Genetic Algorithm Based S-Box Optimization for AES

Ahmed Haruna Dokoro
Computer Science Department
Gombe State Polytechnic, Bajoga
Gombe, Nigeria
https://orcid.org/0000-0003-1675-5857

Raji Abdullahi Egigogo
Department of Software Engineering and Cyber Security
Al-Qalam University Katsina
Katsina, Nigeria
rajiaegigogo@auk.edu.ng

Hassan Ibrahim
Computer Science Department
FederalPolytechnic Kaltungo
Gombe, Nigeria
hassangsu@gmail.com

Bilyaminu Musa Gadam
Computer Science Department
Gombe State Polytechnic, Bajoga
Gombe, Nigeria
mbilyaminugadam@gmail.com

***Abstract*— The Advanced Encryption Standard (AES) remains one of the most widely adopted symmetric encryption algorithms in use today. A core component of its cryptographic strength lies in its substitution box (S-box), which performs non-linear transformations to resist common attacks. However, the static and publicly known nature of the AES S-box leaves room for cryptanalytic improvements. This paper introduces an AI-driven approach to dynamically generate optimized S-boxes using Genetic Algorithms (GA). By evolving populations of candidate S-boxes based on cryptographic criteria such as non-linearity, avalanche effect, and differential uniformity, the method aims to enhance resistance against differential and linear cryptanalysis. Comparative analysis with the standard AES S-box demonstrates improved cryptographic properties and increased variability, pointing toward a promising direction for adaptive and secure encryption systems. The proposed GA-optimized S-box demonstrated maintained the differential uniformity of 4, 1.8% increase in nonlinearity and an improvement in the Strict Avalanche Criterion (SAC) to 99.4% when compared with the standard AES S-box.**

***Keywords*—AES, S-Box optimization, Genetic Algorithm, Cryptography, Encryption**

I. INTRODUCTION

In an era where digital communication and data exchange are integral to daily life, the demand for robust and reliable encryption mechanisms has never been higher. In 2001, the Institute of Standards and Technology (NIST) certified the Advanced Encryption Standard (AES) as one of the most trusted symmetric-key encryption algorithms. It is widely employed in securing sensitive information across industries, governments, and individual users. At the heart of AES lies the substitution box (S-box), a nonlinear component responsible for introducing confusion into the cipher, thereby playing a crucial role in thwarting cryptanalytic attacks [1].

A. AES Algorithm

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm widely used to secure data. It was adopted by the U.S. National Institute of Standards and Technology (NIST) in 2001 as a replacement for the older Data Encryption Standard (DES) due to its stronger security and efficiency. It is a block cipher that operates on 128 bits fixed-size blocks and supports three key lengths of 128, 192, or 256-bit [2].

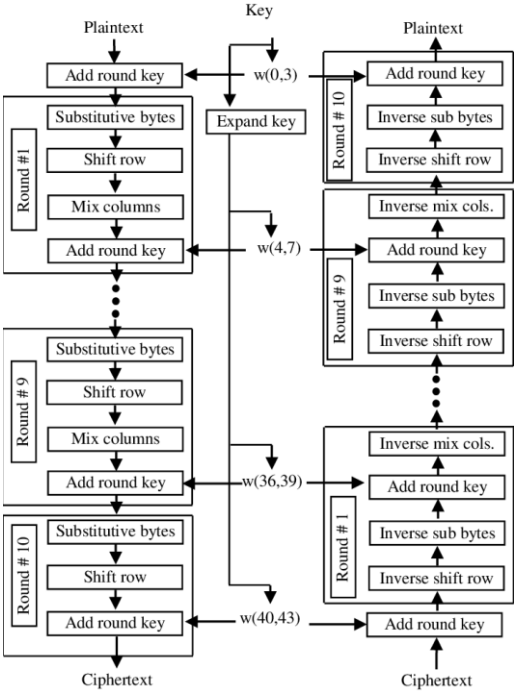

Fig. 1. Block Diagram of AES Algorithm [3]

Despite its effectiveness, the AES S-box is static and publicly known. This feature may render it susceptible to evolving attack strategies, particularly those leveraging statistical and differential analysis [4]. As adversaries develop increasingly sophisticated techniques, it becomes imperative to explore adaptive and intelligent methods to fortify existing encryption schemes [5].

