SJER Scientific Journal of Engineering Research

futuristech PT. Teknologi Futuristik Indonesia

Received: April 7, 2026; Revised: June 10, 2026; Accepted: June 16, 2026; Published: June 19, 2026; Current version: June 19, 2026. e-ISSN: 3109-1725
Digital Object Identifier 10.64539/sjer.v2i4.2026.491

Article

Neural Differential Cryptanalysis of GIFT-128 and ASCON via Deep Learning

Muhammad Ahmad[1,*], Hua Zhou[1], Muhammad Usman[2], Tanzeela Bibi[1], Haider Ali[3], Maryum Shahzadi[4], Farah Javed[5]

[1] Department of Electronics and Information Engineering, School of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, 210044, China; e-mail: muhammadah315@gmail.com (M. Ahmad), hzhou@nuist.edu.cn (H. Zhou), tanzeelabibi26@gmail.com (T. Bibi).
[2] Department of Computer Application Technology, School of Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China; e-mail: usman7610@outlook.com (M. Usman).
[3] Department of Hydrology and Water Resources, School of Hydrology and Water Resources, Nanjing University of Information Science and Technology, Nanjing, 210044, China; e-mail: haideraliises@yahoo.com (H. Ali).
[4] Department of Statistics and Mathematics, School of Statistics, Government College University (GCU), Lahore 54000, Pakistan; e-mail: sostatsfri@gmail.com (M. Shahzadi).
[5] Department of Human Nutrition and Food Technology, Superior University, Raiwind Road, Lahore 54000, Pakistan; e-mail: farahjaved@superior.edu.pk (F. Javed).
* Correspondence Author

The authors received no financial support for the research, authorship, and/or publication of this article.

Abstract: Differential analysis is a pivotal method for assessing the security of block ciphers; it distinguishes a cipher from a random permutation by tracing the propagation of plaintext differences. Traditional analytical methods face limitations when applied to complex algorithms, whereas the feature extraction capabilities of deep learning have opened up new avenues for cryptanalysis. To facilitate the security assessment of block ciphers, this paper proposes a novel construction method for a neural differential distinguisher that integrates traditional differential analysis with deep learning techniques. Regarding dataset construction, a multi-ciphertext-pair triplet input format is adopted to preserve differential features while capturing correlations across ciphertext pairs. The network architecture is based on Convolutional Neural Networks (CNNs) and incorporates a Residual Shrinkage Network to construct a deep dilated structure and a multi-scale feature fusion mechanism. Experimental results on the GIFT-128 and ASCON-PERMUTATION lightweight permutation-based cryptographic algorithm demonstrate the efficacy of this approach: for GIFT-128, the 6-round distinguisher reached a maximum accuracy of 99.70%, and the 7-round distinguisher reached 95.47% when using 32 ciphertext pairs; for the 4-round analysis of ASCON, the accuracy rate reached a maximum of 53.54%. These results validate the effectiveness of deep learning methods in the analysis of cryptographic security.

Keywords: Deep Learning; Differential Analysis; Block Cipher; Neural Distinguisher; GIFT-128; ASCON.

1. Introduction

Differential analysis [1] is a crucial methodology for assessing the security of block cipher algorithms. Its core principle involves identifying potential vulnerabilities within an algorithm by analyzing the propagation and evolution of plaintext differences (specifically, the XOR result of two plaintexts) as they traverse the encryption process. Building upon this analytical framework, various extensions have been developed including higher-order differential analysis [2], impossible differential analysis [3], and related-key differential analysis yielding numerous significant theoretical breakthroughs in the realm of algorithm analysis [4].

Deep learning extracts features from data through hierarchical information processing to facilitate decision-making. In 2019, Gohr [5] presented a seminal work at the

CRYPTO conference, marking the first successful integration of deep learning with traditional differential cryptanalysis. Specifically, Gohr designed a Deep Residual Network (ResNet) tailored for the SPECK32/64 cipher, constructing an 8-round neural differential distinguisher that demonstrated superior accuracy compared to traditional methods. This achievement validated the utility of applying deep learning to differential cryptanalysis and opened up new avenues for research within this domain. Much like traditional differential analysis, deep learning-based differential analysis places its primary focus on the construction of the distinguisher; the higher the accuracy of the distinguisher and the greater the number of rounds it cover the lower the computational complexity required to recover the encryption key, thereby yielding more effective results in algorithm analysis.

In 2021, Chen and Yu [6] significantly enhanced the accuracy of 5-to-7-round neural differential distinguishers for the SPECK32/64 algorithm by incorporating multiple pairs of ciphertexts as training samples for their neural network. The following year, Hou *et al*. [7] further optimized the accuracy of neural differential distinguishers for both the SPECK and SIMON algorithms by integrating multi-output differentials into their training datasets. In 2023, Lu *et al*. [8] successfully constructed distinguishers for the SIMON cipher applicable under both single-key and related-key scenarios by leveraging inferred information derived from the initial two rounds of encryption neural differential distinguisher, achieving a significant improvement in accuracy. In the same year, Bao *et al*. [9] effectively enhanced the accuracy of the distinguisher for the SIMON32/64 algorithm by configuring the neural network inputs as a linear combination of the outputs from the previous round. In 2024, Shen *et al*. [10] proposed a novel model that significantly improved the accuracy of neural distinguishers targeting the GIFT and ASCON algorithms. Inspired by the GoogleNet architecture, Zhang *et al*. [11] further bolstered the performance of the SPECK algorithm's distinguisher through multi-scale feature fusion, thereby highlighting the efficacy of this structural design in cryptanalysis. Based on differential analysis and sample feature analysis, Wang *et al*. [12] effectively boosted the accuracy of network distinguishers for the SPECK and SIMON algorithms by generating negative samples using fixed differentials rather than random ones. Furthermore, Seok and Lee [13] leveraged machine learning techniques to optimize the selection of input differentials, thereby further enhancing the accuracy of the SIMON and SPECK algorithms.

