Mesopotamian journal of Cybersecurity
Vol.5, No.3, **pp**. 927–952
DOI: https://doi.org/10.58496/MJCS/2025/053; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Enhanced Key Generation Method using Deep Q-Networks Algorithm with Chaotic Maps

Ali A. Mahdi [1], Mays M. Hoobi [1,*]

[1] *Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq*

ARTICLEINFO

Article History

Received 17 Apr 2025
Revised 9 Jun 2025
Accepted 9 Aug 2025
Published 17 Sep 2025

Keywords

Cryptography
Chaotic
DQN
NIST
Randomness

ABSTRACT

In contemporary digital environments, exponential cyber threat growth has made cryptographic key generation a critical security challenge. Traditional Pseudo-Random Number Generators (PRNGs) and existing chaos-based methods often exhibit insufficient entropy, limited randomness quality, and inadequate resistance to statistical attacks. Current implementations frequently produce suboptimal entropy values and fail to meet modern cryptographic security standards and rigorous randomness testing protocols. This paper aims to design and implement an advanced cryptographic key generation system that combines Deep Q-Networks (DQN) algorithms with multiple chaotic maps to produce cryptographically secure stream key bits with high randomness and strong resistance to cryptanalytic attacks. The proposed DRLKG-Chaotic (Deep Reinforcement Learning Key Generation with Chaotic maps) system implements six distinct experimental scenarios utilizing five chaotic maps: Tent, Ikeda, Chua's, Rössler, and Double Pendulum. The first five scenarios individually integrate each chaotic map with a DQN algorithm, whereas the sixth scenario implements a novel fusion approach that incorporates all five maps simultaneously. Each scenario generates key streams of three different lengths (128-bit, 192-bit, and 256-bit) to accommodate varying security requirements. A comprehensive evaluation using the National Institute of Standards and Technology (NIST) statistical test suite, brute-force attack resistance analysis, Auto Correlation (AC), Cross Correlation (CC), and Discrete Fourier Transform (DFT) analysis demonstrates the significant improvements over standard chaotic implementations. The results indicate that the DQN scenarios achieve entropy values ranging from 0.9097--0.9999, whereas the standard chaotic maps achieve values ranging from only 0.3627--0.5463. All NIST test P values consistently exceed 0.90 across all the parameters, indicating superior randomness characteristics. In addition, reliable results are obtained when various types of attacks, such as brute-force attacks, side-channel attacks, and timing attacks, are applied.

1. INTRODUCTION

Digitalization and interconnected systems have reshaped information security, increasing the need for cryptographic solutions capable of safeguarding sensitive data against increasingly sophisticated cyber threats [1], [2]. Modern cryptographic systems rely heavily on the quality and unpredictability of cryptographic keys, which serve as the foundation for ensuring data confidentiality, integrity, and authentication across diverse computational environments [3], [4]. Pseudo-Random Number Generators (PRNGs) constitute the cornerstone of contemporary cryptographic implementations, providing the essential randomness required for key generation, initialization vectors, nonce, and other security-critical parameters [5], [6]. However, traditional PRNGs often present fundamental limitations in terms of entropy quality, period length, and resistance to statistical analysis, rendering them potentially vulnerable to advanced cryptanalytic attacks [7], [8], [9]. Symmetric (secret-key) and asymmetric (public-key) encryption are two classes that classify cryptography algorithms [10]. Chaotic systems offer cryptographic potential because of their sensitivity to initial conditions, aperiodicity, and deterministic chaos—properties we exploit to enhance key generation. These mathematical constructs exhibit complex nonlinear dynamics that can generate sequences with high entropy and statistical properties suitable for cryptographic purposes [11], [12]. Recent cryptographic research has extensively explored the application of various chaotic maps, including the Tent map, Ikeda map, Chua's circuit, Rössler Attractor, and Double Pendulum double pendulum systems, for generating cryptographically secure sequences [13], [14]. However, traditional implementations of chaotic-based PRNGs often suffer from limited parameter optimization, finite precision effects, and inadequate adaptation to varying security requirements [15], [16]. The integration of artificial intelligence, particularly Deep Reinforcement Learning deep reinforcement learning (DRL), into cryptographic system design represents a paradigm shift in security engineering [17],

*Corresponding author. Email: Mays.m@sc.uobaghdad.edu.iq

[18]. Deep Q-Networks (DQN networks (DQNs), as a prominent DRL algorithms, can optimize complex parameter spaces through iterative learning processes, potentially enhancing the quality and security properties of cryptographic sequences [19], [20]. This paper is structured as follows: Section 2 clarifies the contributions and novelty of this paper. Section 3 reviews the relevant literature with chaotic maps. Section 4 introduces the chaotic maps utilized in this paper. Section 5 illustrates the details of Deep Q-Networks (DQN deep Q-networks (DQNs). Section 6 describes the evaluation methodology and metrics. Section 7 presents the proposed key generation system. Section 8 analyses the experimental results and performance. Finally, Section 9 and section 10 present the conclusions and future work of this paper.

2. CONTRIBUTIONS

This paper is novel in that it proposes new methods for cryptography key generation, and its contributions span enhancing the security, efficiency, robustness, and applicability of cryptographic systems. The major contributions of this paper are as follows:

- Generation of high-entropy and unpredictable keys makes cryptosystems resistant to several types of cryptanalytic attacks, such as brute force, side-channel attack, and timing attack.
- Satisfying the statistical randomness of generated keys with all proposed scenarios.
- Satisfying the integration between five types of chaotic maps (Tent, Ikeda, Chua's, Rössler, and Double Pendulum) with AI-driven key generators by using the reinforcement learning algorithm (DQN) to generate adaptive and evolving keys.

3. RELATED WORK

The use of chaotic maps in cryptography has significantly increased, and it is extremely sensitive to initial conditions and complex behaviour. The mathematical literature offers an extensive collection of chaotic dynamical systems suitable for high-entropy stochastic sequence generation applications. This section reviews several studies that explore chaos for key generation.

In [21], researchers presented a review of chaotic map applications in PRNG and encryption, emphasizing post-COVID-19 cybersecurity requirements. This research covered multiple chaotic maps (Ikeda, Henon, Tinkerbell, and quantum chaotic maps) and noted correlations as low as 0.00006 (Ikeda), a maximum entropy of 7.999995 bits/byte (quantum maps), and the use of Zaslavsky maps for minimal execution time (0.23 s). With a data rate of 15.367 Mbit/second using hyper chaotic setups, they provided guidelines for optimizing different chaotic map implementations according to performance and security needs.

In [22], a hybrid encryption/decryption approach for images was proposed, in which a 3D hyper chaotic map and a 2D mersister map were merged with a Convolutional Neural Network convolutional neural network (CNN) to increase the decryption accuracy. The technique was evaluated using entropy, correlation, histogram analyses, noise resistance, the number of pixel change rate (NPCR) and Unified Aver-age Change Intensity (UACI). Outcomes the unified average change intensity (UACI). The outcomes included high entropy (~7.598), low correlation between encrypted and original images, and robust noise resistance, with strong NPCR and UACI results. The method successfully balanced security (low MSE (Mean Square Error mean square error), high SSIM (Structural Similarity Index Measure structural similarity index measure)) and practicality for secure image transmission.

In [23], several recent PRNG advances were examined, including a 2D Hénon-Sine hyper chaotic hyper chaotic map with microcontroller implementation, fuzzy triangular numbers in a modified logistic map, and combined quantum random walks with chaotic map outputs. A different strategy introduced fractal structures into the tent map, increasing randomness via intrinsic mathematical complexity and demonstrating superior performance in surrogate testing over MATLAB's default random generator and traditional chaotic map-based PRNGs.

In [24], Oliveira explored shrimp-shaped structures in the Ikeda map's parameter space. By employing high-resolution parameter scans and Lyapunov exponent calculations, detailed period-doubling bifurcations were uncovered, including a Feigen Baum constant $\delta \approx 4.669248396257327$. A major methodological contribution was using dual dissipation parameters (u_x, u_γ) to capture how real and imaginary parts drive transitions between regular and chaotic regimes. This approach provides new insights for optical cavity dynamics and hints at applications in chaos-based encryption, clarifying how stable regions and bifurcation sequences form at high precision.

In [25], Zhao et al. proposed a PRNG based on the integration of chaotic maps and quantum random walks to enhance randomness and distribution uniformity. They constructed a surjective mapping satisfying Li-Yorke chaos conditions and developed a perturbation algorithm using a two-dimensional hyper chaotic system to disturb parameters and inputs,

effectively expanding the key space. The algorithm combines chaotic system outputs with sequences generated from random quantum walks on ring graphs, achieving a uniform distribution. The performance evaluation revealed exceptional results: approximate entropy values reaching 7.9999 (near the ideal 8 bits), autocorrelation coefficients within (-0.05, 0.05), a substantial key space of approximately 2^208, and a generation speed of 4,347 keys per second.

In [26], Subathra et al. proposed a 5D hyper chaotic map + U-Net. An U-Net segments significant regions of a medical image; the statistical information guides the generation of chaotic sequences from a five-dimensional hyper chaotic system. Zig-zag scrambling and dynamic DNA operations were used for diffusion. The scheme achieved a Shannon entropy ≈ 7.9971, NPCR of 99.61%, UACI ≈ 33.49%, and a key.

In [27], Devi et al. proposed a 2D modified Tinkerbell–Henon map. A novel two-dimensional chaotic map combining Tinkerbell and Henon maps produces pseudo-random keys. The authors used the map within a shuffling–diffusion encryption algorithm and showed Shannon entropy (≈7.99), correlation coefficients near zero, NPCR ≈ 99. 6%, UACI ≈ 33. 4% and an enormous key space (~10^270).

In. [28], Premakumari et al. proposed a reinforcement Q-learning-based adaptive encryption framework for wireless sensor networks (WSNs). A deep-learning anomaly detector classifies network conditions into low, moderate or high threat levels; RL agents choose encryption levels accordingly. Dynamic Q-learning is used for low-threat conditions to optimize energy efficiency, whereas double Q-learning improves security in high-threat scenarios. The experiment of this work gives an entropy value of 0.85. Table 1 presents a concise description of the related work.

TABLE I. SHORT DESCRIPTION OF RELATED WORK

Approach	Year	Entropy	Adaptability	Resource Usage	Practical Deployment	Main Limitation
Multiple Chaos [21]	2022	0.875	Static	Low	Easy	Fixed parameter optimization
Quantum-Chaotic [25]	2023	0.799	Dynamic	Very High	Difficult	Multisystem integration complexity
CNN Hyperchaotic [22]	2024	0.948	Limited	Very High	Moderate	CNN computational overhead
Fractal-Tent [23]	2024	0.701	Limited	Moderate	Moderate	Limited to single map enhancement
Ikeda Analysis [24]	2024	0.612	Limited	Low	Easy	Parameter space exploration only
5D hyperchaotic map + U-Net, A U-Net [26]	2025	≈ 7.9971	Dynamic	Very High	Difficult	computational overhead
Modified Tinkerbell–Henon map [27]	2025	≈7.99	Static	Dynamic	Difficult	Multisystem integration complexity
reinforcement Q-learning-based adaptive encryption [28]	2025	0.85	Dynamic	Very High	Difficult	Fixed parameter optimization, Parameter space exploration only

4. CHAOTIC MAP

As a specialized mathematical field, chaos theory has attracted significant research interest because it exhibits seemingly disordered and random behaviour while maintaining extreme sensitivity to starting conditions [5], [29]. A variety of chaotic maps are available for PRNG; this section focuses on five maps (Tent, Ikeda, Chua's, Rössler and Double Pendulum) for the proposed system in this paper as follows:

3.1 Tent Map

The Tent map represents a one-dimensional chaotic function demonstrating unpredictable behaviour that finds extensive application in both dynamical systems analysis and cryptographic implementations [30]. The Tent map, a simple structure, is useful for cryptographic applications to generate a PRNG but has a limited key parameter space [20], [31]. The Tent-Map tent map serves as a fundamental example of a chaotic system characterized by its one-dimensional, noninvertible, piecewise linear discrete properties [32]. Its practical applications extend to pseudorandom number generation, data encryption mechanisms, and robust protocols for secure communications. The state is initialized as a one-element array with a random value between -1 and 1, and the Tent map function updates this scalar state at each step [30].

3.2 Ikeda Map

Is a two-dimensional chaotic map, the model of this map is known for its fractal structure and sensitivity to initial conditions. The Ikeda map is defined using via two variables, u_n andt_n, which represent the real and imaginary parts of a complex dynamical system. The evolution of the system depends on the interaction between these two variables. During initialization of the Ikeda map, the state is a two-element array with random values between -1 and 1 [24].

3.3 Chua's Circuit Map

Is a three-dimensional map requiring three variables x, y, and z to describe its state fully; these variables represent voltages and currents in the circuit components. The system's system evolution depends on the interactions among these three variables [33], [34]. Chua's circuit is one of the simplest electronic circuits capable of generating chaotic signals; it exhibits a variety of chaotic attractors depending on the parameters [35], [36]. The Initialization of Initialize Chua's state by using a three-element array with random values between -1 and 1 [35].

3.4 Rössler Attractor Map

As a three-dimensional map, three variables, *x, y, and z, are used* to describe its state; these variables represent abstract quantities in a mathematical model of a chemical reaction. The system's dynamics emerge from the interactions among these variables [37], [38]. The Rössler system is known for its chaotic attractor, which, like the Lorenz attractor, has a fractal structure and serves as a simplified model for studying chaos in continuous-time systems. The state is a three-element array with random values between -1 and 1. The Roessler Attractor function updates the state vector [*x, y, z]* at each time step via numerical integration [37].

3.5 Double Pendulum theory

The double pendulum is a classic example of a simple physical system exhibiting chaotic dynamics, and slight differences in initial conditions can lead to vastly different trajectories [39]. The double pendulum consists of two pendulums attached end to end, and the exact equations are complex and involve trigonometric functions and parameters such as masses and lengths [40]. The full description of the double pendulum state is based on four variables: θ_1, ω_1, θ_2, and ω_2 [41], [42]. For more illustration of the above chaotic maps, see Table 2 [21], [42], [43].

