Journal of Research High School

Vol. 2026 (1) 40 – 46
ISSN 2688-3651 [online]

DNN-based Intelligent Key Recovery Attack with Differential Cryptanalysis on DES

Zixuan Fu[1] *

[1]Issaquah High School, Issaquah, WA, USA
*Corresponding Author: zixuan1002.fu@gmail.com

Advisor: Xiutao Feng, fengxt@amss.ac.cn

Received September 14, 2025; Revised October 5, 2025; Accepted October 20, 2025

Abstract

Deep neural networks (DNNs) have emerged as powerful tools for modeling complex data structures and extracting subtle statistical across a wide range of domains. This paper leveraged this capability by integrating artificial intelligence (AI) with differential cryptanalysis to perform key recovery attacks on the Data Encryption Standard (DES). The paper utilized high probability differential paths on reduced-round DES to construct training datasets of ciphertext differentials. A deep neural network for binary classification is trained to distinguish ciphertext differentials from reduced-round DES and random permutations. The binary classifier is known as a distinguisher in cryptography and learns subtle patterns that classical methods often miss. The distinguisher incorporated into a key-recovery procedure by evaluating candidate subkeys for the final encryption round after removing ("peeling off") the last round of DES on rounds $r \in \{6, 8, 9\}$. The proposed DNN-based distinguisher achieved up to 96.80% training accuracy, which outperforms classical statistical methods while demonstrating the current limits of key recovery on higher reduced-round DES.

Keywords: Deep neural network, Artificial intelligence, Differential attack, Block cipher, Data encryption standard

1. Introduction

In the past decade, AI has developed rapidly and become a powerful tool with the creation of OpenAI (Cousera Staff, 2025), Chat-GPT (Open AI, 2022), etc. In general, modern artificial intelligence refers to computer systems that imitate human intelligence to perform tasks that usually require human intelligence, such as pattern recognition and reasoning. Machine learning (ML) is a branch of AI that enhances its application and extends capabilities to create algorithms that follow patterns in data without requiring explicit code to describe those heuristics (Phothilimthana et al., 2023). Deep learning is a type of ML that utilizes artificial neural networks with multiple layers of processing to extract patterns from datasets (GeeksforGeek, 2025). In the field of cybersecurity, deep learning is used for malware detection (Bensaoud, 2024), network intrusion detection (Ashiku and Dagli, 2021), and even cryptanalysis (OWASP, n.d.). It evaluates small correlations and variations in the data that the human eye may not be able to detect.

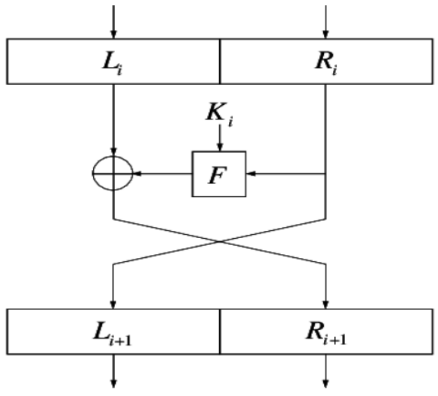

Figure 1. DES Feistel network (Hernandez-Castro et al., 2006).

The Data Encryption Standard, or DES, (National Bureau of Standards, 1977) is a symmetric block cipher algorithm

Vol. 2026 (1) 40 – 46
ISSN 2688-3651 [online]

designed in the 1970s by IBM and subsequently adopted by the U.S. government as a federal standard. DES encrypts 64-bit blocks of plaintext and uses a 56-bit secret key applying sixteen rounds of transformation, in a Feistel network.

Figure 1 shows one round of the Feistel network. A Feistel network is a symmetric encryption structure that transforms plaintext into ciphertext through a series of iterative processing rounds. Each round splits the data into two halves (left and right), and the round function generates a transformation of both halves by mixing the key with one half, then swapping them and repeating the process. Although DES was once considered secure, improvements in both computational methods and cryptanalysis have diminished its functionality. However, it serves as a viable option for teaching new students about encryption and attack methods in a less complex environment.

Differential cryptanalysis, introduced by Eli Biham and Adi Shamir in the early 1990s, was a major turning point in the study of symmetric encryption (1991). It demonstrated that block ciphers like DES that were once considered secure could be vulnerable to statistical analysis. It utilizes how small differences in plaintext affect ciphertext output. The technique quickly became one of the most extensively studied forms of cryptanalysis and significantly influenced the design of future encryption algorithms.

