



A novel deep learning-based statistical randomness evaluation test methodology for cryptographic applications

Seyfullah Kaner¹ · Ali Murat Garipcan² · Ebubekir Erdem³

Received: 4 June 2025 / Accepted: 4 September 2025 / Published online: 17 October 2025
© The Author(s) 2025

Abstract

The security of cryptographic systems is directly linked to the statistical randomness properties of the random numbers used. Traditional statistical randomness tests can be limited in evaluating the properties of these numbers and can require long processing times. While widely used test suites such as the National Institute of Standards and Technology (NIST) Special Publication (SP) 800–22 play a crucial role in assessing data randomness, they are slow on large data sets, can only evaluate certain statistical properties, and fail to detect complex data patterns and dependency structures. In this paper, we propose a new deep learning (DL) based method to overcome these limitations. In the study, bit sequences produced by an FPGA-based true random number generator (TRNG) and different pseudo-random number generators (PRNGs) were converted into image format, and classification experiments were carried out on the AlexNet, ResNet50, and EfficientNetB0 architectures. The results showed that AlexNet, with 87% accuracy and 99% recall, was by far the most successful method compared to the other models. Furthermore, the ablation analysis revealed that the components of data augmentation, early stopping, and cross-validation play a critical role in the model's stability and generalizability. Providing a reliable randomness evaluation with high accuracy, this new method, as an alternative to traditional statistical test suites, demonstrates that AI-based solutions can enhance the effectiveness and accuracy of randomness assessment in cryptographic applications. The success of DL methods in complex data structures offers significant potential for improving the security of modern cryptographic systems.

Keywords Deep learning · Statistical randomness testing · Cryptography · Artificial intelligence · Random numbers

1 Introduction

In the information age, rapidly developing technologies have made the need for data security and encryption mechanisms critical. Cryptography, one of the fundamental disciplines that meets these requirements, employs advanced mathematical methods and algorithms to ensure the

confidentiality, integrity, and authenticity of information. However, the reliability of cryptographic systems is tightly bound to the statistical randomness properties of the random numbers used, which are directly linked to their unpredictability. One of the fundamental building blocks of today's modern cryptographic systems, Kerckhoffs's Principle, emphasizes that a system must remain secure even when all its components are known except for the key. (Muhammad and Özkaynak 2021). This principle demonstrates that the security of cryptographic systems depends not only on algorithms but also on the generation of strong and unpredictable random numbers. This clearly reveals that secure cryptographic systems must rely not only on robust mathematical methods but also on reliable key generation techniques that ensure high accuracy and randomness (Özkaynak 2014). Therefore, the production of reliable and unpredictable random numbers continues to be one of the most significant technical and mathematical challenges faced by modern cryptography.

✉ Ebubekir Erdem
aberdem@firat.edu.tr

Seyfullah Kaner
kaner@alparslan.edu.tr

Ali Murat Garipcan
alimuratgaripcan@munzur.edu.tr

¹ Department of Computer Engineering, Muş Alparslan University, Muş, Turkey

² Department of Computer Engineering, Munzur University, Tunceli, Turkey

³ Department of Computer Engineering, Firat University, Elazığ, Turkey

The security of cryptographic systems depends not only on strong algorithms but also on reliable key generation that ensures statistical randomness properties. Accordingly, Random Number Generators (RNGs) are indispensable components for reliable key generation and encryption security in modern cryptographic systems. However, it is not enough for the random numbers generated by RNGs to have a random appearance; these numbers must have strong statistical properties associated with unpredictability, independence, and non-reproducibility, which can be defined by mathematical tools and methods (Özkaynak 2014). In order to verify the reliability and statistical adequacy of random number generators, many test methods have been developed, including Diehard (Marsaglia 2003), NIST SP 800–22 (Bassham et al. 2010), AIS20/AIS31 (Schindler 2023), ENT (Walker 2008), FIPS 140–2 (Caddy 2011) and TestU01 (L’Ecuyer and Simard 2007). However, conventional approaches primarily assess the surface-level statistical characteristics of generated sequences, often failing to capture intricate dependency structures. This shortcoming highlights the need for more powerful and comprehensive verification methods in the face of the increasing complexity of cryptographic systems. In this context traditional cryptographic randomness tests face challenges in meeting the evolving demands of modern cryptographic systems. In particular, widely used tests such as Diehard, NIST SP 800–22, and TestU01 only evaluate certain statistical properties, while failing to adequately analyze long-term dependencies and complex data relationships. Hence, there is an increasing demand for innovative methodologies that enhance randomness testing by incorporating deeper validation and advanced analytical techniques (Teh et al. 2020).

Traditional statistical randomness evaluation methods, while effective in analyzing certain fundamental statistical metrics, struggle to adapt to the growing complexity of cryptographic systems. These methods are generally effective in assessing fundamental properties of bit sequences, such as regularity and independence, yet they fail to detect complex dependency relationships and long-term patterns. Furthermore, performance bottlenecks emerge when processing large datasets, leading to increased computational overhead and reduced accuracy. This clearly shows that existing methods only provide superficial analysis and have a limited perspective on complex data structures. Moreover, since these approaches are mostly based on fixed rules and independence assumptions, they lead to decreased accuracy levels in variable and dynamic data environments. In today's cryptographic applications, there is a growing need for more flexible and powerful methods that can effectively analyze complex dependency structures and unpredictable patterns in data. The development of innovative solutions that can overcome the shortcomings

of existing methods has become a critical necessity to strengthen the security of modern cryptographic systems (Applebaum 2016; Ali et al. 2022; Hamza et al. 2020).

Artificial intelligence (AI) is playing an increasingly critical role in the field of cryptography, as in almost every field of technology today. With advanced capabilities including the analysis of complex datasets, detecting patterns and modeling unpredictable structures, AI offers innovative solutions for securing cryptographic systems. The relationship between these two disciplines is deepening at both theoretical and practical levels, and the powerful computational capacity offered by AI contributes significantly to the processes of analyzing the vulnerabilities of cryptographic protocols, detecting complex dependency structures and developing systems that are more resistant to attacks. Especially in the digital world, where data security is becoming more and more important every day, AI and cryptography stand out as two critical fields that complement each other (Rana et al. 2022; Coutinho et al. 2018; Torkzadehmahani et al. 2022). This mutual interaction enables both the development of stronger measures against existing security threats and the creation of next-generation security frameworks. In particular, AI-based systems have the potential to provide a much more comprehensive assessment than traditional randomness tests by identifying complex dependencies in data and analyzing unpredictable patterns (Choithani et al. 2024; Marwala and Xing 2018).

AI is increasingly gaining increasing adoption in cryptographic applications, providing effective solutions in critical processes such as data security (Alani 2019), verification and optimization, the design and analysis of encryption protocols (Coutinho et al. 2018), and intrusion detection systems (Capuano et al. 2022). In particular, AI and machine learning (ML) algorithms play a crucial role in tasks such as analyzing complex encryption structures, detecting potential vulnerabilities, and evaluating predictive attack scenarios. However, the advanced modeling and prediction capabilities of AI enable cryptographic protocols more flexible and robust (Sharma et al. 2019). In this context, the relationship between AI and cryptography is not only limited to strengthening existing systems, but also has the potential to serve as the foundation for next-generation security approaches (Choithani et al. 2024; Giudici 2018).

In this context, AI-based approaches present a robust alternative to overcome the limitations of traditional methods in statistical randomness assessment processes and expand the analysis capacity. In particular, DL algorithms can perform fast and highly accurate evaluations on large datasets thanks to their ability to detect complex dependency structures and unpredictable patterns (Coutinho et al. 2018; Khalid et al. 2023). Models such as CNN allow analyzing random sequences of numbers by converting them into image format and can effectively reveal patterns

that traditional statistical tests cannot detect. In this way, AI-based testing methods not only enhance processing time but also improve the accuracy of randomness assessment processes by providing more comprehensive and reliable analyses (Marwala and Xing 2018; Capuano et al. 2022). Furthermore, the automatic learning and adaptation capabilities of AI enhance adaptability to diverse data environments by making testing processes dynamic. Such innovative methods offer significant potential to improve the reliability of cryptographic systems and create more robust solutions against future threats. From a broader perspective, given the increasing complexity of modern cryptographic systems, the role of AI in providing security is becoming increasingly crucial (Choithani et al. 2024; Giudici 2018).

Consequently, traditional approaches to secure cryptographic systems fall short in addressing the increasingly complex threats and dynamic data environments in the modern world. In critical processes such as reliable randomness assessment and strong key generation, AI-assisted methods offer solutions that exceed the limits of existing methods thanks to both DL algorithms and advanced analysis techniques. The advanced modeling, prediction and adaptation capabilities of AI not only increase the reliability of existing systems, but also contribute to shaping future cryptographic standards. In this context, this study aims to improve the security level of modern cryptographic systems by demonstrating the advantages of AI-based methods over traditional statistical evaluation approaches. It is envisaged that the new perspective that this study will bring to the literature will provide important contributions in terms of both theoretical and practical applications.

In this study, an AI-assisted innovative approach with a hierarchical structure, illustrated in Fig. 1, is proposed to overcome the limitations of traditional methods and provide

a more comprehensive evaluation in randomness analysis. The proposed method utilizes not only raw bit sequences generated by an FPGA-based RO-TRNG (Altera Cyclone IV EP4CGX150DF31) but also post-processed FPGA data and sequences obtained from a PRNG, including both NIST-compliant and non-compliant sequences. This enabled the creation of a balanced dataset from four distinct data classes, ensuring that an equal number of samples from each class were used for unbiased model training. The collected data were processed using deep learning-based models, with AlexNet's 17-layer optimized architecture evaluated as the primary model, alongside comparative analyses with ResNet50 and EfficientNetB0 architectures. For use as model input, each bit sequence was transformed into an image format; sequences structured into specific matrices were converted into 224×224 pixel color images. By leveraging the strong representational capabilities of deep learning architectures on image data, this approach enabled the analysis of randomness properties not only statistically but also in terms of structural and dependency-based patterns.

Among the deep learning-based approaches employed in this study, the AlexNet, ResNet50, and EfficientNetB0 models were comparatively evaluated. The AlexNet model demonstrated the most successful performance among the models, achieving 87% accuracy and 99% recall during training and testing, providing a robust solution in terms of both speed and reliability. Although ResNet50, due to its deep architecture, possesses the capacity to learn complex features, it exhibited lower accuracy and a tendency to generate higher false positives. EfficientNetB0, on the other hand, presented a more balanced performance in terms of accuracy and F1 score but still lagged behind AlexNet's success. The TRNG- and PRNG-derived data used in the training process were evaluated in both raw

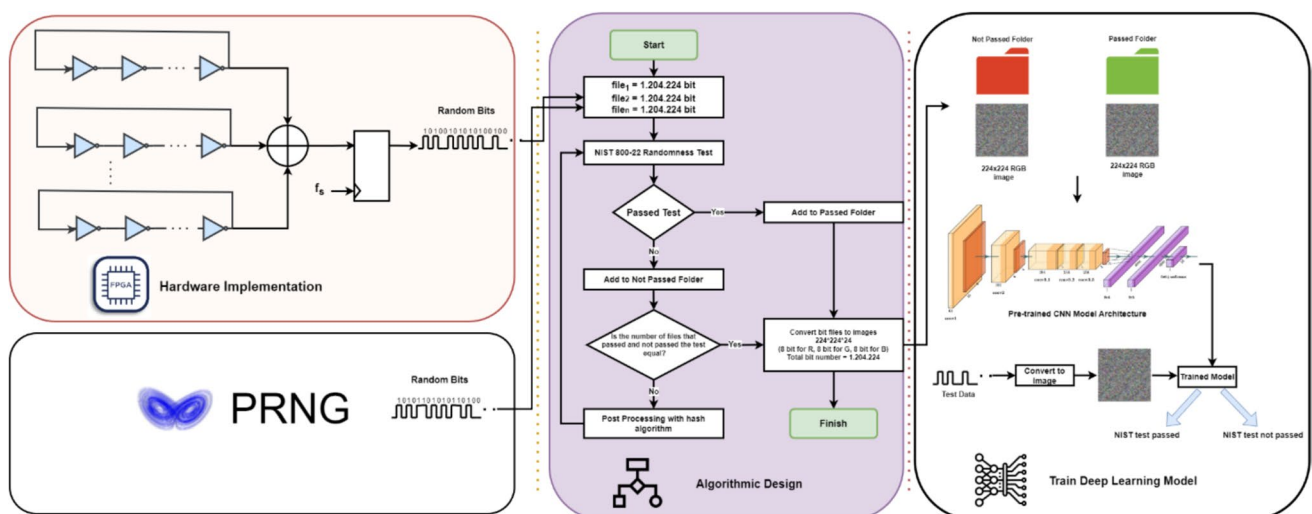


Fig. 1 Modeling the randomness test with DL

and post-processed forms, thereby incorporating not only real-time experimental data but also various artificial data sources into the analysis. This comprehensive approach not only meets the randomness evaluation requirements of modern cryptographic systems but also provides a more flexible, reliable, and extensive analytical process compared to traditional methods. The findings of the study demonstrate that deep learning methods yield effective results across critical performance metrics such as accuracy, precision, sensitivity, and processing speed. Consequently, the proposed approach introduces an innovative perspective to randomness analysis—a fundamental component of cryptographic security—and offers a robust solution capable of addressing the technological demands of the future.

