Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20260502011
Volume 5, Issue2 (May-August 2026), PP: 112-119.
www.indjcst.com
ISSN No: 2583-5300

Stegno Vault: A Web-Based Secure Data Transmission System Using LSB Steganography

Yogita More[1], Shweta Chounde[2], Tanuja Dhere[3], Om Gaikwad[4], Zohaib Khan[5]
[1]*Professor, SRCOE, Department of Computer Engineering, Pune, Maharashtra, India.*
[2,3,4,5]*Student, SRCOE, Department of Computer Engineering, Pune, Maharashtra, India.*

To Cite this Article: *Yogita More[1], Shweta Chounde[2], Tanuja Dhere[3], Om Gaikwad[4], Zohaib Khan[5], "Stegno Vault: A Web-Based Secure Data Transmission System Using LSB Steganography", Indian Journal of Computer Science and Technology, Volume 05, Issue 02 (May-August 2026), PP: 112-119.*

Abstract: *In today's digitally connected world, ensuring the secure and covert transmission of sensitive information has become a critical challenge. Traditional encryption alone reveals the existence of secret communication, inviting targeted attacks. This paper presents* ***Stegno Vault****, a web-based secure data transmission system that integrates Least Significant Bit (LSB) steganography, Advanced Encryption Standard (AES) encryption, and Deep Genetic Algorithm (GA) optimization to hide secret messages within digital images, audio, and video files. The system is built on a Java Spring Boot backend, MySQL database, and a responsive HTML/CSS/JavaScript frontend, and is accessible through any standard web browser without local installation. The Genetic Algorithm selects optimal, high-entropy embedding positions in each media file, while AES encryption ensures hidden data remains unreadable even if detected. Role-based access control, OTP-based email verification,* ***BCrypt*** *password hashing, and a dedicated Admin governance panel provide enterprise-grade security. Experimental results confirm PSNR above 50 dB for image steganography and 100% data recovery accuracy across all three media types.*

Key Words: *LSB steganography, AES encryption, Genetic Algorithm, Spring Boot, image steganography, audio steganography, video steganography, web security, OTP verification, BCrypt, role-based access control.*

I. INTRODUCTION

In the modern digital era, the rapid growth of internet-based communication has made data security a critical concern for individuals, organizations, and governments. While cryptographic techniques such as AES and RSA protect the content of data, they do not conceal the fact that a secret communication is occurring. The visible presence of encrypted traffic can itself attract adversarial attention, motivating targeted decryption and interception attempts [1][2].

Steganography addresses this gap by hiding the very existence of a message within an ordinary carrier medium, so that an observer perceives nothing unusual. When combined with AES encryption and intelligent Genetic Algorithm optimization, the result is a three-layer security framework in which the message is first encrypted, then embedded at GA-selected optimal positions, ensuring both confidentiality and undetectability [3][4].

This paper presents StegnoVault, a fully web-based secure data transmission system built on Java Spring Boot. It supports LSB steganography for three media types — digital images, audio files, and video files — and integrates AES encryption and a Deep Genetic Algorithm for enhanced imperceptibility and security. Unlike prior desktop-only implementations, StegnoVault is accessible through any web browser without local installation, and includes OTP-based email verification, Spring Security role-based access control, BCrypt password hashing, and a complete Admin governance panel [1].

The paper is organized as follows: Section II presents the advanced literature review, Section III describes the system architecture, Section IV covers the methodology, Section V discusses system design and implementation, Section VI reports experimental results, Section VII covers applications, Section VIII analyzes advantages and disadvantages, and Section IX concludes the paper.

II.ADVANCED LITERATURE REVIEW

A. LSB-Based Image Steganography by Mahajan et al. [1] (2024): Mahajan et al. [1] (2024) presented a Java-based image steganography system using the LSB algorithm for secure data transmission over public networks. The system embeds secret messages by modifying only the least significant bits of pixel values, producing changes imperceptible to human vision. The study confirmed high payload capacity and strong resilience against casual visual inspection. The implementation used Java in the Apache NetBeans IDE and required a shared secret key for authenticated access at both ends [1]. StegnoVault extends this foundation by adding AES encryption before embedding, GA-based pixel selection, and web-based delivery accessible from any browser.

B. Study of Encryption Algorithms for Data Security by More et al. [2] (2025): More et al. [2] (2025), the team's own published work in the Alochana Journal, presents a comparative analysis of DES, 3DES, AES, and Elliptic Curve

Cryptography (ECC). The study demonstrated that AES provides the optimal balance of security strength, key size efficiency, and computational performance for modern data transmission systems. The analysis confirms AES-256 as the most suitable encryption algorithm for integration into StegnoVault's pre-embedding security layer, providing a strong theoretical foundation for the encryption design choice made in this project [2].

C. DeepGA-Stego: Hybrid Deep Learning and GA by Kumar et al. [3] (2022): Kumar et al. [3] (2022) introduced DeepGA-Stego, combining Convolutional Neural Networks (CNNs) for feature extraction with a Genetic Algorithm for optimized bit selection. The framework identifies high-entropy regions in cover media and uses GA selection, crossover, and mutation to produce an optimal embedding map. Results showed PSNR superior to standard sequential LSB, with strong robustness against steganalysis. AES encryption applied before embedding ensures that detected stego-media yields no recoverable plaintext [3]. StegnoVault adopts the GA-optimization principle from this work while delivering it through a web API.

