RLND: A ResNet and LSTM based neural distinguisher for lightweight block ciphers

Jie Liu*, Junjie Xu, Yunhao Qiu, and Qibo Liu

Northwestern Polytechnical University, China
lucky_jiel@163.com, jjxu@mail.nwpu.edu.cn, qiu-yunhao@mail.nwpu.edu.cn, liuqibo@mail.nwpu.edu.cn

Abstract. Block ciphers are widely used to ensure the security of data in the digital age, yet the techniques for analyzing them continue to advance. Recently, neural distinguishers have garnered significant attention due to their higher accuracy compared to traditional methods when analyzing block ciphers. In this work, we propose a novel neural distinguisher, named RLND, which can extract both spatial and temporal features. Specifically, we design a multi-scale convolutional kernel-based residual network (MSCK-ResNet) to fuse spatial features across different scales through varied receptive fields. To capture temporal dependencies, we construct a multi-layer LSTM (ML-LSTM) module with varying numbers of neurons. The optimal configuration of RLND, including the number of MSCK-ResNet blocks and ML-LSTM layers, is determined experimentally to maximize accuracy. We evaluate the performance of the proposed distinguisher on five lightweight block ciphers, demonstrating that it achieves up to 0.5% and 5.5% higher accuracy with single and multiple ciphertext pairs, respectively. This highlights the strong potential of RLND for practical key recovery.

Keywords: Neural distinguisher · Lightweight block cipher · Differential analysis · Key recovery

1 Introduction

Differential cryptanalysis is an efficient chosen-plaintext attack [1], which was first proposed in 1991. The core idea is to perform statistical analysis by studying how input differences propagate to output differences. Matsui [2] proposed a branch-and-bound search algorithm to find the best differential characteristics. Additionally, several automated search tools, such as Mixed Integer Linear Programming (MILP) [3, 4] and Constraint Programming (CP) [5, 6], have played a crucial role in addressing such challenges. Recently, machine learning-based differential distinguishers have gained more attention due to their high efficiency in distinguishing difference pairs from random pairs [7, 8].

In Crypto 2019, Gohr [7] first proposed a machine learning-based differential distinguisher, which achieved a high success rate in key recovery. He verified the

* Corresponding author.

differential distribution model using a Markov model and trained a deep residual network on cipher pairs with specified input differences and random cipher pairs. Furthermore, he proposed a multi-stage key recovery attack on round-reduced Speck 32/64 to validate the efficiency of the proposed neural distinguisher. This work was considered a significant breakthrough at the intersection of cryptanalysis and machine learning. To further improve the accuracy of neural distinguishers, researchers have deeply studied dataset construction methods and neural network structures. Chen et al. [8] proposed using multiple ciphertext pairs instead of a single pair as training samples, effectively improving the accuracy of round-reduced Speck 32/64 neural distinguishers. Hou et al. [9] further improved this by using output differential pairs to replace multiple ciphertext pairs, achieving better accuracy. Gohr [10] summarized related works and proposed a method for converting single-pair distinguishers to multiple-pair distinguishers, known as the combine method. He noted that, for 5 round-reduced Speck 32/64, an accuracy improvement of approximately 0.5% could be considered significant. Additionally, Zhang et al. [11] discovered that using the right half of the penultimate round ciphertext in Speck 32/64, which can be obtained at no cost, significantly improved the accuracy of distinguishers by incorporating it into the ciphertext pairs. In 2023, Liu et al. cascaded the ciphertext that was decrypted by one round with a random key to the ciphertext pair [12], which also enhanced accuracy.

Another approach is to optimize the neural network. Zhang et al. introduced the inception layer into Gohr's neural network, achieving improvements on Speck 32/64 and Simon 32/64 [11]. Bao et al. [13] employed a deep dense network and a Squeeze-and-Excitation network to construct effective distinguishers for Simon 32/64. Furthermore, Baksi [14] proposed a non-Markovian model based on a machine learning distinguisher and compared it with MLP, LSTM, and CNN model-based distinguishers. Baksi and Wang [15] discussed neural networks using MLP [16], LSTM [17], and CNN [18] separately. The results showed that MLP outperformed LSTM, while CNN performed poorly. Bose et al. [19] explored the distinguishers using LSTM and Transformer, respectively. Jiang et al. [20] introduced the attention mechanism in to RegNet to improve the accuracy. These methods inspire the improvement of the neural network for the distinguisher.

Common attacks against ciphers include chosen-plaintext attacks (CPA), known-plaintext attacks (KPA), and ciphertext-only attacks (COA). CPA assumes that the attacker has access to the encryption system and can control the input plaintext. In this scenario, the key can be recovered using carefully crafted plaintext-ciphertext pairs. Although CPA is relatively easy to implement, its assumption is often unrealistic in many practical situations. In a KPA, a set of plaintext-ciphertext pairs is obtained randomly rather than chosen by the attacker. This type of attack poses a broader threat to cryptosystems than CPA but is more challenging to execute in practice. In a COA, the attacker only has access to a limited number of ciphertexts without any corresponding plaintexts, making it the most challenging scenario due to its minimal assumptions. Cryptanalysis for standard cryptosystem such as AES and ASCON usually em-

ploies the differential attacks. Zhang et al. applied the neural distinguisher on the key recovery of DES, Chaskey, PRESENT [21]. Mishra et al. explored the neural distinguisher based key recovery attack on lightweight ciphers GIFT and PRIDE [22, 23]. The deep learning based key recovery attack for ciphers with SPN structure is is explored by Kimura et al. [24].

Motivations. Although neural distinguishers have outperformed traditional ones, there is still considerable gap for improvement in accuracy. Many existing approaches rely solely on individual neural network architectures. For example, ResNet structures are primarily designed to extract spatial features, while LSTM networks excel at capturing temporal dependencies in input sequences. Combining these complementary structures and optimizing them in an integrated framework could lead to higher distinguishing accuracy. Furthermore, the complexity of recovering the plaintext of the standard ciphers is equivalent to breaking the underlying cipher. Therefore, we explore the integration of ResNet and LSTM to simultaneously extract spatial and temporal features, enabling the development of more effective neural distinguishers.

Our Contributions. In this work, we propose a novel neural distinguisher with higher accuracy capturing both spatial and temporal features. The optimal network and parameters are simulated and compared with other neural distinguishers:

(i) We design a neural distinguisher based on ResNet and LSTM, denoted RLND. A new residual network based on a multi-scale convolution kernel is designed, defined as MSCK-ResNet, to extend the receptive field and combine information from different scales. Furthermore, we construct a multi-layer LSTM (ML-LSTM) model, which consists of a reshape layer and three multi-layer LSTM layers. Different multi-layer LSTM layers and neuron numbers are designed to explore the influence of LSTM architecture on the accuracy of the neural distinguisher.