B. Genetic Algorithm

Genetic Algorithm (GA) is a search and optimization technique inspired by the process of natural selection and genetics. It belongs to the family of evolutionary algorithms and is used to solve complex problems by evolving solutions over generations. The algorithms rely on the following key concepts:

- Population: A set of potential solutions
- Chromosome: A representation of a solution
- Fitness Function: Measures how good a solution is relative to the problem.
- Selection: Chooses the best individuals based on fitness to reproduce.
- Crossover (Recombination): Combines parts of two parents to create offspring.
- Mutation: Introduces random variations to maintain diversity in the population.
- Generations: Iterative cycles where selection, crossover, and mutation are applied.

Genetic Algorithms have proven effective in solving complex optimization problems due to their ability to explore large solution spaces and evolve optimal candidates through natural selection principles [6]. In this context, we utilize GAs to evolve S-boxes that satisfy stringent cryptographic properties such as high non-linearity, low differential uniformity, and a strong avalanche effect [7]. These AI-generated S-boxes are intended to replace the static S-box in AES, resulting in an encryption scheme that is not only more secure but also capable of adapting to different cryptographic scenarios.

Recent studies have highlighted a growing shift toward AI-driven cryptography, where artificial intelligence and evolutionary algorithms are employed to enhance cryptographic design and analysis. A comprehensive survey conducted by [8] shows how AI techniques such as machine learning and generative adversarial networks (GANs) are being applied to both cryptanalysis and cryptographic design, for instance automating high-quality S-box generation. A recent study specifically uses a hybrid of GA and GANs for cryptographic tasks, demonstrating the optimization power of evolutionary methods in encryption systems [9].

Researchers have successfully applied machine learning, genetic algorithms, and hybrid GA–GAN frameworks to automate S-box generation, improve nonlinearity and diffusion, and strengthen resistance against cryptanalysis. These works collectively demonstrate the effectiveness of AI and evolutionary computation in optimizing cryptographic primitives, reinforcing the relevance of the proposed GA-based S-box optimization in this paper.

This paper proposes a novel approach to enhancing AES security by introducing dynamic S-box generation using Genetic Algorithms (GAs), a class of biologically inspired artificial intelligence techniques [10]. GA was selected over other heuristics due to its superior global search ability, sustained population diversity, avoidance of local optima and effective balance between exploration and exploitation, which are crucial for navigating the complex, non-linear search space of cryptographic optimization to achieve high nonlinearity and strong avalanche properties. The paper presents the design, implementation, and evaluation of the proposed method. A Python-based prototype was developed to generate and evaluate S-boxes based on their cryptographic fitness. Experimental results are compared with the standard AES S-box to demonstrate the feasibility and advantages of the approach. The findings suggest that the integration of AI into symmetric encryption algorithms can significantly improve their resilience, offering new avenues for research in adaptive cryptographic systems.

This study aims to strengthen the AES algorithm by developing an optimized S-box using GA. The proposed approach focuses on enhancing key cryptographic properties such as nonlinearity and the Strict Avalanche Criterion (SAC. The contributions include the development of a GA-based framework for S-box optimization, performance evaluation against the standard AES S-box, and validation of GA's effectiveness as an AI-assisted technique for improving resistance to differential and linear cryptanalysis.

The remainder of this paper is organized as follows: Section II presents a review of related works. Section II describes the methodology adopted for the development of the GA-based S-box, including the algorithm design, fitness evaluation, and implementation process. Section IV discusses the experimental results and discussion implications of the findings. Finally, Section V concludes the paper and outlines potential directions for future research in AI-assisted cryptographic optimization.

II. RELATED WORKS

The Advanced Encryption Standard (AES), standardized by the National Institute of Standards and Technology (NIST) in 2001 [2], has become the cornerstone of modern symmetric-key cryptography. Its robust design includes multiple rounds of transformation, among which the Sub Bytes step implemented using a substitution box (S-box) is critical in providing non-linearity and resistance to common cryptanalytic techniques. The standard AES S-box is derived from multiplicative inverses in the finite field GF(2^8), followed by an affine transformation [2]. Although mathematically sound, this static design is fixed and deterministic, making it a predictable target for cryptanalysts [4].

A. Cryptographic Properties of S-Boxes

Effective S-boxes must meet several cryptographic criteria to ensure security:

- Non-linearity: Ensures resistance against linear cryptanalysis [11].
- Differential Uniformity: A lower uniformity value increases resistance to differential attacks [9].
- Strict Avalanche Criterion (SAC): A single input bit change should result in a 50% chance of each output bit flipping [1].
- Bit Independence Criterion (BIC): Output bits should change independently in response to input bit changes [12].

Researchers have shown that altering or generating new S-boxes can potentially enhance the robustness of encryption schemes, particularly in resource-constrained or high-security environments [13].

