IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 August 2022, accepted 30 August 2022, date of publication 5 September 2022, date of current version 13 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3204175

RESEARCH ARTICLE

Automated Deep Learning BLACK-BOX Attack for Multimedia P-BOX Security Assessment

ZAKARIA TOLBA[1], MAKHLOUF DERDOUR[2], MOHAMED AMINE FERRAG[3], (Senior Member, IEEE), S. M. MUYEEN[4], (Senior Member, IEEE), AND MOHAMED BENBOUZID[5], (Fellow, IEEE)

[1]Laboratory of Mathematics, Informatics and Systems (LAMIS), Larbi Tebessi University, Tebessa 12022, Algeria
[2]Networks and Systems RSI Laboratory–Annaba, University of Oum El Bouaghi, Oum El Bouaghi 04000, Algeria
[3]Department of Computer Science, Guelma University, Guelma 24000, Algeria
[4]Department of Electrical Engineering, Qatar University, Doha 2713, Qatar
[5]UMR CNRS 6027, University of Brest, 29238 Brest, France

Corresponding authors: Mohamed Amine Ferrag (ferrag.mohamedamine@univ-guelma.dz) and S. M. Muyeen (sm.muyeen@qu.edu.qa)

The publication of this article was funded by Qatar National Library.

ABSTRACT Resistance to differential cryptanalysis is a fundamental security requirement for symmetric block ciphers, and recently, deep learning has attracted the interest of cryptography experts, particularly in the field of block cipher cryptanalysis, where the bulk of these studies are differential distinguisher based black-box attacks. This paper provides a deep learning-based decryptor for investigating the permutation primitives used in multimedia block cipher encryption algorithms.We aim to investigate how deep learning can be used to improve on previous classical works by employing ciphertext pair aspects to maximize information extraction with low-data constraints by using convolution neural network features to discover the correlation among permutable atoms to extract the plaintext from the ciphered text without any P-box expertise. The evaluation of testing methods has been conceptualized as a regression task in which neural networks are supervised using a variety of parameters such as variations between input and output, number of iterations, and P-box generation patterns. On the other hand, the transfer learning skills demonstrated in this study indicate that discovering suitable testing models from the ground is also achievable using our model with optimum prior cryptographic expertise, where we contribute the results of deep learning in the field of deep learning based differential cryptanalysis development.Various experiments were performed on discrete and continuous chaotic and non-chaotic permutation patterns, and the best-performing model had an MSE of $1.8217e^{-04}$ and an R^2 of 1, demonstrating the practicality of the suggested technique.

INDEX TERMS Cryptanalysis, deep learning, convolution, deconvolution, plaintext, ciphertext, block cipher, P-Box, attack.

I. INTRODUCTION

A. MOTIVATION AND GOALS OF THIS PAPER

Block ciphers are famous cryptographic alternatives that improve data confidentiality while also providing the framework for a wide range of other cryptographic algorithms and network protocols [1].

A block cipher uses a key-dependent transformation to handle axed-length (block) data, which frequently includes generic operations like substitution and permutation. The key-dependent modification should be carried out over several rounds until the complete ciphertext is generated (multiple rounds).A key scheduling technique is used to produce round keys from a master key for each encryption round. Among the algorithms that use this technique are the modern block ciphers, whose basic structures are divided into several categories, including the generalized Feistel structure (GFS), addition-XOR-rotate (ARX), and substitution-permutation network (SPN). Aside to that, permutation techniques (P-boxes) are widely applied to various

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

image and video encrypted communication processes, and they are recommended as a straightforward solution in the design of business and privacy engineering fields in social-use network systems where the recommended security level, as well as the potential cost demanded for a potential threat, are both low.

On the other hand,cryptanalysis is an assessment step in the process of the development of those ciphers that assists developers in designing more secure cryptosystems as well as measuring the overall operational efficiency of the proposed algorithm. This step could be performed during the construction process of cipher architecture or even after the final deployment.From this perspective, attempts to reduce the complicated task of cryptanalysis toward a computational analysis requiring only a basic understanding of cryptography have inspired scientists to explore the use of machine learning. Rather than relying heavily on cryptanalysts to develop a stronger cryptosystem structure, machine learning models have been widely implemented using data provided by the cryptosystem itself to simplify this audit step [1].

The early implementations of machine learning models in cryptanalysis focused primarily on training the models to imitate cipher behavior under the assumption of an available secret key, and while deep learning has recently piqued the interest of cryptography experts, particularly those specializing in block cipher cryptanalysis, the great majority of research has concentrated on deep learning-based black-box cipher attacks [2]. A block cipher is properly secured enough for effective use in cryptography if it has proven tolerance to different breaking cryptanalysis strategies over a specified time period, and it should be highlighted that resistance to differential cryptanalysis is among the most important security necessities for symmetric block ciphers. In this sense, differential cryptanalysis methodologies have recently been used for something altogether new: training machine learning algorithms for cryptanalysis applications. In this context, Rivest [3] studied the interactions between machine learning and cryptanalysis many years ago, whereas Gohr [4], who provided the first effective application of deep learning in the field of traditional cipher attacks, which has only recently gained popularity. By applying a machine learning-based differential distinguisher developed on differential data, the greatest cryptanalytic attack to date was achieved against a round-reduced Speck32/64. His results demonstrated that machine learning distinctions outperformed classical differential distinctions, opening the path for further study in the subject. As a result, further investigation is devoted to understanding the possibilities and limits of machine learning, which is often designed to evaluate the strength of cryptographic systems. In the classical investigation works on permutation techniques used for image encryption [43], [44], [46], [54], the researchers employed a known plain attack (KPA) to restrict the number of pairs required to partially or completely infer the permutation used key. The problem is that all of these studies concentrate on generating pictures with a uniform color distribution to reinforce the results while minimizing computation time and storage space. This uniform distribution of elements does not occur in natural images when using images without unified distribution or when deploying permutation algorithms in the operational mode, which reflects the work's weakness. Despite the higher recovery performance, it's indeed insufficient and cannot properly determine the appropriate permutation aspects. In the frequent circumstances of non-uniform color distribution, where the calculated size of the key search space rises appropriately, the outcome does not appear to be sufficiently pleasing. Furthermore, those linked research appears to be infeasible against lower color number ciphers, particularly white-and-black images, where reconstructing the real shape of the encrypted picture is impossible. However, these studies do not address another critical issue in the field of cryptography because all prior research is based on a black box attack that uses a predetermined number of (cipher/plain) pairs encrypted by permutation with no specification of the permutation rounds number or key generator pattern type.

B. CONTRIBUTIONS

This paper extends prior researches [43], [44], [46], [54] to overcome their drawbacks by using deep learning to assess P-box permutation methods and technologies widely used in multimedia encryption. Most studies that fit this condition in the literature attempt to recover the whole plain form of a particular cipher using classical research approaches and various optimization methods to find the used key or most parts of it by using black-box attacks. At the same time, those methods make it impractical in many situations because they go into important details and constraints of the optimization algorithms and parameters of methods used, whereas cryptanalysis's real purpose is limited to the acquisition of just the cipher sense and its basic concept, which can be reached by black-box based deep learning attacks without any complicated task of cryptanalysis and hard algorithm details. Furthermore, these approaches appear to be hard to reuse. On the other hand, our technique provides a relatively basic process that can easily be reused in the testing processes. Image files, unlike text files, have unique characteristics such as large data capacity, redundancy, and strong adjacent pixel correlation that necessitate the use of specialized strategies to deal with them in the encryption process to break the correlation of adjacent pixels. Among these are permutation algorithms, which are based on non-linear systems, and chaos theory.This technique appears to be beneficial in transferring media files and high-resolution pictures across insecure channels.

We explore the advantages of deep learning cryptanalysis techniques on the evaluation process, employing convolution neural network features ,by a black box attack, to discover the correlation among permutable entities to effectively and efficiently extract the plaintext from the ciphered text without any P-box knowledge.

We provide an automated decryptor based on deep convolutional neural networks that outperforms related work that

relies on traditional methods. We define a decryptor as a neural network model with the capability of decrypting a plaintext/plainimage without knowledge of additional details such as the small difference distribution [54],number of iterations, and P-box generation patterns.

– In our scenario, we focused on data that did not have a unified distribution. Because the data-driven methodologies utilized in prior research to demonstrate their potential benefits are based on the unified distribution of atoms, which fully matches the theoretical tests, which is a performance mismatch between the theoretical approaches and measured experimental execution. We show that even with limited data distribution, our neural decryptor successfully employs ciphertext pair aspects to maximize information extraction with low-data constraints in terms of the absence of uniform distribution that were not addressed in previous differential attack works implemented in previous studies. [43], [44], [46], [54].

