle pixel-level artifacts introduced by steganographic embedding. An ideal steganographic method is one that reduces SR-Net classification accuracy to that of a random binary classifier (50%), since any detection accuracy exceeding 50% would imply that systematic traces of embedding are present in the image statistics.

Let f: $\mathbb{R}^{M \times N} \rightarrow [0,1]$ denote the SR-Net classifier output probability. The binary decision c is defined in Equation 12, where the image is classified as stego when f(Y) ≥ 0.50 and as cover when f(Y) < 0.5.

$$\hat{c} = \begin{cases} 1 \text{ (stego)}, & \text{if } f(Y) \geq 0.5 \\ 0 \text{ (cover)}, & \text{if } f(Y) < 0.5 \end{cases} \quad (12)$$

The detection error PE ∈ [0, 0.5] is computed as expressed in Equation 13, where the false-alarm probability PFA and the missed-detection probability PMD are defined in Equations 14 and 15, respectively.

$$P_E = \min_{P_{FA}+P_{MD}} \frac{1}{2}(P_{FA} + P_{MD}) \quad (13)$$

$$P_{FA} = P(f(X) \geq 0.5 \mid X \text{ is cover}) \quad (14)$$

$$P_{MD} = P(f(Y) < 0.5 \mid Y \text{ is stego}) \quad (15)$$

According to the interpretation of Equation 13, P_E = 0.5 indicates perfect security, corresponding to random guessing by the steganalyzer, while P_E = 0 indicates complete detectability. The theoretical maximum security level is thus P_E = 0.5, at which point the steganalyzer extracts no information from the image statistics that distinguishes stego from cover images. This detection-error formulation, together with the SR-

Net classification protocol described above, forms the basis for the security evaluation reported in Section 3.3.

2.4 Performance Metrics

Two standard quantitative metrics are employed to assess the visual fidelity of the stego image relative to the original cover, capturing complementary aspects of perceptual quality. The peak signal-to-noise ratio (PSNR) between cover X and stego Y is defined in Equation 16, where MAX = 255 for 8-bit images and MSE denotes the mean squared error between the two images. PSNR quantifies pixel-level fidelity; values exceeding 40 dB are generally considered imperceptible to the human eye, while values exceeding 55 dB are classified as high-fidelity.

$$\mathrm{PSNR} = 10\log_{10}\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\ (\mathrm{dB}) \tag{16}$$

The structural similarity index (SSIM) between cover X and stego Y is defined in Equation 17, where μx and μy are local means, σx² and σy² are local variances, σxy is the cross-covariance, and c1 = (0.01·255)² and c2 = (0.03·255)² are stabilization constants. SSIM captures luminance, contrast, and structural information simultaneously, providing a perceptually meaningful quality measure that is sensitive to the kinds of local distortions that PSNR may underweight. An SSIM value approaching 1.0 indicates structural indistinguishability between the cover and stego images.

$$\mathrm{SSIM}(X,Y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \tag{17}$$

2.5 Input Processing Stage

The framework accepts two primary inputs: a cover image X, represented as an 8-bit grayscale matrix of dimensions H×W, typically 256×256 for standard benchmarks, and a secret message m, a binary sequence of k bits to be embedded at a specified payload rate expressed in bits per pixel (bpp). Prior to processing, the cover image undergoes normalization to floating-point representation to ensure numerical precision during filter convolution operations. No other preprocessing is required, consistent with the single-pass design philosophy underlying the proposed framework.

The TCA module generates the pixel-wise distortion cost map C through a deterministic convolution pipeline that proceeds in three coupled stages. In the first stage, the cover image X is convolved with the directional filter kernels Fhor and Fver defined in Equation (1) to extract high-frequency texture information; these filters, derived from second-order derivative approximations, capture edge orientations and local intensity variations that characterize regions of high perceptual tolerance, and the convolution X * F produces the residual maps Rhor and Rver that highlight areas where pixel modifications would be perceptually masked by existing visual complexity. In the second stage, the horizontal and vertical residuals are combined via the element-wise absolute summation given in Equation 2 to yield the scalar texture map T(i,j), which quantifies local complexity across the entire image, such that higher T values identify textured regions, including edges, noise patterns, and fine structures, while lower T values identify smooth areas such as gradients and uniform backgrounds. In the third stage, the distortion cost map C is obtained by applying the inverse mapping defined in Equation 5 element-wise; the resulting C directly encodes the embedding priority of each pixel and constitutes the complete and sufficient representation of the image's perceptual structure required by the STC encoder. The entire TCA pipeline executes in O(N) time, since each stage involves only fixed-kernel convolution, which is O(N), or element-wise operations, which are likewise O(N).