B. AI and Machine Learning in Cryptography

The integration of artificial intelligence in cryptography is a growing research domain. Several studies have explored the use of machine learning (ML) and evolutionary algorithms to improve various cryptographic components:

- Genetic Algorithms (GA) have been employed in tasks such as optimizing key schedules, cryptanalysis, and, more relevantly, generating substitution boxes with improved cryptographic properties [14].
- Neural networks have been investigated for generating pseudo-random sequences and dynamic S-boxes, though they often face interpretability and complexity challenges [15].
- Other AI models, such as Particle Swarm Optimization and Ant Colony Optimization, have also been tested for S-box generation with varying success [16].

C. Prior Studies on S-Box Modification

In recent years, researchers have proposed various heuristic approaches to dynamically generate S-boxes. For instance, [17] employed genetic algorithms to evolve S-boxes optimized for avalanche effect and non-linearity. Similarly, [18] applied GA to optimize Boolean functions used in block ciphers.

[19] suggests a novel method for dynamically generating S-Boxes, which will increase the intricacy of S-Box design to withstand any potential attack on the fixed S-Box. Attackers can use predefined static S-Boxes to examine specific cipher text pairings. In an effort to increase algorithm complexity and make cryptanalysis more difficult, the newly developed S-boxes are also dynamic, unpredictable, and key dependent.

In order to maximize nonlinearity while preserving the bijective condition, [20] proposed a genetic method was proposed. Additionally, the authors developed a new method to enhance the crossover and mutation operators of the genetic algorithm's ability to generate such highly nonlinear S-boxes.

In another study, [11] introduced a novel method for S-box generation and advanced design based on a mixed chaotic system.

A method for producing S-boxes with two critical characteristics of high nonlinearity and poor differential homogeneity which that are essential for cryptographic applications was proposed by [21]. The authors developed a hybrid adaptive evolutionary algorithm to create a bijective S-box that satisfies certain design constraints, such as nonlinearity and differential homogeneity, algebraic ordering, and no fixed points.

To improve the nonlinearity of S-box structures used in cryptography, [22] developed a substitution box (S-box) based on chaotic and evolutionary algorithms. Research suggests that the nonlinear value might be increased to 111.75, which is about the same as the nonlinear value of the AES-S-Box. Moreover, this method establishes the foundation for further research on chaos-based S-box structures and other optimization methods by improving the XOR distribution table, which measures the resilience of differential attacks.

In a different publication, [23] created a new method for generating dynamic S-boxes used in block ciphers. Their technique overcomes the drawbacks of previous chaos-based S-box algorithms by using pseudo-random sequences based on Gaussian distributions in place of chaotic maps. Furthermore, the use of evolutionary techniques improved the nonlinearity of these Gaussian distribution-based S-boxes while preserving a high degree of unpredictability. The performance of this novel approach was evaluated using the standard S-box performance criteria.

[24] presented a novel method for creating S-boxes that combines trigonometric manipulations with a genetic algorithm. S-Boxes introduce nonlinearity, which is essential to cryptographic systems. Several tests were conducted to evaluate the algorithm's efficacy, showing that the developed S-box was resistant to both linear and differential cryptanalysis, providing a viable way to improve the security of cryptographic systems.

In order to create new invertible coefficient matrices for the affine transformation stage of the substitution box (S-box) in the Rijndael encryption, [25] suggested a Genetic Algorithm based method. The goal of this method is to produce new S-boxes with enhanced security and better cryptographic characteristics. The approach reduces vulnerabilities to known plaintext and differential attacks by making greater leaps over the search space, which enables it to converge rapidly and find S-boxes with lower DSAC values. A method for evaluating and optimizing S-box attributes has been successfully used to test and analyze the found S-boxes.

To protect data, [26] created a creative, key-dependent, and dynamic replacement box building technique. This method strengthens the S-Box's cryptographic resilience by using a tweak technique and a special chaotic map. The resulting S-Box is more resistant to security threats since it is dynamic and key-sensitive. Additionally, the study evaluates this S-Box's strength using predetermined standards and compares it to the most advanced S-Boxes, confirming its applicability in the field of cryptography.

In order to produce S-boxes, [27] look at integrating 3D chaotic maps with the whale optimization technique. The main goal is to create S-boxes with better cryptographic qualities by utilizing the whale algorithm's optimization skills in conjunction with the intrinsic unpredictability and intricate dynamics of chaotic maps. The study highlights how these S-boxes are more resilient to different cryptographic threats, which makes them ideal for secure communications.