– Also, this study focuses on applying machine learning techniques to expand the generally used model of differential cryptanalysis . We contribute to this area of study by studying the capabilities of deep learning to support differential cryptanalysis to measure the security of block ciphers. The assessment of block cipher security in which artificial neural networks are developed and employs a multitude of elements such as input and output distinctions, the number of iterations, and P-boxes used patterns. – The transfer learning skills exhibited in this study, on the other hand, demonstrate that determining approved input differences from scratch is also achievable by the networks with optimal background cryptography knowledge.

C. OUTLINE

The following is the structure of this document. Section 2 contains the famous published research on the use of machine learning in cryptanalysis. Section 3 provides an overview of the P-box block cipher for multimedia, as well as a brief notation and definition of the cipher. Many classical attacks on permutation ciphers are presented. In Section 4, many experiments to test the effectiveness of training DL-decryptors are presented. In Section 5, considerable experimental evidence is explored.Finally, in section 6, a brief overview of the relationship between our solution and previous literature works is provided.

II. RELATED WORKS OF USE MACHINE LEARNING IN CRYPTANALYSIS

Aron Gohr employed machine learning to construct an 8-round differential distinguisher for the SPECK32/64 cipher in 2019 [4], and based on it, an 11-round attack that surpassed earlier conventional techniques was built. Gohr's core aim was to use artificial intelligence to create new cryptanalysis attacks. By analyzing the output differences of the ciphertexts for a certain plaintext difference, he constructed a neural classifier in SPECK32/64 to discriminate between a block cipher and a random permutation.

Then, he evaluated this neural distinguisher against the famous SPECK32/64 all-in-one difference distribution database, which is also able to commute because of the small block size of the encryption, and the results showed that ML-distinguishers are an acceptable model behind it.

Recently, machine learning has been deployed to perform linear cryptanalysis. Hou *et al.* applied machine learning to achieve a linear attack on the DES encryption [5], employing known plaintext and ciphertexts. The findings show that in the DES cipher, a neural network can distinguish the XOR distribution of a linear expression. Other attacks, such as integral, have also been studied in connection with machine learning [6].

Modern studies in this field are not restricted to block ciphers: Liu *et al.* [7] use deep learning to evaluate the security of Xoodyak hash mode variations against preimage attacks. They developed a model to predict the message of a hash function for one round of permutation and reported that the accuracy was great. However, as the number of rounds increases, the efficacy of the deep learning preimage attack decreases.

With modest success, a similar method is employed to cryptanalysis lightweight cryptographic algorithms, FeW and PRESENT [9], [16] deep learning models were trained, verified, and validated on data that included plaintext, ciphertext, and intermediate round data created with the same encryption key. In the work of [10], he developed a learning algorithm to recover the secret keys of the Caesar and Vigenere poly-alphabetic and substitution ciphers. In [11] also generative adversarial networks have been employed to break these traditional cryptosystems. Machine learning algorithms and classification skills have been used to detect cryptographic algorithms from ciphertexts in the works of [12] and [13]. Classifiers were trained using known ciphertexts produced by a collection of six widely used cryptographic methods. Benamira et al [8] conducted a more detailed investigation of the operation of ML-based distinguishers, focusing on what information they employ, Their results demonstrate that these machines not only perform the differential distribution on ciphertext combinations but that the distinguisher is influenced by the penultimate or ante-penultimate round. They suggest a new pure cryptanalysis distinguisher with the same accuracy as Gohr's neural distinguisher based on their findings. [1] investigated the influence of block cipher characteristics on prediction accuracy by training deep learning algorithms to estimate the amount of active S-boxes for GFS cryptosystems. Deep learning has been used in both of these strategies, rather than just simpler, conventional machine learning techniques.

Further machine learning algorithm distinguishers and cryptanalysis against Simon, Speck, and non-Markov ciphers have also been introduced [14], [15]. The findings in [64] add another contribution by investigating the capability of linear and nonlinear machine learning classifiers in evaluating block cipher security in cryptanalysis using machine learning. According to their findings, machine learning models identify

a given block cipher result as secure or insecure depending on the number of active S-boxes. Nonlinear machine learning model types outperform linear models when evaluating inputs from previously observed ciphers during training, achieving prediction accuracy of up to 93%. But when evaluating inputs from other unobserved ciphers, nonlinear models outperform linear models with an accuracy of up to 71 %. These cryptanalysis works motivate us to investigate the use of deep learning techniques to overcome traditional cryptanalysis challenges in the most efficient manner possible.

III. BACKGROUNDS

A. PERMUTATION P-BOXES

A basic encryption scheme includes two major alternatives for accomplishing Shannon's confusion and diffusion metrics: substitution (S-box) and permutation (P-box) phases [17]. While the permutation stage changes the locations of atoms, the substitution stage changes the values of atoms. These two alternatives appear to be able to deal effectively with digital multimedia encryption while minimizing distraction, accidental deletion, and obfuscations caused by the encryption operation, and it must be noted that many block-cipher multimedia cryptographic structures involve independent processes for all permutation and substitution procedures [18].

Many block ciphers, including AES, substitution-permutation networks (SPN), and generalized Feistel structures (GFS), lightweight block ciphers, employ public permutation, and it is also a symmetric cipher for secret permutation, with the key being the secret for generating the permutation sequence [1].

B. DEFINITION AND NOTATION

Definition:
The function $P : B^n \rightarrow B^m$
P is called P-BOX if there exists a sequence $(i_k)_k^m = 1$,
Where $i_k \in \{1, \ldots, n\}$
such that for all $b \in B^n$ and $k = 1, \ldots, m$
we have: $P(b)_k = b_{ik}$.

- It just says: the K^{-th} item of $P(b)$ represents the i_K^{-th} item of b.
- P-BOX is a special type of S-BOX.
- P-BOXES permute, repeat or discard the elements of the input but do not change them.
- Because P-BOXES are a special type of S-BOXES, we denote that "$n = m$".

For multimedia permutation, the image M is represented as a two-dimensional array of positive integers with size (MN).

- Each element of M entry variable M_s is an image pixel s defined as: $M_{ls}(i, j)$.
- Where (i, j) and l are the pixel coordinates in M and the intensity value, respectively.
- Each element position is shifted to another place during the permutation step, to reconstruct the ciphered image without distortion, this technique must be a one-to-one correspondence.
- The secret permutation phase could be seen as a two-dimensional table, with each table member holding the new element location.
  Prm is a permutation table (P-box) of size MN that is defined as in following:

$$Prm = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ p_{21} & \cdots & p_{2n} \\ & & \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ p_{m1} & \cdots & p_{mn} \end{bmatrix}$$

- The permutation Prm, represented by the function T_k in the following : $C = MT_k = T_k(M)$.
- T_k is a bijection function that converts each element e_{ij} of M with location (i, j) to e'_{ij} a new location (i', j') referring to a key k over a set number of rounds with $i \neq i'$ and $j \neq j'$ and $i, i' \in \{1, \cdots m\}$ and $j, j' \in \{1, \cdots n\}$.

$$C = \begin{bmatrix} e_{11} & \cdots & e_{1n} \\ e_{21} & \cdots & e_{2n} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ e_{m1} & \cdots & e_{mn} \end{bmatrix} T_k$$

$$C = \begin{bmatrix} e'_{11} & \cdots & e'_{1n} \\ e'_{21} & \cdots & e'_{2n} \\ & & \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ e'_{m1} & \cdots & e'_{mn} \end{bmatrix}$$

- The permutation relationship can be expressed as a matrix $Prm[MN]$ that contains the cipher image pixels' positions of every clear image.
- This notion assumes that those cryptographic algorithms based on permutation can be deciphered using the same technique.
- As a result, the starting location of pixel p must be determined using the inverse function T_k^{-1}.
- The T_k function as well as its inversion T_k^{-1} are formed by the encryption private key k and have the same dimension as the considered plaintext witch is the block cipher size (P-box size).
- As a result, this framework reveals that image encryption reflects a symmetric block cipher, with an input size of (MN) as well as a key size of (MN).
- We conclude that all permutation approaches will be included within the $[MN]!$ possible scenarios that also reflect the greatest number of selected plaintexts that outcomes in a conclusion, and that permutation cryptanalysis should thus concentrate on those dimensions as a problem space [54].

Permutation methods are frequently included as multimedia content encryption techniques, and they are strongly

advised as an attractive option in the composition of marketing and security engineering disciplines in insecure channels where the requested security measures and the related impacts of a security threat are generally low [54].

Because of its simplicity, flexibility, size conservation, and effectiveness in masking apparent visual information, permutation is a common method in many secure multimedia systems. It is divided into two sorts, dependent on whether or not it is linked to a secret key: public permutations and secret permutations.

