Differential Cryptanalysis using Neural Network

By ASHU KHODWAL

WORD COUNT	6302	TIME SUBMITTED	26-JUL-2022 05:28PM
		PAPER ID	88839023

Differential Cryptanalysis using Neural Network

Dissertation submitted in partial fulfilment for the award of the degree

Master of Technology in Computer Science

by

ASHU KHODWAL

Roll No.: CS2018
M.Tech, 2nd year

Under the supervision of
Dr. Malay Bhattacharyya

Computer and Communication Sciences Division
INDIAN STATISTICAL INSTITUTE

July, 2022

CERTIFICATE

This is to certify that the work presented in this dissertation titled "Differential Cryptanalysis using Neural Network", submitted by Ashu Khodwal, having the roll number CS2018, has been carried out under my supervision in partial fulfilment for the award of the degree of Master of Technology in Computer Science during the session 2021-22 in the Computer and Communication Sciences Division, Indian Statistical Institute. The contents of this dissertation, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

21/07/2022

Dr. Malay Bhattacharyya
Assistant Professor, Machine Intelligence Unit
Associate Member, Centre for Artificial Intelligence and Machine Learning
Associate Member, Technology Innovation Hub on Data Science, Big Data Analytics, and Data Curation
Indian Statistical Institute, Kolkata

Acknowledgements

First and foremost, I take this opportunity to express my sincere thankfulness and deep regard to *Dr. Malay Bhattacharyya*, for the impeccable guidance, nurturing and constant encouragement that he had provided me during my post-graduate studies. Words seem insufficient to utter my gratitude to him for his supervision in my dissertation work. Working under him was an extremely knowledgeable experience for a young researcher like me.

I also thank the CSSC and ISI Library for extending their supports in many different ways in my urgent need.

I shall forever remain indebted to my parents, teachers and friends for supporting me at every stage of my life. It is their constant encouragement and support that has helped me throughout my academic career and especially during the research work carried out in the last one year.

Ashu Khodwal.

Date: 21-07-2022

Ashu Khodwal
Roll No.: CS2018
M.Tech, 2nd year
Indian Statistical Institute

Abstract

Recently, deep learning has enabled significant advancements on a variety of challenging tasks, from machine translation and autonomous driving to mastering a variety of abstract board games. Practical machine learning approaches in cryptography have primarily centred on side-channel analysis. In this dissertation, we attempt to train neural networks with the unique characteristics of a round-reduced Speck. The purpose of training neural networks is to differentiate between random data and the output of Speck with a specific input difference. We initially determine the expected effectiveness of a few multiple-differential distinguishers for round-reduced Speck32/64 that utilise the full Markov model of Speck32/64 in order to assess the potency of these machine-learned distinguishers, i.e. all differential characteristics following a given input difference. Up to around eight rounds past our selected input difference, a decently high detection efficiency is attained. We also performed cryptanalysis on a different cipher, namely KATAN, with the same approach but achieved no success, thereby indicating the strength of KATAN.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Cryptography

A method of encoding messages or information to make sure that only the intended recipients can read and understand it. In computer science, the term "cryptography" refers to safe information and communication methods that use mathematical principles and a system of calculations based on rules, or algorithms, to change messages in ways that are challenging to read. These deterministic algorithms are employed in the creation of cryptographic keys, digital signature, online browsing on the internet, and private communications like email and credit card transactions.

1.2 Terminologies

The major components of a cryotography-assisted communication (see Fig. 1) are listed below.

- **Plaintext:** Anything that people can understand and/or relate to is considered plaintext. It is in plaintext if you can understand what is written.
- **Ciphertext:** It is also known as encrypted text, is made up of a random assortment of letters and numbers that are incomprehensible to humans. A plaintext message is input into an encryption algorithm, which then processes the plaintext to create a ciphertext. Through the process of decryption, the ciphertext can be reversed to yield the original plaintext.
- **Encryption:** A process/method by which readable massage/information is converted into secret code that hides the actual information's meaning.
- **Decryption:** A process/method of converting an encrypted message back to its original (readable) format.
- **Key:** A string of characters called a cryptographic key is used in an encryption method to change data so that it appears random. It locks (encrypts) data, just like a real key, so that only someone with the proper key may unlock (decrypt) it.

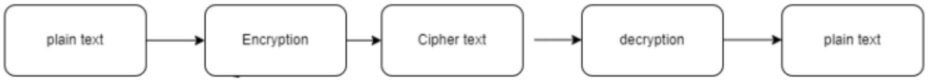


Figure 1: The cryptographic approach of secure communication.

1.3 The need for data encryption

- Privacy- Communication and data at rest cannot be read by anybody outside the intended recipient or the legitimate data owner thanks to encryption. This makes it difficult for hackers, ad networks, Internet service providers, and occasionally governments to intercept and read private information.
- Security- Whether the data is in transit or at rest, encryption aids in the prevention of data breaches. The data on a lost or stolen corporate device won't be compromised because the hard disc is adequately encrypted. Similar to this, encrypted communications allow communicating parties to share private information without disclosing it.
- Data integrity- On path attacks and other harmful activities are also deterred by encryption. Encryption, coupled with other integrity safeguards, ensures that data sent over the Internet hasn't been altered on route to the recipient.
- Authentication- Among other things, public key encryption can be used to prove that the private key listed in a website's TLS certificate belongs to the website's owner. This enables website visitors to be certain that they are linked to the legitimate website.
- Regulations- All of these factors make it mandatory for businesses handling user data to maintain encryption in accordance with numerous industry and governmental laws. HIPAA, PCI-DSS, and the GDPR are a few examples of regulatory and compliance standards that demand encryption.

1.4 Types of cryptography

Cryptography have mainly two different types:

- Secret Key Cryptography
  There is only one key, and it is used by all communicating parties for both encryption and decryption. Encryption and decryption with the same key(see Fig. 2).

- Public Key Cryptography
  Asymmetric encryption, often known as public key encryption, uses two keys: one is used for encryption and the other for decryption. The encryption key is shared openly for use by anybody, however the decryption key is kept secret (thus the term "private key") hence the "public key" name(see Fig. 3).

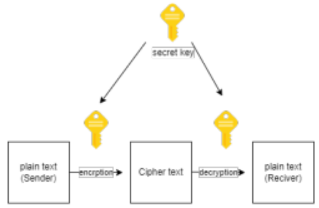


Figure 2: The principle of secret key cryptography.

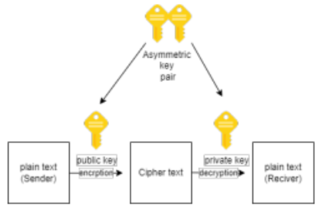


Figure 3: The principle of public key cryptography.

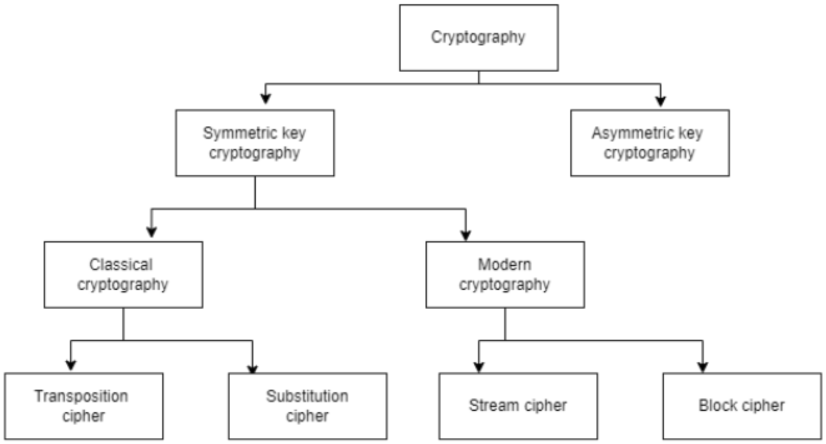


Figure 4: Classification of different cryptographic approaches.