1.1 Literature review and problem definition

In recent years, the relationship between DL and cryptography has strengthened, and remarkable studies have been carried out in this field. In the development of pseudorandom number generators (PRNGs) used in cryptographic applications, the advantages of chaotic systems have come to the forefront. In this context, Irfan et al. (2020) succeeded in increasing the randomness properties with a hyperchaotic logistic map-based PRNG design. Similarly, Kuznetsov et al. (2024) explored the potential of DL in cryptographic key generation based on biometric data and presented security solutions for the post-quantum era. On the other hand, Zaid et al. (2020) showed how DL-based neural networks can be effectively used in side-channel attacks. Zhao and Fan (2019) proposed a ML-based approach to detect different block cipher algorithms using Euclidean distance based feature extraction methods. Examining the impact of modern attack techniques on cryptographic devices, Gu et al. (2020) emphasized the importance of secure key generation and robust randomness testing. In addition, Perin et al. (2021) aimed to improve the accuracy of key recovery processes by developing a DL-based system that can recover private keys subjected to specific attacks.

Randomness testing is one of the cornerstones of cryptographic security, and traditional test suites used in this area include NIST SP 800–22 (Bassham et al. 2010), Dieharder (Brown 2006), and TestU01 (L'Ecuyer and Simard 2007). In addition, NIST SP 800–90B (Turan et al. 2018) provides minimum entropy estimation methods to evaluate the quality of entropy sources. On the other hand, tests such as frequency (Knuth 1997) and autocorrelation (Gentle 2003) are often used to analyze randomness deviations in generated sequences of numbers. However, these traditional methods have been reported to be inadequate, especially for high-dimensional data and for analyzing complex dependencies (L'Ecuyer and Simard 2007). At this point,

the integration of DL into randomness testing emerges as an important alternative. DL architectures such as AlexNet, with their automatic learning processes and feedback mechanisms, both improve evaluation accuracy and provide significant advantages in cryptographic applications (Hinton et al. 2006).

On the other hand, there are also studies to improve the performance of traditional methods. For example, a study on speeding up the NIST test suite has shown significant performance improvements (Sýs and Říha 2014). However, different approaches have been developed for applying ML and DL-based methods to randomness testing (Feng and Hao 2020; Li et al. 2020). Fischer (2018) proposed the use of artificial neural networks to evaluate cryptographically secure pseudorandom number generators (CSPRNG) and achieved superior performance compared to traditional tests. Similarly, Nagy and Suciu (2021) have succeeded in improving the accuracy of statistical tests in the NIST test suite by supporting them with neural networks. Wu et al. (2011) developed statistical hypothesis tests based on Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) measurements for the evaluation of image encryption algorithms.

The high modeling capacity offered by DL is not only limited to increasing the reliability of random number generators, but has also been used in the analysis of cryptographic structures. In this direction, DL-based analyses have been performed for the prediction of active numbers in S-box structures of block ciphers (Idris et al. 2021). Similarly, ML methods have been utilized to evaluate the robustness of optical continuous variable quantum random number generators (QRNG) against classical noise (Truong et al. 2018).

Recent advancements in image steganography have demonstrated the potential of color transformation techniques for secure information embedding. Li et al. (2024) introduced a novel approach that leverages de-colorization and re-colorization processes to conceal data within color channels, effectively enhancing resistance against conventional steganalysis. Their method not only preserves visual fidelity but also exploits the perceptual characteristics of color conversion to obfuscate hidden patterns. Although the present study focuses on evaluating the randomness of TRNG and PRNG outputs using deep learning models, the conceptual parallels with Li et al.'s work—particularly in terms of leveraging transformation domains for security—highlight the broader applicability of AI-driven methodologies in cryptographic systems.

Moreover, recent studies combining chaotic systems with neural networks for randomness generation and cryptographic applications have drawn attention. In these approaches, sequences obtained from chaotic maps are integrated with AI-based models, providing advantages

such as high entropy, wide chaotic parameter ranges, and fast computation, particularly in the areas of image encryption and pseudorandom number generation (Gao et al. 2024, 2025a). These studies align with our proposed approach, which relies on deep learning-based classification of FPGA-based TRNG and PRNG data, reflecting current trends in the literature.

In addition to randomness evaluation studies, recent research has also examined the security of data hiding and encryption schemes against known-plaintext attacks. For instance, investigations into reversible data hiding in encrypted images have revealed potential vulnerabilities when block permutation and co-modulation techniques are applied, as well as limitations under redundant space transfer approaches (Lingfeng et al. 2022; Qu et al. 2024). Although the present study focuses on TRNG and PRNG randomness analysis with deep learning models, considering such findings highlights the broader relevance of robustness against known attack vectors in cryptographic systems.

Furthermore, approaches such as Neural Cryptography, where neural networks are employed for secure key exchange, have introduced new perspectives to the field. AI-based models such as autoencoders and Generative Adversarial Networks (GANs) have also been effectively used for encrypting multimedia data (Nitaj and Rachidi 2023; Oseni et al. 2021). In conclusion, the literature demonstrates that DL provides significant contributions across a wide range of cryptographic applications, from randomness testing and secure key generation to complex attack analysis. Considering the increasing volume and diversity of data, DL-assisted approaches are expected to play a critical role in enhancing the reliability and resilience of cryptographic systems in the future.

1.2 Motivation and contributions

Tests for statistical randomness assessment in cryptography use different mathematical methods to determine the statistical properties of random numbers. These tests, however, lack a single mathematical definition of randomness, making it impossible to create a universal test or a perfect method. While each test analyzes certain properties (e.g. NIST smoothness and independence, DIEHARD complex dependencies), it may ignore other aspects. This leads to contradictions or discrepancies between the results of the tests due to the focus on different and limited number of statistical properties on the same sample sets. Furthermore, limitations of the tests, such as different data requirements, inability to detect dependencies and patterns, inconsistencies in mathematical assumptions, and false positive/negative results, may cause a sequence that succeeds in one test to fail in another test. These limitations necessitate the use of

multiple tests together in randomness analysis and suggest the need to develop more comprehensive approaches. This is the main motivation for this study and necessitates the development of generalizable AI-based methods to overcome the statistical limitations of existing tests. In this way, it is aimed to create more scalable, fast and accurate randomness evaluation processes that can detect complex dependency structures with high accuracy and meet the requirements of modern cryptographic systems. The core contributions of the proposed method, whose hierarchical representation is shown in Fig. 1, can be summarized as follows:

- The reliance on experimental data not only from theoretical models but also directly from TRNG outputs produced in real-time FPGA environments and sequences obtained from PRNGs increases the reliability of the method and supports its applicability in cryptographic applications.
- The proposed approach provides the ability to detect complex dependency structures and long-term patterns in random numbers with high accuracy, which are often undetectable by traditional statistical tests.
- In this study, the AlexNet, ResNet50, and EfficientNetB0 architectures were comparatively examined; the results demonstrated that AlexNet achieved the highest performance with 87% accuracy and 99% recall, while the other two architectures exhibited relatively lower performance.
- Additionally, ablation analyses detailed the contributions of components such as data augmentation, early stopping, and cross-validation to the performance and stability of the proposed method.
- The proposed method offers a scalable solution by optimizing processing time on large datasets and provides the potential for generalizable randomness analysis.

In terms of contribution to the literature, this methodology fills an important gap in randomness evaluation by offering advanced detection capabilities. This DL-based approach stands out for its applicability on various sources of randomness, such as hardware RNGs, quantum RNGs, etc., strengthening the design and security of cryptographic protocols. At the same time, by opening the door to new areas of research, such as transfer learning or adversarial robustness, this methodology not only improves existing applications but also lays a solid foundation for future research. In this context, it has the potential to be a benchmark for setting the next generation of standards and redefining randomness testing in terms of sensitivity, accuracy and scalability.

1.3 Paper organisation

The remaining sections of this paper are organized as follows: In Sect. 2, the concept of randomness, random number generators, and the role of randomness in cryptography are discussed, along with the limitations of existing traditional testing methods. Section 3 introduces the dataset generated using FPGA-based RO-TRNG and various PRNG sources, detailing the transformation and preprocessing steps as well as the data distribution. Section 4 presents the experimental studies conducted on the AlexNet, ResNet50, and EfficientNetB0 models, including the training and validation processes, hyperparameter settings, cross-validation methods used, and performance evaluation metrics. This section also provides findings related to confusion matrices, prediction outputs, training and validation losses, and classification performance through figures and tables. In Sect. 5, the obtained results are discussed, and the comparative performance of the models is evaluated. Finally, the last section summarizes the overall conclusions of the study, highlights the strengths and limitations of the proposed methodology, and provides recommendations for future research directions.

2 Theoretical background

This section addresses the fundamental principles of existing randomness tests used in cryptographic applications, the traditional limitations of these tests, and the potential contributions of AI-based innovative approaches. Additionally, the structure, functionality, and limitations of widely used test suites, such as NIST SP 800–22, are explained to discuss the extent to which these tests meet modern cryptographic requirements. Furthermore, the superior performance of DL methods, particularly CNN-based approaches, in analyzing complex data structures is elaborated, with a focus on the application of models such as AlexNet in randomness testing. In this context, the section aims to present the theoretical background underpinning the study and to explain the technical foundations of the methods employed.

2.1 Randomness, random numbers and cryptography

In cryptography, randomness plays a critical role in information security. Randomness is a fundamental element that increases the resilience of a system against attacks by making it unpredictable. The security of cryptographic algorithms depends on the unpredictability of the keys and other secret parameters used. For example, if the key used in an encryption algorithm contains a certain regularity or

predictability, attackers can guess this key and the security of the system can be compromised. Therefore, cryptographic keys are based on high-quality random values obtained from trusted random number generators. The security of today's modern cryptographic systems relies heavily on the statistical properties of the random numbers used (Simion 2020; Manucom et al. 2019; Mannalatha et al. 2023).

There are some basic requirements that random numbers used in cryptographic systems must meet. First, they must be unpredictable and unpredictable by outside observers. Secondly, the numbers generated must be independent of each other and there must be no patterns or correlations between them. Also, the random numbers must fit a certain distribution. For example, each number is expected to have an equal probability of being chosen. If these requirements are not met, serious security vulnerabilities can arise in cryptographic systems (Mengdi et al. 2021; Aljohani et al. 2019).

The basic statistical requirements for random numbers used in cryptographic systems also determine the characteristics of the generators that produce them. Properties such as unpredictability, independence and statistical stability are required not only for the generated numbers, but also for the algorithm or hardware that generates them. PRNGs generate random sequences at high speed from a given seed value based on mathematical algorithms. However, their deterministic nature limits their use in high-security situations. TRNG generates completely unpredictable and high-entropy numbers by exploiting physical processes such as electronic noise or quantum events. Therefore, TRNGs have become an indispensable component of modern cryptographic systems. Consequently, randomness analysis and generation must be rigorously evaluated not only in terms of the numbers generated, but also in terms of the architecture and operation of the generators supporting this process (Jacak et al. 2021; Garipcan and Erdem 2020; Koç, 2009).

2.2 Randomness test suites and their limitations

Assessing the suitability of random number sequences for cryptographic applications is possible by verifying their statistical adequacy. This verification is usually done with observational statistical tests, but these tests do not take into account the methods by which the number sequences are obtained and the internal structure of the generator. These testing tools try to detect erroneous distributions in the number sequences and sub-patterns that show statistical dependence. The ability of an ideal random number generator to generate numbers with equal probability is a significant challenge for the random numbers being tested. Therefore, it is difficult to conclusively determine whether a generator produces ideal random numbers based on

observational test results alone. Furthermore, these tests can produce false positives and lead to erroneous assessments of random numbers.

Randomness tests usually consist of three basic steps to assess whether a sequence has certain statistical properties. In the first step, the statistical properties of the tested sequence of numbers are analyzed by defining a variable function of the form $F(X)$. In the second step, the probability distribution of the sequence of the form $X(\sigma) = tp, \sigma \in \Omega, X : \Omega \rightarrow R$ is calculated. This is done in order to check whether the tested sequence of numbers conforms to a given statistical distribution. In the last step, the test statistic is evaluated in terms of the null hypothesis (H_0). The null hypothesis (H_0) assumes that the examined number generator is indistinguishable from a true random number generator and this hypothesis is tested. The alternative hypothesis (H_a) asserts that the number generator is not random. In this process, the acceptance or rejection of the null hypothesis allows one to decide whether the tested sequence satisfies the randomness criteria. The types of errors that can be encountered for both cases are shown in Table 1 (Bassham et al. 2010; Akcengiz et al. 2024).