According to the digitized multimedia's particular format, the quantity of items that could be shifted is higher than its analog counterpart. Such items include bits [19], [20], [21], [22], bit-planes [21], [22], pixels [23], [24], [25], pixel blocks [23], [26], [27], transform coefficients [23], [28], [29], [30], variable-length code-words [30], tree nodes [31] motion vectors [30], [32], prediction errors [23], [32], [33], and several different sorts of those types [30], [32].

The encryption strategy is implemented directly on the picture frame in the spatial domain, and efforts in this aspect rely on direct modification of the picture intensities. The encryption in this field eliminates the correlation between the pixels. Using the opposite procedure, the pixel intensities in the picture could be fully restored without any errors.

Schemes in the frequencies field of cryptographic algorithms are focused on altering the frequency of the picture by transformations. Hence, recuperation of the original picture pixel intensities in the decryption process frequently results in distortion and information degradation.

The permutation transitional period is often carried out by using a variety of chaotic and non-chaotic strategies, such as using the logistic and Arnold's map [34], [35] for discrete chaotic maps. Furthermore, the Lorenz attractor and Chen's hyper-chaotic system are used to generate chaotic permutations [35], [36] with continuous attractors. A range of different techniques, including the chess-based horse movement [35] and the trajectory of a water wave movement [37], could be processed to create non-chaotic alternatives.

C. MULTIPLE CLASSICAL ATTACKS ON PERMUTATION CIPHERS

For a long time, researchers have been interested in the security of permutation encryption techniques. Because the unique architecture of analog video signals limits the flexibility of building elaborate permutations, the cipher text-only attack (COA) has an effect on Numerous permutation-only broadcast-TV devices [39].

For example, Bertilsson *et al.* [38] presented another approach to the famous architecture of Matias and Shamir [40], wherein every structure of a video is examined over to the other multiple pseudo-random space-filling curves to selectively retrieve the contents of the video by using the connection between subsequent frames.

Whenever secret permutations are used to encrypt multimedia content information, the situation appears to improve dramatically.

Furthermore, a cipher text-only attack (COA) can still be successful if certain correlations emerge between the items to be permuted. In [41], Li *et al.* showed a COA attack on row-column shuffled pictures by taking advantage of any connection between distinct rows and columns.

The above approach has been further expanded in [42] to break permutation-only picture encryption of pixel bits.

However, if the atoms of each element L are not low and the entropy contained in each element is significant, it is evident that finding for all $(L!)$ possibilities is very complex, and it hence COA is practically infeasible.

As a result, several research strategies propose building more complicated ways to generate secret permutations in order to provide higher security while also satisfying various additional application-dependent needs [25], [26], [30], [32].

Despite efforts to improve the resilience of permutation-only ciphers to cipher text-only attacks, most cryptographic algorithms of this type are vulnerable to plaintext attacks.

The impact of a chosen-plaintext attack (CPA), in which the adversary obtains the cipher text of a chosen plaintext, is increased by making all elements of the plaintext distinct from one another (input difference).

From the results of [43], Jolfaei *et al.* [44] demonstrated that the minimum constraint on the number of selected plaintext to completely extract the fundamental permutation pattern is $(Log_r l)$, where r seems to be the number of potential intensity.

It was hard to calculate how often these known plaintexts are required to properly break the fundamental permutation pattern in the case of a targeted attack (KPA), which represents an attack model that varies from CPA only in the implication that the adversary cannot choose the plaintext arbitrarily.

In general, L is much larger than r in multimedia data. According to the pigeonhole principle, certain values in $0, 1, \ldots, r_1$ must occur more than once. The same pixel quantity of 0 should occur approximately 512 times within the permutation encrypted ciphered image when a known plainimage of size (512x512) has a uniform distribution. As a result of witnessing this clear picture and the accompanying cipher one, there must be ([512]!) possibilities for one item in the permutation pattern whose pixel value corresponds to zero [54].

– Viewing many more pairs of known plainimages and encrypted images instinctively should eventually remove the uncertainty from such experiences [54].

Li *et al.* gave a quantitative analysis of the known plaintext attack on permutation-only multimedia algorithms in [43] by expanding the work in [45]. Their approach consists of two phases: partitioning a permutation sequence into various groups based on the atom values in every plaintext/ciphertext combination and finding the intersection of the sets between different pairings.

By establishing a tree structure, this approach was enhanced in terms of storage and computational complexity in [46]. Both of these studies came to the conclusion that the number of known plaintext is in the rang of ($Log_r L$) [54].

These two publications are famous for the development and study of lightweight multimedia encryption systems due to their universality [22], [44], [49], [50], [51], [52], [53]. From the standpoint of composite representation, Leo Yu Zhang re-analyzes the KPA attack on permutation-only ciphers.

Bianchi *et al.* [47], [48] confirmed it in a series of publications, suggesting that it is used to minimize the dimension of secret text and increase the speed of linear processes on ciphered data created using additive homomorphic cryptograms.

He provides a complete theoretical study of the KPA cryptanalysis on permutation-only systems using composite representation. Many KPA algorithms are implied by the composite representation, one of which outperforms the well-known ''optimal'' approach in terms of faster computation with the same storage. Leo Yu *et al.* [54] present a complete theoretical investigation of the KPA attack on permutation-only ciphers, in contrast to prior work [43], [44], [46], Many KPA algorithms are implied by the composite representation, one of which outperforms the recognized ''optimal'' approach in terms of faster computation with the same storage [54].

IV. OUR NEURAL DECRYPTOR

A. DATA SETS

For training and cracking tests, the MNIST [55] and FASHION MNIST [56] data sets are utilized.

Specifically, we use 60000 plain pictures from the MNIST and FASHION MNIST data sets to generate our encrypted images, which represent the training set, and the remaining 10000 ciphered images from the same dataset for the test set.

The non-uniform distribution of colors in each image would be the main motivation for selecting such a dataset where the intensity of the picture is stored as a number between 0 and 255, providing each pixel with 256 different possibilities. The number 0 represents black, whereas the value 255 represents white, and the intermediate values are grayscale levels ranging from black to white.

The histogram is an effective instrument in image processing because it illustrates the intensity (or color) distribution of a picture.

The distribution of color intensity created by the MNIST and Fashion MNIST databases has been calculated, as illustrated in the graphics below. In which we can see the table values as well as the shades of gray that describe the image in the absence of a uniform distribution.

Figures 1 and 2 demonstrate 16 samples from the fashion MNIST and MNIST data sets, respectively, while tables 3 and 4 indicate the intensity distribution of their corresponding samples from the two data sets.

B. DEEP LEARNING MODEL

1) NEURAL DECRYPTOR

The model computes the difference between various inputs and outputs based on the dataset, using numerous parameters such as batch size and pixel values, and the present key characteristic is the correlation between adjacent pixels in two-dimensional space.The decryptors' objective is to extract the visual difference between the inputs and the outputs, which would be formally defined as follows:

- $\Delta d = Input_1 \otimes Input_2$
  where $Input_1$ and $Input_2$ are two distinctive plainimages and $\otimes$ denotes to the dissimilar function.
  Because we work in a two-dimensional space, the dissimilar function represents the distance between two images.For more details, if we place two images that have the same size on top of each other, the function represents the number of pixels in the same position in the two images that have different colors and this can be noted as follows:
  $Input_1 = Img_1$ and $Input_2 = Img_2$.
  The dissimilar function represents the number of pixels P_{ij} with the condition $P1_{ij} \neq P2_{ij}$ and $P1_{ij} \in Img_1$, $P2_{ij} \in Img_2$.
- $\Delta d' = Output_1 \otimes Output_2$,For an outputting distinction, $\Delta d'$ can be alternatively constructed by exploiting a pair of relative cipher images $Output_1$ and $Output_2$.
- The underlined I-iteration differential pathway like the propagation of Δd to $\Delta d'$ after i iteration of permutation is represented by: $\Delta d \xrightarrow{i} \Delta d'$.
- Every differential direction must have a specific probability of holding: $Pr(\Delta d \xrightarrow{i} \Delta d') = a^{-p}$ in the case of unified distribution of the intensity I with $I \in \{0, \ldots, a\}$.

witch the current work seeks to reduce because differential patterns can be used as a statistical or quantitative distinguisher [43], [44], [46], [54] for permutation cryptanalysis attacks.

In the case of a non-unified distribution, every differential direction must have a specific probability of holding:

$Pr(\Delta d \xrightarrow{i} \Delta d') = a^{-b}$ with $b > p$, posing a performance gap in current works [43], [44], [46], [54]. We show in this paper that, even with the absence of data distribution, our neural decryptor successfully employs aspects of ciphertext pairs that are not addressed by the previous differential works.

