A Machine Learning-Based Framework for Assessing Cryptographic Indistinguishability of Lightweight Block Ciphers

Jimmy Dani
Texas A&M University
College Station, TX, USA
danijy@tamu.edu

Kalyan Nakka
Texas A&M University
College Station, TX, USA
kalyan@tamu.edu

Nitesh Saxena
Texas A&M University
College Station, TX, USA
nsaxena@tamu.edu

***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 the two messages encrypted with the same key. This research investigates the ability of machine learning (ML) in assessing 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[1], a novel ML-based framework designed to assess the cryptographic indistinguishability of lightweight block ciphers, specifically the SPECK32/64 and SIMON32/64 encryption algorithm in CBC mode (Cipher Block Chaining), under Known Plaintext Attacks (KPA). Our approach involves training ML models using ciphertexts from two plaintext messages encrypted with 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 exists in the ciphertexts generated by considered lightweight block ciphers. Furthermore, we demonstrate that in ML algorithms with all the possible combinations of the ciphertexts for given plaintext messages 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.

***Index Terms*—Lightweight Block Ciphers, Cryptanalysis, Deep Learning**

I. Introduction

Indistinguishability is the basis for building secure encryption systems. Concretely, indistinguishability means that the adversary can not tell the difference between the ciphertexts 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, stands out for its 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 learn-

[1]We refer to our attack framework as MIND-Crypt which stands for "***M***achine learning based framework for assessing ***IND***istinguishability of ***Crypt***ographic Algorithms."

ing 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 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 assess 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. 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 & 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 ($\approx$50%). These findings suggests 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 cryptographic indistinguishability of popular lightweight block ciphers by leveraging multiple state-of-the-art deep learning architectures, including ResNet, CNN, LSTM, and BiLSTM. Our experiments demonstrate that all evaluated ML models consistently achieve accuracies equivalent to random guessing ($\approx$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.

Reproducibility. Our code is publicly available [25].

II. Background & Preliminaries

A. Lightweight Block Ciphers

1) SPECK32/64 Block Cipher: SPECK is a family of lightweight block ciphers, denoted as SPECKM/N where *M*, *N* are block size and key size respectively in bits, developed by Beaulieu, Treatman-Clark, Shors, Weeks, Smith and Wingers [26] 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 $F : \mathbb{F}_2^{2k} \times \mathbb{F}_2^{2k} \rightarrow \mathbb{F}_2^{2k}$, 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, 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) 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 [26], 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].

III. Threat Model & Assumptions

Our study investigates the security of the SPECK32/64 and SIMON32/64 lightweight block ciphers in CBC mode (Cipher Block Chaining) against Known Plaintext Attacks (KPA). 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 gains excess 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 I) 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 deep learning 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.

IV. MIND-Crypt: Design & 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 Cipher Block Chaining (CBC) mode.

A. Framework Design

Our primary objective is to investigate whether machine learning (ML) algorithms can identify statistically meaningful

patterns or cryptographic leakage in ciphertexts generated by these lightweight block ciphers. Deep learning 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 [27] (CNNs), Long-Short Term Memory (LSTM) [28] networks, Bidirectional LSTM (BiLSTM) [29] networks, and Residual Neural Networks (ResNets) [23] to comprehensively evaluate cryptographic indistinguishability of lightweight block ciphers. The details of these DL architectures is as follows:

1) Convolutional Neural Networks (CNNs): CNNs, introduced by LeCun et al. [27] 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 [30] and malware analysis [31].

2) Long-Short Term Memory (LSTM): LSTMs were introduced by Hochreiter and Schmidhuber [32] 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.

3) Bidirectional Long-Short Term Memory (BiLSTM): BiLSTM network architecture proposed by Graves and Schmidhuber [29] enhancs 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 subtle statistical differences across ciphertext sequences, thereby providing a comprehensive evaluation capability for the presence or absence of cryptographic leakage or patterns.

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.