(ii) To validate the effectiveness of LSTM in capturing temporal characteristics, we design an ablation scheme, which cascades an improved MLP to the output of ResNet, denoted RMND. Different numbers of fully connected (FC) layers and neurons are used in the MLP layer of the RMND. We implemented the ablation experiment for the proposed RLND and compared it with other distinguishers, which shows that RLND learns additional temporal features compared to RMND and CNN-based distinguishers.

(iii) We simulate the proposed neural distinguishers to determine the optimal network structure and parameters. Then, they are compared with existing methods across different block ciphers. The results demonstrate that the proposed RLND significantly outperforms distinguishers based solely on CNN or LSTM architectures, especially when tested on input datasets of varying sizes. Additionally, the proposed RLND is applied to five different block ciphers. The result shows that the maximum advantage can reach 0.9% for Speck 32/64, and 8.02% for Simeck 32/64, and 2.97% for Chaskey, and 6.67% for PRESENT 64/80, and 3.14% for DES.

Organization. The remainder of the paper is organized as follows: Section 2 provides an overview of Gohr's neural distinguisher and introduces the five block ciphers under study. Section 3 presents the proposed RLND neural distinguisher in detail. Section 4 describes the simulation experiments used to determine the optimal network structures and parameters, and evaluates the accuracy of RLND for different ciphers. Section 5 concludes the paper and outlines potential directions for future research.

2 Preliminaries

2.1 Notation

In this work, $P = (P_0, P_1)$ and $C = (C_0, C_1)$ are defined as a plaintext/ciphertext pair. ΔP denotes the plaintext difference. N and M denote the number of samples in the training and test sets, respectively. The number of rounds is denoted as r. The symbol $\mathcal{ND}$ is used to represent a neural distinguisher. The performance of neural distinguishers is evaluated using accuracy as the metric.

2.2 Five Block Ciphers

Speck 32/64 [25] is a lightweight block cipher with block size 32 bits, where the modular addition is the only non-linear component. Chaskey [26] is a message authentication code (MAC) algorithm with intermediate state size 128, and its non-linear component is also the modular addition. PRESENT 64/80 [27, 28] is another lightweight block cipher with a block size of 64 bits. Its non-linear component is a 4×4 S box. The Simeck [29] family ciphers are designed by Yang et al. to improve the efficiency of Simon. All of them are alternative proposals for the lightweight block cipher program of NIST.

These ciphers cover both addition-based nonlinear structures (such as Speck and Chaskey) and S-box-based substitution-permutation structures (such as PRESENT and DES). They are widely adopted in the evaluation of neural distinguishers, providing a comprehensive benchmark for testing the accuracy and generalization of RLND.

2.3 Gohr's Neural Distinguisher

In [7], Gohr proposed a $\mathcal{ND}$ against round-reduced Speck 32/64, which aims to distinguishing two classes of ciphertext pairs, defined by formula (1):

$$Y(C_0, C_1) = \begin{cases} 1, & \text{if } P_0 \oplus P_1 = \alpha, \\ 0, & \text{if } P_0 \oplus P_1 \neq \alpha \end{cases} \tag{1}$$

where (C_0, C_1) is the ciphertext pair corresponding to the plaintext pair (P_0, P_1), and Y is the label of (C_0, C_1).

Gohr's network consists of four modules. The initial convolution model applied a convolution layer with 32 filters to the input ciphertext pair (C_L^r, C_R^r)

and $(C_L^{r'}, C_R^{r'})$, using the Relu function to extract features from the filters, followed by a 1D-CNN with kernel size of 3. The output is then processed by a convolution layer with 32 filters in the ResNet module. This produces a feature vector of size 32×16, *which is passed to three MLPs in the prediction model to obtain the final result.*

Therefore, Gohr's $\mathcal{ND}$ is a binary classifier for 64-bit inputs. A r-rounds distinguisher can be defined as follows:

$$\begin{cases} \Pr(Y^r = 1 \mid X^r) = p(F(X^r)) \\ X^r = (C^r, C^{r'}) \end{cases} \tag{2}$$

where, $F(X^r)$ denotes the features captured from the r-round ciphertext pair, and $p(\cdot)$ is a posterior probability function.

3 The Proposed Neural Distinguisher

In this section, we detail the proposed new neural distinguisher, RLND, and its ablation scheme, RMND. In the proposed RLND, we design a multi-scale convolutional kernel-based residual tower to extract features at multiple scales. Different LSTM layers with varying numbers of neurons are employed to construct diverse LSTM modules. The details are as follows.

3.1 Neural Distinguisher using spatial and Temporal Learning

The encryption process of block ciphers involves both spatial and temporal operations. However, current CNN-based distinguishers can only capture spatial features. LSTM, on the other hand, employs gating mechanisms such as the forget gate, input gate, and output gate, allowing it to selectively extract meaningful features while filtering out irrelevant or noisy information. This makes LSTM particularly suitable for the intricate structure of block cipher encryption. By leveraging these gating mechanisms, LSTM enables effective and selective feature extraction from the input data.

To address the limitations of existing methods, we propose a new neural distinguisher, RLND, which consists of a multi-scale convolution-based residual network and a multi-layer LSTM module. The architecture of the proposed network is illustrated in Fig. 1. Specifically, to effectively capture cryptographic features from multiple ciphertext pairs, RLND adopts the same two-dimensional convolution input format as proposed in [8].

In the proposed RLND, we design a new MSCK-ResNet module and a new ML-LSTM module. The details are as follows:

MSCK-ResNet module The proposed MSCK-ResNet consists of different ResNet blocks. The initial convolution kernel size of each MSCK-ResNet blocks is set to 1, with the remaining components unchanged. To capture features at

