Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

An efficient differential analysis method based on deep learning

Ying Huang, Lang Li *, Ying Guo, Yu Ou, Xiantong Huang

College of Computer Science and Technology, Hengyang Normal University, Hengyang, 421002, China
Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang Normal University, Hengyang, 421002, China

A R T I C L E I N F O

Keywords:
Differential analysis
Deep learning
MLP
ResNet

A B S T R A C T

Differential analysis is a vital tool for evaluating the security of cryptography algorithms. There has been a growing interest in the differential distinguisher based on deep learning. Various neural network models have been created to increase the accuracy of distinguishing between ciphertext and random sequences. However, few studies have focused on differential analysis at the design stage of cryptographic algorithms. This paper presents an appropriate model for differential analysis of block ciphers. The model is similar to multilayer perceptron (MLP) models in simplicity and clarity. It also introduces a shortcut connection that enables one to learn more information about the differential analysis dataset. The model is used to predict the minimum number of active S-boxes (AS), linking differential analysis results to algorithm features. This model and two classical neural network models are compared under fair experimental conditions. The findings indicate that our model predicts the AS values with an accuracy of 97%. It can effectively predict the results of differential analysis. In addition, the differential analysis dataset is constructed for SPN structure cryptographic algorithms. It can be used for further differential analysis studies based on deep learning.

1. Introduction

There are wide applications for wireless sensor networks and radio frequency identification technologies. The security of sensitive data has become increasingly important on resource-constrained devices. These devices pursue low consumption and high efficiency. It inevitably leads to the reduction of security. Lightweight block cipher plays a critical role in maintaining encryption-sensitive data. The security of lightweight block ciphers mainly depends on the ability to resist known attacks. Since differential analysis is being used to analyze DES, the method has been widely used for various block ciphers. Nowadays, differential analysis is still one of the most popular cryptanalysis methods. Differential analysis uses statistical methods to analyze the security of cryptographic algorithms. The most important in differential analysis is to find a high probability differential characteristic. The difference characteristic is the possible propagation trail, which is obtained through the round function of block cipher. The difference characteristic consists of the difference path and its probability, where the difference path consists of an input difference and an output difference. The details are shown in Definition 1. Researchers can get a differential distinguisher through the high probability differential characteristic. The differential distinguisher that can distinguish longer rounds is an excellent aid to the analysis of block ciphers. In traditional differential analysis, the differential distinguisher relies more on possible defects in the structure of the algorithm itself, such as the round function. The process of differential analysis relies heavily on manual derivation. It significantly slows down the process of cryptanalysis. In recent years, the use of automatic search techniques to find differential distinguishers has gradually become the mainstream.

Deep learning plays an essential role as one of the most intelligent technologies. It tries to discover some concealed rules in dataset information. Researchers in cryptography have focused on side-channel attack [1] and algorithm component design [2] using deep learning. There have only been several studies that have applied deep learning to traditional cryptanalysis. It has not been explored much until the research of Gohr [3]. Gohr proposed a 9-round differential distinguisher for SPECK [4] using ResNet [5]. This demonstrates that deep learning can produce very powerful cryptographic distinguishers. The researcher also developed a strategy for key recovery attacks for deep learning distinguishers. It is a significant foundation for the subsequent development of deep learning distinguisher. Following the important research published by Gohr, the current research works can be described from two perspectives.

(1) Researchers apply the deep learning distinguisher to several cryptographic algorithms to validate the efficiency. Jain [6] used the multilayer perceptron network to distinguish between random sequences and 3–6 rounds of PRESENT [7] ciphertexts. Bellini et al. [8] proposed two deep learning distinguishers for TEA [9] and RAIDEN

* Corresponding author at: College of Computer Science and Technology, Hengyang Normal University, Hengyang, 421002, China.
E-mail address: lilang@hynu.edu.cn (L. Li).

https://doi.org/10.1016/j.comnet.2023.109622
Received 13 June 2022; Received in revised form 29 January 2023; Accepted 6 February 2023
Available online 9 February 2023

Table 1
The comparison of previous works.

Paper	Summary
Gohr [3]	The effectiveness of neural networks in differential analysis.
Jain et al. [6]	The distinguishing effect of MLP model.
Bellini et al. [8]	Comparison of neural network distinguisher and traditional distinguisher.
Tian et al. [11]	The distinguishing effect of different deep learning models on SIMON32/64.
Wang et al. [12]	The distinguishing effect of FNN and CNN models.
Baksi et al. [13]	The effectiveness of deep learning distinguishers in non-Markovian ciphers.
Hou et al. [14]	The distinguisher accuracy for different input difference patterns.
Yadav et al. [15]	An extended distinguisher framework using machine learning.
Chen et al. [16]	The distinguisher accuracy for multiple ciphertext pairs patterns.
Benamira et al. [17]	The detailed analysis and a thorough explanation of Gohr's work.
Guo et al. [18]	The extract method for more accurate features in neural distinguisher.
Gohr et al. [19]	The automatic optimization of neural differential distinguishers.

[10]. It is shown that the deep learning distinguisher is significantly better than the classical distinguisher. Tian achieved 7, 8, and 9 rounds of deep learning distinguishers for SIMON32/64 [11]. This study also successfully completed a key recovery attack on 15 rounds of SIMON32/64. Wang et al. designed a 6- to 9- round differential distinguisher applied to SIMON32/64 [12]. The feedforward neural network and convolutional neural network can distinguish 6 and 7 rounds of SIMON32/64 ciphertext pairs with high probability. Baksi et al. show that deep learning can also construct distinguishers for non-Markovian ciphers [13]. It points out that deep learning can reduce the distinguisher complexity. Hou et al. explored the effect of different input difference patterns on the accuracy of deep learning distinguisher [14]. It helps to find the appropriate input difference for training a higher accuracy distinguisher. This work also proposed an 11-round key recovery attack against SIMON32 with over 90% success rate. Yadav et al. developed an extension framework for classical differential distinguishers using machine learning [15]. This distinguisher can distinguish more rounds with lower data complexity. Chen et al. proposed a new neural distinguisher that uses the features of multiple ciphertext pairs [16].

(2) Researchers focus on the interpretability of deep learning distinguishers and study if the accuracy is affected by certain cryptographic properties. Benamira et al. provided a full analysis and comprehensive description of Gohr's findings [17]. It is shown that the deep learning distinguisher typically relies on the difference distribution of ciphertext pairs, but also on the difference distribution of the penultimate and antepenultimate rounds. Guo et al. designed a key recovery attack strategy by introducing the concept of neutral bits [18]. It is more suitable for neural network distinguishers. This research also studied ways to extract features that are more accurate and offered a trade-off strategy for modifying parameters. 2022, Gohr's new work provided an idea for automatically optimizing deep learning distinguisher [19]. This work also provided practical optimization strategies for different ciphers. This work also pointed out that the accuracy is closely related to the mean absolute distance between the output difference distribution and the uniform one.