Today, differential cryptanalysis is less commonly used in real-world cryptographic attacks because modern ciphers have evolved to defend against it. That said, the method still holds an important place in the field of cryptographic research and education. It offers a clear framework for understanding how ciphers work internally and provides a meaningful starting point for testing novel approaches with machine learning and AI. In this study, differential cryptanalysis is revisited in a controlled setting using reduced-round DES to explore how AI can enhance and automate this classic technique.

Traditionally, differential cryptanalysis has relied on manual analysis or statistical techniques to identify useful input-output differences and connect them to potential subkey candidates. However, recent advancements in deep learning have introduced a new approach: training neural networks to distinguish between valid ciphertext differences and random noise. These models learn subtle statistical relationships in ciphertext pairs that would be difficult for humans to detect. Research by Gohr (Gohr, 2019) and subsequent work in recent years has demonstrated that neural networks can outperform traditional techniques in identifying differential patterns, particularly in reduced-round versions of DES and other lightweight ciphers (Gerault et al., 2024). By integrating AI into cryptanalysis, the process becomes more scalable and more effective at targeting intermediate encryption layers.

This paper outlined a research process for developing an AI-based differential attack on an r-round reduced version of DES. The core contribution lies in detailing a methodology to can AI-based differential distinguisher for $(r-1)$ rounds of DES and utilize this distinguisher to mount a key-recovery attack on the r round subkey K_r.

This research provided a framework for understanding and implementing AI-enhanced cryptanalytic techniques. The subsequent sections will cover the preliminaries of DES and differential analysis, methodology, results, and conclusion with further applications.

2. Preliminaries

2.1 The DES Algorithm

DES operates on 64-bit blocks of plaintext and uses a 56-bit key with 8 parity bits. It follows a Feistel network, meaning the input block is split into two halves and undergoes repeated transformations over 16 rounds. DES encrypts a 64-bit plaintext $\underline{P}$ by first applying an initial permutation (IP), then dividing the result into two 32-bit halves: L_0 and R_0 that represent the left 32 bits and the right 32 bits respectively. For each round $i = 1$ to r, the following Feistel operations are performed:

$$L_i = R_{i-1}, R_i = L_{i-1} \oplus f(R_{i-1}, K_i).$$

In the equation, K_i is the 48-bit round key from the 56-bit key, and the function f includes expansion, XOR with K_i, sixteen S-box substitution, and a permutation. The output ciphertext is $C = R_r \parallel L_r$, and this process is denoted as $C = DES_{r(P,K)}$. For more details, please refer to the paper (ScienceDirect, n.d.).

Vol. 2026 (1) 40 – 46
ISSN 2688-3651 [online]

2.2 Differential Idea and Main Differential Path of DES

Differential cryptanalysis is a statistical attack technique used to break block ciphers by studying how small differences in the plaintext affect the resulting differences in the ciphertext. It has proven to be an effective technique of symmetric-key cryptanalysis, especially with Feistel structures like the Data Encryption Standard (DES). The basic idea is to examine how a certain amount of difference in pairs of plaintexts is affected through rounds of the cipher when creating ciphertext pairs with corresponding differences. A differential path is a sequence of an input-output difference pair ($\Delta P \rightarrow \Delta C$) that occurs with a probability significantly higher than random.

In DES, due to the structure of its S-boxes and Feistel network, certain bitwise differences are more likely to persist across rounds, due to the structure of its S-boxes and Feistel network. These high-probability differentials allow an attacker to guess the intermediate values more effectively for subkey recovery. By targeting the output of the $(r-1)^{th}$ round, the attacker can bypass the final round's permutation and focus on recovering round-specific key bits through partial decryption and differential analysis.

For this research, high-probability differential paths through $r-1$ rounds of DES are utilized, where r ∈ {6, 8, 9}. The following table summarizes the selected differentials used for training and key recovery attacks:

Table 1. Plaintext differentials for reduced r-round DES.

r	$\Delta P (= \Delta_{in})$
6	0080000 04000000
8	405C0000 04000000
9	84411346 405C0000

2.3 Introduction to AI Model

The research uses a binary classifier based on a neural network that acts as an intelligent distinguisher. The model is trained on two datasets, with one containing positive samples (D_0), which are differences resulting from known differential paths, and negative samples (D_1), which randomized values to simulate false data and noise for the model to distinguish from.