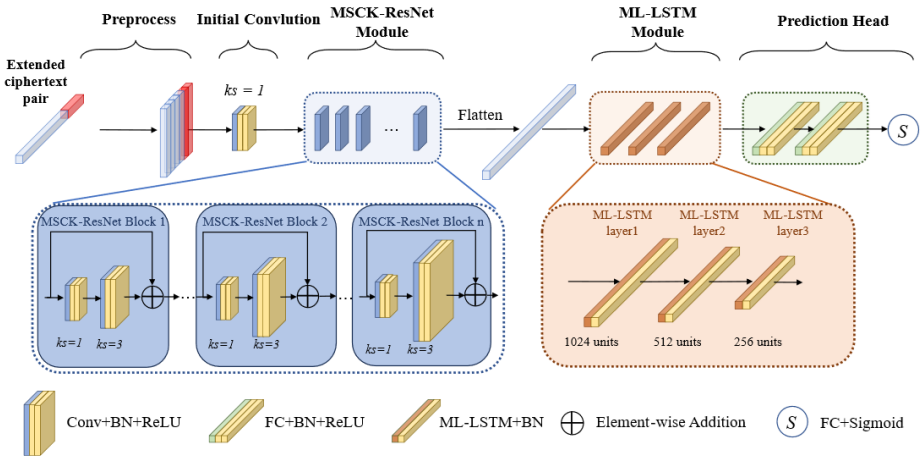

Fig. 1. The architecture of the proposed neural distinguisher RLND.

multiple scales, the convolution kernel size increases in steps of 2, as defined by the following formula:

$$k_{s_{dep}} = 3 + 2(dep - 1) \tag{3}$$

where *dep* represents the number of layers in the current residual block. The experiments show that though larger kernels contribute useful information for feature extraction, the performance gain plateaus when the kernel size reaches 5. This observation is consistent with the findings in prior research [8].

ML-LSTM module Considering that the capacity of learning of LSTM module is affected by the number of the neurons in hidden layer, we design a multi-layer LSTM module using different number of the neurons in each ML-LSTM layers. The detail is shown in Fig. 2.

For each ML-LSTM layer, the forget gate is defined by formula (4):

$$z_t^{(f)} = \sigma(U_f \cdot [r_{t-1}, s_t] + b_f) \tag{4}$$

Here, r_{t-1} represents the hidden state from the previous time step, which encodes the historical information. s_t is the input at the current time step, which contains the spatial features extracted by MSCK-ResNet module. The weight matrix U_f and bias vector b_f are learnable parameters. The input gate $z_t^{(i)}$ and the candidate memory $\tilde{m}_t$ are defined as the definition in [17]. The cell state is defined by the following formula:

$$c_t = z_t^{(f)} \cdot c_{t-1} + z_t^{(i)} \cdot \tilde{m}_t \tag{5}$$

where c_{t-1} is the previous cell state. The cell state ensures that only the most relevant information is retained over time. Each ML-LSTM layer is followed by

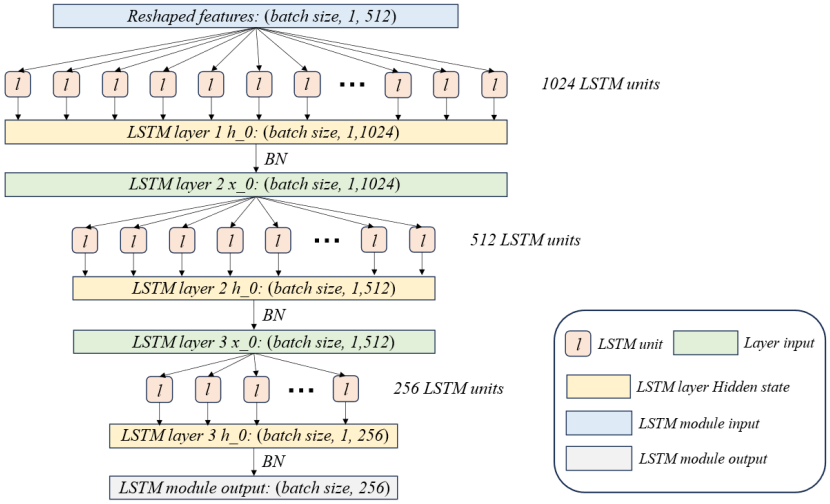

Fig. 2. The network structure of the ML-LSTM module.

a Batch Normalization (BN) to restrain the overfitting cause by the ML-LSTM layer. Additionally, a reshape layer is introduced to match the output of ResNet with the input format of ML-LSTM $(batchsize, t, len)$, where $t \in \{1, 2, 4, 8, 16\}$ and $len = 512/t$. The ML-LSTM module contains different LSTM layers is shown in Table 1.

Table 1. Four different LSTM modules

LSTM module	Number of layers	Number of neurons
ML-LSTM$_1$	3	1024, 512, 256
ML-LSTM$_2$	4	1024, 512, 256, 256
ML-LSTM$_3$	5	1024, 512, 256, 256, 128
ML-LSTM$_4$	5	1024, 512, 256, 256, 256

By using different LSTM modules in Table 1, we obtain four neural distinguishers, which are denoted as RLND$_1$, RLND$_2$, RLND$_3$, and RLND$_4$, respectively.

3.2 Neural Distinguisher based on MSC-MLP

The results in [10] show that the accuracy of $\mathcal{ND}$ based solely on LSTM is lower than that of $\mathcal{ND}$ based on MLP. To further investigate how the temporal features in the input sequence affect the model's prediction accuracy and validate the effectiveness of the RLND, we design a multi-scale convolutional MLP (MSC-MLP) and replace the ML-LSTM module with it. It contains three MLP layers,

in which each MLP layer uses different size of fully connected (FC) layers. We construct a neural distinguisher RMND by varying the number of MLP blocks and layers in Table 2.

Table 2. The MLP modules with different components

RMND	Number of MLP block	Number of MLP layer	Number of neurons
RMND_1	1	3	1024, 512, 256
RMND_2	3	3	1024, 512, 256
RMND_3	5	1	64
RMND_4	10	1	64
RMND_5	5	1	128
RMND_6	10	1	128

The RMND models based on these MLP blocks are denoted as RMND_i. The structure of RMND_1 is shown in Fig. 3, and the others can be derived using the same method, as presented in Table 2.

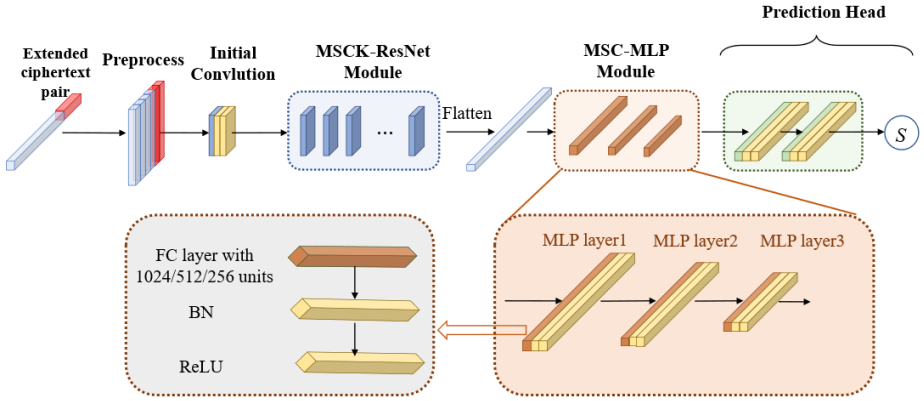

Fig. 3. The network structure of the RMND module.