α let be the predefined significance level for the null hypothesis (H_0). In this case, the statistical randomness test is evaluated based on the probability $X(\sigma) = tp$ and is calculated as follows.

$$P(X(\sigma) = t_p) \begin{cases} \leq \alpha & \sigma \text{ is not random.} \\ > \alpha & \sigma \text{ passed the test.} \end{cases} \quad (1)$$

In statistical randomness tests, the initial assumption that the sequences being tested come from a random source is called the null hypothesis (H_0). The null hypothesis assumes that the sequence is derived from a random number generator, and the test is performed to evaluate the accuracy of this assumption. The tests work on a given randomness statistic and the values of this statistic are used to decide whether H_0 should be accepted or rejected. The alternative hypothesis (H_a) states that the sequence is not random. If the test criteria are not met H_0 is rejected and H_a is accepted. For each test, the distribution of possible values of the statistic under the randomness assumption is determined and the theoretical reference distribution is calculated mathematically. In the final step, the critical value

is compared with the test statistic to decide whether the null hypothesis should be rejected.

Although conventional randomness tests provide an important foundation for cryptographic systems, they are limited by the complexity and increasing security requirements of modern systems. In particular, tests such as NIST SP 800–22 analyze basic properties such as independence and uniformity, while ignoring complex dependency structures and long-term patterns. More advanced tools, such as DIEHARD and TestU01, offer a wider range of analyses, but can increase computational costs and show limited flexibility on large data sets.

These shortcomings make it clear that modern and scalable solutions need to be developed. In this context, DL-based methods are considered as a powerful alternative that can complement the shortcomings of conventional tests, offering broader analysis capabilities and high accuracy rates. With its capacity to detect complex dependency structures and efficiently handle large data sets, DL has the potential to provide a more flexible and comprehensive solution that meets modern cryptographic needs.

2.3 Deep learning and convolutional neural network architecture

DL, as a sub-branch of ML, has a wide range of applications today with its capacity to model complex data patterns and relationships. Especially in areas such as big data analytics, image processing, natural language processing and cyber security, significant success has been achieved. This technology, derived from artificial neural networks, has the ability to discover hidden patterns in the data and make predictions using these patterns thanks to its multilayer network structures. The success of DL is based on the development of optimization algorithms that increase the complexity and accuracy of these models, along with access to large datasets and high processing power (Kim et al. 2022; Mehmood et al. 2024; Kishore et al. 2023).

CNN is one of the most widely used and effective architectures of DL models, especially in visual data processing. CNNs are characterized by their ability to extract meaningful features from input data without human intervention. The architecture consists of different components: convolutional layers, pooling layers and fully connected layers. Convolutional layers extract important features from the data, while pooling layers function to reduce size and prevent overlearning. The feature extraction provided by these layers enables the model to learn complex dependency structures and detect patterns in the data. CNNs can be used in a wide range of applications such as image recognition, medical image analysis and natural language processing, providing optimized solutions in terms of accuracy and computational efficiency compared

Table 1 Hypothesis test outcomes for randomness evaluation

Randomness Source	Conclusion	
	H_0 Acceptance Status	H_a Acceptance Status
Generator Random	No Error	Type I Error
Generator Not Random	Type II Error	No Error

to traditional neural networks (Das et al. 2019; Liu et al. 2022; Salem et al. 2018).

The use of CNN in randomness testing offers significant advantages in overcoming the shortcomings of traditional statistical tests. In particular, the ability to detect complex dependency structures and identify long-term data relationships makes the integration of CNNs into cryptographic security systems important (Priya et al. 2024). Mathematically, the fundamental operation of a CNN, convolution, extracts specific features from an input matrix through the application of filters and can be expressed as shown in Eq. 2:

$$y_{ij} = f\left(\sum_{m,n} x_{i+m,j+n} \cdot w_{m,n} + b\right) \quad (2)$$

Here x represents the data points in the input matrix, w denotes the learned filters, b the bias value and f the activation function. This operation allows CNNs, when analyzing random number sequences, to go beyond fundamental randomness criteria such as statistical independence and uniformity. Dependency relationships and patterns that traditional tests may overlook can be examined in greater depth due to the multilayered structure of CNNs.

In this study, different deep learning architectures were comparatively evaluated to enhance randomness testing, specifically employing the AlexNet (Krizhevsky et al. 2017), ResNet50 (He et al. 2016) and EfficientNetB0 (Tan and Le 2019) models. AlexNet is a widely recognized CNN architecture in deep learning and, with its 17-layer optimized modern version, stands out in analyzing the randomness levels of number sequences. ResNet50, on the other hand, possesses the capacity to learn more complex features through residual connections, which are designed to mitigate learning losses caused by deep consecutive layers. EfficientNetB0 is an optimized CNN architecture that provides a balanced performance in terms of accuracy while maintaining relatively low computational cost. The combined evaluation of these models enables a comparative analysis across architectures with varying depths and parameter optimizations, rather than relying on a single model.

In this study, different deep learning architectures were employed to advance randomness testing beyond the capabilities of traditional methods. The multilayered structures of AlexNet, ResNet50, and EfficientNetB0 provide the ability to analyze a wide spectrum of features, ranging from low-level characteristics to high-level abstractions. In particular, converting bit sequences into image format enables a more detailed and visually supported analysis of the data. This approach allows both the optimization of processing speed and the more reliable evaluation of the tested sequences. Thanks to their GPU-based parallel

processing capabilities and deep learning strengths, all three models facilitate efficient analysis of large-scale datasets, offering innovative solutions aligned with the security requirements of modern cryptographic systems (Tan et al. 2021; Deebak and Hwang 2025; Tolba 2024).

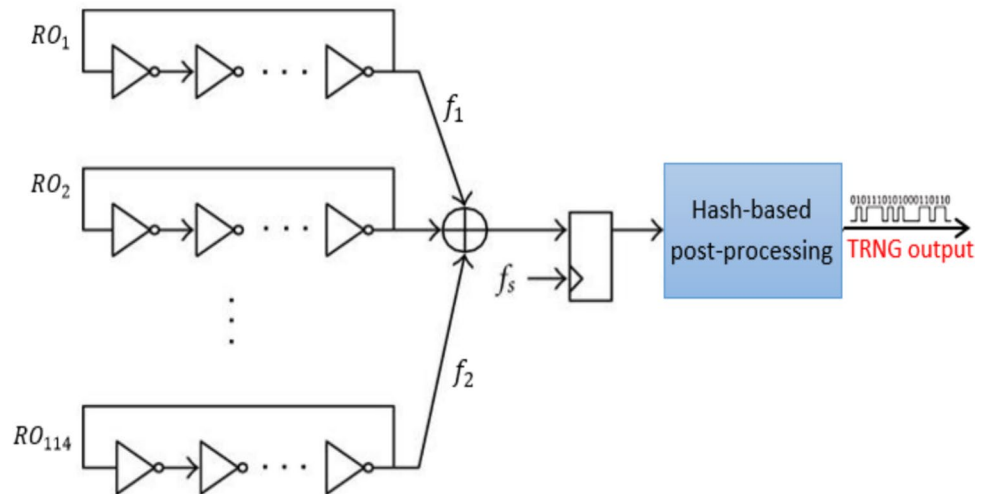
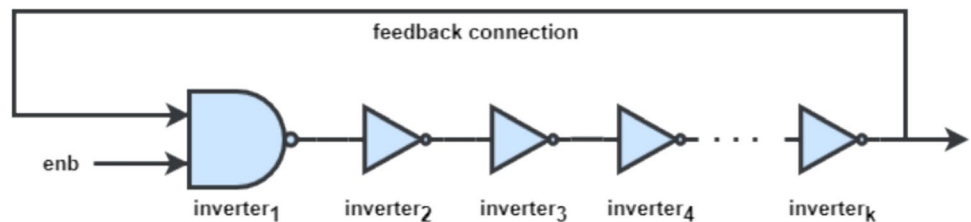
In conclusion, the CNN-based approaches employed in this study have transcended the limitations of traditional statistical tests, providing a more comprehensive and highly accurate evaluation process. The findings indicate that AlexNet outperformed the other models with 87% accuracy and 99% recall, whereas ResNet50 exhibited a tendency for false positives despite lower accuracy, and EfficientNetB0 demonstrated a more balanced but relatively lower performance. Deep learning models such as AlexNet not only enable the rapid and efficient analysis of large-scale datasets but also detect dependencies and patterns that may be overlooked by conventional tests. This underscores the potential of deep learning methods to enhance the accuracy and reliability of randomness testing in cryptographic systems.

2.4 Jitter based true random number generation

In this work, we adopt a jitter-based approach for random number generation and choose an RO-based TRNG architecture known as the Sunar (Sunar et al. 2007) model. This architecture is chosen because it can be easily integrated due to its simple hardware design, as well as its ability to provide the non-deterministic definition of the noise source in TRNG by itself and thus generate a random bit stream at high speed. This method, which is widely used in the literature, has the advantage that it can be implemented in integrated devices without the need for analog-to-digital conversion.

The TRNG architecture, schematically shown in Fig. 2, comprises 114 parallel-connected ROs consisting of 13 logic gates, whose combinational description is given in Fig. 3. The phase jitter generated by these oscillators is the result of a combination of local phenomena, such as thermal and flicker noise, and phenomena caused by the operating conditions of the device (e.g. temperature and voltage variations). As shown in the figure, the oscillator outputs are sampled with the help of a D Flip-Flop (DFF) combined with an XOR network to generate true random numbers at bit level.

The number of clock signals in the RO cycle can be calculated using Eq. 3, assuming constant time for each inverter. In practice, however, the propagation and routing delay in each logic gate introduces an unstable delay that causes periodic irregularities in the signals. This irregularity is called phase jitter and acts as a source of randomness. These random variations in the clock signals result from a combination of local (thermal and flicker noise) and global

Fig. 2 RO-based TRNG architecture**Fig. 3** RO combinational notation

(power supply and temperature variations) phenomena of the device and are unpredictable within any sampling period (Sunar et al. 2007). Therefore, in addition to Eq. 3, the mathematical definitions in Eqs. 4 and 5 are necessary to obtain a more accurate definition for RO. In Eqs. 4 and 5, D_i and R_i are fixed delay times corresponding to the nominal supply and temperature level of the digital device. In both equations, real random delay variations ($\Delta d_{Li} - \Delta r_{Li}$ and $\Delta d_{Gi} - \Delta r_{Gi}$) due to local and global events, respectively, cause jitter. The direction and magnitude of these variations in the clock signals are inherently unpredictable for any sampling period (Garipcan and Erdem 2022).

$$T = 2 \sum_{i=1}^k (d_i + r_i) \quad (3)$$

$$d_i = D_i + \Delta d_i = D_i + \Delta d_{Li} + \Delta d_{Gi} \quad (4)$$

$$r_i = R_i + \Delta r_i = R_i + \Delta r_{Li} + \Delta r_{Gi} \quad (5)$$

In Eq. 3, d_i is the latency of the logic gate, r_i is the gate routing delay, k is the number of logic gates in the loop. In Eqs. 4 and 5, D_i and R_i represent the constant delay in the nominal conditions of the device, Δd_{Li} and Δr_{Li} represent the random variation due to local events, and Δd_{Gi} , Δr_{Gi} represent the variation due to global events. The TRNG

architecture, from which the data used in the training and validation processes of the proposed DL model were obtained, was implemented in real time on the Altera Cyclone IV EP4CGX150DF31 FPGA development board, and the experimental setup is as shown in Fig. 4.

2.5 Random number generation with PRNG

Alongside TRNG, datasets were also obtained from PRNG to enable comparative analysis in this study. These datasets consisted of both high-quality sequences that passed the NIST SP 800–22 test suite and sequences deliberately modified to fail certain NIST tests by introducing artificial distortions. This approach allowed the proposed deep learning-based classification method to be evaluated not only on truly random sequences but also on artificially patterned and insecure sequences.

The generation of PRNG data employed a locally biased strategy. In this method, a dominant bit was selected for each block and repeated with a certain probability (bias = 0.8). This created an imbalanced bit distribution within the block, producing PRNG sequences that specifically failed NIST's Frequency Within a Block and Serial tests. This strategy was chosen because it is both straightforward to implement and widely recognized in the literature, allowing for controlled generation of synthetic sequences that would not pass NIST

Fig. 4 Real-time experimental setup of the TRNG architecture



tests. The mathematical representation of this approach is given in Eq. 6:

$$x_i = \begin{cases} b & \text{with probability } \alpha \\ 1 - b & \text{with probability } 1 - \alpha \end{cases} \quad (6)$$

Here, b denotes the dominant bit, and α represents the bias ratio.

Additionally, high-quality PRNG sequences without any artificial distortions were generated using Python's secrets library. These sequences successfully passed the NIST tests and were used to evaluate the models' ability to correctly classify secure random sequences.

The dataset created through these strategies enabled a comprehensive training and testing process using both hardware-generated TRNG data and synthetic PRNG sequences. This approach ensured that the model could effectively distinguish between "random" and "non-random" bit sequences with high accuracy.