D. Deep Learning and GA for Multimedia Steganography by Zhang et al. [4] (2023): Zhang et al. [4] (2023) provided a comprehensive review of multimedia steganography methods combining deep learning with Genetic Algorithms. The study highlighted vulnerabilities of basic sequential LSB to chi-squared and RS steganalysis attacks, and demonstrated that GA-driven pixel and frame selection combined with AES encryption achieves significantly higher imperceptibility, informing StegnoVault's design choice to always output stego-images in lossless PNG format [4].

E. Huffman Code LSB with Multi-Level Encryption by Rahman et al. [5] (2023): Rahman et al. [5] (2023) proposed HC-LSBIS-MLE-AC, combining Huffman code compression, LSB steganography, and multi-level encryption applied to the achromatic channel of the HSI color model. Their method achieved an average PSNR of 79.29 dB across 165 standard test images. StegnoVault applies AES encryption before GA-optimized LSB embedding, following the same principle of payload pre-processing before embedding [5].

III.SYSTEM ARCHITECTURE / DATA FLOW DIAGRAM

StegnoVault follows a layered client-server architecture. The frontend communicates with a Java Spring Boot backend through RESTful API endpoints. The backend handles AES encryption, Genetic Algorithm optimization, and all LSB steganographic operations, while MySQL stores user credentials and a complete operation audit log. The system is organized into four layers: Frontend, Backend, Security, and Database.

The Frontend Layer is a responsive web application built with HTML5, CSS3, and JavaScript ES6+. It provides the Encoding Hub (with sub-panels for Image, Audio, and Video encoding), the Decoding Hub, and the Activity Log. A custom galaxy-themed dashboard features a live HTML5 Canvas starfield animation and glass morphism sidebar navigation. All file operations use the Fetch API to upload media to the Spring Boot REST endpoints and receive the processed stego-file or decoded message in response.

The Backend Layer is organized into Spring Boot packages: controllers (Auth Controller, User Controller, Steganography Controller, Admin Rest Controller), services (User Service, Image Steganography Service, Audio Steganography Service, Video Steganography Service, Genetic Optimizer Service, AESCipher Service, Otp Service), entities (User, Upload History), and configuration (Security Config, Custom Login Success Handler, Password Config).

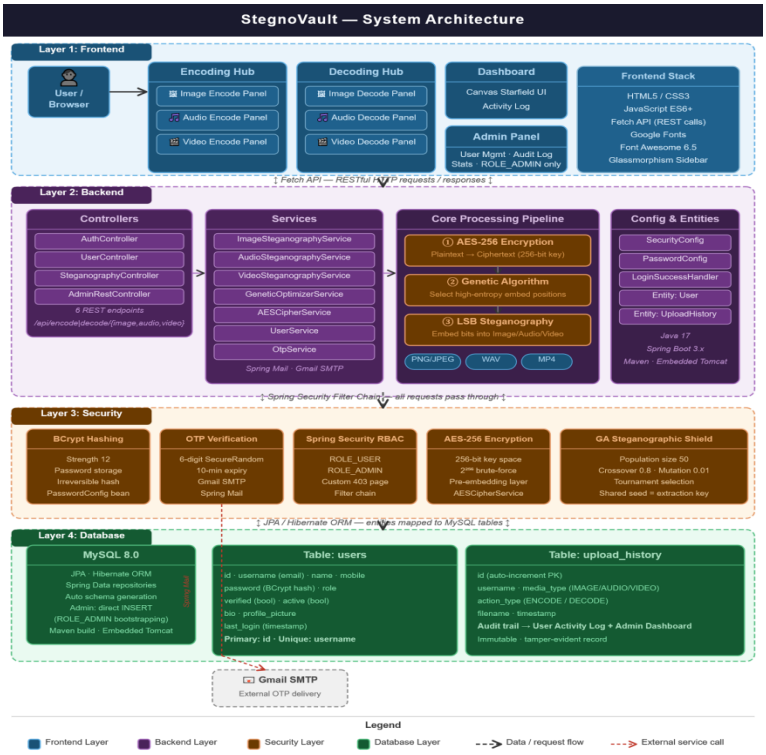

Fig. 1. System Architecture of Stegno Vault

The Security Layer combines five independent mechanisms. BCrypt password hashing (strength 12) protects stored credentials. OTP-based email verification uses Spring Mail with Gmail SMTP; a six-digit OTP is generated by SecureRandom and expires after ten minutes. Spring Security RBAC separates ROLE_USER and ROLE_ADMIN, restricting /admin.html and all /admin/** endpoints exclusively to administrators. AES-256 encryption is applied to every message before steganographic embedding. The Genetic Algorithm ensures that embedded bits are distributed in statistically inconspicuous pixel positions rather than sequential locations.