B. 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 having 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 with CBC mode. Our DL models are trained for 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 for 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 mecha-

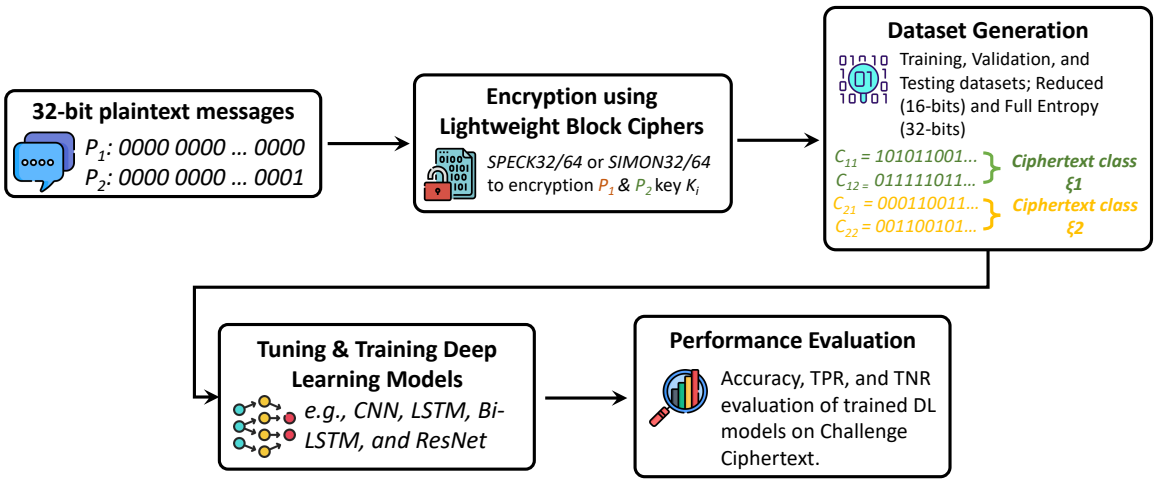

Fig. 1. The MIND-Crypt assessment framework - Investigating the indistinguishability of SPECK32/64 and SIMON32/64 lightweight block ciphers. Two plaintext messages encrypted with the same key, represented in binary format, form the basis for training a DL model.

nisms 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 $\mathcal{P}_2$.

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.

V. 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.

A. 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]. Our objective was to investigate the principle of indistinguishability, which required control over the inputs provided for encryption. In our experimental setup, we aimed to align closely with the methodologies previously established by Gohr, particularly regarding the generation of cryptographic components.

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.

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 with randomly generated initialization vectors (IVs) and applying an XOR operation to the messages before encryption.
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.

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.

B. Experiment Settings

The implementation of the MIND-Crypt was conducted using the Python programming language, leveraging the open-source library TensorFlow [39] for the development, training, and evaluation of the deep learning 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 hours. The search space for the hyperparameters is detailed in Table I and II.

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 machine learning 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]. This work examines the potential of DL techniques to serve as distinguishers, contributing to the broader understanding of cryptographic security in the using machine learning.

C. 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 classification—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.

True Positive Rate (TPR), or sensitivity, specifically measures the model's precision in identifying ciphertexts that genuinely belong to class ξ_1. This metric is crucial for cryptographic applications as it reflects the model's ability to capture the unique characteristics expected from ξ_1 under particular encryption conditions. High TPR is vital, especially in situations where failing to correctly identify a ciphertext from ξ_1 could pose significant security threats.

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).

VI. Results

In our experiment, we evaluated the cryptographic indistinguishability of two lightweight block ciphers, SPECK32/64 and SIMON32/64 using four DL architecture: ResNet, CNN, LSTM, and BiLSTM. We conducted assessments under both round-reduced and standard (full-round) configurations. The classification performance for each configuration is summarized in Table III.

In the round-reduced configuration, both ciphers demonstrate strong indistinguishability. For SPECK32/64, models such as ResNet, LSTM, and BiLSTM achieved near-random performance (accuracy ≈50%, ROC-AUC ≈50%), with zero precision and recall, indicating an inability to differentiate ciphertexts of $\mathcal{P}_1$ or $\mathcal{P}_2$. The CNN model showed marginal improvement (recall = 0.0356) but remained ineffective, as evidenced by its low F1-score (0.0665). Similarly, for SIMON32/64, ResNet and CNN models exhibited balanced but random-like accuracy (≈50%), with CNN marginally better at detecting ciphertexts of $\mathcal{P}_2$ (recall = 0.2235) but compromised

TABLE I
MODEL-SPECIFIC HYPERPARAMETER SEARCH SPACE

Hyperparameter	LSTM-based Models (LSTM, 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 II
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)

by high false positives (FPR = 0.3086). LSTM and BiLSTM models entirely failed, reinforcing security of the lightweight block ciphers.

