Mesopotamian journal of Cybersecurity
Vol.5,No.1, **pp**. 301–318
DOI: https://doi.org/10.58496/MJCS/2025/019; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Robust and Efficient Methods for Key Generation using Chaotic Maps and A2C Algorithm

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

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

ARTICLE INFO

Article History

Received 01 Mar 2025
Accepted 31 Mar 2025
Published 15 Apr 2025

Keywords

Cryptography

Chaotic

NIST

RNG

Randomness

ABSTRACT

In the current digital landscape, information security has become a critical necessity given the escalating frequency and sophistication of cyberattacks across global computing networks. Cryptography is the science and practice of securing information by transforming it into a format that is unreadable or inaccessible to unauthorized parties. The strength of the cryptography algorithm lies in the strength of used encryption key. Recently, the nonlinear behavior of chaotic maps has been utilized as a random source to generate robust key stream bits for cryptographic purposes. The aim of this paper is to introduce an efficient and robust system for generating strong key stream bits against different types of attacks. The proposed system implemented six proposed scenarios for generating a highly strong key stream bits based on different 5-types of chaotic maps (Tent, Ikeda, Chua's, Rössler and Double Pendulum). In the first five scenarios, use each map with Deep-Reinforcement Learning (DRL) algorithm called Advantage Actor-Critic (A2C), while in the sixth scenario, the fusion of all above five chaotic maps with Advantage Actor-Critic (A2C) are used at the same time. For each scenario of the proposed system generates three different lengths of key stream bits (128-bit, 192-bit, and 256-bit). To evaluate the robustness, randomness, and effectiveness of the proposed system, several types of tests were applied like NIST, brute-force attack, Auto Correlation (AC), Cross Correlation (CC), and Discrete Fourier Transform (DFT). All the obtained results indicated to the highly robustness and highly strength of generated key stream bits for all six proposed scenarios of the proposed system.

1. INTRODUCTION

The exponential expansion of internet technologies has created unprecedented demand for robust information protection across diverse domains. Consequently, cybersecurity challenges have emerged as critical research priorities in recent literatures [1], [2]. Cryptographic mechanisms such as hash functions, encryption, and digital signatures often incorporate Pseudo-Random Number Generator (PRNG) in their design for critical tasks like key generation, One-Time Passwords (OTP)[3], [4], [5]. Most cryptographic algorithms depend particularly on PRNG sequences for cipher keys, many of which are in use in cryptography and digital communications today, that making the keys difficult to predict[6], [7]. Encryption is important because it allows securely preserve data that you do not want anyone else to have access to[8], [9]. Symmetric (secret-key) and asymmetric (public-key) encryption are two classes that are used for classifying cryptography algorithms[10]. Moreover, generating cipher keys randomly is considered an essential principle in cryptography and information security science to protect confidentiality, integrity, and availability. To convert plain-text to cipher-text, the encryption operation requires an algorithm with strong key [11], [12]. It has been observed recently, chaotic maps have been used in various cryptographic systems. Chaotic maps find applications in various domains of information security, including stream ciphers [13], [14], block ciphers [14], [15], hashing [15], [16], steganography, and digital watermarking [17], [18], [19], [20]. The remaining sections of this paper are structured as follows. Section 2 presents a comprehensive review of the literatures that use chaotic maps. Section 3 provides an introduction to the chaotic maps used in this paper. Section 4 describes the concepts of Artificial Intelligence (AI) in addition to Deep Reinforcement Learning (DRL) and Advantage Actor-Critic (A2C) algorithm. Section 5, illustrates the evaluation metrics used in this paper. The proposed system for stream key bits generation is discussed in Section 6. In Section 7, the experimental results are presented. Finally, section8, presented the conclusions of this paper.

*Corresponding author. Email: ali.abdulameer1201a@sc.uobaghdad.edu.iq

2. RELATED WORK

The use of chaotic maps in cryptography has significantly increased, and it is extreme sensitivity to initial conditions, and complex behavior. The mathematical literature offers an extensive collection of chaotic dynamical systems suitable for high-entropy stochastic sequence generation applications. This section reviews several literatures explores chaos for key generation.

In [18], the researcher introduced a modified logistic map with a larger control parameter space. The Control of Chaos (CoC) technique employed the robust chaotic logistic map to generate Cryptographically Secure Pseudorandom Number Generator (CSPRNG). The proposed method was also implemented on the various chaotic maps, such as Circle map, Iterative map, Tent map, and Singer maps. The NIST has been successfully implemented on these maps entails CSPRNG.

The research [6] , introduces an innovative approach for stochastic sequence generation that leveraging interconnected map lattice architectures. This methodology incorporates adaptively parameterized generalized symmetric mapping functions within a network framework where individual mapping nodes establish connections with adjacent elements, substantially enhancing output complexity. Comprehensive validation through multiple assessment methodologies confirmed the system's robustness and demonstrated its expansive cryptographic key space characteristics.

In [21], researchers developed a hardware-based True Random Number Generator (TRNG) for real-time video encryption by combining four chaotic models: Lorenz, fractional-order Chen-Lee, and discrete-time Logistic and Tent maps. Specialized analog circuits generated random bits at 25 Mbps, whereas an FPGA pipeline performed real-time XOR-based encryption. Validations with NIST SP 800-22, FIPS 140-1, and chi-square tests confirmed high throughput and robust security metrics, highlighting the scheme's effectiveness for real-world video encryption applications.

In [20], a novel implementation based on a Robust Chaotic Tent Map (RCTM) was introduced, featuring a modified mathematical framework that uses modulo and scaling operations to expand the key space to 2^{199} well above the NIST-recommended minimum of 2^{128}. Rigorous testing via NIST SP 800-22, ENT, and TestU01 suites, along with differential analysis, demonstrate the strong cryptographic qualities of the system.

3. CHAOTIC MAPS

As a specialized mathematical field, chaos theory has attracted significant research interest because it exhibits seemingly disordered and random behavior while maintaining extreme sensitivity to starting conditions [6], [22]. A variety of chaotic maps are available for PRNG, this section focuses on five maps (Tent, Ikeda, Chua's, Rössler and Double Pendulum) are used in the proposed system in this paper as follows: -

3.1 Tent Map

The Tent map represents a one-dimensional chaotic function demonstrating unpredictable behavior that finds extensive application in both dynamical systems analysis and cryptographic implementations [23]. The Tent map, a simple structure, is useful for cryptographic applications to generate PRNG but limited key parameter space [20]. The Tent-Map serves as a fundamental example of chaotic system, characterized by its one-dimensional, noninvertible, piecewise linear discrete properties [24]. Its practical applications extend to pseudorandom number generation, data encryption mechanisms, and robust protocols for secure communications[55]. 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 [23].