The samples encode differences in intermediate values after partial decryption. The neural network learns to classify whether a sample is likely to follow a valid differential path. During the attack, this model helps evaluate subkey guesses by distinguishing meaningful patterns from random ones.

3. AI-based Differential Attack

3.1 General Idea

The goal of the AI-based differential attack is to recover the last few round subkeys of the reduced-round DES cipher by combining differential cryptanalysis with a neural network classifier.

The attack contains two main phases. Phase one's goal is to build an AI Distinguisher for $(r-1)$ Rounds with a deep neural network. This will be trained to distinguish between output differences that result from a specific input differential Δ_{in} after $(r-1)$ rounds of DES encryption (labeled as D_0) for n number of samples. Then, randomly generated on non-characteristic differences (labeled as D_1) for n number of samples as well.

Using known plaintext-ciphertext pairs encrypted under an unknown key K, the r-th round subkey K_r is guessed. For each guess, the ciphertexts are partially decrypted by one round to obtain the state differences at the output of the $(r-1)$-th round. These differences are fed as features to the neural network that acts as an intelligent distinguisher. Guesses for K_r that consistently produce differences classified as D_0 by the distinguisher are considered strong candidates for the correct subkey.

3.2 Dataset Generation and Training Method

The construction of effective training datasets D_0 and D_1 is critical for the AI distinguisher's performance. The experiment starts with searching various literature for an optimal differential route for all r. A high-probability differential characteristic for $(r-1)$ rounds of DES must be selected. This is denoted as $\Delta_{in} \rightarrow \Delta_{out}(r-1)$, where Δ_{in} is the chosen 64-bit input plaintext difference, and $\Delta_{out}(r-1)$ is the expected 64-bit difference pattern at the output of the $(r-1)$-th round.

Vol. 2026 (1) 40 – 46
ISSN 2688-3651 [online]

The specific generation of the true differential Dataset D_0 first starts with randomly generating a plaintext P_0 and a DES key K. Then the P_0 will be encrypt for $(r-1)$ rounds using the key K. Let the output be $(L_{0(r-1)}, R_{0(r-1)})$. This will allow us to calculate $P_1 = P_0 \oplus \Delta_{in}$. Encrypt P_1 for $(r-1)$ rounds using the same key K to get $(L_{1(r-1)}, R_{1(r-1)})$. Based on this, the four 32-bit difference components can be compute for the training sample: $\delta_1 = L_{0(r-1)} \oplus R_{0(r-1)}$,$\delta_2 = R_{0(r-1)} \oplus R_{1(r-1)}$, $\delta_3 = L_{0(r-1)} \oplus R_{1(r-1)}$, $\delta_4 = R_{0(r-1)} \oplus L_{1(r-1)}$. This leads to the sample for D_0 :the tuple $(\delta_1, \delta_2, \delta_3, \delta_4)$. This procedure will be repeated to generate samples n for the dataset D_0. These samples are labeled as belonging to the "true differential" class.

For the data generation of the random differentials Dataset D_1, samples in D_1 represent patterns that should not be confused with the true characteristic. In order to do so, take sample data like those in D_0 , e.g., $(\delta_1, \delta_2, \delta_3, \delta_4)$ derived from an $(L_{0(r-1)}, R_{0(r-1)})$ and $(L_{1(r-1)}, R_{1(r-1)})$ pair, randomly generate a 32-bit random number T. From this, the D_1 sample is as: (δ1 ⊕T, δ2 ⊕T, δ3 ⊕T, δ4 ⊕T). This procedure will be repeated to generate samples n for the dataset D_1. These samples are labeled as belonging to the "false differential" class.

Once D_0 and D_1 are sufficiently large (sample size n = 1,000,000 in the experiment), an AI classification model is selected. The combined dataset containing samples from D_0 and D_1 with their labels) is used to train model D. The model learns to distinguish inputs characteristic of D_0 from those characteristics of D_1.

When a test data tuple (δ1′, δ2′, δ3′, δ4′) is given to the trained distinguisher D, if D classifies it as belonging to D_0, it's considered a potential true differential pattern. Otherwise, D will classify it as D_1, which is considered random or not matching the characteristic.