Although the aforementioned studies have achieved significant progress regarding specific algorithms, their methodologies still suffer from limitations: they are difficult to transfer to novel cryptographic algorithms, and their feature extraction mechanisms fail to fully exploit the inherent correlations among ciphertexts. To address these limitations, this paper integrates traditional differential analysis with deep learning to propose a novel construction framework for neural differential distinguishers. We investigate the input structures of these distinguishers and develop adaptive dataset construction methods and specialized network architectures tailored to two distinct cryptographic algorithm structures. The proposed framework is designed to preserve differential characteristics while enabling the extraction of correlations across multiple ciphertext pairs. While S-box design is important for cipher strength [14]-[17], our work focuses on distinguisher construction and does not propose new S-boxes.

Section 2 of this paper provides an over-view of the GIFT-128 and ASCON-PERMUTATION algorithms, as well as relevant foundational concepts; Section 3 proposes a construction method for neural differential distinguishers; Section 4 and 5 apply the aforementioned method to construct neural differential distinguishers for the GIFT-128 and ASCON-PERMUTATION algorithms, respectively; finally, the paper concludes with a summary and an outlook on future research.

2. Prerequisites

2.1. GIFT-128 Algorithm

GIFT-128, designed by Banik *et al*. [18], is a lightweight block cipher based on a Substitution-Permutation Network (SPN) structure; it is widely applied in scenarios such as Internet of Things (IoT) device security and data protection in embedded systems. This algorithm features a 128-bit block size and a 128-bit key size, performing a total of 40 rounds of encryption operations. The round function con-sists of three ordered operations: SubCells, PermBits, and AddRoundKey.

- SubCells: Implements a nonlinear substitution on the state via an invertible 4×4 S-box.
- PermBits: Executes distinct bit permutations on the 32-bit state word.
- AddRoundKey: Performs a bitwise XOR operation between the round key and the state, where the round key con-sists of two 32-bit segments ($U \| V$) extracted from the key state.

2.2. ASCON-Permutation: A Lightweight Permutation-Based Cryptographic Algorithm

Ascon is a permutation-based lightweight authenticated encryption algorithm designed by Dobraunig *et al*. [19]; it employs the Sponge Construction to realize a unified framework for data encryption and integrity protection. As the winning algorithm of the CAESAR competition and a finalist candidate in the NIST Lightweight Cryptography Standardization project, its design is widely recognized as a benchmark for balancing security and efficiency in modern lightweight cryptography. The state

size of Ascon's operation is 320 bits (composed of five 64-bit words: $x0, \ldots, x4$), and the state register is $S = x0 \,||\, x1 \,||\, x2 \,||\, x3 \,||x4$. its primary component is a 320-bit permutation, implemented through the use of specific constants and a defined number of rounds.

Consistent with [10], this paper focuses exclusively on the 320-bit permutation. Each round of the permutation p comprises three steps: the Constant Addition layer (pC), the Substitution layer (pS), and Linear Diffusion layer (pL).

- Constant Addition (pC): In each round, a round constant c_r is added to the state register word x2; the specific constant value varies depending on the current round number.
- Substitution Layer (pS): The substitution layer pS updates the state by applying a 5-bit S-box in parallel 64 times. The specific mapping rule is as follows: for the five internal 64-bit words x_0, x_1, x_2, x_3, x_4, each S-box independently processes the values of these five words at the same bit position specifically, the input bits for the j-th S-box ($0 \le j < 64$) are $x_0[j], x_1[j], x_2[j], x_3[j], x_4[j]$, where x0 corresponds to the most significant bit.
    - Linear Diffusion Layer: Each 64-bit register word xi undergoes diffusion via a linear function ∑i(xi); that is, $xi \leftarrow \sum i\,(xi), i=0, \ldots, 4$.

$x0 \leftarrow \sum 0\,(x0) = x0 \oplus (x0 >>> 1\ 9) \oplus (x0 >>> 2\ 8)\ x1 \leftarrow \sum 1\,(x1) = x1 \oplus (x1 >>> 61) \oplus (x1 >>> 39)$

$x2 \leftarrow \sum 2\,(x2) = x2 \oplus (x2 >>> 1) \oplus (x2 >>> 6)$

$x3 \leftarrow \sum 3\,(x3) = x3 \oplus (x3 >>> 10) \oplus (x3 >>> 17)$

$x4 \leftarrow \sum 4\,(x4) = x4 \oplus (x4 >>> 7) \oplus (x4 >>> 41)$

where x >>> i denotes a right cyclic shift of the 64-bit word x by i bits.

2.3. Differential Analysis

Differential cryptanalysis, proposed by Biham and Shamir, is a cryptanalytic method based on chosen-plaintext attacks. Its core lies in studying the propagation patterns of a specific input difference Δ_{in} to an output difference Δ_{out} within an iterated cipher system. This method has now become one of the benchmark metrics for evaluating the security of block ciphers; its effectiveness relies on the construction of efficient differential distinguishers.