3.2 Ikeda Map

Is a two-dimensional chaotic map, the model of this map is known for its fractal structures and sensitivity to initial conditions. The Ikeda map is defined via variables u_n and t_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 [25].

3.3 Chua's Circuit Map

Is a three-dimensional map, requires three variables x, y and z to describe its state fully, these variables represent voltages and currents in the circuit components. The system's evolution depends on the interactions among these three variables [26], [27]. 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 [28], [29]. Initializes the Chua's state using three-element array with random values between -1 and 1 [28].

3.4 Rössler Attractor Map

Is 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 [30], [31]. The Rössler system is known for its chaotic attractor, which, like the Lorenz attractor, exhibits 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 [30].

3.5 Double Pendulum theory

The double pendulum is a classic example of a simple physical system exhibiting chaotic dynamics and small differences in initial conditions can lead to vastly different trajectories [32]. The double pendulum consists of two pendulums attached end to end, and the exact equations are complex and involve trigonometric functions and parameters like masses and lengths [33]. The state of the double pendulum is fully described by four variables θ_1, ω_1, θ_2, and ω_2 [34], [35] .

For more illustration the details of the above chaotic maps, see Table 1 [35], [36], [37].

TABLE I. DETAILS OF TENT, IKEDA, CHUA'S, 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	$\begin{cases} \dfrac{d^2\theta_1}{dt^2} = \dfrac{-g(2m_1 + m_2)\sin\theta_1 - m_2 g \sin(\theta_1 - 2\theta_2) - 2\sin(\theta_1 - \theta_2) m_2 \left[\dot{\theta}_2^2 l_2 + \dot{\theta}_1^2 l_1 \cos(\theta_1 - \theta_2)\right]}{l_1\left(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2)\right)}, \\ \dfrac{d^2\theta_2}{dt^2} = \dfrac{2\sin(\theta_1 - \theta_2)\left[\dot{\theta}_1^2 l_1 (m_1 + m_2) + g(m_1 + m_2)\cos\theta_1 + \dot{\theta}_2^2 l_2 m_2 \cos(\theta_1 - \theta_2)\right]}{l_2\left(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2)\right)} \end{cases}$
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

4. EXPLAINABLE ARTIFICAL INTELLIGENCE (AIX)

Artificial Intelligence (AI) encompasses various subfields, among which Machine Learning (ML), Deep Learning (DL), and Deep Reinforcement Learning (DRL) are prominent. Understanding their distinctions and interrelationships is crucial for leveraging their capabilities effectively[56]. This section provides an explanation of DRL, compares it with DL and ML, and illustrates why DRL is advantageous in certain contexts. Table 2 illustrates the comparison among ML, DL, and DRL [38].

TABLE II. COMPARISON AMONG ML, DL, AND DRL

Definition	
ML	Algorithms that parse data, learn from it, and make decisions or predictions.
DL	Subset of ML using neural networks with multiple layers to model complex patterns.
DRL	Combines DL with reinforcement learning to make sequential decisions.
Learning	
ML	Broad field including supervised, unsupervised, and reinforcement learning.
DL	Focuses on neural networks with many layers to model complex patterns.
DRL	Applies deep learning techniques within the reinforcement learning framework.
Data Dependency	
ML	Relies on structured datasets; performance depends on data quality.
DL	Requires large volumes of data due to numerous parameters.
DRL	Learns from interactions; can generate data through exploration.
Complexity and Computation	
ML	Computationally less intensive; suitable for simpler patterns.
DL	Computationally intensive; requires significant processing power.
DRL	Highly computational; combines the demands of DL and RL.
Applications	
ML	Spam detection, recommendation systems, basic image recognition.
DL	Advanced image and speech recognition, language translation.
DRL	Robotics, game playing, autonomous driving.
Interpretability	
ML	Generally, more interpretable due to simpler models.
DL	Often considered a "black box" due to complex architectures.
DRL	Interpretation is challenging due to the combination of deep networks and learning policies.

In the realm of DRL, the Advantage Actor-Critic (A2C) algorithm is a prominent method that combines the strengths of both policy-based and value-based approaches [39]. A2C is an advancement over traditional actor-critic methods, which designed to improve learning stability and efficiency [40]. In A2C, value-based methods learn value functions to estimate the expected return and policy-based methods learn policy functions that map states to actions directly. Finally, the actor-critic methods combine both approaches, where the actor decides actions, and the critic evaluates them [41]. Within actor, the policy function selects actions based on the current state, and within critic, the value function that evaluates the action taken by the actor by estimating the expected return [42]. The advantages of using actor-critic methods are stability (the critic guides the actor, reducing variance in policy updates) and efficiency (combining the benefits of policy gradients and value function approximation) [43]. The Advantage Actor-Critic (A2C) algorithm is a significant method in DRL, offering a balance between learning efficiency and implementation simplicity [44].
By synchronizing updates across multiple workers and utilizing advantage estimates for policy updates, A2C achieves stable and scalable learning in various environments [45].
Table 3 illustrates the definition of A2C parameters [46].

TABLE III. A2C HYPER PARAMETERS DEFINITIONS

	Hyper parameter	**Definition**
1	**Learning Rate**	Controls the step size during optimization; needs to balance convergence speed and stability.
2	**Number of Steps**	Determines the trajectory length for updates; affects bias-variance trade-off in advantage estimates.
3	**Discount Factor (gamma)**	Balances the importance of immediate vs. future rewards; higher values encourage long-term strategies.
4	**GAE Lambda**	Adjusts bias and variance in advantage estimation; higher values use more future information.
5	**Entropy Coefficient**	Encourages exploration by adding randomness; important for preventing premature convergence.
6	**Value Function Coefficient**	Balances the training focus between the policy and value networks.
7	**Maximum Gradient Norm**	Stabilizes training by preventing large gradient updates; helps avoid exploding gradients.
8	**Using RMSProp**	Chooses the optimizer; RMSProp is adaptive and suitable for noisy updates.
9	**Normalize Advantage**	Affects the scaling of advantage estimates; normalization can improve stability.

A2C agent consists of the following layers:

1- *Input Layer:*
The purpose of input layer is to receive the state representation from the environment. The number of input layers is typically one input layer, while the number of neurons in the input layer equals the size of the state vector [47].

2- *Hidden Layers:*
The purpose of hidden layers is to process the input data to extract meaningful features that help the agent make decisions. The exact number of hidden layers can vary based on the complexity of the task, commonly; two or three hidden layers [48].
Typically, the number of neurons is chosen based on empirical performance; for example, the first hidden layer consists of 128 neurons and second hidden layer consists of 128 neurons. The activation functions like ReLU (Rectified Linear Unit) are commonly used after each hidden layer [49].