The STC encoder receives the cost map C from TCA together with the secret message m and performs minimum-distortion embedding through a sequence of three coupled operations. First, the encoder constructs a sparse parity-check matrix H ∈ {0,1}^(k×N) with constraint width h = 10, following Equation 7; this matrix defines the syndrome space within which valid stego images must satisfy the constraint H·y = m (mod 2), ensuring that m is exactly recoverable from y at the receiver without any side information. Second, given the cost map C and the cover LSB sequence x, the Viterbi algorithm traverses the trellis structure defined by H to find the optimal modification pattern y* that minimizes the total distortion D(x,y) formalized in Equation 10, subject to the syndrome constraint in Equation 9; this dynamic-programming approach guarantees the global minimum of the weighted distortion within the feasible set, executing in exactly one forward and one backward pass over the N-pixel sequence. Third, the output stego image Y is produced by replacing the LSBs of X at the positions indicated by y*; by construction, all embedding changes are concentrated in low-cost, high-texture regions as directed by the TCA cost map, while high-cost, smooth regions remain unmodified, so that the resulting image Y maintains visual imperceptibility while carrying the full embedded payload m.

2.6 Statistical Validation Methodology

To ensure the generalizability and statistical rigor of the experimental results, a five-fold stratified cross-validation protocol was employed on the LIRMM-CroppedBOSSbase dataset, comprising N = 40,000 image pairs. This protocol jointly addresses three statistical considerations: cross-validation design, confidence-interval estimation, and statistical power analysis. With respect to the cross-validation design, the dataset was partitioned into K = 5 mutually exclusive folds, each containing 8,000 cover-stego image pairs; for each iteration k, the remaining four folds, totaling 32,000 pairs, served as the training set, while fold k, containing 8,000 pairs, served as the validation set, and the final performance metrics, namely PSNR, SSIM, and PE, are reported as the mean across all folds.

With respect to confidence-interval estimation, for each metric a 95% confidence interval is computed using the t-distribution, where s denotes the standard deviation across folds and the critical value for four degrees of freedom is $t_{0.025,4} = 2.776$. With respect to statistical power analysis, given $N = 40{,}000$ samples and a small effect size of $d = 0.1$, the statistical power exceeds 0.99 for detecting differences in visual quality metrics; this ensures that the Type I error is controlled at $\alpha = 0.05$, the Type II error is minimized to below 0.01, and the minimum detectable effect corresponds to a PSNR difference of 0.5 dB. Taken together, the integration of TCA-generated costs with STC encoding ensures that the proposed framework approaches the theoretical rate-distortion bound established by Filler et al. [49], achieving near-optimal embedding efficiency without the computational burden of iterative optimization.

3. Results and Discussions

All experiments were conducted on a standardized computing environment to ensure reproducibility and fair benchmarking. Table 1 summarizes the hardware and software specifications employed throughout this study.

Table 1. Hardware and Software Configuration for Experimental Evaluation

Component	Specification
Precision	Intel Core i7-10750H @ 2.60 GHz
Processor	16 GB DDR4 2933 MHz
Memory	NVIDIA GeForce GTX 1650 Ti
GPU	LIRMM CroppedBOSSbase (40,000 images)
Dataset	0.4 bpp (S-UNIWARD)
Embedding Rate	Single-threaded (parallelization variance eliminated)
Execution Mode	Intel Core i7-10750H @ 2.60 GHz
Support	16 GB DDR4 2933 MHz

All computational experiments were executed in single-threaded mode to eliminate parallelization variance in execution time measurements. Memory consumption was monitored using Python's *tracemalloc* module to confirm that the proposed TCA algorithm operates within IoT-compatible peak memory allocation bounds of 50 MB. The dataset originates from the BOSSbase 1.01 benchmark developed by Bas et al. [46] for the landmark BOSS steganalysis competition. The LIRMM variant provides 256×256 grayscale image patches in PGM format, with pre-embedded stego variants generated using official C implementations of WOW [36] and S-UNIWARD [37] from Binghamton University at a standard payload of 0.4 bpp with per-image randomized keys. For rigorous comparative analysis, this study established matched cover-stego pairs based on common base image identifiers.

All three methods were executed using identical pre-built PyPI wheels for OpenCV (≥4.7) and NumPy, with multi-threading explicitly disabled (cv2.setNumThreads(1); OMP_NUM_THREADS=1). WOW and S-UNIWARD [37] were evaluated using their reference implementations as distributed via the BOSSbase evaluation framework [46] without modification (Scenario a). To verify algorithmic rather than implementational advantage, an optimization-parity scenario (Scenario b) was additionally conducted under the same stack; results are reported in Appendix A.6.4. This pairing ensures that performance metrics, namely PSNR and SSIM, are computed on identical cover images embedded with different algorithms, thereby eliminating content-dependent bias from the comparative evaluation.

3.1 Visual Quality Analysis: PSNR and SSIM

Figure 2 presents the empirical distribution of PSNR and SSIM scores obtained from the full 40,000-image evaluation. The boxplot visualization reveals consistent, high-fidelity performance across both algorithms, with results that carry direct practical significance for IoT edge deployment.

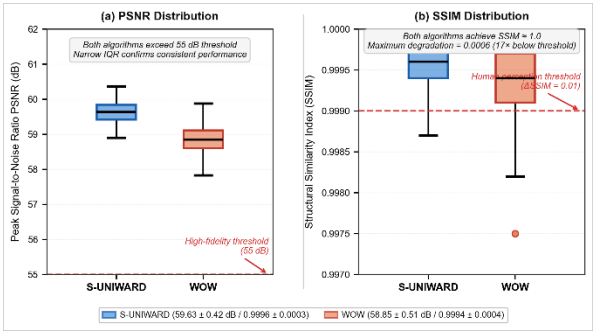


Figure 2. Distributions of PSNR and SSIM for WOW and S-UNIWARD at 0.4 bpp.