2 Block Cipher

A block cipher uses a block of plaintext bits to create an equivalent sized block of ciphertext bits. In the specified system, the block size is fixed. The encryption scheme's strength is not directly impacted by the block size selection. The length of the key has an impact on the cipher's strength(see Fig. 5).

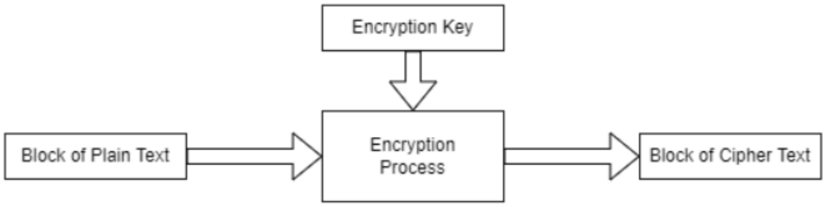


Figure 5: The encryption process in block ciphers.

2.1 Block Cipher Schemes

There are several different block ciphers in use today. Many of them are well-known to the public. Below is a list of the most well-known and prominent block ciphers.

- Digital Encryption Standard (DES) block cipher that was widely used in [9] the 1990s. Currently, it is regarded as a "broken" block cipher, mostly because of its tiny key size.
- Triple DES This alternative approach is based on applying DES repeatedly. Although still regarded, it is ineffective in comparison to the newer, faster block ciphers that are currently accessible.
- [9] Advanced Encryption Standard (AES) It is a relatively new block cipher that is based on the AES design competition winner Rijndael.

2.2 Feistel Block Cipher

A design approach is used to create several distinct block ciphers. One illustration of a Feistel Cipher is DES. The encryption and decryption processes of a cryptographic system based on the Feistel cipher structure employ the same algorithm.
The Feistel structure is used in the encryption process, with each round of processing the plaintext consisting of a substitution step and a permutation step.

- Each round's input block is split into two parts, referred to as the left half and the right half respectively by the letters L and R.
- The right half of the block R, moves through each round unaltered. However, the action that the left half, L, is dependent on R and the encryption key. First, we use the encryption function f, which requires the keys K and R as inputs. The output of the function is f(R,K). The result of the mathematical function is then XORed with L.

- Instead of using the entire encryption key during each round in a practical Feistel Cipher implementation, like DES, a round-dependent key (a subkey) is derived from the encryption key. As a result, even though all of these subkeys are connected to the original key, each round uses a different key.

- The modified L and unmodified R are switched at the permutation phase at the end of each round. As a result, the R from the current round would be the L for the following one. And the output L from the previous round serves as R for the subsequent one.

- A round is formed by the above substitution and permutation procedures. The algorithm design specifies the number of rounds.

- The two subblocks, R and L are concatenated in this order after the final round to create the ciphertext block (see Fig. 6).

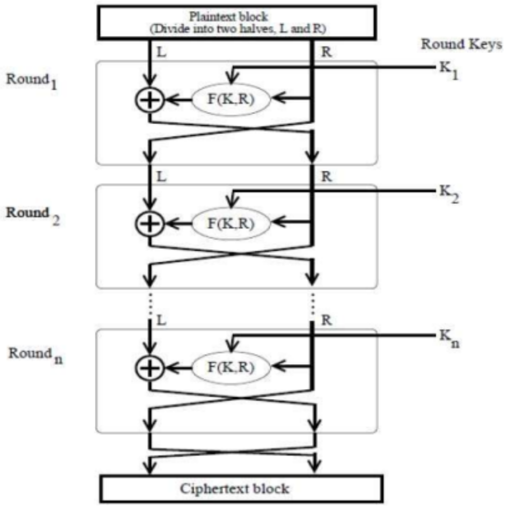


Figure 6: The Feistel structure.

The decryption in Feistel cipher is essentially the same. The Feistel structure starts with a block of ciphertext rather than plaintext, and from there, the process proceeds exactly the same way as it is shown in Fig. 6. Though the procedure is essentially the same but not exactly. The subkeys used in encryption are utilized in reverse order during decryption, which is the only difference. In the final phase of the Feistel Cipher, the letters L and R must be switched. The resulting ciphertext cannot be decrypted using the same technique if they are not switched.

3 Differential Cryptanalysis

Differential cryptanalysis investigates how differences spread across ciphers. Let a function $f : F_2^b \rightarrow F_2^b$ and x_1, x_0 be two different inputs for f with a difference $\Delta x = x_1 \oplus x_0$. Let $y_1 = f(x_1)$ and $y_0 = f(x_0)$ and a difference $\Delta y = y_1 \oplus y_0$. Then, We are curious in the probability of a transition from Δx to Δy ($\Delta x \rightarrow \Delta y$):

$$\mathbb{P}(\Delta x \xrightarrow{f} \Delta y) := \frac{\#\{x | f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^b}$$

4 Machine Learning

Machine learning is a subfield of artificial intelligence (AI) and computer science. It focuses on using data and algorithms to simulate how humans learn, gradually increasing the accuracy of the system. The learning mechanism in a machine learning model can be of two main types - supervised and unsupervised. These are explained below.

- **Supervised Learning:** In the process of developing artificial intelligence (AI), supervised learning involves training a computer system on input data that has been tagged for a certain output.
- **Unsupervised Learning:** The process of using artificial intelligence (AI) algorithms to find patterns in data sets including numeric data points that are neither categorised nor labelled.

Note that there could be semi-supervised learning too.

4.1 Working process of Machine Learning

Machine learning algorithm into three main steps.

- **A Decision Process:** In general, predictions or classifications are made using machine learning algorithms. Your algorithm will generate an estimate about a pattern in the input data based on some input data, which can be labelled or unlabeled.
- **An Error Function:** An error function is used to assess how well the model predicts. If there are known examples, an error function can compare them to gauge the model's correctness.

- **A Model Optimization Process:** Weights are changed to lessen the difference between the known example and the model estimate if the model can better fit the data points in the training set. Until an accuracy level is reached, the system will repeat this assessment and optimisation procedure, automatically by updating weights(see Fig. 7).

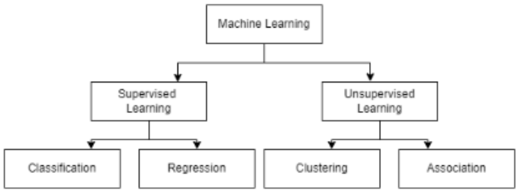


Figure 7: An overview of machine learning.

5 Deep Neural Networks (DNN)

The main problem tackled by DNN is, given a dataset $D = (x_0, y_0), \ldots, (x_n, y_n)$, with $x_i \in X$ being samples and $y_i \in [0, \ldots, l]$ being labels, to find the optimal parameters θ^* for the DNN_θ model, with the parameters θ such that:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=0}^{n} L(y_i, DNN_\theta(x_i))$$

and L standing for the loss function. Since it cannot be expressed literally, the approximation will rely on the optimization algorithm selected, such as stochastic gradient descent. As they have a significant impact on the final quality of the solution, hyper-parameters of the problem parameters whose value is utilized to regulate the learning process also need to be altered.

