





# Recent Advances in Machine Learning for Differential Cryptanalysis

Isabella Martínez<sup>1</sup>, Valentina López<sup>1</sup>, Daniel Rambaut<sup>2</sup>, Germán Obando<sup>3</sup>,  
Valérie Gauthier-Umaña<sup>4</sup> (✉) , and Juan F. Pérez<sup>5</sup> 

<sup>1</sup> Ressolve, Bogotá, Colombia

<sup>2</sup> Opencs, Bogotá, Colombia

<sup>3</sup> Departamento de Ingeniería Electrónica, Universidad de Nariño, Nariño, Colombia

<sup>4</sup> Systems and Computing Engineering Department, Universidad de los Andes,  
Bogotá, Colombia

[ve.gauthier@uniandes.edu.co](mailto:ve.gauthier@uniandes.edu.co)

<sup>5</sup> Department of Industrial Engineering, Universidad de los Andes, Bogotá, Colombia

**Abstract.** Differential cryptanalysis has proven to be a powerful tool to identify weaknesses in symmetric-key cryptographic systems such as block ciphers. Recent advances have shown that machine learning methods are able to produce very strong distinguishers for certain cryptographic systems. This has generated a large interest in the topic of machine learning for differential cryptanalysis as evidenced by a growing body of work in the last few years. In this paper we aim to provide a guide to the current state of the art in this topic in the hope that a unified view can better highlight the challenges and opportunities for researchers joining the field.

**Keywords:** Differential cryptanalysis · Machine learning · Survey

## 1 Introduction

Since the seminal paper of Rivest [24], a large number of machine learning approaches for cryptanalysis have been proposed. While a steady flow of progress has been sustained since, the recent paper of Gohr [12] has provided a new flurry of excitement as it manages to employ recent deep learning architectures to improve upon known results in differential cryptanalysis.

Given that the space of cryptographic systems is quite large, and that there exist many machine learning techniques, it can be hard to gather a clear picture of the current state of the art in the field of machine learning for cryptanalysis. This paper aims to provide a guide to the current state of the art in this field, especially in regards to *differential* cryptanalysis, where most of the recent work has centered.

## 1.1 Related Work

In spite of the increasing number of works in the field of machine learning for cryptanalysis there are very few surveys on the topic. In [4], the authors provide a detailed analysis of the strengths and weaknesses of various machine learning techniques and their effectiveness in breaking different types of cryptographic systems. However, being over a decade old, it does not consider the recent progress in the field. A more recent survey [1] provides a wider overview of machine learning in cryptography, considering not just cryptanalysis but also image steganalysis, side-channel analysis, power analysis attacks, and encrypted traffic classification. With this wider spectrum of applications, the topic of cryptanalysis receives a corresponding limited attention span, thus focusing on only the most relevant contributions according to the authors. Finally, the most recent survey [33] considers the dichotomy between neural networks and cryptography, and in particular covers the application of neural computing to attack cryptographic systems. The survey considers a number of contributions in this area but misses the proposal of neural distinguishers by Gohr [12] and others, which is a key development in the area. In this survey paper we provide an up-to-date guide to these and other recent advances in the field of machine learning for differential cryptanalysis.

In the following we provide some background definitions to summarize the main concepts in both differential cryptanalysis and machine learning. Next, we survey existing works in the area of machine learning for differential cryptanalysis, with a focus on the most recent results.

## 2 Foundations

In this section we provide a brief summary of definitions of symmetric cryptography and machine learning necessary for the discussion on how these topics coalesce around cryptanalysis.

### 2.1 Symmetric Cryptography

Symmetric-key algorithms use the same key for encryption and the decryption. One type of symmetric-key algorithms are stream ciphers, where the bits of the message are XORed with a pseudorandom secret key, which is expensive but fast, such as ARC4. Another type are block ciphers, where the message is divided into blocks of the same size. Block ciphers are typically made of several rounds, which are almost the same, except for some special values (called round constants) and the round key. These round keys are built from the secret key using an algorithm called “key schedule”.