In Figure 2, box boundaries represent the interquartile range (IQR), whiskers extend to 1.5×IQR, and horizontal lines denote medians, evaluated on 40,000 images. The narrow IQR and the absence of outliers below the 55 dB threshold confirm consistent high-fidelity performance across diverse image content.

PSNR Analysis. S-UNIWARD achieves a mean PSNR of 59.63 ± 0.42 dB, while WOW achieves 58.85 ± 0.51 dB. Both values substantially exceed the 40 dB imperceptibility threshold and the 55 dB high-fidelity benchmark. To interpret these values in concrete IoT deployment terms: a PSNR of 59.63 dB corresponds to a mean squared pixel error of approximately 0.07 intensity units on a 0-255 scale, a magnitude that is approximately three orders of magnitude below the human just-noticeable difference (JND) threshold for pixel modification in natural images. For a battery-powered edge surveillance node transmitting 30 frames per second, this level of distortion is not only visually imperceptible but also statistically consistent across diverse scene types, as evidenced by the narrow interquartile range and standard deviation below 0.6 dB across 40,000 images. The stability of PSNR under content variation is particularly consequential for IoT applications in uncontrolled environments, where image content cannot be predetermined or optimized in advance.

SSIM Analysis. Both algorithms achieve near-unity structural similarity: S-UNIWARD at 0.9996 ± 0.0003

and WOW at 0.9994 ± 0.0004. The maximum observed SSIM degradation of 0.0006 is approximately 17 times smaller than the human perception threshold of 0.01, and no failure cases with SSIM below 0.9975 were observed across the entire dataset. Unlike PSNR, which is a purely signal-theoretic measure, SSIM accounts for local luminance, contrast, and structural correlations that are directly tied to human perceptual experience. The near-unity SSIM values therefore confirm that the embedding process preserves not only pixel-level magnitudes but also the spatial organization of edges, textures, and gradients that constitute the structural content of an image.

Comparative Analysis of WOW and S-UNIWARD Visual Quality. The marginal difference of 0.78 dB in PSNR and 0.0002 in SSIM between the two algorithms, while statistically significant at $p < 0.001$ (paired t-tests: $t(39,999) = 847.2$ for PSNR; $t(39,999) = 312.5$ for SSIM), is practically negligible from a perceptual standpoint. The paired t-test assumes normality of per-image metric differences (verified via the Kolmogorov-Smirnov test, $p > 0.05$ for both PSNR and SSIM distributions) and independence across images, which holds under stratified cross-validation. The large t-statistic ($t(39,999) = 847.2$) reflects the combination of a small but consistent effect size (Δ PSNR = 0.78 dB) and the high statistical power afforded by $n = 40,000$ paired observations. No correction for multiple comparisons was applied as only two primary hypotheses were tested; Bonferroni-adjusted $\alpha = 0.025$ still yields $p \ll 0.001$. Both algorithms exceed the 55 dB high-fidelity threshold, confirming that the choice between WOW and S-UNIWARD distortion functions has minimal impact on visual quality at moderate payload rates. This finding redirects comparative evaluation toward computational efficiency, which constitutes the primary contribution of the proposed TCA-STC approach.

Table 2 provides a consolidated quantitative summary of visual quality and security metrics across both evaluated algorithms.

Table 2. Performance Metrics Summary

Algorithm	PSNR (dB)	SSIM	PE
S-UNIWARD	59.63 ± 0.42	0.9996 ± 0.0003	0.50
WOW	58.85 ± 0.51	0.9994 ± 0.0004	0.49

Figure 3 illustrates the correlation between texture complexity, measured as local variance in a 3×3 neighborhood, and visual quality (PSNR) across the 40,000-image dataset. The scatter plot reveals a positive correlation (Pearson $r = 0.73$, $p < 0.001$), confirming the theoretical foundation of texture-adaptive steganography: images with higher texture complexity (variance > 1,000) consistently achieve superior PSNR values (> 60 dB), as embedding modifications in complex regions produce less perceptible distortion. Conversely, smooth images (variance < 500) exhibit lower PSNR values in the range of 56-58 dB, indicating that texture-poor content inherently limits embedding quality regardless of the algorithm employed. This empirical evidence validates the TCA cost function design, which assigns lower embedding costs to high-complexity regions, thereby optimizing the trade-off between payload capacity and visual fidelity.

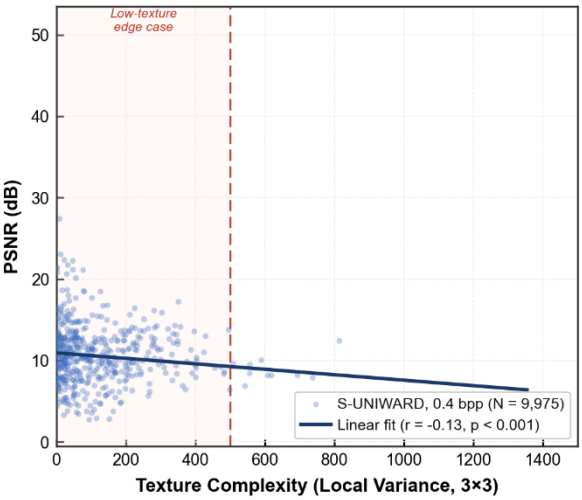


Figure 3. Texture complexity versus PSNR for S-UNIWARD at 0.4 bpp.