3- *Output Layers*
The purpose of output layers is to provide the final outputs of the network, the policy and the value estimation. The number of output layers are two output layers one for the actor and one for the critic [50].

5. EVALUATION METRICS

There are several tests use to evaluate the robustness and randomness of generated stream key bits; this section shows the metrics used in this paper.

5.1 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 behavior, 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 [51].

5.2 Brute-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 $>2^{128}$ possibilities) that render brute-force attacks infeasible [52]. Moreover, chaotic keys exhibit high key sensitivity, meaning 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 show that chaos-driven neural key generators achieve vast key spaces and successfully resist exhaustive (brute-force) key searches [53].

5.3 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 non-zero shift) is desired. If a chaotic key sequence had significant autocorrelation at some lag, it would indicate repeating patterns or predictability a vulnerability for attackers[54].

5.4 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 [43].

5.5 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) [51]. In the context of chaotic map keys, passing this DFT test indicates the sequence has no discernible periodicity [52].

6. PROPOSED SYSTEM (DRLKG-Chaotic)

Recent cryptographic researches witnessed a growing emphasis on chaos-based sequence generation algorithms for security applications. Despite this trend, a significant challenge remains: numerous 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 the 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 DRL algorithm A2C with different scenarios, the proposed system called (DRLKG-Chaotic), shortest for Deep Reinforcement Learning Key Generation with Chaotic maps as shown in Figure 1.

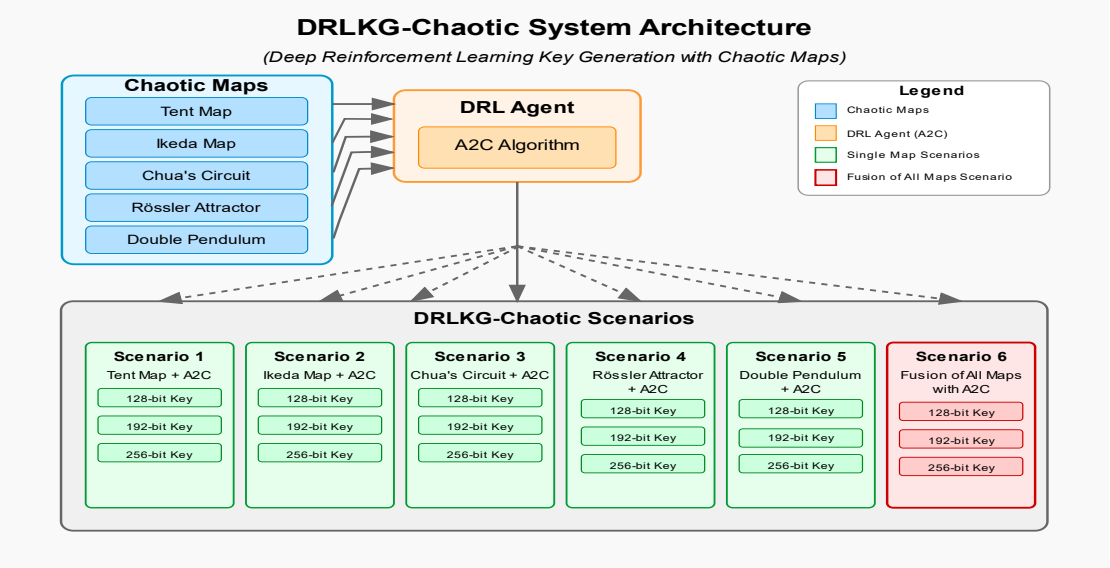


Fig. 1. Deep Reinforcement Learning with Chaotic Maps Scenarios

The methodology for implementing the proposed system DRLKG-Chaotic basically used five chaotic maps (Tent, Ikeda, Chua's, Rössler and Double Pendulum). This proposed system represented by six scenarios that is in the first five scenarios used each one of five chaotic maps with A2C agent separately, finally the remaining scenario implemented by using the fusion of A2C with five maps at the same time. The last scenario used to increase the complexity of the generated stream key bits. For each one 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 A2C is used. It is crucial to select parameters that promote near-optimal randomness; therefore the following subsections illustrate the parameters used with each scenario.

6.1 Scenario-1: Tent Map-A2C

For this implementation, the algorithm processes input value denoted as x. within the range [0,1] & 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 to 1.9999 during the training phase. Additionally, a one-dimensional Tent-map, have been implemented with A2C agent to generate several strong stream key bits with different three lengths (128, 192, and 256 bits) [24].

6.2 Scenario-2: Ikeda Map-A2C

In this scenario used Ikeda map with A2C agent to generate several strong stream key bits with different three lengths (128, 192, and 256-bits). At initialization of Ikeda map, the state is a two-element array with random values between -1 and 1. The Ikeda map is implemented with parameters as illustrated in Table 4.

TABLE IV. IKEDA MAP-A2C PARAMETERS

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

6.3 Scenario-3: Chua's Circuit -A2C

In this scenario, Chua's map used with A2C agent to produce number of strong randomness stream key bits with different three lengths (128, 192, and 256-bits). The initialization Chua's state by using 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 illustrated in Table 5.

TABLE V. CHUA'S CIRCUIT -A2C PARAMETERS

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

6.4 Scenario-4: Rossler Attractor-A2C

This scenario is the fourth scenario proposed in DRLKG-Chaotic system for randomness stream key bits generation by using Rossler Attractor with A2C 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 numerical integration. The values of Roessler Attractor parameters in this scenario illustrated in Table 6.

TABLE VI. ROSSLER ATTRACTOR-A2C PARAMETERS

Parameter	Value
a	0.2
b	0.2
c	5.7

6.5 Scenario-5 Double Pendulum - A2C

In this scenario a new efficient type of chaotic map used with A2C agent to generate several robust stream bit key. 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 numerical integration. The parameters values set as illustrated in Table 7.

TABLE VII. DOUBLE PENDULUM - A2C

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 scenarios from 6.1 to 6.5.