5.1 Residual Network (ResNet)

The Residual Blocks idea was created by this design to address the issue of the vanishing/exploding gradient. In this network, we use a technique called skip connections. Skip connection bypasses some levels in between to link layer activations to subsequent layers. This creates a Res block. These Res blocks are stacked to create ResNets. The strategy behind this network is to let the network fit the residual mapping rather than have layers learn the underlying mapping. Let the network fit instead of, say, the original mapping $H(x)$ (see Fig. 8).

$$F(x) := H(x) - x$$

which gives

$$H(x) := F(x) + x.$$

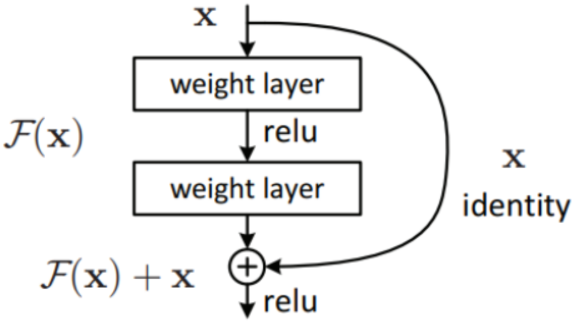


Figure 8: Residual Block.

The benefit of including this kind of skip link is that regularization will skip any layer that degrades architecture performance. As a result, training an extremely deep neural network is possible without encountering issues with vanishing or expanding gradients.

6 Problem Statement

Unlike ML, which explores patterns from the given data, cryptography aims to hide the underlying patterns in the data [7]. Principally, one can deduce cryptanalysis as ML tasks, be it supervised or unsupervised. This provides a way of testing the strength of a cryptoraphic approach because the stronger the cryptosystem, the easier the deduced learning problem. Therefore, a provably good cryptosystem is possibly convertible to a hard learning problem [5] and vice versa [6, 3]. Thus, the ML approaches can play a major role in testing the hardness of cryptanalysis for a given cryptographic approach.

In this thesis, we aim to address the problem of differential cryptanalysis. Differential cryptanalysis is a kind of black box cryptanalysis that aims to understand how differences in plaintexts relate to the differences in ciphertexts. In CRYPTO 2019, Gohr made a novel and successful attempt to apply deep learning for differential cryptanalysis against NSA block cipher Speck32/64, achieving a higher accuracy than the pure differential distinguishers [4]. Gohr trained a deep neural network using the labeled ciphertext pairs as training data. Data with label 0 is random content, while data with label 1 comes from an encrypted plaintext pair with a defined input difference. Then the trained model was used to distinguish between the real ciphertext pairs and random pairs. On applying the trained model to Speck32/64[2], a higher accuracy than the pure differential distinguishers is obtained.

7 Gohr's Architecture

7.1 Introduction of SPECK

Basic notations: Here, $\oplus$, $\wedge$, and $\boxplus$ stand for "exclusive-OR", "bit-wise logical AND", and "modular addition" respectively. $\ll$ and $\gg$ respectively stand for a left-to-right bit rotation, whereas $a||b$ represents the joining of two bit strings, a and b.
In SPECK The plaintext $(l_0||r_0) \leftarrow P$ is used to initialize the 32-bit internal state, which is divided into a 16-bit left and 16-bit right part. These parts are denoted by l_i and r_i for round i, respectively. The cipher's round function is then a relatively straightforward Feistel structure with bitwise XOR and 16-bit modular addition.

In Fig. 9, where k_i represents the 16-bit subkey for round i and where $\alpha = 7$, $\beta = 2$. The final ciphertext C is then obtained as $C \leftarrow (l_{22}||r_{22})$. A key scheduling method that is quite similar to the round function is used to generate the subkeys as follows.

$$l_{i+1} = ((l_i \gg \alpha) mod(r_i)) \oplus k_i$$

$$r_{i+1} = ((r_i \ll \beta)) \oplus l_{i+1}$$

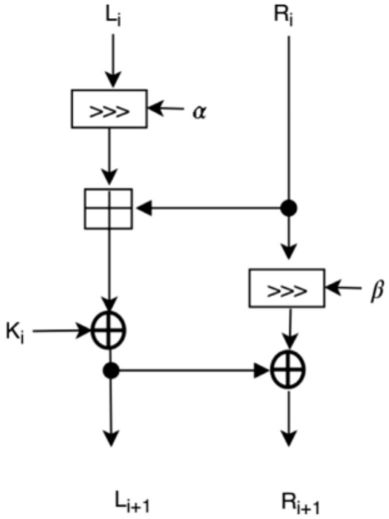


Figure 9: The round function of SPECK.

7.2 Definitions

- The incoming data is transformed linearly by a linear neural network, which works as follows: $out = inp.A^T + c$. Theta is equal to (A, c) in this case. The perceptron layer or dense layer are other common names for the linear neural network.

- Convolution is applied to a fixed (multi-)temporal input signal by a one-dimensional convolutional neural network (1D-CNN). Multiple linear neural networks (one for each filter) applied to different portions of the input can be seen as the 1D-CNN operation. This component is sliding, has a kernel size, stride-size pitch, and padding-dependent start and finish locations.

- Batch Normalisation: To expedite the training process, training samples are often randomly gathered in batches. Thus, normalising the overall tensor in accordance with the batch dimension is common place, can be used after the convolution layer to lessen internal covariate shift, which efficiently solves the gradient disappearance issue and expedites network training.

- Activation functions: The two activation functions that we use here are the Rectified Linear Unit (ReLU), $ReLU(x) = max(0, x)$, the Sigmoid, s $Sigmoid(x) = (x) = 1/(1 + exp(-x))$

- Residual Network (ResNet): For greater accuracy, ResNet can train a deeper CNN model. Establishing "shortcuts (skip) connections" between the front layer and the back layer is the main goal. It is made up of a number of Residual blocks. A residual block can be expressed as: $x_n + 1 = x_n + F(x_n)$. It has two parts: the direct mapping part and the residual part. $F(x_n)$ is the residual part(see Fig. 10).

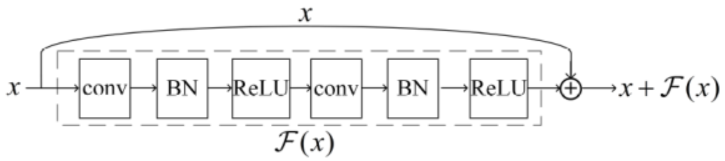
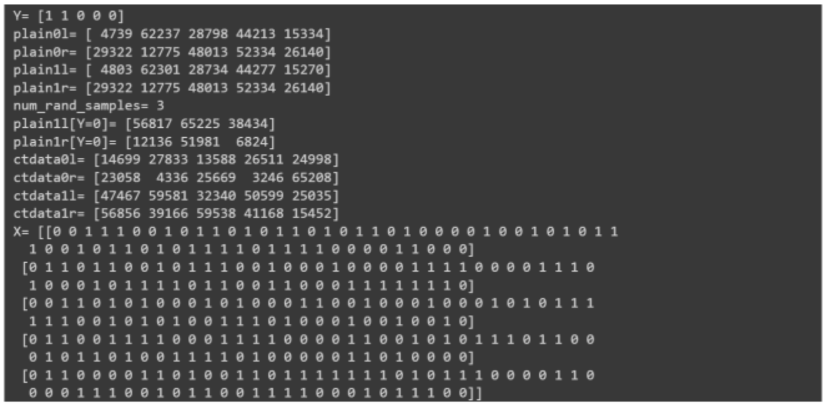


Figure 10: The residual block used in Gohr's work.

7.3 Data Generation

A vector of binary valued real/random labels Y_i and uniformly distributed keys K_i and plaintext pairings P_i with the input difference $\Delta = 0x0040/0000$ were obtained by using the random number generator (/dev/urandom) to produce training and validation data. If Y_i was set, the plaintext pair P_i was then encrypted for k rounds to create training or validation data for k-round Speck otherwise, the second plaintext of the pair was changed to a newly generated random plaintext.