- **Differential Characteristic:** An i-round differential characteristic $\Omega = (\beta 0, \beta 1, \ldots, \beta i\text{-}1, \beta i)$ for an iterated block cipher refers to the condition where, given an input pair (X, X^*) with a difference value satisfying $X \oplus X^* = \beta 0$, the difference values of the intermediate states (Yj, Yj^*) during the i-round encryption process satisfy $Yj \oplus Yj^* = \beta j\ (1 \le j \le i)$.
- **Differential Probability:** The probability $DP(\alpha, \beta)$ corresponding to an i-round differential (α, β) for an iterated block cipher refers to the likelihood that, given an input pair (X, X^*) satisfies $X \oplus X^* = \alpha$, the difference of the output pair (Yi, Yi^*) after i rounds of satisfies $Yj \oplus Yj^* = \beta$, where the round keys are independent and uniformly ran-dom. When the probability of an (r-1)-round differential characteristic significantly exceeds the probability associated with a random permutation (namely $\frac{1}{2^n}$), an (r-1)-round differential distinguisher is constructed.

While S-box design remains an important aspect of cipher security, this work focuses specifically on neural differential distinguisher construction rather than proposing new S-box mechanisms. Previous studies have explored various approaches for improving S-box cryptographic properties. Khan *et al*. [20] investigated natural randomness-based S-box construction methods, while Alali *et al*. [21] utilized graph-theory metrics to construct secure S-boxes with desirable cryptographic characteristics. Ye and Chen [22] demonstrated that effective S-box substitution strategies can improve cipher security properties. Furthermore, Jamal *et al.* [23], Saleem *et al*. [24], and Ahmad *et al*. [25] explored hybrid and chaos-based S-box approaches aimed at enhancing nonlinearity, unpredictability, and security characteristics in lightweight cryptographic applications.

When the probability of an (r-1)-round differential characteristic is observed to significantly exceed the probability associated with a random permutation namely $\frac{1}{2^n}$ an (r-1) round differential distinguisher is constructed. This distinguisher per-forms a binary classification by analyzing the joint distribution patterns of (Δ_{in}, Δ_{out}), distinguishing between two categories: a genuine (r-1)-round. encryption process and a random permutation. Subsequently, by utilizing this distinguisher, it is possible to progressively recover portions of the secret.

Key Information: The higher the prob-ability of a differential distinguisher, the less data is required to recover the key; conversely, the longer the path covered by the distinguisher, the fewer computational resources are needed for the attack.

2.4. Neural Differential Distinguisher

The neural differential distinguisher achieves the efficient classification of cryptographic data through deep learning; fundamentally, it bears an intrinsic similarity to classification tasks found in fields such as image recognition and speech processing. The neural differential distinguisher [5] operates by training a neural network model to precisely distinguish between real ciphertext pairs (generated by encrypting plaintext pairs with a fixed input difference) and random ciphertext pairs (generated by encrypting random plaintext pairs). Once trained, Key recovery is beyond the scope of this paper, as we focus solely on distinguisher construction, where it demonstrates superiority over traditional methods in terms of both classifica-

tion accuracy and attack complexity. The construction of the training dataset follows the procedure outlined below:

a) Randomly generate a master key K;
b) Construct a plaintext pair (P, P∗) that satisfies a specific input difference of Δ in;
c) Encrypt the plaintext pair (P, P∗) to generate a ciphertext pair (C, C∗), which serves as a training sample.

During the neural network training process, each sample is assigned a label Y, taking a value of either 0 or 1. A value of 1 indicates that the data was generated by encrypting a plaintext pair (P, P∗) with an input difference of Δ_{in}, whereas a value of 0 indicates that the data was generated by encrypting a random pair.

The distinguisher evaluates the input ciphertext; if it determines that the ciphertext was generated from a fixed difference, it outputs 1; otherwise, it outputs 0. The performance of the distinguisher refers to its ability to correctly identify positive and negative samples, with classification accuracy serving as the primary evaluation metric. When the validation accuracy exceeds the 50% probability associated with random guessing, the model is deemed to be a valid neural differential distinguisher. A high-accuracy distinguisher model not only enhances the efficiency of key recovery but also significantly reduces the complexity of the attack. Based on this methodology, Gohr successfully recovered the round keys for the 11-round SPECK32/64 algorithm, Key recovery is not performed in this work, as our focus is on distinguisher accuracy.

3. Construction Method for Neural Differential Distinguishers

Deep learning models have demonstrated significant advantages in the field of cryptanalysis. By employing deep neural networks to perform feature learning and pattern recognition on cipher text data, it is possible to capture the specific regularities exhibited by input differentials with greater precision, thereby constructing a more com-prehensive feature library of differential paths. This section investigates the construction method for neural differential distinguishers; its fundamental framework comprises three components: dataset construction, network architecture configuration, and model training.

3.1. Dataset Construction

Supervised learning constitutes a vital branch of deep learning, centered on the utilization of labeled training datasets. Within such a dataset, each sample is associated with a specific label.

A set of binary labels, Y, of length n is created, where Y = 1 denotes a valid differential pair, and Y = 0 denotes a random noise pair. Subsequently, the elements of Y are paired with each corresponding set of elements in the set P*. For negative samples (cases where Y = 0), the corresponding P* is re-placed with a random plaintext P**.

Next, the plaintext pairs are encrypted using the target encryption algorithm to yield ciphertext pairs (C, C*). The specific encryption process is defined as follows:

$$C = Encrypt(K, P)$$
$$C^* = Encrypt(K, P^*) \text{ , } (if\ Y = 1);$$
$$C^* = Encrypt(K, P\_random), (if\ Y = 0)$$

Finally, each ciphertext pair is trans-formed into a triplet (CC*, C, C*), where-in CC* explicitly preserves the differential features. For a sample comprising S ciphertext pairs, these S triplets are concatenated along the feature dimension to form the final input tensor.

During the dataset construction phase, we adopt a multi-ciphertext-pair differential input structure, namely the (CC*, C, C*) format under a multi-ciphertext setting. The detailed workflow is illustrated in Figure 1, and the corresponding dataset structure is shown in Figure 2.