Algorithm-1: Proposed A2C With Single Chaotic Maps Algorithm
Input: • Key Length (KL): Number of bits to be generated (128, 192, 256). • Chaotic Map Select (CMS): Choose specific type (Tent, Ikeda, Chua's, Rössler Attractor, double Pendulum). • Parameters controlling for each chaotic maps (e.g., coefficients for Rössler Attractor, Chua's Circuit, etc.). • Hyperparameters for A2C agent according to table-8.
Output: Generated Stream Key bits (GSK).
Step 1: Environment Initialization 1.1 Initialize chaotic states $S \in R^n$, where n represent the dimension for specific chaotic map as follows: • Tent $\rightarrow n = 1$ • Ikeda $\rightarrow n = 2$ • Chua/Rössler $\rightarrow n = 3$ • Double pendulum $\rightarrow n = 4$ 1.2 Create an environment with a dimensional state vector $S \in R^n$, then full this state by seed generation using True Random Number Generation (TRNG).
Step 2: Chaotic maps implementation 2.1 Apply the CMS with default parameter. 2.2 Update parameter from read the scalar action and clips.
Step 3: Training model 3.1 Instantiate an A2C agent with policy of MLP, then call the CMS 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 taring agent.
Step 5: End

6.6 Scenario-6 A2C- 5maps Fusion

In this scenario, increase the randomness and complexity of generated stream key bits for three different lengths (128, 192, and 256-bits) by using the fusion of A2C with five maps. The training operation of A2C method implemented by updating which made to maximize the expected return using policy gradients then loss function includes terms for the advantage and, optionally, entropy regularization. Updates are made to minimize the mean squared error between the predicted value and the actual return and the loss function is typically the value loss, which is the squared difference between the estimated value and the target value.

Figure 2 Illustrates the implementation of connection among different layers of actor neural network. This connection typically structured as a Multi-Layer Perceptron (MLP), each neuron in one layer is connected to every neuron in the next layer. In addition to shared hidden layers between actor and critic allow both networks to learn from common features. After the shared layers, the network splits into two separate output layers for the actor and critic.

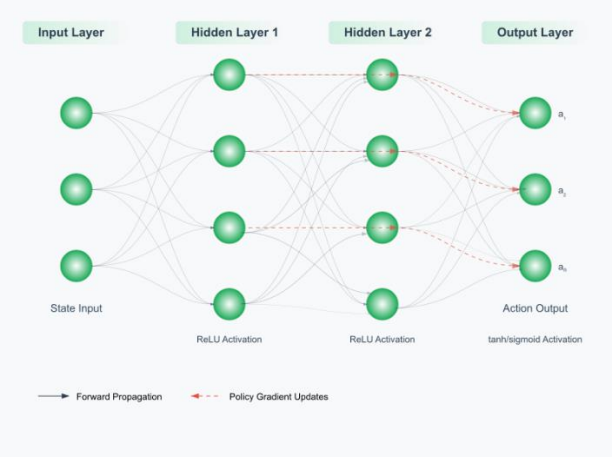


Fig. 2. Actor Neural network (Fully connected layers)

The implementation of connection among different layers of critical neural network illustrated in Figure 3. This figure effectively illustrates the processes state and action inputs separately, then merges them in the hidden layers, produces a single Q-value output and use backpropagation to learn the value function.

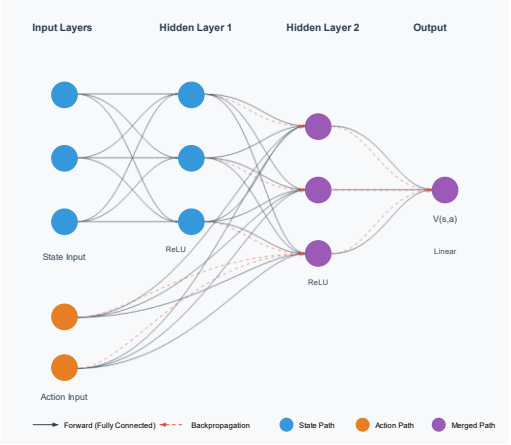


Fig. 3. Critic Neural network (Fully connected layers)

Figure 4 Illustrates the interaction between Actor and Critic networks, the network architectures partitioned to the following parts: -

- Actor: 3-layer neural network (input → hidden → output).
- Critic: 3-layer neural network with dual inputs (state and action).
- Target Netwo*rks: Delayed copies of both Actor and Critic.

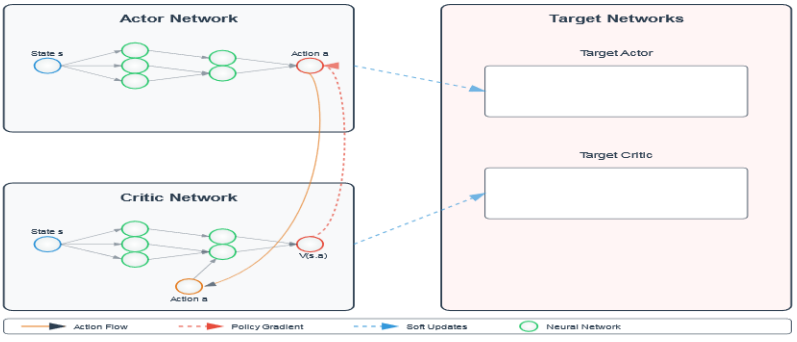


Fig. 4. Interaction between Actor and Critic networks in a deep reinforcement learning

The interactions show that; the Actor produces actions based on state input; then; these actions are fed to Critic along with state and Critic evaluates action quality. The learning mechanisms occur when the policy gradient flows from Critic to Actor, soft updates maintain the target networks, and dual input processing in Critic network.
Additionally, the A2C parameters used with all scenarios of DRLKG-Chaotic system as illustrated in Table 8.

TABLE VIII. A2C HYPER PARAMETERS VALUES

Hyper parameter		**Values**
1	Learning Rate	0.0003
2	Number of Steps	5
3	Discount Factor (gamma)	0.99
4	GAE Lambda	1.0
5	Entropy Coefficient	0.0
6	Value Function Coefficient	0.5
7	Maximum Gradient Norm	0.5
8	Using RMSProp	True
9	Normalize Advantage	False

The combination of A2C 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 2 represents the proposed algorithm for scenario 6.6.

Algorithm-2: Proposed A2C Fusion Algorithm

Input:

- Key Length (KL): Number of bits to be generated (128, 192, 256).
- Parameters controlling for each chaotic maps (e.g., coefficients for Rössler Attractor, Chua's Circuit, etc.).
- Hyperparameters for A2C agent according to table X.

Output: Generated Stream Key bits (GSK).

Step 1: Environment Initialization

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

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 full this state by generated values from step 2.1.

Step 2: Chaotic maps implementation

2.1 Apply Tent map, the output of the Tent map is used 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 an A2C 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 taring agent.