3.3 Specific Steps of the Attack on r-round DES

After the AI distinguisher D for $(r-1)$ rounds is trained, it's used to attack the $r-th$ round subkey K_r. This requires multiple plaintext-ciphertext pairs (P_{0i}, C_{0i}) and (P_{1i}, C_{1i}) such that all pairs are encrypted with the *same unknown* key K and the plaintexts satisfy the chosen input differential: $P_{0i} \oplus P_{1i} = \Delta_{in}$.

The attack will start with by iterating through all possible candidate values for a K_{guess} in each for a given keyspace (keyspace = 1024 in the experiment) to guess the correct subkey K_r.

For each guessed K_{guess} and for each pair of ciphertexts, (C_{0i}, C_{1i}): let $C_{0i} = R_{0r}||L_{0r}$ *and* $C_{1i} = R_{1r}||L_{1r}$. To obtain the output of the $(r-1)$-th round for P_{0i}, compute: $* R_{0(r-1)'} = L_{0r} * L_{0(r-1)'} = R_{0r} \oplus f(L_{0r}, Kguess)$. Similarly, for P_{1i} using C_{1i} and K_{guess} $* R_{1(r-1)'} = L_{1r} * L_{1(r-1)'} = R_{1r} \oplus f(L_{1r}, Kguess)$.

Using the decrypted states, calculate the four difference component $* \delta_{1'} = L_{0(r-1)'} \oplus L_{1(r-1)'} * \delta_{2'} = R_{0(r-1)'} \oplus R_{1(r-1)'} * \delta_{3'} = L_{0(r-1)'} \oplus R_{1(r-1)'} * \delta_4 = R_{0(r-1)} \oplus L_{1(r-1)'}$ The input to the distinguisher is the tuple $(\delta_{1'}, \delta_{2'}, \delta_{3'}, \delta_{4'})$. In the experiment, this tuple is fed into the trained AI distinguisher D. If D classifies the tuple as belonging to D_0, then K_{guess} is recorded as a key candidate, and given a score near 1. Otherwise, the guess is considered wrong for this pair and scored near 0.

After processing multiple plaintext-ciphertext pairs, the K_{guess} that consistently results in a D_0 classification is considered a strong candidate. If the number of pairs is sufficient, the correct K_r should accumulate a significantly higher score with a normalized score range from 0 to 1. This is then verified by enumerating remaining candidates by full trial decryption or other methods to find the correct subkey.

4. Experiment Results

4.1 Six Round Attack

To further illustrate the model's performance for each round, the following figures showcase a loss and accuracy plots provide a visual evaluation of the deep neural network's learning progress over 40 epochs. All the graphs exhibited similar outcomes that support the conclusion of the effectiveness of the neural network in training.

Table 2: Accuracy and loss of neural network for r round attack over epochs trained using selected plaintext differentials

r	Accuracy	Loss
6	96.80%	0.1542
8	94.26%	0.1930
9	85.99%	0.4091

Vol. 2026 (1) 40 – 46
ISSN 2688-3651 [online]

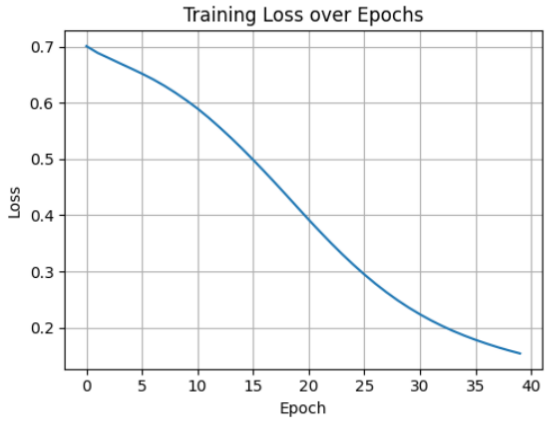

Figure 2. Visualization of training loss over epochs for 6 round attack.

