cryptography

MDPI

Article

MIND-Crypt: A Machine Learning Framework for Assessing the Indistinguishability of Lightweight Block Ciphers Across Multiple Modes of Operation †

Jimmy Dani *, Kalyan Nakka and Nitesh Saxena

Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA; kalyan@tamu.edu (K.N.); nsaxena@tamu.edu (N.S.)

* Correspondence: danijy@tamu.edu

† This article is an extended version of our paper presented at the 22nd Annual International Conference on Privacy, Security, and Trust (PST2025) on 26–28 August 2025.

Abstract

Indistinguishability is a fundamental principle of cryptographic security, crucial for securing data transmitted between Internet of Things (IoT) devices. This principle ensures that an attacker cannot distinguish between the encrypted data, also known as ciphertext, and random data or the ciphertexts of two messages encrypted with the same key. This research investigates the ability of machine learning (ML) to assess the indistinguishability property in encryption systems, with a focus on lightweight ciphers. As our first case study, we consider the SPECK32/64 and SIMON32/64 lightweight block ciphers, designed for IoT devices operating under significant energy constraints. In this research, we introduce MIND-Crypt (a Machine-learning-based framework for assessing the INDistinguishability of Cryptographic algorithms), a novel ML-based framework designed to assess the cryptographic indistinguishability of lightweight block ciphers, specifically the SPECK32/64 and SIMON32/64 encryption algorithms in CBC, CFB, OFB, and CTR modes, under Known Plaintext Attacks (KPAs). Our approach involves training ML models using ciphertexts from two plaintext messages encrypted with the same key to determine whether ML algorithms can identify meaningful cryptographic patterns or leakage. Our experiments show that modern ML techniques consistently achieve accuracy equivalent to random guessing, indicating that no statistically exploitable patterns exist in the ciphertexts generated by the considered lightweight block ciphers. Although some models exhibit mode-dependent bias (e.g., collapsing to a single-class prediction in CBC and CFB), their overall accuracy remains at random guessing levels, reinforcing that no meaningful distinguishing patterns are learned. Furthermore, we demonstrate that, when ML algorithms are trained on all possible combinations of ciphertexts for given plaintext messages, their behavior reflects memorization rather than generalization to unseen ciphertexts. Collectively, these findings suggest that existing block ciphers have secure cryptographic designs against ML-based indistinguishability assessments, reinforcing their security even under round-reduced conditions.

Keywords: lightweight block ciphers; cryptanalysis; deep learning

Check for updates

Academic Editor: Rongxing Lu and Ali Ghorbani

Received: 18 December 2025
Revised: 24 January 2026
Accepted: 4 February 2026
Published: 10 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Indistinguishability is the basis for building secure encryption systems. Concretely, indistinguishability means that the adversary cannot tell the difference between the cipher-

texts corresponding to two plaintexts with a probability significantly better than 0.50. It is an important notion underlying encryption security since it implies that the adversaries are unable to decipher any useful information about the plaintext given the ciphertext. Moreover, a broken indistinguishability property exposes deterministic or predictable patterns in the encryption process, making the system susceptible to more effective attacks, such as ciphertext-only attacks where the plaintext is deciphered without the key. This not only undermines the trust and reliability of the cryptographic system but also paves the way for practical decryption techniques that could exploit this predictability. Therefore, preserving indistinguishability is essential to maintain the overall integrity and security of encryption schemes.

Lightweight Block Ciphers. The Internet of Things (IoT) exemplifies a domain where cryptography's vital role is particularly pronounced, due to its explosive growth and the evolving capabilities of connected devices. With projections estimating about 40 billion devices connected by 2030 [1–4], the diversity of applications—from smart home devices enhancing residential convenience and security to advanced systems in healthcare monitoring and industrial IoT (IIoT)—is transforming traditional industries. However, many IoT devices operate under constraints of processing power and memory, necessitating cryptographic solutions that optimize security without imposing significant computational burdens. Among lightweight block ciphers, the SPECK32/64 and SIMON32/64 ciphers, designed by the National Security Agency, stand out for their operational efficiency and simplicity, tailored specifically to meet the needs of these resource-constrained environments [5–7].

Cryptanalysis and Machine Learning. As cryptographic systems evolve in complexity and sophistication, so too does cryptanalysis—the study and practice of deciphering codes, ciphers, and encrypted messages without the use of actual key. This discipline has seen significant advancements through a variety of techniques, reflecting the ongoing arms race between cryptography and cryptanalysis. Traditional methods such as side-channel attacks [8–11], fault injection attacks [12–15], mathematical analysis [16–18], and brute-force attacks [19–22] have continually been refined in tandem with advancements in cryptographic techniques. However, as cryptographic algorithms become more complex, the effectiveness of these traditional approaches is increasingly challenged, necessitating newer methodologies. This evolving landscape has sparked considerable interest in integrating machine learning with cryptanalysis, offering novel approaches to breaking cryptographic systems and presenting new challenges to their robustness.

In 2019, Gohr [23] proposed a differential attack on round-reduced SPECK32/64, focusing on the development of neural distinguishers that could effectively distinguish ciphertexts differing by a specific difference delta from random text. This approach leveraged deep learning (DL), specifically deep residual neural networks, which demonstrated superior performance compared to traditional cryptographic distinguishers. Further enhancing the practicality of his method, Gohr integrated a novel key search policy based on Bayesian optimization, significantly improving the efficiency of key recovery processes. Following Gohr's work, Benamira et al. [24] conducted detailed analysis and showed neural distinguisher developed by Gohr generally relies on the differential distribution on the ciphertext pairs but also on the differential distribution in penultimate and antepenultimate rounds. This approach not only showcased DL's potential in enhancing traditional cryptanalysis but also emphasizes the need to probe deeper into the cipher's behavior by exploring the notion of indistinguishability. Unlike prior research focused primarily on differential cryptanalysis, our approach uniquely targets indistinguishability—an essential property underpinning robust encryption—and systematically assesses it against advanced machine learning methods.

Our research investigates the potential of ML techniques to assess the indistinguishability of lightweight block ciphers, specifically SPECK32/64 and SIMON32/64 under CBC, CFB, OFB, and CTR modes of operation. Compromising indistinguishability renders the cipher fundamentally insecure. This process involves training a deep learning model on ciphertexts from two distinct messages, $\mathcal{P}_1$ and $\mathcal{P}_2$, and aims to determine if a challenge ciphertext belongs to message $\mathcal{P}_1$ or $\mathcal{P}_2$.

Focus of Our Research. In contrast to Gohr [23], our research shifts the focus from differential attack strategies to the broader concept of indistinguishability within lightweight block ciphers (e.g., SPECK32/64 and SIMON32/64). Unlike Gohr's approach, which targets specific, known differential paths for key recovery, our study employs ML to assess whether a model can distinguish between ciphertexts of two plaintext messages encrypted using the same key. Our analysis demonstrates that achieving a generalized ML-based indistinguishability is fundamentally more challenging than exploiting predefined differential characteristics. Consequently, our results highlight that existing lightweight block ciphers remain robust, as current ML methods fail to compromise their indistinguishability.

To illustrate the practical implications of our research, consider a scenario involving a smart home security system that utilizes the SPECK32/64 or SIMON32/64 cipher to encrypt data from sensors such as motion detectors and window sensors. If indistinguishability were compromised, an adversary might differentiate encrypted sensor signals, distinguishing, for instance, whether ciphertext originates from motion sensors detecting indoor movement or window sensors detecting window openings. Such an ability would pose severe privacy risks, enabling unauthorized parties to infer sensitive patterns (e.g., movements), without explicitly decrypting the messages.

Formally, in our study, we address the following research question: *Can ML techniques compromise the indistinguishability property of lightweight block ciphers?* Our findings provide strong evidence that current lightweight block cipher implementations are secure against ML-based indistinguishability assessments.

When designing MIND-Crypt, we considered assumptions typical of the Known Plaintext Attack (KPA) scenario, where the attacker has access to both plaintexts and their corresponding ciphertexts encrypted under the same key. Here, the primary focus of an attacker is to identify if the challenge ciphertext belongs to message $\mathcal{P}_1$ or $\mathcal{P}_2$, thus testing the fundamental indistinguishability of the considered encryption schemes. Our objective is not to demonstrate vulnerability but to investigate whether subtle leakages might be exploited by ML. We study both its standard configuration and round-reduced versions to understand if these variations affect resistance to ML.

Our Methodology and Experiments. We approach this challenge by framing the task as a binary classification problem, where the ML classifier is trained on previously known ciphertexts $\mathcal{C}_1$ and $\mathcal{C}_2$ corresponding to two fixed plaintexts $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, and using the trained model to predict whether any new challenge ciphertexts correspond to $\mathcal{P}_1$ or $\mathcal{P}_2$. To train the model, the attacker generates ciphertexts of these messages by encrypting them under the same key.

Our experiments show that the performance of the ML models remains consistently around random guessing levels (≈50%). These findings suggest that ML models are unable to extract meaningful patterns from ciphertexts produced by lightweight encryption schemes. Consequently, our results emphasize that ML techniques, despite their advanced capabilities, cannot challenge the indistinguishability property cryptographic algorithms.