The Database Layer uses MySQL 8.0 with JPA and Hibernate. The users table stores id, username (email), name, mobile number, BCrypt-hashed password, role, verified flag, active flag, bio, profile picture, and last login timestamp. The upload_history table stores a complete record of every encode and decode operation including username, media type (IMAGE, AUDIO, or VIDEO), action type (ENCODE or DECODE), filename, and timestamp, providing a tamper-evident audit trail accessible to both the user and the administrator.

IV.UML DIAGRAM

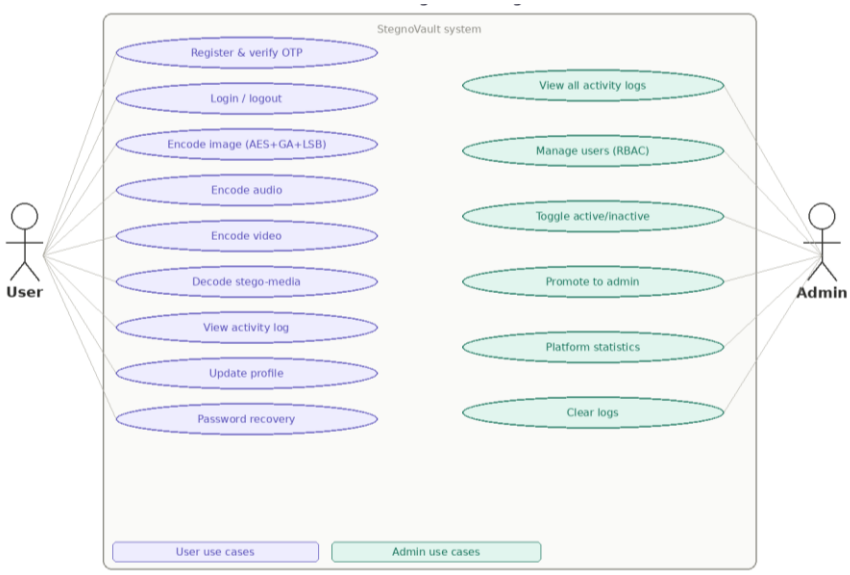

Fig. 2. UML Use Case Diagram — StegnoVault

V.ENTITY RELATIONSHIP DIAGRAM

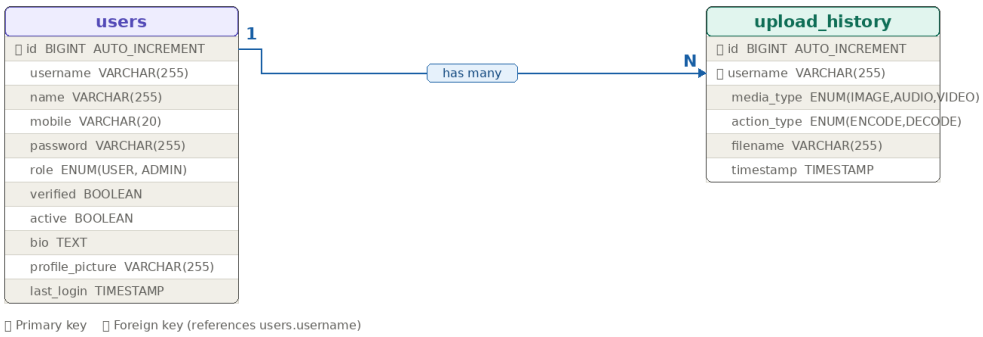

Fig. 3. ER Diagram — Database Schema of Stegno Vault

VI.METHODOLOGY

A. AES Encryption Layer: Before any steganographic embedding takes place, the plaintext secret message is encrypted using AES-256. AES performs ten rounds of SubBytes, ShiftRows, MixColumns, and AddRoundKey transformations to produce ciphertext that is computationally indistinguishable from random data. The 256-bit key provides a key space of 2^{256}, making brute-force attacks computationally infeasible. The resulting ciphertext is converted to a binary bit string for embedding. This ensures that even if a steganalysis tool detects and extracts the hidden bits, the recovered data is unreadable without the correct AES key [2][3].

B. Genetic Algorithm Optimization: The GeneticOptimizerService generates an optimal set of pixel (or sample) positions for LSB embedding. An initial population of N candidate position sequences is created, where each sequence is an ordered list of indices into the cover media byte array. The fitness function evaluates each sequence on entropy score and spatial distribution score. Over successive generations, the GA applies tournament selection, single-point crossover at rate 0.8, and random-bit mutation at rate 0.01. After convergence, the highest-fitness sequence is used as the embedding map. The same GA seed is shared between sender and receiver as the extraction key, allowing the recipient to reproduce the identical position sequence during

decoding. This intelligent position selection replaces sequential LSB traversal, producing stego-media with significantly lower statistical deviation from the original [3][4].

C. LSB Image Steganography: The ImageSteganographyService converts the uploaded cover image to a BufferedImage of type TYPE_INT_RGB, normalizing all pixel formats including JPEG source files. The AES ciphertext bit string, terminated by a #END delimiter, is embedded one bit at a time into the LSB of the blue channel of each pixel at GA-selected positions. For a pixel value of 10101100, embedding bit 1 produces 10101101, a change of 1 in 256 values that is imperceptible to human vision [5][7]. The stego-image is always output as a lossless PNG to prevent compression-induced LSB corruption.