3 Implementation of deep learning model and findings

In this section, the randomness levels of the bit sequences generated by the TRNG were evaluated using both the NIST SP 800–22 test suite and a deep learning (DL)–based method. By combining the reliability provided by the NIST test suite with the ability of DL models to detect patterns in complex data structures, a comprehensive analysis was conducted. The evaluation was carried out in three stages: First, the fundamental statistical properties of the bit sequences were examined using the NIST SP 800–22 test suite, and the randomness levels were determined based on

the p-values. Next, the same bit sequences were converted into image format to enable analysis with CNN-based deep learning models, which were trained to detect data dependencies. The performance of the model was assessed using the Area Under the Curve (AUC) metric. Finally, the NIST test suite and the DL model were compared in terms of accuracy, processing time, and efficiency. The results demonstrate that the DL-based method, when combined with the NIST test suite, can provide a new perspective for randomness analysis and, in particular, optimize processing time for large-scale datasets.

3.1 Acquisition and preprocessing of the dataset

In this section, the creation and preprocessing of the dataset used in the experimental study are presented. As a real-time setup, the RO-based TRNG shown in Fig. 2 was employed, and bit sequences obtained at the hardware level were used as one of the primary data sources for the proposed models. Additionally, for comparative analysis, bit sequences generated from a standard PRNG were also included in the dataset. The raw data obtained from both TRNG and PRNG were subjected to preprocessing steps according to the logical scheme presented in Fig. 5, in order to enhance the reliability of logical validation and to prepare the data for the model training process. Bit sequences labeled based on their statistical randomness properties were subsequently transformed into image format, resulting in the formation of both training and testing datasets.

The study utilized not only the raw FPGA-based RO-TRNG data but also post-processed FPGA outputs and standard PRNG-generated sequences. Given the high time cost associated with generating FPGA data, 2,000 sample files were produced for each class to ensure a balanced

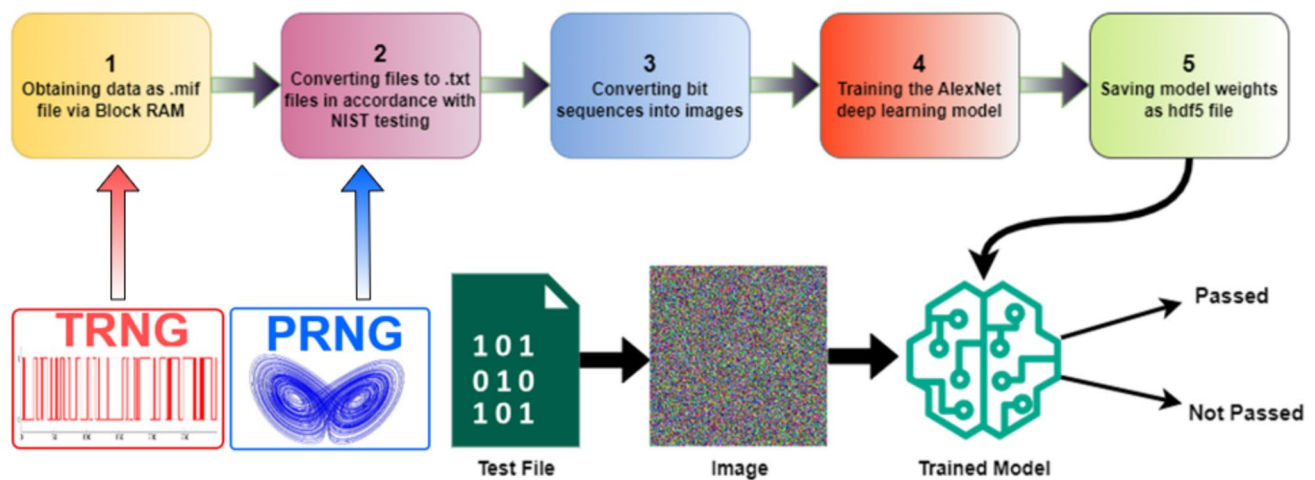


Fig. 5 Preprocessing stages of bit sequences obtained from TRNG and PRNG for training with DL model

distribution, resulting in a total of 8,000 files forming a balanced dataset. These four classes correspond to: raw FPGA data, post-processed FPGA data, PRNG sequences passing the NIST tests, and PRNG sequences failing the NIST tests. This approach ensured that the model was not solely dependent on a single hardware-based source, enabling a broader evaluation by considering diverse data types and levels of randomness.

In the first step of the preprocessing stage, the bit sequences with the “.mif” extension obtained from the FPGA environment were converted into a format compatible with the NIST SP 800–22 test suite. In this phase, 4000. mif files, each containing 1, 04, 224 bits, were transformed into equivalent “.txt” test files with a single row and single column format, as illustrated in Fig. 6, to facilitate statistical randomness analysis. The transformed test files were then

subjected to the NIST SP 800–22 tests for labeling. During this labeling process, files that met the test criteria were classified as “passed,” while those that failed were labeled as “not passed.” This step represents a critical preprocessing procedure to ensure that the data is suitable for input into the deep learning model.

In the next step, the test files were converted into RGB image data with dimensions of 224×224 pixels. During this transformation, each pixel was represented by a 24-bit data block, where the first 8 bits were assigned to the red (R) channel, the next 8 bits to the green (G) channel, and the final 8 bits to the blue (B) channel. This configuration was implemented according to the classical RGB (TrueColor) model, using a fixed 8-bit color depth for each channel. The bit sequences were scanned row by row (left to right, top to bottom) and mapped to the pixel matrix, thereby preserving

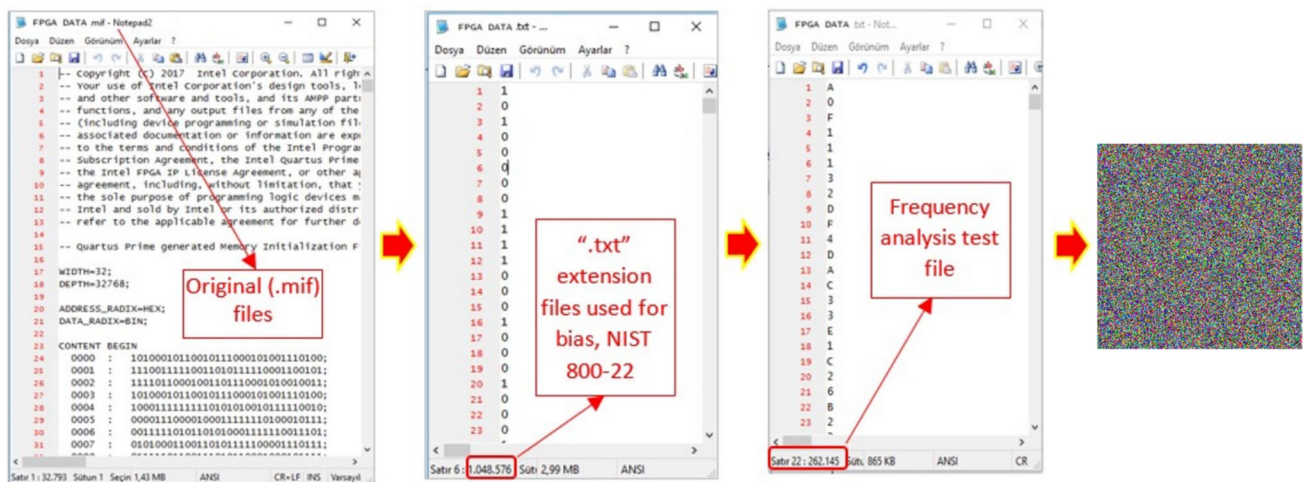


Fig. 6 Transformation stages of real random number/bit sequences for statistical randomness and DL model

the data order corresponding to Big-Endian format. This approach prevented encoding errors related to endianness or color depth inconsistencies. The methodology followed for converting the bit sequences into images is illustrated in Fig. 7.

Ensuring a fixed and consistent mapping of the data in this manner is critical for classification performance. Even a minor deviation in the RGB channel order or a distortion in color depth can lead to misinterpretation of the structural patterns that the model is attempting to learn. Therefore, the conversion process was carried out using a total of 1,204,224 bits of data per image, with meticulous checks to ensure that no gaps or missing data remained at the end of the files. Additionally, try-except mechanisms were implemented during encoding to detect potential errors; any samples containing corrupted or incomplete data were

excluded from the system, ensuring that only valid and structurally consistent images were included in the training process. This approach allowed the model to perform reliable randomness analysis both statistically and visually. The parameter set for the test criteria and the metrics related to sequence lengths are provided in Table 2.

After completing the conversion and preprocessing steps, the dataset was evenly split to ensure balanced classification during the model's learning process. The structure of the dataset is presented in Table 3. Out of the total 8,000 images, 6,400 were allocated for training and 1,600 for validation. The dataset was evenly divided between the "passed" and "failed" classes, with each class containing 3,200 training images and 800 validation images. This distribution was optimized to ensure that the model accurately learns and generalizes from both randomness-compliant and

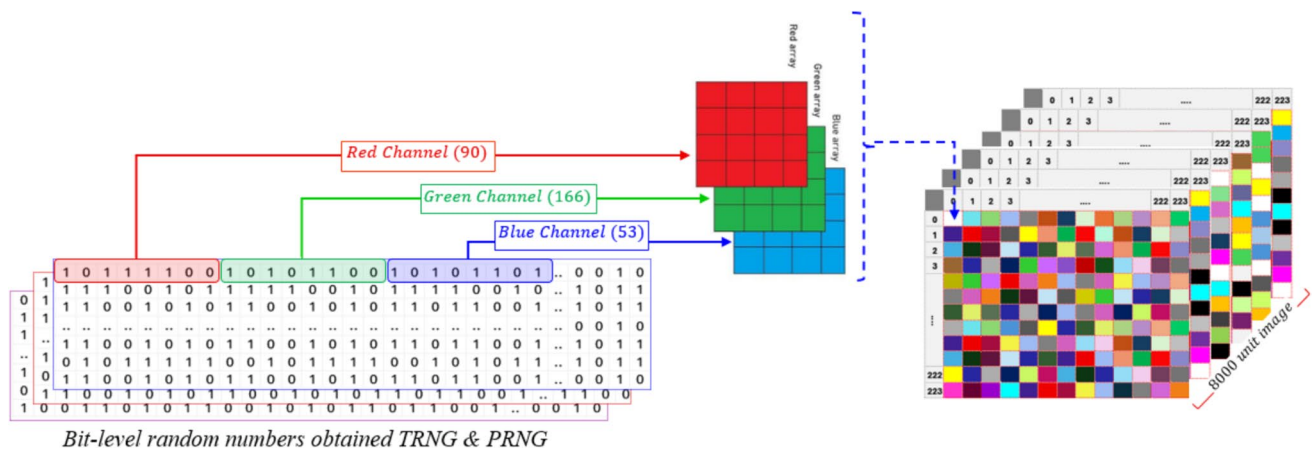


Fig. 7 Converting bit strings to image format

Table 2 Probabilistic distributions and input sizes of NIST 800–22 test criteria

Test No	Test Name	Reference Distribution	Size
1	Frequency test	<i>Halfnormal</i>	$n \geq 100$
2	Frequency test within a block	$\chi^2(\text{chi} - \text{square})$	$n \geq 100$
3	Run test	χ^2	$n \geq 100$
4	Test for the longest run of ones in a block	χ^2	$n \geq 128 - 750,000$
5	Binary matrix rank	χ^2	$n \geq 38,912$
6	Discrete Fourier transform	<i>Normal</i>	$n \geq 100$
7	Non-overlapping template matching	χ^2	$n \geq 10^6$
8	The overlapping template matching test	χ^2	$n \geq 10^6$
9	Maurer's universal statistical test	<i>Halfnormal</i>	$n \geq 387,340 - 109$
10	Linear complexity test	χ^2	$n \geq 10^6$
11	Serial test	χ^2	$m \leq \log_2(n) - 2$
12	Approximate entropy test	χ^2	$m \leq \log_2(n) - 2$
13	Cumulative sums	<i>Normal</i>	$n \geq 100$
14	Random Excursions	χ^2	$n \geq 10^6$
15	Random Excursions Variant	<i>Halfnormal</i>	$n \geq 10^6$

Table 3 Data set structure

Class	Number of training data sets	Number of validation data sets	Explanation
TRNG-Passed the test	1,600 image	400 image	FPGA-based TRNG data passing NIST tests after post-processing
TRNG-Failing the test	1,600 image	400 image	Raw FPGA-based TRNG data failing NIST tests
PRNG-Passed the test	1,600 image	400 image	High-quality PRNG data generated using the Python <code>secrets</code> library (NIST-compliant)
PRNG-Failing the test	1,600 image	400 image	PRNG data generated with local bias methods, failing NIST tests

non-compliant bit sequences. As a result, overfitting to any single class during training was prevented, enabling statistically balanced learning.

To mitigate overfitting during training, Dropout and early stopping mechanisms were employed. A Dropout rate of 50% was applied, randomly disabling certain neurons to facilitate a more balanced learning process. The early stopping mechanism was activated with a minimum of 50 epochs, terminating training when an increasing trend in validation loss (`val_loss`) was observed. This ensured that the model did not overfit the training data, preserving the optimal weights based on validation performance. Examination of the training and validation curves revealed that convergence in the AlexNet model stabilized around epochs 60–70, whereas validation performance in the ResNet50 and EfficientNetB0 models remained limited to the 50–56% range. These criteria indicate that potential overfitting situations were systematically identified and early stopping was applied according to precise rules.