By constructing the dataset using this method, the input samples are designed to encompass not only the inherent features of the ciphertexts themselves but also the differential relationships existing between pairs of ciphertexts, thereby enhancing the model's sensitivity to differential features. According to the findings presented in [26], the decision-making mechanism of a neural differential distinguisher fundamentally relies on the differential distribution characteristics of ciphertext pairs; essentially, during the training phase, the neural differential distinguisher constructs a high-precision approximation of the cryptographic differential distribution table and utilizes this information directly to classify ciphertext pairs. This conclusion underscores the necessity of explicitly incorporating ciphertext differential information during the dataset construction process. Furthermore, the design involving the concatenation of multiple cipher-text pairs enables the model to capture the correlations existing among these pairs. When utilizing multiple ciphertext pairs as input since multiple pairs correspond to a single label the statistical features across these pairs are further aggregated, thereby mitigating the impact of random noise. Consequently, compared to scenarios in-volving single ciphertext pairs, inputs comprising multiple pairs exhibit lower variance, thereby achieving the objective of enhancing the accuracy of the differential distinguisher.

3.2. Network Architecture Configuration

Inspired by the Residual Network (ResNet) architecture introduced by Gohr [5] for neural differential cryptanalysis, this paper proposes a modified residual network architecture named RSBG-ResNet (Residual Shrinkage Block with Gated Mechanism). The proposed architecture extends the conventional ResNet framework by incorporating adaptive residual shrinkage and feature gating mechanisms to improve feature extraction capability for neural differential distinguishers. Through its residual

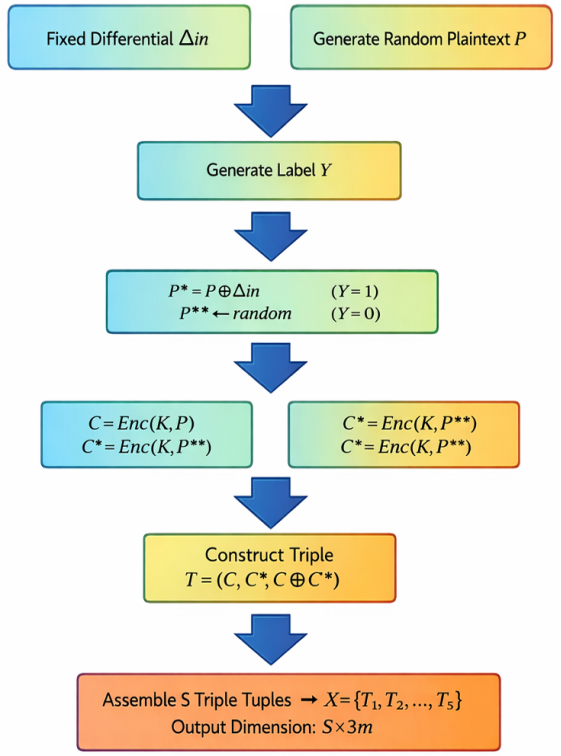

Figure 1. Dataset Construction Process.

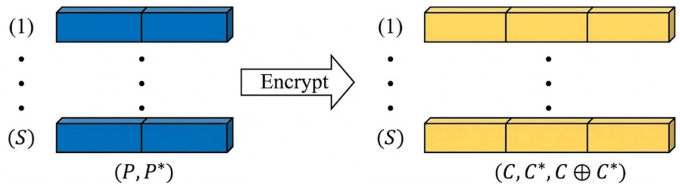

Figure 2. Dataset format.

connection mechanism, the proposed RSBG-ResNet alleviates the vanishing gradient problem commonly encountered in deep convolutional neural networks, thereby improving training stability and classification performance.

This paper adopts a modified Residual Network architecture, the core innovation of which lies in its adaptive threshold mechanism. This architecture begins with the input data dimensions; a reshaping layer adjusts the data's shape to form a feature sequence suitable for subsequent convolutional operations. An initial convolutional layer employs small-sized kernels to perform preliminary feature ex-traction on the input features, utilizing Batch Normalization and the ReLU activation function to enhance feature representational capacity. Subsequently, the network employs two additional fully connected layers to perform a nonlinear mapping on these initial convolutional features, thereby enhancing the model's feature adaptability and generating an initial feature representation.

The main body of the network consists of multiple Residual Shrinkage Blocks (RSBs); each RSB module comprises two convolutional layers, Batch Normalization, and the ReLU activation function, serving the purpose of deep feature extraction. The module incorporates a soft-thresholding mechanism; by computing the absolute values of the residual path and integrating global average pooling, fully connected layers, and a scaling fac-tor generator network, it dynamically adjusts feature weights to achieve feature selection and noise suppression. To pre-vent dimensional mismatches, whenever the number of input and output channels differs, the network employs 1×1 convolutions to adjust the input, thereby ensuring the efficacy of the residual connections. Compared to Residual Shrinkage Networks designed for general image processing tasks, the parameter settings here have been specifically adapted for cryptanalysis tasks: key dimensions such as the number of output channels are uniformly adjusted to be integer multiples of the cryptographic algorithm's block length. This alignment precisely matches the network structure to the output length of the cryptographic algorithm, thereby enhancing the network's capacity to learn intra-block correlational features. Simultaneously, a transition from a two-dimensional to a one-dimensional representation is executed, establishing a block-sequential processing paradigm tailored for ciphertext data streams; this approach not only accommodates the inherent structural characteristics of ciphertext data but also significantly reduces the total number of model parameters.