These studies further expand the application of deep learning in differential analysis. Table 1 shows the comparison table of current differential analysis literature. However, little research has looked at the relationship between deep learning and the security of cryptographic algorithms throughout the design process. It means that few studies have used deep learning to predict the minimum number of active S-boxes (AS) of an algorithm. The traditional differential analysis methods are divided into the high probability search algorithms [20] and automatic analysis method [21]. The high probability search algorithm has a high accuracy rate while it takes a long time. The automated analysis approach requires the assistance of mathematical tools. The cost and time required are dictated by the size, scale, and number of iterations of cryptographic algorithms. The two methods both need to construct differential analysis model for specific cryptographic algorithms. The SPN structure is one of the classical structures which include AES [22], QTL [23], and SKINNY [24] et al. algorithms. This paper proposes a deep learning model named DABC model for classical SPN structure algorithms. DABC model combines the simplicity of fully connected layer with the learning ability of shortcut connection module. It allows DABC model to learn more potential patterns of differential analysis dataset.

The rest of the paper is organized as follows. Section 2 introduces related principles, including differential analysis, mixed-integer linear programming (MILP) and multilayer perceptron and ResNet network. Section 3 discusses the benefits and limitations of traditional differential analysis and gives a deep learning model applied to differential analysis. Section 4 lists the differential analysis dataset, the network model hyperparameters, the training process, and the evaluation metrics. Section 5 gives the AS prediction result of several neural networks and the number of hidden layers of the DABC model and the model complexity and training time. Section 6 outlines the main contributions of this paper and considers future directions.

2. Related principles

2.1. Differential analysis

Differential analysis is a kind of selective plaintext attack in which cryptography attacker tries to get the certain key. The goal of differential analysis is to obtain maximum possible keys by obtained differential characteristics.

Definition 1. The difference path $(\Delta X, \Delta Y)$ is written as $\Delta X \rightarrow \Delta Y$ in n bit block cipher C, the probability of differential characteristic is defined in Eq. (1) as follows.

$$Pr_C(\Delta X \rightarrow \Delta Y) = \frac{|X|C(X_1) \bigoplus C(X_2) = \Delta Y, X \in 0, 1^n|}{2^n} \quad (1)$$

A difference characteristic requires the probability of a plaintext difference, a ciphertext difference, and the difference pair consisting of the plaintext difference and the output difference. The plaintext difference $\Delta X = X_1 \oplus X_2$, when X_1 and X_2 are the plaintext pairs. Similarly, the difference ciphertext $\Delta Y = Y_1 \oplus Y_2$, when Y_1 and Y_2 are the ciphertext pairs. $Pr_C(\Delta X \rightarrow \Delta Y)$ is the probability that when the difference of an input pair is ΔX, the difference of the output of its corresponding output pair is ΔY. The difference probability of random plaintext difference ΔX to corresponding ciphertext difference ΔY is 2^{-n}. If finding the probability of plaintext–ciphertext differential pairs $(X' \rightarrow Y')$ is much smaller than 2^{-n}, cryptography attacker can obtain large amounts of plaintext–ciphertext pairs $Prc(X_1, X_2, Y_1, Y_2)$. Then, attackers calculate the likelihood of candidate keys using these plaintext–ciphertext pairs obtained previously. It is believed that cryptography attackers can obtain sensitive information about specific keys.

This approach is used to evaluate the security of lightweight block ciphers, such as LBlock [25], Shadow [26], and GIFT [27]. Generally, the maximum differential probability of 4 bits S-box is 2^{-2}. The maximum differential probability is an index to measure the resisted ability for differential analysis. It can be said that block cipher can resist differential analysis when the maximum differential probability satisfies $Pr_C(\Delta X \rightarrow \Delta Y) = 2^{n\times(-2)} \leq 2^{AS\times(-2)}$.

2.2. Mixed Integer Linear Programming

Mixed Integer Linear Programming (MILP) is a mathematical computational analysis method for optimization problems. Mouha et al. used MILP to evaluate the security of the Enocoro stream algorithm. Sun et al. further improved the application of MILP in the differential analysis of block cipher [28]. Since 2014, MILP has been widely used in security analysis methods for block ciphers, such as differential analysis [29], linear analysis [30], and conditional cube attack [31]. In this paper, MILP is used to collect differential analysis datasets. The algorithm components are constrained as shown below.

Definition 2. Considering n bits string $\Delta = (\Delta_0, \Delta_1, \dots, \Delta_{n-1})$ and the difference variable $x = (x_0, x_1, \dots, x_{n-1})$. The difference value $\Delta_i = 0$, when the difference variable $x_i = 0$. The difference value $\Delta_i \neq 0$, and the difference variable $x_i = 1$. This definition is shown in Eq. (2) as follows.

$$x_i = \begin{cases} 0 & \text{if } \Delta_i = 0 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

(1) Constraints of Nonlinear Transformations. The input and output vectors of S-box are defined as $\Delta a = (\Delta a_0, \Delta a_1, \dots, \Delta a_{n-1})$, $\Delta b = (\Delta b_0, \Delta b_1, \dots, \Delta b_{n-1})$. The value of A_t is used to evaluate whether S-box is active. The inequality constraints are shown in Eq. (3) as follows.

$$\begin{cases} \Delta a_0 - A_t \leq 0 \\ \Delta a_1 - A_t \leq 0 \\ \Delta a_2 - A_t \leq 0 \\ \Delta a_3 - A_t \leq 0 \end{cases} \tag{3}$$

The four input differences Δa_0, Δa_1, Δa_2, and Δa_3 exist in a non-zero state when $A_t = 1$. The inequality constraint is shown in Eq. (4) as follows.

$$\Delta a_0 + \Delta a_1 + \Delta a_2 + \Delta a_3 - A_t \geq 0 \tag{4}$$

(2) Constraints of Linear Transformations. The number of difference branch B^* is the output branch of the linear transformation operation L. The input differential variables in the linear transformation are $(x^L_{i_0}, x^L_{i_1}, \dots, x^L_{i_{n-1}})$, then $(x^L_{o_0}, x^L_{o_1}, \dots, x^L_{o_{n-1}})$ are the output differential variables. d^* is a dummy variable that takes zero or one. $x^L_{i_0}, x^L_{i_1}, \dots, x^L_{i_{n-1}}, x^L_{o_0}, x^L_{o_1}, \dots, x^L_{o_{n-1}}$ are zero, d^* takes zero, and vice versa, d^* takes one. The inequality constraints of linear transformation are expressed as follows in Eq. (5).

$$\begin{cases} x^L_{i_0} + x^L_{i_1} + \cdots + x^L_{i_{n-1}} + x^L_{o_0} + x^L_{o_1} + \cdots + x^L_{o_{n-1}} \geq B^* d^* \\ d^* \geq x^L_{i_0}, \dots, d^* \geq x^L_{i_{n-1}}, d^* \geq x^L_{o_0}, \dots, d^* \geq x^L_{o_{n-1}} \end{cases} \tag{5}$$

2.3. Multilayer Perception

Multilayer Perceptron (MLP) is a kind of feedforward neural network [32]. It comprises three layers: an input layer, an output layer, and possible hidden layers. MLP models are some directed graphs which are consisted of multiple nodes. Each node is a neuron (or processing unit) with nonlinear activation function excluding input nodes. The predicted result is given in the output layer. The sizes of neuron parameters may be manually adjusted to get task-appropriate results. Hence, MLP can handle many non-convex and non-linear problems. Its advantage is the ability to solve complex problems without a lot of time.