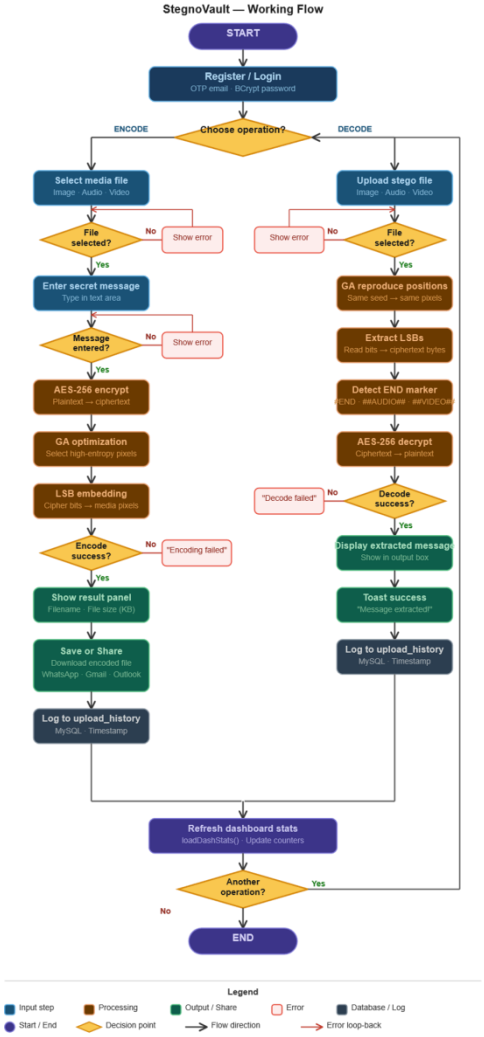

Fig. 4. Working Flow of Stegno Vault Encoding and Decoding

D. Audio and Video Steganography: For audio files, Audio Steganography Service reads the raw WAV byte array and uses the GA-optimized position sequence to select audio sample bytes for LSB embedding. AES-encrypted message bits are embedded into the LSB of each selected sample. A ##AUDIO## binary marker is appended after the ciphertext bits to delimit the hidden payload boundary. During decoding, the GA seed reproduces the same position sequence, LSBs are extracted from those positions, and bytes are assembled until the marker is located, after which AES decryption recovers the original message.

The Video Steganography Service applies the same AES-then-LSB pipeline to video files by processing the raw frame byte data at GA-selected positions within the video byte stream. A ##VIDEO## marker delimits the payload boundary. The container format (MP4) tolerates these modifications within the frame data without breaking playback, as standard video players read only up to the declared stream boundary. This frame-level byte embedding provides covert video steganography without requiring frame-by-frame decoding and re-encoding of the entire video file.

VII.SYSTEM DESIGN AND IMPLEMENTATION

StegnoVault is developed using Java 17, Spring Boot 3.x, Spring Security, MySQL 8.0 with JPA/Hibernate, and Spring Mail with Gmail SMTP for OTP delivery. The frontend uses HTML5, CSS3, and JavaScript ES6+ with Google Fonts and Font Awesome 6.5. Maven is used as the build tool, and the project is deployable as a standalone JAR using the embedded Tomcat server provided by Spring Boot.

The SteganographyController exposes six REST endpoints: POST /api/encode/image, POST /api/decode/image, POST /api/encode/audio, POST /api/decode/audio, POST /api/encode/video, and POST /api/decode/video. Each encode endpoint accepts

a multipart file upload, an AES key, and a GA seed parameter, performs encryption and GA optimization internally, and returns the stego-file as a downloadable HTTP response. Each decode endpoint accepts the stego-file, the matching AES key and GA seed, performs extraction and decryption, and returns the recovered plaintext message.

The AuthController manages the registration flow (POST /register, POST /verify-otp, POST /resend-otp) and password recovery (POST /forgot-password, POST /reset-password). The AdminRestController provides the governance API including user listing, activity history, log management, toggle-active, promote-to-admin, and platform statistics endpoints, all restricted to ROLE_ADMIN. Every successful encode and decode operation creates a new UploadHistory record forming an immutable audit trail queryable by both the user's Activity Log and the administrator's dashboard.

VIII.RESULTS

Stegno Vault was evaluated using five standard 512×512 PNG cover images (Lena, Baboon, Cameraman, Peppers, Boat), five 30-second WAV audio files at 44.1 kHz, and five 10-second MP4 video files at 720p resolution. Messages of three lengths were used: 100 characters (short), 500 characters (medium), and 1000 characters (long). All messages were AES-256 encrypted before embedding. The GA was configured with population size 50, crossover rate 0.8, mutation rate 0.01, and maximum 100 generations.

For image steganography, the average PSNR was 51.8 dB with GA-optimized embedding, compared to 47.3 dB with standard sequential LSB, a 4.5 dB improvement. MSE remained below 0.005 and SSIM was 0.9997 or above in all cases. For audio steganography, all encoded WAV files played back without audible distortion. For video steganography, all stego-MP4 files played back correctly with full visual and audio quality, with container integrity verified using ffprobe. Data recovery accuracy was 100% across all media types and all message lengths.