Step 5: End

7. RESULTS DISCUSSION

This section presents a comprehensive analysis of experimental findings derived from multiple evaluation methodologies implemented to measure the statistical performance metrics of the DRLKG-Chaotic framework. Three experiments implemented in the proposed system DRLKG-Chaotic, these experiments illustrated as below: -

7.1 Experiment-1

In this experiment the standard five maps are implemented separately (Tent, Ikeda, Chua's, Rössler and Double Pendulum) to generate the stream key bits with different three lengths (128, 192, and 256-bits), the results of NIST tests of this experiment indicated that the produced stream key bits are generated with a low level 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, relatively low performance, such that Tent map: $0.3627 \leq H \leq 0.3762$, Ikeda map: $0.4480 \leq H \leq 0.4850$, Chua's Circuit: $0.4766 \leq H \leq 0.5323$, Rössler Attractor: $5076 \leq H \leq 0.5144$ and Double Pendulum $0.5216 \leq H \leq 0.5168$. As illustrated in this table the highest entropy value is 0.5463 with Rössler Attractor, by the same way of analysis, all the results obtained of the NIST tests of this table are not optimal and 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.

7.2 Experiment-2

The second experiment represented by applying the first five scenarios of proposed DRLKG-Chaotic by using five types of chaotic maps with A2C agent for stream key bits generation.

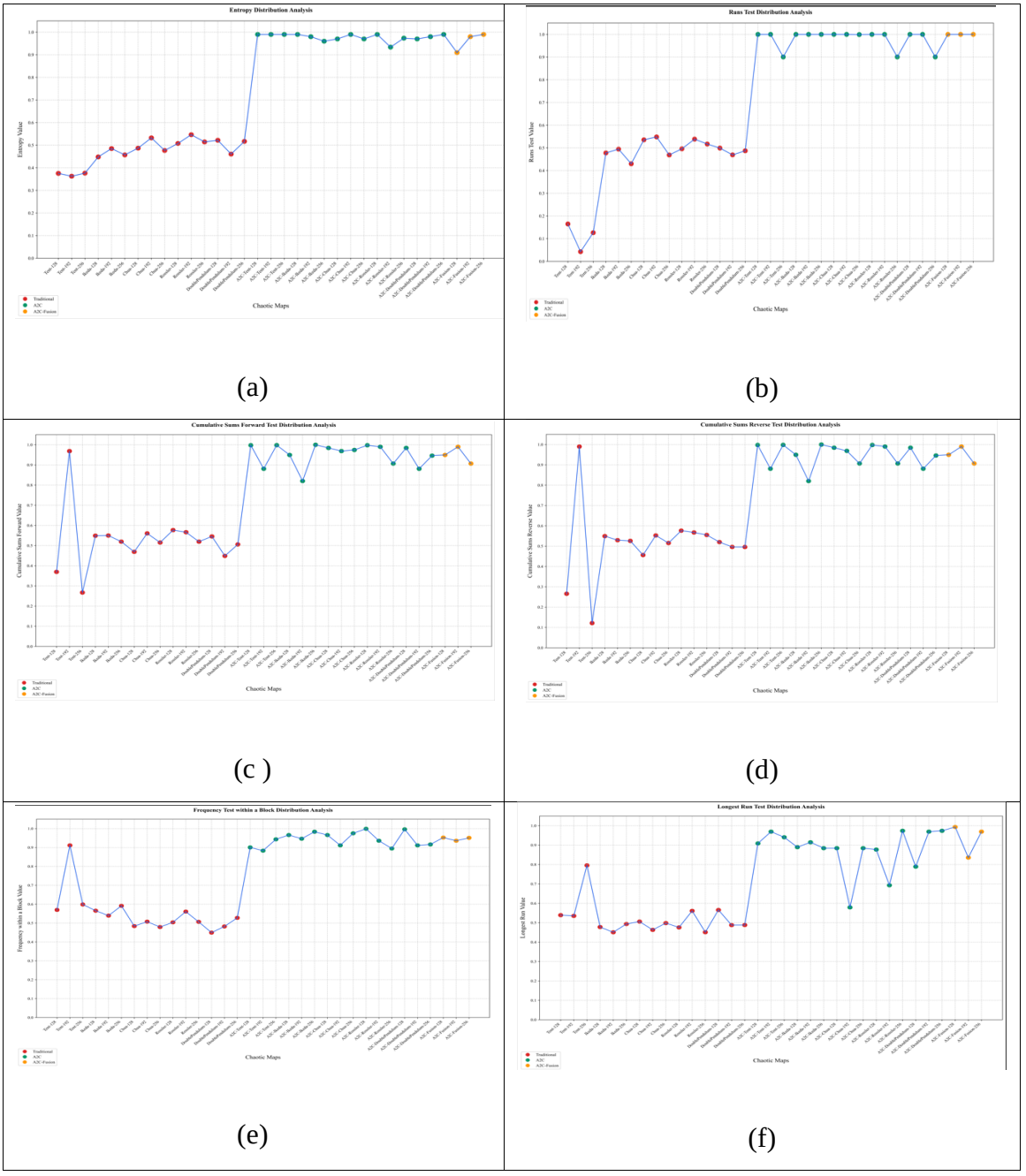
7.3 Experiment-3

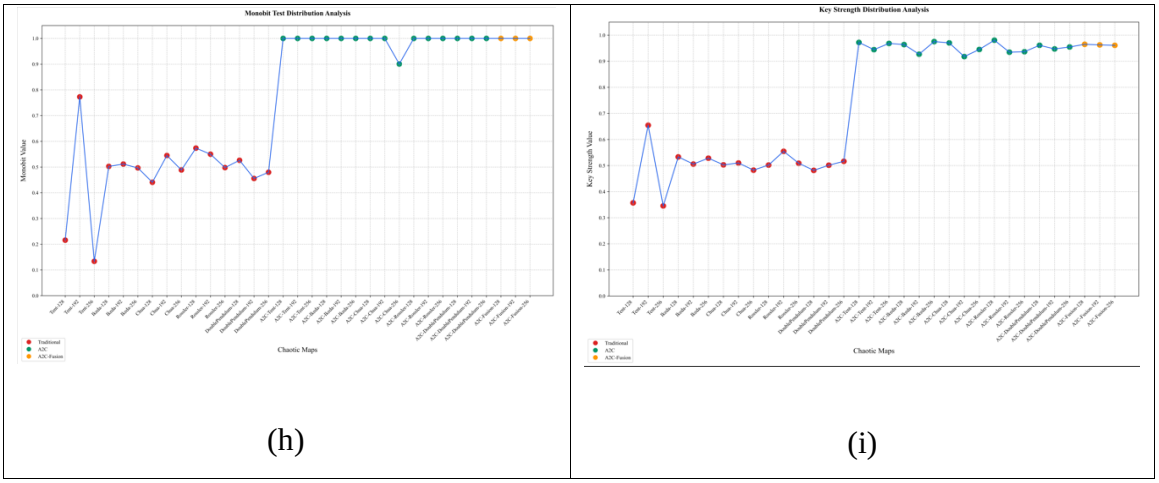
This experiment represents scenario-6 of the proposed DRLKG-Chaotic, where used A2C fusion for all five chaotic maps. The results of NIST tests for the generated stream key bits of the last two experiments, illustrated in Table 10.