Our Contributions and Summary of Results. The main contributions and findings are summarized as follows:

1. **A Novel Machine Learning Framework:** We designed MIND-Crypt, a novel machine-learning-based framework that utilized ML techniques to investigate the indistinguishability of lightweight block ciphers. More specifically, we leverage DL to implement MIND-Crypt.
2. **Comprehensive Evaluation of Cryptographic Indistinguishability:** We evaluate the cryptographic indistinguishability of SPECK32/64 and SIMON32/64 across four widely used block cipher modes of operation (CBC, CFB, OFB, and CTR) using multiple state-of-the-art DL architectures. Our experiments demonstrate that all evaluated ML models consistently achieve accuracies equivalent to random guessing (≈50%), clearly indicating their inability to detect meaningful cryptographic leakage or statistical patterns.
3. **Analysis of Memorization vs. Generalization:** We provide a detailed analysis distinguishing memorization from generalization in DL model predictions, leveraging reduced-entropy datasets specifically designed to study memorization effects.
4. **Security Assurance for IoT Devices:** Our results provides practical assurance, demonstrating that lightweight block ciphers such as SPECK32/64 and SIMON32/64 are secure against ML-based indistinguishability attacks in realistic, resource-constrained IoT environments.

This article is an extended version of our conference paper presented at the 21st Annual International Conference on Privacy, Security and Trust (PST 2025) [25], where we introduced the MIND-Crypt framework and evaluated it only for the CBC mode of operation. A detailed summary of the extensions in this journal version is provided in Appendix B.

Reproducibility. Our code is publicly available [26].

2. Background and Preliminaries

In this section, we provide important context in the form of basic background on block cipher SPECK32/64, Residual Neural Networks, and Transfer Learning.

2.1. Lightweight Block Ciphers

A block cipher is a deterministic permutation that operates on fixed size blocks of data. Since plaintexts are typically longer than a single block and semantic security is required, block ciphers are used in conjunction with modes of operation. A mode of operation specifies how encryption is applied across multiple blocks and how randomness is incorporated to prevent information leakage. In this work, we consider four widely deployed block cipher modes: Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR) for SPECK32/64 and SIMON32/64 lightweight block ciphers.

2.1.1. SPECK32/64 Block Cipher

SPECK is a family of lightweight block ciphers, denoted as SPECKM/N, where M and N are block size and key size, respectively, in bits, developed by Beaulieu, Treatman-Clark, Shors, Weeks, Smith, and Wingers [27] for NSA. It is an add-rotate-xor (ARX) cipher with operations like modular addition (mod 2^k) $\boxplus$, bitwise addition $\oplus$, and bitwise rotation (left $\ll$ and right $\gg$) applied on k-bit words, aimed to build efficient cipher for software implementations in IoT devices [7]. The round function of SPECK $F : \mathbb{F}_2^{2k} \times \mathbb{F}_2^{2k} \to \mathbb{F}_2^{2k}$ computes the next round state (L_{i+1}, R_{i+1}) using a k-bit subkey K and current round state (L_i, R_i) as $L_{i+1} = ((L_i \gg \alpha) \boxplus R_i) \oplus K$ and $R_{i+1} = (R_i \ll \beta) \oplus L_{i+1}$. Here, α and β are rotation constants ($\alpha = 7, \beta = 2$ for SPECK32/64 and $\alpha = 8, \beta = 3$ for remaining). The ciphertext is produced from the input plaintext by employing this round function for a

fixed number of times (22 rounds for SPECK32/64). Further, the design of SPECK32/64 balances security with minimal computation overhead, making it an ideal candidate for studying indistinguishability in resource-constrained IoT devices [6,7].

2.1.2. SIMON32/64 Block Cipher

SIMON is a family of lightweight block ciphers, denoted as SIMONM/N, where M represents the block size in bits, and N denotes the key size in bits. SIMON was designed by Beaulieu, Shors, Smith, Treatman-Clark, Weeks, and Wingers for the NSA [27], specifically optimized for efficient implementation in hardware-constrained environments, such as embedded systems [7]. SIMON employs a balanced Feistel network structure, particularly suited for hardware efficiency due to its simplicity, minimal gate count, and compact area utilization.

For SIMON32/64, the cipher employs a word size of 16 bits (thus a 32-bit block size) and a 64-bit key. The SIMON32/64 variant uses 32 rounds of encryption, providing adequate security for resource-constrained devices. The minimalistic and serialized design makes it highly suitable for hardware implementations where area minimization and power efficiency are critical, such as embedded IoT platforms [6,7].

3. Threat Model and Assumptions

Our study investigates the security of the SPECK32/64 and SIMON32/64 lightweight block ciphers. We consider a known plaintext passive adversary model on block ciphers instantiated with four modes of operation: CBC, CFB, OFB, and CTR. We primarily focus on an attacker's ability to distinguish between the ciphertexts of two different messages encrypted using the same key. This is particularly relevant for IoT devices that operate under significant energy constraints and require efficient and lightweight cryptographic solutions like the SPECK32/64 or SIMON32/64 cipher.

In our attack model, we consider a passive attack scenario where the attacker observes multiple ciphertexts, all encrypted with the same key, without performing active attacks such as Chosen-Ciphertext Attacks (CCA). To illustrate the practical implications of violating indistinguishability (briefly noted in Section 1) in cryptographic systems, consider a smart home security system that uses the SPECK32/64 or SIMON32/64 lightweight block cipher to encrypt data from various constrained IoT sensors around the house. These sensors—including motion detectors, cameras, and window sensors—regularly send encrypted data to a central monitoring system. Adopting a passive attack scenario enhances the practical relevance of our assessment, as it represents a realistic threat where attackers merely observe ciphertexts without active manipulations, commonly encountered in practical IoT security environments.

Mathematically, we denote the plaintext by $\mathcal{P}$, the ciphertext by $\mathcal{C}$, and the secret key by $\mathcal{K}$. The encryption function $\mathcal{E}_K$ uses the key $\mathcal{K}$ to transform plaintext into ciphertext. A cipher maintains indistinguishability if no polynomial-time adversary can distinguish between the ciphertexts of two different plaintexts encrypted with the same key with a probability significantly better than 0.5.

The attacker selects two different fixed plaintexts, $\mathcal{P}_1$ and $\mathcal{P}_2$ (e.g., "heat" or "cool" commands that adjusts the temperature using thermostat), which are encrypted using the same secret key $\mathcal{K}$, resulting in ciphertexts $\mathcal{C}_1$ and $\mathcal{C}_2$. Subsequently, the attacker employs a DL model, trained with multiple instances of ciphertexts $\mathcal{C}_1$ and $\mathcal{C}_2$. This model is then utilized to classify new challenge ciphertexts, determining whether they correspond to $\mathcal{P}_1$ or $\mathcal{P}_2$, potentially breaching the indistinguishability property of the encryption scheme.

Our model extends these concepts by allowing the attacker to simulate data generation without direct access, avoiding the active manipulation typical of CCA. The attacker aims

to identify patterns, anomalies, or relationships in the ciphertexts that differentiate those corresponding to two distinct, same-byte-length plaintexts. Successfully differentiating ciphertexts beyond chance agreement signifies vulnerabilities in the block cipher, whereas failure to do so would validate the cipher's robustness under passive attack settings.

4. MIND-Crypt: Design and Methodology

In this section, we introduce MIND-Crypt, a machine-learning-based assessment framework designed to evaluate the cryptographic indistinguishability of lightweight block ciphers, specifically SPECK32/64 and SIMON32/64, operating in CBC, CFB, OFB, and CTR modes.

4.1. Framework Design

Our primary objective is to investigate whether ML algorithms can identify statistically meaningful patterns or cryptographic leakage in ciphertexts generated by these lightweight block ciphers. DL models have demonstrated significant promise for solving complex classification problems in cybersecurity, such as malware detection, intrusion detection, and traffic classification. In this study, we utilized multiple DL architectures, namely, Convolution Neural Networks [28] (CNNs), Long-Short Term Memory (LSTM) [29] networks, Bidirectional LSTM (BiLSTM) [30] networks, and Residual Neural Networks (ResNets) [23], to comprehensively evaluate cryptographic indistinguishability of lightweight block ciphers. The details of these DL architectures are as follows:

4.1.1. Convolutional Neural Networks (CNNs)

CNNs, introduced by LeCun et al. [28], can effectively extract hierarchical spatial features from input data via convolutional layers. CNNs leverage multiple convolutional layers to automatically identify hierarchical patterns within the input data, which reduces the reliance on manual feature extraction. Although CNNs have historically been applied extensively in image recognition tasks, their capability to capture subtle local statistical dependencies also makes them well suited for security research. CNNs are highly effective for classification tasks involving structures, grid-like data. These models have successfully improved classification accuracy for security problems such as network intrusion detection [31] and malware analysis [32].

4.1.2. Long-Short Term Memory (LSTM)

LSTMs were introduced by Hochreiter and Schmidhuber [29] are a type of recurrent neural network capable of learning sequential dependencies and long-term temporal patterns. LSTM architectures employ specialized gating mechanisms that include input, output and forget gates to effectively preserve long-range dependencies within sequential data, addressing the vanishing gradient problem common to traditional RNNs. For ciphertext indistinguishability assessment, the sequential characteristics of ciphertext bits are critically important. The intrinsic ability of LSTM networks to capture long-range sequential patterns makes them particularly suitable for analyzing cryptographic ciphertexts generated through block ciphers.

4.1.3. Bidirectional Long-Short Term Memory (BiLSTM)

BiLSTM network architecture proposed by Graves and Schmidhuber [30] enhances traditional LSTM architectures by simultaneously processing input sequences in both forward and backward directions. This bidirectional processing allows BiLSTM networks to leverage past and future context at each point in a given sequence, significantly improving their ability to capture complex dependencies. In cryptographic indistinguishability analysis, the direction-agnostic nature of BiLSTMs may offer additional sensitivity in detecting sub-

tle statistical differences across ciphertext sequences, thereby providing a comprehensive evaluation capability for the presence or absence of cryptographic leakage or patterns.