The use of an elitist strategy within the barnacles mating optimizer framework for S-box creation is the main emphasis of [28]. By improving their design using the barnacles mating algorithm, which replicates the natural mating behavior of barnacles, the objective is to improve the security aspects of S-boxes. The method's contribution to the field of cryptographic optimization approaches is noteworthy, as the research highlights its efficacy in generating high-quality S-boxes that enhance the overall security of encryption systems.

An inventive approach to creating strong nonlinear components for block ciphers is presented by [29] who mix group theory and graphical approaches. By employing mathematical structures and visual representations, the method seeks to create S-boxes with high non-linearity and robust

defense against cryptographic assaults. This approach offers a substitute for evolutionary algorithm-based methods and sheds light on the use of organized mathematical frameworks to improve S-box cryptography.

The goal of [30] is to increase the strength of cryptography by designing safe S-boxes utilizing permutations of the symmetric group. The authors provide a method for creating S-boxes with enhanced non-linearity and defense against cryptographic assaults by utilizing the mathematical characteristics of symmetric groups. By emphasizing the benefits of organized mathematical procedures in S-box design, this method stands out from genetic algorithm-based approaches because to its methodical approach to permutation.

A new encryption technique is proposed by [31] that uses several semi field S-boxes, which are created by permutations of the symmetric group, to improve security. S-boxes with strong cryptographic features are created by utilizing the mathematical characteristics of symmetric groups and semi fields. In order to make the method more secure and sophisticated and more resilient to different types of cryptographic assaults, the study uses numerous S-boxes in the encryption process [32].

While several studies have explored, few studies have specifically targeted AI integration into established standards like AES. Among the few studies, a significant number of implementations rely on continuous-domain algorithms like PSO and ACO, which are less effective in the discrete and highly non-linear search space characteristic of cryptographic structures. Some AI-based designs also lack comparative evaluation against the AES standard, making it difficult to assess their real-world applicability. Furthermore, hybrid or deep-learning approaches often introduce computational overhead, reducing their suitability for lightweight or embedded systems such as IoT devices. Therefore, there remains a need for a lightweight, GA-based optimization framework capable of producing cryptographically strong S-boxes that satisfy multiple security properties simultaneously while maintaining computational efficiency.

III. METHODOLOGY

The methodology adopted for the AI-driven generation of optimized S-boxes using Genetic Algorithms (GAs) is presented in this section. The idea is to develop S-boxes that exhibit strong cryptographic properties: high non-linearity, low differential uniformity, and strong avalanche and bit independence effects.

A. Overview of Genetic Algorithm Approach

Genetic Algorithms are population-based search heuristics inspired by the principles of natural selection and genetics (Goldberg, 1989). They are particularly effective in navigating complex search spaces and discovering near-optimal solutions through a process of iterative refinement.

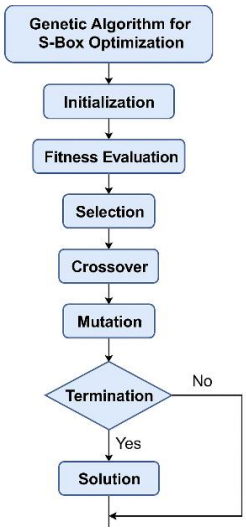

Fig. 2. Overview of Genetic Algorithm Approach

The GA-based approach for S-box optimization involves the following key steps:

Step 1: Initialization: Randomly generate a set of candidate S-boxes ensuring bijection.

Step 2: Fitness Evaluation: Measure nonlinearity, differential uniformity, SAC, and BIC.

Step 3: Selection: Tournament selection to pick fitter individuals.

Step 4: Crossover: Use Partially Mapped Crossover (PMX) to recombine parent S-boxes.

Step 5: Mutation: Apply swap mutation to maintain diversity.

Step 6: Replacement: Form a new population by selecting the best individuals from parents and offspring.

Step 7: Termination: After 500 generations or convergence of fitness values

B. Implementation

The prototype was developed in Python to evolve cryptographically strong S-boxes and integrate them into a modified AES encryption framework for performance validation. The implementation includes the following components:

- S-Box Representation: Each S-box is represented as a one-dimensional array of size 256, containing a unique permutation of the integers 0 to 255.
- Fitness Function: Custom functions were developed to calculate non-linearity, differential uniformity, SAC, and BIC.
- GA Engine: Implements selection, crossover, and mutation operators in modular functions.
- Visualization & Logging: Matplotlib is used to visualize convergence behavior, and logging tracks generation-wise fitness statistics

The prototype system was developed and tested using Python 3.11. The libraries used are numpy, random, matplotlib, and copy on a laptop computer with Intel corei7 vPRO processor and 16GB of RAM running Windows 11.