There are several families of block ciphers, such as the ones based on substitution-permutation networks (SPN), examples of which include the standard AES, Serpent and PRESENT. In SPNs, each round involves an XOR between the message and the round key followed by an substitution box (S-Box)

and a permutation box (P-Box) that is in charge of distributing the outputs of the S-box to many other S-boxes in the next round. Another family is the one constructed using a Feistel structure, such as DES, TEA, Blowfish and Twofish. In this structure, the message is divided in left and right parts. The right part is the input of a so-called  $F$  function and its output is XORed with the left part of the message. The  $F$  function is made by a non-linear function that has an SPN structure and hides the relation between the ciphertext and the key ensuring Shannon's property of confusion. A simpler set of block ciphers are those in the ARX (Add-Rotate-Xor) family, such as SPECK and SIMON. The symmetric cryptography used in constrained environments (e.g. IoT) is called lightweight. The ARX cipher is considered to be a lightweight, fast, and secure encryption method. It is widely used in applications that require high-performance cryptography, such as embedded systems, mobile devices, and network security protocols. Examples include Salsa20, ChaCha and SPECK, SIMON and SIMECK. Lightweight cryptosystems based on other structures include PRESENT and GIFT64.

Block ciphers structures are often used as the base of Message Authentication Code (MAC) schemes, like Chasky, and for the construction of Hash functions. Also, some permutation primitives can be used to build high-security block ciphers, stream ciphers, and MAC's, authenticated ciphers, hash functions, etc. with a unified hardware. Examples include Gimli, KNOT and ASCON.

Symmetric ciphers can also be found in different so-called modes, like for example Electronic Codebook (ECB) and Cipher Block Chaining (CBC) modes. In ECB mode, each block is encrypted independently while in CBC mode, each plaintext is XORed with the ciphertext. They have different advantages and some of them can provide more security.

## 2.2 Symmetric Cryptanalysis

There are a number of different techniques to analyze the security of symmetric cryptosystems, such as brute force attacks, linear and differential cryptanalysis, among others.

On the one hand, linear cryptanalysis looks for linear relationships between the plaintext, ciphertext and the key and by analyzing these relationships, an attacker can deduce information about the secret key. On the other hand, differential cryptanalysis methods depend on identifying high-probability differences between plaintext and ciphertext pairs. These differences, also known as characteristics, are essential for successful attacks. The idea was first introduced in 1990 [6] where it was used to attack DES. Since then, differential cryptanalysis has been considered to attack a wide range of symmetric key cryptosystems, including AES, Blowfish, and Twofish. The goal is to find techniques that are aimed at tracking differences across the transformation network, identifying instances where the cipher deviates from random behavior, and exploiting such properties to recover the secret key. Let  $E : \{0, 1\}^n \rightarrow \{0, 1\}^m$  be a map. A differential transition for  $E$  is a pair  $(\Delta_{in}, \Delta_{out})$  in  $\{0, 1\}^n \times \{0, 1\}^m$ . The probability of the differential transition is defined as

$$P(\Delta_{\text{in}} \rightarrow \Delta_{\text{out}}) = \frac{\text{Card}(\{x \in \{0, 1\}^n : E(x) \oplus E(x \oplus \Delta_{\text{in}}) = \Delta_{\text{out}}\})}{2^n}.$$

Considerable research has been devoted to improving methods for finding characteristics. In the seminal paper by Wang et al. [30], characteristics of the hash function SHA-0 were manually constructed using knowledge of the hash function’s structure. Later advancements were made by De Canniere and Rechberger in [10], and by Leurent in [18] and [19], who enhanced Wang’s approach by imposing constraints on the plaintexts. Additionally, Chen et al. introduced a ranking system for discovering differential characteristics and applied it to lightweight block ciphers [7]. Furthermore, novel methods utilizing Mixed-Integer Linear Programming have been proposed to enhance the process of characteristic finding, as demonstrated by Mouha et al. in [22], Sun et al. in [29], and Zhao et al. in [32].

For cryptographic systems built based on the iteration of a cipher block, a differential trail or differential characteristic is a sequence of differential transitions. Thus, these characteristics show how the differences in the input propagate through the internal components of the primitive. The higher the probability of the differential characteristic, the more pairs that can be studied to recover the key, which increases the insecurity of the encryption, since for the basic differential attack, one particular output difference is expected to be more frequent, which means the cipher output can be distinguished from a random one. Any dependence of the differential probability on the key is usually suppressed. Further details can be found in [11, 26] and [28].