Following the stacking of these modules, the network utilizes a flattening layer to convert the multi-dimensional feature maps into a one-dimensional feature vector. Subsequently, fully connected layers are employed to progressively reduce the feature dimensionality, ultimately yielding the final prediction results. The over-all network design synthesizes the strengths of both residual learning and attention mechanisms: it effectively mitigates the vanishing gradient problem commonly encountered in deep networks while simultaneously leveraging soft-thresholding to dynamically focus on salient features, thereby enhancing both the model's expressive power and its generalization performance. The specific architecture of the RSBG-ResNet is illustrated in Figure 3.

The training parameters were configured as follows: a training set comprising 1×10^7 samples and a validation set comprising 1×10^6 samples were randomly generated. The validation set was utilized solely to assess the model's generalization capability and was excluded from the parameter optimization process. The raw data underwent preprocessing to convert it into the input format required by the neural network; the dataset was processed in batches with a batch size of 1×10^4, spanning a total of 120 training epochs. A cyclical learning rate strategy was adopted, featuring a step size of 30 epochs, a base learning rate of 1×10^{-4}, and a maximum learning rate of 2×10^{-3}. The Mean Squared Error (MSE) was selected as the loss function following [5] (Gohr), who demonstrated its

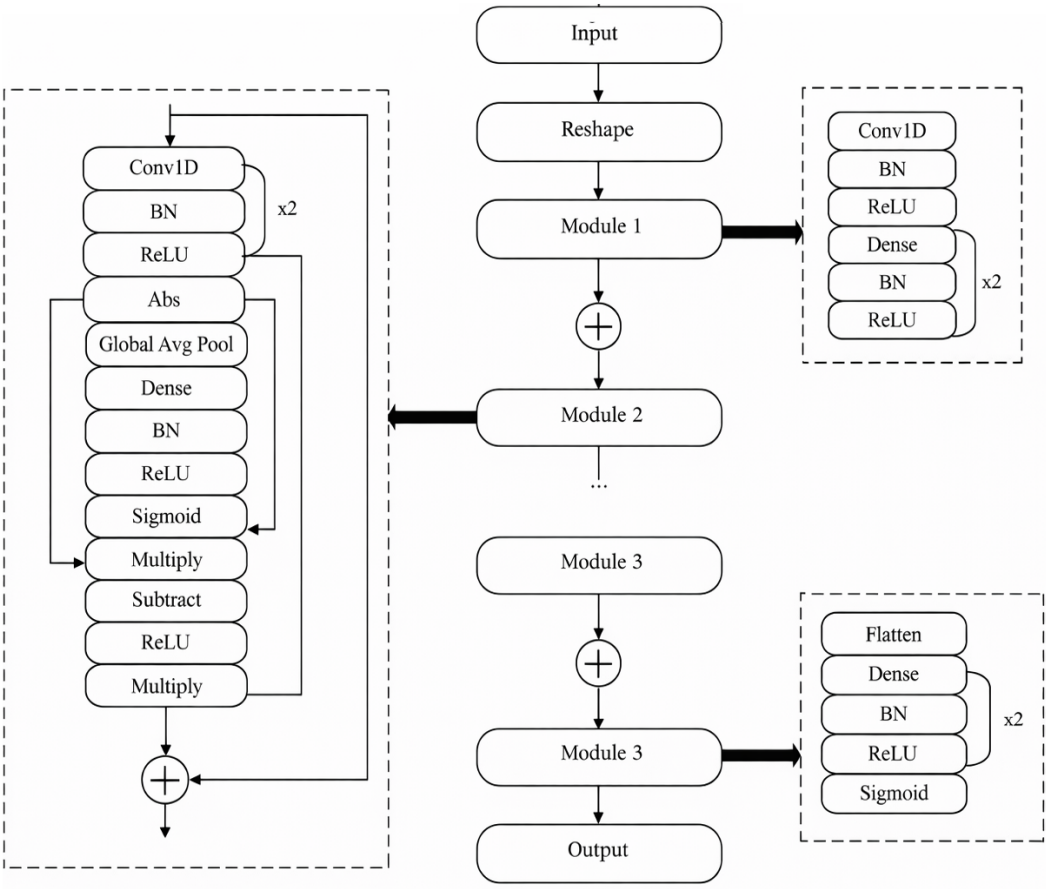

Figure 3. Network structure of RSBG-ResNet.

effectiveness for neural distinguishers despite BCE being standard for classification, and the Adam algorithm was employed as the optimizer, supplemented by L2 regularization (with a coefficient of $\lambda = 1 \times 10^{-4}$) to constrain model complexity. To further validate the effectiveness of the proposed network architecture, we take the 6-round GIFT-128 algorithm as an example. We select the standard ResNet architecture (as used in prior work) to serve as a baseline model for a comparative experiment against our proposed RSBG-ResNet architecture. The experimental results, obtained under identical training parameter settings, are presented in Table 1.

4. Application of the Neural Differential Distinguisher for GIFT-128

Based on the construction method for neural differential distinguishers presented in Section 3, this section constructs a neural differential distinguisher for the GIFT-128 algorithm, addressing two key aspects: dataset construction and network architecture selection.

The GIFT-128 algorithm is based on an SPN structure; its round function sequentially performs substitution and permutation operations to produce a 128-bit ciphertext. It employs a data construction paradigm specifically, "ciphertext pairs + ciphertext differences" to precisely capture the nonlinear properties of the S-boxes and the patterns of bit permutations.

For the 6-round and 7-round variants of GIFT-128, and taking into account the algorithm's structural characteristics, the input differentials are prioritized to favor fixed differentials with low Hamming weights. During differential propagation, this approach minimizes the occurrence of unnecessary "active bits" resulting from nonlinear operations, thereby facilitating propagation with higher probabilities and enabling the analysis of a greater number of algorithm rounds.