Data sets made up of 10^7 samples were created in this fashion for training. Preprocessing was done to change the data's format to match the network's requirements(see Fig. 11).



```
Y= [1 1 0 0 0]
plain0l= [ 4739 62237 28798 44213 15334]
plain0r= [29322 12775 48013 52334 26140]
plain1l= [ 4803 62301 28734 44277 15270]
plain1r= [29322 12775 48013 52334 26140]
num_rand_samples= 3
plain1l[Y=0]= [56817 65225 38434]
plain1r[Y=0]= [12136 51981  6824]
ctdata0l= [14699 27833 13588 26511 24998]
ctdata0r= [23058  4336 25669  3246 65208]
ctdata1l= [47467 59581 32340 50599 25035]
ctdata1r= [56856 39166 59538 41168 15452]
X= [[0 0 1 1 1 0 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 1
  1 0 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 1 1 0 0 0 0 1 1 0 0 0]
 [0 1 1 0 1 1 0 0 1 0 1 1 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0
  1 0 0 0 1 0 1 1 1 1 0 1 1 0 0 1 1 0 0 0 1 1 1 1 1 1 1 0]
 [0 0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 1 0 1 1 1
  1 1 1 0 0 1 0 1 0 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 0 1 0]
 [0 1 1 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 0 0
  0 1 0 1 1 0 1 0 0 1 1 1 1 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0]
 [0 1 1 0 0 0 0 1 1 0 1 0 0 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 0 0 0 0 1 1 0
  0 0 0 1 1 1 0 0 1 0 1 1 0 0 1 1 1 1 0 0 0 1 0 1 1 1 0 0]]
```

Figure 11: Data points for SPECK.)

7.4 Architecture

Gohr's neural distinguisher has mainly three components as listed below.

1. **Input Block (Block 1):** It has a batch normalisation, a ReLU activation function, and a 1D-CNN with a single kernel size (see Fig. 13).

2. **Res Block (Blocks 2-i):** It has one to ten layers, with two 1D-CNNs with 3-kernel sizes making up each layer. Each layer is then followed by batch normalisation and a ReLU activation algorithm (see Fig. 14).

3. **Prediction block (Block 3):** Its nonlinear final classification block is made up of three perceptron layers and is divided into two sections by batch normalisation and ReLU functions. A sigmoid function completes the structure (see Fig. 15).

The input to the first convolution block (Block 1) is a 4×16 matrix, where each row having 16-bit value in this order $(C_l, C_r, C_l^{'}, C_r^{'})$, a convolution layer with 32 filters is then applied. The kernel size of this 1D-CNN is 1 thus, it maps this $(C_l, C_r, C_l^{'}, C_r^{'})$ to

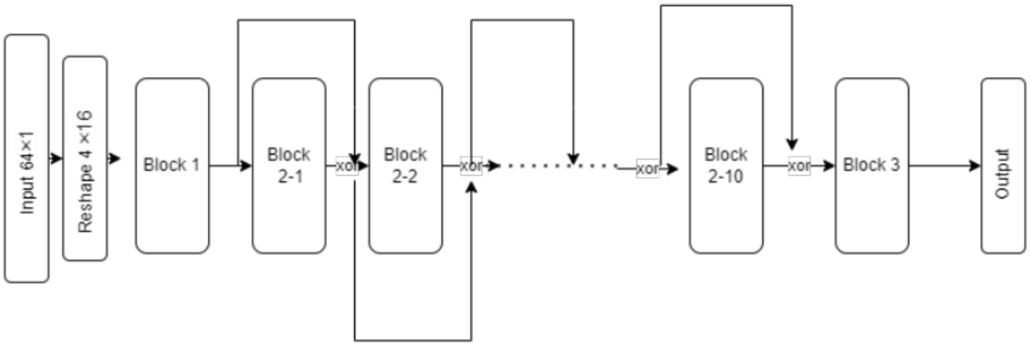


Figure 12: The whole pipeline of Gohr's deep neural network.

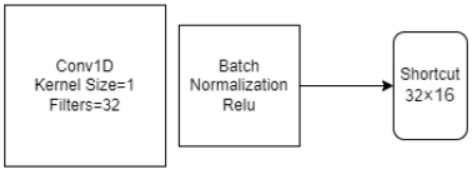


Figure 13: Initial convolution Block(Block-1) of Gohr's network.

$(filter1, filter2, ..., filter32)$ And Each filter is a non-linear combination of the features $(C_l, C_r, C_l^{'}, C_r^{'})$ after the ReLU activation function depending on the value of the inputs and weights learned by the 1D-CNN. In the residual block, the output of the first block is connected with the input and combined with the output of the next layer. Both 1D-CNNs in the residual blocks (Blocks 2-i) contain a kernel of size 3 and a temporal dimension that is invariant across layers thanks to padding and strides of size 1 and 3. To avoid the pertinent input signal from being wiped away across layers, the output of each layer is connected to the input and added to the output of the following layer. A (32×16) feature tensor is the result of a residual block. .

The attened output tensor of the residual block is the input for the final classification block. With batch normalisation and ReLU activation functions for the first two layers and a final sigmoid activation function performing the binary classification, this 512×1 vector is sent into three perceptron layers (also known as MLPs).

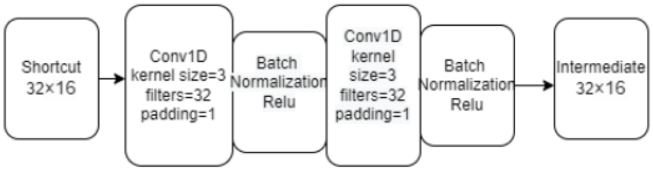


Figure 14: The residual block(Block(2-i)) of Gohr's network.

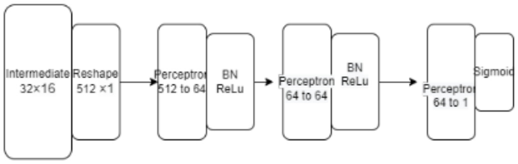


Figure 15: Final block of Gohr's network.

7.5 Results

Basic Training Pipeline-200 epoch were performed on the 10^7 dataset. The datasets were processed in 5000 batches. The final 10^6 samples were held back for validation. The L2 weights regularization was used to optimise against mean square error loss with a minimal penalty. with regularization parameter $c = 10^5$, using the Adam algorithm with default parameters in Keras . A cyclic learning rate schedule was used, setting the learning rate l_i for epoch i to $l_i := \alpha + (ni)mod(n+1)(\alpha - \beta)/n$,
with $\alpha = 10^{-4}$, $\beta = 2 \times 10^{-3}$ and n = 9. The networks obtained at the end of each epoch were stored and the best network by validation loss was evaluated against a test set of size 10^6 not used in training or validation. These are accuracies of neural distinguishers for the rounds 5, 6, 7 and 8.

round	test accuracy
5	0.929
6	0.788
7	0.616
8	0.514

Table 1: Accuracy for different rounds (using Gohr's architecture).

8 Proposed Architecture

8.1 Architecture

In this Architecture we propose 'Stocastic Approach' for skip connections.
It has mainly three components as listed below.

Input Block (Block-1): It has a batch normalisation, a ReLU activation function, and a 1D-CNN with a single kernel size(see Fig. 17).

Res Block (Blocks 2-i): It has one to ten layers, with two 1D-CNNs with 3-kernel sizes making up each layer. Each layer is then followed by batch normalisation and a ReLU activation algorithm(see Fig. 18).

Prediction block (Block 3): Its nonlinear final classification block is made up of three perceptron layers and is divided into two sections by batch normalisation and ReLU functions. A sigmoid function completes the structure .

In this approach we use probability. Suppose P is the probability of each skip connection. For Gohr's model, we take $P = 1$.