4.1.4. Residual Neural Networks (ResNets)

He et al. [33] introduced ResNets to address the vanishing gradients problem in deep neural network (DNN) training by utilizing residual blocks. These blocks, featuring stacked convolutional layers with skip connections, allow the network to learn residual functions, focusing on differences rather than complete transformations. ResNets have been successfully applied in various security applications [34–37].

In cryptanalysis, ResNet models are effective at identifying complex patterns, which helps with tasks such as automated cipher breaking and differential cryptanalysis. Their architecture allows for more accurate and efficient prediction of differential characteristics, enhancing encryption analysis and vulnerability insights. A prominent example is the work of Gohr et al. [23], who leveraged deep residual neural networks to identify differential characteristics in round-reduced versions of lightweight block ciphers such as SPECK32/64. Their findings highlighted that ResNets could surpass traditional cryptanalytic methods in specific scenarios involving reduced cipher complexity. Adrien et al. [24] discuss how machine learning, including ResNets, advances cryptanalytic and cyber defense techniques.

4.2. Framework Implementation

Figure 1 illustrates our assessment framework, detailing the entire process from message selection and ciphertext generation to ML-based assessment. Initially, two plaintext messages $\mathcal{P}_1$ and $\mathcal{P}_2$, each measured in byte length and differing by exactly one bit, are encrypted multiple times using either SPECK32/64 or SIMON32/64 ciphers under a fixed encryption key k using CBC, CFB, OFB, and CTR modes. Our DL models are trained for the binary classification task of separating ciphertexts into two classes: ξ_1 and ξ_2. To explain, ξ_1 includes the ciphertexts of $\mathcal{P}_1$, labeled as $\mathcal{C}1_i$ ($\mathcal{C}1_i = Enc_k(\mathcal{P}_1)$), where $i \in \{1, 2, \ldots, n\}$. Similarly, ξ_2 includes the ciphertexts of $\mathcal{P}_2$, labeled as $\mathcal{C}2_i$ ($\mathcal{C}2_i = Enc_k(\mathcal{P}_2)$) for $i \in \{1, 2, \ldots, n\}$. It should be noted that the Initialization Vectors (IVs) are used only as a part of encryption process and not included in the training data of the DL model. This design choice ensures that the model learns to identify any intrinsic properties or subtle differences in the ciphertext generated from $\mathcal{P}_1$ and $\mathcal{P}_2$, without relying on external factor of the IVs.

Following ciphertext generation, we convert the ciphertexts into binary format, adhering to the data preparation methods described by Gohr et al. [23] for examining differential attacks on SPECK32/64. Utilizing this methodology, we feed these binary ciphertexts into a DL model. While Gohr et al. [23] demonstrated the effectiveness of ResNet models in identifying differential characteristics within ciphertexts, their approach primarily leveraged spatial hierarchical features through convolutional residual blocks. To thoroughly assess cryptographic indistinguishability, we employ diverse DL architectures capable of capturing different types of patterns or subtle biases within ciphertext data. Specifically, we selected CNN architectures for their proven efficiency in extracting spatial and local feature patterns. Additionally, we included LSTM and BiLSTM networks due to their capability to detect sequential dependencies and temporal correlations that might remain undetected by purely convolution-based architectures. The combination of spatial (CNN), sequential (LSTM/BiLSTM), and hierarchical (ResNet) learning mechanisms ensures a robust, multi-dimensional analysis, providing comprehensive insights into security of lightweight block ciphers against varied ML-based cryptanalytic approaches. Each DL model in our framework is trained for binary classification to distinguish ciphertexts derived from plaintexts $\mathcal{P}_1$ and $\mathcal{P}_2$.

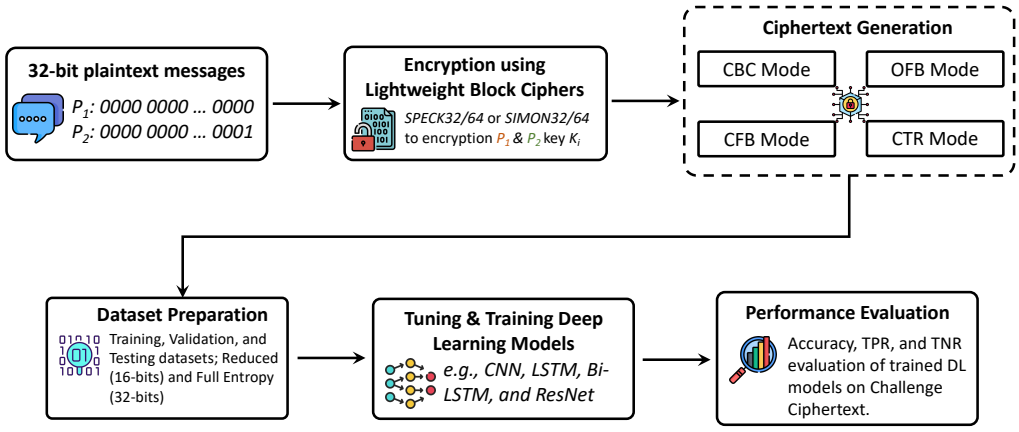

Figure 1. The MIND-Crypt assessment framework—investigating the indistinguishability of SPECK32/64 and SIMON32/64 lightweight block ciphers across four modes of operation (CBC, CFB, OFB, and CTR). Two plaintext messages encrypted under the same key are processed through each mode, generating ciphertext datasets used to train and evaluate deep learning models.

Finally, the trained ML models are evaluated on unseen challenge ciphertext samples. By analyzing model predictions and systematically comparing their performance against a random guessing baseline (≈50% accuracy), we provide empirical insights into whether state-of-the-art ML techniques can uncover meaningful cryptographic vulnerabilities. Rather than demonstrating exploitable weakness, our comprehensive assessment highlight the robustness of lightweight block cipher designs against ML-based indistinguishability attacks.

5. Assessing Lightweight Block Ciphers Using MIND-Crypt

In this section, we describe how our proposed MIND-Crypt framework can be utilized for assessing lightweight block ciphers. We describe the datasets, experiment settings, and evaluation metric considered for assessing our framework.

5.1. Description of the Dataset

In our study, we evaluated the effectiveness of the MIND-Crypt by utilizing a publicly available implementation of SPECK32/64 provided by Gohr [23] and SIMON32/64 implementation [38].

For the CBC mode of SPECK32/64, we retained a modified version of Gohr's original implementation. This choice was made to preserve direct experimental comparability with Gohr's prior results. Maintaining this implementation for CBC of SPECK32/64 ensures that any observed differences between the original work and our extended evaluation can be attributed to the learning framework and experimental design rather than to software level discrepancies.

To this end, we made several modifications to the original SPECK32/64 implementation provided in [23]:

1. **Encryption Mode:** We shifted from the Electronic Code Book (ECB) mode used in Gohr's original code to CBC mode. This change involved encrypting the messages using CBC mode with randomly generated initialization vectors (IVs).
2. **Key Usage:** Unlike the original implementation that used varying keys, we utilized a single, fixed key securely generated using Gohr's methodology. This consistency was vital for comparing the indistinguishability of outputs. This approach allowed us to isolate the impact of message variation on ciphertext indistinguishability without key variability influencing the results.

3. **Generating IVs:** We employed the `frombuffer` module in `NumPy` library in conjunction with Python's `os.urandom` to generate cryptographically secure IVs, mirroring Gohr's method.
4. **Correctness:** To ensure the correctness of our modifications, we decrypted the ciphertexts to verify that they reverted accurately to the original plaintexts, labeled '0' and '1'.
5. **Message Selection:** We chose two specific messages of identical 32-bit length, differing by only a single bit at the binary level, labeled '0' and '1'. This allowed us to directly assess the effect of minimal input variation on the encryption output.

Importantly, both implementations realize the same algorithmic specification of SPECK32/64 and differ only in software structure and experimental interfacing. The round functions, key schedules, and encryption operations are identical. Consequently, the use of two implementations does not introduce cryptographic bias and does not affect the validity of the indistinguishability evaluation.

Additionally, to distinguish between memorization and generalization behaviors exhibited by ML models, we conducted a proof-of-concept evaluation using a simplified cryptographic setup. Specifically, we intentionally reduced the entropy in the SPECK32/64 encryption algorithm from the standard 32 bits to 16 bits. This reduction created an artificially weakened cryptographic scenario, significantly decreasing the complexity and thereby increasing the potential for identifiable statistical patterns. We emphasize that this simplified experiment was conducted solely for analyzing ML model behaviors regarding memorization versus genuine generalization and was not intended as a realistic assessment of the cipher's actual indistinguishability or security under standard cryptographic conditions.

For exploring indistinguishability using DL, we collected 10^7 training samples, 10^6 samples each for validation and testing across '$\mathcal{R}$' rounds of encryption schemes. Each dataset segment maintained an equal number of samples from two classes, representing ciphertexts of two distinct plaintext messages encrypted with the same key. The training data was used to train a DL model, while the testing data was utilized to evaluate the performance of the trained model on an unseen dataset. This allowed the DL model to detect subtle differences in ciphertexts of the selected messages. To facilitate the learning process, the ciphertexts were represented as 32-bit binary vectors, providing a consistent input format for the DL.