Synthesizing the aforementioned considerations and incorporating the dataset construction methodology described in Reference [10] experiments were conducted using a fixed input differential with a Hamming weight of 1 (specifically, 0x00000000080000000000000000000000). To investigate the impact of the quantity of ciphertext pairs on model performance, datasets containing varying scales of ciphertext pairs were further constructed and subsequently utilized to train the neural differential distinguisher.

Experimental results demonstrate that, under identical ciphertext pair conditions, the RSBG-ResNet architecture employed in this paper effectively enhances the accuracy of the neural differential distinguisher, thereby validating the efficacy of this network structure. The comparative results of these experiments are presented in Table 2 and 3, as well as in Figure 4.

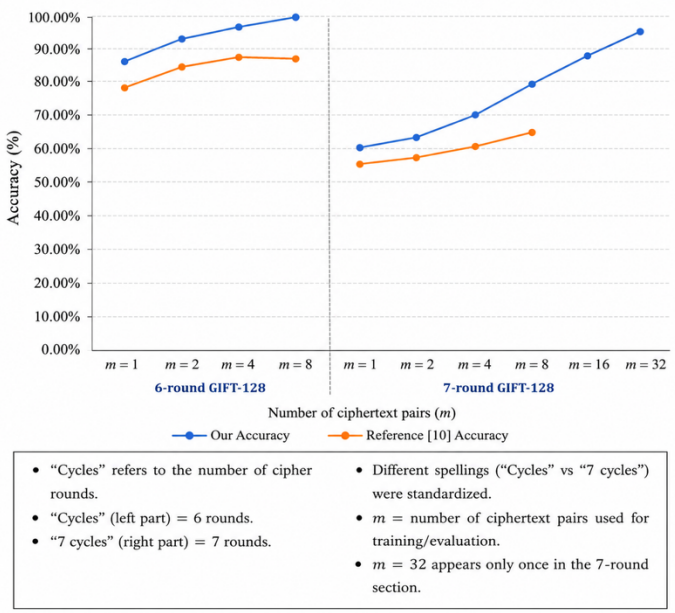

Figure 4. Accuracy comparison of neural differential distinguishers for 6-round and 7-round GIFT-128 under different numbers of ciphertext pairs (m).

Table 1. Accuracy comparison of Neural Differential Distinguishers for 6-round GIFT-128 Under Different Network Architectures.

Number of Ciphertext Pairs	ResNet Accuracy (%)	RSBG-ResNet Accuracy (%)
1	78.83	86.36
2	87.67	93.80
4	96.04	97.93
8	98.90	99.70

Table 2. Accuracy comparison of neural differential distinguishers for 6-round GIFT 128.

Number of Ciphertext Pairs	Original Text Accuracy (%)	Reference [10] (%)
1	86.36	78.36
2	93.80	84.77
4	97.93	88.00
8	99.70	87.12

Table 3. Accuracy comparison of neural differential distinguishers for 7-round GIFT 128.

Number of Ciphertext Pairs	Original Text Accuracy (%)	Reference [10] (%)
1	60.85	56.06
2	63.60	55.64
4	70.54	50.17
8	79.71	50.06
16	88.31	-
32	95.47	-

A comprehensive analysis indicates that the proposed neural differential distinguisher achieves consistent performance improvements for the 6-round and 7-round GIFT-128 settings. It is observed that as the number of cipher-text pairs increases, the accuracy of the distinguisher improves correspondingly; furthermore, for equivalent scales of ciphertext pairs, the experimental results generally outperform those reported in Reference [10]. Specifically, with 8 pairs of ciphertexts over 6-rounds, the proposed distinguisher achieves an accuracy of 99.70% achieving, to the best of our knowledge, im-proved accuracy compared with [10] for this specific round count. For 8 pairs of ciphertexts over 7-rounds, the distinguisher demonstrates a noticeable improvement relative to Reference [10]. Additionally, this study marks the first exploration of ciphertext pair scales of 16 and 32 within the 7-round algorithm, achieving stable distinguishing performance. An analysis comparing two distinct distinguisher architectural de-signs reveals that the newly constructed data format facilitates superior extraction of features embedded within the ciphertext pairs, thereby enabling the neural network model to learn a greater number of latent features and further enhancing the accuracy of the distinguisher.

5. Application of the Neural Differential Distinguisher to ASCON-Permutation Lightweight Permutation-Based Cryptographic Algorithm

Focusing on the 4-round ASCON-Permutation lightweight permutation-based cryptographic algorithm and taking into account its unique permutation structure, which yields a 320-bit ciphertext output we adopt a strategy involving ciphertext pairs combined with ciphertext differences to better extract correlative features between ciphertexts. To minimize the occurrence of unnecessary active bits and enable the differential to propagate through a greater number of rounds with higher probability, we utilize input differentials with low Hamming weights. Consequently, during the dataset construction phase, we generated a fixed differential with a Hamming weight of 1 by XORing the 64-bit register x0 with the differential value 0x00000001; we then constructed distinct datasets comprising ciphertext pairs of varying scales. By applying the network architecture proposed in Section 4, we trained the corresponding neural differential distinguishers. The specific comparative results are presented in Table 4 and Figure 5.