4 Test of the Proposed Neural Distinguisher

To achieve the highest accuracy for the $\mathcal{ND}$ and validate the advantage of the proposed neural network module, we test the accuracy of the RLND with different time steps and the RMND with different MLP modules, using a single ciphertext pair from the 5-round reduced Speck 32/64 cipher.

4.1 Dataset generation and Simulation parameters

Consider k plaintext pairs with difference ΔP encrypted by the r rounds cipher E with k keys generated randomly to obtain k ciphertext pairs, where each ciphertext pair is denoted as $(C_L^r, C_R^r, C_L^{r'}, C_R^{r'})$. These k ciphertext pairs are regarded by a ciphertext group with size k, and labeled by $Y = 1$ as a positive sample. If the plaintext differences of k plaintext pairs are random, the corresponding ciphertext group is labeled as $Y = 0$, denoted as a negative sample. In this work, we extend the ciphertext pairs by adding the right part of the ciphertext pairs decrypted by one round with a random key, denoted by $(C_L^r, C_R^r, C_L^{r'}, C_R^{r'}, C_R^{r-1}, C_R^{r-1'})$.

Training parameters: the size of the training set is $N = 10^7/k$, the size of the testing set is $M = 10^6/k$, $k = 1$, and the number of epochs is 200. The i-th learning rate is defined as be

$$l_i = \frac{(n - i \bmod 10) \bmod (n+1)}{n_{\text{epoch}}}(\beta - \alpha) \tag{6}$$

where $\alpha = 10^{-4}$, $\beta = 2 \cdot 10^{-3}$, and the cycle period $n = 10$. L1 is used for batch normalization with a parameter of 10^{-5}. The batch size is set to 20000.

4.2 Accuracy of the Proposed RLNDs

Since the proposed RLNDs use different time steps and LSTM modules, we first test their accuracy with different time steps to determine the optimal value. The training procedure for the distinguisher for 8-rounds is as follows:

Step 1: The 7-round of neural distinguisher ND_7 is obtained using common methods, where the dataset is collected according to the method described in Section 4.1 with input difference $\delta = (0x0040, 0x0000)$.

Step 2: Then, 10^7 and 10^6 plaintext-pairs with input difference of $\delta = (0x8000, 0x840a)$ are encrypted 4 rounds to generate the training set and validation set, respectively. The distinguisher ND_7 is trained with a fixed learning rate of 10^{-4} with 10 epochs.

Step 3: Subsequently, the 10^9 and 10^8 plaintext-pairs with input difference of $\delta = (0x8000, 0x840a)$ are encrypted for 8 rounds to generate the training set and the validation set, respectively. The distinguisher ND_7 is retrained for three epochs with learning rates of 10^{-4}, 10^{-5}, and 10^{-6}, respectively. Finally, we obtain the final 8-round's neural distinguisher ND_8 for Speck32/64.

The accuracy of RLND_1 for different round-reduced Speck 32/64 is shown in Table 3.

Here, Acc_i is the accuracy of the i-th round-reduced Speck 32/64. From Table 3, we observe that the accuracy of RLND_1 reaches its highest value of 93.36% when the time step equals to 1. The accuracy decreases as the time step increases. Similar results are observed for the other RLNDs. Therefore, we simulate the accuracy of all proposed RLNDs with a time step of 1 and ciphertext-pair number of 1. The results are shown in Table 4.

Table 3. Accuracy of RLND_1 with different time step

Time step	1	2	4	8	16
Acc_5	**93.36%**	93.18%	93.17%	93.10%	93.10%
Acc_6	**79.09%**	78.76%	78.91%	78.92%	78.88%
Acc_7	**61.85%**	60.88%	61.80%	61.26%	61.06%
Acc_8	**51.54%**	51.00%	51.06%	50.99%	50.50%

Table 4. Accuracy of RLNDs with different LSTM module

$\mathcal{ND}$	RLND_1	RLND_2	RLND_3	RLND_4	Gohr [7]	DD1 [14]
Acc_5	**93.36%**	93.36%	93.36%	93.35%	92.9%	92.9%
Acc_6	**79.09%**	79.07%	79.06%	79.04%	78.9%	78.9%
Acc_7	**61.85%**	61.81%	61.84%	61.84%	61.7%	61.8%
Acc_8	**51.54%**	51.51%	51.52%	51.52%	51.47%	51.2%

Table 4 implies that the number of LSTM modules has little effect on the accuracy of the RLNDs. Therefore, we select RLND_1 as optimal $\mathcal{ND}$. Furthermore, we compare RLND_1 with $\mathcal{ND}$ based solely on LSTM and MLP, which is shown in Table 5.

Table 5. Compare RLND_1 with LSTM and MLP

Acc	$\mathcal{ND}$	$k=1$	$k=2$	$k=4$	$k=8$	$k=16$
Acc_5	LSTM [17]	88.65%	95.51%	98.58%	99.59%	99.16%
	MLP [16]	89.18%	95.48%	98.61%	99.61%	99.14%
	RLND_1	**93.36%**	**97.87%**	**99.08%**	**99.80%**	**99.87%**
Acc_6	LSTM [17]	69.76%	78.04%	83.77%	89.25%	85.59%
	MLP [16]	69.76%	78.05%	83.74%	89.21%	86.05%
	RLND_1	**79.09%**	**87.40%**	**93.15%**	**92.78%**	**90.44%**
Acc_7	LSTM [17]	52.19%	54.98%	50.28%	51.49%	50.60%
	MLP [16]	52.19%	52.86%	50.30%	51.34%	50.78%
	RLND_1	**61.85%**	**64.93%**	**65.18%**	**53.69%**	**50.47%**
Acc_8	LSTM [17]	×	×	×	×	×
	MLP [16]	×	×	×	×	×
	RLND_1	**51.54%**	**51.82%**	**52.38%**	**52.69%**	**53.04%**

The results above show that the accuracy of RLND_1 is higher than that of DD1 and the distinguishers based solely on LSTM and MLP for all tested numbers of input ciphertext pairs. For the 5 round-reduced Speck 32/64, its accuracy reaches the highest when $k = 16$, being 0.71% and 0.73% higher than the neural distinguishers using LSTM and MLP, respectively. For the 6 and 7 round-reduced Speck 32/64, its accuracy is highest when $k = 4$. Specifically, for the 6-round reduced Speck 32/64, it is about 1.38% and 1.41% higher than the other two neural distinguishers, respectively. For the 7 round-reduced Speck 32/64, it is about 14.9% and 14.88% higher than the other two neural distinguishers,

respectively. For the 8 round-reduced Speck 32/64, only the RLND_1 achieves the distinguishable accuracy. Therefore, the proposed RLND_1 demonstrates a significant advantage in accuracy compared to other neural distinguishers.