5.2. Experiment Settings

The implementation of the MIND-Crypt was conducted using the Python v3.9 programming language, leveraging the open-source library TensorFlow [39] for the development, training, and evaluation of the DL model. To optimize the neural network's hyperparameters, we employed Optuna [40], a software framework designed for efficient and automatic hyperparameter optimization. Specifically, we utilized Optuna's `TPESampler`, which implements the Tree-structured Parzen Estimator (TPE) algorithm—a Bayesian optimization approach that models the objective function using two separate densities to efficiently navigate the hyperparameter search space [41]. The hyperparameter search process was configured to execute up to 100 trials or terminate if the search duration exceeded 200 h for each combination of cipher, mode, and DL architecture. Specifically, hyperparameter optimization was performed independently for SPECK32/64 and SIMON32/64, across all four modes of operation (CBC, CFB, OFB, and CTR), and for each of the four DL architectures (ResNet, CNN, LSTM, and BiLSTM), resulting in comprehensive model tuning tailored to each experimental configuration. The search space for the hyperparameters is detailed Tables 1 and 2.

Table 1. Model-specific hyperparameter search space.

Hyperparameter	LSTM-Based Models (LSTM and BiLSTM)	CNN-Based Model (1D CNN)
No. of LSTM Layers	{2, 3, 4, 5, 6, 7, 8, 9}	–
LSTM Cells in Each Layer	{200, 300, 400, 500}	–
No. of Convolution Layers	–	{2, 3, 4, 5, 6, 7, 8, 9}
No. of Filters	–	{2, 4, 8, 16, 32, 64, 128, 256}
Kernel Size	–	{2, 3, 4, 5, 7, 9, 11, 13, 15, 17, 19, 21}
Convolution Stride Size	–	{2, 3, 4, 5, 7, 9, 11, 13, 15, 17, 19, 21}
Pool Size	–	{1, 2, 3, 4}
Pool Stride Size	–	{2, 3, 4}

Table 2. Common hyperparameter search space across all models.

Hyperparameter	Search Space
decay	{0.05, 0.1, 0.2, 0.3}
Dropout Rate	{0.05, 0.1, 0.2, 0.3, 0.4}
Activation Function	{Softsign, ELU, Selu, ReLU, Tanh}
No. of Dense Layers	{1, 2, 3, 4, 5, 6, 7, 8, 9}
No. of Neurons in FC Layer	{256, 512, 1024, 2048, 4096}
Activation Function in FC Layer	{Softsign, ELU, Selu, ReLU, Tanh}
Dropout Rate FC	{0.05, 0.1, 0.2, 0.3, 0.4}
optimizer	{RMSprop, Adagrad, Adam, Adamax, Nadam, SGD}
Epochs	{100, 200, 300}
Batch Size	{256, 512, 1024}
Learning Rate	[0.00001, 0.01] (log scale)

DL Model Training for Indistinguishability Assessment. To study cryptographic indistinguishability of ciphertexts, we implemented and trained four distinct DL architectures: ResNets [23], CNN, LSTM, and BiLSTM networks. The ResNet architecture developed by Gohr was specifically selected due to its success in identifying differential characteristics in reduced-round versions of SPECK32/64 cipher. We adapted Gohr's ResNet model for our binary classification task. This adaptation aimed to assess whether ML could effectively distinguish ciphertexts generated from two distinct plaintexts, $\mathcal{P}_1$ and $\mathcal{P}_2$, encrypted using same key.

We extended the assessment to include CNN, LSTM, and BiLSTM architectures, commonly employed in image and sequence processing. These models were adapted to process ciphertext data by converting inputs into binary vector representations, facilitating sequential (LSTM/BiLSTM) or spatial (CNN) feature extraction. This methodology ensured a comparative analysis of DL architectures in the context of ciphertext indistinguishability. Detailed architectural specifications and hyperparameter settings for the ResNet model are available in Gohr [23].

5.3. Evaluation Metrics

To evaluate the efficacy of the MIND-Crypt across different settings, we performed a comprehensive assessment using a DL model to classify ciphertexts into two distinct classes, ξ_1 and ξ_2. This evaluation employs three key metrics: Accuracy, True Positive Rate (TPR), and True Negative Rate (TNR), similar to the metrics considered in studies by [23,24] that explore differential attacks in the SPECK32/64 encryption scheme. Furthermore, accuracy, TPR, and TNR were specifically chosen because they collectively provide clear insights into model biases, detection capabilities, and overall effectiveness in distinguishing ciphertext classes.

Accuracy gauges the model's overall effectiveness at correctly classifying ciphertexts belonging to class ξ_1 or ξ_2. We calculate accuracy as the proportion of correct classifi-

cation—both true positives and true negatives—out of the total ciphertexts examined. A higher accuracy value reflects superior model performance in discriminating accurately between ciphertexts associated with classes ξ_1 and ξ_2.

The True Positive Rate (TPR), also known as Recall or Sensitivity, measures the proportion of ciphertext samples that truly belong to the class ξ_1. In our experiments, a higher TPR indicates that the classifier more consistently identifies ciphertext from ξ_1, reducing false negatives and improving the reliability of the empirical distinguishers.

The True Negative Rate (TNR), or Specificity, evaluates the model's accuracy in classifying ciphertexts into class ξ_2 when they do not belong to class ξ_1. This measure is essential for ensuring the model effectively identifies ciphertexts that do not adhere to the characteristics of class ξ_1, thus preventing false positives. A high TNR underscores the model's reliability in excluding non-conforming encryption outputs, pivotal for upholding robust cryptographic defenses.

In addition to these key metrics, we also provided detailed analysis using Precision, Recall, F1-Score, Receiver Operating Characteristic Area Under the Curve (ROC-AUC), False Negative Rate (FNR), and False Positive Rate (FPR) in Appendix A.

6. Results

We evaluated the cryptographic indistinguishability of the SPECK32/64 and SIMON32/64 lightweight block ciphers using four DL architectures: ResNet, CNN, LSTM, and BiLSTM. Experiments were performed under both round-reduced (five-round) and standard full-round configurations. In addition, four modes of operation were examined: CBC, CFB, OFB, and CTR. The resulting classification performance for all combinations is summarized in Tables 3 and 4.

Table 3. Indistinguishability assessment for SPECK32/64 and SIMON32/64 in **round-reduced** (5-round) configuration using MIND-Crypt.

CBC and CFB							
		CBC			**CFB**		
Cipher	**DL Model**	**Accuracy**	**TPR**	**TNR**	**Accuracy**	**TPR**	**TNR**
SPECK32/64	ResNet	0.5000	0.0000	1.0000	0.5003	0.0355	0.9650
	CNN	0.5003	0.0355	0.9650	0.4998	0.0434	0.9562
	LSTM	0.5000	0.0000	1.0000	0.5000	0.0000	1.0000
	BiLSTM	0.5000	0.0000	1.0000	0.5000	0.0000	1.0000
SIMON32/64	ResNet	0.5002	0.4947	0.5057	0.5000	0.0000	1.0000
	CNN	0.4993	0.2235	0.7750	0.4998	0.0470	0.9526
	LSTM	0.5000	0.0008	0.9991	0.5000	0.0000	1.0000
	BiLSTM	0.5000	0.0000	1.0000	0.4999	0.4481	0.5518
OFB and CTR							
		OFB			**CTR**		
Cipher	**DL Model**	**Accuracy**	**TPR**	**TNR**	**Accuracy**	**TPR**	**TNR**
SPECK32/64	ResNet	0.5000	0.4999	0.5000	0.4993	0.2235	0.7750
	CNN	0.5001	0.8012	0.1990	0.5005	0.2062	0.7948
	LSTM	0.5000	0.0000	1.0000	0.5000	0.0002	0.9998
	BiLSTM	0.4997	0.0806	0.9188	0.5000	0.9997	0.0003
SIMON32/64	ResNet	0.5053	0.9991	0.0017	0.4993	0.2235	0.7750
	CNN	0.5000	0.0000	1.0000	0.5000	0.0612	0.9388
	LSTM	0.5000	0.0000	1.0000	0.5000	0.0000	1.0000
	BiLSTM	0.4997	0.0806	0.9188	0.5000	0.0000	1.0000

Across all ciphers, modes, and DL architectures, the experimental results show that classification accuracy remains clustered around random guessing (50%). These results indicate that the ciphertexts produced by the evaluated encryption schemes do not reveal structural features or distinguishing patterns that can be exploited by the DL models.

Table 4. Indistinguishability assessment for SPECK32/64 and SIMON32/64 in **standard** configuration using MIND-Crypt.

CBC and CFB							
		CBC			**CFB**		
Cipher	**DL Model**	**Accuracy**	**TPR**	**TNR**	**Accuracy**	**TPR**	**TNR**
SPECK32/64	ResNet	0.5000	1.0000	0.0000	0.4997	0.9489	0.0505
	CNN	0.4997	0.9489	0.0505	0.5002	0.0826	0.9178
	LSTM	0.5000	0.0000	1.0000	0.5000	0.0000	1.0000
	BiLSTM	0.4999	0.0000	0.9999	0.4999	0.0167	0.9831
SIMON32/64	ResNet	0.5000	1.0000	0.0000	0.4999	0.0720	0.9278
	CNN	0.4999	0.0720	0.9278	0.4999	0.7994	0.2003
	LSTM	0.5000	0.4999	0.5000	0.5000	0.0000	1.0000
	BiLSTM	0.5000	0.9996	0.0004	0.5001	0.9765	0.0237
OFB and CTR							
		OFB			**CTR**		
Cipher	**DL Model**	**Accuracy**	**TPR**	**TNR**	**Accuracy**	**TPR**	**TNR**
SPECK32/64	ResNet	0.5000	0.0000	1.0000	0.5000	0.9996	0.0004
	CNN	0.5002	0.9297	0.0707	0.5002	0.8517	0.1487
	LSTM	0.5000	0.0000	1.0000	0.5000	0.0002	0.9998
	BiLSTM	0.5000	0.0258	0.9741	0.4999	0.9987	0.0012
SIMON32/64	ResNet	0.5000	0.4999	0.5000	0.5002	0.4947	0.5057
	CNN	0.4999	0.9949	0.0049	0.4997	0.9413	0.0581
	LSTM	0.5000	0.0000	1.0000	0.5000	0.0000	1.0000
	BiLSTM	0.5000	0.0000	1.0000	0.5000	1.0000	0.0000

6.1. Round-Reduced Configuration

In the round-reduced configuration, every combination of cipher, mode, and DL architecture exhibits behavior consistent with random guessing (Table 3). Accuracies remain centered around 0.50, while the observed pairs of TPR and TNR are often highly unbalanced, reflecting model bias rather than meaningful discriminative capability.