MLP model is stated as $y = F(x, \theta)$, where x is the input, θ is composed of weights ω and biases b. MLP try to use algorithms to change the weights and eliminate bias during training. The simplest MLP model has only one hidden layer in extreme cases. The output can be expressed as in Eq. (6).

$$y = \sum_{j=1}^{3} \left(\omega_j \left(x_1 + x_2\right) + b_j\right) \tag{6}$$

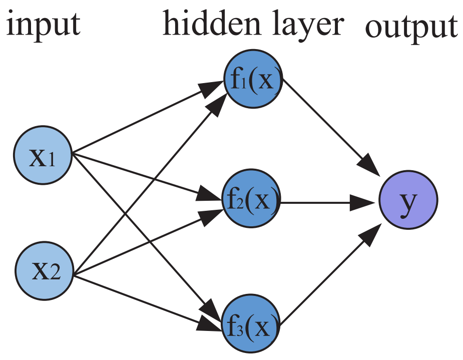


Fig. 1. The simplest MLP model of one hidden layer.

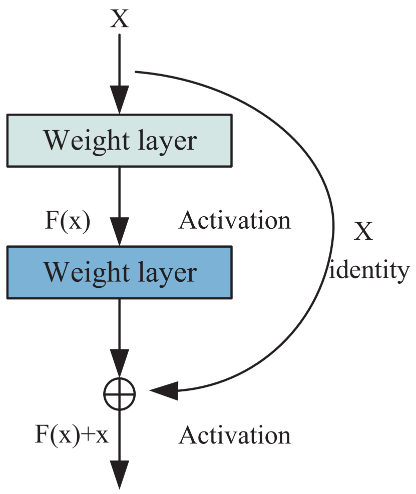


Fig. 2. The shortcut connection module.

where ω_j is the weight corresponding to the output of the $f_j(x)$ neuron at the preceding layer and b_j is the bias. There are two weights in Eq. (6), ω_j and b_j, which represent the output of the all neurons and their respective biases. The simplest MLP model of on hidden layer can be seen in Fig. 1.

2.4. ResNet

Convolutional neural networks (CNN) can simplify complex problems, reduce the dimension of many parameters into a small number of parameters [33]. Therefore, convolutional neural networks are suitable for many fields such as computer vision, object detection, and cryptanalysis. ResNet is a classical convolutional neural network that uses the shortcut connection module to enhance model efficiency. ResNet is ability to learn conventional information by adding some shortcut connection modules. Therefore, the learning objectives and difficulties are simplified. The shortcut connection module is shown in Fig. 2. Where x is the input and $f(x)$ is the output of the first network layer. x and $f(x)$ are directly retained in subsequent network layers. It allows subsequent neural networks to learn new features from the original input x.

To better evaluate the effectiveness of the DABC model in predicting AS, we introduce the ResNet model to compare. ResNet is widely used in differential cryptanalysis based on deep learning. The prediction AS task is a kind of differential analysis from the cryptographer's perspective. There are certain values in exploring the effectiveness of ResNet in predicting AS task for cryptanalysis. Therefore, the Res18 network is used to perform this task.

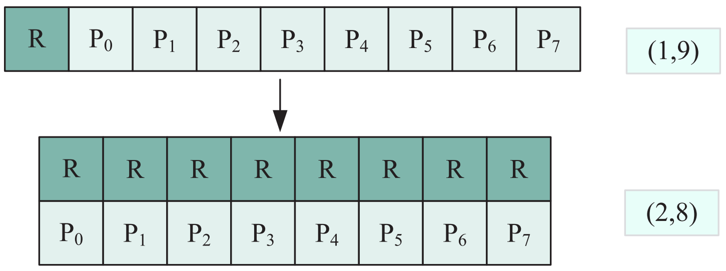


Fig. 3. Data processing of DABC model with 8 branch P-permutation, where R denotes Round and $P_i (i = 0, 1, \dots, 7)$ denotes P-permutation.

3. Deep learning model for differential analysis

3.1. Traditional differential analysis

Differential analysis is a crucial analysis method for the block cipher. The paper studies the propagation pattern of the input plaintext pairs in encryption process. Traditional differential analysis methods can be divided into two methods:

(1) The high probability search algorithms build differential analysis models for different algorithms. These algorithms calculate the estimated values for each round as branch definition conditions. It derives the maximum differential probability from the actual values calculated at the end. The actual differential probabilities for the nth round need to rely on the actual maximum differential probabilities obtained in the previous n-1 rounds. However, these algorithms have a low ratio of correctness to time.

(2) Automatic search methods rely on mathematical tools to establish linear constraints and filter branches. The linear and nonlinear components of cryptographic algorithm are transformed into a set of constraint inequalities. This approach enables the core part of algorithms to be extracted.

The traditional differential analysis methods finds optimal solutions in finite time. These methods come at the expense of accuracy. However, traditional differential analysis have certain limitations: (1) Insufficient applicability of traditional methods. Traditional differential analysis needs to analyze specific cryptographic algorithms. It need to build specific search algorithm or establish automated analysis model. (2) Long time for traditional differential analysis. Therefore, it is worth finding an efficient differential analysis using deep learning.

3.2. Deep learning framework for differential analysis

This paper proposes an efficient differential analysis method based on deep learning, namely the DABC model. The goal of our study is to predict AS. This metric can calculate the differential probability Pr_C. It is the ability to evaluate whether block cipher algorithm is secure against differential analysis.

(1) The design principle of DABC model.

MLP is the base model that is used in almost all networks. It is the first model that tried to handle different deep learning tasks. MLP model is chosen as the base model for the differential analysis task. However, the MLP model does not achieve the expected results for predicting AS. The shortcut connection module is used in many popular network models, such as Transformer [34], and ResNet. It allows the network to learn information about both direct mapping and the residual components. This module allows the network to retain the original information during the training process. It also increases the new knowledge gained in the network. This module can potentially improve the model performance without slowing down the learning efficiency. We add the shortcut connection module to the MLP model for better learning the information. Thus, the DABC model is formally proposed.

Table 2
The data input shapes for two different models, where $DABC^1$ model is applying to the 8-branch P-permutation SPN structure algorithm and $DABC^2$ model is applying to the 16-branch P-permutation SPN structure algorithm.

Model	Data input shape
$DABC^1$	(batchsize, 2, 8)
$DABC^2$	(batchsize, 2, 16)

(2) The data input of DABC model

This paper selected the SPN structure algorithm as the experimental object. The SPN structure is a classical structure of block cipher algorithms, such as AES [35], KLEIN [36] et al. The specific value of S-box is independent of the AS. Differential propagation is related to the round function of block cipher, such as the permutation layer (P-permutation). P-permutation affects the differential propagation of cryptographic algorithms. Round is a factor that directly affects the effectiveness and security performance. Round and P-permutation are selected as two features of the differential analysis dataset. The DABC model networks are designed for applying to 8-branch and 16-branch P-permutation, respectively. Table 6 shows five samples of 8-branch SPN structure algorithms. It can be seen that the Round feature occupies 1 bit and the P-permutation feature occupies 8 bits. DABC model learns the two features with different effects, due to the two features occupying different positions. In order to make the DABC model more balanced to learn the hidden information behind Round and P-permutation, we expand Round bits to the number of bits of P-permutation, as shown in Fig. 3. The sample dimensions are deformed from (1, 9) to (2, 8) of the 8-branch P-permutation SPN algorithms. The two dimensions represent Round and P-permutation features, respectively. Thus, the data input shape of DABC models for 8-branch P-permutation and 16-branch P-permutation are shown in Table 2.