3.2 Model training and hyperparameter optimization

After completing the preprocessing and data transformation steps, the training of deep learning-based models was carried out. The bit sequences, prepared to be compatible with the model input, were represented as 224×224 pixel RGB images and analyzed using three different CNN architectures. The first of these, AlexNet, a pioneering model in the field of image classification, was used in the experiments in its optimized 17-layer version. The second model, ResNet50, adopts the “residual connection” structure to address the vanishing gradient problem in deep networks, enabling the effective training of deeper layers. The third model, EfficientNetB0, offers a more balanced architecture with parameter efficiency and scalability, delivering competitive performance at a relatively low computational cost.

During the training of these models, various hyperparameter optimization techniques were applied to improve performance and maintain generalization capabilities. Learning rates, activation functions, and weight

updates were carefully tuned, and regularization methods such as early stopping and dropout were employed to prevent overfitting. Upon completion of the training process, the performance of all three models was comparatively evaluated on the test datasets.

Following training, the models’ performance was assessed using the test data. The input images were processed by the CNNs for feature extraction, and weight updates were optimized to produce the hierarchical model shown in Fig. 8, capable of making accurate predictions with high accuracy on the test data. Within this context, AlexNet demonstrated higher accuracy and more stable learning performance based on the input images, whereas ResNet50 and EfficientNetB0, despite having certain advantages, lagged behind AlexNet in overall classification performance.

During the training process, data augmentation techniques were applied to enhance the generalization capability of the models and prevent overfitting. As part of data augmentation, the images were rotated by 90° increments (90° , 180° , 270°) to increase diversity. This approach was chosen to preserve the statistical structure of the bit sequences while allowing the model to evaluate them from different orientations, thereby improving rotational invariance. More complex augmentation methods (e.g., adding noise, scaling) were not applied, ensuring that the integrity of the bit sequences for randomness analysis was maintained.

In this study, three different CNN architectures were employed for deep learning-based randomness analysis: AlexNet, ResNet50, and EfficientNetB0. The hyperparameters used during model training are presented in Table 4. The learning rate was updated every 5 epochs to prevent the model from getting trapped in local minima, and early stopping was applied to mitigate overfitting. Additionally, 5-fold cross-validation was utilized to further enhance the generalization ability of the models. Throughout training, the network weights were updated using the backpropagation algorithm, ensuring the selection of optimal parameters.

Among the employed architectures, AlexNet is a widely used CNN model in deep learning, performing multi-layered convolution and pooling operations to extract features from input images. ResNet50, with its 50-layer structure

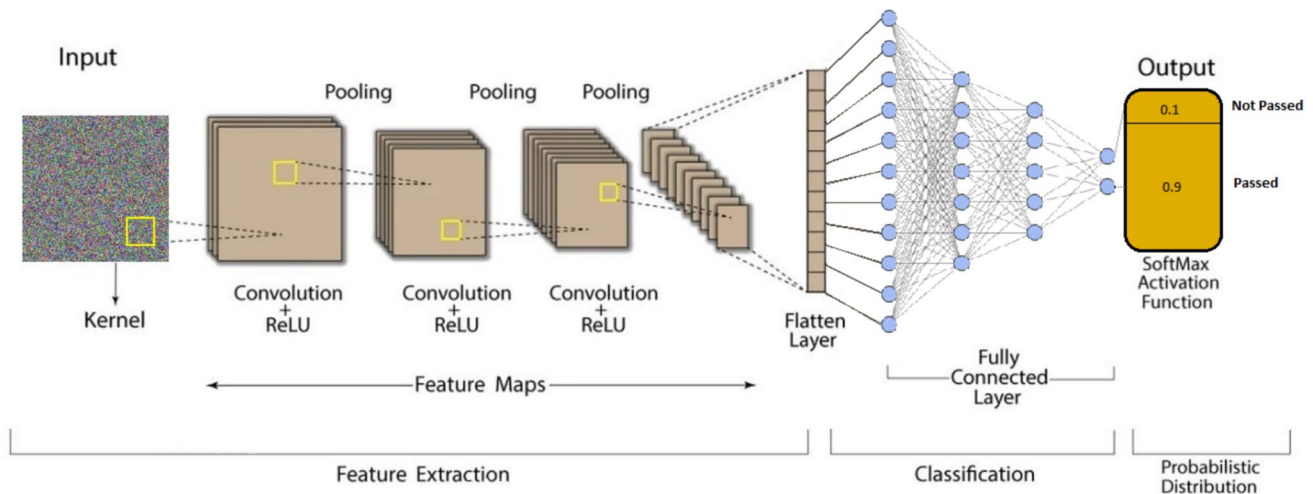


Fig. 8 Feature extraction and classification process of the proposed DL model for randomness analysis

Table 4 Hyperparameters used in training the model

Hyperparameter	Value
Learning Rate	0.001(Updated every 5 epochs)
Number of Epochs	100
Batch Size	16
Optimization Method	Adam
Verification Method	5 layered cross validation
Early Stopping	10 if there is no improvement throughout the epoch
Trainer Size	6, 400 image
Validation Set Size	1, 600 image

and residual connections, enables the effective training of deeper networks. EfficientNetB0, on the other hand, provides a more balanced architecture through a scaling strategy that optimizes parameter efficiency. Evaluating these three architectures together allows for a comparative analysis across CNN models with varying depths and parameter optimization strategies.

The learning rate specified in the table was updated every five epochs to prevent the model from getting trapped in local minima and to ensure a more stable learning process. Additionally, early stopping was applied to prevent overfitting. The Adam optimization algorithm was employed to enhance training efficiency and optimize parameter updates. The training and validation datasets were split in an 80 : 20 ratio to prevent model bias and improve generalization capability.

The careful selection of hyperparameters in the proposed DL model ensured a balanced performance of the model in metrics such as accuracy, precision, recall and F1 score. The

weights obtained during the training process were recorded and the generalization ability of the model was tested using the cross-validation method.

Processing of the bit-sequence datasets and the training of the deep learning model were carried out on a system equipped with an 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30 GHz processor. To enhance generalization and mitigate overfitting, a 5-fold cross-validation procedure was applied. During each training phase, the model was tested on different subsets to strengthen its generalization capability.

During the hyperparameter optimization process, the learning rate was initially set to 0.001 and updated every five epochs to ensure a more stable and efficient learning. The total training was planned as 100, but early stopping method was applied to avoid unnecessarily long training time. Data augmentation techniques were used to make the model more robust to different data variations. In this context, the images in the training data were diversified by rotating the images in multiples of 90° degrees (90°, 180°, 270°). The reason for choosing this method is to preserve the statistical structure of the bit sequences and contribute to a more generalizable learning of the model against random bit sequences generated from different angles.

3.3 Validation process

The validation process of the proposed DL model was carried out in order to evaluate the generalization capability of the model and its performance on real-world data. In the validation phase, the performance of the parameters learned during the training of the model on new and unprecedented data sets is analyzed. In this process, a 5-fold cross-validation

method was applied to avoid overfitting and to obtain the best overall performance.

The dataset is divided into 80% training and 20% validation to ensure a balanced evaluation of the model. With the cross-validation method, the model was trained and tested on different subsets at each iteration. This method prevented the model from memorizing specific data samples and ensured generalizable learning.

In the cross-validation process, the model was trained with the four subsets and validated with the one subset in each iteration. After the validation was completed for all subsets, the average accuracy, precision, recall and F1 score were calculated. By applying the early stopping method, the training process was terminated when there was no improvement in the verification loss during 10 epochs.

In order to increase the generalization capability of the model and to prevent false learning structure, the following techniques were applied:

- **Balanced Dataset:** In order to prevent the model from biased learning, it was ensured that the "passed" and "not passed" classes had an equal number of instances. Since the imbalance of the dataset may change the model's tendencies, balancing techniques were applied in the data preprocessing stage.
- **Dropout Technique:** To prevent the model from developing overdependence on certain neurons, some neurons were randomly disabled during training. This method allowed the model to explore different neuronal connections, which helped to increase its generalization capacity and prevent overlearning.
- **Data Augmentation:** Data augmentation was applied to increase the generalization capability of the model. However, only 90 degrees and multiples of degrees were returned as (90°, 180°, 270°) in order to avoid distorting the structure of the bit strings and causing mislabeling errors. This limitation was set as a strategy to ensure the transformation invariance of the model while maintaining the statistical integrity of the randomness analysis

As a result of this validation process, the generalization capability of the model was successfully tested, the tendency to overlearn was minimized and more consistent results were obtained. Cross-validation increased the reliability of the test results and improved the model's fit to real-world data.

4 Experimental results

This section provides a comprehensive evaluation of the proposed deep learning models based on their performance metrics. Key classification measures, including accuracy, precision, recall, and F1-score, were calculated to analyze the

overall performance of the models. In addition, the loss and accuracy curves during training and validation were examined, and the learning dynamics of the models were visualized to assess performance variations. The evaluation of the models is summarized in Table 5.

In this study, not only AlexNet but also ResNet50 and EfficientNetB0 architectures were comparatively evaluated. The experiments demonstrated that the AlexNet model achieved the highest performance, with 87% accuracy, 79% precision, 99% recall, and an 88% F1-score. In contrast, the ResNet50 model exhibited only near-random prediction performance with 50% accuracy and generated numerous false positives due to its low precision. EfficientNetB0 provided relatively balanced results with 56% accuracy and a 69% F1-score, yet it lagged behind AlexNet.

To further assess the effectiveness of the proposed deep learning-based approach, traditional ML methods were also tested. Support Vector Machine (SVM) and Random Forest (RF) algorithms were employed for comparison. Both methods reached an overall accuracy of 84%, but they demonstrated low precision values (76 – 77%) in correctly classifying bit sequences that failed the NIST tests (i.e., non-secure sequences), indicating vulnerability to critical classification errors. In contrast, the proposed AlexNet architecture reliably distinguished both random and non-random bit sequences, achieving over 87% accuracy, a more balanced precision–recall relationship, and higher F1-scores. These results highlight that the proposed approach not only delivers superior overall accuracy but also provides a more consistent and reliable assessment in terms of cryptographic security. Consequently, AlexNet emerges as the most suitable and effective architecture for the problem addressed in this study.

In DL models, training loss is an error metric that indicates how much the model's predictions on the training data set deviate from the true values. Validation loss is the error rate on a validation set that the model has not seen before and is critical for assessing the generalization ability of the model. The loss function is typically computed using functions such as Binary Cross-Entropy Loss (Log Loss) and is defined as shown in Eq. 7:

$$Loss = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)] \quad (7)$$

Table 5 Performance metrics of DL models

Model	Accuracy	Precision	Recall	F1 Score
AlexNet	0.87	0.79	0.99	0.88
ResNet50	0.50	0.50	1.00	0.67
EfficientNetB0	0.56	0.57	0.89	0.69

Here, y_i is the actual label, $\hat{y}_i$ denotes the predicted probability, N is the total number of samples. Training loss is computed by calculating this function on training data, while validation loss is calculated on validation data that the model has not seen before. If the validation loss increases continuously while the training loss decreases, the model may be overfitting. Conversely, if both loss values are high, the model may be underfitting. Similarly, training accuracy measures the model's correct prediction rate on the training set, and validation accuracy measures the model's performance on the validation dataset. Training and validation accuracy are calculated as shown in Eq. 8:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

The Confusion Matrix is a tool used to evaluate the performance of classification models, encompassing True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. Metrics derived from these components determine the model's key evaluation criteria, including accuracy, precision, recall, and F1 score. The mathematical definitions of accuracy and precision are given in Eqs. 8 and 9, representing the overall success rate of the model and the proportion of correctly predicted positive samples, respectively. The recall value, presented in Eq. 10, indicates how well the model identifies actual positive cases. The F1 score, which is the harmonic mean of precision and recall, is calculated according to Eq. 11, providing a comprehensive measure of the model's overall performance.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (10)$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

To evaluate the generalization capabilities of the models, classification performance metrics were calculated and summarized in Table 6. This table presents the True Positive Rate (TPR), False Positive Rate (FPR), and Area Under the Curve (AUC) values. Equations 12, 13, and 14 respectively illustrate how TPR, FPR, and AUC are computed.

$$\text{TPR} = \frac{TP}{TP + FN} \quad (12)$$

$$\text{FPR} = \frac{FP}{FP + TN} \quad (13)$$

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})d(\text{FPR}) \quad (14)$$

The results indicate that AlexNet demonstrated the most balanced and reliable performance, achieving a TPR of 0.99, FPR of 0.25, and an AUC of 0.87. The EfficientNetB0 model showed limited success with a TPR of 0.89 and an AUC of 0.56. In contrast, ResNet50 assigned all samples to the positive class theoretically, resulting in a TPR of 1.00 and FPR of 1.00, and consequently an AUC of 0.50, which corresponds to random guessing.