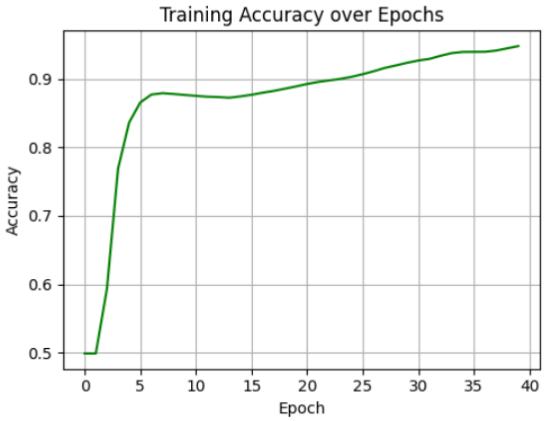

Figure 3. Visualization of training accuracy over epochs for 6 round attack.

4.2 Eight Round Attack

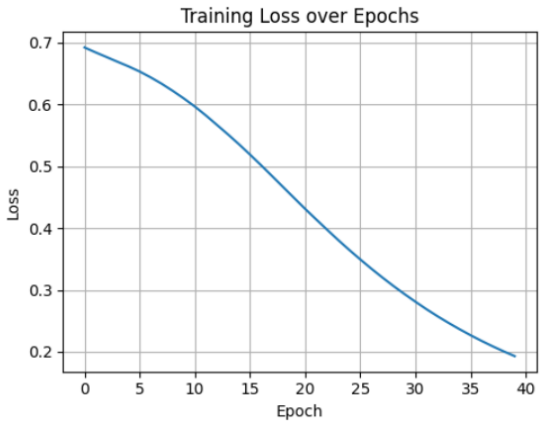

Figure 4. Visualization of training loss over epochs for 8 round attack.

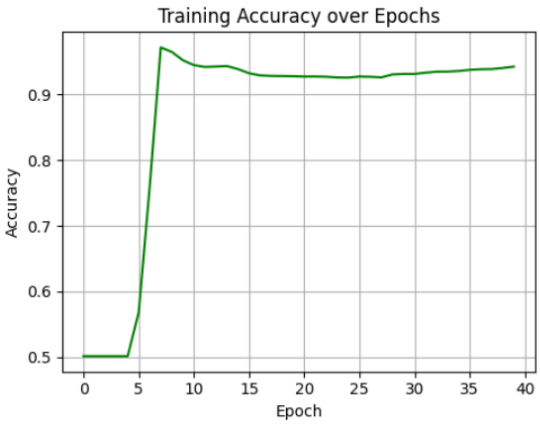

Figure 5. Visualization of training accuracy over epochs for 8 round attack.

4.3 Nine Round Attack

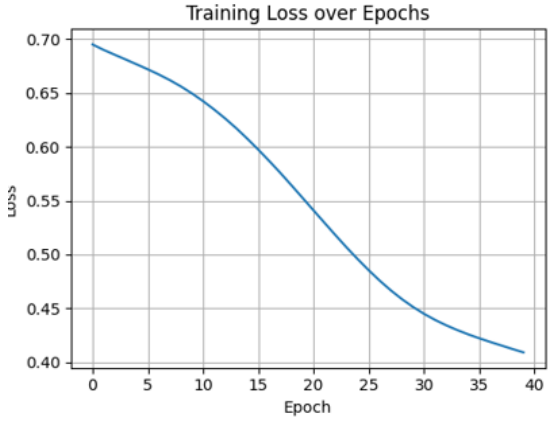

Figure 6. Visualization of training loss over epochs for 9 round attack.

Figure 7. Visualization of training accuracy over epochs for 9 round attack.

Vol. 2026 (1) 40 – 46
ISSN 2688-3651 [online]

5. Discussion

This paper demonstrated a research outline for utilizing a neural network as a differential distinguisher to facilitate a key recovery attack on several *r*-round reduced DES. The methodology selected an appropriate differential characteristic, generated datasets D_0 and D_1 and used this data to train the neural network classifier. Finally, the trained model evaluated subkey guesses in the attack on the final round. The plots in figures 2 and 3 for 6 round attack show the training performance of the neural network over 40 epochs, with loss steadily decreasing from ~70% to ~15% and accuracy rising from ~50% to over 94%. This indicates successful convergence and strong learning of differential features for the DES distinguisher, and a similar pattern is observed for 8 and 9 round attack for figures 4, 5, 6 and 7. It highlighted the neural network's success and capacity to capture complex statistical relationships inherent in the encryption process for DES.

5.1 Limitation

Despite achieving high classification accuracy during training and testing phases, a notable limitation of this work lay in the relatively low success rate during the key recovery phase. While the distinguisher performed well in differentiating between real and fake differential features, this did not translate into effective subkey identification during attacks for the higher rounds.