(3) The detailed structure of DABC model

The overall structure of DABC model for the 8-branch SPN block cipher is shown in Fig. 4. DABC model has eight hidden layers, where each hidden layer has 256/512 neural units for 8-branch and 16-branch P-permutation. Each hidden layer is considered a block. The block structure is illustrated in Fig. 5. Block is a module to illustrate the hidden layer. It consists of a fully connected layer, data normalization operation, and activation function.

The InstanceNorm1D function processes the data according to Eq. (7).

$$y = \frac{x - E[x]}{\sqrt{\mathrm{var}[x] + \varepsilon}} * \gamma + \beta \tag{7}$$

where ε is a number added to avoid convergence of the denominator or taking 0. The default value is 1e−5. γ and β usually are two variables that can be learned to compensate for normalization impact. The two variables can be sufficiently flexible to learn to trim due to the possibility of standardization mistakes.

ELU function has outperformed ReLU [37] and its variants, such as Leaky-ReLU(LReLU) and Parameterized-ReLU(PReLU). ELU function is shown in Eq. (8). The default value of Alpha is 1.0. The problem

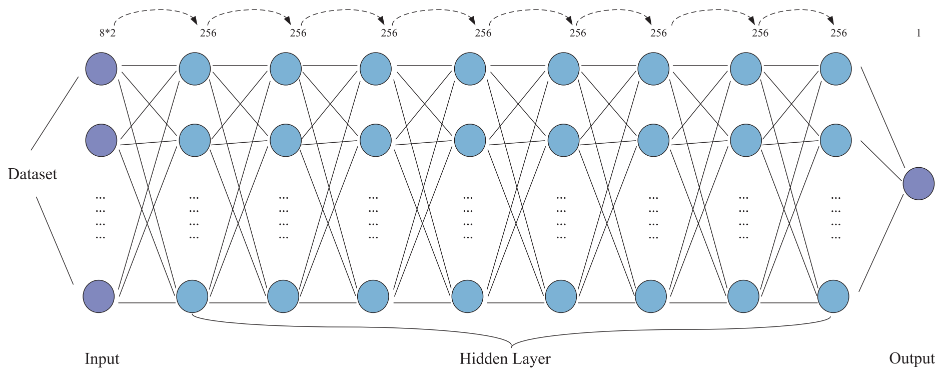


Fig. 4. The overall structure of DABC model for 8-branch SPN block cipher.

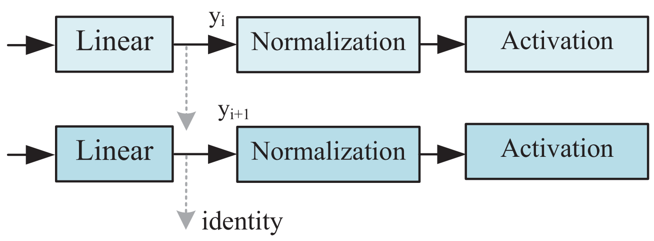


Fig. 5. The detailed structure of block module used in DABC model hidden layers.

means that a large gradient will be generated in backpropagation when abnormal input occurs. This large gradient will cause the gradient to disappear. ELU does not have the problem of neuron death. It has a faster training time than other linear non-saturating activation like ReLU and its variants.

$$ELU = \begin{cases} x & x > 0 \\ \text{alpha} * (e^x - 1) & x \le 0 \end{cases} \tag{8}$$

The shortcut connection module is used in the block. It allows the deep learning network to learn more information about the features $y_i (i = 1, 2, \dots, 7)$. The output data of the fully connected layers in the first five-block modules are retained. The output data is transmitted to the subsequent hidden layer in the first seven blocks. The output layer is a fully-connected layer with only one neuron. This neuron outputs the predicted AS value.

For the 8-branch and 16-branch P-permutation SPN structure cryptographic algorithms, the DABC models have some differences in neurons due to the difference in the number of features. The hierarchical structure of the DABC modules are shown in Tables 3 and 4, respectively.

4. Training and evaluating

4.1. Differential analysis datasets

The differential analysis datasets are constructed to evaluate AS for 8-branch and 16-branch P-permutation SPN structure algorithms. For space considerations, this subsection only illustrates the differential analysis dataset for the 8-branch P-permutation SPN structure algorithms. The linear and nonlinear components constitute the round function of block cipher. The linear component includes operations such as P-permutation and column obfuscation, and the nonlinear layer is usually Sbox substitution. AS is used to evaluate the ability of the cryptographic algorithm to resist differential analysis. In general, the specific S-box does not affect the results of differential analysis. Round and P-permutation are the main reasons that affect its results. Thus, Round and P-permutation are chosen as the features of differential analysis datasets, and AS as the label. The following are the specific collection steps:

Table 3
The architecture and parameters of $DABC^1$ model applied to 8-branch SPN structure block ciphers.

Layer(type)	Output shape	Param #
Flatten	[1,16]	0
Linear-1	[1,256]	4352
Linear-2	[1,256]	65 792
Linear-3	[1,256]	65 792
Linear-4	[1,256]	65 792
Linear-5	[1,256]	65 792
Linear-6	[1,256]	65 792
Linear-7	[1,256]	65 792
Linear-8	[1,256]	65 792
Linear-9	[1,256]	65 792
InstanceNorm1D-1	[1,256]	512
InstanceNorm1D-2	[1,256]	512
InstanceNorm1D-3	[1,256]	512
InstanceNorm1D-4	[1,256]	512
InstanceNorm1D-5	[1,256]	512
InstanceNorm1D-6	[1,256]	512
InstanceNorm1D-7	[1,256]	512
InstanceNorm1D-8	[1,256]	512
InstanceNorm1D-9	[1,256]	512
InstanceNorm1D-10	[1,256]	512
Linear-10	[1,1]	257
Total Params: 535 553		

Table 4
The architecture and parameters of $DABC^2$ model applied to 16-branch SPN structure block ciphers.

Layer(type)	Output shape	Param #
Flatten	[1,32]	0
Linear-1	[1,512]	16 896
Linear-2	[1,512]	65 792
Linear-3	[1,512]	65 792
Linear-4	[1,512]	65 792
Linear-5	[1,512]	65 792
Linear-6	[1,512]	65 792
Linear-7	[1,512]	65 792
Linear-8	[1,512]	65 792
Linear-9	[1,512]	65 792
InstanceNorm1D-1	[1,512]	1024
InstanceNorm1D-2	[1,512]	1024
InstanceNorm1D-3	[1,512]	1024
InstanceNorm1D-4	[1,512]	1024
InstanceNorm1D-5	[1,512]	1024
InstanceNorm1D-6	[1,512]	1024
InstanceNorm1D-7	[1,512]	1024
InstanceNorm1D-8	[1,512]	1024
Linear-10	[1,1]	513
Total Params: 2 127 873		

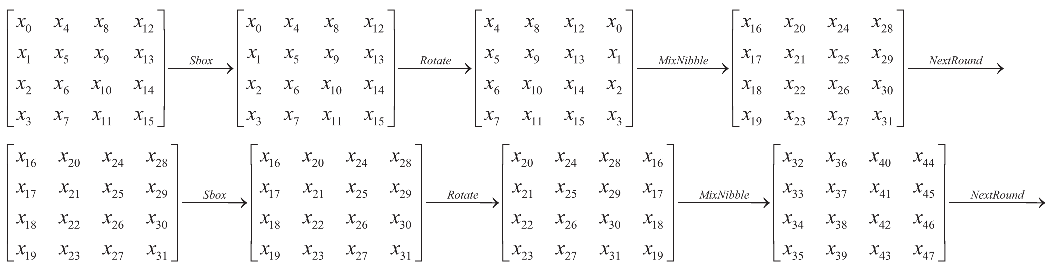


Fig. 6. The differential propagation path of KLEIN.