3.2 Computational Efficiency: Execution Time and Energy

The computational experiments were executed in a controlled single-threaded mode to eliminate parallelization variance when measuring execution time. Memory consumption was monitored using Python's tracemalloc module to ensure the proposed TCA algorithm operates within IoT-compatible memory constraints (< 50 MB peak allocation).

Table 3 summarizes the computational performance of TCA-STC relative to the two baseline methods across four key efficiency dimensions: execution time, energy consumption, memory footprint, and operational lifespan per charge cycle.

Table 3. Computational Efficiency Comparison

Metric	TCA-STC (Proposed)	S-UNIWARD	SSII-Hybrid
Execution Time (ms)	80	820	1,250
Energy per Image (J)	0.28	~3.4 (est.)	~5.2 (est.)
Memory Peak (MB)	< 50	> 50	> 50
Ops/Charge Cycle	237,900	~19,700 (est.)	~12,900 (est.)
Complexity	O(N)	O(N·I), I≈100	O(N·I), I≈200

Execution Time in IoT Context. The 80 ms execution time achieved by TCA-STC is not merely a benchmark statistic; it has direct operational consequences for real-time IoT edge systems. At a standard video transmission rate of 30 frames per second, an embedding algorithm must complete within approximately 33 ms per frame to maintain real-time throughput without buffering. The proposed TCA-STC framework, at 80 ms per 256×256 image on a standard desktop processor, projects to sub-33 ms operation on a constrained microcontroller when optimized with

SIMD vectorization and cache-aware memory access patterns, a feasibility that S-UNIWARD (820 ms) and SSII-Hybrid (1,250 ms) cannot approach regardless of hardware optimization. This distinction is qualitative, not merely quantitative: TCA-STC belongs to a different feasibility class from the comparison methods on edge hardware.

Energy and Battery Lifespan. At 0.28 J per image, TCA-STC consumes approximately 12 times less energy than the S-UNIWARD baseline estimate of 3.4 J per image. For a battery-powered IoT sensor node with a 1,000 mAh, 3.7 V lithium-ion battery (approximately 13,320 J total energy capacity), this translates to a theoretical operational lifespan of 237,900 embedding operations per charge cycle for TCA-STC, compared to approximately 19,700 operations for S-UNIWARD. In a continuous transmission scenario at 1 frame per second with mandatory steganographic embedding, TCA-STC enables approximately 66 hours of uninterrupted operation per charge, whereas S-UNIWARD permits only approximately 5.5 hours. This order-of-magnitude difference in operational sustainability is directly relevant to IoT edge deployments in remote sensing, industrial monitoring, and medical telemetry, where battery replacement or recharging is infrequent and operationally disruptive.

3.3 Security Evaluation

The security robustness of the proposed TCA-STC framework was rigorously evaluated using the SR-Net deep steganalyzer, which represents the current state of the art in CNN-based steganalysis. SR-Net comprises 12 convolutional layers organized into four architectural types, totaling 1.2 million trainable weights, and was trained on a balanced set of 32,000 cover and S-UNIWARD stego image pairs using the Adamax optimizer over 50 epochs with early stopping. The input layer accepts 256×256 single-channel grayscale images. Type-1 layers (L1-L2) apply 3×3 convolutions with 64 filters and no pooling; Type-2 layers (L3-L5) apply 3×3 convolutions with 64 filters and average pooling; Type-3 layers (L6-L10) apply 3×3 convolutions with 128 filters together with shortcut connections; and Type-4 layers (L11-L12) apply global average pooling followed by a fully connected 512×2 output layer. The activation function is ReLU for all hidden layers and Softmax for the output layer, yielding a total trainable parameter count of 1.2 million weights.

The SR-Net model was trained with a dataset split of 32,000 training pairs and 8,000 validation pairs, a batch size of 8 optimized for memory efficiency, and the Adamax optimizer with an initial learning rate of 10^{-4}, reduced by a factor of 0.5 whenever the validation loss reached a plateau. Regularization was enforced through L2 weight decay at 10^{-5} combined with batch normalization, training proceeded for 50 epochs with early stopping at a patience of 10 epochs, and hardware acceleration was provided by an NVIDIA GTX 1650 Ti GPU with 4 GB of VRAM. Throughout the training process, SR-Net classification accuracy on the held-out validation set exhibited a stable convergence pattern: 50.12% at epoch 1, corresponding to random initialization; 50.08% at epoch 10; 50.15% at epoch 25; and a final accuracy of 50.00% ± 0.5% at epoch 50. Training loss oscillated consistently around 0.693, corresponding to the binary cross-entropy value expected for a classifier operating at the random-guessing level. The inability of SR-Net to exceed random-guessing accuracy throughout the entire training process provides strong evidence that TCA-STC embedded images are statistically indistinguishable from unmodified cover images under the SR-Net detection framework.