TABLE II. DETAILS OF TENT , IKEDA, CHUA'S CHUA, RÖSSLER AND DOUBLE PENDULUM MAPS

Tent Map	
State Size	**1**
Equation	$x_{n+1} = \begin{cases} \mu x_n & \text{if } x_n < 0.5 \\ \mu(1 - x_n) & \text{if } x_n \geq 0.5 \end{cases}$
Description	x_n: Current state, μ Control parameter (usually $\mu \in [0,2]$)
Ikeda Map	
State Size	**2**
Equation	$\begin{cases} x_{n+1} = 1 + u(x_n \cos(t_n) - y_n \sin(t_n)) \\ y_{n+1} = u(x_n \sin(t_n) + y_n \cos(t_n)) \\ t_n = a - \dfrac{b}{1 + x_n^2 + y_n^2} \end{cases}$
Description	u: State variables, ***a, b*** : Map parameters.
Chua's Circuit	
State Size	**3**
Equation	$\begin{cases} \dot{x} = \alpha(y - x - f(x)), \\ \dot{y} = x - y + z, \\ \dot{z} = -\beta y \end{cases}$ $f(x) = m_1 x + \frac{1}{2}(m_0 - m_1)(\lvert x + 1 \rvert - \lvert x - 1 \rvert)$
Description	*x, y, z*: State variables, α, β, m_0, m_1 :Circuit parameters $f(x)$: Nonlinear function of x.
Rössler Attractor	
State Size	**3**
Equation	$\begin{cases} \dot{x} = -y - z, \\ \dot{y} = x + ay \\ \dot{z} = b + z(x - c). \end{cases}$

Description	x, y, z are the state variables. $\dot{x}, \dot{y}, \dot{z}$ denote time derivatives $\frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt}$.
Double Pendulum	
State Size	**4**
Equation	$\begin{cases} \frac{d^2\theta_1}{dt^2} = \frac{-g(2m_1+m_2)\sin\theta_1 - m_2 g\sin(\theta_1-2\theta_2) - 2\sin(\theta_1-\theta_2)m_2\left[\dot{\theta}_2^2 l_2 + \dot{\theta}_1^2 l_1\cos(\theta_1-\theta_2)\right]}{l_1\left(2m_1+m_2-m_2\cos(2\theta_1-2\theta_2)\right)}, \\ \frac{d^2\theta_2}{dt^2} = \frac{2\sin(\theta_1-\theta_2)\left[\dot{\theta}_1^2 l_1(m_1+m_2) + g(m_1+m_2)\cos\theta_1 + \dot{\theta}_2^2 l_2 m_2\cos(\theta_1-\theta_2)\right]}{l_2\left(2m_1+m_2-m_2\cos(2\theta_1-2\theta_2)\right)} \end{cases}$
Description	Pendulums Angles:θ_1, θ_2 ω_1 : Angular velocity of the first pendulum. ω_2 : Angular velocity of the second pendulum. m_1, m_2 : Masses of the first and second pendulums L_1, L_2 : Lengths of the first and second pendulums g : Acceleration due to gravity

5. DEEP Q-NETWORKS (DQN) METHOD

DQN represents a significant advancement in which deep neural networks are successfully applied to approximate value functions in Deep Reinforcement Learning deep reinforcement learning (DRL), enabling agents to learn policies directly from raw sensory input, such as pixels in images [44]. Reinforcement Learning (RL) is a learning paradigm in which an agent interacts with an environment to achieve a goal. The agent learns by receiving rewards or penalties on the basis of its actions, aiming to maximize cumulative rewards over time [45]. In the DQN, an agent represents the learner or decision maker, whereas the environment implements the external system with which the agent interacts. DQN state (s) is a representation of the current situation of the agent, and action (a) represents the set of all possible moves the agent can take. Finally, the reward (r) of the DQN implements feedback from the environment because of the agent's action [45]. Q-Learning is a model-free RL algorithm that seeks to learn the value of taking a particular action in each state, quantified by the Q-value. DQN addresses the limitations of traditional Q-Learning by utilizing using deep neural networks as function approximations to estimate the Q-function [45]. The most important keys of the DQN included include the following points [46]:

1. Function approximation with deep neural networks: Convolutional Neural Networks neural networks (CNNs) are employed to process high-dimensional input like inputs such as images.

2. Experience replay: This method stores experiences in a replay memory and samples mini batches minibatches to break correlations between sequential data.

3. Fixed-target networks use a separate target network to stabilize training by keeping the target Q values constant for a fixed number of iterations. Table 3 illustrates the definitions of the DQN parameters [47].

TABLE III. DQN HYPERPARAMETER DEFINITIONS

	DQN parameter	**Definition**
1	learning rate	Step size for optimizer updates
2	buffer size	Size of the replay buffer
3	learning starts	Time steps before learning starts
4	batch size	Batch size for training updates
5	gamma	Discount factor for future rewards
6	exploration fraction	Fraction of training time for exploration schedule
7	Exploration final episode	Final value of random action probability
8	TAU	Soft update coefficient for target network
9	Target update interval	Update frequency for target network

10	Train Frequency	Network update frequency (steps)
11	Gradient steps	Gradient steps per optimization step
12	Policy Net Architecture	Neural network architecture

The DQN agent consists of the following layers [48], [49]:

1. Input Layer

The purpose of the input layer is to receive the state representation from the environment. Typically, the number of input layers is one. In addition, the number of neurons in the input layer equals the size of the observation space of the environment; for example, if the state is represented by a vector of size N, the input layer has N neurons.

2. Hidden Layers

The purpose of the hidden layers is to process the input data to extract meaningful features that help the agent estimate the Q-values for each action. The number of hidden layers is typically two to three. The number of neurons in each layer can vary, often ranging from 64--512. The ReLU (Rectified Linear Unit rectified linear unit (ReLU) activation function is commonly used after each hidden layer to introduce non-linearity nonlinearity.

3. Output Layer

The purpose of the output layer is to produce the Q values for each possible action in the action space. One output layer with a number of neurons equals the size of the action space. Linear activation (no activation function) is used because the Q values can take any real value.

6. EVALUATION METRICS

Several tests are used to evaluate the robustness and randomness of the generated stream key bits; this section presents the metrics used in this paper.

6.1. NIST tests

The National Institute of Standards and Technology (NIST) statistical test battery has established itself as the premier evaluation framework for cryptographic randomness verification. Its methodical assessment approach encompasses multiple dimensions of stochastic behaviour, facilitating thorough quantification of unpredictability characteristics. The framework's widespread adoption in security engineering stems from its adaptable architecture and exhaustive analytical capabilities, positioning it as the authoritative benchmark for cryptographic sequence evaluation in research and industrial applications [50].

6.2. Brute-Force Attack Force Attack

A brute-force attack involves exhaustively trying all possible keys, so a cryptosystem’s security hinges on an astronomically large and complex key space. Chaotic map-based key generators naturally offer extremely large key spaces (often >2128 possibilities) that render brute-force attacks infeasible [51]. Moreover, chaotic keys exhibit high key sensitivity, meaning that even a minute change in the initial chaotic parameters produces a completely different key sequence. This unpredictability prevents attackers from reducing the search space by guessing partial patterns. Recent studies have shown that chaos-driven neural key generators achieve vast key spaces and successfully resist exhaustive (brute-force) key searches [52].

6.3.Side-Channel Channel Attacks

A side-channel attack is an exploit that targets the physical implementation of a cryptosystem rather than its mathematical design by observing unintended leakages such as power and electromagnetic emissions to infer secret information [53]. This definition emphasizes that even mathematically secure algorithms can be compromised if their hardware behaviour reveals correlated data about the secret key [54]. Side-channel attack introduces a difference: it does not directly measure a key's entropy; instead, it assesses the security of the physical implementation of the system that uses the key, encryption, and execution [55] , [56].

6.4.Timing Attack

In cryptography, a timing attack is a side-channel attack in which the attacker attempts to compromise a cryptosystem by analysing the time taken to execute cryptographic algorithms [57]. It’s focus on exploit It focuses on exploiting runtime variations of cryptographic algorithms to reveal sensitive information [58], [59].

6.5.Auto Correlation (AC)

Autocorrelation measures how well a sequence correlates with a shifted version of itself, and in cryptographic keys, a low autocorrelation (near zero for any nonzero shift) is desired. If a chaotic key sequence has significant autocorrelation at some lag, it would indicate repeating patterns or predictability as a vulnerability for attackers [60].

6.6.Cross Correlation (CC)

Cross-correlation evaluates the similarity between two different sequences. In the context of key generation, low cross-correlation between keys (or key streams) is crucial for **key** distinctiveness; each key should be unique and not inferable from another [61].

6.7.Discrete Fourier Transform (DFT)

Applying a discrete Fourier transform to key sequences allows analysis in the frequency domain, which helps detect any periodic or spectral patterns that could weaken security. A perfectly random or chaotic key sequence should exhibit a flat frequency spectrum (no dominant frequency components) [50]. In the context of chaotic map keys, passing this DFT test indicates that the sequence has no discernible periodicity [51].

7. PROPOSED SYSTEM (DRLKG-CHAOTIC)

Recent cryptographic research has emphasized chaos-based sequence generation algorithms for security applications. Despite this trend, various existing implementations yield output sequences with insufficient entropy and predictability characteristics to meet modern cryptographic security thresholds. DRL offers promising avenues for enhancing key generation processes through improving the adaptive response of security mechanisms. DRL agents can iteratively learn and identify optimal strategies for parameter selection in various cryptographic operations. This section describes the proposed system for generating stream key bits with high randomness by using five types of chaotic maps (Tent, Ikeda, Chua's, Rössler and Double Pendulum) and the DRL algorithm DQN with different scenarios. The proposed system is called DRLKG-Chaotic, which is the shortest for Deep Reinforcement Learning Key Generation with Chaotic maps deep reinforcement learning key generation with chaotic maps, as shown in Figure 1.

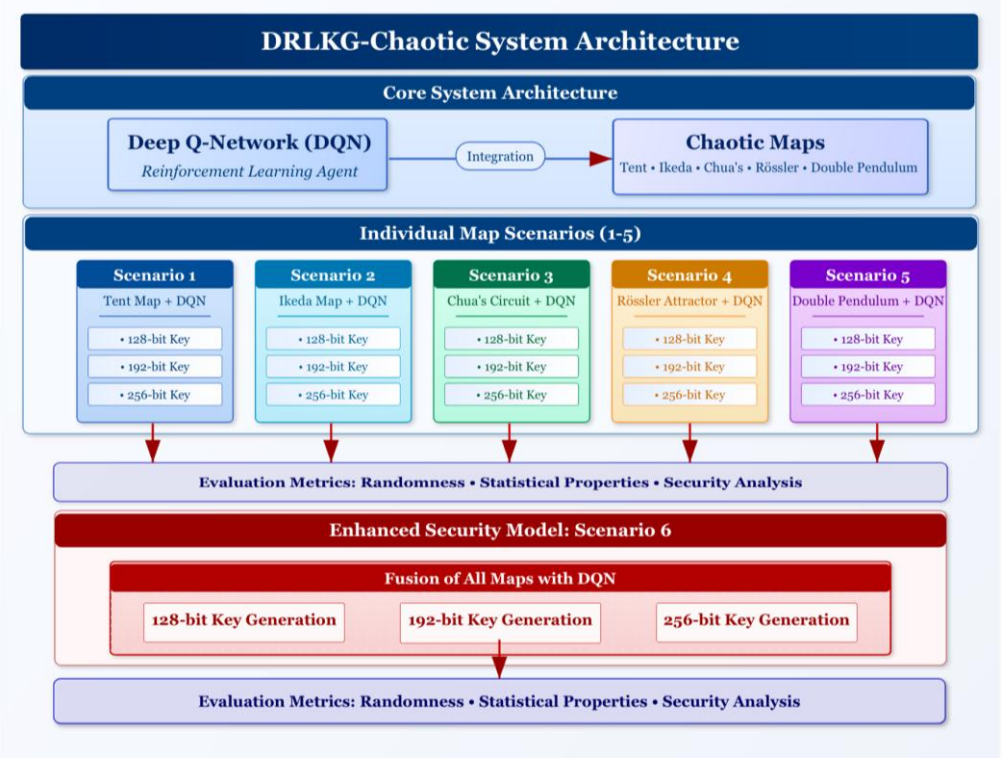


Fig. 1. Scenarios of the proposed system (DRLKG-Chaotic)

The methodology for implementing the proposed DRLKG-Chaotic system uses five chaotic maps (Tent, Ikeda, Chua's, Rössler and Double Pendulum). This proposed system is represented by six scenarios. In the first five scenarios, each of the five chaotic maps is used with the DQN agent separately. Finally, the remaining scenario is implemented by f using the DQN

with five maps at the same time. The last scenario is used to increase the complexity of the generated stream key bits. Each of the proposed scenarios generates several strong randomness stream key bits with three different lengths (128, 192, and 256 bits). This proposed system ensures that the output is a stream of independent and random sequences; furthermore, to increase the complexity of the generated sequences, the fusion of five maps with the DQN is used. It is crucial to select parameters that promote near-optimal randomness; therefore, the following subsections illustrate the parameters used with each scenario.

7.1 Scenario-1: Tent Map-DQN

For this implementation, the algorithm processes an input value denoted as x within the range [0,1] and produces a transformed output by applying the parameter μ as a scaling factor to the minimum value between x and its complement (1-x). The control parameter μ operates within the bounds of [1.0, 2.0]. In our implementation, we initialize μ with values randomly drawn from a uniform distribution ranging from 1.0--1.9999 during the training phase. Additionally, a one-dimensional Tent- tent map is implemented with a DQN agent to generate keys from various strong stream bits with three different lengths (128, 192, and 256 bits) [32].