For SPECK32/64, this behavior remains consistent across the CBC, CFB, OFB, and CTR modes. In particular, under CBC and CFB, the ResNet, LSTM, and BiLSTM models yield TPR near zero and TNR near one, revealing that the models have collapsed to trivial single class predictions even though their overall accuracy remains close to 0.50. CNN shows small deviations in CBC and CFB with TPR values around 0.03 to 0.04 and TNR values around 0.95 to 0.97, but these deviations are too small to suggest significant structure. Similar behavior appears in OFB and CTR, where accuracy remains at random guessing levels, and the imbalance between TPR and TNR only reflects changes in prediction bias rather than improved distinguishability.

For SIMON32/64, the results closely parallel those of SPECK32/64. In all four modes, overall accuracy remains at 0.50 for every DL model. ResNet produces roughly balanced TPR and TNR values near 0.5 across CBC, CFB, OFB, and CTR, consistent with purely random behavior. CNN shows slightly elevated TPR values around 0.22 to 0.25 in CBC and CFB, but these are offset by TNR values around 0.75 to 0.78, resulting in an accuracy of 0.50. LSTM and BiLSTM again collapse to deterministic predictions with TPR near zero

and TNR near one or the reverse, maintaining random guessing accuracy. The OFB and CTR experiments confirm that changing the mode of operation does not provide additional distinguishing power in the round-reduced setting.

Overall, these round-reduced experiments show that the DL models cannot exploit the reduced round structure of SPECK32/64 or SIMON32/64 under any mode of operation. Any increase in TPR is always offset by a corresponding decrease in TNR, resulting in accuracy near 0.50.

6.2. Standard Configuration

In the standard (full-round) configuration, the DL models continue to show no meaningful distinguishing advantage (Table 4).

For SPECK32/64, the ResNet model achieves an accuracy of approximately 0.50 across CBC, CFB, OFB, and CTR, but with TPR near one and TNR near zero, meaning that the model predicts every ciphertext as belonging to the same class. This complete bias invalidates the appearance of a high TPR as evidence of distinguishing capability. The CNN model behaves similarly in CBC and CFB, where its TPR is around 0.95 and its TNR is around 0.05, again yielding an accuracy of 0.50. The LSTM and BiLSTM models display the complementary bias, with TPR near zero and TNR near one or the reverse, confirming that they also resort to trivial classification. The OFB and CTR modes do not alter this pattern: accuracy remains at random guessing levels accompanied by highly skewed TPR and TNR pairs.

For SIMON32/64, the standard configuration results match those for SPECK32/64. ResNet yields TPR near one and TNR near zero across all four modes, with accuracy fixed at 0.50, indicating a systematic preference for a single ciphertext class. CNN shows slightly higher TNR values than in the SPECK case, approximately 0.93 in several modes, but its TPR remains low (around 0.07 to 0.08), keeping the accuracy at 0.50. LSTM and BiLSTM exhibit skew patterns similar to those observed in SPECK32/64 but again never exceed the accuracy expected from random guessing. As before, switching from CBC to CFB, OFB, or CTR produces no meaningful improvement.

Overall, these standard configuration experiments confirm that full-round SPECK32/64 and SIMON32/64 remain indistinguishable from random from the perspective of the DL architectures evaluated.

6.3. Summary of Indistinguishability Results

Across all the evaluated modes of operation (CBC, CFB, OFB, and CTR), no measurable distinguishing advantage is observed for any DL architectures under either the round-reduced or standard configurations. While certain models exhibit strong prediction bias towards a single class in specific configuration, such behavior consistently results in compensating trade-offs between TPR and TNR, confirming such bias does not constitute meaningful distinguishability. Importantly, no mode yields simultaneous improvement in both metrics, which is required for a DL model to be a successful distinguisher.

Overall, these results confirm that the choice of mode of operation does not introduce exploitable statistical structures detectable by modern DL models under KPA settings.

7. Discussions

Our experimental results consistently show that ML models fail to surpass random guessing when distinguishing ciphertexts produced by lightweight block ciphers. To better understand these results, we conducted detailed analysis exploring whether models genuinely learn cryptographic patterns or merely memorize overlapping ciphertext samples.

Analysis of Memorization vs. Generalization: Why ML models Fail to Identify Patterns. Lightweight block ciphers such as SPECK32/64 and SIMON32/64 generate 32-bit ciphertexts, producing approximately 2^{32} (over 4 billion) possible ciphertext outputs for a given plaintext message under full entropy conditions. Exhaustively analyzing such an enormous dataset to detect cryptographic leakage or statistical patterns is computationally prohibitive and practically infeasible due to extensive resources required. Therefore, to conduct computationally manageable evaluation, we intentionally restricted randomness of the initialization vectors (IVs) to 16 bits. Since ciphertext variability directly depends on IV randomness, this restriction reduced the ciphertext space to approximately 2^{16} (65,536) unique ciphertexts, creating a controlled yet meaningful experimental scenario to test if ML models genuinely learn or merely memorize ciphertext patterns.

In our primary experiments with SPECK32/64, we observed that when ML models were trained on datasets containing extensive oversampling—intentional duplication of ciphertext samples to explicitly test memorization capabilities of ML models—the models achieved nearly 99% accuracy. This accuracy reflects memorization of duplicate ciphertext entries rather than genuine generalization. Conversely, when models were trained with only a limited number of unique ciphertext pairs without extensive duplication, accuracy dropped sharply to approximately random guessing (≈50%) when evaluated on unseen ciphertext pairs. However, these models could still correctly classify ciphertext pairs exactly matching those in the training set, further underscoring the effect of memorization.

To systematically investigate memorization versus genuine generalization in DL models, we performed a detailed analysis on datasets generated with 16-bit IV randomness. Our training dataset comprises 800,000 ciphertext samples, with an equal split (400,000 each) between ciphertexts of $\mathcal{P}_1$ and $\mathcal{P}_2$. Within these samples, $\mathcal{P}_1$ has 65,395 unique ciphertexts, while $\mathcal{P}_2$ has 65,375 unique ciphertexts. The testing dataset contains a total of 100,000 ciphertexts samples, equally distributed between $\mathcal{P}_1$ and $\mathcal{P}_2$. Specifically, ciphertexts corresponding to $\mathcal{P}_1$ include 34,974 unique samples, and those corresponding to $\mathcal{P}_2$ include 35,049 unique samples, resulting in combined total of 70,023 unique ciphertexts in the test set.

In our controlled experiment with reduced entropy (16-bits instead of 32-bits), we selected subsets containing 5000 ciphertext samples per class (10,000 samples in total) from the training dataset. Within this subset, $\mathcal{P}_1$ had 4819 unique ciphertexts, and $\mathcal{P}_2$ had 4815 unique ciphertexts, with 366 redundant samples. Upon examining overlaps between training subset and the complete testing dataset, we identified 5307 overlapping ciphertext samples. Specifically, 2659 samples of $\mathcal{P}_1$ and 2684 samples of $\mathcal{P}_2$ appeared in both training and testing datasets, constituting approximately 5% overlap. Such overlaps are crucial, as they directly enable memorization effects by allowing the model to recognize previously encountered samples.

Evaluating the DL model trained on these subsets, we obtained an overall accuracy of about 53.72%. The cross-validation accuracy was around 52.6%, slightly above random guessing (50%), indicating a minimal memorization effect. To further clarify whether the model's performance resulted from genuine generalization or memorization, we conducted detailed sample-by-sample analysis. Among the 70,023 unique ciphertexts samples in the testing dataset, the model correctly classified 53.58% of them. However, when isolating samples unique only to the testing dataset (thus excluding overlapping training samples), the accuracy sharply dropped to 49.90%, equivalent to random guessing.

This analysis conclusively demonstrates that ML models fail to identify meaningful cryptographic patterns or statistically exploitable leakage under artificially simplified cryptographic conditions. The observed marginal improvements in accuracy above random

chance are entirely due to memorization of overlapping ciphertext samples, rather than genuine generalization by the ML algorithm.

Overall, the inability of state-of-the-art ML models to surpass random guessing underscores not a deficiency of ML techniques but rather highlights the inherent robustness and strength of cryptographic indistinguishability within lightweight block cipher designs.

8. Related Work

Linear and Differential Cryptanalysis. Albrecht et al. [42] introduced a unified framework that synergistically incorporates various differential cryptanalysis techniques, including standard, truncated, and impossible differentials. These methods are particularly effective in extending the capabilities of known attacks against lightweight block ciphers such as KATAN-32. Following a similar thematic exploration, Dinur et al. [43] and Blondeau et al. [44] refined differential cryptanalysis techniques specifically for a round-reduced version of SPECK, highlighting potential weaknesses of these ciphers under constrained operational conditions. In parallel, Ashur et al. [45] examined the SPECK cipher using linear cryptanalysis, revealing vulnerabilities across various block sizes and demonstrating that linear approximations could be exploited to undermine the cipher's integrity. Complementing these analyses, Biryukov et al. [46] developed a branch-and-bound method that identifies linear and differential trails in ARX-based ciphers. They specifically applied this approach to enhance cryptanalytic attacks against SPECK. Further studies on the operational constraints of these ciphers also support these findings [42,47].