### 2.3 Machine Learning

Machine learning (ML) comprises a large number of methods that enable computers to learn patterns from datasets. It has found many applications in computer vision, natural language processing, voice recognition, among others. In cryptography, supervised learning, a type of ML, helps recognize encrypted data patterns and detect malicious activity. In particular, neural networks are ML models composed of a number of nodes linked by weighted connections. Neural networks are trained by adjusting connection weights and biases to minimize a loss function,  $L(\theta)$ , which can be represented as

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N \mathcal{L}(y_i, f(x_i; \theta)),$$

where  $N$  denotes the number of samples,  $y_i$  represents the observed output,  $f(x_i; \theta)$  the predicted output, and  $\mathcal{L}(y_i, f(x_i; \theta))$  measure the difference between the observed and the predicted outputs. The objective of training the neural network is to find the optimal parameters  $\theta^*$  that minimize the loss function, i.e.,

$$\theta^* = \arg \min_{\theta} L(\theta). \quad (1)$$

As discussed in the next section, many of the ML approaches for cryptanalysis rely on neural networks to build so-called neural distinguishers.

### 3 Machine Learning Approaches for Differential Cryptanalysis

This section presents a number of recent approaches for differential cryptanalysis based on machine learning methods. We classify these approaches in two categories: attacks on SIMON and SPECK, as many methods centered on these, and attacks on other cryptosystems. Table 1 provides a summary of the methods considered, classified by the cryptosystem attacked and the type of neural network employed.

**Table 1.** Comparative Table for Distinguisher Approach

		Neural network			
		Residual Net	Multilayer Perceptron	Convolutional	LSTM
Cryptosystem	SPECK	[2, 12, 14]	[31]	[8]	
	SIMON	[13, 14, 20]	[31]		
	SIMECK	[20]	[15]		
	GIMLI		[3]	[3]	[3]
	ASCON		[3]	[3]	[3]
	KNOT		[3]	[3]	[3]
	CHASKEY		[3]	[3, 8]	[3]
	TEA		[5]	[5]	
	RAIDEN		[5]	[5]	
	PRESENT		[15]		
	GIFT64		[31]		
	DES			[8]	

#### 3.1 Attacks for SIMON and SPECK

As mentioned before, the recent contribution by Gohr [12] has become a keystone in the area of machine learning for differential cryptanalysis. The paper presents a novel technique to improve attacks on the round-reduced version of the Speck32/64 block cipher using deep learning. The author argues that existing attacks on the cipher are limited by the complexity of the differential and linear approximations used to analyze the cipher. To overcome this limitation, the proposed approach consists of training a deep neural network to predict the cipher output for a given input, considering a specific number of rounds. The author demonstrates the effectiveness of the approach by applying it to up to eight rounds of Speck 32/64 and shows that it can improve the success rate of

the attack compared to existing techniques. The author also shows that the approach is robust to noise and can generalize to larger numbers of rounds, although it is limited by the amount of training data needed, which increases with the number of rounds. A number of papers have extended and improved Gohr’s work recently, as we describe next.

Next, Hou et. al. [13] use deep residual neural networks to train differential distinguishers for SIMON32 with eight and nine rounds. They investigate how patterns in the input differences affect the model accuracy. Even though the input differences they employ result from differential characteristics with the same probability, they discover that their accuracy is different. Employing this distinguisher, the paper develops an 11-round SIMON32 last subkey recovery attack, extends the 9-round SAT-based distinguisher to an 11-round distinguisher, and subsequently suggests a 13-round SIMON32 attack. They employ a Ghor-inspired architecture and discover that for SIMON32, the input difference has a significant impact on the model’s performance. The suggested attacks have a success rate of over 90% and require about 45 s to obtain the last subkey. To locate the key, they employ a Bayesian search technique.

In [2], the authors look for alternatives to the Gohr distinguisher that are either smaller or perform better. They are able to successfully prune to one layer, resulting in a network that performs within 1% of the original network. Convolutional autoencoders, a type of neural network, are trained on the ciphertext pairs in order to test whether preparing the input improves performance. They find out, though, that the network was no longer sufficiently sophisticated to extract pertinent data from the input, and employ LIME [23] to further assess the significance of the characteristics. They use iterative and one-shot trimming techniques and note that at least 90% of the 10-layer network (and even the 1-layer network) may be pruned without degrading average performance, and that some of the resulting networks are superior to the original network. Additionally, they investigated whether all 64 input bits were required using LIME, and discovered that each feature’s significance is very limited and no region in the bit space has a significant impact on the ranking.