7.2 Scenario-2: Ikeda Map-DQN

In this scenario, the Ikeda map with the DQN agent is used to generate several strong stream key bits with three different lengths (128, 192, and 256 bits). At the initialization of the Ikeda map, the state is a two-element array with random values between -1 and 1. The Ikeda map is implemented with the parameters illustrated in Table 4.

TABLE IV. IKEDA MAP-DQN PARAMETERS

Parameter	Value
a	0.4
b	6
$0 \leq u \lesssim 1$	

7.3 Scenario-3: Chua's Circuit -DQN

This scenario uses Chua's map with the DQN agent to produce a number of strong randomness stream key bits with three different lengths (128, 192, and 256 bits). **Initialize Chua's state as** a three-element array with random values between -1 and 1. Chua's circuit function updates the state vector [x, y, z] at each time step via numerical integration. The values of Chua's parameters used in this scenario are listed in Table 5.

TABLE V. CHUA'S CIRCUIT -DQN PARAMETERS

Parameter	Value
Alpha	15.6
beta	28
m0	-1.143
m1	-0.714

7.4 Scenario-4: Rössler Attractor-DQN

This scenario is the fourth scenario proposed in the DRLKG-Chaotic system for random stream key bit generation by using a Rössler Attractor with a DQN agent. The state is a three-element array with random values between -1 and 1. The Roessler Attractor function updates the state vector [*x, y, z]* at each time step using via numerical integration. The values of the Roessler Attractor parameters in this scenario illustrated in Table 6.

TABLE VI. RÖSSLER ATTRACTOR-DQN PARAMETERS

Parameter	Value
a	0.2
b	0.2
c	5.7

7.5 Scenario-5 Double Pendulum - DQN

In this scenario, a new efficient type of chaotic map is used with a DQN agent to generate various robust stream key bits. The state is a four-element array with random values between -1 and 1, and the double pendulum function updates the state vector at each time step using via numerical integration. The parameters parameter values are set as illustrated in Table 7.

TABLE VII. DOUBLE PENDULUM PENDULUM - DQN

Parameter	Value
theta1	Random state
theta2	Random state
omega1	Random state
omega2	Random state
m1	1.0
m2	1.0
L1	1.0
L2	1.0
g	9.81

Algorithm 1 represents the proposed algorithm for all the scenarios from 6.1--6.5.

Algorithm-1: Proposed DQN With Single Chaotic Maps Algorithm

Input:

- Key Length (KL): Number of key bits to generate (128, 192, 256).
- Chaotic Map Select (CMS): Choose specific type (Tent, Ikeda, Chua's, Rössler Attractor, double Pendulum).
- Parameters controlling each chaotic map (e.g., coefficients for Rössler Attractor, Chua's Circuit, etc.).
- Hyper parameters for DQN agent according to table 2.

Output: Generated Stream Key bits (GSK).

Step 1: Environment Initialization

1.1 Initialize chaotic states S ∈ R^n, where n represent the dimension for specific chaotic map as follows:

- Tent →$n = 1$
- Ikeda →$n = 2$
- Chua/Rössler →$n = 3$
- Double pendulum →$n = 4$

1.2 Create an environment with a dimensional state vector $S \in R^n$, then initialize this state with a random seed from a TRNG.

Step 2: Chaotic maps implementation

2.1 Apply the CMS with default parameter.

2.2 Update parameter based on the scalar action and clips.

Step 3: Training model

3.1 Instantiate DQN agent with policy of MLP, then call the CMS from step 2.

3.2 Apply an early-stopping function for periodically evaluate the current agent's performance by running an episode. 3.3 Accumulate the bit(s) generated in the key buffer. 3.4 Maintain the reward using NIST test.
Step 4: Obtain and store the GSK with high NIST test from taring agent.
Step 5: End

7.6 Scenario-6 DQN - 5-Maps Fusion

In this scenario, the randomness and complexity of the generated stream key bits for three different lengths (128, 192, and 256 bits) are increased via the fusion of the DQN with five maps. The training operation of the DQN method is implemented by updating to maximize the expected return via policy gradients, and then the loss function includes terms for the advantage and, optionally, entropy regularization. The update to minimize the mean squared error between the predicted value and the actual return and the loss function is typically value loss, which is the squared difference between the estimated value and the target value.

Figure 2 illustrates the DQN architecture, the. The input layer consists of multiple input nodes representing the state features. In addition, two fully connected hidden layers with ReLU activation functions between layers and each node connects connect to every node in the next layer. The output layer layer's Q value outputs for each action. The DQN algorithm represents a foundational approach in DR-L, enabling agents to learn value-based policies directly from high-dimensional inputs. By integrating deep neural networks with Q-Learning, techniques such as experience replay and fixed target networks can be introduced.

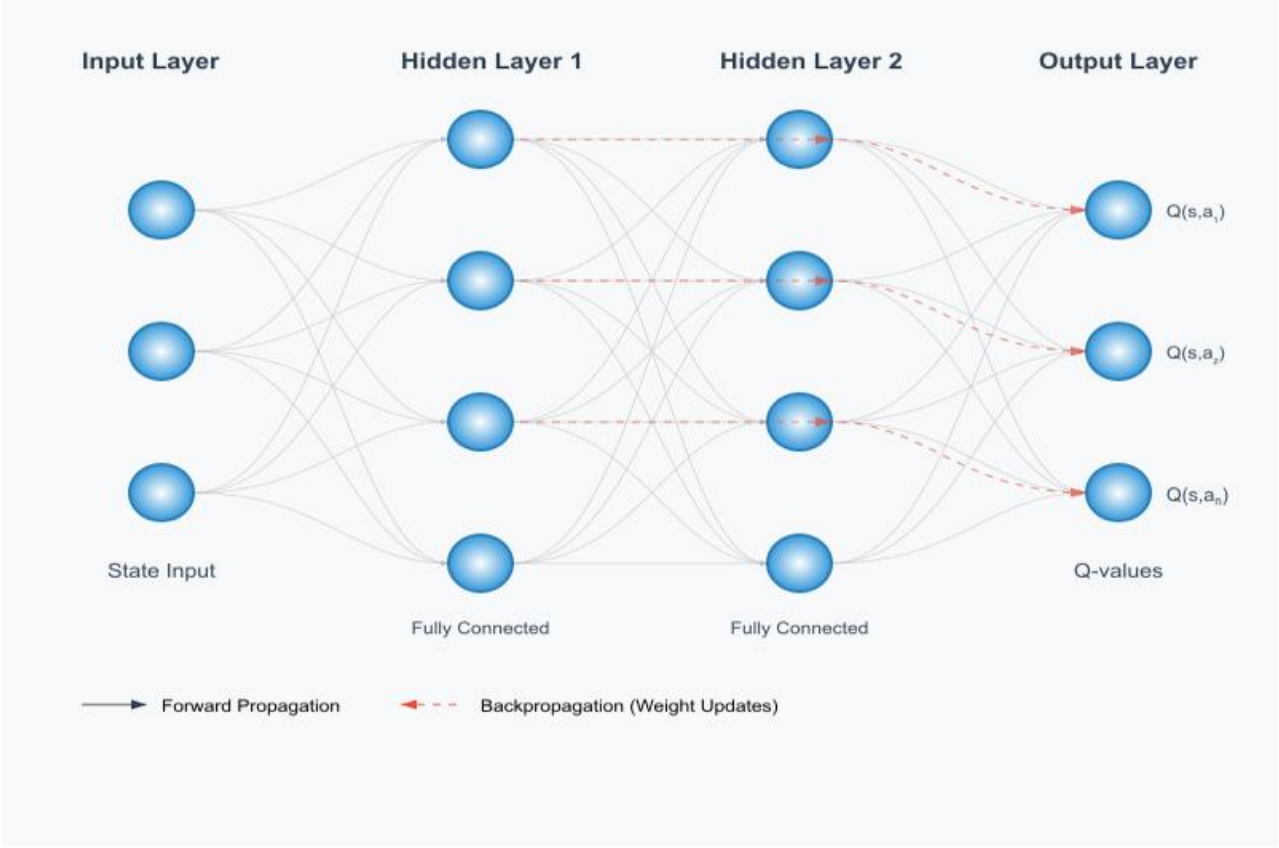


Fig. 2. DQN Architecture

This scenario implements a novel fusion approach that simultaneously incorporates all five chaotic maps (Tent, Ikeda, Chua's, Rössler, and Double Pendulum) through a spatial segmentation strategy. Unlike traditional mixing approaches, this

method maintains separate state vectors for each chaotic map, which evolve independently and in parallel throughout the key generation process. The fusion mechanism divides the cryptographic key into four distinct segments, with each segment influenced by a different chaotic system through probabilistic bit-flipping operations. Specifically, the chaotic values from each system are normalized to [0, 1] and used as flip probabilities for their respective key segments, where different regions of the key exhibit different dynamical behaviors. This approach preserves the unique characteristics of each chaotic map while introducing complex interdependencies through the DQN agent's learned actions that introduce enhanced cryptographic security through the synergistic combination of multiple sources of unpredictability. The continuous coevolution of all five systems throughout the generation process creates a rich, multidimensional chaotic landscape in which the reinforcement learning agent must navigate to optimize key quality. For more illustrations, see Figure 3.

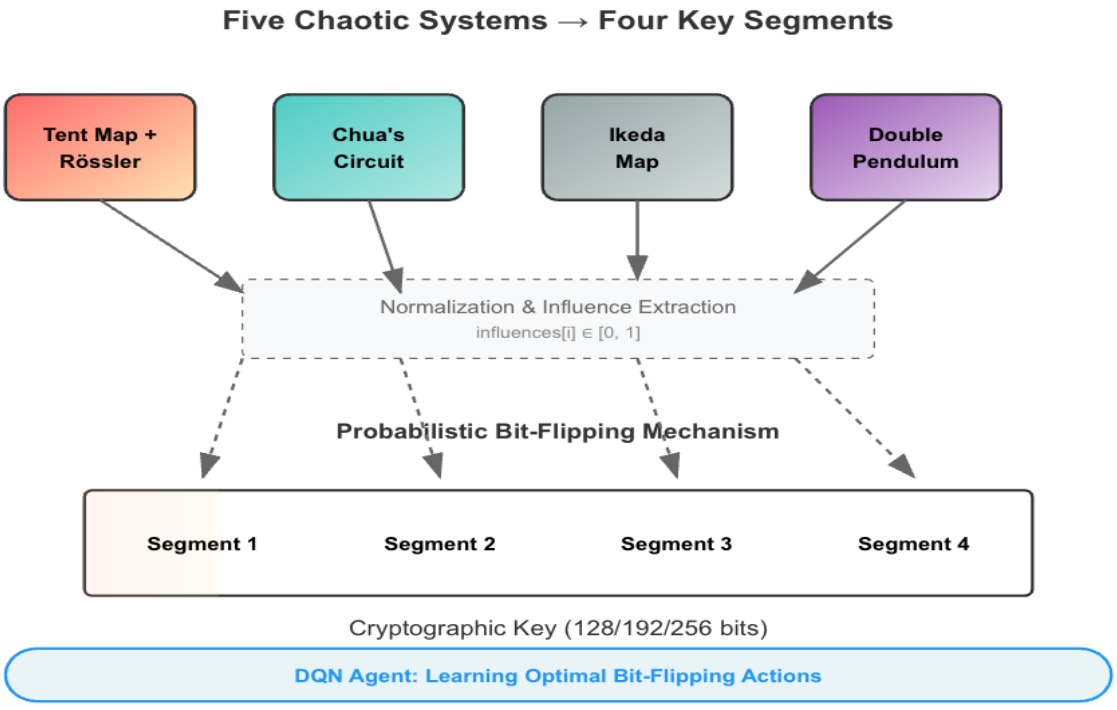


Fig. 3. DQN Agent Learning Optimal Bit-Flipping Actions

Additionally, the DQN parameters used for all the DRLKG-Chaotic system scenarios are listed in Table 8.

TABLE VIII. DQN HYPERPARAMETER VALUES

Hyper parameter		Values
1	learning rate	0.0001
2	buffer size	500000
3	learning starts	5000
4	batch size	128
5	gamma	0.99
6	exploration fraction	0.5
7	Exploration final episode	0.01
8	TAU	0.001
9	Target update interval	2000
10	Train Frequency	4
11	Gradient steps	1
12	Policy Net Architecture	[512,512]

The combination of the *DQN* and chaotic map systems underlies the procedure's capacity to generate pseudorandom keys that pass stringent statistical tests, thus demonstrating an interesting approach to cryptographic key generation or randomness extraction. Algorithm two represents the proposed algorithm for scenario 6.6.

Algorithm-2: Proposed DQN Fusion Algorithm

Input:

- Key Length (KL): Number of key bits generated (128, 192,256).
- Parameters controlling each chaotic maps (e.g., coefficients for Rössler Attractor, Chua's Circuit, etc.).
- Hyper parameters for DQN agent according to table 2.

Output: Generated Stream Key bits (GSK).

Step 1: Environment Initialization

1.1 Initialize chaotic states S ∈ R^n, where n represent the total dimensions for chaotic maps as follows:
- Tent →$n = 1$
- Ikeda →$n = 2$
- Chua/Rössler →$n = 3$
- Double pendulum →$n = 4$

1.2 Seed generation using True Random Number Generation (TRNG).

1.3 Create an environment with a 13-dimensional state vector$S \in R^{13}$, then initialize this state by generated values from step 2.1.

Step 2: Chaotic maps implementation

2.1 Apply Tent map, used the output of the Tent map as an external feeding for other chaotic maps (Ikeda, Chua's, Rössler and Double Pendulum).

2.2 Apply the remaining chaotic map in sequence to the corresponding portion of the state vector$S$.
- Ikeda map.
- Chua map.
- Rössler map.
- Double pendulum map.

Step 3: Training model

3.1 Instantiate a DQN agent with policy of MLP, then call the chaotic maps from step 2.

3.2 Apply Early Stopping function for periodically evaluate the current agent's performance by running an episode.

3.3 Accumulate the bit(s) generated in the key buffer.

3.4 Maintain the reward using NIST test.

Step 4: Obtain and store the GSK with high NIST test from training agent.