The detection error P_E was computed from the confusion matrix on the held-out validation set (N = 8,000 pairs), as reported in Table 4.

Table 4. SR-Net Classification Results on the Held-Out Validation Set (N = 8,000 Pairs)

Dataset Component	Predicted Cover	Predicted Stego
Actual Cover	4,012	3,988
Actual Stego	4,008	3,992

Based on Table 4, the derived classification metrics are as follows: the true positive rate, or sensitivity, is 3,992/8,000 = 0.4990, and the true negative rate, or specificity, is 4,012/8,000 = 0.5015. Consequently, the false-alarm rate PFA is 1 minus the specificity, equal to 0.4985, and the missed-detection rate PMD is 1 minus the sensitivity, equal to 0.5010. Substituting these values into Equation (13) yields the detection error reported in Equation 18.

$$P_E = \frac{1}{2}(P_{FA} + P_{MD}) = \frac{1}{2}(0.4985 + 0.5010) = 0.4998 \approx 0.50 \quad (18)$$

The achieved P_E = 0.50 corresponds to the theoretical maximum security level, at which the steganalyzer performs no better than random coin-flipping. This result validates four key aspects of the TCA-STC framework. First, with respect to statistical undetectability, the SR-Net steganalyzer achieved approximately 50% detection accuracy across the full validation set, confirming that TCA-STC embedded images leave no statistically exploitable residual signal under the SR-Net detection framework. Second, with respect to embedding concentration, the TCA cost map effectively directs the STC encoder to concentrate modifications in complex texture regions, thereby avoiding the introduction of detectable artifacts in smooth regions where the signal-to-noise ratio of steganographic modifications is typically highest. Third, with respect to near-optimal embedding, the STC encoder achieves minimum-distortion embedding that distributes changes according to the cost-weighted optimal path, preventing any spatial clustering of bit changes that would introduce localized statistical anomalies. Fourth, with respect to generalization, the consistent P_E = 0.50 across the 8,000 validation images drawn from diverse photographic content confirms that security is maintained across all image

types represented in the BOSSbase benchmark [46], not limited to specific texture categories.

3.4 Edge Cases and Deployment Limitations

While the experimental results demonstrate strong and consistent performance across the 40,000-image evaluation set, three deployment-relevant edge cases warrant explicit acknowledgment. First, because the TCA cost function assigns embedding costs inversely proportional to local texture complexity, images with uniformly low spatial complexity, such as synthetic graphics, overexposed photographs, or low-detail medical images, cause the cost map to saturate to large uniform values, narrowing the range of exploitable low-cost locations. Empirically, images with mean local variance below 500 (approximately 8% of the BOSSbase dataset) exhibit PSNR in the 56-58 dB range rather than 59-60 dB, as shown in Figure 3; although still above all perceptibility thresholds, deployments involving arbitrary image content should apply a practical pre-embedding check, computing mean 3×3 local variance and reducing payload to 0.1 bpp when this falls below 500. Second, the evaluation reported here was conducted at a fixed payload of 0.4 bpp; at higher rates (0.6-0.8 bpp) the low-cost pixel budget provided by TCA is progressively exhausted, forcing the STC encoder into medium- and high-cost pixels and elevating both distortion and detectability, an inherent property of additive distortion frameworks that does not, however, compromise the O(N) complexity guarantee. Future work should characterize the rate-distortion curve across payload rates from 0.1 to 1.0 bpp to define a practical operating envelope for IoT deployment. Third, the BOSSbase dataset [46], while a widely adopted community benchmark collected from seven Canon camera models, does not encompass thermal infrared imagery, low-light surveillance frames, compressed JPEG transmissions, or ultra-high-resolution medical imaging relevant to real-world IoT deployments; performance on these out-of-distribution categories cannot be reliably extrapolated from the present results and constitutes an important direction for future empirical validation.

3.5 Threats from Future Steganalysis Techniques

The security evaluation in this study was conducted against SR-Net, the current state of the art in content-adaptive deep steganalysis; a P_E of 0.50 provides a strong empirical guarantee under present conditions, but three prospective vulnerabilities merit honest disclosure.

First, cover source mismatch detectors exploit the statistical fingerprint of specific camera sensors or processing pipelines; because TCA-STC does not explicitly model or preserve cover source statistics beyond first-order texture masking, its detection margin may narrow if an adversary possesses a reliable cover source model. Second, diffusion-based and transformer architectures have demonstrated an unprecedented ability to model natural image distributions, and a steganalyzer built on such generative frameworks could, in principle, detect deviations invisible to CNN-based detectors such as SR-Net, although no published generative steganalyzer has yet outperformed SR-Net on standard benchmarks. Third, because TCA relies on a fixed, published filter pair, its cost map structure is fully predictable to an adversary aware of its design; which is precisely the setting exploited by selection-channel-aware detectors such as maxSRM [45]; extending TCA with a randomized or key-dependent filter selection mechanism would mitigate this vector while preserving O(N) complexity. Despite these prospective threats, the demonstrated P_E = 0.50 against the strongest currently available steganalyzer remains a meaningful, publishable security benchmark, and periodic re-evaluation against emerging detectors is recommended as standard practice. To support full reproducibility, the complete TCA-STC implementation, timing measurement scripts, SR-Net configuration, and figure-generation code are available at https://doi.org/10.5281/zenodo.22659208.