Table 5
The AS of the 2 to 11 rounds of KLEIN.

Round	2	3	4	5	6	7	8	9	10	11	12
AS	5	8	15	18	20	24	30	34	36	39	46

Table 6
Five samples of 8-branch SPN structure algorithms.

Round	P-permutation	AS
4	1,2,3,4,5,6,7,0	15
6	2,1,7,0,4,5,3,6	15
7	0,3,5,6,4,7,1,2	23
8	3,7,6,0,1,4,5,2	30
11	4,1,0,6,3,5,7,2	50

(1) Differential analysis of KLEIN.

KLEIN-64 is a classical 8-branch P-permutation SPN structure algorithm. The KLEIN-64 algorithm is referred to as KLEIN for convenience. The round function of KLEIN includes AddRoundKey, S-box substitution (Sbox), P-permutation (Rotate), and column confusion (MixNibble) operations. The AddRoundkey operation is irrelevant to differential analysis. The modules with linear diffusion are P-permutation and column confusion. The automatic differential model of KLEIN is built according to MILP method. P-permutation operation is performed without changing the value of difference variables. The confusion matrix of KLEIN is a fourth-order MDS matrix with the linear diffusion layer reaching the optimal state. The number of branches is $B = 5$. The inequality constraints of column confusion are expressed as follows in Eq. (5). The module with confusion effect is Sbox substitution operation. If the half-byte input differential is non-zero, then the half-byte output differential is also non-zero in Sbox substitution. In this case, A_t is denoted as 1 and vice versa. The inequality constraint is used as shown in Eq. (9). Therefore, the constraints for S-box do not generate new differential variants.

$$A_t = \begin{cases} 1 & \Delta x \neq 0 \\ 0 & \Delta x = 0 \end{cases} \tag{9}$$

The differential propagation path of KLEIN is shown in Fig. 6. After comparing the relevant literature [38], the automatic differential analysis model is proved to be correct. The AS for the 2 to 11 rounds KLEIN algorithm is shown in Table 5.

(2) Generation of differential data samples.

Round and P-permutation are two features of the differential analysis datasets. P-permutation affects the diffusion effect of block cipher algorithms. It also affects the performance of differential analysis. P-permutation is the algorithm component that affects AS. In other words, the change of P-permutation means that new algorithms are generated. The P-permutation of random generated replaced KLEIN P-permutation to create novel cryptographic algorithm. A total of 12,500 data samples are generated through step 2. Five samples of the differential analysis dataset for 8-branch P-permutation SPN structure algorithms are shown in Table 6.

Table 7
The hyperparameter setting of DABC model.

Hyper parameters	Values
Optimizer/learning rate	SGD/0.001
Loss function	MSE
Epochs	5000
Batch size	2500

4.2. Hyper-parameter setting

Hyperparameters are used to control model training, which can only be specified by design relying on human experience and cannot be estimated from the data. These parameters need to be assigned or initialized before training. Model performance depends heavily on the hyperparameters, these hyperparameters cannot be applied to all models. Setting different hyperparameters for different deep learning models can maximize model performance. Usually, the impact of hyperparameters on a model is known. How to set appropriate hyperparameters and how these parameters influence training are unknown and challenging.

Hyperparameter optimization is the most common problem in deep learning. It is the process of configuring hyperparameters. The better the hyperparameter tuning is, the better the resulting model. The chosen hyperparameters of this model are learning rate/optimizer, loss function, epoch value, and batch size. Any change in parameters affects the training phase. The learning rate/optimizer regulates the size of weight updates and model training speed and accuracy. It is simple to cause the goal function to vary substantially if the learning rate is too high, making it difficult to determine the optimum. Convergence will be too sluggish and take too long if the learning rate is set too low. Batch-size parameter significantly affects the model's optimization and performance. The correct of Batch-size is the key to finding a balance between memory efficiency and memory capacity. Epoch is a hyperparameter that defines the determination of training time. This affects the time spent on training.

The parameter setting has a nontrivial impact on the neural network model. The hyperparameters of the differential analysis model are determined after tuning and optimization. Hyperparameter selection is made to define the optimizer/learning rate, loss function, optimizer, number of epochs, and batch size for this regression task. The hyperparameters are chosen in Table 7.

4.3. Training process

The training model is to calculate the difference between the model output and the target, iteratively changing the weight function. By repeating a long training procedure, the total loss reaches the desired level. Algorithm 1 is used to implement the training and evaluating phases for differential analysis. It stores the best performance model after certain epochs.

Algorithm 1: The training and evaluating phases of differential analysis model

```
Input: training dataset $D_{train}$, validation dataset $D_{validation}$, test dataset $D_{test}$, number of epochs $N_{epoch}$, P-permutation $P$, Round $R$, the actual value of AS $AS$.
Output: The model $M_{DABC}$ for differential analysis.
1: Set training set and validation set
//The format of set is $feature_1$, $feature_2$, $label$.
2: for $i \rightarrow 1$ to $75,000$ and $j \rightarrow 1$ to $25,000$ and $k \rightarrow 1$ to $25,000$ do
3:     set $D_{train} = P_i, R_i, AS_i$ and $D_{validation} = P_j, R_j, AS_j$
4:         $D_{test} = P_k, R_k, AS_k$
5: end for
//Training the differential analysis model $M_{train}$.
6: for e = 1 to $N_{epoch}$ do
7:     training $M_{train}$ on training dataset $D_{train}$
8:     evaluating $M_{train}$ on validation dataset $D_{validation}$
9:     adjusting the hyperparameters of $M_{train}$
10: end for
//Testing the differential analysis model $M_{train}$
11: evaluating the differential analysis model $M_{train}$ on test dataset $D_{test}$
12: $M_{DABC} = M_{train}$
13: Return $M_{DABC}$
```

4.4. Evaluating metrics

The performance of neural network model needs to be evaluated on a test dataset. m is the sample size of test dataset. y_i is the actual AS value. $\bar{y}$ is the average AS value of the test dataset. $f(x_i)$ is the predicted AS value.

Two performance evaluation indicators are root mean square error RMSE and coefficient of determination R^2. RMSE measures the error between predicted values and actual values. It is very sensitive to some extreme errors generated by the model. Therefore, RMSE gives a good indication of the model precision. RMSE score is as close to zero as possible for neural network models. It represents the difference between predicted and actual AS in this task. RMSE is calculated as shown in Eq. (10).