As indicated by the experimental results presented in Table 4 and Figure 5, the proposed method demonstrates modest but stable gains over the baseline method in the 4-round ASCON-PERMUTATION setting. The experiments demonstrate that as the number of ciphertext pairs increases, the improvement becomes more noticeable. Given the same data scale, the proposed method demonstrates competitive performance relative to Reference [10]. Taking 32 ciphertext pairs as an example, the accuracy rate of the proposed method reaches 53.54%—an improvement of 3.52%—and achieves higher accuracy than the compared baseline configuration at both the 16-pair and 32-pair scales. In scenarios involving a low number of ciphertext

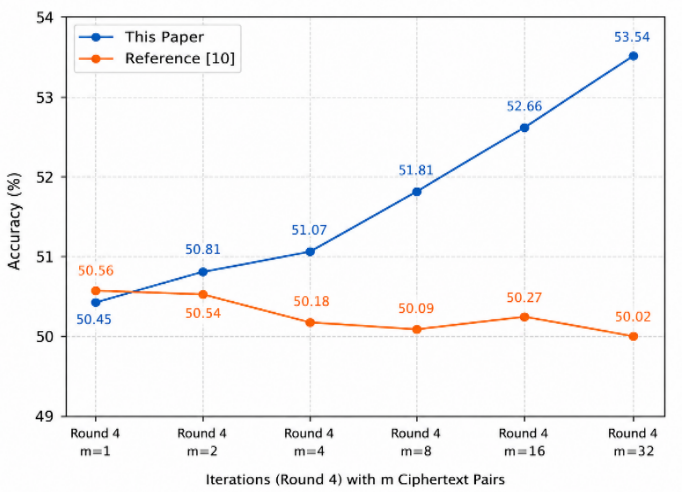

Figure 5. Comparative results of neural differential for ASCON-PERMUTATION.

Table 4. Accuracy comparison of neural differential distinguishers for ASCON-PERMUTATION.

Number of Ciphertext Pairs	Original Text Accuracy (%)	Reference [10] (%)
1	50.45	50.56
2	50.81	50.54
4	51.07	50.18
8	51.81	50.09
16	52.66	50.27
32	53.54	50.02

pairs, the accuracy of the proposed method is comparable to the benchmark established in Reference [10]; this suggests that there remains room for improvement in the method's feature extraction capabilities when operating with limited data. An analysis of the discriminator architecture reveals that the new data format achieves accuracy gains of 1.72%, 2.39%, and 3.52% for input sets of 8, 16, and 32 ciphertext pairs, respectively. These results suggest that the proposed input representation and network architecture can improve feature extraction capability under larger ciphertext-pair settings.

6. Conclusions

This paper proposes a construction method for neural differential distinguishers that integrates a multi-ciphertext-pair input structure with a residual shrinkage network. This approach successfully enhances the accuracy of neural differential distinguishers when applied to the GIFT-128 and ASCON-PERMUTATION lightweight permutation-based cryptographic algorithm. By adopting the (CC*, C, C*) multi-ciphertext-pair input format, the method explicitly preserves differential features and captures correlations across ciphertext pairs; furthermore, the incorporation of an adaptive residual shrinkage network significantly improves feature extraction capabilities.

This study presents a neural differential analysis framework integrating dataset construction, network architecture, and training strategy for lightweight cryptographic analysis. It offers a new paradigm for the security assessment of lightweight ciphers and demonstrates the applicability of deep learning techniques in neural differential cryptanalysis. The current work, however, presents certain limitations: the "black-box" nature of deep learning models results in a lack of interpretability regarding the underlying cryptographic mechanisms driving the decision-making process, and the framework's transferability across different algorithms has not yet been systematically validated. Future research will focus on dynamic feature fusion mechanisms, cross-algorithm transfer learning, and enhanced interpretability, aiming to further expand the depth and breadth of applications for neural differential analysis.

7. Declarations

7.1. Author Contributions

Muhammad Ahmad: Conceptualization, Methodology (Lead), Writing Original Draft; **Hua Zhou:** Supervision, Project Administration; **Muhammad Usman:** Conceptual Guidance, Formal Analysis; **Tanzeela Bibi:** Data Visualization, Review and writing; **Haider Ali:** Investigation and Review; **Maryum Shahzadi:** Data Collection and Software Validation; **Farah Javed:** Critical Review, Writing-Review, Editing.

7.2. Institutional Review Board Statement

Not applicable.

7.3. Informed Consent Statement

Not applicable.

7.4. Data Availability Statement

The data used in this study is available on request from the corresponding author.

7.5. Acknowledgment

Not applicable.

7.6. Conflicts of Interest

The authors declare no conflict of interest.

8. References

[1] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, pp. 3–72, 1991. https://doi.org/10.1007/BF00630563.

[2] X. Lai, "Higher order derivatives and differential cryptanalysis," *Communications and Cryptography*, 1994, pp. 227–233. https://doi.org/10.1007/978-1-4615-2694-0_23.

[3] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials," in *Advances in Cryptology EUROCRYPT '99*. Berlin: Springer, 1999, vol. 1592, pp. 12–23. https://doi.org/10.1007/3-540-48910-X_2.

[4] E. Biham, "New types of cryptanalytic attacks using related keys," *Journal of Cryptology*, vol. 7, pp. 229–246, 1994. https://doi.org/10.1007/BF00203965.

[5] A. Gohr, "Improving attacks on round-reduced Speck32/64 using deep learning," in *Advances in Cryptology—CRYPTO 2019*, 2019, pp. 150–179. https://doi.org/10.1007/978-3-030-26951-7_6.

[6] Y. Chen and H. Yu, "A new neural distinguisher model considering derived features from multiple ciphertext pairs," *The Computer Journal*, vol. 66, no. 6, pp. 1419–1433, 2023. https://doi.org/10.1093/comjnl/bxac019.

[7] Z. Hou, J. Ren, and S. Chen, "Improved neural distinguisher for cryptanalysis," [Online]. Available: https://eprint.iacr.org/2021/1017.