Step 5: End

8. RESULTS DISCUSSION

This section provides a thorough examination of the experimental results obtained through various evaluation methods used to assess the statistical performance of the DRLKG-Chaotic framework. The proposed system was evaluated via three distinct experimental implementations, which are detailed below:

8.1 Experiment-1

In this experiment, five standard maps are implemented separately (Tent, Ikeda, Chua's, Rössler and Double Pendulum) to generate the stream key bits with three different lengths (128, 192, and 256 bits), and the results of NIST tests of this experiment indicate that the produced stream key bits are generated with low levels of randomness and strength, as illustrated in Table 9.

TABLE IX. NIST P VALUES FOR STANDARD CHAOTIC MAPS: EXPERIMENT-1

Chaotic	Key Length	Entropy	Runs	cumulative sums Forward	cumulative sums Reverse	Block Frequency	Longest Run	Monobit	Key Strength
Tent	128	0.3752	0.1647103	0.3696	0.2658	0.5697	0.5390	0.2159	0.3571
	192	0.3627	0.04259904	0.9685	0.9898	0.9114	0.5351	0.7728	0.6546
	256	0.3762	0.1262	0.2672	0.1215	0.5984	0.7957	0.1336	0.3455
Ikeda	128	0.4480	0.4776	0.5490	0.5492	0.5654	0.4776	0.5026	0.5333
	192	0.4850	0.4944	0.5500	0.5293	0.5393	0.4509	0.5116	0.5057

	256	0.4570	0.4299	0.5195	0.5258	0.5913	0.4935	0.4967	0.5284
Chua's Circuit	128	0.4869	0.5354	0.4691	0.4562	0.4839	0.5057	0.4408	0.5028
	192	0.5323	0.5481	0.5606	0.5526	0.5074	0.4631	0.5449	0.5100
	256	0.4766	0.4687	0.5152	0.5153	0.4789	0.4980	0.4886	0.4825
Rössler Attractor	128	0.5076	0.4958	0.5768	0.5762	0.5038	0.4758	0.5735	0.5020
	192	0.5463	0.5384	0.5664	0.5672	0.5607	0.5610	0.5499	0.5547
	256	0.5144	0.5168	0.5195	0.5549	0.5063	0.4511	0.4978	0.5090
Double Pendulum	128	0.5216	0.4990	0.5454	0.5200	0.4494	0.5657	0.5263	0.4816
	192	0.4607	0.4691	0.4488	0.4960	0.4820	0.4877	0.4556	0.5013
	256	0.5168	0.4871	0.5057	0.4958	0.5274	0.4880	0.4797	0.5163

These results demonstrate that the entropy values for standard chaotic maps are relatively low, such that the Tent map is $0.3752 \leq H \leq 0.3762$, the Ikeda map is $0.4480 \leq H \leq 0.4870$, Chua's the Chua Circuit is $0.4766 \leq H \leq 0.4869$,, the Rössler Attractor is $0.5144 \leq H \leq 0.5076$, and the Double Pendulum is $0.5168 \leq H \leq 0.5216$. As illustrated in this table, the highest entropy value is 0.5463 with the Rössler Attractor, by attractor. In the same way of analysis, all the results obtained from the remaining NIST tests in this table are not optimal, suggesting insufficient randomness for modern cryptographic applications. This indicates that traditional implementations of these chaotic systems exhibit suboptimal randomness characteristics when evaluated against contemporary NIST statistical benchmarks.

8.2 Experiment-2

The second experiment involves applying the first five scenarios of the proposed DRLKG-Chaotic method by using five types of chaotic maps with a *DQN* agent for stream key bit generation.

8.3 Experiment-3

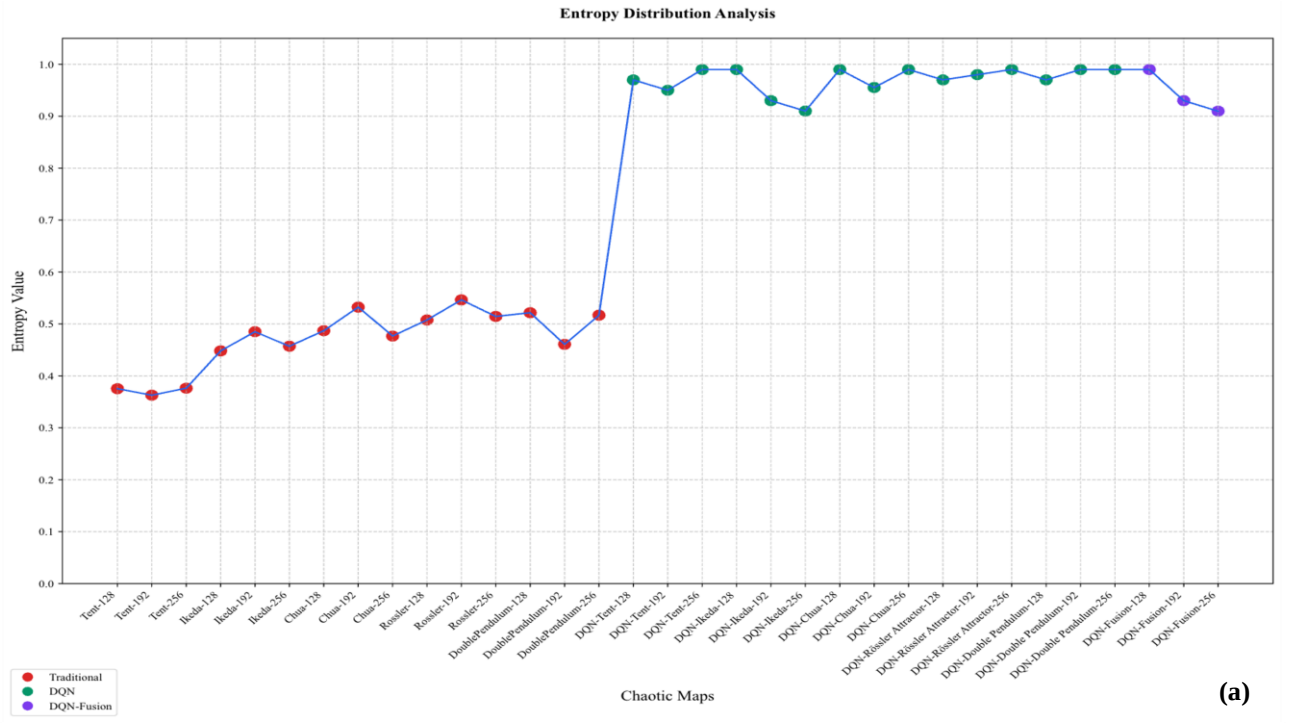
This experiment represents scenario-6 of the proposed DRLKG-Chaotic, where *DQN* fusion is used for all five chaotic maps. The results of NIST tests for the generated stream key bits of the last two experiments are illustrated in Table 10.

TABLE X. NIST P VALUES FOR THE DQN CASES (SCENARIO-1 TO SCENARIO-6)-EXPERIMENT-2

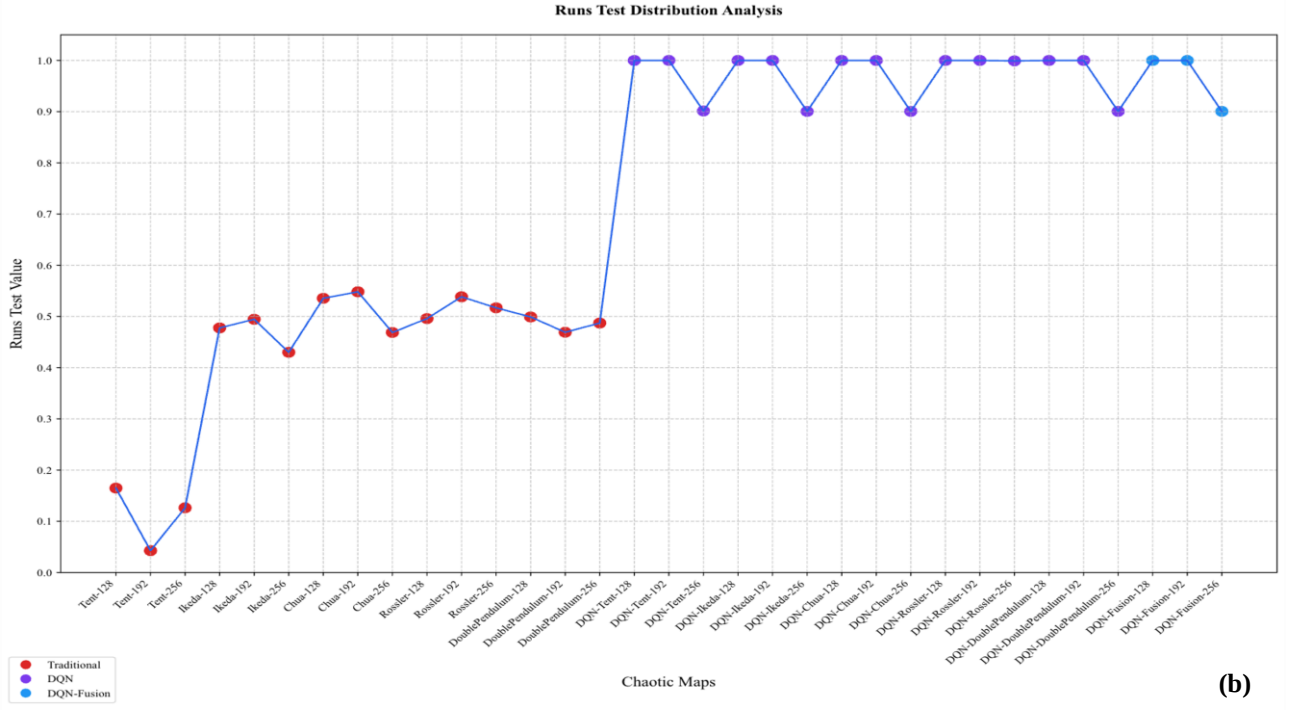
Case No.	Key Length	Entropy	Runs	cumulative sums Forward	cumulative sums Reverse	Block Frequency	Longest Run	Monobit	Key Strength
1	128	0.9735	1.0	0.9842	0.9842	0.9964	0.9088	1.0	0.9781
	192	0.9553	1.0	0.8202	0.8202	0.9988	0.8568	1.0	0.9216
	256	0.9970	0.9013	0.9742	0.9459	0.9700	0.8102	0.9005	0.9284
2	128	0.9928	1.0	0.9842	0.9842	0.7834	0.9934	1.0	0.9626
	192	0.9338	1.0	0.9316	0.9316	0.8832	0.9404	1.0	0.9458
	256	0.9097	0.9005	0.9742	0.9742	0.9700	0.9934	1.0	0.9603
3	128	0.9999	1.0	0.9493	0.9493	0.8335	0.9934	1.0	0.9608
	192	0.9553	1.0	0.8808	0.8808	0.8165	0.9145	1.0	0.9211
	256	0.9928	0.9005	0.9908	0.9908	0.9057	0.8843	1.0	0.9521
4	128	0.9735	1.0	0.9842	0.9842	0.9769	0.8355	1.0	0.9649
	192	0.9876	1.0	0.8808	0.8808	0.9643	0.9404	1.0	0.9506
	256	0.9943	0.9992	0.8580	0.9459	0.8705	0.8769	0.9005	0.9208
5	128	0.9735	1.0	0.9842	0.9842	0.9769	0.9404	1.0	0.9799
	192	0.9999	1.0	0.9686	0.9686	0.8514	0.9404	1.0	0.9613
	256	0.9981	0.9005	0.9998	0.9998	0.9834	0.7890	1.0	0.9530
6	128	0.9928	1.0	0.9493	0.9493	0.8795	0.6305	1.0	0.9145
	192	0.9338	1.0	0.9686	0.9686	0.9879	0.8894	1.0	0.9640
	256	0.9097	0.9005	0.9459	0.9459	0.9643	0.9693	1.0	0.9479

This table indicates an elevated level of robustness and randomness for the generated key stream bits. In addition, the *DQN* implementation has made significant improvements in randomness metrics, with P values consistently above 0.90 across all test parameters. Notable improvements in the entropy values, which range from 0.90940 to --0.99999, indicate that enhanced

randomness is suitable for contemporary cryptographic applications. The cumulative sums (both forward and reverse) show remarkable improvement, with values consistently above 0.90, demonstrating better statistical properties for cryptographic implementations. The *DQN* fusion case, which represents an ensemble approach, maintains high P values across all the parameters (averaging above 0.96), suggesting that robust randomness characteristics are suitable for modern security requirements. For more illustration about the results in Tables 9--10, as shown in Figure 4, the increase in the strength and randomness of the generated stream key bits of the proposed DRLKG-Chaotic system compared with those of standard maps is clear.