According to Table 6, AlexNet exhibited the strongest discriminative capability with an AUC of approximately 0.87, owing to its combination of low FPR and high TPR. EfficientNetB0 demonstrated a moderate discriminative power (AUC $\approx$ 0.56), whereas ResNet50's AUC of 0.50 indicates that this architecture performed classification largely at the level of random guessing.

In the study, not only statistical tests but also complexity measures were employed to analyze the structural characteristics of the generated sequences. Spectral entropy (SE), C0 complexity, and permutation entropy (PE) were calculated. The results show that the raw FPGA data had $SE = 13.52$, $C0 = 0.0017$, and $PE = 0.7705$, whereas PRNG sequences that failed the randomness tests exhibited lower complexity ($SE = 12.00$, $PE = 0.5977$). Post-processed FPGA sequences ($SE = 13.57$, $PE = 0.7737$) and PRNG sequences that passed the randomness tests ($SE = 13.57$, $PE = 0.7736$) both demonstrated similarly high entropy levels. These findings indicate that FPGA-based data possess strong characteristics in terms of complexity measures, comparable to systems based on chaotic maps reported in the literature (Gao et al. 2025b, 2025c). Thus, the proposed approach was validated not only through classical statistical tests but also via complexity analyses.

In an ideal model, validation accuracy should be close to training accuracy, and validation loss should remain low. The analysis of these metrics is crucial for early stopping, hyperparameter optimization, and enhancing generalization capability. Training/validation losses, accuracy curves, confusion matrix results, and prediction outputs for the three CNN architectures are collectively presented in Fig. 9. These visualizations provide a comprehensive comparison of each model's generalization capacity and classification performance.

Table 6 Classification performance of DL models

Model	TPR	FPR	AUC
AlexNet	0.99	0.25	0.87
ResNet50	1.00	1.00	0.50
EfficientNetB0	0.89	0.77	0.56

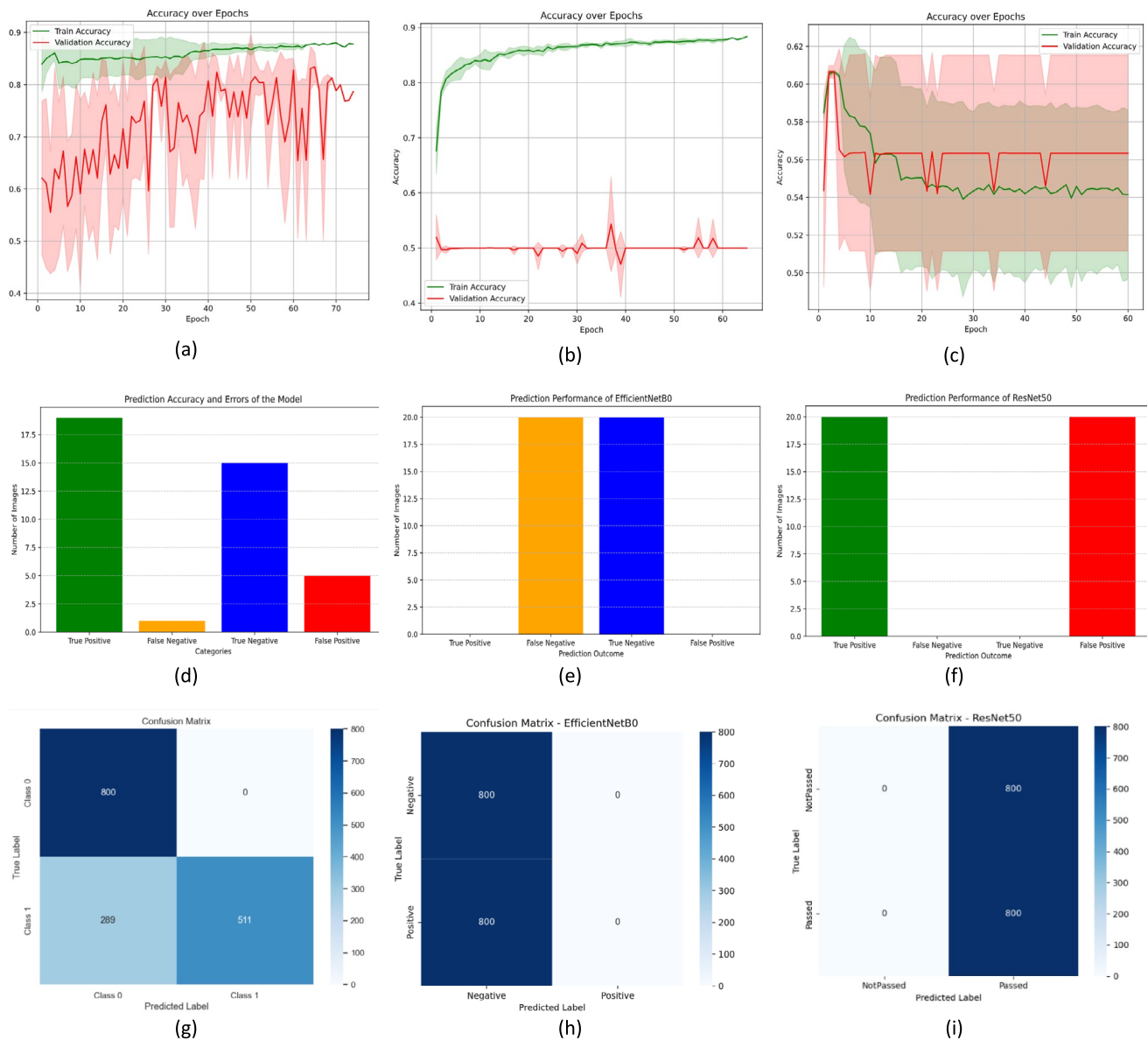


Fig. 9 DL model accuracy, confusion matrix and prediction results (a) AlexNet Accuracy, (b) EfficientNetB0 Accuracy (c) ResNet50 Accuracy, (d) AlexNet Prediction, (e) EfficientNetB0 Prediction, (f) ResNet50 Prediction, (g) AlexNet Confusion Matrix, (h) EfficientNet

Figure 9a–c illustrate the training and validation accuracy curves across epochs for AlexNet, EfficientNetB0, and ResNet50, respectively. As shown in Fig. 9a, the AlexNet model achieved a gradual increase in training accuracy, reaching approximately 0.90, while the validation accuracy, despite some fluctuations, stabilized around 0.80. This indicates that the model not only learned the training data but also generalized effectively to the validation set. The fivefold cross-validation results yielded an average accuracy of ≈ 0.87 and an F1 score of ≈ 0.88 .

In contrast, as presented in Fig. 9b, the EfficientNetB0 model's training accuracy quickly rose to $\approx 0.88 - 0.89$; however, its validation accuracy remained near chance

level (~ 0.50) throughout the training process. This marked divergence suggests that the model overfit the training data and failed to generalize to unseen data. The cross-validation mean accuracy was ≈ 0.566 with a standard deviation of ≈ 0.083 . Similarly, Fig. 9c shows that ResNet50 plateaued at a training accuracy of $\approx 0.54 - 0.58$, with validation accuracy around ≈ 0.55 . The wide uncertainty range indicates unstable generalization behavior on the validation set. Cross-validation mean accuracy was ≈ 0.563 with a standard deviation of ≈ 0.052 .

In summary, only AlexNet demonstrated meaningful and sustained improvement in validation accuracy, successfully capturing the discriminative structure of the dataset. Despite

their deeper and more complex architectures, EfficientNetB0 and ResNet50 overfit the limited dataset, resulting in lower validation performance.

Figure 9d–f display the prediction outputs over epochs for the three models. AlexNet outputs in Fig. 9d show a clear separation between classes and well-defined decision boundaries, demonstrating the model's capability to correctly capture discriminative patterns in the dataset. In contrast, EfficientNetB0 predictions in Fig. 9e exhibit high fluctuations and class ambiguities during validation, indicating limited generalization ability. ResNet50 predictions in Fig. 9f are similarly unstable, with the model failing to meaningfully separate classes and showing high uncertainty at decision boundaries.

Figure 9g–i present the confusion matrices corresponding to each model. The AlexNet confusion matrix in Fig. 9g shows high true positive (TP) and true negative (TN) rates with very few misclassifications, confirming the model's ability to accurately distinguish both sequences that satisfy and fail the randomness criteria. EfficientNetB0, shown in Fig. 9h, exhibits a high false positive (FP) rate particularly for the negative class, consistent with its near-chance validation performance. ResNet50, as illustrated in Fig. 9i, demonstrates imbalanced TP and TN rates, indicating a failure to meaningfully separate classes.

Overall, evaluation of both confusion matrices and prediction outputs indicates that only AlexNet provides consistent, balanced, and reliable results, whereas

EfficientNetB0 and ResNet50 fail to maintain robust classification accuracy due to poor generalization on the validation set.

The ablation analysis presented in Fig. 10 was conducted to assess the contributions of key components affecting the performance of the AlexNet model, which achieved the highest accuracy in this study. In this context, the individual effects of rotational data augmentation, early stopping, and 5-fold cross-validation were analyzed. The results indicate that removing data augmentation increased the model's average validation accuracy to 94.72%, suggesting that this component effectively limits overfitting. However, the high standard deviation of 5.52% implies reduced stability in the absence of augmentation. Therefore, data augmentation contributes to long-term generalizability of the model.

When the early stopping mechanism was removed, the validation accuracy increased only marginally to 88.16%, and this change was not statistically significant. This finding highlights that early stopping primarily prevents unnecessarily prolonged training and ensures efficient use of training time.

Finally, omitting cross-validation resulted in 100% accuracy, suggesting potential overlap between training and test data or insufficient test data for proper generalization. This outcome demonstrates that without cross-validation, the model becomes prone to overfitting, and the obtained high accuracy may not generalize to real-world scenarios.

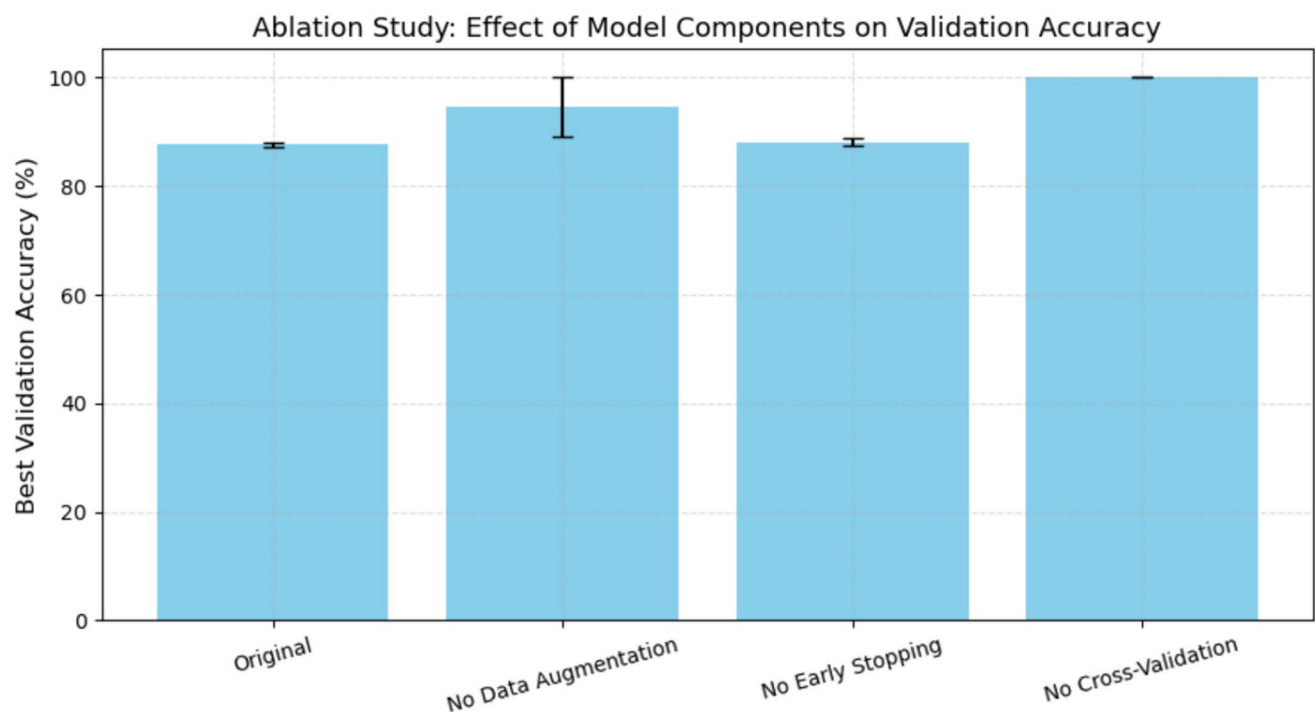


Fig. 10 AlexNet ablation analysis

Overall, the ablation analysis reveals that each component affects both the performance and stability of the model. This indicates that all architectural elements employed in the proposed approach collectively contribute to a balanced and generalizable learning process.