An improvement to neural distinguishers based on SAT/SMT is proposed in [14], where new distinguishers for SIMON and SPECK are proposed. Specifically, the approach is able to consider large-size block-based ciphers, which leads to key recovery attacks for 13 rounds of SIMON32/64, and 14 rounds of SIMON48/96, as well as an attack on 13 rounds of SIMON64/128 using an 11-round neural distinguisher. Unlike Gohr [12], which uses text and ciphertext pairs as samples, the authors in [14] design the distinguisher by taking several differences, arranged in a matrix, as a sample. The matrix is treated as an image, and each output difference is treated as an objective feature, such that if all output differences of the matrix are from the same input difference, the sample is labeled with one, and zero otherwise. They show experimentally that the improvement in the accuracy of distinguishers is due to learning more features from the relationship between the output differences.

More recently, [20] proposes an improved differential neural distinguisher for the SIMON and SIMECK block ciphers. The proposed method improves upon previous neural distinguishers by incorporating related-key attacks, which can improve the accuracy of the distinguisher when the attacker has access to related keys. On the 8-round SIMON-64 cipher with a *single* key, the proposed neural distinguisher achieved an accuracy of 99.98%, while on the same cipher with related keys, it achieved an accuracy of 94.9%. On the six round SIMECK-64 cipher, the proposed neural distinguisher achieved an accuracy of 99.92% for a single-key attack, and of 91.7% for a related-key attack.

### 3.2 Attacks on Other Systems

Baksi et. al. [3] propose distinguishers for non-Markov ciphers to replicate “all-in-one” differentials, i.e., differentials that take into account the impact on all output differences under the same input difference. The paper presents distinguishers based on deep learning for eight rounds of Gimli-Hash, Gimli-Cipher, and Gimli-Permutation, three rounds of the Ascon-Permutation, ten rounds each of the Knot-256 and Knot-512 permutations, and four rounds of the Chaskey-Permutation. The paper compares different net architectures and experimentally shows that the multi-layer perceptron (MLP) outperforms both CNN and LSTM networks in terms of precision.

In [15], Jain et. al. propose a distinguisher based on deep neural networks for PRESENT-80 and Simeck64/128, attaining excellent precision for six and seven rounds, respectively. They pick a few input differentials and closely follow the steps in [3]. Additionally, they test four differential distinguisher models, the first of which makes use of the design recommended in [3] while the second makes use of the authors’ own architecture, with input differentials selected at random. Models three and four employ the same architectures as models one and two, respectively, but with selected differentials instead of random ones. Their tests show model four outperforms the rest with a validation precision of 0.86, 0.76, 0.39 and 0.27 for three, four, five and six rounds of PRESENT, respectively. For SIMECK, this precision is 1, 1, 0.83, 0.48 and 0.27 for three, four, five, six and seven rounds, respectively.

Yadav and Kumar [31] aim to create a framework for ML-based distinguishers to tackle Feistel, SPN, and ARX block ciphers. They apply it to SPECK, SIMON, and GIFT64, lowering the amount of data complexity needed for 9, 12, and 8 rounds, respectively. The paper proposes the first SIMON 12-round distinguisher with a complexity lower than  $2^{32}$ . They name this approach as hybrid Differential-ML distinguisher, which combines traditional differential distinguishers with ML models to tackle more rounds. They take advantage of Gohr’s suggested design [12] and Baksi’s improvements [3]. The network is trained using the differences directly rather than ciphertext pairs and employ a multi-layer perceptron architecture. The results show that the hybrid approach is able to increase the number of rounds considered without the need for much more data.

The Tiny Encryption Algorithm (TEA) and its evolution, RAIDEN, are put to the test in [5] using two deep learning-based distinguishers. Compared to Speck32/64, in TEA and RAIDEN the block and key sizes are both doubled. The neural distinguishers proposed are based on a multi-layer perceptron architecture and a convolutional architecture, which are shown to outperform traditional ones based on differential trails and statistical methods. Additionally, the paper shows that the loss can be greatly reduced while keeping the number of training samples constant. This is done by breaking the problem into two phases: first, a time-distributed network that treats each 32-bit chunk individually, and second, a fully connected layer. With this approach, the paper shows that the neural distinguishers are able to improve upon traditional bitflip and differential distinguishers for up to six rounds of TEA, with as little as  $10^3$  samples. This result holds too for a large number of rounds employing a larger number of samples.