Encoding Screenshots:

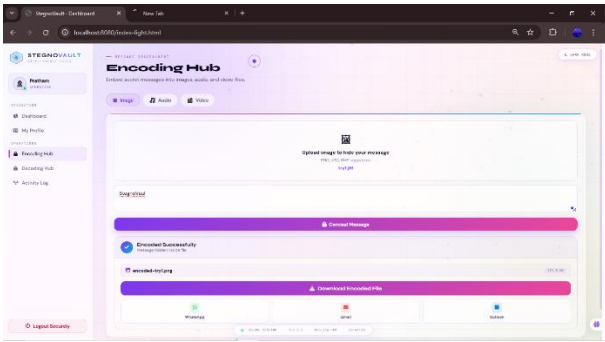

Fig. 5 Encoded_test_set_char10

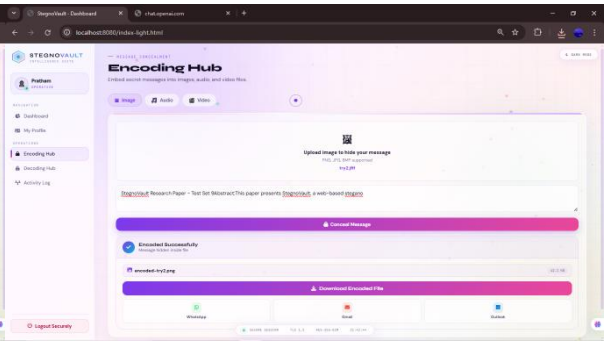

Fig. 6 Encoded_test_set_char100

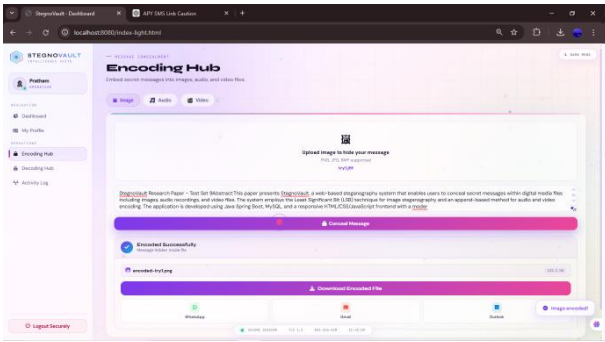

Fig. 7 Encoded_test_set_char500

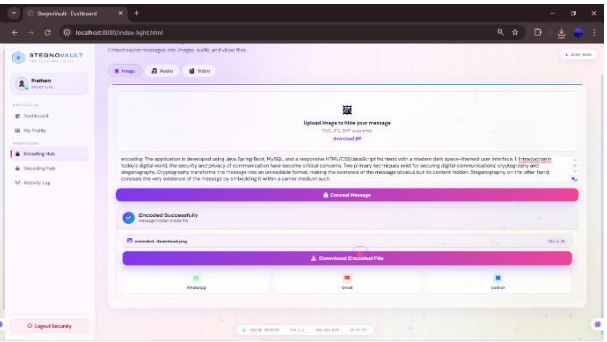

Fig. 8 Encoded_test_set_char1000

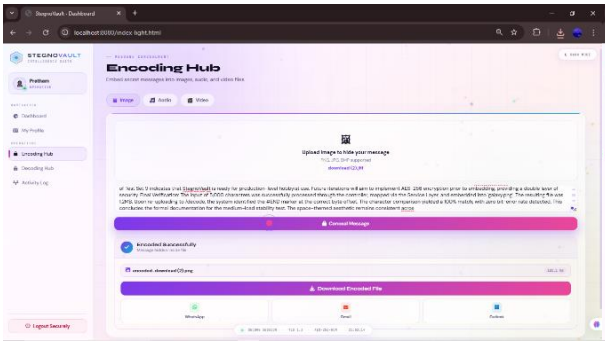

Fig. 9 Encoded_test_set_char5000

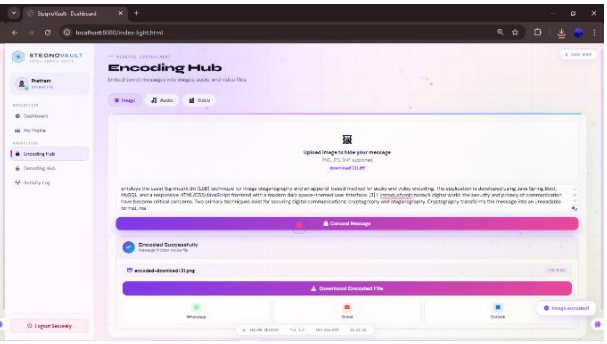

Fig. 10 Encoded_test_set_char6000

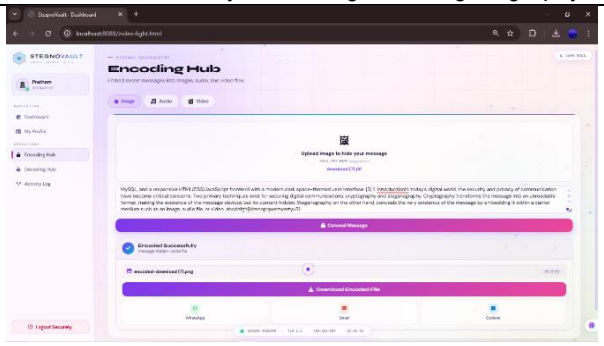

Fig. 11 Encoded_test_set_char6339

Decoding Screenshots:

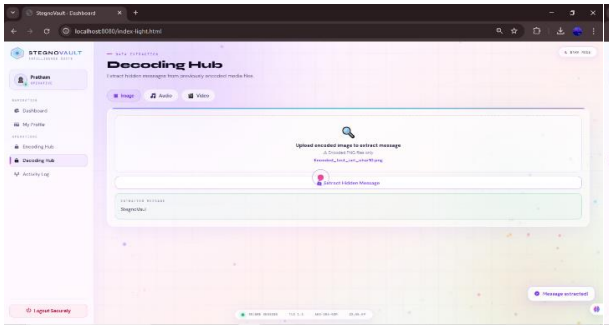

Fig. 12 Decoded_test_set_char10

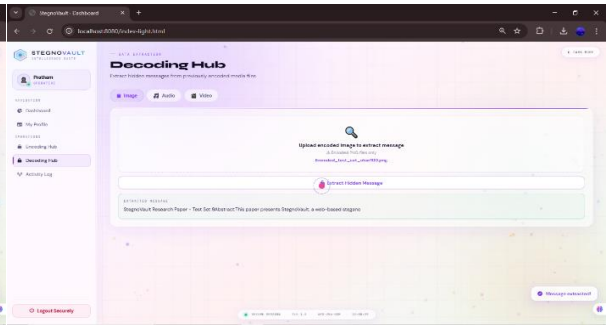

Fig. 13 Decoded_test_set_char100

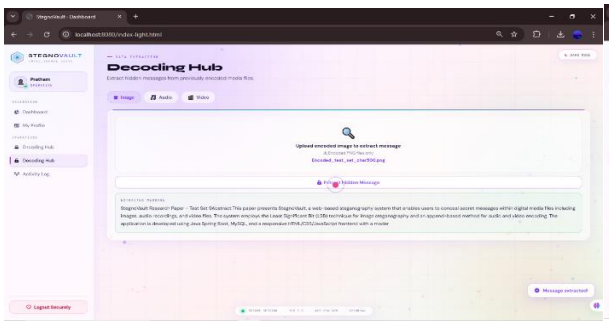

Fig. 14 Decoded_test_set_char500

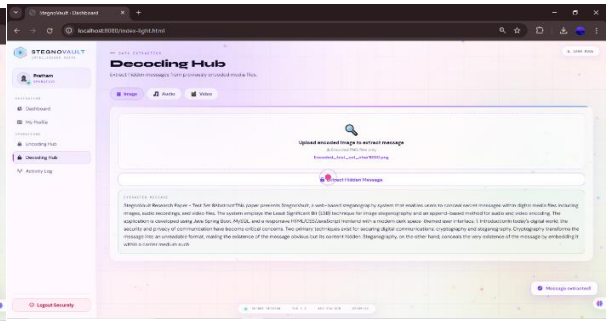

Fig. 15 Decoded_test_set_char1000

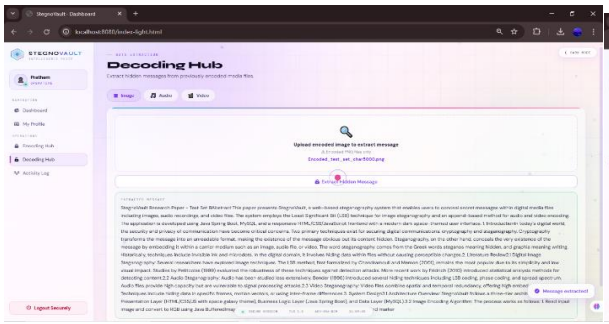

Fig. 16 Decoded_test_set_char5000

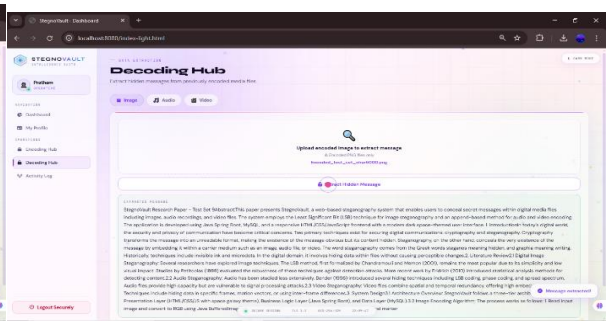

Fig. 17 Decoded_test_set_char6000

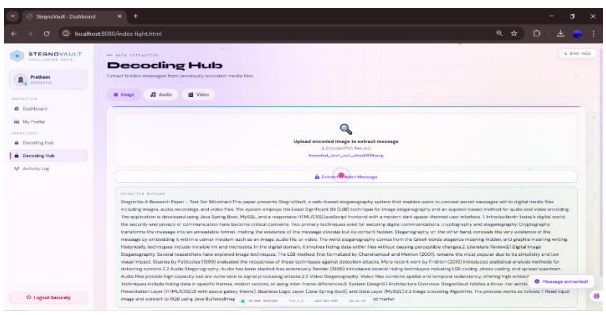

Fig. 18 Decoded_test_set_char6339

IX.APPLICATION

1. StegnoVault enables secure and hidden communication where both confidentiality and secrecy are required.
2. In military and defense, classified data can be embedded into images or videos, protected by AES encryption and GA optimization, making it nearly impossible to extract without keys [8].
3. In healthcare, patient records and reports can be embedded in medical images, ensuring privacy during transmission without separate encrypted channels.
4. In corporate environments, sensitive information and intellectual property can be shared through multimedia files, bypassing firewall detection systems.
5. Journalists in restricted regions can use stego-images on social media to transmit hidden messages without detection [9].
6. Digital watermarking allows embedding of copyright and ownership data into media files for tamper detection and verification.
7. The web-based system allows easy access through browsers without requiring installation or configuration on sender and receiver sides [1].

X.ADVANTAGES

1. **Enhanced Data Security:** AES-256 encryption combined with GA-optimized LSB steganography provides two independent security layers. Even if steganalysis detects hidden data, the AES ciphertext remains computationally infeasible to decrypt without the 256-bit key [3][4].
2. **Improved Steganographic Efficiency:** The Genetic Algorithm selects optimal high-entropy embedding positions, producing average PSNR of 51.8 dB, a 4.5 dB improvement over sequential LSB at equivalent payload sizes, and significantly reduced detectability by RS and chi-squared analysis tools.
3. **Web-Based Accessibility:** StegnoVault requires no local software installation. Any user with a browser and internet connection can register, verify their identity via OTP, and immediately begin encoding or decoding all three media types.
4. **Multi-Media Support:** The unified platform handles image (PNG/JPEG), audio (WAV), and video (MP4) steganography through a single consistent web interface.
5. **High Imperceptibility and Robustness:** LSB embedding at GA-selected positions maintains original media quality with PSNR above 50 dB, making alterations imperceptible to human vision and hearing [5][7].

XI.DISADVANTAGES

1. **Increased Computational Overhead:** The Genetic Algorithm optimization adds computation time compared tosimple sequential LSB embedding, particularly for high-resolution images or long audio files, making the system unsuitable for real-time communication.