In line with these findings, the experimental results show distinct performance differences among the deep learning architectures used. The AlexNet model achieved the highest overall performance, with 87% accuracy, 79% precision, 99% recall, and 88% F1 score, demonstrating strong sensitivity by correctly identifying nearly all positive instances. ResNet50, on the other hand, exhibited only random-guess-level performance with 50% accuracy; despite a recall of 100%, its low precision resulted in a high number of false positives. EfficientNetB0 provided a more balanced outcome with 56% accuracy and 69% F1 score, but still lagged behind AlexNet. Overall, these analyses confirm AlexNet's superior performance compared to ResNet50 and EfficientNetB0.

4.1 Statistical reliability analysis of model performances

In this study, bootstrap analysis was applied to evaluate the performance reliability of the models. Bootstrap is a method

that allows examining statistical reliability by repeatedly generating subsamples from the original dataset using sampling with replacement. In this analysis, 1000 iterations were performed, with 80% of the test set included in each bootstrap sample. For each bootstrap iteration, the relevant metrics were calculated to obtain their distributions.

The primary purpose of the bootstrap method is to estimate the distribution of a performance measure θ based on samples derived from the original dataset. Accordingly, the metrics for the bootstrap samples created from n observations, $(D_1^*, D_2^*, \dots, D_B^*)$ are repeatedly computed as defined in Eq. 15:

$$\hat{\theta}_b^* = f(D_b^*), b = 1, 2, \dots, B \quad (15)$$

Here, $f(\cdot)$ denotes the selected metric, and D_b^* denotes the selected metric, and b -th. bootstrap sample. The resulting set $\{\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_B^*\}$ forms the empirical distribution of the performance measure. From this distribution, the mean, standard deviation, and confidence intervals can be calculated to assess the statistical reliability of the model.

Figure 11 presents the bootstrap distributions of F1 scores for the three different deep learning architectures. The plot shows that the F1 scores of the AlexNet model are

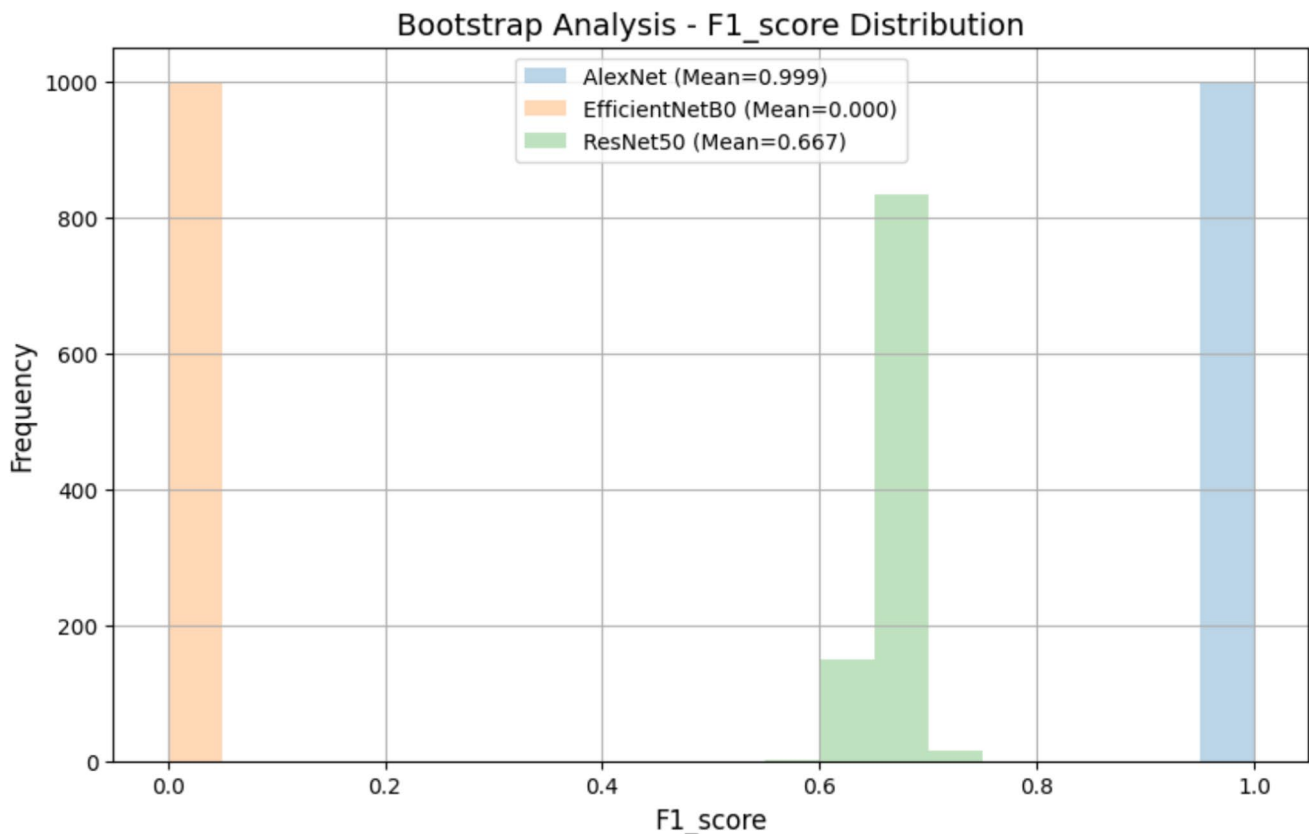


Fig. 11 F1 score bootstrap analysis of the deep learning models used

almost entirely close to 1, indicating very high accuracy and consistency. In contrast, the F1 scores for the EfficientNetB0 model are clustered around 0, demonstrating that this model failed to perform effective classification on the given dataset. The ResNet50 model exhibits F1 scores distributed approximately between 0.6 and 0.7, indicating a moderate level of performance. These results highlight that bootstrap analysis allows for a more comprehensive understanding of model sensitivity to sampling variability, beyond reporting a single accuracy or F1 score. In particular, the high and stable performance of AlexNet, the moderate but variable performance of ResNet50, and the low performance of EfficientNetB0 are clearly observed through the bootstrap distributions.

5 Results and conclusions

In this study, a deep learning–based randomness evaluation method was proposed to overcome the limitations of conventional statistical test suites, particularly those of NIST SP 800–22. Bit sequences obtained from FPGA-based RO-TRNG and various PRNG sources were selected with a length of 1,204,224 bits and transformed into 224×224 RGB images for analysis using CNN-based models. Comprehensive experiments conducted on AlexNet, ResNet50, and EfficientNetB0 architectures showed that the AlexNet model achieved the highest overall performance, with 87% accuracy, 79% precision, 99% recall, and 88% F1 score. Despite its deep structure, ResNet50 reached only 50% accuracy, performing close to random guessing; although its recall was high, its low precision resulted in a substantial number of false positives. EfficientNetB0 demonstrated a more balanced performance with 56% accuracy and 69% F1 score, yet it remained inferior to AlexNet.

The experimental findings indicate that the proposed AlexNet-based approach not only provides the statistical validation offered by conventional tests such as NIST SP 800–22 but also captures structural dependencies and unpredictable patterns within datasets. Consequently, the model simultaneously delivers high accuracy, low error rates, and short processing times—features critical for modern cryptographic systems.

The findings obtained in this study also provide a significant contribution in the context of cryptographic security. While traditional statistical tests are effective in evaluating the basic statistical properties of random number generators, they may fall short in capturing complex dependency structures and long-term patterns that are critical against cryptographic attacks. The proposed deep learning–based approach, however, is capable of detecting such intricate structures, thereby revealing not only the level of randomness but also the security value

of the generated sequences for cryptographic protocols. This capability enhances the practical applicability of the method in high-security applications such as key generation, session management, and digital signatures, thus directly contributing to the overall security framework of systems. In conclusion, the proposed deep learning–based randomness evaluation method presents an innovative solution that addresses the shortcomings of traditional statistical tests. The approach optimizes processing time for large-scale datasets, enabling more comprehensive, reliable, and scalable analyses. The results suggest that AI-assisted testing methods can serve as a powerful alternative in cryptographic security processes and may contribute to the development of new standards in the future. As a future projection, the proposed methodology can also be extended to include evaluations under cryptanalysis-oriented scenarios such as adversarial robustness, side-channel analysis, and known-plaintext attacks, and these aspects are planned to be comprehensively investigated in subsequent studies.

Acknowledgements The authors sincerely thank the editor(s), and the anonymous reviewers contributed positively to this paper's academic development with their constructive and quick feedback.

Author contributions Seyfullah Kaner: Investigation, Methodology, Software. Ali Murat. Garipcan: Conceptualization, Methodology, Software, Writing – original draft. Ebubekir Erdem: Validation, Supervision, Writing – review & editing.

Data availability Data are available from the corresponding author by reasonable request.

Declarations

Competing interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

References

- Akcengiz Z, Aslan M, Doğanaksoy A, Sulak F, Uğuz M (2024) LS-14 test suite for long sequences. *Hacettepe J Math Stat* 53:230–250. <https://doi.org/10.15672/hujms.1190807>
- Alani MM (2019) Applications of machine learning in cryptography. In: *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*. pp. 23–27. ACM, New York, NY, USA. <https://doi.org/10.1145/3309074.3309092>
- Ali RS, Akif OZ, Jassim SA, Farhan AK, El-Kenawy ESM, Ibrahim A, Ghoneim ME, Abdelhamid AA (2022) Enhancement of the CAST block algorithm based on novel S-box for image encryption. *Sensors* 22:8527. <https://doi.org/10.3390/S22218527>
- Aljohani M, Ahmad I, Basher M, Allassafi MO (2019) Performance analysis of cryptographic pseudorandom number generators. *IEEE Access* 7:39794–39805. <https://doi.org/10.1109/ACCESS.2019.2907079>
- Applebaum B (2016) Cryptographic hardness of random local functions: survey. *Comput Complex* 25:667–722. <https://doi.org/10.1007/S00037-015-0121-8>
- Bassham LE, Rukhin AL, Soto J, Nechvatal JR, Smid ME, Barker EB, Leigh SD, Levenson M, Vangel M, Banks DL, Heckert NA, Dray JF, Vo S (2010) A statistical test suite for random and pseudorandom number generators for cryptographic applications. Gaithersburg, MD. <https://doi.org/10.6028/NIST.SP.800-22r1a>
- Brown RG (2006) Dieharder: a Gnu public license random number tester. Manual, Duke University Physics Department. Available at: <https://rurban.github.io/dieharder/manual/dieharder.pdf>
- Caddy T (2011) FIPS 140–2. *Encyclopedia of Cryptography and Security*. Springer US, Boston, MA, pp 468–471. https://doi.org/10.1007/978-1-4419-5906-5_205
- Capuano N, Fenza G, Loia V, Stanzione C (2022) Explainable artificial intelligence in cybersecurity: a survey. *IEEE Access* 10:93575–93600. <https://doi.org/10.1109/ACCESS.2022.3204171>
- Choithani T, Chowdhury A, Patel S, Patel P, Patel D, Shah M (2024) A comprehensive study of artificial intelligence and cybersecurity on Bitcoin, crypto currency and banking system. *Ann Data Sci* 11:103–135. <https://doi.org/10.1007/s40745-022-00433-5>
- Coutinho M, de Albuquerque RO, Borges F, Villalba LJG, Kim TH (2018) Learning perfectly secure cryptography to protect communications with adversarial neural cryptography. *Sensors (Switzerland)* 18:1–16. <https://doi.org/10.3390/s18051306>
- Das I, Singh S, Gupta S, Banerjee A, Mohiuddin MG, Tiwary S (2019) Design and implementation of secure ATM system using machine learning and crypto–stego methodology. *SN Appl Sci* 1:976. <https://doi.org/10.1007/s42452-019-0988-0>
- Deebak BD, Hwang SO (2025) Privacy-preserving learning model using lightweight encryption for visual sensing industrial IoT devices. *IEEE Trans Emerg Top Comput Intell*. <https://doi.org/10.1109/TETCI.2024.3523771>
- Feng Y, Hao L (2020) Testing randomness using artificial neural network. *IEEE Access* 8:163685–163693. <https://doi.org/10.1109/ACCESS.2020.3022098>
- Fischer T (2018) Testing cryptographically secure pseudo random number generators with artificial neural networks. *Proc. - 17th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 12th IEEE Int Conf Big Data Sci Eng Trust* 2018. 1214–1223. <https://doi.org/10.1109/TRUSTCOM/BIGDATA.2018.00168>
- Gao S, Liu J, Iu HHC, Erkan U, Zhou S, Wu R, Tang X (2024) Development of a video encryption algorithm for critical areas using 2D extended Schaffer function map and neural networks. *Appl Math Model* 134:520–537. <https://doi.org/10.1016/j.apm.2024.06.016>
- Gao S, Zhang Z, Iu HHC, Ding S, Mou J, Erkan U, Toktas A, Li Q, Wang C, Cao Y (2025c) A parallel color image encryption algorithm based on a 2-D logistic-rulkov neuron map. *IEEE Internet Things J* 12:18115–18124. <https://doi.org/10.1109/IJOT.2025.3540097>
- Gao S, Ding S, Iu HHC, Erkan U, Toktas A, Simsek C, Wu R, Xu X, Cao Y, Mou J (2025) A three-dimensional memristor-based hyperchaotic map for pseudorandom number generation and multi-image encryption. *Chaos*. <https://doi.org/10.1063/5.0270220>
- Gao S, Iu HHC, Erkan U, Simsek C, Toktas A, Cao Y, Wu R, Mou J, Li Q, Wang C (2025) A 3D Memristive Cubic Map with Dual Discrete Memristors: Design, Implementation, and Application in Image Encryption. *IEEE Trans. Circuits Syst. Video Technol.* 1–1. <https://doi.org/10.1109/TCSVT.2025.3545868>
- Garipcan AM, Erdem E (2020) A TRNG using chaotic entropy pool as a post-processing technique: analysis, design and FPGA implementation. *Analog Integr Circ Sig Process* 103:391–410. <https://doi.org/10.1007/S10470-020-01605-0>
- Garipcan AM, Erdem E (2022) DESSB-TRNG: a novel true random number generator using data encryption standard substitution box as post-processing. *Digit Signal Process*. <https://doi.org/10.1016/j.DSP.2022.103455>
- Gentle JE (2003) Random number generation and monte carlo methods, 2nd edn. Springer, New York, NY. <https://doi.org/10.1007/b97336>
- Giudici P (2018) Fintech risk management: a research challenge for artificial intelligence in finance. *Front Artif Intell* 1:1–6. <https://doi.org/10.3389/frai.2018.00001>
- Gu R, Wang P, Zheng M, Hu H, Yu N (2020) Adversarial attack based countermeasures against deep learning side-channel attacks. *J Univ Sci Technol China* 50:1343–1358. <https://doi.org/10.3969/j.issn.0253-2778.2020.10.006>
- Hamza R, Yan Z, Muhammad K, Bellavista P, Titouna F (2020) A privacy-preserving cryptosystem for IoT e-healthcare. *Inf Sci (Ny)* 527:493–510. <https://doi.org/10.1016/j.ins.2019.01.070>
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2016–Decem, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. *Neural Comput* 18:1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>
- Idris MF, Teh JS, Yan JLS, Yeoh W-Z (2021) A deep learning approach for active S-box prediction of lightweight generalized feistel block ciphers. *IEEE Access* 9:104205–104216. <https://doi.org/10.1109/ACCESS.2021.3099802>
- Irfan M, Ali A, Khan MA, Ehatisham-ul-haq M, Shah SNM, Saboor A, Ahmad W (2020) Pseudorandom number generator (PRNG) design using hyper-chaotic modified robust logistic map (HC-MRLM). *Electronics* 9(1):104. <https://doi.org/10.3390/ELECTRONICS9010104>
- Jacak MM, Jóźwiak P, Niemczuk J, Jacak JE (2021) Quantum generators of random numbers. *Sci. Reports* 2021 111. 11, 1–21. <https://doi.org/10.1038/s41598-021-95388-7>
- Khalid N, Qayyum A, Bilal M, Al-Fuqaha A, Qadir J (2023) Privacy-preserving artificial intelligence in healthcare: techniques and applications. *Comput Biol Med* 158:106848. <https://doi.org/10.1016/j.combiomed.2023.106848>
- Kim H, Kim D, Yi O (2022) Convolution neural network-based sensitive security parameter identification and analysis. *Wirel Commun Mob Comput* 2022:9584894. <https://doi.org/10.1155/2022/9584894>
- Kishore RH, Sekhar AC, Patro P, Chaganti P (2023) A novel homomorphic and matrix operation for randomization encryption schemes for privacy in cloud computing architecture. *J Theor Appl Inf Technol* 101:1038–1053
- Knuth DE (1997) The art of computer programming: fundamental algorithms, volume 2. Addison-Wesley Professional, Boston, MA