2) THE CHOICE OF MACHINE LEARNING MODEL

Test evaluation is conceptualized as a regression problem for a supervised model in which layers of the model are trained by many characteristics such as variations between input and output, number of iterations, and P-box generation patterns.

Deep learning algorithms are used to find a decryptor because they can detect hidden structures in digital information besides the need for explicit intentional feature extraction engineering.

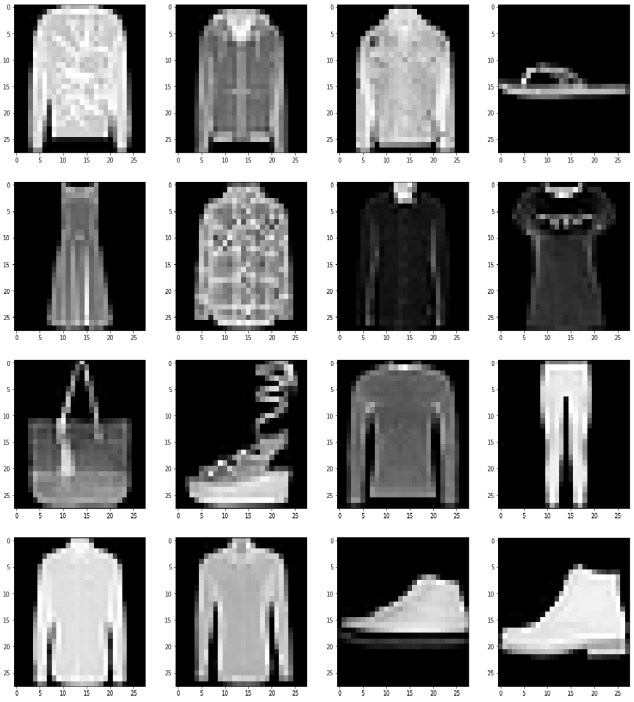


FIGURE 1. Fashion MNIST samples.

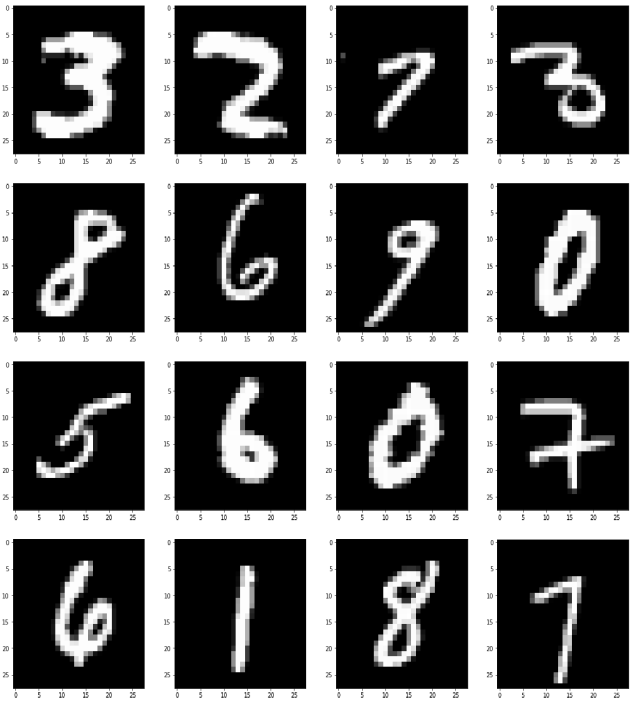


FIGURE 2. MNIST samples.

We experimented with several neural network types, such as the basic Multi-Layer Perceptron (MLP), deep neural network (DNN), convolutional neural network (CNN), and long-short-term memory network (LSTM).

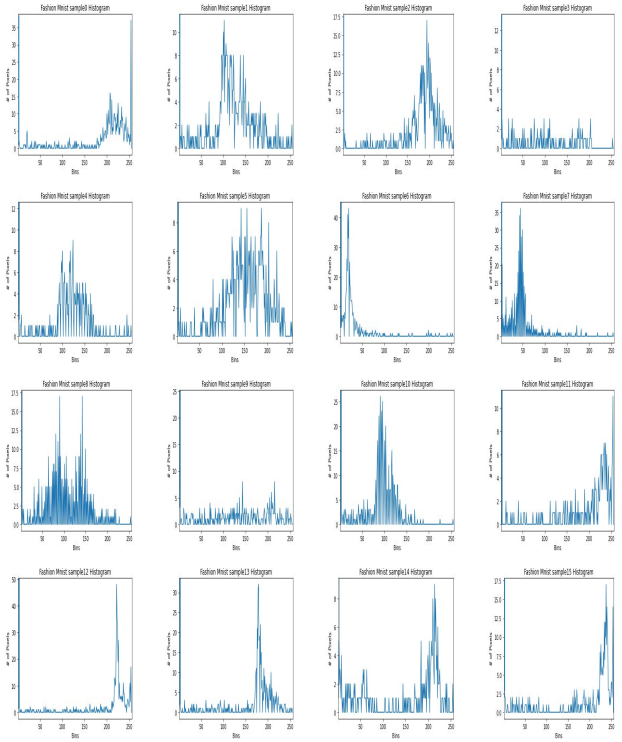


FIGURE 3. Corresponding Fashion Mnist samples color intensity distribution.

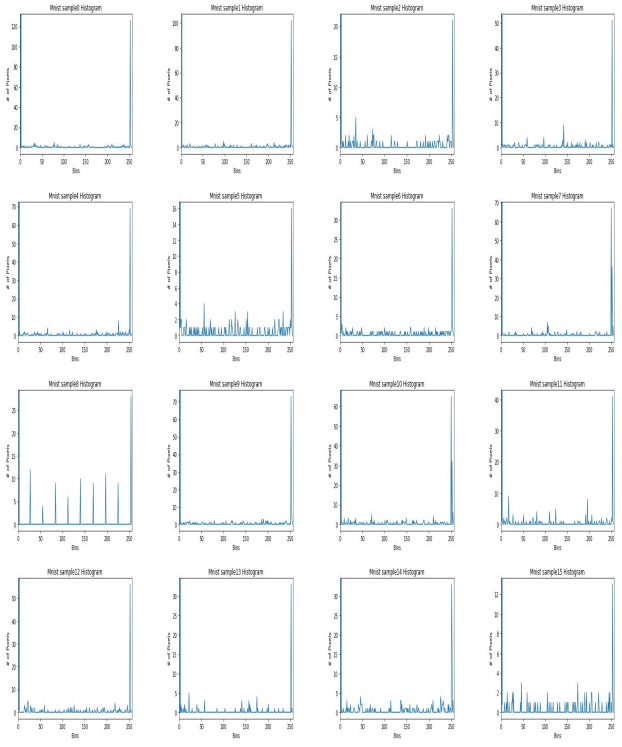


FIGURE 4. Corresponding MNIST samples color intensity distribution.

To get the highest accuracy and learning speed, we investigated the width (number of neurons for every layer) and depth (number of hidden layers) of the latter.

We also performed experiments including several sorts of activation functions as well as the weights initialization technique.

After many experiments, we found that the CNNs [57] are appropriate for the task of identifying a decryptor .

The main reason for this choice is that CNNs are designed to recognize patterns in input data, which aids the differential process, and it works for every input where two-dimensional data is connected in any manner.

Convolution is based on three key techniques that can help in the enhancement of a machine learning approach, which are:

- Sparse interactions: Classical neural networks employ matrix multiplication by a table of features with distinctive parameters specifying the relationship between each incoming and outgoing unit. This signifies that each element of output communicates with each element of entry, which is not the case with convoluted neural networks. This is accomplished by making the core lower than the entry.
  This ensures that the model must store fewer parameters, which reduces the model's memory requirements and improves its efficiency. This also means that the calculation of the departure requires fewer operations. These efficiency improvements are often significant. Despite the limited distribution of data, this property enables our proposed model to establish a strong distinction between the corresponding inputs and outputs, as well as the different color intensities [57].
- Parameter sharing: This refers to the use of the same characteristic for more than one function in a model. In a convolution network of neurons, each core element is used at each entry location. The parameter sharing performed by the kernel size ensures that instead of learning a set of different parameters for each location, we just discover one set, which minimizes the model's size of the data requirements even further [57].
- Equivariant representations: In the case of convolution, the layer has equivariant interpretations due to the special property of parameter sharing.
  When a function is said to be equivariant, it appears to mean that even if the input changes, so too does the output. In particular, a function $H(x)$ is equivalent to a function K if $H(K(x)) = K(H(x))$. This property enables our proposed model to establish a strong link between the plain and cipher pairs [57].

3) ARCHITECTURE

We proceed by recovering permutation pattern information from cipher images using a convolutional encoder network. Furthermore, using a symmetric deconvolutional generator network, we construct encrypted pictures from the features to match their equivalent plainimages. To decrypt the P-box encryption technique, we must first involve a strong mapping function that can be expressed as the inverse transform between encrypted and plainimages.