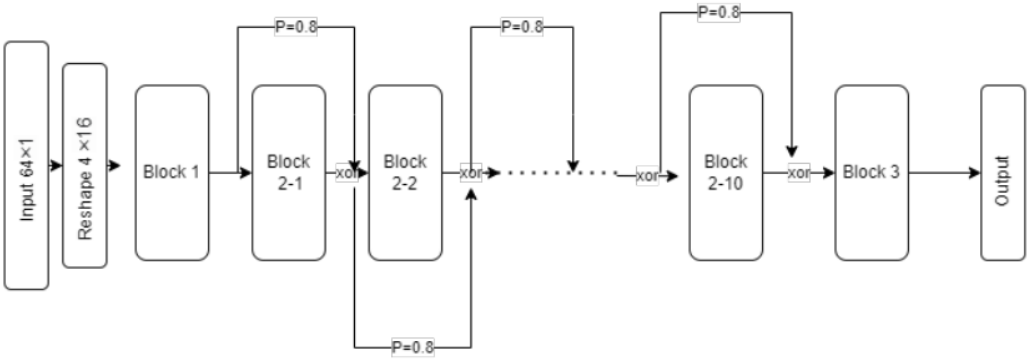


Figure 16: Architecture for P=0.8 probability.

The input to the first convolution block (Block 1) is a 4×16 matrix, where each row having 16-bit value in this order $(C_l, C_r, C_l^{'}, C_r^{'})$, a convolution layer with 32 filters is then applied. The kernel size of this 1D-CNN is 1, thus, it maps this $(C_l, C_r, C_l^{'}, C_r^{'})$ to $(filter1, filter2, ..., filter32)$ And Each filter is a non-linear combination of the features $(C_l, C_r, C_l^{'}, C_r^{'})$ after the ReLU activation function depending on the value of the inputs and weights learned by the 1D-CNN. The output of the first block is connected to the input and added to the output of the subsequent layer in the residual block . Both 1D-CNNs in the residual blocks (Blocks 2-i) contain a kernel of size 3 and a temporal dimension that is invariant across layers thanks to padding and strides of size 1 and 3.Here for each skip connection we assign P probability. To avoid the pertinent input signal from being wiped away across layers, the output of each layer is connected to the input and added to the output of the following layer. A (32×16) feature tensor is the result of a residual block. .

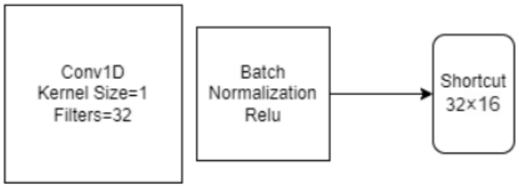


Figure 17: Block-1 for propose Architecture.

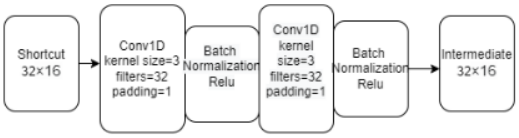


Figure 18: Block 2-i for propose Architecture.

The attened output tensor of the residual block is the input for the final classification block with batch normalisation and ReLU activation functions for the first two layers and a final sigmoid activation function performing the binary classification, this 512×1 vector is sent into three perceptron layers (also known as MLPs).

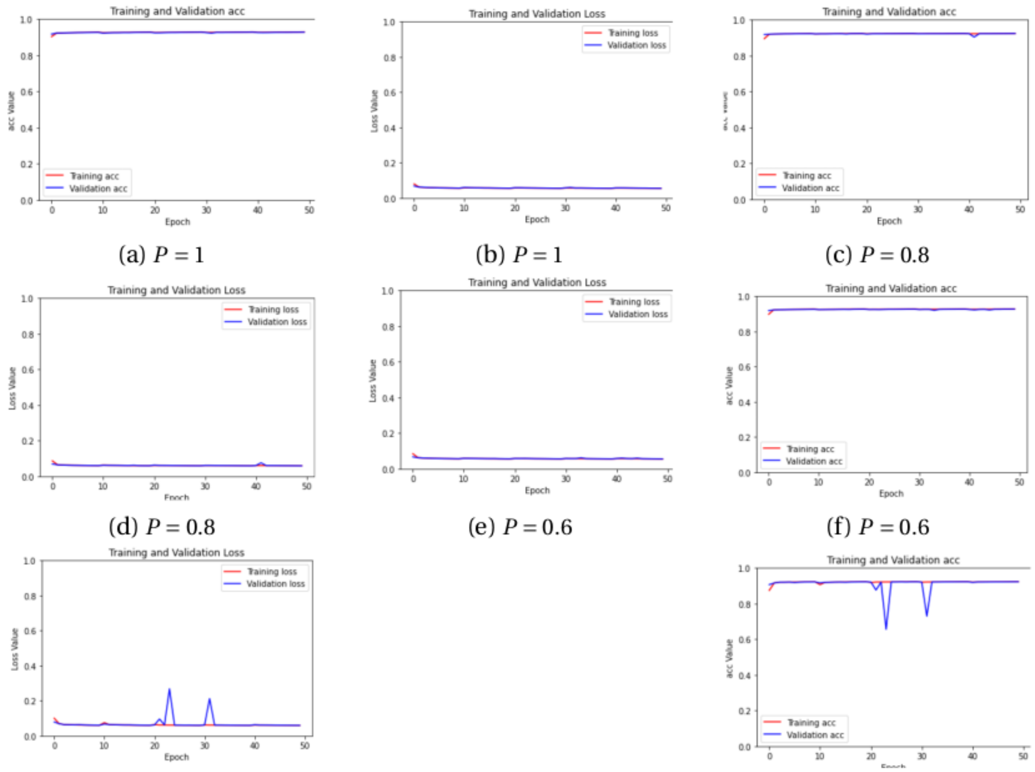
First If we take $P = 0.8$ that mean we drop 20 percent skip connections and keep 80 percent skip connection. By this way we only use 80 percent skip connection and getting same results.We try it with different - different P and getting almost same results. After that We try it for the probability $P = 0.8, 0.6$ and 0.4.

8.2 Results

Basic Training Pipeline-For training we take 10^7 data points .Training was run for 50 iterations. Batches of 5000 records per dataset were processed. The 10^6 samples were held back for validation. Based on L2 weights regularisation (with regularisation parameter), optimization was done against mean square error loss plus a modest penalty $c = 10^{-5}$) using the Adam algorithm with default parameters in Keras The learning rate was established using a cyclic schedule.learning rate l_i for epoch i to

$$l_i := \alpha + (ni) mod (n+1)(\alpha - \beta)/n,$$

with $\alpha = 10^{-4}$, $\beta = 2\times10^{-3}$ and n = 9. The networks that were obtained at the end of each epoch were saved, and the top network by validation loss was evaluated against a test set with a 10^6 size that was not used for training or validation. We get same accuracy as Gohr but we use less probability of skip connection.