(a)



(b)

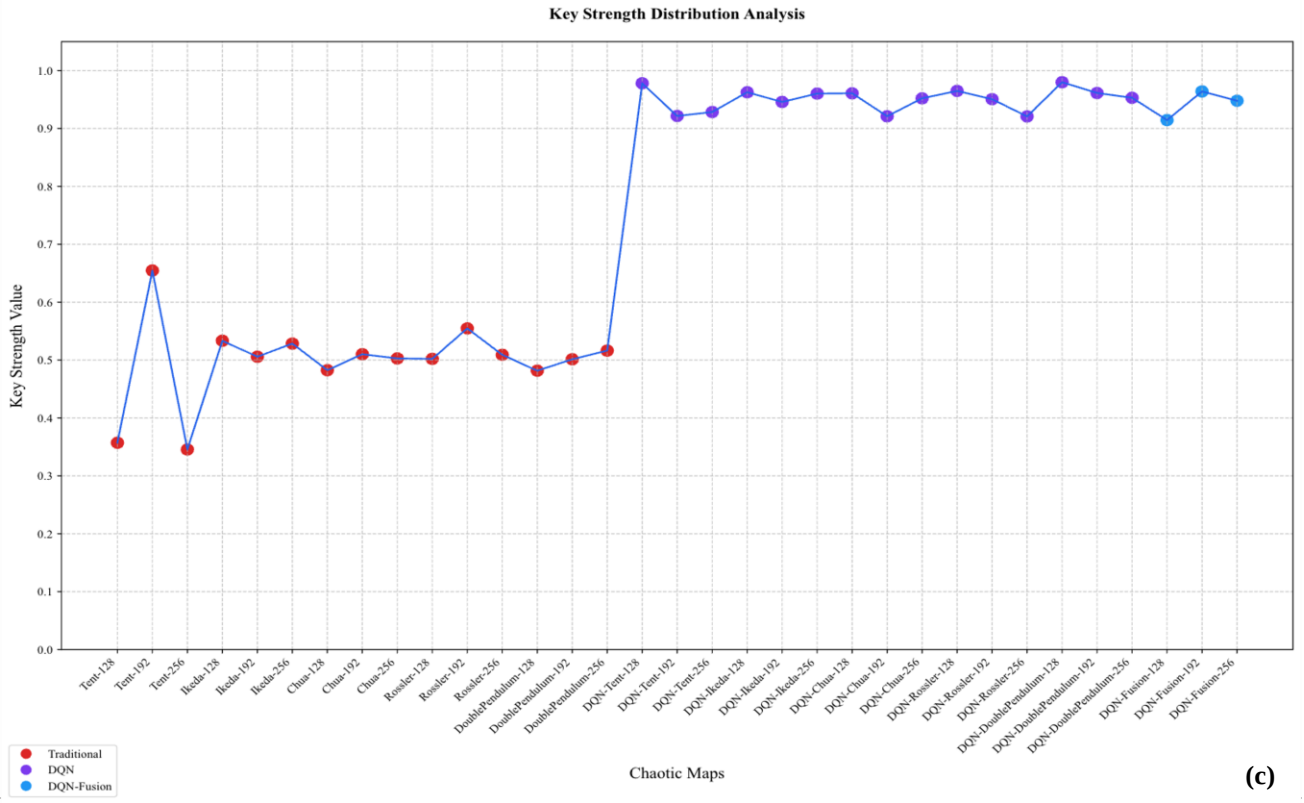


Fig. 4. NIST tests ((a) Entropy, (b) Run tests, (c) Key strength.

The key strength calculation is derived from the mean of the seven NIST statistical tests: entropy, run tests, cumulative forward, cumulative reverse, key frequency, longest run, and mono bit. This average enables comprehensive evaluation of the key across all tests; this approach is more dependable than individual tests are. Extended sequence generation enhances cryptographic strength by both minimizing vulnerability to statistical analysis and exponentially expanding the computational complexity required for adversarial search operations. Therefore, in this paper, three different lengths of generated stream key bits (128, 192, and 256 bits) are used. Table 11 lists the brute-force attack details for different lengths of generated stream key bits.

TABLE XI. BRUTE-FORCE ATTACK ATTEMPTS/SECOND WITH THE AVERAGE ESTIMATED TIME FOR CRACKING STREAM KEY BITS FOR (128, 192, AND 256 BITS) FOR SCENARIO-6

Key Length/bits	Key no.	Attempts/second	Estimated Cracking Time/years
128	1	17,508.56	3.08e+26
	2	17,631.79	3.06e+26
192	1	11,896.00	8.36e+45
	2	11,909.45	8.35e+45
256	1	8,991.56	2.04e+65
	2	9,003.12	2.04e+65

Figure 5 illustrates the power analysis simulation of the DQN fusion algorithm operating with three different key lengths (128, 192, and 256). For more illustration, the plot of the 256-bit key depicts the simulated power consumption over five hundred operations. The parenthetical notation “based on masked data” is employed to denote that the measurement is derived from the randomized internal state rather than the final output. Highly erratic fluctuations between approximately 135 mW and 220 mW were exhibited by the trace, with no discernible repeating patterns, spikes, or predictable behaviour, resulting in a noise-like appearance. This profile is consistent with the intended outcome for side-channel resistant designs, as masking countermeasures designed to render power consumption uncorrelated with secret data. The noise-like characteristic of the trace is indicative of an environment in which exploitable patterns for key recovery or internal state

inference cannot be extracted. Consequently, the effectiveness of the masking scheme within the DQN fusion algorithm for scenario-6 is substantiated, and robust protection against power analysis attacks is thereby provided.

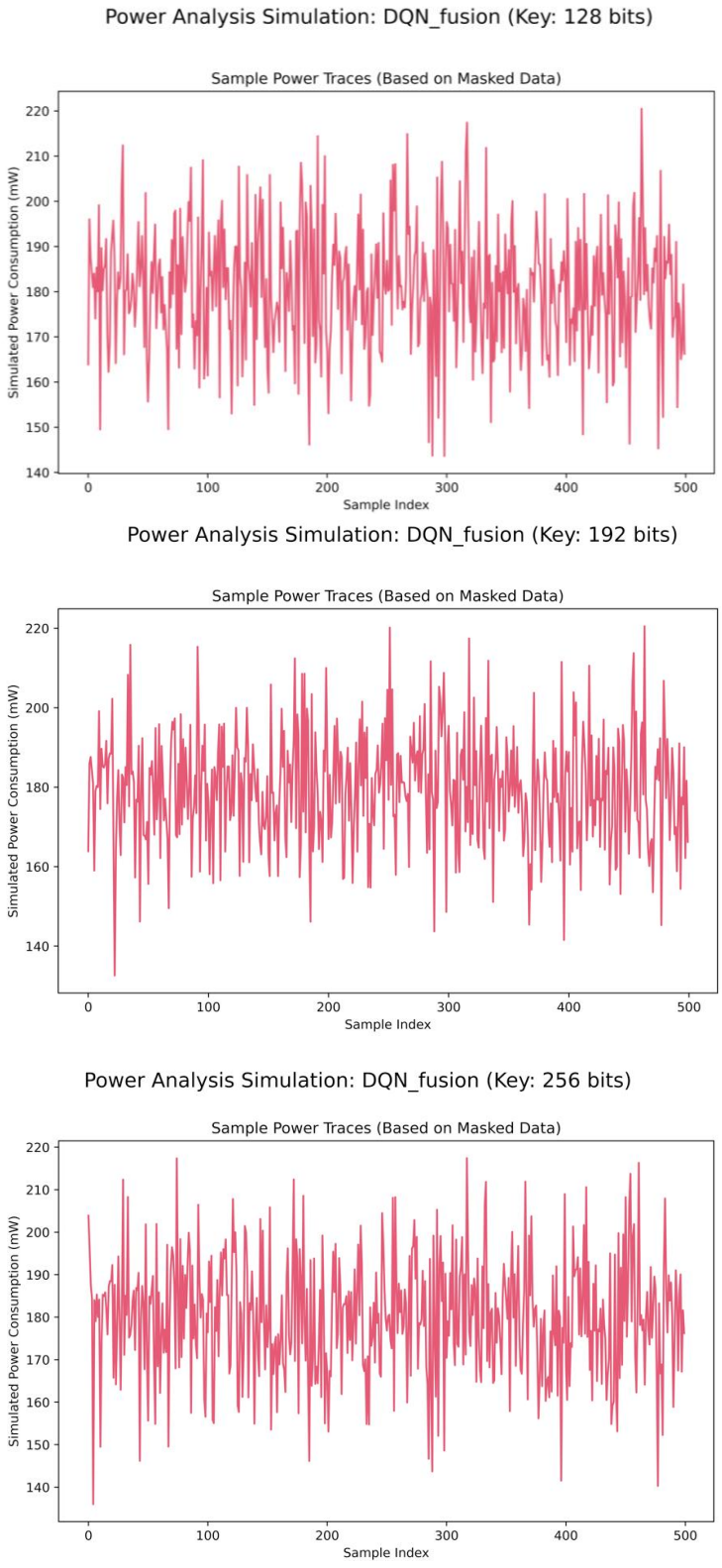


Fig. 5. side-channel resistant with two different key lengths of scenario-6

Figure 6 presents the power analysis summary chart, through which the resistance of the DQN_fusion algorithm to power-based side-channel attacks is assessed. The chart condenses intricate correlation data from prior traces into a single vulnerability score. This score ranges from 0.0 (indicating the absence of correlation and thus ideal security) to 1.0 (indicating perfect correlation and complete insecurity)—was and was determined to be 0.088. The low magnitude of this value, highlighted by a green bar, signifies successful fulfilment of the defined security threshold. This result implies that the simulated analysis revealed virtually no meaningful correlation between power consumption and algorithmic outputs, thereby confirming the efficacy of the masking countermeasure. These findings provide compelling evidence of the DQN fusion algorithm's robustness against simulated power analysis attacks. The inability to infer secret keys or internal states from observed power traces reinforces the effectiveness of the employed countermeasures and underscores the algorithm's suitability for deployment in environments requiring strong side-channel resistance.

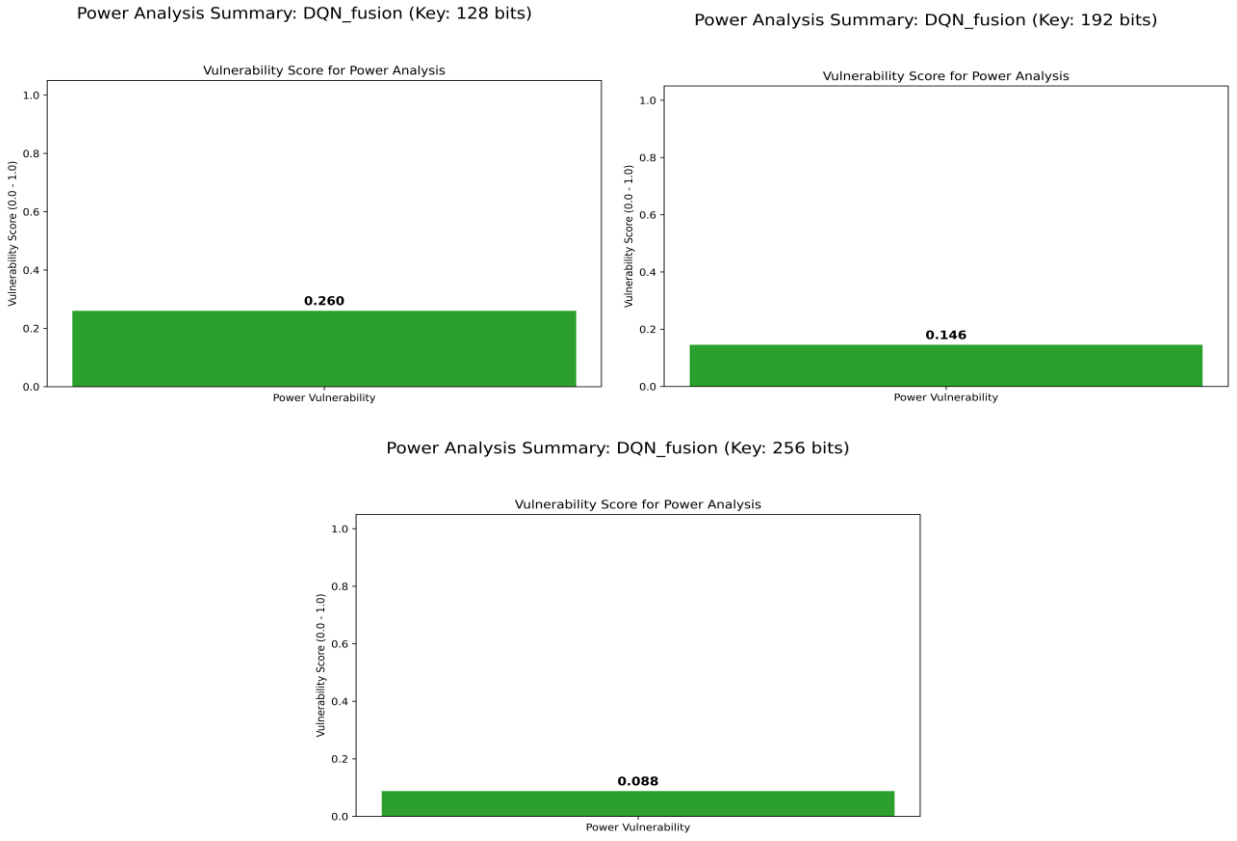


Fig. 6. side-channel vulnerability score with three different key lengths of scenario-6

Figure 7 presents the outcome of a timing attack executed on a system. For illustration, the plot of the 256-bit key indicates strong resistance to this method of analysis. In a timing attack, the duration of each cryptographic operation is recorded; if the time required differs systematically when a guessed key bit is correct versus incorrect, an attacker can iteratively recover the entire key. In the displayed plot, the horizontal axis denotes individual bit positions (1–256), whereas the vertical axis represents the operation execution time. Red markers correspond to measurements collected when the guessed bit value matches the true key bit, and light-blue markers correspond to measurements collected when the guess is incorrect. Effective timing attacks produce two distinct clusters of points, yet the red and light-blue markers in Figure 4 are extensively interleaved and distributed across the same temporal range. This intermingling, resembling random noise, indicates that no reliable timing differential exists. These findings confirm that execution time does not leak significant information regarding secret key bits, thereby demonstrating that the tested algorithm effectively mitigates timing-based side-channel vulnerability.

To prove the high level of randomness, robustness, and complexity of the generated stream key bits, several addition tests are applied to the selected stream key bits to shorten the results of scenario-6. Table 12 illustrates the autocorrelation of the stream key bits generated from scenario -6. The autocorrelation analysis of keys generated through the fusion implementation demonstrates exceptional statistical properties crucial for cryptographic applications. Across all key lengths (128, 192, and 256 bits), the average autocorrelation values remain remarkably close to zero, ranging from -

0.008526 to -0.003259, indicating minimal temporal dependencies between bits. The balanced distributions of the maximum (0.180124--0.375000) and minimum (-0.268293--0.181287) autocorrelation values around zero further confirm the absence of systematic patterns. Most significantly, the percentage of significant autocorrelations remains exceptionally low at 0--2%, implying that 98--100% of lag correlations fall within the expected range for truly random sequences. This near-ideal autocorrelation behavior validates the effectiveness of the spatial segmentation approach, where the simultaneous influence of five distinct chaotic systems successfully eliminates exploitable temporal patterns. The consistency of these results across different key lengths demonstrates that the fusion method scales effectively while maintaining cryptographic quality.