4.3 Ablation studies of the Proposed RLNDs

To validate the effectiveness of the ML-LSTM module, we conducted an ablation experiment. First, we tested the accuracy of the proposed RMNDs with different MLP modules to determine the optimal ablation scheme based on the MLP. The results are shown in Table 6.

Table 6. Accuracy of RMNDs with different MLP module

$\mathcal{ND}$	RMND_1	RMND_2	RMND_3	RMND_4	RMND_5	RMND_6
Acc_5	**93.25%**	93.18%	93.18%	93.20%	93.20%	93.10%
Acc_6	**79.01%**	77.71%	79.00%	78.97%	78.97%	77.71%
Acc_7	**61.68%**	59.74%	61.46%	61.70%	61.72%	59.57%
Acc_8	**51.54%**	50.05%	51.11%	51.41%	51.01%	50.10%

Table 6 shows that accuracy improves with an increase in the number of MLP layers in the MLP block. However, simply increasing the number of MLP blocks does not lead to higher accuracy. RMND_1 has the highest accuracy than other RMNDs for 5 and 6 round of Speck 32/64, while has a little lower than RMND_4 for 7 and 8 round of Speck 32/64. Therefore, it is used to implement the ablation test for the proposed RLND_1.

In the ablation test, we design four different ablation schemes, as shown in Table 7, denoted as OI_RLND_1, SR_RLND_1, NML_RLND_1 and RMND_1, respectively. Their difference compared to the RLND_1 are as follows: OI_RLND_1 uses the original cipher input, SR_RLND_1 uses the standard ResNet module, NML_RLND_1 does not include the ML-LSTM module, and RMND_1 replaces the ML-LSTM module with the MSC-MLP module.

Table 7. Detail of different ablation schemes

Component	Gohr	RLND_1	OI_RLND_1	SR_RLND_1	NML_RLND_1	RMND_1
Origin ciphertext	√		√			
Extended ciphertext		√		√	√	√
Preprocess	√	√	√	√	√	√
Initial Convolution	√	√	√	√	√	√
Original ResNet	√			√		
MSCK-ResNet module		√	√		√	√
ML-LSTM module		√	√			
MSC-MLP module						√
Prediction head	√	√	√	√	√	√

The accuracy of different ablation schemes using different ciphertext pairs for 5 round-reduced Speck 32/64 is compared in Table 8.

Table 8. Accuracy of different ablation schemes

	Accuracy					
k	Gohr	$RLND_1$	OI_RLND_1	SR_RLND_1	NML_RLND_1	$RMND_1$
1	92.90%	**93.36%**	93.26%	93.16%	93.20%	93.29%
2	97.68%	**98.05%**	97.85%	97.82%	97.74%	97.87%
4	99.08%	**99.65%**	99.19%	99.16%	99.25%	99.05%
8	99.74%	**99.95%**	99.78%	99.77%	99.76%	99.68%
16	98.47%	**99.99%**	98.89%	99.70%	99.80%	99.86%

From Table 7 and Table 8, we can see that the proposed neural distinguisher $RLND_1$ achieves the highest accuracy for all numbers of input ciphertext pairs. Additionally, its accuracy increases with the number of inputted ciphertext pairs. This indicates that the spatial and temporal features captured by the proposed neural network are effective in improving the neural distinguisher.

4.4 Comparison of Feature Selection Mechanisms in LSTM and MLP Modules

To investigate the effect of using either an LSTM module or an MLP module for feature processing after the same residual convolutional block, we compared RMND and RLND. They differ only in the intermediate processing module: RLND adopts a stack of three LSTM layers, whereas RMND replaces them with fully connected (MLP) layers of comparable scale. The effectiveness of the LSTM-based feature selection mechanism is evaluated from the following two perspectives.

First, we examine the models' sensitivity and selectivity with respect to input features through weight-based importance distributions. After training, the weight matrices of the final LSTM/MLP layer in each model are extracted. For each input feature, the absolute values of the corresponding connection weights are averaged, and features with importance values greater than 0.0005 are selected for analysis. The resulting feature-importance distributions for RMND and RLND are shown in Fig. 4 and Fig. 5, respectively. Although the MLP module activates a larger number of high-importance features (43 out of 512), the LSTM selects fewer features (31 out of 512), indicating stronger selectivity. Meanwhile, the average importance of the features selected by the LSTM is noticeably higher than that of the MLP (0.00177 vs. 0.00148). In addition, the histogram of the LSTM-based model exhibits a more concentrated and stable distribution, suggesting that the LSTM is more effective in extracting salient features.

Second, we further validate the rationality of the proposed LSTM module. As shown in Fig. 6, when the time step is reduced to 1, the LSTM module

degenerates into a single-step gated nonlinear transformation. Nevertheless, the experimental results show that its behavior remains clearly different from that of a standard MLP. This observation suggests that the LSTM architecture can still preserve beneficial gated feature interactions even in this degenerated setting, thereby enabling the model to emphasize informative features while suppressing irrelevant or noisy ones.

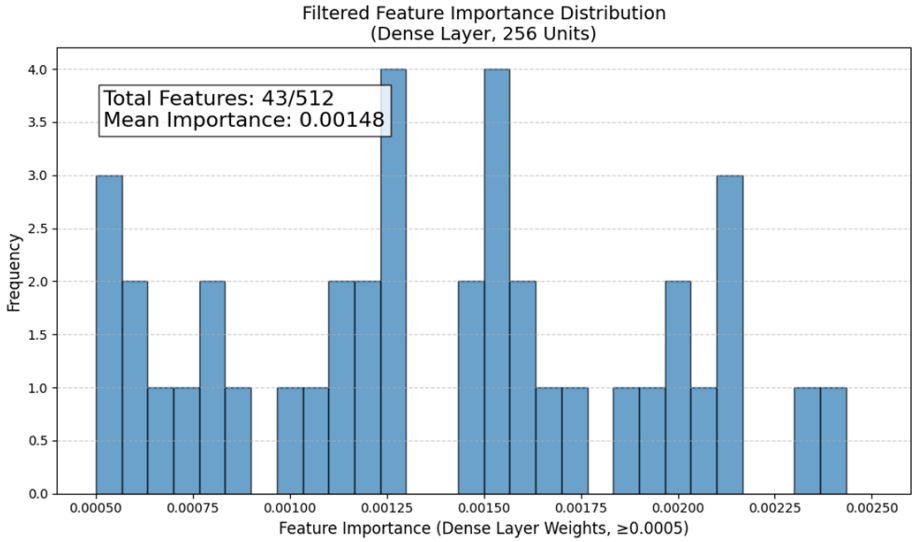

Fig. 4. Filtered feature importance distribution of the final dense layer in the RMND

4.5 Accuracy Test of Speck32/64

In the test, we use an input difference $\Delta P = (0x0040, 0x0000)$. The dataset is generated using the method in Section 4 by different k. The results are shown in Table 9.