- Deep convolutional neural networks (CNN) [57] are used to mimic such complex inverse characteristics. In Fig.5, the system is split into convolutional and deconvolutional groups.
- The input is encrypted images specifically mentioned as X in convolutional groups, and we start generating six convolutional layers to quantify input image composition that gets low-dimensional characteristic features, with the operating condition described as $Y = O(X)$.
- All these characteristic features will be used to define the dense layer parameters for profound understanding to detect hidden features in data sets without the need for intentional feature selection.
- Form the dense layer we reverse the convolution stage in deconvolutional groups and reestablish the basic images with good accuracy.
- The inverted procedure is expressed by the following equation: $X = H(Y)$.
- The regenerated images are compared to corresponding ground truth plainimages presented as objective T, with the error function $MeanSquaredError(MSE)$.

4) HYPER-PARAMETERS

When deploying machine learning algorithms, the hyper-parameters that make the biggest difference for a particular task must be chosen. These parameters are often determined experimentally by analyzing multiple network topologies and adhering to best practices. There are automated ways of tuning the hyper-parameters [60], but they demand significant resources that can be difficult to replicate. Following that, we provide the results of the manual architectural search.

The remaining hyper-parameters that were correctly applied in our interesting experiments are listed below:

- Initial Learning rate: 0.1.
- Batch Size: 2000.
- Epochs: fixed in 1500.
- Trainable parameters: 679 338.
- Weights initialization: Xavier Initialization [61], also known as Glorot Initialization, is a neural network initialization strategy. Biases are set to zero, and for each level, the weights W_{ij} are established as:
$$W_{ij} = Ds\left[-\frac{1}{\sqrt{Prv}}, \frac{1}{\sqrt{Prv}}\right] \quad (1)$$
  where Ds is a uniform distribution and Prv is the dimension of the preceding layer (the number of columns in W).
- Optimizer: As an optimizer, we used the Adam algorithm [58]. Since it slightly differs from the classical gradient descent we presented before, we give a brief explanation here. We denote two sequences:
$$x_t = \gamma_1 x_{(t-1)} + (1-\gamma_1) f_t \quad (2)$$
$$y_t = \gamma_2 y_{(t-1)} + (1-\gamma_2) f_t^2 \quad (3)$$
  x_t and y_t are respectively 1^{st} order (mean) and 2^{nd} order (variance) gradient estimates.

and $f_t = \nabla E\theta^{(t-1)}$
where $\theta(t)$ represents as before our trainable parameters, E is our loss function and γ_1, γ_2 are constants.

- Error function:
  - The mean squared error (MSE) [59] of a regression line indicates how near it is to a set of values by squaring the distances between the values and the regression line (the ''errors''). Squaring is essential to remove any negative parameters. Larger differences are also found to be more significant. The mean squared error is so named because we are computing the average of a sequence of errors. The smaller the *MSE*, the more accurate the prediction. Below is the description of the mean squared error:

$$MSE = \frac{1}{a}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{4}$$

    With:
    * n is the number of items.
    * Σ is summation notation.
    * y_i represent original ground of truth or observed y-value.
    * $\hat{y}_i$ is the predicted y-value from the model.
  - Instead of single-point predictions, the quantile loss function is used to forecast intervals or ranges of predictions. The quantile regression loss function is used to predict quantiles, as both the title and subheading indicate. A quantile is indeed the value from which a particular amount of observations in a group are derived.
- The coefficient of determination ($R-squared$) is a metric that offers information about a model's goodness of fit. In the framework of regression, this is a measurable statistic about how well the linear regression accurately simulates the correct information. It's indeed significant when using a quantitative model to better estimate outcomes or validate hypotheses. There are other versions (see reminder below); the one shown here is the most commonly used:

$$R - squared = 1 - \frac{\text{SSR}}{\text{SST}}, \tag{5}$$

$$= 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}. \tag{6}$$

  with *SSR* is the sum squared regression and *SST* represents the total sum of squares. The complete sum of squares would be the sum of the information's distance from the average squared, whereas the sum multiplied regression is the total of the residuals squared. It would only handle values between 0 and 1 because it is a percentage.
  The residual, according to each measurement, is the difference between the estimated and observed values of the parameter y.

$$\text{Residual} = \text{actual } y \text{ value} - \text{predicted } y \text{ value}, \tag{7}$$

TABLE 1. Convolutional groups parameters.

Conv	Filters	Kernels	Strides	Padding	Parameters
1	140	(1,1)	(1,1)	valid	280
2	112	(1,1)	(1,1)	valid	15792
3	84	(1,1)	(1,1)	valid	9492
4	56	(1,1)	(1,1)	valid	4760
5	28	(1,1)	(1,1)	valid	1596
6	1	(1,1)	(1,1)	valid	29

TABLE 2. Dense layer parameters.

Dense Layer	Neurons	Trainable parameters
1	784	615440

$$r_i = y_i - \hat{y}_i. \tag{8}$$

A negative residual indicates that the desired value was too great, whereas a positive residual indicates that the value obtained was too lower. A regression line's goal is to minimize the sum of residuals.
For calculating residuals, recognizing that:$r_i = y_i - \hat{y}_i$ and understanding that the regression contains the equation: $\hat{y}_i = a + bx_i$
The residual of observation is calculated as follows: $r_i = y_i - \hat{y}_i = y_i - (a + bx_i)$

- Activation function: The linear activation function was chosen for any situation in which activation is roughly proportional to the input. It is also known as ''no activation'' or ''identity function''.
  The function makes zero variations to the weighted combination of the parameters; it really just returns the value that was therefore provided. In our situation, using this function well preserves the parameters generated by the Adam optimizer and strengthens the effectiveness of the convolution features.

5) PADDING, STRIDES, KERNELS AND FILTERS

- The first important Conv-2D measurement is the total of filters that the convolutional layer should receive.
- The depth of the kernel, which is a 2-tuple indicating the size of the 2D convolution frame, is the next essential factor that must be supplied to the Conv-2D class. The kernel size must be an integer value as well.
- The strides configuration is a pair of integers that describes the movement of the convolution along the input volume's x and y dimensions.
- The padding argument of the Conv-2D class could have one of the two possible paratetrs: valid or the same. By using the valid measurement, the entry dimension is not zero-padded, so the spatial perception has been restricted naturally through the use of convolution.

Figure 5 illustrates the model architecture and tables 1, 2 and 3 represent the convolutional groups parameters, Dense layer and De-convolutional groups parameters respectively.

6) THE TRAINING GOALS OF THE DEEP LEARNING MODEL

One of the main goals of image encryption algorithms is to break down the correlation between adjacent pixels as

TABLE 3. De-convolutional groups parameters.

Deconv	Filters	Kernels	Strides	Padding	Parameters
1	28	(1,1)	(1,1)	valid	56
2	56	(1,1)	(1,1)	valid	1624
3	84	(1,1)	(1,1)	valid	4788
4	112	(1,1)	(1,1)	valid	9520
5	140	(1,1)	(1,1)	valid	15820
6	1	(1,1)	(1,1)	valid	141

TABLE 4. Pseudo code of the model training algorithm.

Algorithm

Input: Ciphered image data set **C** , Plainimage data set **P**, loss functions **L**, initial learning rate **r**, number of epochs **E**, Optimizer **O**, Initializer **I** , Batch size **B** .
Output: Generated image **G** .
Initialize the weights and bias of the algorithm by **I** technique
For e = 1 to E do
  For i = 1 to |C| do
    -Extract the i^{ith} sample c_i cipher image from the dataset **C**.
    -Extract the i^{ith} sample p_i plainimage from the dataset **P** corespondent to the ciphered image c_i .
    -Forward propagate the sample c_i through the model M to obtain the output generated image **G**.
    -Compute the loss L using the output generated image**G** and the labeled p_i plainimage from the dataset **P** .
    -Back-propagate the loss L through the M model with **O**.
    -Update the weights and bias of the M model using the **O** and **B**.
    -Update the learning rate **r**.
    -Save the best weights and bias of the M model .
  end for i
end for e
return the generated image **G**

much as possible. Because images contain important visual information that can be seen by simply overlaying the correlations of adjacent pixels. In order to predict clear images from encrypted images by implicit exploration of this correlation, one of the important features of our model is to rediscover and reconstruct this correlation by training the model with several different input and output (plain/cipher) pairs for permutation feature extraction.

The main goal is to train a model that can predict a clear image from its corresponding ciphered image. but at the same time to find a model that can be trained with different characteristics to distinguish the clear image from its encrypted counterpart, whatever the permutation technique and the number of rounds.