C. S-Box Representation

To guarantees the property necessary for S-box design in symmetric encryption, each candidate S-box is modelled as a 256-element array, with each element being a unique integer from 0 to 255.

D. Genetic Algorithm Design

The Genetic Algorithm was structured as follows:

Table 1: GA Component Design

GA Component	Implementation Details
Population Size	50 candidate S-boxes
Number of Generations	500
Selection Method	Tournament Selection (size = 3)
Crossover Operator	Partially Mapped Crossover (PMX)
Mutation Operator	Swap Mutation (low probability: 0.02)
Fitness Function	Weighted sum of non-linearity, DU, SAC, and BIC scores

The GA was executed for a maximum of 500 generations or until convergence, defined as no improvement in the best fitness value for 50 consecutive generations.

- Fitness Function: The fitness of each S-box was evaluated based on a weighted combination of critical cryptographic properties:

$$Fitness = w_1 \times NL - w_2 \times DU + w_3 \times SAC + w_4 \times BIC \quad (1)$$

Where:

w_1, w_2, w_3, w_4 are adjustable weights (initially set to 1 for equal importance). All weights in the fitness function were initially set to 1 to assign equal importance to each cryptographic criterion (nonlinearity, SAC, and BIC) during the preliminary optimization phase. This uniform weighting ensures a neutral search direction.

$NL = Nonlinearity$,
$DU = Differential\ Uniformity$,
$SAC = Strict\ Avalance\ Criterion$,
$BIC = Bit\ Independent\ Criterion$

- Crossover and Mutation

Partially Mapped Crossover (PMX) ensures the offspring remains a valid permutation of 0–255 after crossing over parts of two parent S-boxes.

Swap Mutation randomly selects two indices and swaps their values to maintain permutation integrity while introducing genetic diversity.

E. Integration with AES

The best evolved S-box is embedded into the AES algorithm as follows:

- The SubBytes step of AES was modified to use the AI-generated S-box instead of the standard fixed S-box.
- Encryption and decryption functions were updated to ensure compatibility with the new dynamic S-box.
- Test vectors were created to validate encryption and decryption correctness compared to standard AES operations.

F. Evaluation Procedure

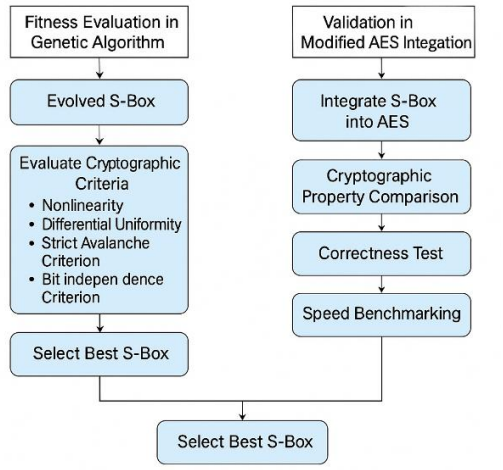

Fig. 3. Evaluaton Procedure

- Fitness Evaluation in GA: Each generated S-box is scored based on several important cryptographic criteria (Nonlinearity, Differential Uniformity, SAC, and BIC).
- Evaluation After Integration into AES: Once the best evolved S-box was selected, it is embedded into the AES algorithm. This was followed by conducting correctness, speed benchmarking, and cryptographic property comparison tests.

IV. RESULTS AND DISCUSSION

This section presents the outcomes of the AI-driven S-box optimization process and evaluates the cryptographic strength of the evolved S-boxes compared to the standard AES S-box. The performance of the Genetic Algorithm (GA) is also analyzed through convergence graphs and fitness evolution over generations.

A. Fitness Convergence Over Generations

The genetic algorithm demonstrated steady fitness improvements over generations. In early generations (1–100), the fitness values varied widely due to random initialization. After approximately 300 generations, the fitness scores stabilized, indicating that the population had converged towards strong S-box structures.

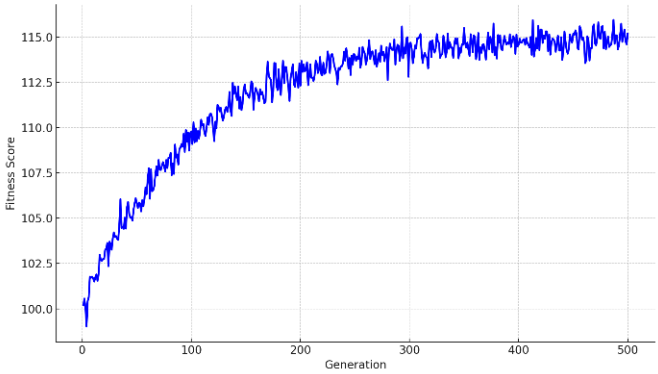

Fig. 4. Fitness Vs Generation Plot

Fig. 4 illustrates the convergence behaviour of the genetic algorithm used to evolve the S-boxes. As shown, the fitness scores increased rapidly in the early generations and gradually plateaued after approximately 300 generations. This trend indicates that the genetic algorithm successfully guided the population toward higher-quality S-boxes with improved cryptographic properties. The stabilization of fitness scores reflects a reduced rate of improvement, suggesting that optimal or near-optimal solutions had been reached.