Here, D represents the traditional distinguisher, and *Gohr22 refers to Gohr's network using knowledge distillation. From Table 9, we can see that compared with Gohr's $\mathcal{ND}$, the accuracy of RLND_1 using single ciphertext pairs improves by approximately 0.46%, 0.3%, and 0.15% for 5, 6, and 7 round-reduced Speck 32/64, respectively. When using multiple ciphertext pairs, the improvement compared to [8] is approximately 0.57% for the 5 round-reduced Speck 32/64 with $k = 4$. For the 6 and 7 round-reduced Speck 32/64 with $k = 8$, the improvement is about 2.67% and 5.55%, respectively. The accuracy of RLND_1 for 8 round-reduced Speck 32/64 is much higher than the distinguishable accuracy 51.0%. while other distinguishers are almost unable to make the distinction.

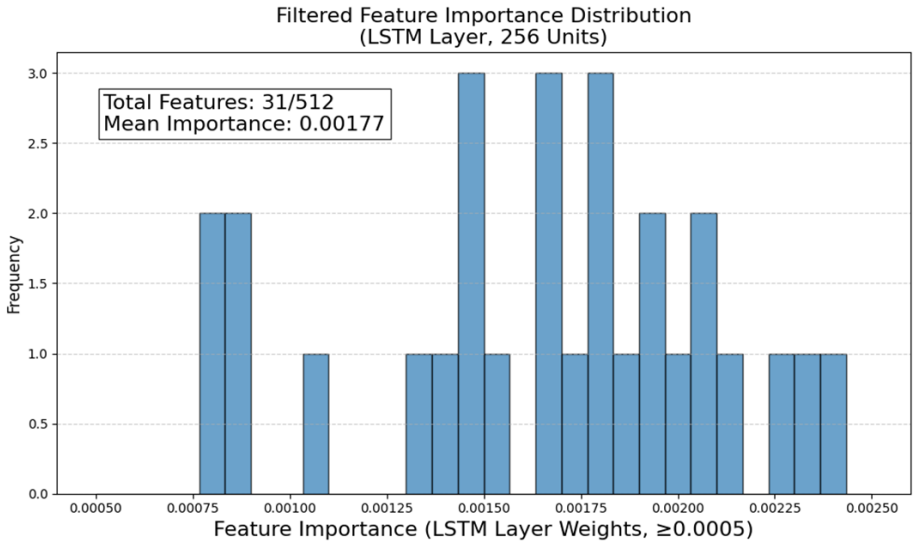

Fig. 5. Filtered feature importance distribution of the final dense layer in the RLND

Additionally, we compared RLND_1 using a single ciphertext-pair with the attention-based model B-C3-HSwish [20] under identical settings. As shown in Table 9, the proposed RLND_1 consistently achieves higher accuracy across all tested rounds.

To further demonstrate the practical impact of the proposed neural distinguisher, we evaluate how the improvement in accuracy translates into the success rate of key recovery attacks. As shown in Table 10, the higher accuracy of RLND_1 effectively enhances the practical key-recovery performance for round-reduced Speck 32/64, increasing the successful recovery rate from 52% (as achieved by Gohr's method) to 55%.

4.6 Accuracy Test of Simeck 32/64

In the experiment on 7, 8, and 9 round-reduced Simeck 32/64, we set the number of new residual block to 5 and $\Delta P = (0x0000, 0x0040)$. The batch size is set to 5000. The results are shown in Table 11:

As shown in Table 11, the accuracy of RLND_1 is 2.49%, 0.51% and 0.81% higher than [8] for the 7, 8 and 9 round-reduced Simeck 32/64 with single ciphertext pairs. For the 7, 8 and 9 round-reduced Simeck 32/64 with multiple ciphertext pairs, the highest accuracy of RLND_1 is 0.32%, 3.72% and 8.02% higher than [8], respectively.

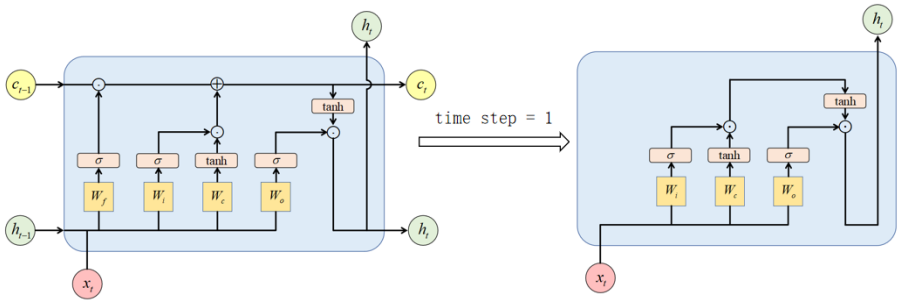

Fig. 6. The architecture of the degenerated LSTM at time step = 1.

4.7 Accuracy Test of Chaskey

In the experiment on 3-round and 4-round reduced Chaskey, we set the number of new residual block to 5 and $\Delta P = (0x8400, 0x0400, 0x0000, 0x0000)$. The batch size is set to 5000. The results are shown in Table 12:

As shown in Table 12, the accuracy of RLND_1 is 2.97% and 0.31% higher than [8] for the 3 and 4 round-reduced Chaskey with single ciphertext pairs. For the 3 and 4 round-reduced Chaskey with multiple ciphertext pairs, the highest accuracy of RLND_1 is 4.21% and 6.28% higher than [8], respectively.

4.8 Accuracy Test of PRESENT64/80

The experiment on 5-round, 6-round and 7-round reduced PRESENT 64/80 is tested in this section. The number of new residual blocks is set to 5, and the batch size is 5000. The input difference is $\Delta P = (0x0000, 0x0000, 0x0000, 0x0009)$ [29]. The results are shown in Table 13.