In the standard configuration, the results highlight systemic biases rather than meaningful discrimination. For SPECK32/64, ResNet achieved perfect recall (1.0) but trivial precision (0.5), reflecting prediction of all samples as ciphertext belonging to ξ_1 (or $\mathcal{P}_1$), which renders it uninformative. The CNN model exhibited high recall (0.9489) but suffered from severe false positives (FPR = 0.9494), undermining its reliability. For SIMON32/64, ResNet mirrored this behavior, while BiLSTM showed extreme bias (TPR = 0.9996, TNR = 0.0004). Across both ciphers, models like LSTM and BiLSTM consistently failed to generalize, with near-zero recall and precision.

The overarching pattern across configurations is the inability of DL models to surpass random guessing (accuracy and ROC-AUC $\approx$50%) underscores the cryptographic strength of SPECK32/64 and SIMON32/64 against the considered DL models. While certain models (e.g., CNN for SPECK32/64 in standard configuration) showed skewed metrics, these reflect algorithmic biases rather than true discriminative capability. The findings confirm that these ciphers maintain strong security evaluated settings.

VII. 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 learns 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 in 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 ($\approx$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 investigation memorization versus genuine generalization in DL models, we performed a detailed analysis on datasets generated with 16-bit IV randomness. Our training dataset comprises of 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,

TABLE III
INDISTINGUISHABILITY ASSESSMENT FOR SPECK32/64 AND SIMON32/64 IN ROUND-REDUCED STANDARD CONFIGURATION USING MIND-CRYPT

Round Reduced										
Cipher	**DL Model**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC-AUC**	**TPR**	**TNR**	**FPR**	**FNR**
SPECK32/64	ResNet	0.5000	0.0000	0.0000	0.0000	0.5008	0.0000	1.0000	0.0000	1.0000
	CNN	0.5003	0.5043	0.0356	0.0665	0.5005	0.0355	0.9650	0.0350	0.9644
	LSTM	0.5000	0.0000	0.0000	0.0000	0.5014	0.0000	1.0000	0.0000	1.0000
	BiLSTM	0.5000	0.0000	0.0000	0.0000	0.5000	0.0000	1.0000	0.0000	1.0000
SIMON32/64	ResNet	0.5002	0.5002	0.4947	0.4974	0.5003	0.4947	0.5057	0.4943	0.5053
	CNN	0.4993	0.4985	0.2235	0.3086	0.4992	0.2235	0.7750	0.3086	0.4992
	LSTM	0.5000	0.5053	0.0009	0.0017	0.4996	0.0008	0.9991	0.0017	0.4996
	BiLSTM	0.5000	0.0000	0.0000	0.0000	0.4991	0.0000	1.0000	0.0000	0.4991
Standard Configuration										
SPECK32/64	ResNet	0.5000	0.5000	1.0000	0.6667	0.5001	1.0000	0.0000	1.0000	0.0000
	CNN	0.4997	0.4999	0.9489	0.6548	0.4996	0.9489	0.0505	0.9494	0.0511
	LSTM	0.5000	0.0000	0.0000	0.0000	0.5001	0.0000	1.0000	0.0000	1.0000
	BiLSTM	0.4999	0.0000	0.0000	0.0000	0.5003	0.0000	0.9999	0.0000	1.0000
SIMON32/64	ResNet	0.5000	0.5000	1.0000	0.6667	0.5000	1.0000	0.0000	1.0000	0.0000
	CNN	0.4999	0.4998	0.0721	0.1260	0.5000	0.0720	0.9278	0.1260	0.5000
	LSTM	0.5000	0.6667	0.0000	0.0000	0.5004	0.4999	0.5000	0.0000	1.0000
	BiLSTM	0.5000	0.5295	0.0005	0.0010	0.5003	0.9996	0.0004	0.0010	0.5003

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 5,000 ciphertext samples per class (10,000 samples in total) from the training dataset. Within this subset, $\mathcal{P}_1$ had 4,819 unique ciphertexts, and $\mathcal{P}_2$ had 4,815 unique ciphertexts, with 366 redundant samples. Upon examining overlaps between training subset and the complete testing dataset, we identified 5,307 overlapping ciphertext samples. Specifically, 2,659 samples of $\mathcal{P}_1$ and 2,684 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.

VIII. RELATED WORK

Linear & 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 [47], [48].

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. [49] 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. [50] 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. [51] explored 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 Systems (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 providing a more comprehensive evaluation of cryptographic indistinguishability of lightweight block ciphers.

IX. Conclusion

In this research, we introduced a machine learning-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 deep learning architectures to assess these ciphers using machine learning.

Our results show that deep learning 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.