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(f(x_i) - \bar{y})^2} \tag{10}$$

R^2 measures the fit goodness of regression models. It indicates the degree of approximation between predicted and actual values. The closer it is to 1, the better the regression fit. Neural network models with a good fit of more than 0.8 are generally thought to be better. The calculation of R^2 is shown in Eq. (11).

$$R^2 = 1 - \frac{\frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2}{\frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y})^2} \tag{11}$$

5. Experiment and analysis

5.1. AS prediction results for 8-branch SPN structure algorithms

The prediction effects of the three models are shown in Fig. 7. These three models are used to predict AS of the 8-branch SPN structure cryptographic algorithms. Fig. 7(a) shows that when the number of AS is greater than 60, the predicted value of AS has a large error by MLP model. The results obtained by Res18 and DABC models are evenly distributed on this curve. When the AS is less than 40, the prediction results of DABC model are more accurate than other models. The comparison of prediction differential analysis performance for 8-branch SPN structure block cipher of MLP, Res18, and DABC models is shown in Table 8. R^2 is used to evaluate the accuracy of the model. The R^2 of DABC model is 0.97122. RMSE is used to evaluate the prediction error of the model. The deviation of DABC model is 3.27018, which is lower than Res18 model and MLP model. Therefore, DABC model has an advantage for applying in 8-branch P-permutation SPN structure algorithms.

Table 8
Comparison of prediction differential analysis performance for 8-branch SPN structure block cipher of MLP, Res18, and DABC models.

Model	R^2	RMSE
MLP	0.93949	4.74166
Res18	0.95445	4.11405
DABC	0.97122	3.27018

Table 9
Comparison of prediction differential analysis performance for 16-branch SPN structure block cipher of MLP, Res18, and DABC models.

Model	R^2	RMSE
MLP	0.93012	5.70577
Res18	0.93159	5.64583
DABC	0.94701	5.08549

This paper also uses the MILP method to compare with the DABC model. We selected 20 sets of cryptographic algorithms and predicted the AS for the first 20 rounds. The prediction time of DABC model is 0.02s. Ignoring the time of building differential model, the calculation time for the MILP method is 19 min.

5.2. AS prediction results for 16-branch SPN structure algorithms

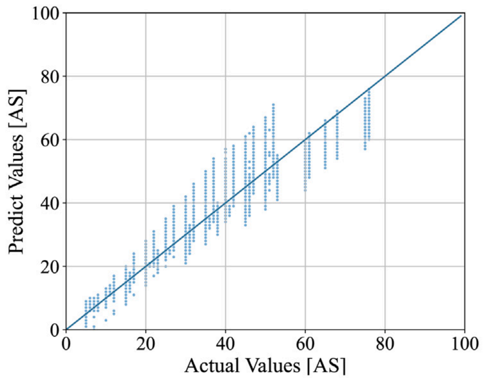
The prediction effects of the three models are shown in Fig. 8. These three models are used to predict AS for 16-branch SPN structure cryptographic algorithms. Fig. 8(a) shows that when the number of AS is between 20 and 40, the predicted value of AS obtained by MLP model is 15–20 larger than the actual value. Fig. 8(b) shows that when the AS range is 20 to 40, the predicted value of AS obtained by Res18 model is 20 larger than the actual value. When AS ranges from 60 to 80, the prediction effect of Res18 model is worse than that of MLP and DABC model. The predicated points are evenly distributed in Fig. 8(c). The predicted effects of the three models on 16-branch P-permutation SPN structure algorithms are shown in Table 9. R^2 is used to evaluate the accuracy of the model. The R^2 of DABC model is 0.94701. RMSE is used to evaluate the prediction error of the model. The deviation of DABC model is 5.08549, which is lower than that of Res18 model and MLP model. It can be said that the DABC model is truly capable of predicting the results of differential analysis for 16-branch P-permutation SPN structure algorithms.

This paper also uses the MILP method to compare with the DABC model. We selected 20 sets of cryptographic algorithms and predicted the AS for the first 20 rounds. The prediction time of the DABC model is 0.05s. Ignoring the time of building differential model, the calculation time for the MILP method is 53 min.

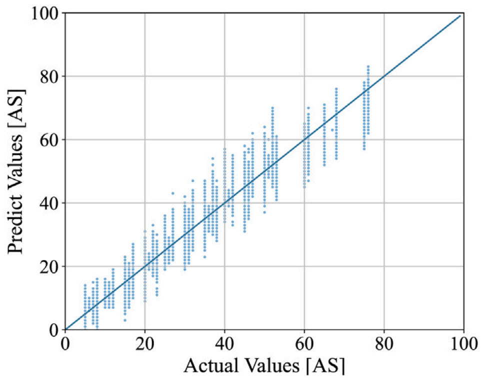
5.3. The depth of DABC model

Deep learning models are sensitive to hyperparameters such as changing the number of hidden layers resulting in the new model. To investigate the effect of the number of hidden layers, we further changed the hidden layers by adding or reducing the blocks to DABC model.

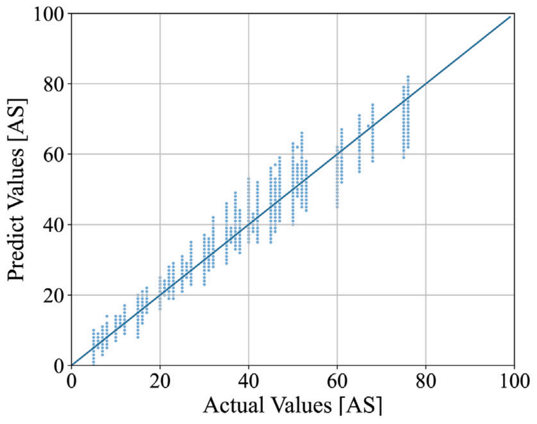
The same setting for hyperparameters as those used in the previous experiment is shown in Table 7. We study the influence of network models with different hidden layers on 8-branch and 16-branch SPN structure algorithms, respectively. As can be seen from Table 10, the network model with 8 hidden layers performs best on the data set of the 8-branch SPN structure. As can be seen from Table 11, the



(a) MLP



(b) Res18



(c) DABC

Fig. 7. Predicted results of AS for 8-branch P-permutation SPN structure cryptographic algorithms. (a) Prediction results for the MLP model, (b) prediction results for the Res18 model, and (c) prediction results for the DABC model. A straight line with y = x exists for each subplot, which is the baseline of the model. The coordinates of the points are (Actual Values, Predict Values). When the point is closer to the y = x line, it indicates that the model predicts AS more accurately.

Table 10
The effect of various hidden layers of DABC model applied for 8-branch SPN structure algorithms.

Numbers	R^2	RMSE	Numbers	R^2	RMSE
1	0.90191	6.03724	6	0.96624	3.54164
2	0.91841	5.50580	7	0.96758	3.47038
3	0.93525	4.90476	8	0.97122	3.27018
4	0.94797	4.39693	9	0.94762	4.41161
5	0.95777	3.96126	10	0.94284	4.60848

Table 11
The effect of various hidden layers of DABC model applied for 16-branch SPN structure algorithms.

Numbers	R^2	RMSE	Numbers	R^2	RMSE
1	0.90705	6.58072	6	0.94950	4.85039
2	0.92770	5.80392	7	0.94930	4.97437
3	0.93308	5.58384	8	0.94701	5.08549
4	0.94051	5.26459	9	0.94193	5.32353
5	0.94511	5.05722	10	0.94058	5.38537

network model with 6 hidden layers performs best on the 16-branch SPN structure. When the depth is greater than 6, the score of R^2 even worse. Meanwhile, RMSE scores goes up. However, the R^2 of these models with more than 6 hidden layers are about to fluctuate from 0.94 to 0.95. The range of RMSE values is between 4.8 and 5.4.