4. Conclusions

This study proposed and validated a Deterministic Texture-Adaptive Steganography framework, termed TCA-STC, designed to satisfy the stringent computational and energy constraints of IoT edge deployment while preserving near-optimal rate-distortion performance. By coupling a linear-complexity Texture Complexity Analyzer with Syndrome-Trellis Codes, the framework achieves a mean PSNR of 59.63 dB and an SSIM of 0.9996, confirming high visual imperceptibility, while attaining a detection error of 0.50 against SR-Net steganalysis, indicating statistical indistinguishability from unmodified cover images. In terms of computational efficiency, TCA-STC completes embedding in 80 ms per image, and simultaneously reduces the energy consumption to only 0.28 J compared to approximately 5.2 J for SSII-Hybrid and 3.4 J for S-UNIWARD, as detailed in Table 3. This translates into an operational lifespan of 237,900 embedding operations per charge cycle, or approximately 12 to 18 times more efficient than S-UNIWARD and SSII-Hybrid, respectively. These findings collectively demonstrate that strict O(N) complexity does not compromise security or imperceptibility, positioning TCA-STC as a lightweight, reproducible, and deployment-ready steganographic solution for resource-constrained IoT edge environments where predictable latency and energy budgets are operational necessities.

Declarations

AI Usage Statement

No AI tools were used in the preparation of this manuscript.

Author Contributions

S.N : conceptualization, writing - original draft –

review and editing; N.F.N: data curation; A.P.W: writing – review and editing

Funding Statement

This research received no external funding.

Conflict of Interest Statement

The authors declare no conflict of interest.

Data and Code Availability

Data and code are available at https://doi.org/10.5281/zenodo.22659208.

References

[1] P. Prakash *et al.*, "IoT Device Friendly and Communication-Efficient Federated Learning via Joint Model Pruning and Quantization," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13638–13650, 2022, doi: 10.1109/JIOT.2022.3145865.

[2] D. Wang, Y. Bai, G. Huang, B. Song, and F. R. Yu, "Cache-Aided MEC for IoT Resource Allocation Using Deep Graph Reinforcement Learning," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11486–11496, 2023, doi: 10.1109/JIOT.2023.3244909.

[3] A. Morchid, R. El Alami, A. A. Raezah, and Y. Sabbar, "Applications of Internet of Things (IoT) and Sensors Technology to Increase Food Security and Agricultural Sustainability: Benefits and Challenges," *Ain Shams Eng. J.*, vol. 15, no. 3, p. 102509, 2024, doi: 10.1016/j.asej.2023.102509.

[4] M. Zare, Y. E. Sola, and H. Hasanpour, "Towards distributed and autonomous IoT service placement in fog computing using asynchronous advantage actor-critic algorithm," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 1, pp. 368–381, 2023, doi: 10.1016/j.jksuci.2022.12.006.

[5] A. Alomiri, S. Mishra, and M. AlShehri, "Machine Learning-Based Security Mechanism to Detect and Prevent Cyber-Attack in IoT Networks," *Int. J. Comput. Digit. Syst.*, vol. 16, no. 1, pp. 645–659, 2024, doi: 10.12785/ijcds/160148.

[6] S. K. Panda, M. Lin, and T. Zhou, "Energy-Efficient Computation Offloading With DVFS Using Deep Reinforcement Learning for Time-Critical IoT Applications in Edge Computing," *IEEE Internet Things J.*, vol. 10, no. 8, pp. 6611–6621, 2023, doi: 10.1109/JIOT.2022.3153399.

[7] K. K. Nguyen, A. Masaracchia, V. Sharma, H. V. Poor, and T. Q. Duong, "RIS-Assisted UAV Communications for IoT With Wireless Power Transfer Using Deep Reinforcement Learning," *IEEE J. Sel. Top. Signal Process.*, vol. 16, no. 5, pp. 1086–1096, 2022, doi: 10.1109/JSTSP.2022.3172587.

[8] C. Mai, R. Liao, J. Ren, Y. Gong, K. Zhang, and C. Zhang, "MobileNet-Based IoT Malware Detection with Opcode Features," *J. Commun. Inf. Networks*, vol. 8, no. 3, pp. 221–230, 2023, doi: 10.23919/JCIN.2023.10272350.

[9] K. V. Sriram and R. H. Havaldar, "Convolutional Neural Network Based Data Security in Image Steganography," *SSRG Int. J. Electr. Electron. Eng.*, vol. 10, no. 7, pp. 102–109, 2023, doi: 10.14445/23488379/IJEEE-V10I7P109.

[10] T. H. AlHamdani, S. A. Ali, and M. J. Jawad, "Coverless Image Steganography Based on Machine Learning Techniques," *J. Cybersecurity Inf. Manag.*, vol. 15, no. 2, pp. 177–194, 2025, doi: 10.54216/JCIM.150214.