The accuracy of RLND_1 is approximately 0.4% and 1.24% higher than [8] for 6 and 7 round-reduced PRESENT 64/80 with single ciphertext-pair. For 6 and 7 round-reduced PRESENT 64/80 with multiple ciphertext-pairs, the highest accuracy of RLND_1 is 12.97% and 9.93% higher than [8], respectively. Furthermore, we compared the proposed model with recent Transformer-based architectures LbEC and TbEC [19] in Table 13. The results demonstrate that RLND_1 exhibits stronger performance, achieving higher distinguishing accuracy than these Transformer-based models.

4.9 Accuracy Test of DES

The experiment on 5-round and 6-round reduced DES is tested in this section. The number of new residual blocks is set to 10, and the batch size is 5000. The input difference is $\Delta P = (0x40080000, 0x04000000)$. The results are shown in Table 14. The accuracy of RLND_1 is approximately 3.23% and 1.05% higher than [8] for 5 and 6 round-reduced DES with single ciphertext pairs. For 5 and

Table 9. Accuracy of different $\mathcal{ND}$ for Speck 32/64

r	$\mathcal{ND}$	Accuracy				
		k=1	k=2	k=4	k=8	k=16
5	D [7]	91.1%	×	×	×	×
	Gohr22 [10]	92.9%	×	×	×	×
	CY[8]	92.9%	97.68%	99.08%	99.91%	99.98%
	B-C3-HSwish [20]	93.20%	×	×	×	×
	RLND$_1$	**93.36%**	**98.05%**	**99.65%**	**99.95%**	**99.99%**
6	D [7]	75.8%	×	×	×	×
	Gohr22 [10]	78.8%	×	×	×	×
	CY[8]	78.8%	86.13%	93.10%	95.62%	98.18%
	B-C3-HSwish [20]	79.00%	×	×	×	×
	RLND$_1$	**79.09%**	**87.64%**	**93.95%**	**98.29%**	**99.29%**
7	D [7]	59.1%	×	×	×	×
	*Gohr22 [10]	61.7%	×	×	×	×
	CY[8]	61.7%	63.93%	68.61%	70.74%	66.94%
	B-C3-HSwish [20]	61.70%	×	×	×	×
	RLND$_1$	**61.85%**	**66.01%**	**69.51%**	**76.29%**	**67.24%**
8	D [7]	×	×	×	×	×
	*Gohr22 [10]	51.40%	×	×	×	×
	CY[8]	×	×	×	×	×
	B-C3-HSwish [20]	×	×	×	×	×
	RLND$_1$	**51.54%**	**51.82%**	**52.38%**	**52.69%**	**53.04%**

Table 10. Comparison of key recovery successful rate

$\mathcal{ND}$	Successful rate
Gohr's Distinguisher [7]	52%
RLND	55%

6 round-reduced DES with multiple ciphertext pairs, the highest accuracy of RLND$_1$ is 3.14% and 2.76% higher than [8], respectively.

From the above, we can see that the proposed neural distinguisher RLND$_1$ achieves higher accuracy for both single and multiple ciphertext pairs, demonstrating strong generalization ability. Typically, the neural distinguisher with the highest accuracy is used to recover the keys. Therefore, the improvement of the proposed RLND$_1$ is 0.46%, 0.3%, and 0.15% for different round-reduced Speck 32/64. Its advantage is 0.32%, 3.72% and 8.02% for different round-reduced Simeck 32/64, 2.97% and 0.31% for different round-reduced Chaskey, 6.67% and 0.18% for different round-reduced PRESENT 64/80, 3.14% and 2.76% for different round-reduced DES. These results highlight the significant advantage of the proposed neural distinguisher over others.

Table 11. Accuracy of different $\mathcal{ND}$ for Simeck 32/64

r	$\mathcal{ND}$	Accuracy				
		k=1	k=2	k=4	k=8	k=16
7	CY[8]	88.23%	99.40%	99.91%	99.99%	99.99%
	RLND$_1$	**90.72%**	**99.72%**	**99.93%**	**99.99%**	**99.99%**
8	CY[8]	90.21%	92.29%	97.70%	97.97%	98.06%
	RLND$_1$	**90.72%**	**96.01%**	**98.91%**	**99.04%**	**98.29%**
9	CY[8]	69.91%	73.81%	74.17%	71.44%	74.33%
	RLND$_1$	**70.72%**	**77.32%**	**76.57%**	**77.43%**	**82.35%**

Table 12. Accuracy of different $\mathcal{ND}$ for Chaskey

r	$\mathcal{ND}$	Accuracy				
		k=1	k=2	k=4	k=8	k=16
3	CY[8]	84.51%	89.58%	95.83%	98.87%	95.70%
	RLND$_1$	**87.48%**	**90.79%**	**96.87%**	**99.01%**	**99.91%**
4	CY[8]	61.61%	65.89%	69.81%	76.03%	77.12%
	RLND$_1$	**61.92%**	**66.43%**	**73.94%**	**77.73%**	**83.40%**

Table 13. Accuracy of different $\mathcal{ND}$ for PRESENT 64/80

r	$\mathcal{ND}$	Accuracy				
		k=1	k=2	k=4	k=8	k=16
5	LbEC [19]	84.65%	×	×	×	×
	TbEC [19]	85.29%	×	×	×	×
	RLND$_1$	**85.35%**	×	×	×	×
6	CY[8]	65.84%	71.98%	79.53%	83.08%	82.59%
	LbEC [19]	65.83%	×	×	×	×
	TbEC [19]	64.91%	×	×	×	×
	RLND$_1$	**66.24%**	**72.57%**	**81.68%**	**89.44%**	**95.56%**
7	CY[8]	54.86%	55.03%	58.53%	57.86%	58.18%
	LbEC [19]	56.24%	×	×	×	×
	TbEC [19]	56.30%	×	×	×	×
	RLND$_1$	**57.08%**	**58.08%**	**60.63%**	**63.76%**	**68.11%**

Table 14. Accuracy of different $\mathcal{ND}$ for DES

r	$\mathcal{ND}$	Accuracy				
		k=1	k=2	k=4	k=8	k=16
5	CY[8]	62.61%	72.09%	83.82%	93.18%	95.85%
	RLND$_1$	**66.24%**	**72.32%**	**84.00%**	**94.32%**	**98.62%**
6	CY[8]	54.86%	55.03%	58.53%	57.86%	58.18%
	RLND$_1$	**55.91%**	**57.79%**	**59.65%**	**58.02%**	**59.56%**

5 Conclusion

To enhance the accuracy of neural distinguishers for block ciphers, we proposed novel neural distinguisher structures and tested their performance on different block ciphers. In this work, a residual tower with a multi-scale convolution kernel was used in the ResNet block of RLND to capture spatial features, while LSTM models with varying numbers of blocks and neurons were employed to extract temporal features. To further validate the advantages of RLND, we also designed the MSC-MLP module with different configurations of fully connected layers and MLP blocks, which served as an ablation scheme for the ML-LSTM module in RLND.