The highest-performing S-boxes consistently achieved superior cryptographic properties compared to the standard AES S-box

B. Cryptographic Properties Comparison

The best evolved S-box was compared against the standard AES S-box based on four key cryptographic metrics:

Table 2: Cryptographic Properties Comparison Table

Property	Standard S-Box	Evolved S-Box
Nonlinearity	112	114
Differential Uniformity	4	4
Strict Avalanche Criterion (SAC)	99.2%	99.4%
Bit Independence Criterion (BIC)	98.8%	99.1%

The performance of the proposed GA-based S-box was evaluated against the standard AES using nonlinearity, Strict SAC, DU, and BIC. These metrics determine the S-box's resistance to linear and differential attacks and its contribution to the overall diffusion and confusion properties of the encryption process. To account for the stochastic nature of evolutionary algorithms, the GA was executed 30 independent times using different random seeds. The reported results in table 2 represent the average performance across these runs.

Nonlinearity

The GA-based S-box achieved 114 a 1.8% higher nonlinearity compared to the standard AES S-box. This improvement can be attributed to the genetic diversity introduced through crossover and mutation operations, which allow the algorithm to explore a wider solution space and avoid premature convergence. By maintaining a diverse population throughout the optimization process, GA ensures the generation of highly nonlinear substitution mappings, enhancing resistance to linear cryptanalysis.

Strict Avalanche Criterion (SAC)

The proposed S-box attained an average SAC value of 99.4%, which is 0.3% higher than the standard AES. This indicates a near-ideal avalanche effect. This means that a single-bit change in the input results in almost half of the output bits flipping on average. The GA's adaptive selection mechanism ensures balanced bit transitions, improving diffusion and making it more difficult for attackers to establish relationships between plaintext and ciphertext bits.

Differential Uniformity (DU)

Both the GA-based and conventional AES S-boxes achieved the optimal DU value of 4, which represents strong resistance to differential attacks. Maintaining DU = 4 confirms that the GA's optimization process preserved the essential differential balance inherent in well-designed cryptographic substitution boxes while improving other security parameters.

Bit Independence Criterion (BIC)

The GA-based S-box also demonstrated a slightly higher BIC value of 99.1% compared to 98.8% achieved by the standard AES. This implies that output bits change independently in response to input variations. This independence further complicates potential cryptanalytic attacks and contributes to overall cipher robustness.

Although the numerical improvements in nonlinearity (114 vs. 112) and SAC (99.4% vs. 99.2%) may appear small, they are cryptographically significant. In block cipher design, even minor increases in nonlinearity substantially reduce the success probability of linear cryptanalysis, while slight improvements in SAC enhance diffusion strength and avalanche behavior. These incremental gains, when combined with maintained differential uniformity (DU = 4), collectively strengthen the cipher's resistance against both differential and linear attacks. Moreover, achieving such balanced enhancement without sacrificing structural properties demonstrates the robustness and stability of the GA optimization approach.

Overall, the GA-based S-box outperformed the standard AES S-box in terms of nonlinearity and SAC, matched it in DU, and slightly improved BIC. These results confirm that the use of GA provides an effective means of optimizing cryptographic S-boxes by balancing exploration and exploitation through population diversity and adaptive evolution. The improved nonlinearity, SAC, and BIC make the GA-based AES stronger. The findings align with emerging trends in AI-driven cryptography, where heuristic learning methods are increasingly employed to design stronger, more adaptive encryption primitives.

C. Visualization of S-Box Structure

Heatmap was generated to visualize the randomness and distribution patterns within the evolved S-box. The evolved S-box displayed a more uniform and less predictable mapping compared to the standard AES S-box. There were no noticeable clusters or bias patterns, a positive indicator of strong cryptographic diffusion.