In terms of differential analysis, the DABC model is designed to analyze cryptographic algorithms for 8-branch and 16-branch SPN structures. Therefore, we need the model to have applicability for the two structures. The number of hidden layers is set to 8, with fewer network model parameters. It facilitates easy model training and light weighting.

5.4. Model complexity and training time

We provide a detailed discussion and analysis of the DABC model complexity. For a fair comparison and in order to rule out the influence of varying platforms, we report the number of model parameters as well as FLOPs (number of add and multiple operations). We also select MLP model and Res18 model for comparison. It can be seen from Tables 12 and 13 that the proposed DABC network consumes less memory than the other two models. Furthermore, DABC model has fewer FLOPs in comparison to other networks. This means that our network has a smaller model scale. The training time spent by DABC model is far less than the other two models.

6. Conclusion

This paper designed an efficient differential analysis model based on deep learning. Traditional differential analysis methods rely on algorithmic structures or the help of mathematical analysis tools. The computation time of specific algorithms is difficult to predict. Unlike traditional differential analysis methods, this paper aims to build a

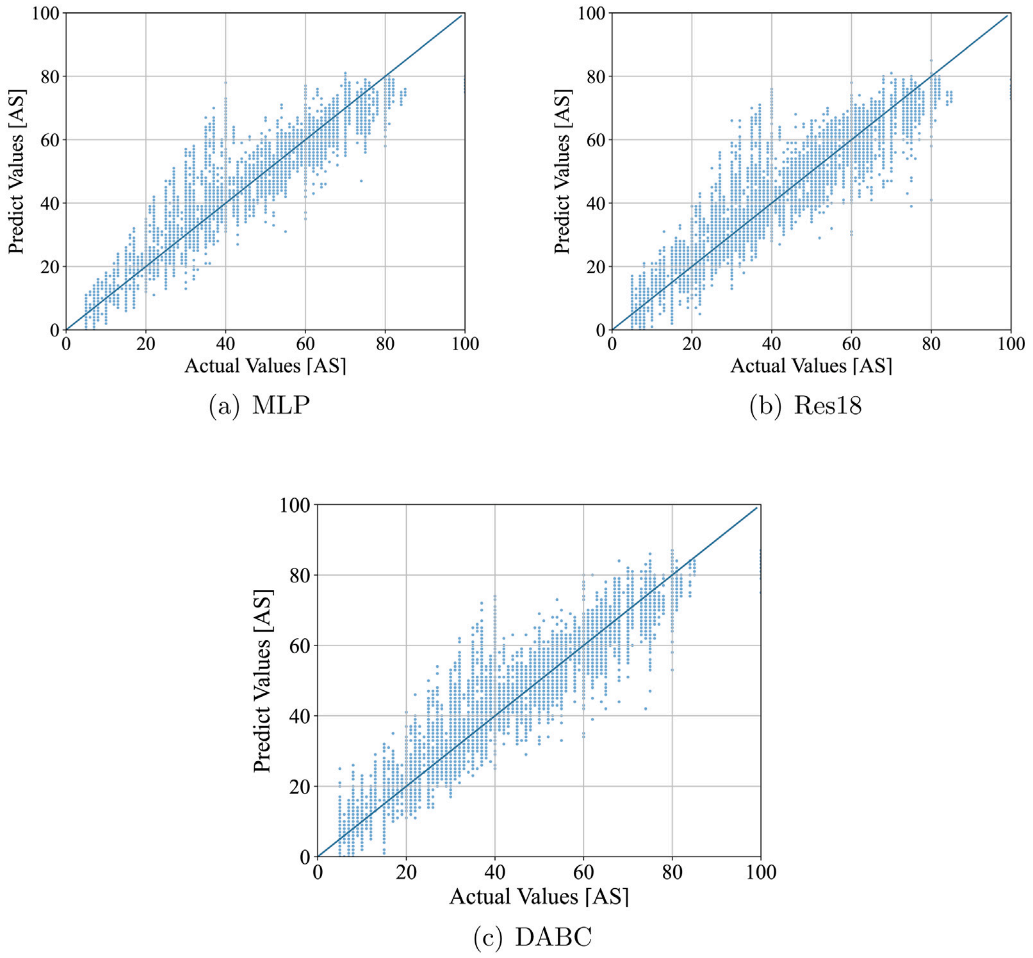


Fig. 8. Predicted results of AS for 16-branch P-permutation SPN structure cryptographic algorithms. (a) Prediction results for the MLP model, (b) prediction results for the Res18 model, and (c) prediction results for the DABC model. A straight line with y = x exists for each subplot, which is the baseline of the model. The coordinates of the points are (Actual Values, Predict Values). When the point is closer to the y = x line, it indicates that the model predicts AS more accurately.

Table 12
The number of parameters and FLOPS and training time of different algorithms for 8-branch P-permutation SPN structure algorithms.

Model	MFLOPs	Params	Training time
MLP	0.46	0.46 M	12.58 h
Res18	3.89	3.85 M	15.58 h
DABC	0.53	0.53 M	6.22 h

Table 13
The number of parameters and FLOPS and training time of different algorithms for 16-branch P-permutation SPN structure algorithms.

Model	MFLOPs	Params	Training time
MLP	1.85	1.85 M	20.00 h
Res18	4.17	3.85 M	17.52 h
DABC	2.11	2.11 M	6.27 h

model that applies to the structure of classical encryption algorithms. The model can accurately predict the minimum number of active S-boxes in a short time. The model combines the simplicity and efficiency of the MLP model and the learning ability of the shortcut connection module. Experimental results show that the DABC model predicts the minimum number of active S-box of the SPN structured encryption algorithm with an accuracy of over 97%. The DABC model can effectively perform differential analysis of cryptographic algorithms for a class of structures. In addition, the differential analysis dataset collected using the MILP method can facilitate the subsequent development of deep learning-based differential analysis.

CRediT authorship contribution statement

Ying Huang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing, Visualization, Reviewing and editing. **Lang Li:** Methodology, Investigation, Resources, Writing, Supervision, Project administration. **Ying Guo:** Validation, Writing. **Yu Ou:** Methodology, Software, Resources, Writing, Visualization. **Xiantong Huang:** Conceptualization, Data curation.

Declaration of 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.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research is supported by the Hunan Provincial Natural Science Foundation of China (2022JJ30103, 2022JJ50016), the Science and

Technology Innovation Program of Hunan Province, China (2016TP1020), Open fund project of Hunan Provincial Key Laboratory of Intelligent Information Processing and Application for Hengyang Normal University, China (2022HSKFJJ011), "the 14th Five-Year Plan" Key Disciplines and Application-oriented Special Disciplines of Hunan Province (Xiangjiaotong [2022] 351).

References

[1] Y. Ou, L. Li, Side-channel analysis attacks based on deep learning network, Front. Comput. Sci. 16 (2) (2022) 1–11.

[2] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, T. Wang, Deepsec: A uniform platform for security analysis of deep learning model, in: 2019 IEEE Symposium on Security and Privacy, SP, IEEE, 2019, pp. 673–690.