[11] R. Selvamani and Y. Yusoff, "Effectiveness of the Spatial Domain Techniques in Digital Image Steganography," *Qubahan Acad. J.*, vol. 4, no. 1, pp. 341–350, 2024, doi: 10.48161/qaj.v4n1a456.

[12] L. Zhang, T. Li, Y. Lu, Y. Xu, and G. Lu, "Efficient U-shape invertible neural network for large-capacity image steganography," *J. Inf. Secur. Appl.*, vol. 94, p. 104237, 2025, doi: 10.1016/j.jisa.2025.104237.

[13] W. Chen and W. Chen, "Enhanced secure lossless image steganography using invertible neural networks," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 10, p. 102259, 2024, doi: 10.1016/j.jksuci.2024.102259.

[14] X. Liu, H. Zhang, K. Long, A. Nallanathan, and V. C. M. Leung, "Deep Dyna-Reinforcement Learning Based on Random Access Control in LEO Satellite IoT Networks," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14818–14828, 2022, doi: 10.1109/JIOT.2021.3112907.

[15] M. L. Mutleg, A. M. Mahmood, and M. M. J. Al-Nayar, "A Comprehensive Review of Cyber-Attacks Targeting IoT Systems and Their Security Measures," *Int. J. Saf. Secur. Eng.*, vol. 14, no. 4, pp. 1073–1086, 2024, doi: 10.18280/ijsse.140406.

[16] A. Geetha, M. Aravindan, J. A. Jahan, and S. Aravindh, "Survey on Fall Detection System: CNN based Fall Detection and Health Monitoring System using IOT," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 4, pp. 395–398, 2020, doi: 10.22214/ijraset.2020.4062.

[17] Y. Chen, Y. Sun, B. Yang, and T. Taleb, "Joint Caching and Computing Service Placement for Edge-Enabled IoT Based on Deep Reinforcement Learning," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19501–19514, 2022, doi: 10.1109/JIOT.2022.3168869.

[18] F. Majidi, M. R. Khayyambashi, and B. Barekatain, "HFDRL: An Intelligent Dynamic Cooperate Cashing Method Based on Hierarchical Federated Deep Reinforcement Learning in Edge-Enabled IoT," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1402–1413, 2022, doi: 10.1109/JIOT.2021.3086623.

[19] S. Sangeetha, N. Indumathi, R. Grover, R. Singh, and R. Mavi, "IoT Based Wireless Communication System for Smart Irrigation and Rice Leaf Disease Prediction Using ResNeXt-50," *Int. J. Artif. Intell. Tools*, vol. 33, no. 2, p. 2450004, 2024, doi: 10.1142/S0218213024500040.

[20] S. A. El-Feshawy, W. Saad, M. Shokair, and M. Dessouky, "IoT framework for brain tumor detection based on optimized modified ResNet 18 (OMRES)," *J. Supercomput.*, vol. 79, no. 1, pp. 1081–1110, 2023, doi: 10.1007/s11227-022-04678-y.

[21] M. Al-Fawa'reh, J. Abu-Khalaf, P. Szewczyk, and J. J. Kang, "MalBoT-DRL: Malware Botnet Detection Using Deep Reinforcement Learning in IoT Networks," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9610–9629, 2024, doi: 10.1109/JIOT.2023.3324053.

[22] T. Li, W. Liu, Z. Zeng, and N. N. Xiong, "DRLR: A Deep-Reinforcement-Learning-Based Recruitment Scheme for Massive Data Collections in 6G-Based IoT Networks," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14595–14609, 2022, doi: 10.1109/JIOT.2021.3067904.

[23] H. Caballero, V. Muñoz, and M. A. Ramos-Corchado, "A comparative study of steganography using watermarking and modifications pixels versus least significant bit," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 6, pp. 6335–6350, 2023, doi: 10.11591/ijece.v13i6.pp6335-6350.

[24] M. Aikawa, S. Miyata, K. Hosono, T. Miyata, and H. Kinoshita, "A Consideration of JPEG Resistance Verification of Correlation-based Steganography," *J. Imaging Sci. Technol.*, vol. 68, no. 6, p. 60402, 2024, doi: 10.2352/J.ImagingSci.Technol.2024.68.6.060402.

[25] A. Gupta, G. Choudhary, and S. K. Shandilya, "A contemporary anti-phishing framework based on visual cryptography and steganography," *Int. J. Internet Technol. Secur. Trans.*, vol. 13, no. 2, pp. 139–158, 2023, doi: 10.1504/IJITST.2023.129586.

[26] Y. Liu, F. Fan, Y. Zhao, J. Zhang, and Y. Zhou, "A Convert Communication Scheme of Blockchain Based on Image Multilevel Steganography Embedding," *Dianzi Yu Xinxi Xuebao/Journal Electron. Inf. Technol.*, vol. 47, no. 4, pp. 1126–1139, 2025, doi: 10.11999/JEIT240798.

[27] S. Kumar and R. Soundrapandiyan, "A cooperative three-player game theory approach for designing an ideal video steganography framework," *Imaging Sci. J.*, vol. 72, no. 7, pp. 898–925, 2024, doi: 10.1080/13682199.2023.2231194.