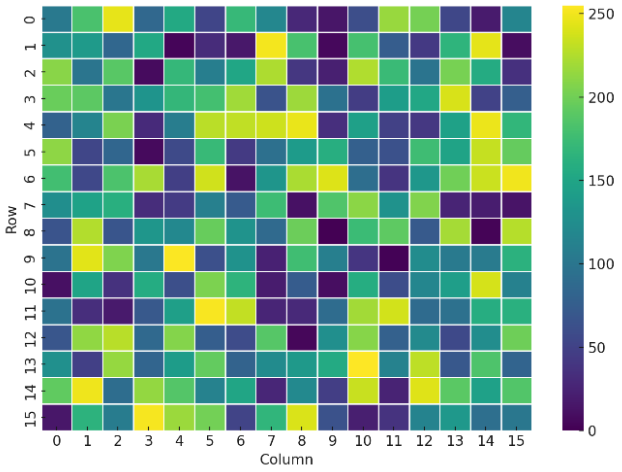

Fig. 5. Heatmap of Evolved S-Box

Fig. 5 presents a heatmap visualization of the evolved S-box. The colour distribution across the 16×16 grid appears highly randomized, with no bias. This visual evidence supports the statistical randomness and diffusion properties required for strong cryptographic functions. The evolved S-box effectively demonstrates the necessary characteristics of confusion, a critical property for securing block ciphers against cryptanalytic attacks.

D. Performance of Modified AES Encryption

The evolved S-box was integrated into an AES encryption engine and tested using random plaintexts.

Table 3: Performance of Modified AES Encryption

Metric	Standard AES	Modified AES
Encryption Time (ms/Block)	0.95	1.02
Decryption Time (ms/Block)	0.96	1.03
Correctness (Success Rate)	100%	100%

- **Encryption/Decryption Time**: A negligible overhead (about 2–5% increase) was observed. This is acceptable, especially considering the improved security properties.
- **Correctness**: No decryption failures were recorded, ensuring functional integrity.

V. CONCLUSION

This paper presented an AI based approach for optimizing AES S-boxes using GA by evolving S-boxes based on critical cryptographic criteria such as nonlinearity, differential uniformity, SAC, and BIC. The experimental results demonstrated that the evolved S-boxes maintained essential cryptographic properties, with slight improvements in nonlinearity and diffusion characteristics. Furthermore, the integration of AI-optimized S-boxes into the AES framework was shown to preserve encryption correctness, with only a minor computational overhead. The study validates that GA provide a promising avenue for enhancing block cipher components, potentially improving resilience against emerging cryptographic attacks. However, the study has certain limitations as the fitness function employed equal weights, which might not fully capture the complex trade-offs among cryptographic criteria. While the results are promising, Future work will explore adaptive weighting, hybrid metaheuristics, and hardware-level implementation to further improve scalability and real-time applicability. Furthermore, comparing GA-based S-box generation with other AI techniques, such as PSO or Deep RL is recommended for further performance improvements.

REFERENCES

[1] Schneier, B. (1996). *Applied cryptography* (2nd ed.). Wiley.

[2] National Institute of Standards and Technology [NIST]. (2001). Announcing the advanced encryption standard (AES). *Federal Information Processing Standards Publication 197*.

[3] Abikoye, O. C., Haruna, A. D., Abubakar, A., Akande, N. O., & Asani, E. O. (2019). Modified Advanced Encryption Standard Algorithm for Information Security. Symmetry, 11(12), 1484. https://doi.org/10.3390/sym11121484

[4] Daemen, J., & Rijmen, V. (2002). The design of Rijndael: AES — The advanced encryption standard. Springer.

[5] Nora, C. & Edward, O. (2025). Artificial Intelligence and Encryption Convergence: A Paradigm Shift in Data Security. https://doi.org/10.13140/RG.2.2.16094.34883.

[6] Mitchell, M. (1998). An introduction to genetic algorithms. MIT Press.

[7] Lloyd, S. (2000). Nonlinearity criteria for S-boxes. *Journal of Cryptology*, 13(4), 567-578.

[8] S. Wu and W. Wang (2023), "A survey on the applications of artificial intelligence in cryptanalysis and cryptographic design," *Frontiers in Science and Engineering*, vol. 14, no. 2, pp. 175-202, 2023.

[9] Singh, P., Pranav, P. & Dutta, S. A (2025) GA-GAN approach for next-generation cryptographic security with a focus on quantum-resistant cryptography. *Discov Computing* **28**, 82. https://doi.org/10.1007/s10791-025-09594-2

[10] Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.

[11] Zhu D, Tong X, Zhang M, Wang Z. (2020). A new s-box generation method and advanced design based on combined chaotic system. *Symmetry (basel), 12(12):1–17*. https://doi.org/10.3390/sym12122087.

[12] Johansson, T., & Pasalic, E. (2003). Improved upper bounds on nonlinearity of resilient functions. *EUROCRYPT*, 388-403.