The problem was designed as a regression problem for a supervised model. It mean that the model will be trained to predict the clear image from its corresponding encrypted image or in other words the inputs of the model are encrypted images and the outputs are images generated to be like the originals plainimages. In the output, the resulting images obtained will be compared to the labeled data, which are the original image of Mnist and fashion Mnist, by an error function which will measure the results of the model. Then, after each iteration, an update to the parameters of the model will be carried out according to the chosen optimizer. The system will save the weights and biases corresponding to the best results obtained at the end of each iteration, and the training

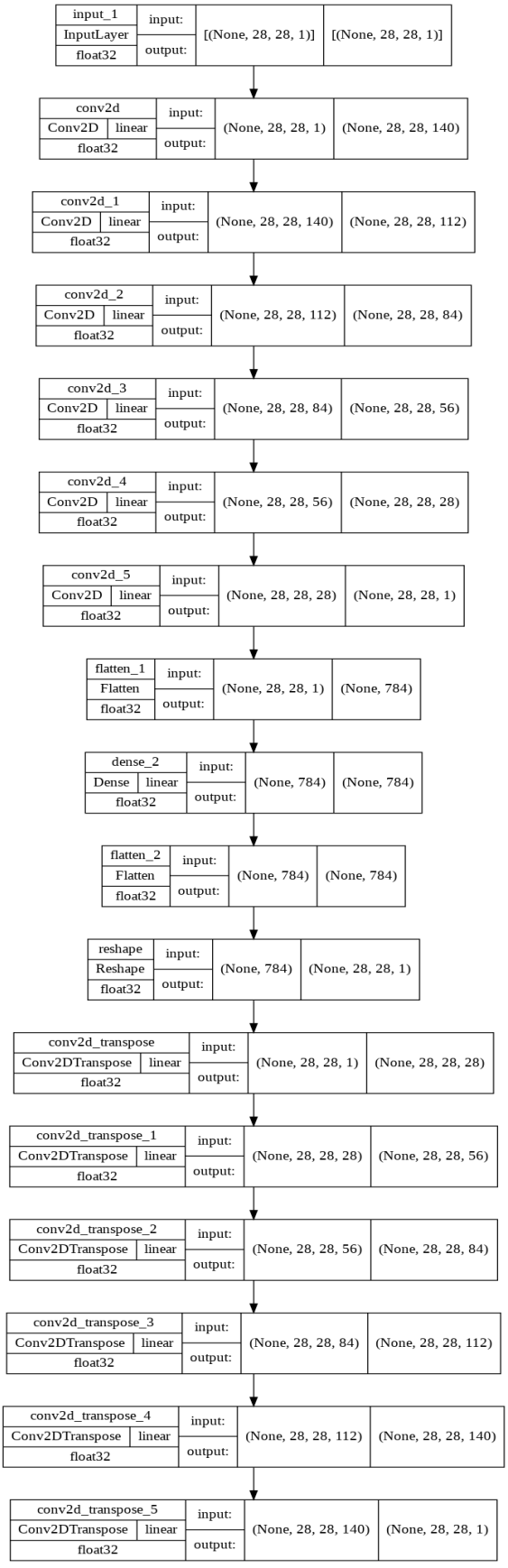


FIGURE 5. Model architecture.

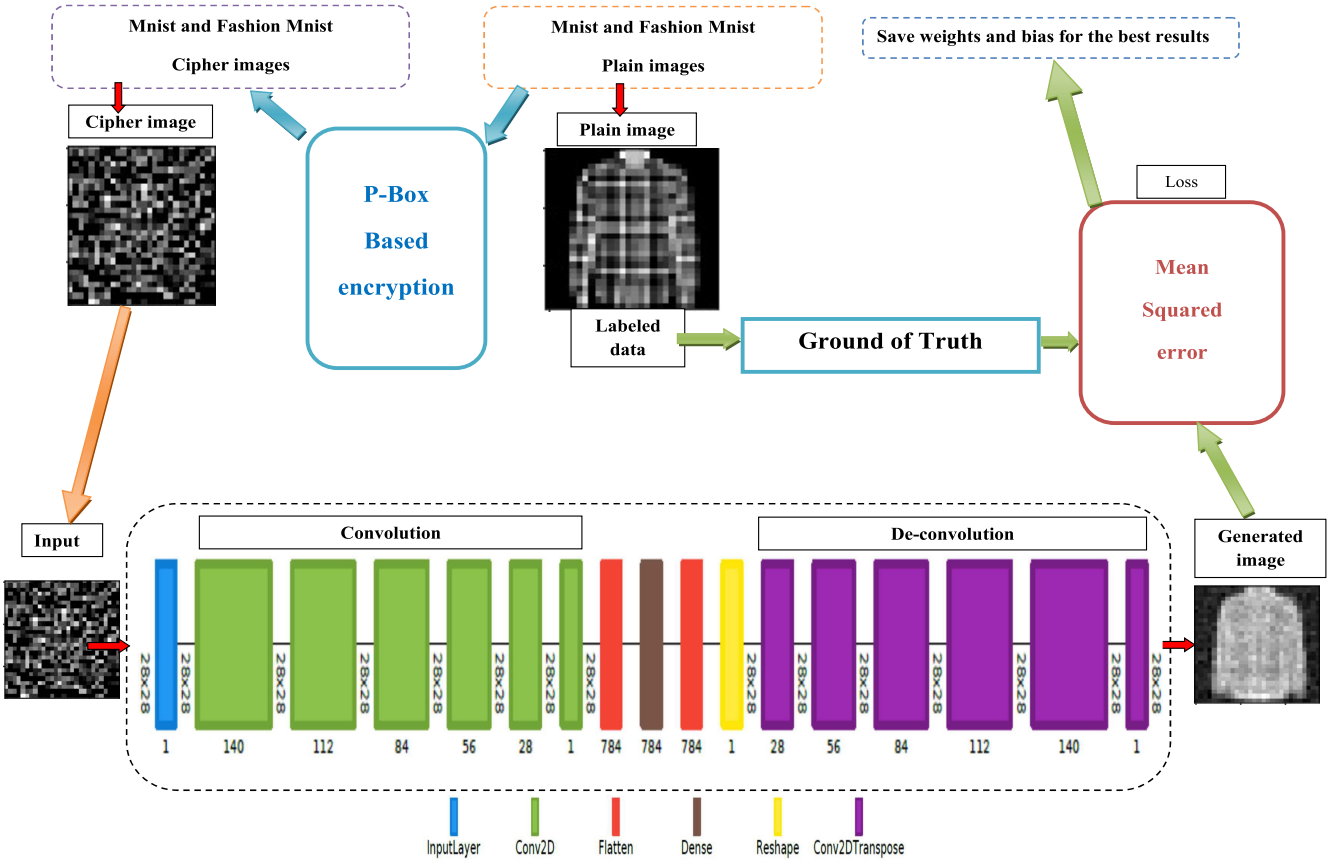


FIGURE 6. The model training process.

will remain in loop until the number of iterations (epochs) completed. To examine permutation pattern measurements in the context of visual cryptography, we used four chaotic and non-chaotic system permutation algorithms on images from the Mnist and Fashion mnist data sets and to strengthen our study and better validate the established model, we employed these patterns as permutation key generators.

Following the production of these permutation keys, the datasets Mnist and Fashion mnist are encrypted by these patterns to produce 60000 encrypted images of each, which are used as inputs to the trained model, and the remaining 10000 images are encrypted for use in the model validation step. To simplify training, the encrypted images are arranged in the same sequence as the clear original images of the Mnist and Fashion mnist datasets. The model additionally makes use of the original images from the Mnist and Fashion mnist datasets as label data. Figure 6 depicts the training procedure, and table 4 provides a pseudo-code of the training algorithm's strategy.

V. EXPERIMENTAL RESULTS

A. PERMUTATION PATTERNS FOR P-BOXES GENERATION

We applied four chaotic and non-chaotic system permutation algorithms on pictures from the Mnist and fashion mnist data sets to investigate permutation pattern measures in the context of visual cryptography. The methodological approach was used to generate P-box permutations of overall pictures with dimensions of 28×28 as follows:

1) DISCRETE CHAOS

The logistic map is utilized in this study case to produce a series of numbers, However, any discrete chaotic map can also be employed in the same manner.

After sorting these values ascendingly, the scoring system for every integer in the sorted series is used to fill the permutation P-BOX.

The standard logistic map with parameter λ looks like this:

$$r_{n+1} = \lambda r_n (1 - r_n) \tag{9}$$

The discrete chaotic system was iterated $(\frac{MN}{spc})$ rounds for P-Box of size MN, where spc is the value of algorithm output parameters and it represents the lowest integer higher than or equal to $(\frac{MN}{spc})$.