[3] A. Gohr, Improving attacks on round-reduced speck32/64 using deep learning, in: Annual International Cryptology Conference, Springer, 2019, pp. 150–179.

[4] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, SIMON and SPECK: Block ciphers for the internet of things, Cryptol. ePrint Arch. (2015).

[5] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[6] A. Jain, V. Kohli, G. Mishra, Deep learning based differential distinguisher for lightweight cipher PRESENT, Cryptol. ePrint Arch. (2020).

[7] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J. Robshaw, Y. Seurin, C. Vikkelsoe, PRESENT: An ultra-lightweight block cipher, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2007, pp. 450–466.

[8] E. Bellini, M. Rossi, Performance comparison between deep learning-based and conventional cryptographic distinguishers, in: Intelligent Computing, Springer, 2021, pp. 681–701.

[9] D.J. Wheeler, R.M. Needham, TEA, a tiny encryption algorithm, in: International Workshop on Fast Software Encryption, Springer, 1994, pp. 363–366.

[10] J. Polimón, J.C. Hernández-Castro, J.M. Estévez-Tapiador, A. Ribagorda, Automated design of a lightweight block cipher with genetic programming, Int. J. Knowl.-Based Intell. Eng. Syst. 12 (1) (2008) 3–14.

[11] W. Tian, B. Hu, Deep learning assisted differential cryptanalysis for the lightweight cipher SIMON, KSII Trans. Internet Inf. Syst. (TIIS) 15 (2) (2021) 600–616.

[12] H. Wang, P. Cong, H. Jiang, Y. Wei, Security analysis of SIMON32/64 based on deep learning, J. Comput. Res. Dev. 58 (5) (2021) 1056.

[13] A. Baksi, Machine learning-assisted differential distinguishers for lightweight ciphers, in: Classical and Physical Security of Symmetric Key Cryptographic Algorithms, Springer, 2022, pp. 141–162.

[14] Z. Hou, J. Ren, S. Chen, Cryptanalysis of round-reduced SIMON32 based on deep learning, Cryptol. ePrint Arch. (2021).

[15] T. Yadav, M. Kumar, Differential-ML distinguisher: Machine learning based generic extension for differential cryptanalysis, in: International Conference on Cryptology and Information Security in Latin America, Springer, 2021, pp. 191–212.

[16] Y. Chen, Y. Shen, H. Yu, S. Yuan, A new neural distinguisher considering features derived from multiple ciphertext pairs, Cryptol. ePrint Arch. (2021).

[17] A. Benamira, D. Gerault, T. Peyrin, Q.Q. Tan, A deeper look at machine learning-based cryptanalysis, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2021, pp. 805–835.

[18] Z. Bao, J. Guo, M. Liu, L. Ma, Y. Tu, Conditional differential-neural cryptanalysis, Cryptol. ePrint Arch. (2021).

[19] A. Gohr, G. Leander, P. Neumann, An assessment of differential-neural distinguishers, Cryptol. ePrint Arch. (2022).

[20] M. Matsui, A. Yamagishi, A new method for known plaintext attack of FEAL cipher, in: Workshop on the Theory and Application of of Cryptographic Techniques, Springer, 1992, pp. 81–91.

[21] N. Mouha, Q. Wang, D. Gu, B. Preneel, Differential and linear cryptanalysis using mixed-integer linear programming, in: International Conference on Information Security and Cryptology, Springer, 2011, pp. 57–76.

[22] A.J. Elbirt, W. Yip, B. Chetwynd, C. Paar, An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 9 (4) (2001) 545–557.

[23] L. Li, B. Liu, H. Wang, QTL: a new ultra-lightweight block cipher, Microprocess. Microsyst. 45 (2016) 45–55.

[24] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, S.M. Sim, The SKINNY family of block ciphers and its low-latency variant MANTIS, in: Annual International Cryptology Conference, Springer, 2016, pp. 123–153.

[25] W. Wu, L. Zhang, LBlock: a lightweight block cipher, in: International Conference on Applied Cryptography and Network Security, Springer, 2011, pp. 327–344.

[26] Y. Guo, L. Li, B. Liu, Shadow: A lightweight block cipher for IoT nodes, IEEE Internet Things J. 8 (16) (2021) 13014–13023.

[27] S. Banik, S.K. Pandey, T. Peyrin, Y. Sasaki, S.M. Sim, Y. Todo, GIFT: a small present, in: International Conference on Cryptographic Hardware and Embedded Systems, Springer, 2017, pp. 321–345.

[28] S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, L. Song, Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES (L) and other bit-oriented block ciphers, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2014, pp. 158–178.

[29] B. Zhu, X. Dong, H. Yu, MILP-based differential attack on round-reduced GIFT, in: Cryptographers' Track at the RSA Conference, Springer, 2019, pp. 372–390.

[30] K. Fu, M. Wang, Y. Guo, S. Sun, L. Hu, MILP-based automatic search algorithms for differential and linear trails for speck, in: International Conference on Fast Software Encryption, Springer, 2016, pp. 268–288.

[31] L. Song, J. Guo, D. Shi, S. Ling, New MILP modeling: improved conditional cube attacks on Keccak-based constructions, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2018, pp. 65–95.

[32] M.I. Jordan, T.M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.

[33] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[35] J. Daemen, V. Rijmen, The Rijndael block cipher: AES proposal, in: First Candidate Conference (AeS1), 1999, pp. 343–348.

[36] Z. Gong, S. Nikova, Y.W. Law, KLEIN: a new family of lightweight block ciphers, in: International Workshop on Radio Frequency Identification: Security and Privacy Issues, Springer, 2011, pp. 1–18.

[37] R. Faster, Towards real-time object detection with region proposal networks, Adv. Neural Inf. Process. Syst. 9199 (10.5555) (2015) 2969239–2969250.

[38] M.B. Ilter, A.A. Selçuk, A new MILP model for matrix multiplications with applications to KLEIN and PRINCE, in: SECRYPT, IEEE, 2021, pp. 420–427.

Ying Huang received her B.E. degree from Hengyang Normal University, Hengyang, China, in 2021. She is currently working toward the Master's degree at Hengyang Normal University, Hengyang, China, since 2021. Her current research interests include embedded systems and information security.

Lang Li received his Ph.D. and Master's degrees in computer science from Hunan University, China, in 2010 and 2006, respectively, and earned his BS degree in circuits and systems from Hunan Normal University, China in 1996. Since 2011, he has been working as a professor in the College of Computer Science and Technology at the Hengyang Normal University, China. His research interests include embedded computing and information security.

YingGuo received her master's and B.E. degree from Hengyang Normal University, Hengyang, China, in 2022 and 2019, respectively. She is currently pursuing a Ph.D. degree at the School of Computer and Information Security, Guilin University Of Electronic Technology, Guilin, China. Her current research interests include embedded systems and information security.

Yu Ou received a bachelor's degree from Hengyang Normal University, Hengyang, China, in 2018 and received a master's degree from the College of Information Science and Engineering, Hunan Normal University, Changsha, China, in 2021. Since 2019, his current research interests include embedded systems and information security.

Xiantong Huang received the B.E. degree in computer science from Hengyang Normal University, Hengyang, China, in 2021, where she is currently working toward the master's degree. Her main research interests include embedded system and information security.