(a) $P = 1$ (b) $P = 1$ (c) $P = 0.8$

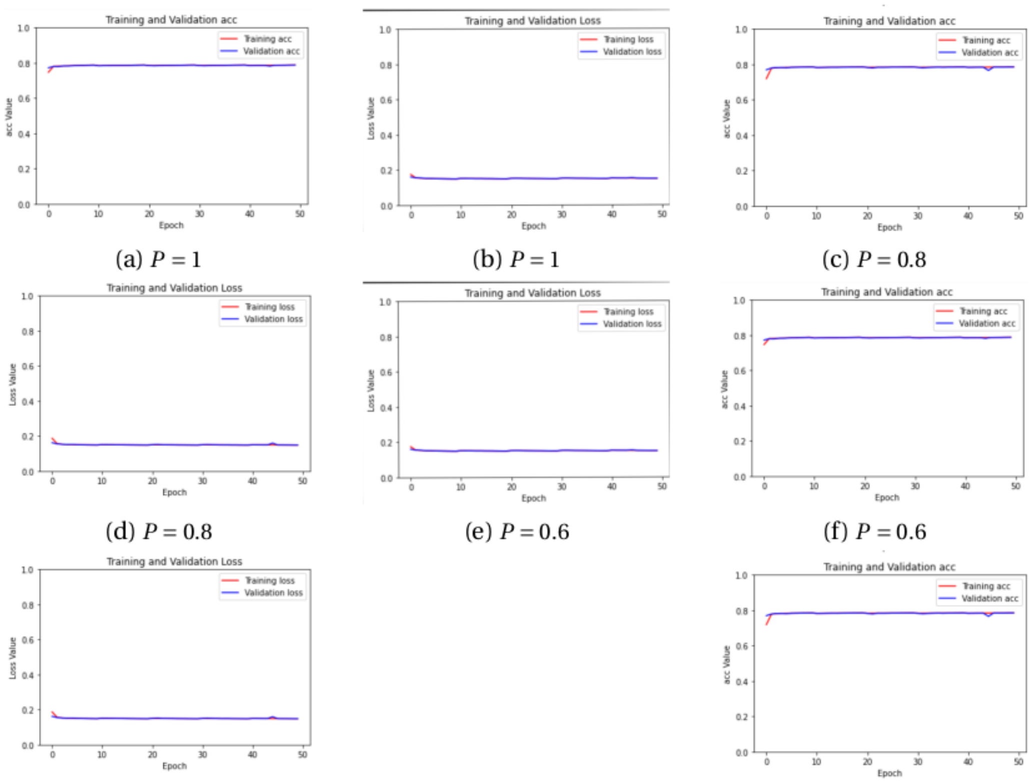
(d) $P = 0.8$ (e) $P = 0.6$ (f) $P = 0.6$

(g) $P = 0.4$ (h) $P = 0.4$

Figure 19: Training/validation Accuracy and loss for round = 5.

Probability of skip	Time(in seconds)	best validation accuracy	test accuracy
1	10594.624483	0.9286370	0.928929
0.8	10475.326285	0.9239220	0.923371
0.6	10715.470632	0.9274809	0.928158
0.4	10399.968406	0.9231290	0.923135

Table 2: Test/validation accuracy and loss for round = 5.



(a) $P = 1$ (b) $P = 1$ (c) $P = 0.8$

(d) $P = 0.8$ (e) $P = 0.6$ (f) $P = 0.6$

(g) $P = 0.4$ (h) $P = 0.4$

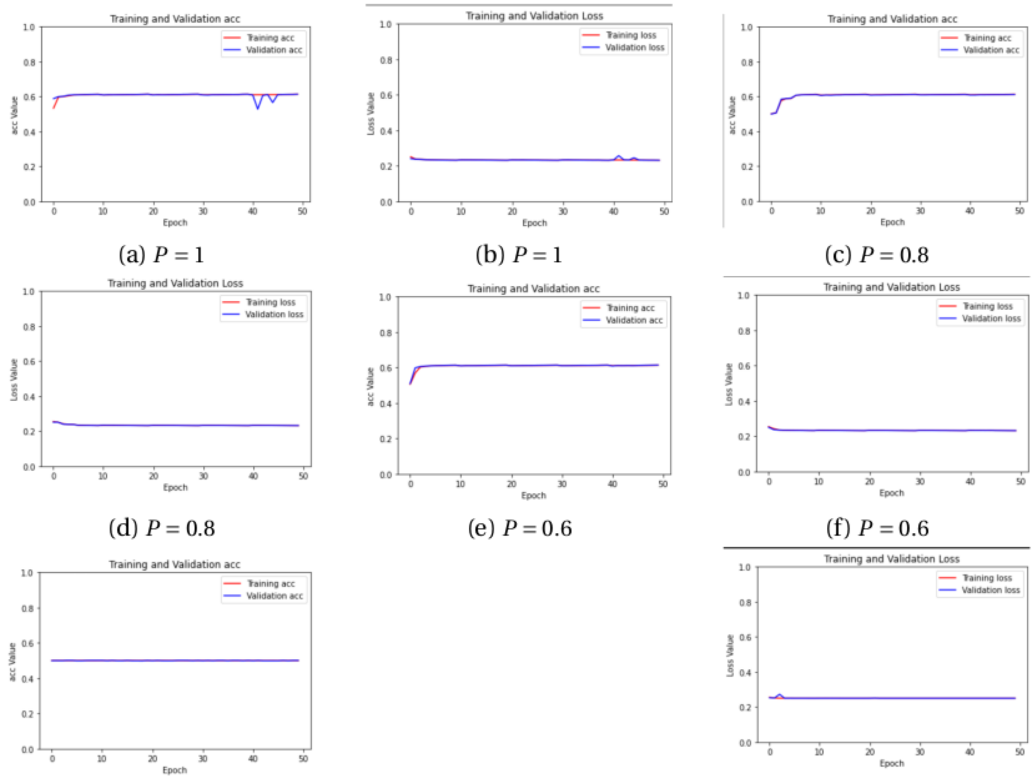
Figure 20: Training/validation accuracy and loss for round = 6.

Probability of skip	Time(in seconds)	best validation accuracy	test accuracy
1	10785.07476634	0.78772199	0.787872
0.8	10415.16607141	0.7869390	0.786619
0.6	10535.71463910	0.78755897	0.787301
0.4	10442.90266807	0.78510	0.786066

Table 3: Test/validation accuracy and loss for round = 6.

Probability of skip	Time(in seconds)	best validation accuracy	test accuracy
1	10510.768701	0.613547	0.611766
0.8	10535.5821011	0.611773	0.611844
0.6	10595.134	0.61383	0.612601
0.4	10237.53489	0.50102299	0.49980

Table 4: Test/validation accuracy and loss for round = 7.



(a) $P = 1$ (b) $P = 1$ (c) $P = 0.8$

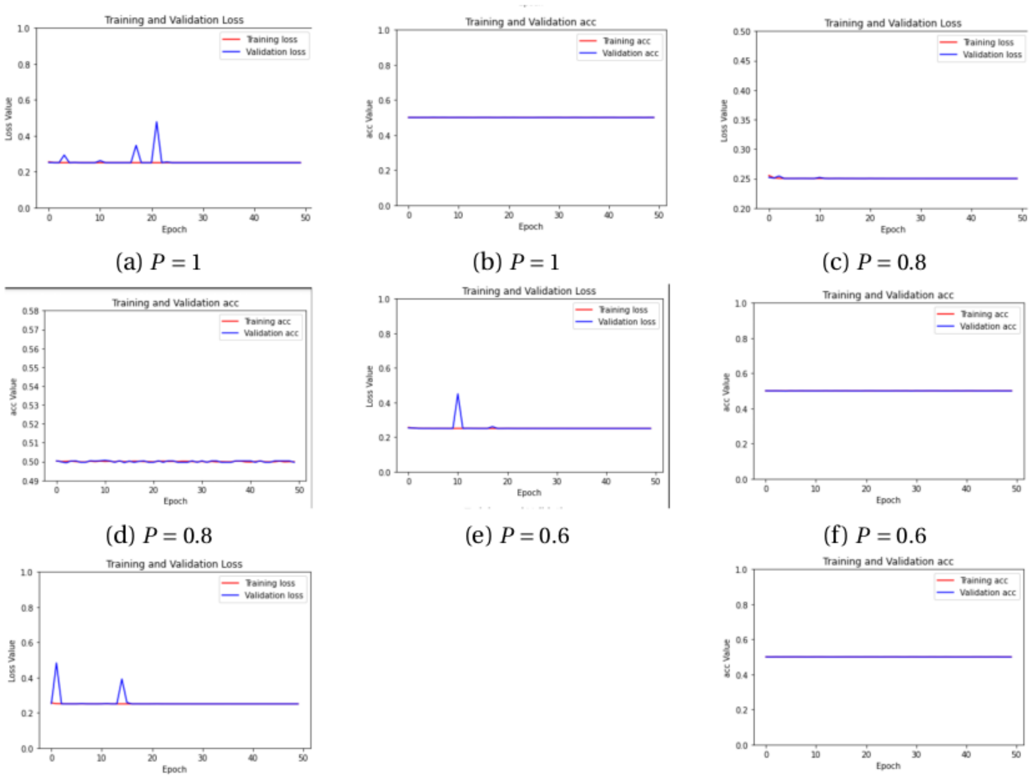
(d) $P = 0.8$ (e) $P = 0.6$ (f) $P = 0.6$

(g) $P = 0.4$ (h) $P = 0.4$

Figure 21: Training/validation accuracy and loss for round = 7.