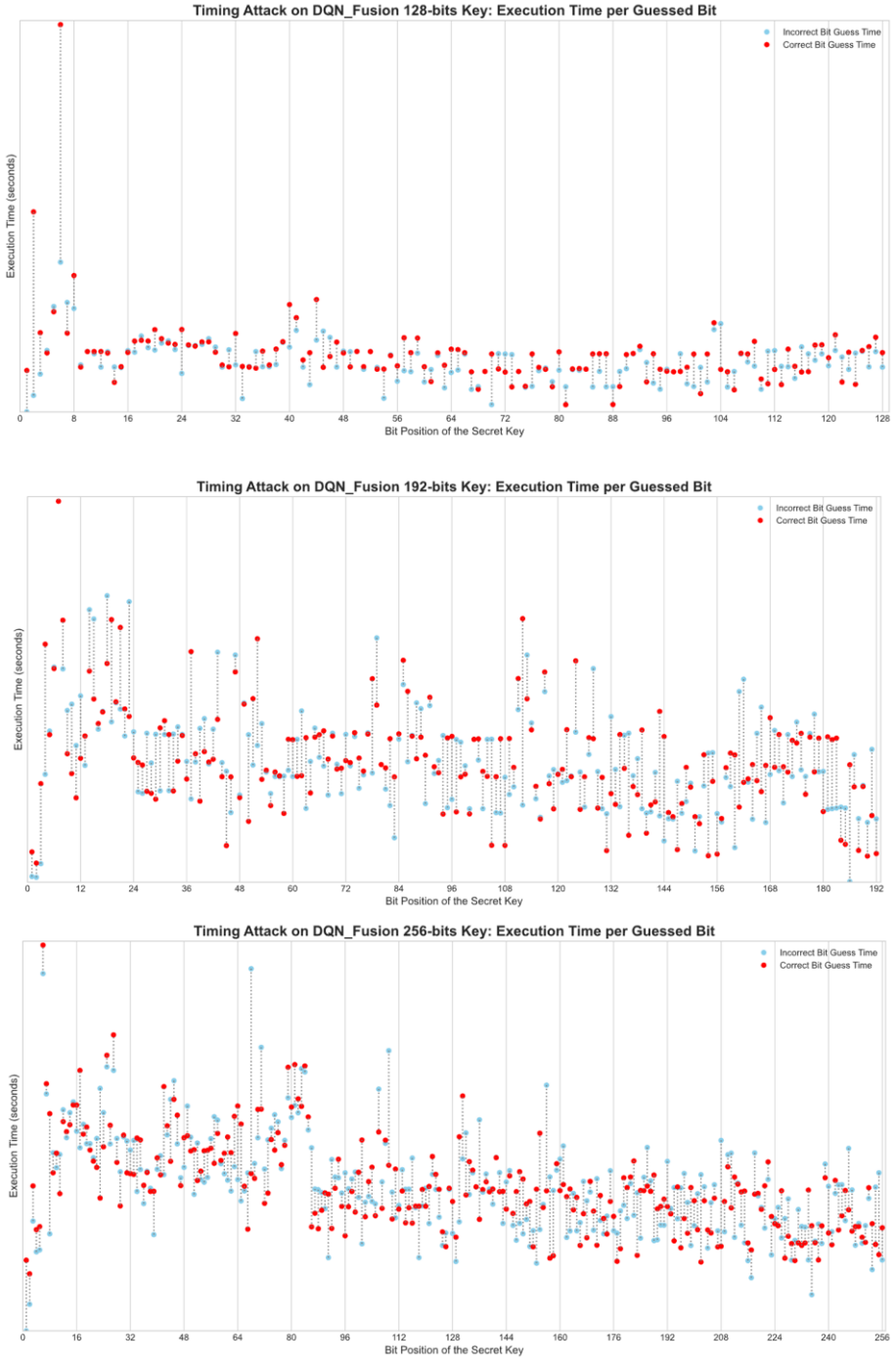
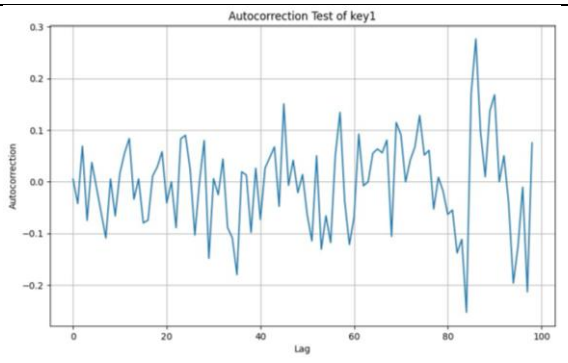
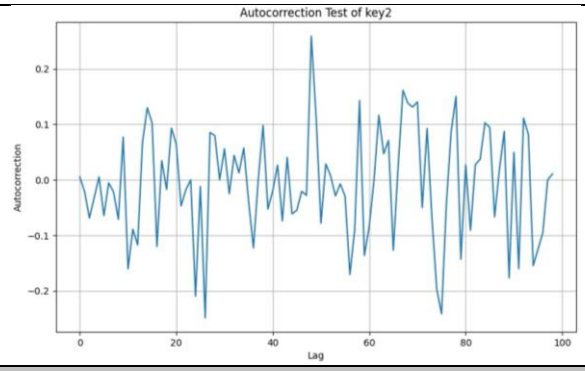
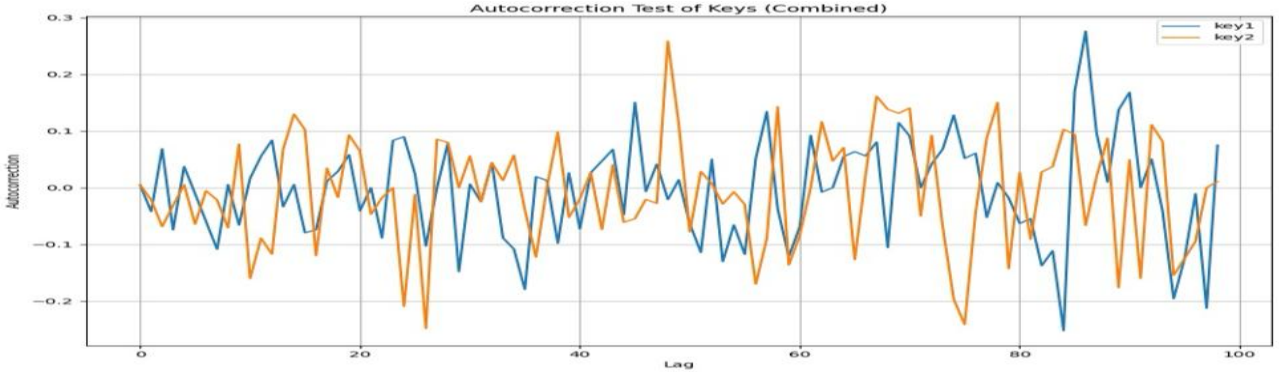
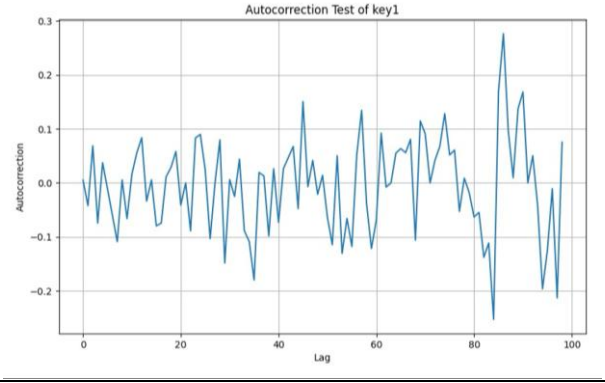
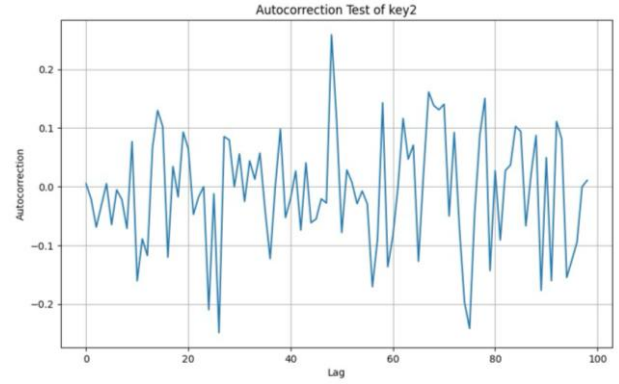


Fig. 7. Timing attack for three different key lengths of scenario-6

TABLE XII. AUTOCORRELATION OF STREAM KEY BITS IN SCENARIO-6

Key Length/bits	Key no.	Average Autocorrection	Maximum Autocorrection	Minimum Autocorrection	Significant Autocorrections%
128 bits	1	-0.003259	0.375000	-0.261538	2%
	2	-0.007065	0.250000	-0.268293	0%
192 bits	1	-0.008526	0.185185	-0.197802	0%
	2	-0.006370	0.207207	-0.200000	0%
256 bits	1	-0.005748	0.295597	-0.184358	1%
	2	-0.006021	0.180124	-0.181287	0%

For more illustration, Figure 8 shows the autocorrelation such that the third column represents the autocorrelation between two selected stream key bits for each scenario with three different key lengths (128, 192, and 256).

Autocorrelation of Key-1	**Autocorrelation of Key-2**
128-bit	
	
Autocorrelation between Key-1 and Key-2	
	
192-bit	
Autocorrelation of Key-1	**Autocorrelation of Key-2**
	

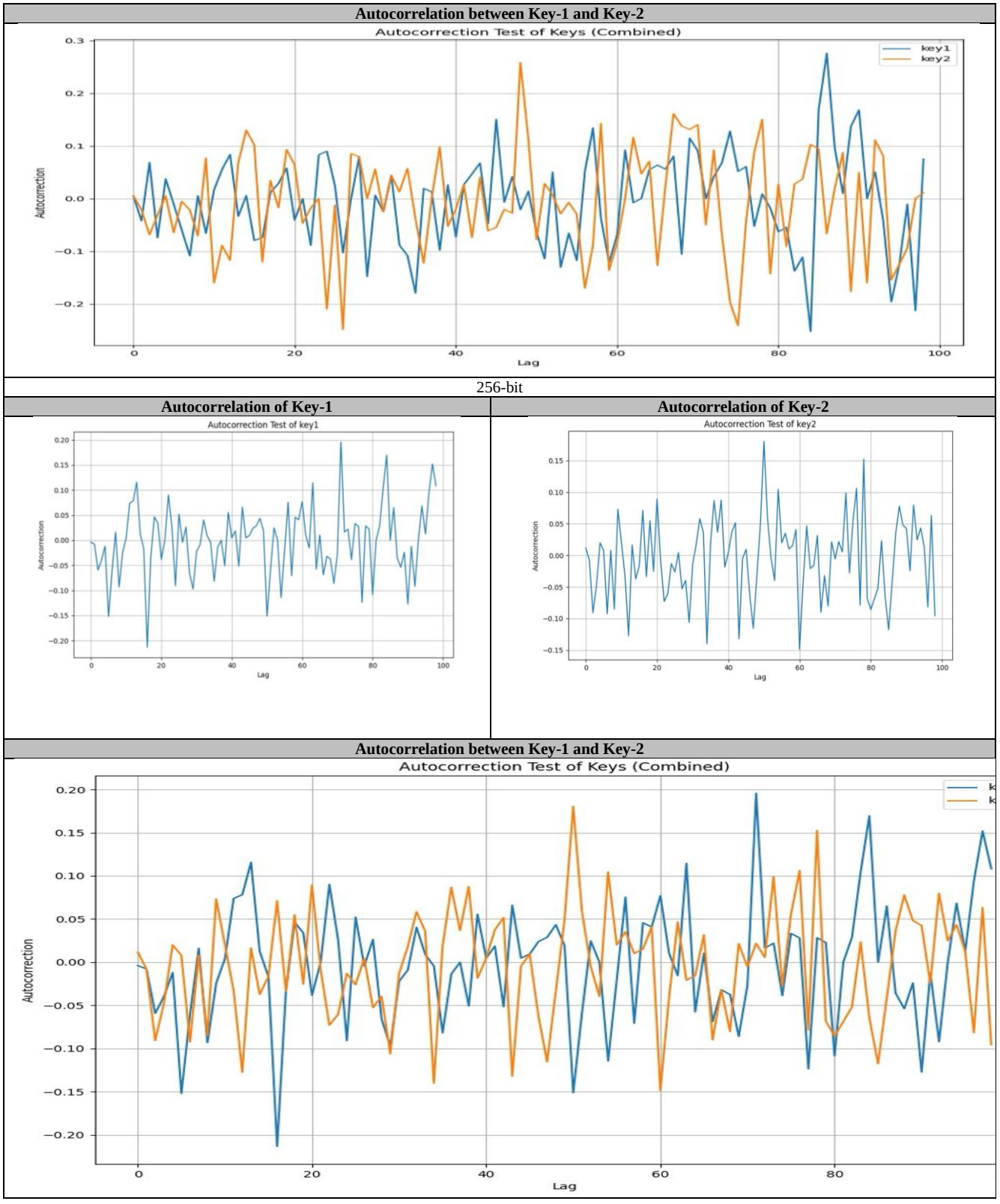


Fig. 8. Autocorrelation between two Keys with three different Lengths in scenario-6

Another indication of the strength of the generated stream key bits of the proposed DRLKG-Chaotic is the use of cross-correlation. From the results of Table 13, there is no robust evidence of correlation for the generated keys.