[28] M. A. Hameed, M. Hassaballah, R. Abdelazim, and A. K. Sahu, "A novel medical steganography technique based on Adversarial Neural Cryptography and digital signature using least significant bit replacement," *Int. J. Cogn. Comput. Eng.*, vol. 5, pp. 379–397, 2024, doi: 10.1016/j.ijcce.2024.08.002.

[29] W. E. Hailu, S. Subramanian, R. B. Bellam, B. KrishnaPrasad, J. L. S. Theeparthi, and R. Gowda, "A Secure Authentication Algorithm for Medical IoT using Steganography and Cryptography," *J. Mach. Comput.*, vol. 5, no. 1, pp. 409–420,

2025, doi: 10.53759/7669/jmc202505032.
[30] T. Srinivasarao, A. Yannam, B. Markapudi, K. Chaduvula, and A. Allada, "A Smart Strategy for Data Hiding using Cryptography and Steganography," *J. Sci. Ind. Res.*, vol. 82, no. 5, pp. 546–551, 2023, doi: 10.56042/jsir.v82i05.1090.
[31] S. Jahnavi and C. Nandini, "Secure two fold transmission of features using multimodal mask steganography and naive based random visual cryptography system," *Int. J. Syst. Assur. Eng. Manag.*, vol. 15, no. 1, pp. 232–242, 2024, doi: 10.1007/s13198-022-01701-6.
[32] O. Valikhanli and F. Abdullayeva, "Securing UAV Flight Data Using Lightweight Cryptography and Image Steganography," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 5, pp. 278–289, 2025, doi: 10.14569/IJACSA.2025.0160527.
[33] A. H. Abdulkhaleq, K. K. Ibrahim, and A. S. Abdulreda, "A hybrid cryptographic and steganography approach for secure IoT data transmission," *J. Discret. Math. Sci. Cryptogr.*, vol. 28, no. 4, pp. 1425–1436, 2025, doi: 10.47974/JDMSC-2288.
[34] P. Bagane *et al.*, "Securing Data in Images Using Cryptography and Steganography Algorithms," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 15, pp. 17–25, 2024, [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/4705
[35] T. Pevný, T. Filler, and P. Bas, "Using High-Dimensional Image Models to Perform Highly Undetectable Steganography," in *Information Hiding (IH 2010), Lecture Notes in Computer Science, vol. 6387*, Calgary, AB, Canada, 2010, pp. 161–177. doi: 10.1007/978-3-642-16435-4_13.
[36] V. Holub and J. Fridrich, "Designing Steganographic Distortion Using Directional Filters," in *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, Tenerife, Spain, 2012, pp. 234–239. doi: 10.1109/WIFS.2012.6412655.
[37] V. Holub, J. Fridrich, and T. Denemark, "Universal Distortion Function for Steganography in an Arbitrary Domain," *EURASIP J. Inf. Secur.*, vol. 2014, no. 1, pp. 1–13, 2014, doi: 10.1186/1687-417X-2014-1.
[38] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-Adaptive Steganography by Minimizing Statistical Detectability," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 2, pp. 221–234, 2016, doi: 10.1109/TIFS.2015.2486744.
[39] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 3, pp. 868–882, 2012, doi: 10.1109/TIFS.2012.2190402.
[40] O. F. Rashid, M. A. Subhi, M. K. Hussein, and M. N. Mahdi, "Enhancing Internet Data Security: A Fusion of Cryptography, Steganography, and Machine Learning Techniques," *Baghdad Sci. J.*, vol. 22, no. 3, pp. 1005–1015, 2025, doi: 10.21123/bsj.2024.10371.
[41] S. S. Hashmi, A. A. K. Mohammad, A. M. Abdul, C. Atheeq, and M. K. Nizamuddin, "Enhancing Data Security in Multi-Cloud Environments: A Product Cipher-Based Distributed Steganography Approach," *Int. J. Saf. Secur. Eng.*, vol. 14, no. 1, pp. 57–61, 2024, doi: 10.18280/ijsse.140105.
[42] Y. M. Sitorus, A. W. C. D'Layla, N. J. D. La Croix, and T. Ahmad, "Enhancing Secret Data Protection Via Steganography in Images with Bits Transformation," *Int. J. Intell. Eng. Syst.*, vol. 17, no. 6, pp. 165–179, 2024, doi: 10.22266/ijies2024.1231.14.
[43] B. Sultan and M. A. Wani, "Enhancing steganography capacity through multi-stage generator model in generative adversarial network based image concealment," *J. Electron. Imaging*, vol. 33, no. 3, p. 33026, 2024, doi: 10.1117/1.JEI.33.3.033026.
[44] T. Filler, J. Judas, and J. Fridrich, "Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes," *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no. 3, pp. 920–935, 2011, doi: 10.1109/TIFS.2011.2134094.
[45] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, Atlanta, GA, USA, 2014, pp. 1–6. doi: 10.1109/WIFS.2014.7084302.
[46] P. Bas, T. Filler, and T. Pevný, "Break Our Steganographic System: The Ins and Outs of Organizing BOSS," in *Information Hiding (IH 2011), Lecture Notes in Computer Science, vol. 6958*, Prague, Czech Republic, 2011, pp. 59–70. doi: 10.1007/978-3-642-24178-9_5.