We then determined the optimal network structure and parameters for RLND through simulations, testing the accuracy of the best RLND on various block ciphers. The results demonstrate that the proposed RLND outperforms other neural distinguishers in both accuracy and generalization.

Future work will focus on further optimizing the structure and parameters of the LSTM to improve accuracy and exploring the features extracted by the LSTM to explain why RLND outperforms other neural distinguishers.

Acknowledgments. This work was supported by the following projects and foundations: 2024 Shaanxi Province key research and development plan project (No. 2024GX-ZDCYL-01-13), 2025 Shaanxi Province key research and development plan project (No. 2025GH-YBXM-025), the Technological Innovation Guidance Program of the Xizang Autonomous Region Project (No. XZ202601JX0 002).

References

1. Biham E. and Shamir A.: Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, vol. 4, no. 1, pp. 3–72 (Jan. 1991).
2. Matsui M.: On correlation between the order of S-boxes and the strength of DES. In: Lecture notes in computer science, pp. 366-–375 (Jan. 1995).
3. Sasaki Y.,Todo Y.: New algorithm for modeling S-Box in MILP based differential and division trail search. In: Lecture notes in computer science, pp. 150—165 (Oct. 2017).
4. Zhou, C., Zhang, W., Ding, T., Xiang, Z.: Improving the MILP-based Security Evaluation Algorithm against Differential/Linear Cryptanalysis Using A Divide-and-Conquer Approach. IACR Transactions on Symmetric Cryptology, pp. 438–469 (Jan. 2020).
5. Sun, S. et al.: Analysis of AES, SKINNY, and Others with Constraint Programming. IACR Transactions on Symmetric Cryptology, pp. 281–306 (Mar. 2017).
6. Gérault, D., Lafourcade, P., Minier, M., Solnon, C.: Revisiting AES related-key differential attacks with constraint programming. Information Processing Letters, vol. 139, pp. 24–29 (Nov. 2018).

7. Gohr, A.: Improving attacks on Round-Reduced Speck32/64 using deep learning. In: Lecture Notes in Computer Science, pp. 150–179 (Aug. 2019).
8. Chen, Y., Shen, Y., Yu, H., Yuan, S.: A new neural distinguisher considering features derived from multiple ciphertext pairs. The Computer Journal, vol. 66, no. 6, pp. 1419–1433 (Mar. 2022).
9. Hou, Z., Ren, J., Chen, S.: Improve neural distinguishers of SIMON and SPECK. Security and Communication Networks, vol. 2021, pp. 1–11 (Dec. 2021).
10. Gohr, A., Leander, G., Neumann, P.: An assessment of differential-neural distinguishers. Cryptology ePrint Archive (2022).
11. Zhang, L., Wang, Z.: Improving differential-neural cryptanalysis. Cryptology ePrint Archive (2022).
12. Liu, J., Ren, J., Chen, S., Li, M.: Improved neural distinguishers with multi-round and multi-splicing construction. Journal of Information Security and Applications, vol. 74, p. 103461 (May. 2023).
13. Bao, Z., Guo, J., Liu, M., Ma, L., Tu, Y.: Enhancing Differential-Neural cryptanalysis. In: Lecture Notes in Computer Science, pp. 318–347 (Jan. 2022).
14. Baksi, A., Breier, J., Dong, X., Yi, C.: Machine learning assisted differential distinguishers for lightweight ciphers. In: Classical and Physical Security of Symmetric Key Cryptographic Algorithms, pp. 141–162 (2020).
15. Wang, G., Wang, G., He, Y.: Improved machine learning assisted (Related-Key) differential distinguishers for lightweight ciphers. In: 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE (Oct. 2021).
16. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature, vol. 323, no. 6088, pp. 533–536 (Oct. 1986).
17. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation, vol. 9, no. 8, pp. 1735–1780 (Nov. 1997).
18. Alzubaidi, L. et al.: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, vol. 8, no. 1 (Mar. 2021).
19. Bose, A., Pal, D., Roy Chowdhury, D.: Deep learning-based differential distinguishers for cryptographic sequences. In: International Conference on Cryptology in India, pp. 114–133. Springer Nature Switzerland, Cham (2024).
20. Jiang, X., Li, M., Kaiyrbek, M., et al.: Enhanced Neural Differential Distinguisher for Speck32/64 Using Attention Mechanisms and Multi Ciphertext Inputs. Informatica, vol. 49, no. 19 (2025).
21. Zhang L., Wang Z.: Improving differential-neural distinguisher model for DES, Chaskey, and PRESENT. arXiv preprint arXiv:2204.06341 (Apr. 2022).
22. Mishra G., Pal S. K., Krishna Murthy S., et al.: Deep learning-based differential distinguisher for lightweight ciphers GIFT-64 and PRIDE. In: Machine Intelligence and Smart Systems: Proceedings of MISS 2021, pp. 245–257 (May. 2022).
23. Rajan R., Roy R. K., Sen D., et al.: Deep learning-based differential distinguisher for lightweight cipher GIFT-COFB. In: Machine intelligence and smart systems: proceedings of MISS 2021, pp. 397–406 (May. 2022).
24. Kimura H., Emura K., Isobe T., et al.: Output prediction attacks on block ciphers using deep learning. In: International Conference on Applied Cryptography and Network Security, pp. 248–276 (Sep, 2022).
25. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK lightweight block ciphers. In: Proceedings of the 52nd Annual Design Automation Conference, pp. 1–5 (Jun. 2015).

26. Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., Verbauwhede, I.: Chaskey: an efficient MAC algorithm for 32-bit microcontrollers. In: Lecture Notes in Computer Science, pp. 306–323 (Nov. 2014).
27. Bogdanov, A. et al.: PRESENT: an Ultra-Lightweight Block Cipher. In: Lecture Notes in Computer Science, pp. 450–466 (2007).
28. Wang, M.: Differential cryptanalysis of Reduced-Round PRESENT. In: Springer eBooks, pp. 40–49 (2008).
29. Yang G., Zhu B., Suder V., et al.: The simeck family of lightweight block ciphers. In: International workshop on cryptographic hardware and embedded systems, pp. 307–329 (Jan. 2015).