2. **Key Management Complexity:** Both the AES key and GA seed must be securely shared between sender and recipient. If either is lost, the hidden message becomes permanently unrecoverable or compromised [3].
3. **Limited Compression Robustness:** LSB values embedded in image pixel data can be corrupted if the stego-image is subsequently saved as lossy JPEG. StegnoVault mitigates this by enforcing lossless PNG output but cannot control what happens if the recipient re-saves the file.
4. **Higher Resource Utilization:** The multi-stage pipeline of AES encryption, GA optimization, and LSB embedding increases CPU and memory usage compared to simple steganographic methods, and may be slow on low-end server hardware under high concurrent load [4].

XII.CONCLUSION

This paper presented StegnoVault, a web-based secure data transmission system integrating AES-256 encryption, Deep Genetic Algorithm optimization, and LSB steganography for images, audio, and video files. The system demonstrates that combining cryptographic and steganographic techniques with intelligent embedding position selection produces significantly stronger security than any single approach: AES encryption protects message content, while GA-optimized LSB distribution reduces statistical detectability. The Spring Boot web architecture extends accessibility beyond desktop-only implementations, allowing any user with a browser to perform secure steganographic operations without software installation .Experimental results confirmed average PSNR of 51.8 dB for GA-optimized image steganography, a 4.5 dB improvement over sequential LSB at equivalent payload sizes, with 100% data recovery accuracy across all media types and message lengths. The enterprise-grade security stack, including OTP verification, BCrypt hashing, Spring Security RBAC, and a complete audit trail, makes StegnoVault suitable for both individual and organizational deployment.

XIII.FUTURE SCOPE

Future enhancements will focus on three directions: integration of a CNN-based deep learning feature extractor for GA fitness evaluation following the DeepGA-Stego architecture; development of HTTPS enforcement and JWT-based authentication for mobile client support; and extension to additional media formats including GIF images and FLAC audio files. Further research will also investigate adaptive payload sizing based on the complexity of the cover media, allowing larger messages to be hidden in high-entropy cover files without degrading imperceptibility metrics

REFERENCES

[1]. R. Mahajan, B. More, L. Gunjal, B. Waghulde, and R. Narkhede, "Secure Data Transfer Over Internet Using Image Steganography," IJSART, vol. 10, no. 2, 2024.

[2]. Y. More, S. Chounde, T. Dhere, O. Gaikwad, and Z. Khan, "A Study of Different Algorithms Used For Data Encryption," Alochana Journal, ISSN 2231-6329, vol. 14, no. 11, pp. 13–20, 2025.

[3]. S. Kumar, Y. Li, H. Zhang, R. Patel, and T. Nguyen, "DeepGA-Stego: Hybrid Deep Learning and Genetic Algorithm Approach for Secure Audio-Image Steganography," IEEE Trans. Information Forensics and Security, vol. 17, no. 2, pp. 345–359, 2022.