- Koç ÇK (2009) About Cryptographic Engineering. In: Koç ÇK (ed) Cryptographic Engineering. Springer US, Boston, MA, pp 1–4. https://doi.org/10.1007/978-0-387-71817-0_1
- Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. *Commun ACM* 60:84–90. <https://doi.org/10.1145/3065386>
- Kuznetsov O, Zakharov D, Frontoni E (2024) Deep learning-based biometric cryptographic key generation with post-quantum security. *Multimed Tools Appl* 83:56909–56938. <https://doi.org/10.1007/S11042-023-17714-7/TABLES/6>
- L'Ecuyer P, Simard R (2007) TestU01. *ACM Trans Math Softw* 33:1–40. <https://doi.org/10.1145/1268776.1268777>
- Li C, Zhang J, Sang L, Gong L, Wang L, Wang A, Wang Y (2020) Deep learning-based security verification for a random number generator using white chaos. *Entropy* 22:1134. <https://doi.org/10.3390/e22101134>
- Li Q, Ma B, Wang X, Wang C, Gao S (2024) Image steganography in color conversion. *IEEE Trans Circuits Syst II Express Briefs* 71:106–110. <https://doi.org/10.1109/TCSII.2023.3300330>
- Lingfeng Q, Fan C, Shanjun Z, He H (2022) Cryptanalysis of reversible data hiding in encrypted images by block permutation and co-modulation. *IEEE Trans Multimed* 24:2924–2937. <https://doi.org/10.1109/TMM.2021.3090588>
- Liu L, Gao M, Zhang Y, Wang Y (2022) Application of machine learning in intelligent encryption for digital information of real-time image text under big data. *EURASIP J Wirel Commun Netw* 2022(1):1–16. <https://doi.org/10.1186/S13638-022-02111-9>
- Mannalatha V, Mishra S, Pathak A (2023) A comprehensive review of quantum random number generators: concepts, classification and the origin of randomness. *Quantum Inf Process*. 2023 22:12. 22, 1–45. <https://doi.org/10.1007/S11128-023-04175-Y>
- Manucom EMM, Gerardo BD, Medina RP (2019) Analysis of Key Randomness in Improved One-Time Pad Cryptography. *Proc. Int. Conf. Anti-Counterfeiting, Secur. Identification, ASID*. 2019-October, 11–16. <https://doi.org/10.1109/ICASID.2019.8925173>
- Marsaglia G (2003) Random number generators. *J Mod Appl Stat Methods* 2:2–13. <https://doi.org/10.22237/jmasm/1051747320>
- Marwala T, Xing B (2018) Blockchain and Artificial Intelligence 6. <https://doi.org/10.48550/arXiv.1802.04451>
- Mehmood A, Shafique A, Alawida M, Khan AN (2024) Advances and vulnerabilities in modern cryptographic techniques: a comprehensive survey on cybersecurity in the domain of machine/deep learning and quantum techniques. *IEEE Access* 12:27530–27555. <https://doi.org/10.1109/ACCESS.2024.3367232>
- Mengdi Z, Xiaojuan Z, Yayun Z, Siwei M (2021) Overview of randomness test on cryptographic algorithms. *J Phys Conf Ser* 1861:012009. <https://doi.org/10.1088/1742-6596/1861/1/012009>
- Muhammad AS, Özkaynak F (2021) Siea: Secure image encryption algorithm based on chaotic systems optimization algorithms and pufs. *Symmetry*. <https://doi.org/10.3390/SYM13050824>
- Nagy I, Suciu A (2021) Randomness testing with neural networks. In: 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE. pp 431–436. <https://doi.org/10.1109/ICCP53602.2021.9733481>
- Nitaj A, Rachidi T (2023) Applications of neural network-based AI in cryptography. *Cryptography* 7:39. <https://doi.org/10.3390/cryptography7030039>
- Oseni A, Moustafa N, Janicke H, Liu P, Tari Z, Vasilakos A (2021) Security and privacy for artificial intelligence: opportunities and challenges. *J ACM* 37:35. <https://doi.org/10.1145/1122445.1122456>
- Özkaynak F (2014) Cryptographically secure random number generator with chaotic additional input. *Nonlinear Dyn* 78:2015–2020. <https://doi.org/10.1007/S11071-014-1591-Y/TABLES/1>
- Perin G, Chmielewski Ł, Batina L, Picek S (2021) Keep it unsupervised: horizontal attacks meet deep learning. *IACR Trans Cryptogr Hardw Embed Syst* 2021:343–372. <https://doi.org/10.46586/TCHES.V2021.I1.343-372>
- Priya S, Abirami SP, Arunkumar B, Mishachandar B (2024) Super-resolution deep neural network (SRDNN) based multi-image steganography for highly secured lossless image transmission. *Sci Rep* 14:1–13. <https://doi.org/10.1038/s41598-024-54839-7>
- Qu L, Li M, Sun Y, Su S, Liu Y, Zhang L (2024) Security analysis of a reversible data hiding scheme in encrypted images by redundant space transfer. *J King Saud Univ* 36:101914. <https://doi.org/10.1016/j.jksuci.2024.101914>
- Rana M, Mamun Q, Islam R (2022) Lightweight cryptography in IoT networks: a survey. *Futur Gener Comput Syst* 129:77–89. <https://doi.org/10.1016/J.FUTURE.2021.11.011>
- Salem M, Taheri S, Yuan J-S (2018) Utilizing transfer learning and homomorphic encryption in a privacy preserving and secure biometric recognition system. *Computers* 8:3. <https://doi.org/10.3390/computers8010003>
- Schindler W (2023) Overview of AIS 20/31. Presentation at NIST Workshop on Random Number Generation, Gaithersburg, MD. Available at: <https://csrc.nist.gov/csrc/media/Presentations/2023/overview-of-ais-2031/images-media/session-2-schindleroverview-of-ais-20-31.pdf>
- Sharma K, Aggarwal A, Singhania T, Gupta D, Khanna A (2019) Hiding data in images using cryptography and deep neural network. *J Artif Intell Syst* 1:143–162. <https://doi.org/10.33969/AIS.2019.11009>
- Simion E (2020) Entropy and randomness: from analogic to quantum world. *IEEE Access* 8:74553–74561. <https://doi.org/10.1109/ACCESS.2020.2988658>
- Sunar B, Martin W, Stinson D (2007) A provably secure true random number generator with built-in tolerance to active attacks. *IEEE Trans Comput* 56:109–119. <https://doi.org/10.1109/TC.2007.250627>
- Sýs M, Říha Z (2014) Faster randomness testing with the NIST Statistical Test Suite. In: *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 272–284. https://doi.org/10.1007/978-3-319-12060-7_18
- Tan M, Le, QV (2019) EfficientNet: Rethinking model scaling for convolutional neural networks. 36th Int. Conf. Mach. Learn. ICML 2019. 2019-June, 10691–10700
- Tan S, Knott B, Tian Y, Wu DJ (2021) CryptGPU: Fast Privacy-Preserving Machine Learning on the GPU. In: 2021 IEEE Symposium on Security and Privacy (SP). pp. 1021–1038. IEEE. <https://doi.org/10.1109/SP40001.2021.00098>
- Teh JS, Alawida M, Sii YC (2020) Implementation and practical problems of chaos-based cryptography revisited. *J Inf Secur Appl* 50:102421. <https://doi.org/10.1016/J.JISA.2019.102421>
- Tolba Z (2024) Cryptanalysis and improvement of multimodal data encryption by machine-learning-based system. *arXiv preprint arXiv:2402.15779*. <https://doi.org/10.48550/arXiv.2402.15779>
- Torkzadehmahani R, Nasirigerdeh R, Blumenthal DB, Kacprowski T, List M, Matschinske J, Spaeth J, Wenke NK, Baumbach J (2022) Privacy-preserving artificial intelligence techniques in biomedicine. *Methods Inf Med* 61:e12–e27. <https://doi.org/10.1055/s-0041-1740630>
- Truong ND, Haw JY, Assad SM, Lam PK, Kavehei O (2018) Machine learning cryptanalysis of a quantum random number generator. *IEEE Trans Inf Forensics Secur* 14:403–414. <https://doi.org/10.1109/TIFS.2018.2850770>
- Turan MS, Barker E, Kelsey J, McKay KA, Baish ML, Boyle M (2018) Recommendation for the entropy sources used for random bit generation. , Gaithersburg, MD. <https://doi.org/10.6028/NIST.SP.800-90B>

- Walker J (2008) ENT: a pseudorandom number sequence test program. Software and documentation, Fourmilab. available at www.fourmilab.ch/random/S
- Wu Y, Noonan JP, Agaian S (2011) NPCR and UACI randomness tests for image encryption. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)* 1(2):31–38. Available at: <https://www.cyberjournals.com/Papers/Apr2011/05.pdf>
- Zaid G, Bossuet L, Habrard A, Venelli A (2020) Methodology for efficient CNN architectures in profiling attacks. *IACR Trans Cryptogr Hardw Embed Syst* 202(1):1–36. <https://doi.org/10.13154/TCHES.V2020.II.1-36>
- Zhao Y, Fan S (2019) Analysis of cryptosystem recognition scheme based on Euclidean distance feature extraction in three machine learning classifiers. *J Phys Conf Ser* 1314:012184. <https://doi.org/10.1088/1742-6596/1314/1/012184>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.