2) CONTINUOUS CHAOS

The Lorenz system is utilized in this study case to produce a series of numbers, However, any continuous chaotic system

can also be employed in the same manner.

$$\begin{pmatrix} \dot{a} \\ \dot{b} \\ \dot{c} \end{pmatrix} \begin{pmatrix} -10 & 10 & 0 \\ 8 & 4 & 0 \\ 0 & 0 & -8/3 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} + \begin{pmatrix} 0 \\ -ac \\ ab \end{pmatrix} \quad (10)$$

To begin, the three output frames are modified to remove short-term dependability. Then, every variable from every sequence is combined together to form a single sequence. To complete the permutation matrix, this series is sorted in ascending order, and a scoring system is supplied for each value in the sorted series. For a P-Box of size MN, the chaotic system is iterated ($\frac{MN}{spc}$) times.

3) GRAY CODE BASED PERMUTATION METHOD (*GCBPM*)

In [64] method, a basic Gray-code-based permutation strategy is used, using the bijective non-linear map described by the following equation as the basis.

$$\gamma = \theta \oplus (\theta \gg (\beta + 1)) \quad (11)$$

where θ is a $k - bit$ number, γ is a Gray-code value of k bits, $\oplus$ is the binary XOR operation, β is an integer, and $\gg$ is the binary right shift. A k-bit number's Gray-code is also a k-bit number.

The image is turned into a one-dimensional array of pixels in order for this code to complete the permutation procedure. This method takes four digits β_1, β_2, δ_1 and δ_2 as input. It is worth noting that δ_1 and δ_2 are k-bit integers. Two Gray-code values, I_1 and I_2, are calculated for each pixel location. Where: $I_1 = GRAY(\theta, \beta 1) \oplus \delta_1$ and $I_2 = GRAY(\theta, \beta 2) \oplus \delta_2$

Then, in the permuted image, take the pixel at position $X1$ and insert it in spot $X2$.

4) COUPLED MAP LATTICE(CML)

Coupled map lattice [63], a dynamical system with discrete time and discrete space, is employed in the manner stated by (12). This system has a long enough period to be employed in crypto-systems, and its output is transformed to integer numbers using (13). The generated numbers are then used to conduct right cyclic shifts to the image's rows and up cyclic shifts to the image's columns.

$$f_{d+1}(k) = (1 - \epsilon)\, \tau\, (f_d\, (k)) + \epsilon \tau\, (f_d\, (k - 1)) \quad (12)$$

$$s_d = mod\left(f_d\left(k\right) 10^{16}, X\right) \quad (13)$$

where $k = 1, 2, \ldots, L$ is the lattice site indices, L is the lattice width, and $f_d(k)$is the constant variable for the Kth site at the instant d. ϵ is the coupling parameter, which is one or zero, X = N for row shifts and X = M for column shifts. When the map τ is chaotic, the entire system is chaotic. The coupled map lattice system is then repeated $Max(MN)$times for a MN-sized P-box.

B. TRAINING EXPERIMENTS, TRANSFER LEARNING AND PREDICTION

All of the following experiments has been carried out on Google Collaboratory with the Back-end Google Compute Engine (Free GPU NVIDIA Tesla K80) and 12 GB RAM employing Python 3.7.13, TensorFlow 2.8, and Keras API,The source code is available from GitHub.[1]

We were using the Keras checkpoint called Call Backs to preserve the much more intended results during every iteration, as well as the weight and bias of the CNN model. To demonstrate the scope of training a machine learning-based decryptor by exploiting significant differences between (Plainimages) and (cipherimages), we set up an experiment in which DL-decryptors are trained in a single round, eight rounds, and sixteen rounds with the following parameters:

First, we trained the model on data from the Mnist data set and the results are presented as following.

1) TRAINING EXPERIMENTS WITH ONE ROUND P-BOX BASED ENCRYPTION

We conduct preliminary experiments on smaller-scale one-permutation ciphers before moving on to larger 8 and 16 rounds equivalents to assess the effectiveness of the experimental evaluation. Mnist data set images are used to produce samples with four distinct permutation mechanisms and the following are the pattern specifications designed to automatically generate one round permutation keys:

- Chaotic system 1: The logistic map with $r_0 = 0.448$ and $\lambda = 3.988$.
- Chaotic system 2: The Lorenz system with $a_0 = 6.293$,$b_0 = -6.749$ and $c_0 = 2.886$.
- Non chaotic system 1: Coupled map lattice with $x_1 = 0.31457$,$y_2 = 0.6532$ and $\epsilon = 0.94$.
- Non chaotic system 2: Gray code based permutation with $d_1 = 1$, $d_2 = 28$, $\delta_1 = 29493$, $\delta_2 = 23749$.

This makes it possible to create a large number of encrypted images in a reasonable amount of time: 60,000 encrypted images for training and 10,000 encrypted images for testing. The structure of the ciphered pictures is described as a two-dimensional array with elements ranging from 0 to 255. The numbers 0 and 255 indicate black and white, respectively, while the intermediate values are grayscale levels ranging from black to white.

Each sample of the dataset used to train a deep learning model contains block cipher-related characteristics. In this initial experience, we have four models. The labeled data of the model is the clear pictures that correspond to the data sets. To simplify training, the encrypted images are arranged in the same sequence as the clear original images of the Mnist data set. The model additionally makes use of the original images from the Mnist dataset as label data. Our experimentation is divided into these main phases:

-With one encryption process cycle, generating the permutation keys of the four permutation patterns structured for the encryption algorithm of the block cipher.

-The generation of cipher Mnist training dataset based on generated keys: 60,000 samples for training and 10,000 samples for model validation.

[1]This paper's supplementary code is accessible at https://github.com/zakariatolba/multimedia-p-box-assessment.git.

TABLE 5. Pseudo code of P-box based permutation encryption algorithm.

Algorithm
Input: Plainimage data set **P**, Number of rounds **R** , P-box Generation pattern **F** .
Output: Generated Ciphered image data set **C** .
For r = 1 to R do
-Generating the permutation key K_r from the **F** permutation patterns.
For i = 1 to |P| do
-Extract the i^{ith} sample p_i plainimage from the dataset P.
-Encrypt the i^{ith} sample p_i image from the dataset P by the key K_r to get c_i.
-Save the encrypted c_i as the i^{ith} sample C_i image of the generated dataset **C**.
end for i
-Use the encrypted data set **C** as the plain data set **P** for the following round **P <– C**.
end for r
return the generated ciphered images data set **C**.

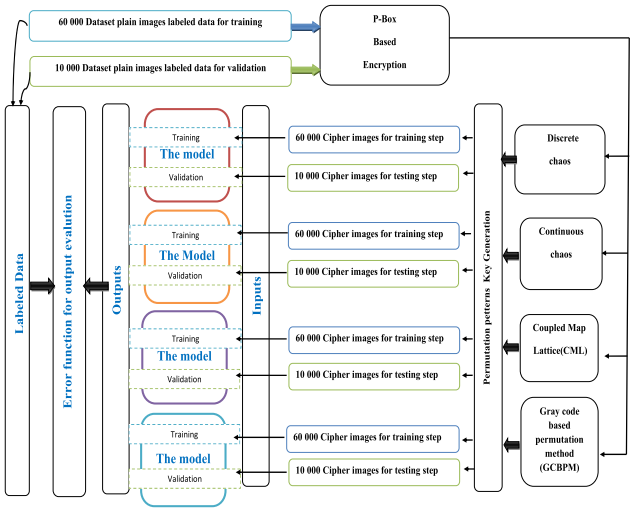


FIGURE 7. Data set generation and labeled data for one round experiments.

-Generation of ciphered Fashion Mnist encryption training datasets based on the same produced keys for the four permutation patterns: 60,000 samples for a transfer learning test and 10,000 samples for each model's prediction.

-Training the four models with the ciphered mnist data sets and saving the best results for each one.

-The reusing of the four models trained by encrypted mnist images as deployment models for prediction images from the ciphered fashion mnist data set.

Figure 7 depicts the data sets generation procedure, and table 6 provides the training results for the Mnist data set.

2) TRAINING EXPERIMENTS WITH 8 ROUNDS P-BOX BASED ENCRYPTION

To conduct a more in-depth investigation and better test our model, we increase the number of rounds of parmation from one to eight in order to complicate the operation of permutation, effectively break the correlation between the pixels, and make the developed models predict the text from the more

TABLE 6. The ciphered Mnist data set one round training results.