ML for Cryptanalysis. Classical cryptanalysis methods, deeply rooted in the mathematical underpinnings of cryptographic algorithms and ciphertexts, Sabaawi et al. [16] extended these traditional techniques by surveying cryptanalysis implementation on ciphers like Caesar, transposition, and Hill. Simultaneously, Khoirom et al. [48] proposed an image encryption scheme based on elliptic curve cryptography and chaotic maps. Their work identified vulnerabilities in the original scheme, leading to an improved version resilient to chosen-plaintext attacks, differential attacks, and statistical attacks, thereby enhancing security and performance in image encryption. This comprehensive exploration spans classical and contemporary approaches, highlighting the evolving landscape of cryptographic techniques for heightened security across diverse applications.

Sikdar et al. [20] conducted a survey on recent cryptanalysis techniques, including brute-force attacks, exploring the growing influence of machine learning in cryptographic methods and suggesting future research directions. Verma et al. [21] delved into the historical significance of brute-force attacks in cybersecurity, emphasizing their enduring relevance for unauthorized data access. Additionally, Mok et al. [22] proposed an intelligent brute-force attack targeting the RSA cryptosystem, simulating and evaluating the effectiveness of their approach in terms of time required for RSA key recovery. Collectively, these works contribute to the understanding and evolution of brute-force cryptanalysis, addressing its challenges and exploring avenues for improved security measures.

While considering side-channel cryptanalysis methods, which focus on the physical characteristics and behaviors of cryptographic devices or implementations, Zhou et al. [8] provided a comprehensive survey covering methods, techniques, and countermeasures in side-channel attacks, evaluating their feasibility and applicability. In a complementary study, Randolph et al. [9] present an in-depth tutorial on power side-channel analysis, spanning the past two decades. The study elucidates fundamental concepts and practical applications of various attacks, such as Simple Power Analysis (SPA), Differential Power Analysis (DPA), Template Attacks (TA), Correlation Power Analysis (CPA), Mutual Information Analysis (MIA), and Test Vector Leakage Assessment (TVLA), along with

the underlying theories. Additionally, the introduction of test statistics as a measure of confidence in detecting side-channel leakage adds depth to these analyses.

Mehmood et al. [49] conducted a comprehensive evaluation of distinguishability on the ciphertexts of AES-128 cipher in CBC and ECB modes. Their methodology involved employing Support Vector Machine, k-Nearest Neighbours, and Random Forest classifiers trained on the frequency distribution of characters in the ciphertexts. The results underscored the susceptibility of the ECB mode, thereby emphasizing the need for robust encryption techniques. Building upon this foundation, Hu et al. [50] explored further by applying Random Forest classifiers to diverse block ciphers, reinforcing the vulnerability of the ECB mode. These studies not only showcase the evolving landscape of machine-learning-based cryptanalysis but also highlight its role in ensuring the resilience of cryptographic algorithms.

Xiao et al. [18] significantly contributed to the field of neural network (NN)-based cryptanalysis by introducing a novel approach that not only focuses on the development of neural distinguishers but also emphasizes metrics for efficacy assessment. Their framework, applied to Cyber-Physical System (CPS) ciphers, adds depth to the understanding of NN-based cryptanalysis.

In summary, while the reviewed literature presents a comprehensive understanding of various cryptanalysis methods, it is noteworthy that the majority of the approaches explores differential attacks, statistical attacks, chosen-plaintext attacks, etc. In contrast to prior research, our work addresses a critical gap in the literature and provides a more comprehensive evaluation of the cryptographic indistinguishability of lightweight block ciphers.

9. Conclusions

In this research, we introduced a ML-based framework, MIND-Crypt, designed specifically to assess the cryptographic indistinguishability of SPECK32/64 and SIMON32/64 lightweight block ciphers. Our investigation utilized various state-of-the-art DL architectures to assess these ciphers using ML.

Our results show that DL models fail to surpass random guessing accuracy ($\approx$50%) in distinguishing ciphertexts of two plaintext messages, $\mathcal{P}_1$ and $\mathcal{P}_2$, encrypted using same key. Our analysis for memorization versus generalization evaluations further revealed that ML models were memorizing ciphertext samples rather than genuinely learning cryptographic patterns. Even in artificially simplified cryptographic environments with deliberately reduced entropy, ML algorithms exhibited no ability to generalize beyond memorized ciphertexts.

These results provide strong empirical evidence that current ML algorithms, despite their advanced pattern-recognition capabilities, remain ineffective in compromising the indistinguishability property of even lightweight cryptographic algorithms. Future research directions could focus on exploring emerging cryptographic algorithms, advanced ML architectures, or quantum-inspired ML methods to monitor and validate cryptographic resilience.

Author Contributions: Conceptualization, J.D. and N.S.; Methodology, J.D., K.N. and N.S.; Software, J.D. and K.N.; Validation, J.D., K.N. and N.S.; Formal analysis, J.D. and K.N.; Investigation, J.D., K.N. and N.S.; Resources, J.D. and K.N.; Data curation, J.D. and K.N.; Writing—original draft, J.D., K.N. and N.S.; Writing—review & editing, J.D., K.N. and N.S.; Supervision, N.S.; Project administration, N.S.; Funding acquisition, N.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science Foundation grant numbers CNS-2154507, OAC-2139358 and CNS-2201465.

Data Availability Statement: The data presented in this study are available in Simon_Speck_Ciphers at https://github.com/inmcm/Simon_Speck_Ciphers (accessed on 3 February 2026), and deep_speck at https://github.com/agohr/deep_speck (accessed on 3 February 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DL	Deep Learning
ML	Machine Learning
IoT	Internet of Things
KPA	Known Plaintext Attack
CCA	Chosen-Ciphertext Attack
TPE	Tree-structured Parzen Estimator
CNN	Convolutional Neural Network
LSTM	Long-Short Term Memory
BiLSTM	Bidirectional Long-Short Term Memory
CBC	Cipher Block Chaining Mode
CFB	Cipher Feedback Mode
OFB	Output Feedback Mode
CTR	Counter Mode
IV	Initialization Vector

Appendix A. Additional Metrics for the Round-Reduced and Standard Configurations

In this section, we provide performance analysis of the MIND-Crypt framework using additional evaluation metrics beyond accuracy, TPR, and TNR. We report Precision, Recall, F1-Score, ROC-AUC, False Positive Rate (FPR), and False Negative Rate (FNR) for both round-reduced and standard configurations of SPECK32/64 and SIMON32/64 ciphers across all evaluated modes of operation (CBC, CFB, OFB, and CTR) and DL architectures (ResNet, CNN, LSTM, and BiLSTM).

Appendix A.1. Round-Reduced Configuration

Table A1 presents additional performance metrics for the round-reduced configuration. The ROC-AUC values consistently cluster around 0.5 across all evaluated cipher-mode and DL architecture combinations, confirming that the models achieve no better than random classification performance. This metric is particularly significant in cryptographic assessment, as ROC-AUC values approaching 0.5 indicate that the classifier cannot distinguish between the two ciphertext classes across any decision threshold.

Table A1. Performance metrics for different modes of operation (CBC, CFB, OFB, and CTR) across ciphers and DL models in **round-reduced** (5-round) configuration.

Mode	Cipher	DL Model	Accuracy	Precision	F1-Score	ROC-AUC	TPR/Recall	TNR	FPR	FNR
CBC	SPECK32/64	ResNet	0.5000	0.0000	0.0000	0.5008	0.0000	1.0000	0.0000	1.0000
		CNN	0.5003	0.5043	0.0665	0.5005	0.0355	0.9650	0.0350	0.9644
		LSTM	0.5000	0.0000	0.0000	0.5014	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.5000	0.0000	0.0000	0.5000	0.0000	1.0000	0.0000	1.0000
	SIMON32/64	ResNet	0.5002	0.5002	0.4974	0.5003	0.4947	0.5057	0.4943	0.5053
		CNN	0.4993	0.4985	0.3086	0.4992	0.2235	0.7750	0.2250	0.7765
		LSTM	0.5000	0.5053	0.0017	0.4996	0.0008	0.9991	0.0017	0.9992
		BiLSTM	0.5000	0.0000	0.0000	0.4991	0.0000	1.0000	0.0000	1.0000