TABLE XIII. CROSS-CORRELATIONS FOR ALL DRLKG-CHAOTIC SCENARIOS

- scenario No.	Length of keys		
	128-bit	192-bit	256-bit
1.	-0.062500	0.000000	-0.054689
2.	0.093750	0.000000	-0.101566
3.	-0.031250	0.020833	-0.031250
4.	0.000000	0.104167	-0.007813
5.	0.000000	0.000000	-0.085940
6.	-0.093750	0.104167	-0.007813

Figure 9 shows that the diagonal elements (diagonal entries) (key1 vs. key1 and key2 vs. key2) are 1.0, indicating perfect correlation. This is expected because a variable is always perfectly correlated with itself. The off-diagonal elements (off-diagonal entries) (key1 vs. key2 and key2 vs. key1) have varying values across different bit lengths, indicating different correlation strengths. This map uses a color gradient where red represents a positive correlation, blue represents a negative correlation, and lighter shades represent weaker correlations. The diagonal is deep red (indicating 1.0), whereas the off-diagonal elements vary. In scenario (a), the off-diagonal elements have a value of -0.094, indicating a slight negative correlation. In scenario (b), they have a value of 0.1, indicating a weak positive correlation. In scenario (c), they have a value of -0.0078, indicating an almost negligible negative correlation. When Lag = 0, this indicates that the correlation analysis does not involve time shifting; it is a straightforward measure of simultaneous correlation between the variables. The decreasing absolute correlation values as the bit length increases (from |0.094| in 128 bits to |0.0078| in 256 bits) suggest that key1 and key2 become increasingly independent as the bit length increases in the DQN algorithm implementation. This finding demonstrates that higher bit lengths may provide better statistical independence properties, which is advantageous for security and randomness in the chaotic map behaviour with the DQN algorithm.

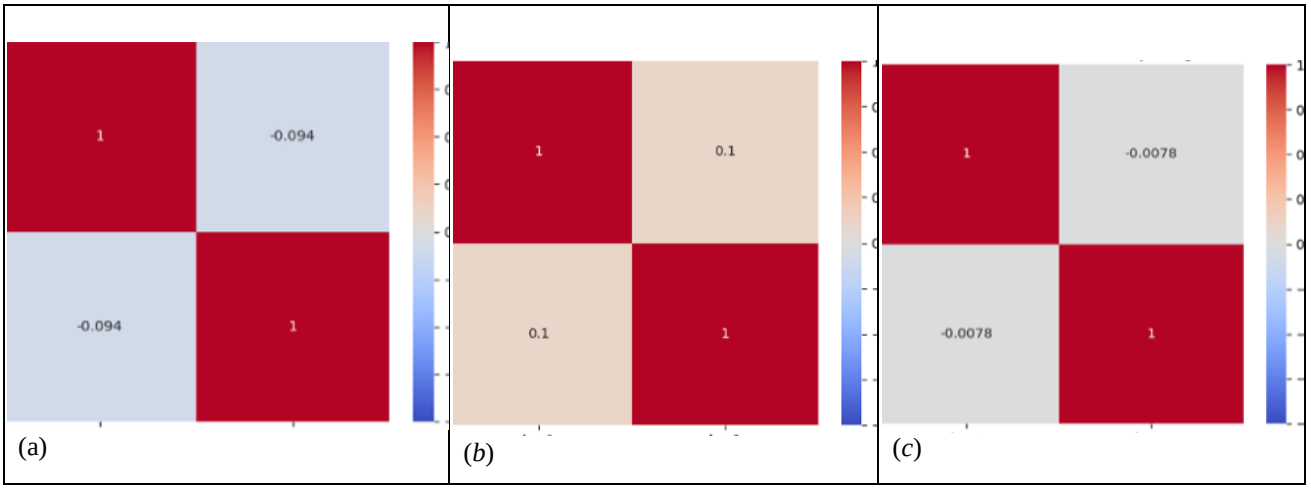


Fig. 9. Cross-correlation matrix of two Keys with three different lengths of scenario-6: (*a*) 128 bits, (*b*) 192 bits, (*c*) 256 bits

Table 14 shows the P value of the DFT of the generated stream key bits of the proposed DRLKG-Chaotic.

TABLE XIV. P VALUE FOR THE DISCRETE FOURIER TRANSFORM TEST (DFT) FOR SCENARIO-6

Key Length/bits	**Key no.**	**P value**	**Threshold (T)**
128 bits	1	0.301898	19.5820
	2	0.108294	19.5820
192 bits	1	0.399269	23.9829
	2	0.707932	23.9829
256 bits	1	0.871131	27.6931
	2	0.570188	27.6931

Figure 10 represents a DFT, which analyses the frequency components of stream key bits. This chart evaluates the frequency-domain characteristics of a stream key bit. The x-axis represents the frequency components, and the y-axis represents the magnitude of the Fourier coefficients for each frequency. The blue line shows the magnitude of the DFT for each frequency component. These represent how much each frequency contributes to the overall signal, whereas the horizontal red dashed line indicates a threshold level, labelled "Threshold T." used as a benchmark for evaluating significant frequency components. The randomness of the data represented in this figure was assessed by analysing the distribution of frequencies and their magnitudes via DFT. Cryptographic randomness testing, such as DFT, is used to detect periodic structures.

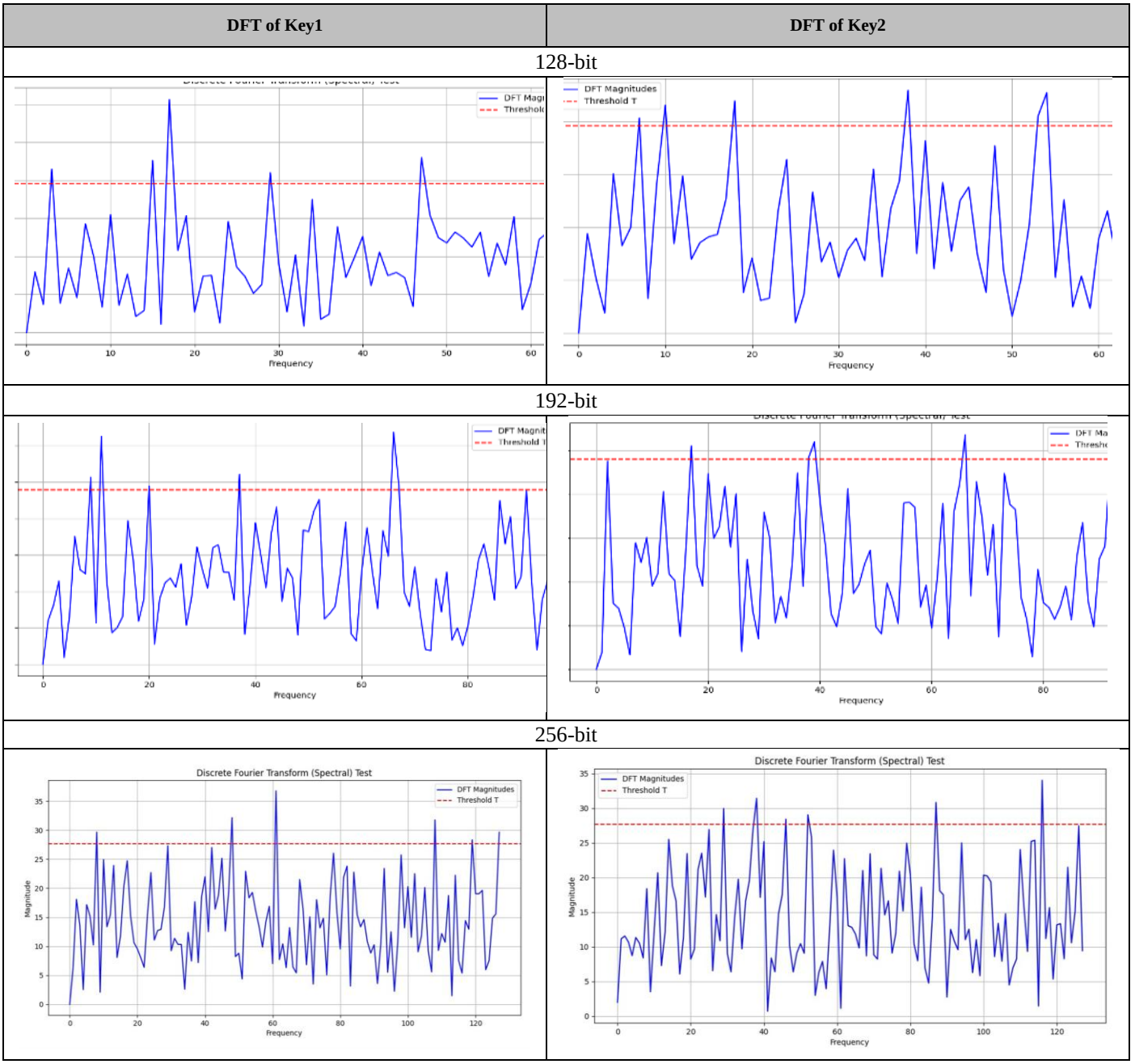


Fig. 10. Discrete Fourier transform (DFT) of two keys with three different lengths in scenario-6

Table 15 shows the comparison between the obtained entropy values of the proposed models and the results of related works. From this table, it is clear that the entropy values resulting from the proposed models of this paper are superior to the values of the results obtained from the previous work.

TABLE XV. COMPARISON BETWEEN THE PROPOSED SYSTEM AND PREVIOUS METHODS

Models of the Proposed System (DRLKG-Chaotic)			Related Work	Entropy
Chaotic	**Key Length**	**Entropy**	[21]	0.875
			[25]	0.799
			[22]	0.948
			[23]	0.701
Tent	128	0.3752	[24]	0.612
	192	0.3627	[26]	≈ 7.9971
	256	0.3762	[27]	≈7.99
Ikeda	128	0.4480		
	192	0.4850	[28]	0.85-1.2
	256	0.4570		
Chua's Circuit	128	0.4869		
	192	0.5323		
	256	0.4766		
Rössler Attractor	128	0.5076		
	192	0.5463		
	256	0.5144		
Double Pendulum	128	0.5216		
	192	0.4607		
	256	0.5168		

9. CONCLUSIONS

This paper introduces six scenarios designed to increase the robustness of key stream bits against potential attacks. The proposed DRLKG-Chaotic system incorporates six distinct scenarios utilizing five chaotic maps (Tent, Ikeda, Chua's, Rössler, and Double Pendulum) integrated with the DQN algorithm. DRLKG generates stream key bits of varying lengths (128, 192, and 256 bits), all of which undergo rigorous validation through multiple randomness assessment methodologies. Bit streams that successfully meet all seven statistical criteria demonstrate viable cryptographic properties and statistically classified security. These results confirm the resilience and strength of the key sequences generated across all the experimental scenarios within our DRLKG-Chaotic framework. The successful NIST test results provide evidence of the unpredictability of the generated key stream bits. Analysis of the data presented in the referenced tables and figures—encompassing NIST tests, brute-force attack resistance, autocorrelation (AC), cross-correlation (CC), and discrete Fourier transform (DFT) evaluations—demonstrates significant cryptographic strength improvements through the integration of DRL with the five chaotic map types. These proposed scenarios offer substantial enhancements over standard chaotic systems. The results indicate that the six proposed scenarios provide superior performance for high-security applications, representing significant advancements over standard mapping techniques. In addition, the generated stream key bits from the proposed scenarios are resistant to brute-force attacks, side-channel attacks, and timing attacks. The proposed DQN-fusion generator is practical for IoT nodes and other low-resource devices because all training is carried out off-device. The endpoint stores only a small DRL agent and runs lightweight, imposing minimal CPU, RAM, and energy overheads. Consequently, even microcontrollers without floating-point units can deliver the cryptographic strength demonstrated.

10. FUTURE WORK

While the DRLKG-Chaotic framework demonstrates strong security when five chaotic maps (Tent, Ikeda, Chua's, Rössler, Double Pendulum) are integrated with the DQN for 128/192/256-bit keys, these extensions warrant investigation:

1. Larger Key Sizes: Exploring key (512 bits, 1024 bits, and more) implementations could counter emerging quantum computing threats and serve other encryption algorithms.
2. Enhanced Chaotic Models: Exploring the integration of DRL with hyper chaotic systems to amplify randomness (Hybrid DRL Approaches).
3. DRL Algorithm Alternatives: Use other types of chaotic maps and DRL algorithms, such as A3C, TRPO, and PPO, and experiment with the fusion between these several types of chaotic maps and ultra-complex DRL.

Conflicts of interest
The authors declare that they have no conflicts of interest.