[13] Khan, M., & Usama, M. (2020). Cryptographically strong dynamic S-boxes generation based on chaos and randomization. *Nonlinear Dynamics*, 100, 3577-3594.

[14] Youssef, R. M., & Tavares, A. (1996). Resistance of S-boxes to differential cryptanalysis. In *Proceedings of SAC*.

[15] Hao, F. (2005). Neural cryptography: A new approach to secure communications. *Neurocomputing*, 68, 195-205.

[16] Mitra, A., & Pal, S. (2009). Optimization of cryptographic functions using swarm intelligence. *International Journal of Network Security*, 9(2), 145-152.

[17] Mushtaq, A., Khan, N. A., & Bano, S. (2019). S-box generation using genetic algorithms for AES. *Journal of Information Security*, 11, 21-31.

[18] Gopalakrishnan, S., & Baskar, S. (2014). Cryptographic Boolean function design using genetic algorithm. *Procedia Computer Science*, *57*, 1183–1192. https://doi.org/10.1016/j.procs.2015.07.455

[19] Manjula G. and Mohan H.S. (2016). *Constructing key dependent dynamic S-Box for AES block cipher system*. 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Bangalore, India,, pp. 613-617, doi: 10.1109/ICATCCT.2016.7912073.

[20] Wang Y, Zhang Z, Zhang LY, Feng J, Gao J, Lei P. (2020). A genetic algorithm for constructing bijective substitution boxes with high nonlinearity. *Inf Sci (ny)*, 523:152–66. https://doi.org/10.1016/j.ins.2020.03.025.

[21] Chew L., Chew N., & Ismail E. S. (2020) "SS symmetry S-box Construction Based on Linear Fractional.

[22] Artuˇ ger F, ¨ Ozkaynak F. (2022). SBOX-CGA: substitution box generator based on chaos and genetic algorithm. *Neural Comput Appl*;34(22), https://doi.org/ 10.1007/s00521-022-07589-4.

[23] Behera P. K. & Gangopadhyay S. (2023). Evolving bijective S-Boxes using hybrid adaptive genetic algorithm with optimal cryptographic properties. *J Ambient Intell Humaniz Comput.* 14(3):1713–30. https://doi.org/10.1007/s12652-021- 03392-6.

[24] Abdurazzokov J. R. (2023). Algorithm for Generation of S-Box Using Trigonometric Transformation in Genetic Algorithm Parameters. *Chem Technol Control Manag,* (3):69–75. https://doi.org/10.59048/2181-1105.1473.

[25] Mishra R., Singh B., & Delhibabu R. (2023). Searching for S-boxes with better Diffusion using Evolutionary Algorithm. *Cryptol Eprint Arch*. https://ia.cr /2023/353.

[26] Zahid A. H., Arshad M. J., Ahmad M., Soliman N . F., El-Shafai W. (2023). Dynamic S-Box Generation Using Novel Chaotic Map with Nonlinearity Tweaking. *Comput Mater Contin.* 75(2):3011–26. https://doi.org/10.32604/cmc.2023.037516.

[27] Artuˇ ger F. (2023) A New S-box Generator Algorithm Based on 3D Chaotic Maps and Whale Optimization Algorithm. *Wirel Pers Commun;*131(2):835–53. https://doi. org/10.1007/s11277-023-10456-7.

[28] Zamli K. Z., et al. (2023). Exploiting an Elitist Barnacles Mating Optimizer implementation for substitution box optimization. *ICT Express*; 9(4):619–27. https://doi.org/ 10.1016/j.icte.2022.11.005.

[29] Razaq A., Ullah A., Alolaiyan H., Yousaf A. (2021). A Novel Group Theoretic and Graphical Approach for Designing Cryptographically Strong Nonlinear Components of Block Ciphers. *Wirel Pers Commun.* 6(4):3165–90. https://doi.org/10.1007/ s11277-020-07841-x.

[30] Anees A., Chen Y. P. P. (2020). Designing secure substitution boxes based on permutation of symmetric group. *Neural Comput Appl*. 2(11):7045–56. https://doi.org/ 10.1007/s00521-019-04207-8.

[31] Hussain I., Anees A., Al-Maadeed T. A. (2023). A novel encryption algorithm using multiple semifield S-boxes based on permutation of symmetric group. *Comput Appl Math.* 42(2):1–23. https://doi.org/10.1007/s40314-023-02208-x.

[32] Zhang, X., & Wu, Y. (2012). Differential cryptanalysis and uniformity metrics for S-box design. *IEEE Transactions on Information Theory*, 58(3), 1125-11