TABLE X. NIST P-VALUES FOR A2C CASES (SCENARIO-1 TO SCENARIO -6)- EXPERIMENT-2

scenario No.	Key Length	Entropy	Runs	cumulative sums Forward	cumulative sums Reverse	Block Frequency	Longest Run	Monobit	Key Strength
1	128	0.99999	1.00000	0.99770	0.99770	0.90042	0.90876	1.00000	0.97208
	192	0.99671	1.00000	0.88082	0.88082	0.88317	0.96927	1.00000	0.94440
	256	0.99813	0.90052	0.99798	0.99798	0.94327	0.94040	1.00000	0.96833
2	128	0.99280	1.00000	0.94927	0.94927	0.96586	0.88940	1.00000	0.96380
	192	0.98756	1.00000	0.82022	0.82022	0.94631	0.91451	1.00000	0.92697
	256	0.96021	1.00000	0.99980	0.99980	0.98344	0.88428	1.00000	0.97536
3	128	0.97347	1.00000	0.98416	0.98416	0.96586	0.88428	1.00000	0.97027
	192	0.99671	1.00000	0.96856	0.96856	0.91141	0.57895	1.00000	0.91774
	256	0.97663	0.99922	0.97420	0.90639	0.97509	0.88428	0.90052	0.94519
4	128	0.99280	1.00000	0.99770	0.99770	0.99885	0.87688	1.00000	0.98056
	192	0.93381	1.00000	0.98987	0.98987	0.93572	0.69290	1.00000	0.93460
	256	0.97347	0.90052	0.90639	0.90639	0.89459	0.97369	1.00000	0.93643
5	128	0.97347	1.00000	0.98416	0.98416	0.99640	0.78903	1.00000	0.96103
	192	0.98756	1.00000	0.88082	0.88082	0.91141	0.96927	1.00000	0.94713
	256	0.99999	0.90052	0.94589	0.94589	0.91608	0.97369	1.00000	0.95458
6	128	0.90940	1.00000	0.94927	0.94927	0.95258	0.99344	1.00000	0.96485
	192	0.98756	1.00000	0.98987	0.98987	0.93572	0.83549	1.00000	0.96264
	256	0.99281	1.00000	0.90639	0.90639	0.95096	0.96927	1.00000	0.96083

This table indicates to a high level of robustness and randomness for the generated key stream bits. In addition, the A2C implementation significant improvements in randomness metrics, with P-values consistently above 0.90 across all test parameters. Notable improvements are observed in the entropy values, which range from 0.90940 to 0.99999, indicating that enhanced randomness 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 A2C 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 of Tables (7-8), see Figure 5. From these figures it's very clear the enhancement of strength and randomness of the generated stream key bits of the proposed DRLKG-Chaotic system compared with standard maps.





(h) (i)

Fig. 5. NIST tests for three experiments ((a) Entropy, (b) Run-tests, (c) Cumulative sums forward, (d) Cumulative sums backward, (e) Frequency, (f)Longest run, (h) Mono bit, (i) Key strength

Extended sequence generation substantially 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 used three different length of generated stream key bits (128, 192, and 256-bits). Table 11 Illustrates 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

For proving the high level of randomness, robustness, and complexity of the generated stream key bits, several addition tests applied of the selected stream key bits, for shorting take the results of scenario-6. Table 12 Illustrates the Auto-Correlation of stream key bits generated from scenario -6.

TABLE XII. AUTO-CORRELATION OF STREAM KEY BITS OF SCENARIO-6

Key Length/bits	**Key no.**	**Average Autocorrection**	**Maximum Autocorrection**	**Minimum Autocorrection**	**Significant Autocorrections%**
128 bits	1	-0.011206	0.419355	-0.310345	3
	2	-0.008196	0.260870	-0.350000	1
192 bits	1	-0.003109	0.244094	-0.226277	0
	2	-0.010475	0.188406	-0.166667	0
256 bits	1	-0.008080	0.177033	-0.190476	0
	2	-0.011206	0.419355	-0.310345	3

For more illustration Figure 6 demonstrates the Auto-Correlation such that the third column represents the Auto-Correlation between two selected stream key bits for each scenario with three different key lengths (128, 192, and 256).