Table A1. *Cont.*

Mode	Cipher	DL Model	Accuracy	Precision	F1-Score	ROC-AUC	TPR/Recall	TNR	FPR	FNR
CFB	SPECK32/64	ResNet	0.5003	0.5043	0.0665	0.5005	0.0355	0.9650	0.0350	0.9644
		CNN	0.4998	0.4978	0.0798	0.5000	0.0434	0.9562	0.0438	0.9566
		LSTM	0.5000	0.0000	0.0000	0.4991	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.5000	1.0000	0.0000	0.4996	0.0000	1.0000	0.0000	1.0000
	SIMON32/64	ResNet	0.5000	0.0000	0.0000	0.4991	0.0000	1.0000	0.0000	1.0000
		CNN	0.4998	0.4979	0.0858	0.5008	0.0470	0.9526	0.0474	0.9530
		LSTM	0.5000	0.0000	0.0000	0.4997	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.4999	0.4999	0.4726	0.4998	0.4481	0.5518	0.4482	0.5519
OFB	SPECK32/64	ResNet	0.5000	0.6667	0.0000	0.5004	0.4999	0.5000	0.5000	0.5001
		CNN	0.5001	0.5000	0.6158	0.5003	0.8012	0.1990	0.8010	0.1988
		LSTM	0.5000	0.0000	0.0000	0.4999	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.4997	0.4982	0.1387	0.4988	0.0806	0.9188	0.0812	0.9194
	SIMON32/64	ResNet	0.5053	0.0009	0.4996	0.0008	0.9991	0.0017	0.9983	0.0009
		CNN	0.5000	0.0000	0.0000	0.4999	0.0000	1.0000	0.0000	1.0000
		LSTM	0.5000	0.5500	0.0000	0.4987	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.4997	0.4982	0.1387	0.4988	0.0806	0.9188	0.0812	0.9194
CTR	SPECK32/64	ResNet	0.4993	0.4985	0.3086	0.4992	0.2235	0.7750	0.2250	0.7765
		CNN	0.5005	0.5011	0.2921	0.5002	0.2062	0.7948	0.2052	0.7938
		LSTM	0.5000	0.4877	0.0004	0.5007	0.0002	0.9998	0.0002	0.9998
		BiLSTM	0.5000	0.5000	0.6666	0.5007	0.9997	0.0003	0.9997	0.0003
	SIMON32/64	ResNet	0.4993	0.4985	0.3086	0.4992	0.2235	0.7750	0.2250	0.7765
		CNN	0.5000	0.5002	0.1091	0.5005	0.0612	0.9388	0.0612	0.9388
		LSTM	0.5000	0.0000	0.0000	0.5009	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.5000	0.0000	0.0000	0.4999	0.0000	1.0000	0.0000	1.0000

The F1-Scores remain significantly low across all configurations, demonstrating that no model achieves stable, balanced classification performance on both ciphertext classes. The FPR and FNR values exhibit a consistent trade-off pattern: any reduction in one error type is systematically offset by a proportional increase in the complementary error type. This symmetrical error distribution is characteristic of random classification behavior and reinforces the conclusion that DL models cannot extract statistically exploitable patterns from round-reduced ciphertexts.

Appendix A.2. Standard Configuration

Table A2 presents additional performance metrics for the standard full-round configuration of SPECK32/64 (22 rounds) and SIMON32/64 (32 rounds). The standard configuration demonstrates behavior consistent with the round-reduced analysis, providing evidence that the additional rounds in full-specification implementations do not introduce exploitable patterns detectable by machine learning.

Notably, the standard configuration exhibits the same fundamental characteristics observed in round-reduced variants: ROC-AUC values remain at approximately 0.5, prediction biases persist across multiple model-mode combinations, and F1-Scores remain substantially low. These statistics indicate that the diffusion and confusion mechanisms inherent in the cipher design operate effectively at both reduced and standard configuration.

The standard configuration shows marginally different bias patterns compared to round-reduced variants, with some mode and DL architecture combinations exhibiting opposing prediction tendencies. However, these differences represent variations in arbitrary model behavior rather than improved distinguishing capability, as evidenced by unchanged accuracies.

Table A2. Performance metrics for different modes of operation (CBC, CFB, OFB, and CTR) across ciphers and DL models in **standard** configuration.

Mode	Cipher	DL Model	Accuracy	Precision	F1-Score	ROC-AUC	TPR/Recall	TNR	FPR	FNR
CBC	SPECK32/64	ResNet	0.5000	0.5000	0.6667	0.5001	1.0000	0.0000	1.0000	0.0000
		CNN	0.4997	0.4999	0.6548	0.4996	0.9489	0.0505	0.9494	0.0511
		LSTM	0.5000	0.0000	0.0000	0.5001	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.4999	0.0000	0.0000	0.5003	0.0000	0.9999	0.0000	1.0000
	SIMON32/64	ResNet	0.5000	0.5000	0.6667	0.5000	1.0000	0.0000	1.0000	0.0000
		CNN	0.4999	0.4998	0.1260	0.5000	0.0720	0.9278	0.0722	0.9280
		LSTM	0.5000	0.6667	0.0000	0.5004	0.4999	0.5000	0.5000	0.5001
		BiLSTM	0.5000	0.5295	0.0010	0.5003	0.9996	0.0004	0.9996	0.0004
CFB	SPECK32/64	ResNet	0.4997	0.4999	0.6548	0.4996	0.9489	0.0505	0.9494	0.0511
		CNN	0.5002	0.5014	0.1419	0.4997	0.0826	0.9178	0.0822	0.9174
		LSTM	0.5000	0.0000	0.0000	0.4999	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.4999	0.4974	0.0323	0.4994	0.0167	0.9831	0.0169	0.9833
	SIMON32/64	ResNet	0.4999	0.4998	0.1260	0.5000	0.0720	0.9278	0.0722	0.9280
		CNN	0.4999	0.4999	0.6152	0.4996	0.7994	0.2003	0.7997	0.2006
		LSTM	0.5000	0.0000	0.0000	0.4999	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.5001	0.5000	0.6614	0.4993	0.9765	0.0237	0.9763	0.0235
OFB	SPECK32/64	ResNet	0.5000	0.0000	0.0000	0.4991	0.0000	1.0000	0.0000	1.0000
		CNN	0.5002	0.5001	0.6504	0.5006	0.9297	0.0707	0.9293	0.0703
		LSTM	0.5000	0.0000	0.0000	0.4999	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.5000	0.4994	0.0491	0.4992	0.0258	0.9741	0.0259	0.9742
	SIMON32/64	ResNet	0.5000	0.6667	0.0000	0.5004	0.4999	0.5000	0.5000	0.5001
		CNN	0.4999	0.4999	0.6152	0.4996	0.9949	0.0049	0.9951	0.0051
		LSTM	0.5000	0.0000	0.0000	0.4999	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.5000	0.0000	0.0000	0.4997	0.0000	1.0000	0.0000	1.0000
CTR	SPECK32/64	ResNet	0.5000	0.5295	0.0010	0.5003	0.9996	0.0004	0.9996	0.0004
		CNN	0.5002	0.5001	0.6302	0.5002	0.8517	0.1487	0.8513	0.1483
		LSTM	0.5000	0.4877	0.0004	0.5007	0.0002	0.9998	0.0002	0.9998
		BiLSTM	0.4999	0.5000	0.6664	0.5003	0.9987	0.0012	0.9988	0.0013
	SIMON32/64	ResNet	0.5002	0.5002	0.4974	0.5003	0.4947	0.5057	0.4943	0.5053
		CNN	0.4997	0.4998	0.6529	0.5001	0.9413	0.0581	0.9419	0.0587
		LSTM	0.5000	0.4999	0.0000	0.4999	0.0000	1.0000	0.0000	1.0000
		BiLSTM	0.5000	0.5000	0.6667	0.5008	1.0000	0.0000	1.0000	0.0000

Appendix B. Summary of Extensions over the PST 2025 Conference Version

The journal manuscript contains substantial new contributions compared with conference version [25], including the following:

1. **Expansion from single-mode (CBC) to comprehensive four-mode evaluation:** The conference version focused solely on the CBC mode for both ciphers. In this journal version, we extend the framework to cover all four standard modes of operation (CBC, CFB, OFB, and CTR), for both SPECK32/64 and SIMON32/64, and for multiple DL architectures in both round-reduced and full-round configurations.
2. **Analysis of mode-dependent bias in ML models:** Beyond reporting aggregate accuracy, we provide detailed, mode-by-mode analysis of classifier behavior, including TPR/TNR, FPR/FNR, and ROC-AUC for each cipher mode and mode combination. We identify and discuss mode-dependent biases, for example, cases where a model collapses to predicting a single class in certain modes or exhibits asymmetric error patterns between "ciphertext" and "random" classes.
3. **Enhanced methodological rigor with mode-specific hyperparameter optimization:** The journal manuscript introduces mode-specific and cipher-specific hyperparameter optimization to ensure that each DL architecture is trained under settings tailored to the underlying data distribution (e.g., learning rates, batch sizes, epochs, dropout rates, etc.). This represents a methodological refinement over the conference version, which used more uniform training configurations. We also provided additional details on

dataset generation and evaluation metrics, thereby strengthening both reproducibility and the credibility of our findings.