Probability of skip	Time(in seconds)	best validation accuracy	test accuracy
1	10707.3002479	0.5011019	0.50032
0.8	10390.6936800	0.50060397	0.499447
0.6	10496.271875	0.5005000	0.499865
0.4	10293.482472	0.50051999	0.500113

Table 5: Test/Validation Accuracy for Round=8.



(a) $P = 1$

(b) $P = 1$

(c) $P = 0.8$

(d) $P = 0.8$

(e) $P = 0.6$

(f) $P = 0.6$

(g) $P = 0.4$

(h) $P = 0.4$

Figure 22: Training/Validation Acuuracy and Loss for round=8.

9 Differential Cryptanalysis on KATAN Cipher

9.1 The KATAN Set of Block Ciphers

The three KATAN cipher versions are KATAN32, KATAN48, and KATAN64. The ciphers of the KATAN family all use the same nonlinear functions and have the same key schedule, which accepts an 80-bit key and 254 rounds.
We begin by describing KATAN32, and then we go on to discuss the distinctions between KATAN48 and KATAN64. The smallest member of this family, KATAN32, has a 32-bit plaintext and ciphertext size. The plaintext is loaded into two registers (L_1) and (L_2), which have a length of 13 bits and 19 bits, respectively. The least significant bit of the plaintext is loaded to bit 0 of the larger register (L_2), and the most significant bit is loaded to bit (L_1) of the smaller register (L_2). The least significant bits of L_1 and L_2 are loaded with the newly computed bits after shifting L_1 and L_2 to the left (bit i is moved to position $i+1$). The ciphertext, which is exported as the contents of the registers after 254 rounds (bit 0 of L_2 is the least significant bit of the ciphertext), is then generated.

Each iteration of KATAN32 employs the nonlinear functions $f_a()$ and $f_b()$. The following definitions apply to the nonlinear functions $f_a()$ and $f_b()$:

$$f_a(L_1) = L_1[x_1] \oplus L_1[x_2] \oplus (L_1[x_3] \cdot L_1[x_4]) \oplus (L_1[x_5] \cdot IR) \oplus k_a$$

$$f_b(L_2) = L_2[y_1] \oplus L_2[y_2] \oplus (L_2[y_3] \cdot L_2[y_4]) \oplus (L_2[y_5] \cdot L_2[y_6]) \oplus k_b$$

Where IR stands for irregular update rule, k_a and k_b are the two subkey bits, and $L_1[x_5]$ is XORed in rounds where the irregular update is utilized.

The definition of k_a for round i is k_{2i}, but k_b is k_{2i+1}. Each variant's choice of the bits x_i and y_j is specified independently and is listed in Table.

The registers L_1 and L_2 are shifted after the nonlinear functions have been computed, where the MSB falls off (into the corresponding nonlinear function) and the LSB is loaded with the output of the second nonlinear function. thus, following the round, the LSB of L_1 is the output of f_b and the LSB of L_2 is the output of f_a.

The 80-bit key is loaded into an LFSR by the KATAN32 cipher's key schedule (and that of its two further variants, KATAN48 and KATAN64) the least significant bit of the key is loaded to position 0 of the LFSR. The round's subkeys k_{2i} and k_{2i+1} are used to produce positions 0 and 1 of the LFSR for each round, and the LFSR is timed twice.

The feedback polynomial used has a minimal hamming weight of 5 and is a basic polynomial (There are no degree 80 primitive polynomials with just three monomials.)

$$X^{80} + X^{61} + X^{50} + X^{13} + 1$$

We point out that because these places make up a complete difference set, they are less likely to be the target of guess and determine attacks than a thorough key search.

To put it another way, if K is the key, then i is the round's subkey. $k_a || k_b = k_{2i} || k_{2i+1}$ where

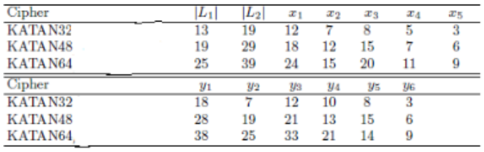
$$k_i = \begin{cases} K_i & \text{for } i = 0 \ldots 79 \\ k_{i-80} \oplus k_{i-61} \oplus k_{i-50} \oplus k_{i-13} & \text{Otherwise} \end{cases}$$

The variations among the different versions of KATAN ciphers include the following:

- The size of the plaintext/ciphertext.
- The size of L_1 and L_2
- The order in which bits enter nonlinear functions
- The frequency of application of the nonlinear functions in each iteration.

While the first distinction is readily apparent, we define the register lengths and the locations of the bits that enter the ciphers nonlinear functions in Table 23. Each variant's choice of the bits x_i and y_j is specified separately and is presented in Table.

For KATAN48, the functions f_a and f_b are applied twice throughout one round of the cipher. After updating the registers, the first pair of f_a and f_b is applied, and then they are applied once again using the same subkeys. Of course, a successful implementation can carry out these two processes simultaneously. Each round of KATAN64 applies f_a and f_b three times (again, with the same key bits).



Cipher	$\lvert L_1 \rvert$	$\lvert L_2 \rvert$	x_1	x_2	x_3	x_4	x_5
KATAN32	13	19	12	7	8	5	3
KATAN48	19	29	18	12	15	7	6
KATAN64	25	39	24	15	20	11	9

Cipher	y_1	y_2	y_3	y_4	y_5	y_6
KATAN32	18	7	12	10	8	3
KATAN48	28	19	21	13	15	6
KATAN64	38	25	33	21	14	9

Figure 23: Parameters defined for the KATAN family of ciphers.

In Figure, we present the structure of KATAN32, which resembles the round function of any form of KATAN (see Fig. 24.

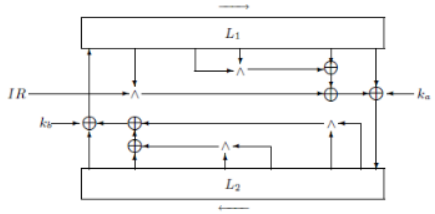
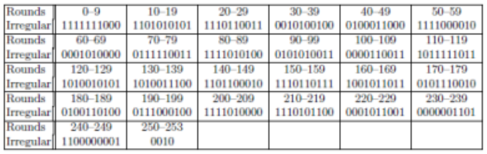


Figure 24: Outline of a round of KATAN ciphers.

Finally, we define the counter that counts the number of rounds in terms of specifications. The feedback polynomial is used to clock the round-counter LFSR once after initialising it to the state of all 1's.

$$X^{80} + X^{61} + X^{50} + X^{13} + 1$$

After 254 more clocks, the encryption process begins and is completed when the LFSR returns to the all-ones state. As previously stated, we employ the LFSR's most important bit to regulate the irregular update (i.e., as the IR signal). The sequence of irregular rounds is provided in Table for the sake of completeness. 1 means that the irregular update rule is used in this round, while 0 means that this is not the case(see Fig. 25).