A simplified version of DES (S-DES), SIMON, and SPECK, which are considered lightweight systems, are considered in [27], which proposes a general deep learning-based cryptanalysis algorithm that starts with pairs of known plaintext and ciphertext and attempts to determine the block cipher key. The authors consider two setups: i) a simplified setup where the key is made of 8 characters, each one corresponding to one of 64 ASCII characters; ii) a general setup where the key can be made of any string of bits. The neural network model proposed is able to fully attack S-DES, but it can only break SIMON and SPECK under the simplified setup. Under this setup, the ML-based attack broke the S-DES cipher with a success probability of 0.9 given  $2^{8.08}$  known plaintexts. Also, it achieved a 0.99 success probability to find 56 bits of Simon32/64 with  $2^{12.34}$  known plaintexts and 56 bits of Speck32/64 with  $2^{12.33}$  known plaintexts, both under the simplified setup.

Another approach is presented in [16], which proposes a deep learning-based cryptanalysis technique for S-DES, S-AES, and S-SPECK. The proposed method utilizes a fully-connected neural network to learn the characteristics of plaintexts and their corresponding ciphertexts in order to predict the key used by the encryption algorithm. The paper introduces an improvement to Gohr’s deep learning model [12] by incorporating skip connections and gated linear units into the neural network structure, enabling a more stable learning. As a result, an average improvement of 5.3% in accuracy is achieved compared to previous works on S-DES [27], while reducing the number of parameters by 93.16%. Furthermore, when applied on S-AES and S-SPECK, the method is able to successfully recover keys of up to 12 bits for S-AES and 6 bits for S-SPECK.

In [8] the authors propose the Extended Differential-Linear Connectivity Table (EDLCT), a tool that describes a cipher and its features relevant to a ciphertext pair. They build various machine learning-based distinguishers, including the neural distinguisher in [12], using these features. They also develop a Feature Set Sensitivity Test (FSST) to identify influential features and create surrogate models based on these features. Experiments on Speck32/64 and DES confirm that the distinguisher learns features corresponding to the



EDLCT. Additionally, the authors explain phenomena related to the neural distinguishers using the EDLCT and demonstrate how machine learning can be used to search for high-correlation differential-linear propagations in the differential-linear attack. The advantages of machine learning in applications such as Chaskey and DES are also showcased.

Another approach is presented in [25], which proposes an artificial neural network-based cryptanalysis method for a simple 8-bit substitution-permutation cipher. The method utilizes a multi-layer perceptron network with a backpropagation learning algorithm to estimate the cipher's inverse function and recover the plaintext. The authors demonstrate the effectiveness of the proposed method on several test cases and show that it outperforms traditional cryptanalysis methods in terms of speed and accuracy.

Finally, we would like to mention an earlier paper [9], which proposes the application of a neural network to S-DES to seek a relationship between plaintext, cipher text and key bits. The network can map the relation between inputs, keys and outputs and to obtain the correct values for the key bits  $k_0$ ,  $k_1$  and  $k_4$ . They also propose new S-boxes, which are more resistant to the differential attack, such that the neural network was not able to point out bits of the key under these S-boxes.

## 4 Discussion

Prior to the breakthrough generated by Gohr's proposal [12], machine learning applications to cryptanalysis were too limited to be considered efficient in a real-world context. What highlights Gohr's work is its ability to achieve a clear improvement in accuracy and data complexity in contrast to traditional differential analysis. Soon after the publication of this proposal, a number of works have emerged exploring different approaches, in particular considering changes on the network type [3], tweaks to the input layer [2, 5, 14] and hybrid approaches [13, 31], to mention just a few.

Out of the many recent works in this direction, the work by Baksi [3] deserves particular attention as it performed a comparison between network architectures to find experimentally that the multi-layer perception had the best results. This result led several authors to focus their research on this type of network architecture. However, machine learning is currently a rapidly evolving field and new architectures are regularly proposed in a variety of areas of application, opening the door for new methods to be considered for differential cryptanalysis.