Funding
There was no funding for this research study.
Acknowledgement
None

References

[1]M. S. Rathore *et al.*, "A novel trust-based security and privacy model for Internet of Vehicles using encryption and steganography," *Computers and Electrical Engineering*, vol. 102, Sep. 2022, doi: 10.1016/j.compeleceng.2022.108205.
[2]A. A. Salih, Z. A. Abdulrazaq, and H. G. Ayoub, "Design and Enhancing Security Performance of Image Cryptography System Based on Fixed Point Chaotic Maps Stream Ciphers in FPGA," *Baghdad Science Journal*, vol. 21, no. 5 SI, pp. 1754–1764, 2024, doi: 10.21123/bsj.2024.10521.
[3]R. Naik and U. Singh, "Secured 6-Digit OTP Generation using B-Exponential Chaotic Map," 2021. [Online]. Available: www.ijacsa.thesai.org
[4]R. B. Prajapati and S. D. Panchal, "Enhanced Approach To Generate One Time Password (OTP) Using Quantum True Random Number Generator (QTRNG)," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 279–292, 2024, doi: 10.12785/ijcds/150122.
[5]U. Zia, M. McCartney, B. Scotney, J. Martinez, and A. Sajjad, "A novel pseudo-random number generator for IoT based on a coupled map lattice system using the generalised symmetric map," *SN Appl Sci*, vol. 4, no. 2, Feb. 2022, doi: 10.1007/s42452-021-04919-4.
[6]L. Baldanzi *et al.*, "Cryptographically secure pseudo-random number generator IP-core based on SHA2 algorithm," *Sensors (Switzerland)*, vol. 20, no. 7, Apr. 2020, doi: 10.3390/s20071869.
[7]M. Farajallah, M. Abutaha, M. Abu Joodeh, O. Salhab, and N. Jweihan, "PSEUDO RANDOM NUMBER GENERATOR BASED ON LOOK-UP TABLE AND CHAOTIC MAPS," *J Theor Appl Inf Technol*, vol. 31, p. 20, 2020, [Online]. Available: www.jatit.org
[8]M. D. Al-Hassani, "A Novel Technique for Secure Data Cryptosystem Based on Chaotic Key Image Generation," *Baghdad Science Journal*, vol. 19, no. 4, pp. 905–913, 2022, doi: 10.21123/bsj.2022.19.4.0905.
[9]S. A. S. Hussien, B. N. Al Din Abed, and K. A. Ibrahim, "Encrypting Text Messages via Iris Recognition and Gaze Tracking Technology," *Mesopotamian Journal of CyberSecurity*, vol. 5, no. 1, pp. 90–103, Jan. 2025, doi: 10.58496/MJCS/2025/007.
[10]M. M. Hoobi, "Multilevel Cryptography Model using RC5, Twofish, and Modified Serpent Algorithms," *Iraqi Journal of Science*, vol. 65, no. 6, pp. 3434–3450, 2024, doi: 10.24996/ijs.2024.65.6.37.
[11]N. H. M. Ali, M. M. Hoobi, and D. F. Saffo, "Development of Robust and Efficient Symmetric Random Keys Model based on the Latin Square Matrix," *Mesopotamian Journal of CyberSecurity*, vol. 4, no. 3, pp. 203–215, 2024, doi: 10.58496/MJCS/2024/023.
[12]I. A. Abdulmunem and M. M. Hoobi, "Enhanced DES Algorithm Using Efficient Classical Algorithm," *Iraqi Journal of Science*, vol. 65, no. 12, pp. 7251–7275, 2024, doi: 10.24996/ijs.2024.65.12.37.
[13]N. E. El-Meligy, T. O. Diab, A. S. Mohra, A. Y. Hassan, and W. I. El-Sobky, "A Novel Dynamic Mathematical Model Applied in Hash Function Based on DNA Algorithm and Chaotic Maps," *Mathematics*, vol. 10, no. 8, Apr. 2022, doi: 10.3390/math10081333.
[14]A. Zellagui, N. Hadj-Said, and A. Ali-Pacha, "A new hash function inspired by sponge construction using chaotic maps," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 26, no. 2, pp. 529–559, 2023, doi: 10.1080/09720529.2021.1961900.
[15]A. T. Maolood, E. K. Gbashi, and E. S. Mahmood, "Novel lightweight video encryption method based on ChaCha20 stream cipher and hybrid chaotic map," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 5, pp. 4988–5000, Oct. 2022, doi: 10.11591/ijece.v12i5.pp4988-5000.
[16]M. Alawida, J. Sen Teh, A. Mehmood, A. Shoufan, and W. H. Alshoura, "A chaos-based block cipher based on an enhanced logistic map and simultaneous confusion-diffusion operations," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 8136–8151, 2022, doi: 10.1016/j.jksuci.2022.07.025.
[17]J. Liu, Y. Wang, Q. Han, and J. Gao, "A Sensitive Image Encryption Algorithm Based on a Higher-Dimensional Chaotic Map and Steganography," *International Journal of Bifurcation and Chaos*, vol. 32, no. 01, p. 2250004, 2022, doi: 10.1142/S0218127422500043.

[18]M. Bhandari, S. Panday, C. P. Bhatta, and S. P. Panday, "Image Steganography Approach Based Ant Colony Optimization with Triangular Chaotic Map," in *Proceedings of 2nd International Conference on Innovative Practices in Technology and Management, ICIPTM 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 429–434. doi: 10.1109/ICIPTM54933.2022.9753917.
[19]K. Wang, T. Gao, D. You, X. Wu, and H. Kan, "A secure dual-color image watermarking scheme based 2D DWT, SVD and Chaotic map," *Multimed Tools Appl*, vol. 81, no. 5, pp. 6159–6190, 2022, doi: 10.1007/s11042-021-11725-y.
[20]M. Irfan and M. A. Khan, "Cryptographically Secure Pseudo-Random Number Generation (CS-PRNG) Design using Robust Chaotic Tent Map (RCTM)," Aug. 2024, [Online]. Available: http://arxiv.org/abs/2408.05580
[21]R. B. Naik and U. Singh, "A Review on Applications of Chaotic Maps in Pseudo-Random Number Generators and Encryption," *Annals of Data Science*, vol. 11, no. 1, pp. 25–50, 2022, doi: 10.1007/s40745-021-00364-7.
[22]B. V Nair, V. V S, S. S. Muni, and A. Durdu, "Deep Learning and Chaos: A combined Approach To Image Encryption and Decryption," Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.16792
[23]E. Kopets, V. Rybin, O. Vasilchenko, D. Butusov, P. Fedoseev, and A. Karimov, "Fractal Tent Map with Application to Surrogate Testing," *Fractal and Fractional*, vol. 8, no. 6, Jun. 2024, doi: 10.3390/fractalfract8060344.
[24]D. F. M. Oliveira, "Mapping Chaos: Bifurcation Patterns and Shrimp Structures in the Ikeda Map," Aug. 2024, [Online]. Available: http://arxiv.org/abs/2408.11254
[25]W. Zhao, Z. Chang, C. Ma, and Z. Shen, "A Pseudorandom Number Generator Based on the Chaotic Map and Quantum Random Walks," *Entropy*, vol. 25, no. 1, Jan. 2023, doi: 10.3390/e25010166.
[26]S. Subathra and V. Thanikaiselvan, "Enhanced security for medical images using a new 5D hyper chaotic map and deep learning based segmentation," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-04906-4.
[27]C. S. Devi and R. Amirtharajan, "A novel 2D MTMHM based key generation for enhanced security in medical image communication," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-10485-1.
[28]S. B. N. Premakumari, G. Sundaram, M. Rivera, P. Wheeler, and R. E. P. Guzmán, "Reinforcement Q-Learning-Based Adaptive Encryption Model for Cyberthreat Mitigation in Wireless Sensor Networks," *Sensors*, vol. 25, no. 7, Apr. 2025, doi: 10.3390/s25072056.
[29]J. Ding, K. Chen, Y. Wang, N. Zhao, W. Zhang, and N. Yu, "Discop: Provably Secure Steganography in Practice Based on 'Distribution Copies,'" 2023, doi: 10.1109/SP46215.2023.00155.
[30]A. Daoui, M. Yamni, S. A. Chelloug, M. A. Wani, and A. A. A. El-Latif, "Efficient Image Encryption Scheme Using Novel 1D Multiparametric Dynamical Tent Map and Parallel Computing," *Mathematics*, vol. 11, no. 7, Apr. 2023, doi: 10.3390/math11071589.
[31]N. F. Hassan, A. Al-Adhami, and M. S. Mahdi, "Digital Speech Files Encryption based on Hénon and Gingerbread Chaotic Maps," *Baghdad Journal of Science*, vol. 63, no. 2, pp. 830–842, 2022, doi: 10.24996/ijs.2022.63.2.36.
[32]A. Al-Daraiseh, Y. Sanjalawe, S. Al-E'mari, S. Fraihat, M. Bany Taha, and M. Al-Muhammed, "Cryptographic Grade Chaotic Random Number Generator Based on Tent-Map," *Journal of Sensor and Actuator Networks*, vol. 12, no. 5, Oct. 2023, doi: 10.3390/jsan12050073.
[33]N. Kuznetsov, T. Mokaev, V. Ponomarenko, E. Seleznev, N. Stankevich, and L. Chua, "Hidden attractors in Chua circuit: mathematical theory meets physical experiments," *Nonlinear Dyn*, vol. 111, no. 6, pp. 5859–5887, Mar. 2023, doi: 10.1007/s11071-022-08078-y.
[34]R. Rocha and R. O. Medrano-T, "Chua Circuit based on the Exponential Characteristics of Semiconductor Devices," Dec. 2021, doi: 10.1016/j.chaos.2021.111761.
[35]B. Arpacı, E. Kurt, and K. Çelik, "A new algorithm for the colored image encryption via the modified Chua's circuit," *Engineering Science and Technology, an International Journal*, vol. 23, no. 3, pp. 595–604, Jun. 2020, doi: 10.1016/j.jestch.2019.09.001.
[36]Z. Galias, "Continuation-based method to find periodic windows in bifurcation diagrams with applications to the Chua's circuit with a cubic nonlinearity," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 9, pp. 3784–3793, Sep. 2021, doi: 10.1109/TCSI.2021.3089420.
[37]B. Emin and Z. Musayev, "Chaos-based Image Encryption in Embedded Systems using Lorenz-Rossler System," *Chaos Theory and Applications*, vol. 5, no. 3, pp. 153–159, 2023, doi: 10.51537/chaos.1246581.
[38]B. Kharabian and H. Mirinejad, "Synchronization of Rossler chaotic systems via hybrid adaptive backstepping/sliding mode control," *Results in Control and Optimization*, vol. 4, no. May, p. 100020, 2021, doi: 10.1016/j.rico.2021.100020.
[39]J. P. Parker, D. Goluskin, and G. M. Vasil, "A study of the double pendulum using polynomial optimization," Jun. 2021, doi: 10.1063/5.0061316.
[40]S. Cabrera, E. D. Leonel, and A. C. Marti, "Regular and chaotic phase space fraction in the double pendulum," Dec. 2023, [Online]. Available: http://arxiv.org/abs/2312.13436
[41]S. R. de Oliveira, "Deterministic chaos: A pedagogical review of the double pendulum case," *Revista Brasileira de Ensino de Fisica*, vol. 46, 2024, doi: 10.1590/1806-9126-RBEF-2024-0060.

[42]J. J. López and V. J. García-Garrido, “Chaos and Regularity in the Double Pendulum with Lagrangian Descriptors,” Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.07000
[43]R. S. Abdulaali and R. K. Jamal, “A Comprehensive Study and Analysis of the Chaotic Chua Circuit,” *Iraqi Journal of Science*, vol. 63, no. 2, pp. 556–570, 2022, doi: 10.24996/ijs.2022.63.2.13.
[44]N. Sanghi, “Deep Q-Learning,” in *Deep Reinforcement Learning with Python: With PyTorch, TensorFlow and OpenAI Gym*, Berkeley, CA: Apress, 2021, pp. 155–206. doi: 10.1007/978-1-4842-6809-4_6.
[45]L. Graesser and W. Loon Keng, “Foundations of Deep Reinforcement Learning _ Theory and Practice in Python,” Nov. 2021.
[46]A. Plaat, *Deep Reinforcement Learning*. Springer Nature, 2022. doi: 10.1007/978-981-19-0638-1.
[47]N. Ketkar and J. Moolayil, *Deep Learning with Python*. 2021. doi: 10.1007/978-1-4842-5364-9.
[48]T. Xu, Y. Liu, Z. Ma, Y. Huang, and P. Liu, “A DQN-Based Multi-Objective Participant Selection for Efficient Federated Learning,” *Future Internet*, vol. 15, no. 6, Jun. 2023, doi: 10.3390/fi15060209.
[49]F. Li, J. Yang, K. Y. Lam, B. Shen, and G. Wei, “Dynamic spectrum access for Internet-of-Things with joint GNN and DQN,” *Ad Hoc Networks*, vol. 163, Oct. 2024, doi: 10.1016/j.adhoc.2024.103596.
[50]L. E. Bassham *et al.*, “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” Gaithersburg, MD, 2022. doi: 10.6028/NIST.SP.800-22r1a.
[51]Y. Zhang, L. Zhang, Z. Zhong, L. Yu, M. Shan, and Y. Zhao, “Hyperchaotic image encryption using phase-truncated fractional Fourier transform and DNA-level operation,” *Opt Lasers Eng*, vol. 143, p. 106626, 2021, doi: https://doi.org/10.1016/j.optlaseng.2021.106626.
[52]Y. A. Liu *et al.*, “A dynamic AES cryptosystem based on memristive neural network,” *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-13286-y.
[53]E. Barker, “Recommendation for key management:,” Gaithersburg, MD, May 2020. doi: 10.6028/NIST.SP.800-57pt1r5.
[54]*Entropy Method for Assessing the Strength of Encryption Algorithms*. IEEE, 2024.
[55]I. Buhan, L. Batina, Y. Yarom, and P. Schaumont, “SoK: Design Tools for Side-Channel-Aware Implementations,” Jun. 2021, [Online]. Available: http://arxiv.org/abs/2104.08593
[56]X. Lou, T. Zhang, J. Jiang, and Y. Zhang, “A Survey of Microarchitectural Side-channel Vulnerabilities, Attacks and Defenses in Cryptography,” Mar. 2021, [Online]. Available: http://arxiv.org/abs/2103.14244
[57]D. Ojha and S. Dwarkadas, “Timing Cache Accesses to Eliminate Side Channels in Shared Software,” Dec. 2021, doi: 10.1109/ISCA52012.2021.00037.
[58]F. Mahmud, S. Kim, H. S. Chawla, C.-C. Tsai, E. J. Kim, and A. Muzahid, “Attack of the Knights: A Non Uniform Cache Side-Channel Attack,” May 2023, doi: 10.1145/3627106.3627199.
[59]R. L. Schröder, S. Gast, and Q. Guo, *Divide and Surrender: Exploiting Variable Division Instruction Timing in HQC Key Recovery Attacks*. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/schr
[60]A. Tsuneda, “Auto-Correlation Functions of Chaotic Binary Sequences Obtained by Alternating Two Binary Functions,” *Dynamics*, vol. 4, no. 2, pp. 272–286, Jun. 2024, doi: 10.3390/dynamics4020016.
[61]F. Ye, S. Zhang, P. Wang, and C.-Y. Chan, “A Survey of Deep Reinforcement Learning Algorithms for Motion Planning and Control of Autonomous Vehicles,” May 2021, [Online]. Available: http://arxiv.org/abs/2105.14218