References

1. State of IoT 2024: Number of Connected IoT Devices Growing 13% to 18.8 Billion Globally. Available online: https://iot-analytics.com/number-connected-iot-devices/ (accessed on 3 February 2026).
2. The Software Strategies Blog. Cisco Internet of Things (IoT) Study, 2024. Available online: https://softwarestrategiesblog.com/tag/cisco-internet-of-things-iot-study/#:~:text=The%20global%20Internet%20of%20Things,B%20in%20global%20IoT%20spending (accessed on 15 August 2024).
3. Statista. Internet of Things (IoT)—Statistics & Facts, 2024. Available online: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (accessed on 15 August 2024).
4. The Future of IoT Development: Trends and Predictions for 2025. Available online: https://imaginovation.net/blog/iot-development-trends-predictions/ (accessed on 3 February 2026).
5. Internet of Things | TechCrunch—Techcrunch.com. Available online: https://techcrunch.com/tag/internet-of-things/ (accessed on 3 February 2026).
6. Appel, M.; Bossert, A.; Cooper, S.; Kußmaul, T.; Löffler, J.; Pauer, C.; Wiesmaier, A. Block ciphers for the iot–simon, speck, katan, led, tea, present, and sea compared. *Proc. Appel Block CF* **2016**, 1–37.
7. Beaulieu, R.; Shors, D.; Smith, J.; Treatman-Clark, S.; Weeks, B.; Wingers, L. SIMON and SPECK: Block Ciphers for the Internet of Things. Cryptology ePrint Archive, Paper 2015/585, 2015. Available online: https://eprint.iacr.org/2015/585 (accessed on 3 February 2026).
8. Zhou, Y.; Feng, D. Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing. Cryptology ePrint Archive, Paper 2005/388, 2005. Available online: https://eprint.iacr.org/2005/388 (accessed on 3 February 2026).
9. Randolph, M.; Diehl, W. Power side-channel attack analysis: A review of 20 years of study for the layman. *Cryptography* **2020**, *4*, 15. [CrossRef]
10. Kelsey, J.; Schneier, B.; Wagner, D.; Hall, C. Side channel cryptanalysis of product ciphers. In *Proceedings of the Computer Security—ESORICS 98*; Quisquater, J.J., Deswarte, Y., Meadows, C., Gollmann, D., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; pp. 97–110.
11. Phan, R.C.W.; Yen, S.M. Amplifying side-channel attacks with techniques from block cipher cryptanalysis. In *Proceedings of the Smart Card Research and Advanced Applications: 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006, Tarragona, Spain, 19–21 April 2006*; Proceedings 7; Springer: Berlin/Heidelberg, Germany, 2006; pp. 135–150.
12. Dutertre, J.M.; Fournier, J.J.; Mirbaha, A.P.; Naccache, D.; Rigaud, J.B.; Robisson, B.; Tria, A. Review of fault injection mechanisms and consequences on countermeasures design. In Proceedings of the 2011 6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), Athens, Greece, 6–8 April 2011; pp. 1–6. [CrossRef]
13. Clark, J.A.; Jacob, J.L. Fault injection and a timing channel on an analysis technique. In *Proceedings of the Advances in Cryptology—EUROCRYPT 2002: International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, 28 April–2 May 2002*; Proceedings 21; Springer: Berlin/Heidelberg, Germany, 2002; pp. 181–196.
14. Barenghi, A.; Breveglieri, L.; Koren, I.; Naccache, D. Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. *Proc. IEEE* **2012**, *100*, 3056–3076. [CrossRef]
15. Shao, C.; Zhao, D.; Li, H.; Cheng, S.; Gao, S.; Yang, L. Detection of security vulnerabilities in cryptographic ICs against fault injection attacks based on compressed sensing and basis pursuit. *J. Cryptogr. Eng.* **2023**, *14*, 57–70. [CrossRef]
16. Al-Sabaawi, A. Cryptanalysis of Classic Ciphers: Methods Implementation Survey. In Proceedings of the 2021 International Conference on Intelligent Technologies (CONIT), Hubli, India, 25–27 June 2021; pp. 1–6. [CrossRef]
17. Lone, P.N.; Singh, D.; Stoffová, V.; Mishra, D.C.; Mir, U.H.; Kumar, N. Cryptanalysis and improved image encryption scheme using elliptic curve and affine hill cipher. *Mathematics* **2022**, *10*, 3878. [CrossRef]
18. Xiao, Y.; Hao, Q.; Yao, D.D. Neural cryptanalysis: Metrics, methodology, and applications in CPS ciphers. In Proceedings of the 2019 IEEE Conference on Dependable and Secure Computing (DSC), Hangzhou, China, 18–20 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–8.
19. Gohr, A. Brute Force Cryptanalysis. Cryptology ePrint Archive, Paper 2022/053, 2022. Available online: https://eprint.iacr.org/2022/053 (accessed on 3 February 2026).
20. Sikdar, S.; Kule, M. Recent Trends in Cryptanalysis Techniques: A Review. In *Proceedings of the International Conference on Frontiers in Computing and Systems*; Springer: Singapore, 2022; pp. 209–222.
21. Verma, R.; Dhanda, N.; Nagar, V. Enhancing security with in-depth analysis of brute-force attack on secure hashing algorithms. In *Proceedings of Trends in Electronics and Health Informatics: TEHI 2021*; Springer: Singapore, 2022; pp. 513–522.

22. Mok, C.J.; Chuah, C.W. An Intelligence Brute Force Attack on RSA Cryptosystem. *Commun. Comput. Appl. Math.* **2019**, *1*.
23. Gohr, A. Improving Attacks on Round-Reduced Speck32/64 Using Deep Learning. In *Proceedings of the Advances in Cryptology—CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 2019*; Proceedings, Part II; Springer: Berlin/Heidelberg, Germany, 2019; pp. 150–179. [CrossRef]
24. Benamira, A.; Gerault, D.; Peyrin, T.; Tan, Q.Q. A Deeper Look at Machine Learning-Based Cryptanalysis. Cryptology ePrint Archive, Paper 2021/287, 2021. Available online: https://eprint.iacr.org/2021/287 (accessed on 3 February 2026).
25. Dani, J.; Nakka, K.; Saxena, N. A Machine Learning-Based Framework for Assessing Cryptographic Indistinguishability of Lightweight Block Ciphers . In Proceedings of the 2025 22nd Annual International Conference on Privacy, Security, and Trust (PST), Los Alamitos, CA, USA, 26–28 August 2025; pp. 1–10. [CrossRef]
26. Mind Crypt. Available online: https://sites.google.com/view/mind-crypt(accessed on 3 February 2026).
27. 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, San Francisco, CA, USA, 7–11 June 2015; pp. 1–6.
28. LeCun, Y.; Bengio, Y., Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1998; pp. 255–258.
29. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
30. Graves, A.; Fernández, S.; Schmidhuber, J. Bidirectional LSTM networks for improved phoneme classification and recognition. In *Proceedings of the International Conference on Artificial Neural Networks, Warsaw, Poland, 11–15 September 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 799–804.
31. Li, Y.; Xu, Y.; Liu, Z.; Hou, H.; Zheng, Y.; Xin, Y.; Zhao, Y.; Cui, L. Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. *Measurement* **2020**, *154*, 107450. [CrossRef]
32. Nethala, S.; Chopra, P.; Kamaluddin, K.; Alam, S.; Alharbi, S.; Alsaffar, M. A Deep Learning-Based Ensemble Framework for Robust Android Malware Detection. *IEEE Access* **2025**, *13*, 46673–46696. [CrossRef]
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
34. Sai Chaitanya Kumar, G.; Kiran Kumar, R.; Parish Venkata Kumar, K.; Raghavendra Sai, N.; Brahmaiah, M. Deep residual convolutional neural Network: An efficient technique for intrusion detection system. *Expert Syst. Appl.* **2024**, *238*, 121912. [CrossRef]
35. Abbas, A.; Pano, V.; Mainland, G.; Dandekar, K.R. Radio Modulation Classification Using Deep Residual Neural Networks. In Proceedings of the MILCOM 2022—2022 IEEE Military Communications Conference (MILCOM), Rockville, MD, USA, 28 November–2 December 2022; pp. 311–317.
36. Shu, Y.; Qin, R.; He, Y.; Li, Y.; Jiang, R.; Wu, Z. Deep Residual Neural Networks with Attention Mechanism for Spatial Image Steganalysis. In Proceedings of the 2022 IEEE 24th International Conference on High Performance Computing & Communications; 8th International Conference on Data Science & Systems; 20th International Conference on Smart City; 8th International Conference on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Hainan, China, 18–20 December 2022; pp. 1721–1727.
37. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
38. Simon—Speck Ciphers. Available online: https://github.com/inmcm/Simon_Speck_Ciphers (accessed on 3 February 2026).
39. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
40. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019.
41. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Granada, Spain, 12–15 December 2011; pp. 2546–2554.
42. Albrecht, M.R.; Leander, G. An All-In-One Approach to Differential Cryptanalysis for Small Block Ciphers. *IACR Cryptol. ePrint Arch.* **2012**, *2012*, 401.
43. Dinur, I. Improved Differential Cryptanalysis of Round-Reduced Speck. *IACR Cryptol. ePrint Arch.* **2014**, *2014*, 320.
44. Blondeau, C.; Gérard, B. Multiple Differential Cryptanalysis: Theory and Practice. In *Proceedings of the Fast Software Encryption Workshop, Lyngby, Denmark, 13–16 February 2011*; Springer: Berlin/Heidelberg, Germany, 2011.
45. Ashur, T.; Bodden, D. Linear cryptanalysis of reduced-round speck. In Proceedings of the 37th WIC Symposium on Information Theory in the Benelux (SITB 2016) and 6th Joint WIC/IEEE Symposium on Information Theory and Signal Processing in the Benelux, Louvain-la-Neuve, Belgium, 19–20 May 2016.

46. Biryukov, A.; Velichkov, V.; Corre, Y.L. Automatic Search for the Best Trails in ARX: Application to Block Cipher Speck. In *Proceedings of the Fast Software Encryption Workshop, Bochum, Germany, 20–23 March 2016*; Springer: Berlin/Heidelberg, Germany, 2016.
47. Abed, F.; List, E.; Lucks, S.; Wenzel, J. Differential Cryptanalysis of Round-Reduced Simon and Speck. In *Proceedings of the Fast Software Encryption Workshop, London, UK, 3–5 March 2014*; Springer: Berlin/Heidelberg, Germany, 2014.
48. Khoirom, M.S.; Laiphrakpam, D.S.; Themrichon, T. Cryptanalysis of multimedia encryption using elliptic curve cryptography. *Optik* **2018**, *168*, 370–375. [CrossRef]
49. Mehmood, Z.; Sultan, A.; Khan, F.; Tahir, S. Machine Learning Based Encrypted Content Type Identification. In Proceedings of the 2023 International Conference on Communication Technologies (ComTech), Rawalpindi, Pakistan, 15–16 March 2023; pp. 117–122. [CrossRef]
50. Hu, X.; Zhao, Y. Block ciphers classification based on random forest. *J. Phys. Conf. Ser.* **2019**, *1168*, 032015. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.