While this paper has focused on machine learning methods for differential cryptanalysis, these methods have been considered in many other areas of cryptography. For instance, [21] examines the use of ML for the identification of encryption algorithms. The objective is to determine the encryption algorithm being used by employing a set of plaintexts and their ciphered versions under several encryption algorithms. Seven encryption methods, each in the EBC and CBC modes, six classification algorithms, and plaintexts in seven distinct languages were employed. Specifically, the paper considers DES, Blowfish, RSA,

ARC4, Rijndael, Serpent, and Twofish as the encryption methods, while the machine learning methods are C4.5, PART, FT, Complement Naive Bayes, Multilayer Perceptron, and WiSARD. The authors find that the classification algorithms are unaffected by the plaintexts' original language, and perform significantly better in ECB than in CBC mode. In ECB mode, the Complement Naive Bayes algorithm displays the best performance with 100% accuracy.

Another prominent application of machine learning in cryptanalysis lies in profiled side-channel analysis [17], where information leaked from a physical cryptosystem is used to break it. Here, convolutional neural networks have shown significant potential, especially by means of adding artificial noise to the input signal, which improves the performance of the neural network and reduces the number of measurements needed to reveal the secret key. These are but a couple of examples of the diverse and promising applications of machine learning in cryptography, from differential cryptanalysis and identification of encryption algorithms to profiled side-channel analysis.

## 5 Conclusion

This survey has shed light on the remarkable progress of machine learning in the field of cryptography, with a particular focus on differential attacks and the utilization of deep learning for distinguishers. These findings emphasize the dynamic and ever-evolving nature of the machine learning domain. As new architectures and algorithms emerge, there is great potential for significant advancements in the performance and efficiency of these distinguishers. The integration of deep learning techniques is a novel approach in the field of cryptanalysis, empowering researchers to tackle complex cryptographic problems and enhancing the security of modern cryptographic systems. As machine learning continues to mature, we can anticipate further breakthroughs, highlighting the importance for researchers and practitioners to remain vigilant and adapt to the evolving landscape of machine learning to stay ahead of potential security threats and leverage the transformative potential of this technology for the betterment of cryptography.

## References

1. Alani, M.M.: Applications of machine learning in cryptography: a survey. In: Proceedings of the 3rd International Conference on Cryptography, Security and Privacy, pp. 23–27 (2019)
2. Băcuieti, N., Batina, L., Picek, S.: Deep neural networks aiding cryptanalysis: a case study of the speck distinguisher. In: Ateniese, G., Venturi, D. (eds.) ACNS 2022. LNCS, pp. 809–829. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-09234-3\\_40](https://doi.org/10.1007/978-3-031-09234-3_40)
3. Baksi, A., Breier, J., Chen, Y., Dong, X.: Machine learning assisted differential distinguishers for lightweight ciphers. In: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 176–181 (2021)