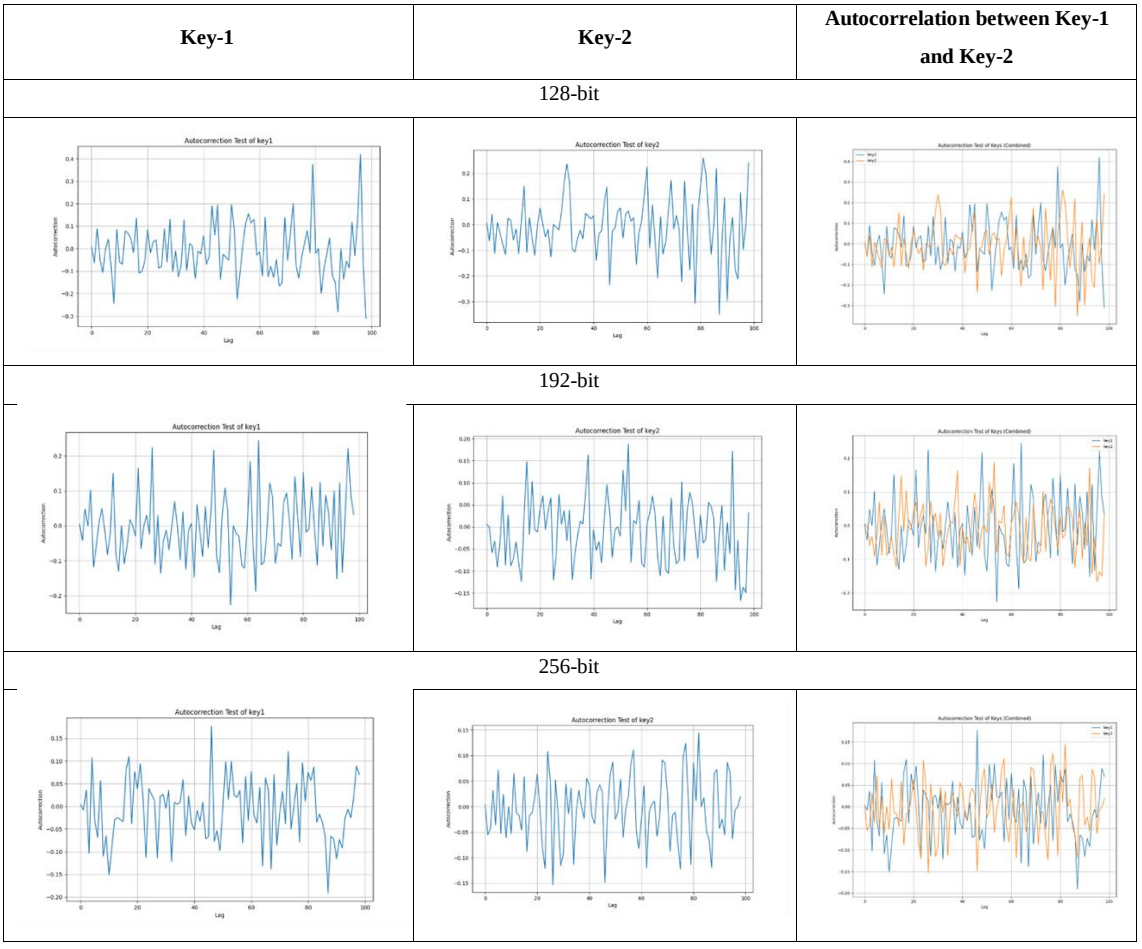


Fig. 6. Auto-Correlation between two Keys with three different Lengths of scenario-6

Another indication for the strength of the generated stream key bits of the proposed DRLKG-Chaotic by using the Cross-Correlation. From the results of the Table 13, it is clear that no strong evidence of correlation for the generated keys.

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

scenario No.	Length of keys		
	128-bit	192-bit	256-bit
1.	0.093750	0.020833	-0.046875
2.	-0.062500	0.020833	-0.039064
3.	-0.062500	-0.062500	0.046817
4.	0.000000	0.125000	0.023438
5.	-0.062500	-0.062500	-0.039064
6.	0.000000	-0.041667	0.023438

Figure 7 demonstrates a map of a cross-correlation matrix for two variables, key1 and key2, at a lag of 0. Here's the analysis, such that the diagonal elements (diagonal entries) represent the (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. While the off-diagonal elements (off-diagonal entries) represents the (key1 vs. key2 and key2 vs. key1) have a value of 0, this indicates a weak positive correlation between key1 and key2. This map uses a color gradient where red represents positive correlation, blue represents negative correlation, and lighter shades represent weaker correlations. The diagonal is deep red (indicating 1.0), while the off-diagonal elements are in light beige weak correlation -0.042. 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 low value (0.023) suggests that key1 and key2 have almost no linear relationship, as the correlation is close to zero, this means weak positive correlation.

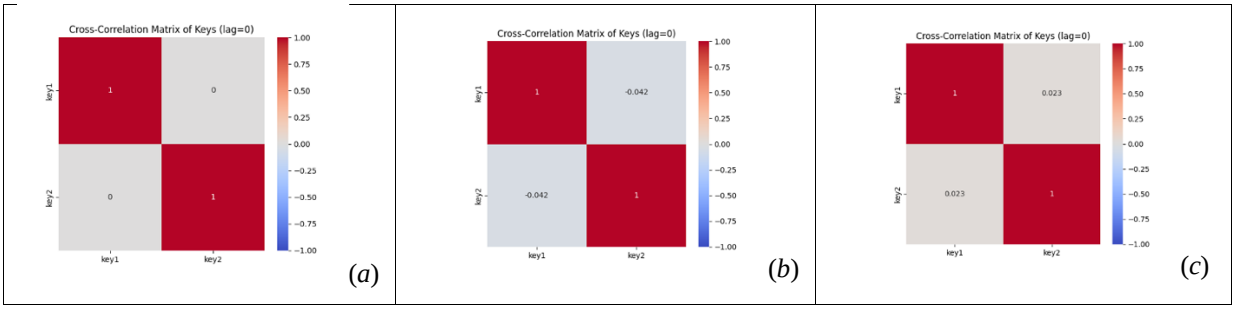


Fig. 7. Cross-Correlation Matrix of two Keys with three different Lengths of scenario-6- (*a*) represents 128-bit, (*b*) represents 192-bit, (*c*) represents 256-bit

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

TABLE XIV. P-VALUE FOR 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	0.908677	19.5820
192 bits	1	0.302895	23.9829
	2	0.925380	23.9829
256 bits	1	0.807748	27.6931
	2	0.570188	27.6931

Figure 8 represents a DFT, which analyzes the frequency components of a stream key bits. This chart evaluates the frequency-domain characteristics of a stream key bit. The x-axis represents frequency components, and the y-axis represents the magnitude of the Fourier coefficients for each frequency. The blue line plots the magnitude of the DFT for each frequency component. These represent how much each frequency contributes to the overall signal, while the horizontal red dashed line indicates a threshold level, labeled as "Threshold T." it used as a benchmark for evaluating significant frequency components. The randomness of the data represented in this figure can be assessed by analyzing the distribution of frequencies and their magnitudes in the DFT. In cryptographic randomness testing like DFT is used to detect periodic structures.