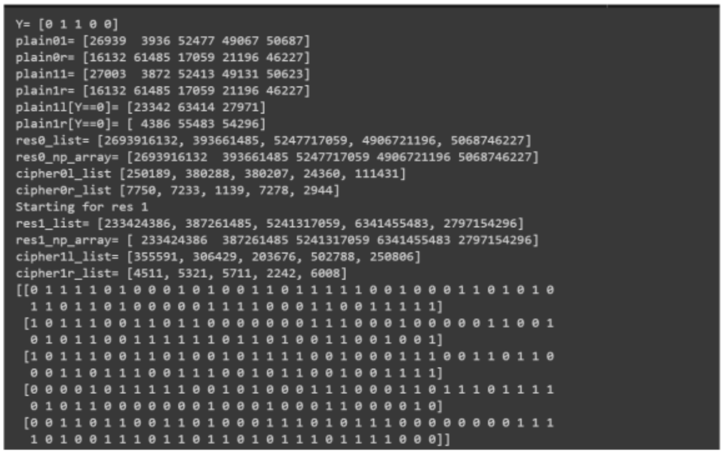


Rounds	0–9	10–19	20–29	30–39	40–49	50–59
Irregular	1111111000	1101010101	1110110011	0010100100	0100011000	1111000010
Rounds	60–69	70–79	80–89	90–99	100–109	110–119
Irregular	0001010000	0111110011	1111010100	0101010011	0000110011	1011111011
Rounds	120–129	130–139	140–149	150–159	160–169	170–179
Irregular	1010010101	1010011100	1101100010	1110110111	1001011011	0101110010
Rounds	180–189	190–199	200–209	210–219	220–229	230–239
Irregular	0100110100	0111000100	1111010000	1110101100	0001011001	0000001101
Rounds	240–249	250–253				
Irregular	1100000001	0010				

Figure 25: The sequence of irragular update in KATAN

9.2 Applying Different Classification Models on KATAN

We take 10^4 training data(see Fig. 26) and 10^3 test data.We try Logistic Regression, support-vector machines (SVM) and random forest classifier (RFC) for different rounds of outputs received from the KATAN cipher and obtained the classification accuracies reported in Table 6.



```
Y= [0 1 1 0 0]
plain0l= [26939  3936 52477 49067 50687]
plain0r= [16132 61485 17059 21196 46227]
plain1l= [27003  3872 52413 49131 50623]
plain1r= [16132 61485 17059 21196 46227]
plain1l[Y==0]= [23342 63414 27971]
plain1r[Y==0]= [ 4386 55483 54296]
res0_list= [2693916132, 393661485, 5247717059, 4906721196, 5068746227]
res0_np_array= [2693916132  393661485 5247717059 4906721196 5068746227]
cipher0l_list [250189, 380288, 380207, 24360, 111431]
cipher0r_list [7750, 7233, 1139, 7278, 2944]
Starting for res 1
res1_list= [233424386, 387261485, 5241317059, 6341455483, 2797154296]
res1_np_array= [ 233424386  387261485 5241317059 6341455483 2797154296]
cipher1l_list= [355591, 306429, 203676, 502788, 250806]
cipher1r_list= [4511, 5321, 5711, 2242, 6008]
[[0 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 1 0 1 1 1 1 1 0 0 1 0 0 0 1 1 0 1 0 1 0
  1 1 0 1 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 1 1 0 0 1 1 1 1 1]
 [1 0 1 1 1 0 0 1 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1
  0 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1 0 0 1 0 0 1]
 [1 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 1 1 1 0 0 1 0 0 0 1 1 1 0 0 1 1 0 1 1 0
  0 0 1 1 0 1 1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 1 1 1 1]
 [0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 1 0 0 0 1 1 1 0 0 0 1 1 0 1 1 1 0 1 1 1 1
  0 1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0]
 [0 0 1 1 0 1 1 0 0 1 1 0 1 0 0 0 1 1 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1
  1 0 1 0 0 1 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0 1 1 1 1 0 0 0]]
```

Figure 26: Data generation approach for KATAN.

Rounds	1	2	3	4	5	6	7	8	9	10
LR	0.52	0.514	0.507	0.506	0.491	0.529	0.509	0.491	0.5	0.494
RFC	0.516	0.528	0.519	0.512	0.498	0.512	0.51	0.505	0.57	0.478
SVM	0.527	0.49	0.511	0.494	0.516	0.505	0.49	0.501	0.507	0.497

Table 6: Test accuracy for different rounds of KATAN cipher.

10 Conclusion and Future Work

We try Gohr's architecture with different skip connection probabilities (to be precise – 0.4, 0.6 and 0.8) to overcome the vanishing gradient problem. With this approach, we obtained almost the same accuracy of differential cryptanalysis. For the fifth, sixth, seventh and eighth rounds, we received test accuracy around 92%, 78%, 61% and 50%, respectively, for different values of the skip probability. We also tried to construct a different architecture for KATAN aiming a good accuracy for round reduced differential attack [1]. We attempted this only for some initial rounds of KATAN without any major success. Thus far our observations through the experiments highlight that KATAN is presumably a better cipher than SPECK. We may improve the given architecture to obtain better accuracy for higher rounds for SPECK and initial rounds of KATAN.

References

[1] Farzaneh Abed, Eik List, Stefan Lucks, and Jakob Wenzel. Differential cryptanalysis of round-reduced simon and speck. In *International Workshop on Fast Software Encryption*, pages 525–545. Springer, 2014.

[2] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The simon and speck families of lightweight block ciphers. *cryptology eprint archive*, 2013.

[3] Avrim Blum. Machine learning theory. *Carnegie Melon Universit, School of Computer Science*, 26, 2007.

[4] Aron Gohr. Improving attacks on round-reduced speck32/64 using deep learning. In *Annual International Cryptology Conference*, pages 150–179. Springer, 2019.

[5] Bharat Lal Jangid. Mutual information and machine learning based distinguishers for pseudo random bit sequences. In *Communication Software and Networks*, pages 435–444. Springer, 2021.

[6] Ronald L Rivest. Cryptography and machine learning. In *International Conference on the Theory and Application of Cryptology*, pages 427–439. Springer, 1991.

[7] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Differential Cryptanalysis using Neural Network

ORIGINALITY REPORT

17%

SIMILARITY INDEX

PRIMARY SOURCES

1	eprint.iacr.org Internet	442 words	7%
2	Christophe Cannière. "KATAN and KTANTAN — A Family of Small and Efficient Hardware-Oriented Block Ciphers", Lecture Notes in Computer Science, 2009 Crossref	110 words	2%
3	"Advances in Cryptology – EUROCRYPT 2021", Springer Science and Business Media LLC, 2021 Crossref	84 words	1%
4	www.iacr.org Internet	70 words	1%
5	Alice James, Avishkar Seth, Subhas Chandra Mukhopadhyay. "Chapter 10 Machine Learning in IoT System", Springer Science and Business Media LLC, 2022 Crossref	66 words	1%
6	www.esat.kuleuven.be Internet	55 words	1%
7	www.cs.technion.ac.il Internet	40 words	1%
8	Heng-Chuan Su, Xuan-Yong Zhu, Duan Ming. "Chapter 1 Polytopic Attack on Round-Reduced	35 words	1%

	Simon32/64 Using Deep Learning", Springer Science and Business Media LLC, 2021 Crossref	
9	Www.tutorialspoint.com Internet	32 words — 1%
10	www.stablx.com Internet	26 words — < 1%
11	Aiman Al-Sabaawi. "Cryptanalysis of Block Cipher: Method Implementation", 2022 International Conference for Advancement in Technology (ICONAT), 2022 Crossref	18 words — < 1%
12	asunix.tufts.edu Internet	16 words — < 1%
13	www.cosic.esat.kuleuven.be Internet	16 words — < 1%
14	cse.iitkgp.ac.in Internet	15 words — < 1%
15	Pallapothala Tejaswini, Priyanshi Singh, Monica Ramchandani, Yogesh Kumar Rathore, Rekh Ram Janghel. "Rice Leaf Disease Classification Using Cnn", IOP Conference Series: Earth and Environmental Science, 2022 Crossref	14 words — < 1%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	< 14 WORDS
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	< 14 WORDS