[8] J. Lu, G. Liu, B. Sun, *et al.*, "Improved (related-key) differential-based neural distinguishers for SIMON and SIMECK block ciphers," *The Computer Journal*, vol. 67, no. 2, pp. 537–547, 2024. https://doi.org/10.1093/comjnl/bxac195.

[9] Z. Bao, J. Lu, Y. Yao, L. Zhang, "More insight on deep learning-aided cryptanalysis," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*. Singapore: Springer, 2023, pp. 436–467. https://doi.org/10.1007/978-981-99-8727-6_15.

[10] D. Shen *et al.*, "Neural differential distinguishers for GIFT-128 and ASCON," *Journal of Information Security and Applications*, vol. 82, p. 103758, 2024. https://doi.org/10.1016/j.jisa.2024.103758.

[11] L. Zhang, Z. L. Wang, and B. C. Wang, "Improving differential-neural cryptanalysis," *IACR Communications in Cryptology*, vol. 1, no. 3, p. 13, 2024. https://doi.org/10.62056/ay11wa3y6.

[12] G. Wang, G. Wang, and S. Sun, "Investigating and enhancing the neural distinguisher for differential cryptanalysis," *IEICE Transactions on Information and Systems*, vol. 107, pp. 1016–1028, 2024. https://doi.org/10.1587/transinf.2024EDP7011.

[13] B. Seok and C. Lee, "A novel approach to construct a good dataset for differential-neural cryptanalysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 1, pp. 246–262, 2024. https://doi.org/10.1109/TDSC.2024.3387662.

[14] Y. Banga, Y. Mahmood, N. Al Mudawi, N. Innab, N. Iqbal, and H. Diab, "Where octagonal geometry meets chaos: A new S-box for advanced cryptographic systems," *PLoS One*, vol. 20, no. 6, Jun. 2025. https://doi.org/10.1371/journal.pone.0320457.

[15] Y. S. Vaz, J. C. B. Mattos, and R. I. Soares, "High throughput-to-area AES: The role of small S-box in lightweight cryptographic design," in *Proc. 2025 IEEE 16th Latin American Symposium on Circuits and Systems (LASCAS)*, 2025. https://doi.org/10.1109/LASCAS64004.2025.10966232.

[16] P. S. Curlin, J. Heiges, C. Chan, and T. S. Lehman, "A survey of hardware-based AES S-boxes: Area, performance, and security," *ACM Computing Surveys*, vol. 57, no. 9, Apr. 2025. https://doi.org/10.1145/3724114.

[17] A. H. Zahid, M. J. Arshad, M. Ahmad, N. F. Soliman, and W. El-Shafai, "Dynamic S-box generation using novel chaotic map with nonlinearity tweaking," *Computers, Materials and Continua*, vol. 75, no. 2, pp. 3011–3026, 2023. https://doi.org/10.32604/cmc.2023.037516.

[18] S. Banik *et al.*, "GIFT: A small present: Towards reaching the limit of lightweight encryption," in *Proceedings of Cryptographic Hardware and Embedded Systems*. Springer, 2017, pp. 321–345. https://doi.org/10.1007/978-3-319-66787-4_16.

[19] C. Dobraunig, M. Eichlseder, F. Mendel, *et al.*, "Ascon v1.2: Lightweight authenticated encryption and hashing," *Journal of Cryptology*, vol. 34, pp. 1–42, 2021. https://doi.org/10.1007/s00145-021-09398-9.

[20] M. F. Khan, K. Saleem, T. Shah, M. M. Hazzazi, I. Bahkali, and P. K. Shukla, "Block cipher's substitution box generation based on natural randomness in underwater acoustics and knight's tour chain," *Computational Intelligence and Neuroscience*, vol. 2022, 2022. https://doi.org/10.1155/2022/8338508.

[21] S. Alali, M. K. Jamil, R. Ali, R. Alotaibi, and W. Albalawi, "Degree, closeness and eigenvector for the construction of cryptographically secure S-boxes," *Ain Shams Engineering Journal*, vol. 16, no. 10, Oct. 2025. https://doi.org/10.1016/j.asej.2025.103559.

[22] J. Ye and Y. Chen, "SC-SA: Byte-oriented lightweight stream ciphers based on S-box substitution," *Symmetry*, vol. 16, no. 8, Aug. 2024. https://doi.org/10.3390/sym16081051.

[23] S. S. Jamal, R. Ali, M. K. Jamil, S. A. Nooh, F. Alblehai, and Gulraiz, "Secure S-box construction with 1D chaotic maps and finite field theory for block cipher encryption," *Alexandria Engineering Journal*, vol. 125, pp. 278–296, Jun. 2025. https://doi.org/10.1016/j.aej.2025.03.109.

[24] M. R. Saleem, A. H. Zahid, and G. Mustafa, "Dynamic S-box construction based on a novel chaotic map and twitching approach," *Mehran University of Engineering and Technology Research Journal*, vol. 44, no. 4, pp. 101–119, 2025. https://doi.org/10.22581/muet1982.0354.

[25] M. Ahmad, H. Zhou, T. Bibi, and H. Ali, "A review of cryptographic solutions and forensic readiness in IoT and network security," *International Journal of Science, Engineering and Technology*, vol. 14, no. 2, 2026. https://doi.org/10.5281/zenodo.19449141.

[26] A. Benamira, D. Gerault, T. Peyrin, *et al.*, "A deeper look at machine learning-based cryptanalysis," in *Advances in Cryptology—EUROCRYPT 2021*. Cham: Springer, 2021, pp. 436–467. https://doi.org/10.1007/978-3-030-77870-5_28.