One possible explanation for this is the over-simplification of the training data used to train the neural distinguisher. The real samples were generated from output values of DES encryption under a fixed differential input, while the fake samples were generated using fully random values that possibly lacked structural correlation to the encryption process. As a result, the neural network may have learned superficial statistical differences without capturing some of the underlying cryptographic relationships. Future works could potentially address this by creating more structurally realistic negative samples in utilizing combinations of both real and XORed components from dataset D_0.

5.2 Further application

The integration of artificial intelligence with differential cryptanalysis has significant potential for application beyond DES. This AI-based methodology can be adapted to other symmetric ciphers where neural network distinguishers often outperform traditional techniques. AI-driven cryptanalysis also enables the development of automated tools to provide fast security evaluation of new ciphers. More generally, machine learning can help identify vulnerabilities in cryptographic implementations and simulate attack scenarios to improve security. While challenges persist in construction of interpretable frameworks with limited results, the AI and cryptanalysis is already transforming how cryptographic systems are analyzed and secured and will continue to improve in the future.

6. Conclusion

This research demonstrated that deep neural networks (DNNs) can significantly improve the process of differential cryptanalysis for reduced-round DES, outperforming classical statistical techniques in distinguishing differential patterns within ciphertexts. By leveraging high-probability differential paths to construct rigorous training datasets, the study developed an AI-based distinguisher that was able to achieve accuracy up to 96.80% for 6 round attack, 94.26% for 8 round attack, and 85.99% for 9 round attack. The findings highlighted that integrating machine learning with traditional cryptanalytic methods enhances both the scalability and effectiveness of attacks on block ciphers, and this offered a promising framework for automating future cryptanalysis.

While some challenges remain, particularly regarding the generation of realistic negative samples and improving key recovery on higher-round DES, the broader significance lies in pioneering new approaches in AI-driven cryptanalysis that can be adapted for testing and securing modern encryption schemes.

Vol. 2026 (1) 40 – 46
ISSN 2688-3651 [online]

References

Ahmed, B. Jugal K., & Mahmoud, B. (2024). survey of malware detection using deep learning. *ScienceDirect*, *16*, https://www.sciencedirect.com/science/article/pii/S2666827024000227

Ashiku, L., & Dagli, C. (2021). Network Intrusion Detection System using Deep Learning. *Procedia Computer Science*, *185*, 239–247, 2-3. https://doi.org/10.1016/j.procs.2021.05.025

Biham, E., & Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1), 3-72.

Data Encryption Standard - an overview | ScienceDirect Topics. (n.d.). Www.sciencedirect.com. https://www.sciencedirect.com/topics/engineering/data-encryption-standard

Gerault, D., et al. (2024). SoK: 6 Years of Neural Differential Cryptanalysis. *Cryptology ePrint Archive*, 19-29. https://eprint.iacr.org/2024/1300

Gohr, A. (2019). Improving Attacks on Round-Reduced Speck32/64 using Deep Learning. *Cryptology EPrint Archive*. https://eprint.iacr.org/2019/037.

Introducing ChatGPT. (2022, November 30) OpenAI. Retrieved May 20, 2025, from https://openai.com/index/chatgpt/

Introduction to Deep Learning. (2025, July 11). GeeksforGeek. Retrieved May 11, 2025, from https://www.geeksforgeeks.org/deep-learning/introduction-deep-learning/

National Bureau of Standards. (1977, January 15). Data Encryption Standard (FIPS Pub. 46). U.S. Department of Commerce. https://csrc.nist.gov/files/pubs/fips/46/final/docs/nbs.fips.46.pdf

Phothilimthana, P. M., & Perozzi, B. (2023). *Advancements in machine learning for machine learning*. Research.google. https://research.google/blog/advancements-in-machine-learning-for-machine-learning/

Rezos., KristenS., & kingthorin. *Cryptanalysis*. OWASP. https://owasp.org/www-community/attacks/Cryptanalysis

What Is OpenAI? Everything You Need to Know. (2023, November 27). Coursera. Retrieved May 18, 2025, from https://www.coursera.org/articles/what-is-openai

Wheedham: An Automatically Designed Block Cipher by means of Genetic Programming - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/llustration-of-a-round-in-a-Feistel-network_fig1_224645711