4. Baragada, S., Reddy, P.S.: A survey on machine learning approaches to cryptanalysis. *Int. J. Emerg. Trends Technol. Comput. Sci. (IJETTCS)* **2**(4), 148–153 (2013)
5. Bellini, E., Rossi, M.: Performance comparison between deep learning-based and conventional cryptographic distinguishers. In: Arai, K. (ed.) *Intelligent Computing. LNNS*, vol. 285, pp. 681–701. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-80129-8\\_48](https://doi.org/10.1007/978-3-030-80129-8_48)
6. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. *J. Cryptol.* **4**, 3–72 (1991)
7. Chen, J., Miyaji, A., Su, C., Teh, J.: Improved differential characteristic searching methods. In: *2nd International Conference on Cyber Security and Cloud Computing*, pp. 500–508. IEEE (2015)
8. Chen, Y., Yu, H.: Bridging machine learning and cryptanalysis via EDLCT. *Cryptology ePrint Archive* (2021)
9. Danziger, M., Henriques, M.A.A.: Improved cryptanalysis combining differential and artificial neural network schemes. In: *2014 International Telecommunications Symposium (ITS)*, pp. 1–5 (2014)
10. De Cannière, C., Rechberger, C.: Finding SHA-1 characteristics: general results and applications. In: Lai, X., Chen, K. (eds.) *ASIACRYPT 2006. LNCS*, vol. 4284, pp. 1–20. Springer, Heidelberg (2006). [https://doi.org/10.1007/11935230\\_1](https://doi.org/10.1007/11935230_1)
11. Ferguson, N., Schneier, B.: *Practical Cryptography*, vol. 141. Wiley, New York (2003)
12. Gohr, A.: Improving attacks on round-reduced speck32/64 using deep learning. In: Boldyreva, A., Micciancio, D. (eds.) *CRYPTO 2019. LNCS*, vol. 11693, pp. 150–179. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-26951-7\\_6](https://doi.org/10.1007/978-3-030-26951-7_6)
13. Hou, Z., Ren, J., Chen, S.: Cryptanalysis of round-reduced simon32 based on deep learning. *IACR Cryptology ePrint Archive* **2021**, 362 (2021)
14. Hou, Z., Ren, J., Chen, S.: Improve neural distinguisher for cryptanalysis. *IACR Cryptology ePrint Archive* **2021**, 1017 (2021)
15. Jain, A., Kohli, V., Mishra, G.: Deep learning based differential distinguisher for lightweight cipher present. *IACR Cryptology ePrint Archive* **2020**, 846 (2020)
16. Kim, H., Lim, S., Kang, Y., Kim, W., Seo, H.: Deep learning based cryptanalysis of lightweight block ciphers, revisited. *Cryptology ePrint Archive* (2022)
17. Kim, J., Picek, S., Heuser, A., Bhasin, S., Hanjalic, A.: Make some noise. Unleashing the power of convolutional neural networks for profiled side-channel analysis. *IACR Trans. Cryptographic Hardw. Embed. Syst.* 148–179 (2019)
18. Leurent, G.: Analysis of differential attacks in ARX constructions. In: Wang, X., Sako, K. (eds.) *ASIACRYPT 2012. LNCS*, vol. 7658, pp. 226–243. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-34961-4\\_15](https://doi.org/10.1007/978-3-642-34961-4_15)
19. Leurent, G.: Construction of differential characteristics in ARX designs application to skein. In: Canetti, R., Garay, J.A. (eds.) *CRYPTO 2013. LNCS*, vol. 8042, pp. 241–258. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-40041-4\\_14](https://doi.org/10.1007/978-3-642-40041-4_14)
20. Lu, J., Liu, G., Sun, B., Li, C., Liu, L.: Improved (related-key) differential-based neural distinguishers for SIMON and SIMECK block ciphers. *Comput. J.* (2023)
21. de Mello, F.L., Xexéo, J.A.M.: Identifying encryption algorithms in ECB and CBC modes using computational intelligence. *J. Univers. Comput. Sci.* **24**, 25–42 (2018)
22. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Wu, C.-K., Yung, M., Lin, D. (eds.) *Inscrypt 2011. LNCS*, vol. 7537, pp. 57–76. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-34704-7\\_5](https://doi.org/10.1007/978-3-642-34704-7_5)

23. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why should I trust you?”: explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144. Association for Computing Machinery, New York (2016)
24. Rivest, R.L.: Cryptography and machine learning. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) ASIACRYPT 1991. LNCS, vol. 739, pp. 427–439. Springer, Heidelberg (1993). [https://doi.org/10.1007/3-540-57332-1\\_36](https://doi.org/10.1007/3-540-57332-1_36)
25. Ruzhentsev, V., Levchenko, R., Fediushyn, O.: Cryptanalysis of simple substitution-permutation cipher using artificial neural network. In: 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), pp. 631–634 (2020)
26. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley, Hoboken (2007)
27. So, J.: Deep learning-based cryptanalysis of lightweight block ciphers. Secur. Commun. Netw. **2020**, 1–11 (2020)
28. Stinson, D.R., Paterson, M.: Cryptography: Theory and Practice. CRC Press, Boca Raton (2018)
29. Sun, S., et al.: Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. Cryptology ePrint Archive (2014)
30. Wang, X., Yu, H., Yin, Y.L.: Efficient collision search attacks on SHA-0. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 1–16. Springer, Heidelberg (2005). [https://doi.org/10.1007/11535218\\_1](https://doi.org/10.1007/11535218_1)
31. Yadav, T., Kumar, M.: Differential-ML distinguisher: machine learning based generic extension for differential cryptanalysis. In: Longa, P., Ràfols, C. (eds.) LATINCRYPT 2021. LNCS, vol. 12912, pp. 191–212. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-88238-9\\_10](https://doi.org/10.1007/978-3-030-88238-9_10)
32. Zhao, H., Han, G., Wang, L., Wang, W.: MILP-based differential cryptanalysis on round-reduced midori64. IEEE Access **8**, 95888–95896 (2020)
33. Zolfaghari, B., Koshiba, T.: The dichotomy of neural networks and cryptography: war and peace. Appl. Syst. Innov. **5**(4), 61 (2022)