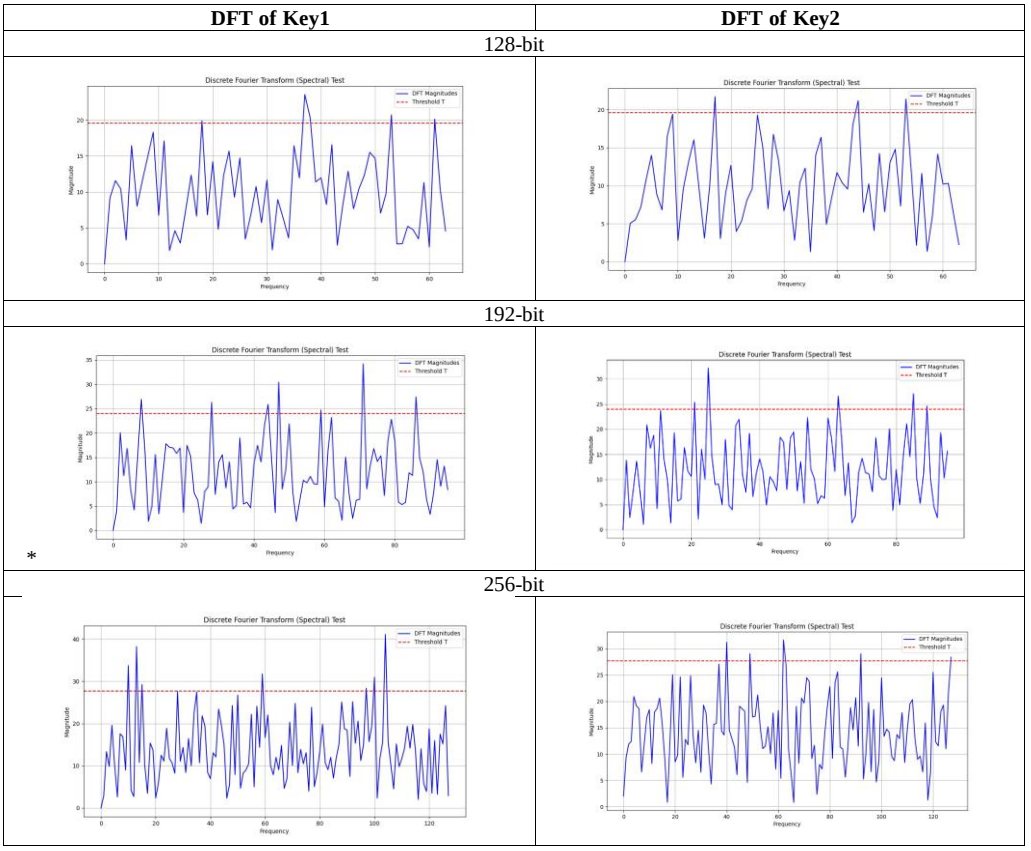


Fig. 8. Discrete Fourier Transform (DFT) of Two Keys with three different Lengths of scenario-6

8. CONCLUSIONS

In this paper, several proposed scenarios have been introduced to generalize the robust and strong key stream bits against the attacks. The proposed DRLKG-Chaotic system used six scenarios with five types of chaotic maps (Tent, Ikeda, Chua's, Rössler and Double Pendulum) and DRL A2C algorithm. In addition, the stream key bits generated by this system included several keys with different lengths (128, 192, and 256-bits). The key streams underwent comprehensive validation through multiple randomness assessment techniques. Bitstreams that successfully satisfied all seven statistical criteria demonstrate cryptographic viability and can be classified as statistically secure. These results confirm the resilience and strength of the key sequences generated across all experimental scenarios of our DRLKG-Chaotic framework. The successful NIST tests proof of unpredictable key stream bits is presented. The results analysis for mentioned tables and figures for all used tests (NIST, brute-force attack, AC, CC, and DFT) demonstrates significant improvements in cryptographic strength through using DRL with different five types of chaotic maps, these proposed scenarios offer improvement over standard chaotic systems. The results proved that, the six proposed scenarios are better for high-security requirements and demonstrate the significant improvements over standard maps.

Conflicts of interest

"No conflicts of interest."

Funding

The presented research did not receive any financial support.

Acknowledgments

The authors extend their sincere thanks to the Foundation for its cooperation and provision of the necessary facilities that contributed to the successful completion of this research.

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] 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.

[6] 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.

[7] 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.

[8] 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

[9] 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.

[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] 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.

[14] 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.

[15] 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.

[16] 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.

[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] E. İnce, B. Karakaya, and M. Türk, "Designing hardware for a robust high-speed cryptographic key generator based on multiple chaotic systems and its FPGA implementation for real-time video encryption," *Multimed Tools Appl*, vol. 83, no. 24, pp. 64499–64532, Jul. 2024, doi: 10.1007/s11042-023-17972-5.

[22] 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.

[23] 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.

[24] 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.

[25] 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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.
[33] 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
[34] 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.
[35] 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
[36] 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.
[37] 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.
[38] R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel, "A Survey of Zero-shot Generalisation in Deep Reinforcement Learning," Nov. 2021, doi: 10.1613/jair.1.14174.
[39] H. Dong, Z. Ding, and S. Zhang, *Deep reinforcement learning: Fundamentals, research and applications*. Springer Singapore, 2020. doi: 10.1007/978-981-15-4095-0.
[40] Kyriakos G., Vamvoudakis, Yan Wan, Frank L. Lewis, and Derya Cansever, *Handbook of Reinforcement Learning and Control*, vol. 325. in Studies in Systems, Decision and Control, vol. 325. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-60990-0.
[41] D. Hortelano *et al.*, "A comprehensive survey on reinforcement-learning-based computation offloading techniques in Edge Computing Systems," Jul. 01, 2023, *Academic Press*. doi: 10.1016/j.jnca.2023.103669.
[42] S. Qin *et al.*, "An Optimized Advantage Actor-Critic Algorithm for Disassembly Line Balancing Problem Considering Disassembly Tool Degradation," *Mathematics*, vol. 12, no. 6, Mar. 2024, doi: 10.3390/math12060836.
[43] 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
[44] A. Plaat, *Deep Reinforcement Learning*. Springer Nature, 2022. doi: 10.1007/978-981-19-0638-1.
[45] L. Canese *et al.*, "Multi-agent reinforcement learning: A review of challenges and applications," Jun. 01, 2021, *MDPI AG*. doi: 10.3390/app11114948.
[46] N. Ketkar and J. Moolayil, *Deep Learning with Python*. 2021. doi: 10.1007/978-1-4842-5364-9.
[47] C. Zhou, B. Huang, H. Hassan, and P. Fränti, "Attention-based advantage actor-critic algorithm with prioritized experience replay for complex 2-D robotic motion planning," *J Intell Manuf*, vol. 34, no. 1, pp. 151–180, Jan. 2023, doi: 10.1007/s10845-022-01988-z.
[48] M. Monaci, V. Agasucci, and G. Grani, "An actor-critic algorithm with policy gradients to solve the job shop scheduling problem using deep double recurrent agents," *Eur J Oper Res*, vol. 312, no. 3, pp. 910–926, Feb. 2024, doi: 10.1016/j.ejor.2023.07.037.
[49] "Custom Policy Network — Stable Baselines 2.10.3a0 documentation." Accessed: Dec. 13, 2024. [Online]. Available: https://stable-baselines.readthedocs.io/en/master/guide/custom_policy.html
[50] "Policy Networks — Stable Baselines 2.10.3a0 documentation." Accessed: Dec. 13, 2024. [Online]. Available: https://stable-baselines.readthedocs.io/en/master/modules/policies.html?highlight=net_arch
[51] 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.
[52] 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.
[53] 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.
[54] 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.
[55] J. Ayad and M. A. Jalil , Trans., "Robust Color Image Encryption Using 3D Chaotic Maps and S-Box Algorithms", *BJN*, vol. 2024, pp. 148–161, Aug. 2024, doi: 10.58496/BJN/2024/015.
[56] L. Hussain, "Fortifying AI Against Cyber Threats Advancing Resilient Systems to Combat Adversarial Attacks", *EDRAAK*, vol. 2024, pp. 26–31, Mar. 2024, doi: 10.70470/EDRAAK/2024/004.