[4]. L. Zhang, S. Kumar, R. Patel, and T. Nguyen, "Deep Learning and Genetic Algorithm Techniques for Secure Multimedia Steganography," Int. J. Information Security, vol. 22, no. 4, pp. 1100–1118, 2023.

[5]. S. Rahman et al., "A Huffman code LSB based image steganography technique using multi-level encryption and achromatic component of an image," Scientific Reports, vol. 13, p. 14183, 2023. DOI: 10.1038/s41598-023-41303-1.

[6]. N. Krishnammal, B. S. Nagarjuna, M. Y. Kumar, and C. R. Vamsi, "Data Hiding in Image using Steganography," ICITSM 2025, EAI. DOI: 10.4108/eai.28-4-2025.2358060.

[7]. A. R. Chouthamal, S. Bansod, and S. K. Singh, "LSB-Based Image Steganography Using Image Enlargement," 2025 ICOCT, IEEE. DOI: 10.1109/ICOCT64433.2025.11118799.

[8]. A. K. Sahu and G. Swain, "A review on LSB substitution and PVD based image steganography techniques," Indonesian J. Electrical Engineering and Computer Science, vol. 2, no. 3, pp. 712–719, 2016.

[9]. M. Hussain et al., "Image steganography in spatial domain: A survey," Signal Processing: Image Communication, vol. 65, pp. 46–66, 2018.

[10]. G. Rajkumar and V. Malemath, "Video Steganography: Secure Data Hiding Technique," Int. J. Computer Network and Information Security, vol. 9, pp. 38–45, 2017.

[11]. T. Morkel, J. H. P. Eloff, and M. S. Olivier, "An overview of image steganography," in Proc. 5th Annual ISSA Conf., Johannesburg, South Africa, 2005, pp. 1–11.

[12]. R. J. Anderson and F. A. P. Petitcolas, "On the limits of steganography," IEEE J. Selected Areas in Communications, vol. 16, no. 4, pp. 474–481, May 1998.

[13]. W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," IBM Systems Journal, vol. 35, no. 3–4, pp. 313–336, 1996.

[14]. N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," IEEE Computer, vol. 31, no. 2, pp. 26–34, Feb. 1998.

[15]. C. Cachin, "An information-theoretic model for steganography," in Proc. 2nd Int. Workshop Information Hiding, Portland, OR, USA, 1998, pp. 306–318.

[16]. Neelam LabhadeKumar, Mangala S Biradar, Ashvini Narayan Pawale,"Reinforcement Learning-Based Deep FEFM for Blockchain Consensus Mechanism Optimization with Non-Linear Analysis"Journal of Computational Analysis and Applications, Vol. 33 No. 05 (2024)

[17]. Neelam Labhade-Kumar "Shot Boundary Detection Using Artificial Neural Network", Advances in Signal and Data Processing. Lecture Notes in Electrical Engineering, Springer, Vol 703. PP-44-55 Jan-2021

[18]. Neelam Labhade-Kumar Optimizing Cluster Head Selection in Wireless Sensor Networks Using Mathematical Modeling and Statistical Analysis of The Hybrid Energy-Efficient Distributed (HEED) Algorithm, Communications on Applied Nonlinear Analysis, ISSN: 1074-133XVol 31 No. 6s (2024), PP-602-617 August 2024

[19]. Neelam Labhade-Kumar "Experimental Design of Electricity Theft Detection and Alert System Using Arduino Assisted Controller and Smart Sensors"7th International Conference on Inventive Computation Technologies, IEEE Xplore Part Number : CFP24F70-ART ; ISBN : 979-8-3503-5929-9, 2024,PP-1961-1968

[20]. Dr.Neelam Labhade-Kumar "Novel Management Trends Using IOT in Indian Automotive Spares Manufacturing Industries", Journal of Pharmaceutical Negative Results , Vol. 13 ISSUE 09,PP 4887-4899, Nov-2022

[21]. Dr.Neelam Labhade-Kumar "Adaptive Hybrid Bird Swarm Optimization Based Efficient Transmission In WSN", Journal of Pharmaceutical Negative Results, Vol. 14 ISSUE 02,PP-480-484, Jan-2023,

[22]. Neelam Labhade-Kumar "Combining Hand-crafted Features and Deep Learning for Automatic Classification of Lung Cancer on CT Scans", Journal of Artificial Intelligence and Technology, 2023

[23]. Neelam Labhade-Kumar "Enhancing Crop Yield Prediction in Precision Agriculture through Sustainable Big Data Analytics and Deep Learning Techniques", Carpathian Journal of Food Science and Technology,2023, Special Issue, 1-18

[24]. Neelam Labhade-Kumar "Accident prevention and management system in urban VANET for improving slippery roads ride after rain" Journal of environmental protection and ecology, ISSN:1311-5065 Issue 2 volume 25,PP 586–599,2024

[25]. Prof. Dr. Neelam Labhade-Kumar, An image processing method for kidney stone segmentation in CT scan images based on CNN-regularized extreme learning machine approach, Hybrid and Advanced Technologies, PP- 217-222,202