Acknowledgement

This work was supported in part by the National Science Foundation under grants CNS-2154507, OAC-2139358, and CNS-2201465.

References

[1] "State of iot 2024: Number of connected iot devices growing 13% to 18.8 billion globally." https://iot-analytics.com/number-connected-iot-devices/, 2024.

[2] Cisco, "Cisco internet of things (iot) study," 2024.

[3] Statista, "Internet of things (iot) - statistics & facts," 2024.

[4] "The future of iot development: Trends and predictions for 2025." https://imaginovation.net/blog/iot-development-trends-predictions/, 2025.

[5] "Internet of Things — TechCrunch — techcrunch.com." https://techcrunch.com/tag/internet-of-things/, 2022.

[6] M. Appel, A. Bossert, S. Cooper, T. Kußmaul, J. Löffler, C. Pauer, and A. Wiesmaier, "Block ciphers for the iot–simon, speck, katan, led, tea, present, and sea compared," *Proc. Appel Block CF*, pp. 1–37, 2016.

[7] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "Simon and speck: Block ciphers for the internet of things," *Cryptology ePrint Archive*, 2015.

[8] Y. Zhou and D. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing," *Cryptology ePrint Archive*, 2005.

[9] M. Randolph and W. Diehl, "Power side-channel attack analysis: A review of 20 years of study for the layman," *Cryptography*, vol. 4, no. 2, p. 15, 2020.

[10] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Side channel cryptanalysis of product ciphers," in *Computer Security — ESORICS 98* (J.-J. Quisquater, Y. Deswarte, C. Meadows, and D. Gollmann, eds.), (Berlin, Heidelberg), pp. 97–110, Springer Berlin Heidelberg, 1998.

[11] R. C. W. Phan and S.-M. Yen, "Amplifying side-channel attacks with techniques from block cipher cryptanalysis," in *Smart Card Research and Advanced Applications: 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006, Tarragona, Spain, April 19-21, 2006. Proceedings 7*, pp. 135–150, Springer, 2006.

[12] J.-M. Dutertre, J. J. Fournier, A.-P. Mirbaha, D. Naccache, J.-B. Rigaud, B. Robisson, and A. Tria, "Review of fault injection mechanisms and consequences on countermeasures design," in *2011 6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, pp. 1–6, 2011.

[13] J. A. Clark and J. L. Jacob, "Fault injection and a timing channel on an analysis technique," in *Advances in Cryptology—EUROCRYPT 2002: International Conference on the Theory and Applications of Cryptographic Techniques Amsterdam, The Netherlands, April 28–May 2, 2002 Proceedings 21*, pp. 181–196, Springer, 2002.

[14] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proceedings of the IEEE*, vol. 100, no. 11, pp. 3056–3076, 2012.

[15] C. Shao, D. Zhao, H. Li, S. Cheng, S. Gao, and L. Yang, "Detection of security vulnerabilities in cryptographic ics against fault injection attacks based on compressed sensing and basis pursuit," *Journal of Cryptographic Engineering*, pp. 1–14, 2023.

[16] A. Al-Sabaawi, "Cryptanalysis of classic ciphers: Methods implementation survey," in *2021 International Conference on Intelligent Technologies (CONIT)*, pp. 1–6, 2021.

[17] P. N. Lone, D. Singh, V. Stoffová, D. C. Mishra, U. H. Mir, and N. Kumar, "Cryptanalysis and improved image encryption scheme using elliptic curve and affine hill cipher," *Mathematics*, vol. 10, no. 20, p. 3878, 2022.

[18] Y. Xiao, Q. Hao, and D. D. Yao, "Neural cryptanalysis: metrics, methodology, and applications in cps ciphers," in *2019 IEEE conference on dependable and secure computing (DSC)*, pp. 1–8, IEEE, 2019.

[19] A. Gohr, "Brute force cryptanalysis." Cryptology ePrint Archive, Paper 2022/053, 2022. https://eprint.iacr.org/2022/053.

[20] S. Sikdar and M. Kule, "Recent trends in cryptanalysis techniques: A review," in *International Conference on Frontiers in Computing and Systems*, pp. 209–222, Springer, 2022.

[21] R. Verma, N. Dhanda, and V. Nagar, "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*, pp. 513–522, Springer, 2022.

[22] C. J. Mok and C. W. Chuah, "An intelligence brute force attack on rsa cryptosystem," *Communications in Computational and Applied Mathematics*, vol. 1, no. 1, 2019.

[23] A. Gohr, "Improving attacks on round-reduced speck32/64 using deep learning," in *Advances in Cryptology – CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II*, (Berlin, Heidelberg), p. 150–179, Springer-Verlag, 2019.

[24] A. Benamira, D. Gerault, T. Peyrin, and Q. Q. Tan, "A deeper look at machine learning-based cryptanalysis." Cryptology ePrint Archive, Paper 2021/287, 2021. https://eprint.iacr.org/2021/287.

[25] "Mind crypt." https://sites.google.com/view/mind-crypt, 2024.

[26] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The simon and speck lightweight block ciphers," in *Proceedings of the 52nd annual design automation conference*, pp. 1–6, 2015.

[27] Y. LeCun and Y. Bengio, *Convolutional networks for images, speech, and time series*, p. 255–258. Cambridge, MA, USA: MIT Press, 1998.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," in *International conference on artificial neural networks*, pp. 799–804, Springer, 2005.

[30] Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng, Y. Xin, Y. Zhao, and L. Cui, "Robust detection for network intrusion of industrial iot based on multi-cnn fusion," *Measurement*, vol. 154, p. 107450, 2020.

[31] S. Nethala, P. Chopra, K. Kamaluddin, S. Alam, S. Alharbi, and M. Alsaffar, "A deep learning-based ensemble framework for robust android malware detection," *IEEE Access*, vol. 13, pp. 46673–46696, 2025.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, Nov. 1997.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[34] G. Sai Chaitanya Kumar, R. Kiran Kumar, K. Parish Venkata Kumar, N. Raghavendra Sai, and M. Brahmaiah, "Deep residual convolutional neural network: An efficient technique for intrusion detection system," *Expert Syst. Appl.*, vol. 238, feb 2024.

[35] A. Abbas, V. Pano, G. Mainland, and K. R. Dandekar, "Radio modulation classification using deep residual neural networks," *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*, pp. 311–317, 2022.

[36] Y. Shu, R. Qin, Y. He, Y. Li, R. Jiang, and Z. Wu, "Deep residual neural networks with attention mechanism for spatial image steganalysis," *2022 IEEE 24th Int Conf on High Performance Computing & Communications*, pp. 1721–1727, 2022.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2015.

[38] "Simon - speck ciphers." https://github.com/inmcm/Simon_Speck_Ciphers, 2019.

[39] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for Large-Scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, (Savannah, GA), pp. 265–283, USENIX Association, Nov. 2016.

[40] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[41] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'11, (Red Hook, NY, USA), p. 2546–2554, Curran Associates Inc., 2011.

[42] M. R. Albrecht and G. Leander, "An all-in-one approach to differential cryptanalysis for small block ciphers," *IACR Cryptol. ePrint Arch.*, vol. 2012, p. 401, 2012.

[43] I. Dinur, "Improved differential cryptanalysis of round-reduced speck," *IACR Cryptol. ePrint Arch.*, vol. 2014, p. 320, 2014.

[44] C. Blondeau and B. Gérard, "Multiple differential cryptanalysis: Theory and practice," in *Fast Software Encryption Workshop*, 2011.

[45] T. Ashur and D. Bodden, "Linear cryptanalysis of reduced-round speck," 2016.

[46] A. Biryukov, V. Velichkov, and Y. L. Corre, "Automatic search for the best trails in arx: Application to block cipher speck," in *Fast Software Encryption Workshop*, 2016.

[47] F. Abed, E. List, S. Lucks, and J. Wenzel, "Differential cryptanalysis of round-reduced simon and speck," in *Fast Software Encryption Workshop*, 2014.

[48] M. R. Albrecht and G. Leander, "An all-in-one approach to differential cryptanalysis for small block ciphers," *IACR Cryptol. ePrint Arch.*, vol. 2012, p. 401, 2012.

[49] M. S. Khoirom, D. S. Laiphrakpam, and T. Themrichon, "Cryptanalysis of multimedia encryption using elliptic curve cryptography," *Optik*, vol. 168, pp. 370–375, 2018.

[50] Z. Mehmood, A. Sultan, F. Khan, and S. Tahir, "Machine learning based encrypted content type identification," in *2023 International Conference on Communication Technologies (ComTech)*, pp. 117–122, 2023.

[51] X. Hu and Y. Zhao, "Block ciphers classification based on random forest," in *Journal of Physics: Conference Series*, vol. 1168, p. 032015, IOP Publishing, 2019.