Patterns	Loss	R^2	MSE
Discrete chaos	0.0842	1	0.1365
Continuous chaos	0.0708	1	0.3144
Gray code based permutation	0.0239	1	0.0052
Coupled map lattice	$8.7997e^{-04}$	1	$1.8217e^{-04}$

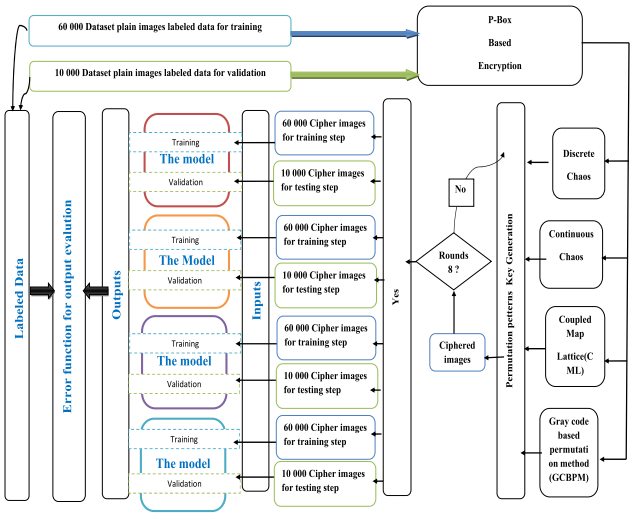


FIGURE 8. Data set generation and labeled data for 8 rounds experiments.

difficult ciphertext. The specifications designed to automatically generate the first round of permutation keys are the same as for the first experiments, but it should be highlighted that in the case of multiple rounds, the key generation stage will be conducted in accordance with the number of rounds. In the case of eight rounds, the first key is created from the baseline parameters specified above, the second key from the first one, the third key from the second one, and so on until the last round.

This experimentation is divided into these main phases:

-After employing the encryption approach given in Table 5 we get:

-The generation of cipher Mnist training dataset based on generated keys: 60,000 samples for training and 10,000 samples for model validation.

-Generation of ciphered Fashion Mnist encryption training datasets based on the same produced keys for the four permutation patterns: 60,000 samples for a transfer learning test and 10,000 samples for each model's prediction.

-Training the four models with the ciphered mnist data sets and saving the best results for each one.

-The reusing of the four models trained by encrypted mnist images as deployment models for prediction images from the ciphered fashion mnist data set.

Figure 8 depicts the data sets generation procedure, and table 7 provides the training results for the Mnist data set.

TABLE 7. The ciphered Mnist data set for 8 rounds training results.

Patterns	Loss	R^2	MSE
Discrete chaos	0.0471	0.9997	1.5738
Continuous chaos	0.0278	0.9994	3.8839
Gray code based permutation	0.0294	0.9998	1.3939
Coupled map lattice	0.0655	1	0.0721

TABLE 8. The ciphered Mnist data set for 16 rounds training results.

Patterns	Loss	R^2	MSE
Discrete chaos	0.1061	0.8507	922.6311
Continuous chaos	15.1446	0.6793	1957.4821
Gray code based permutation	0.2193	0.9998	1.3684
Coupled map lattice	0.0724	0.9996	2.3448

3) TRAINING EXPERIMENTS WITH 16 ROUNDS P-BOX BASED ENCRYPTION

To conduct a more in-depth investigation and better test our model, we increase the number of rounds of parmation from "8" to "16" in order to complicate the operation of permutation, effectively break the correlation between the pixels, and make the developed models predict the text from the more difficult ciphertext. The specifications designed to automatically generate the first round of permutation keys are the same as for the first experiments, but it should be highlighted that in the case of multiple rounds, the key generation stage will be conducted in accordance with the number of rounds. In the case of 16 rounds, the first key is created from the baseline parameters specified above, the second key from the first one, the third key from the second one, and so on until the last round. This experimentation is divided into these main phases:

-We receive after employing the encryption approach given in Table 6: -The generation of cipher Mnist training dataset based on generated keys: 60,000 samples for training and 10,000 samples for model validation.

-Generation of ciphered Fashion Mnist encryption training datasets based on the same produced keys for the four permutation patterns: 60,000 samples for a transfer learning test and 10,000 samples for each model's prediction.

-Training the four models with the ciphered mnist data sets and saving the best results for each one.

-The reusing of the four models trained by encrypted mnist images as deployment models for prediction images from the ciphered fashion mnist data set.

Figure 9 depicts the data sets generation procedure, and table 8 provides the training results for the Mnist data set.

4) TRANSFER LEARNING EXPERIMENTS

The objective of the patterns which used to produce the p-boxes is really to construct permutation keys in such a way that they seem random (pseudo-random generators), but these patterns simultaneously allow for the inverse operation, which is decryption without loss of data. The model is not distinct in itself, but it improves in the identification of decryptors . In other terms, a model trained on data encrypted with one round CML is distinguishable from a P-box-based CML one-round encryption algorithm. All of the models have the same architecture, layers, and hyper parameters, but the key difference between them is the parameters acquired during the training process (weights and bias).

After training the four models with the Mnist data set, we attempted to employ learning transfer by using the weights and bias of the models from the first model trained by the Mnist data set for one, eight, and sixteen rounds as deployment models for the Fashion Mnist models for one, eight, and sixteen rounds with the same permutation patterns and the same algorithms parameters, respectively.

The most remarkable conclusion is that, without any training, the assessment process converges toward desirable findings and the error function is reduced.

It should be highlighted that by combining transfer learning with optimal prior cryptographic competence, it is also possible to develop acceptable decryptors from the ground up by utilizing the transfer learning techniques described in this paper. The results obtained are presented in tables 9, 10 and 11 respectively. Furthermore, the experience of learning transfer reusability to improve model performance

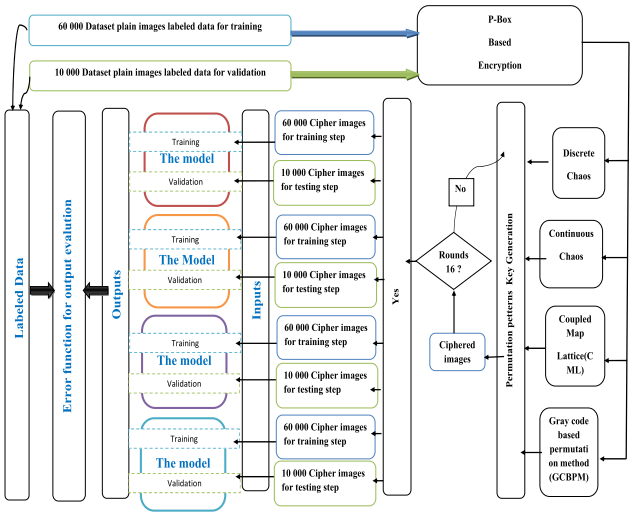


FIGURE 9. Data set generation and labeled data for 16 rounds experiments.

TABLE 9. Transfer learning experiments with one round P-box based encryption.

Patterns	Loss	R^2	MSE
Discrete chaos	0.1867	1	0.3265
Continuous chaos	2.4533	0.9782	175.4175
Gray code based permutation	0.0480	1	0.0725
Coupled map lattice	0.0014	1	0.0015

TABLE 10. Transfer learning experiments with 8 rounds P-box based encryption.

Patterns	Loss	R^2	MSE
Discrete chaos	0.0270	0.9993	5.8473
Continuous chaos	0.1995	0.9994	5.0487
Gray code based permutation	0.0769	0.9994	4.6592
Coupled map lattice	0.0530	1	0.1907

TABLE 11. Transfer learning experiments with 16 rounds P-box based encryption.

Patterns	Loss	R^2	MSE
Discrete chaos	0.0992	0.7599	1940.4117
Continuous chaos	8.3224	0.9223	623.6430
Gray code based permutation	0.5570	0.9994	4.9375
Coupled map lattice	0.0714	0.9991	7.6776

is the best proof of the concept distinguishability emphasized in this research.

5) PREDICTION

The prediction was accomplished by combining the search results; in other words, with models trained by the Mnist data set, the estimation was done by the Fashion Mnist data set. in which both models are predicted with the same weight and bias parameters.

As an illustration of our results, the figures below reflect a sample of the best results obtained by the model of the CML-encrypted images from the two data sets and the prediction of the corresponding images as an example of our results. Visually, we can also see a distinction in the distribution of colors in the encrypted images (Fig10, Fig12), like those of their counterparts dictated by the predicted model of simple images (Fig11, Fig13).

Besides, the images (Fig 14, Fig15) represent the prediction of the encrypted images of the worst results of the chaotic discrete models with 16 iterations, where it is very clear visually the degradation of image quality in prediction.

VI. DISCUSSIONS

A. MEASURING THE DECRYPTOR RESULTS

To examine and fully understand the research results, we need to have a measuring tool that quantifies the visual original plaintext from the Mnist and Fashion mnist data sets to the prediction results, allowing us to demonstrate the attack using a more reliable technique.

In order to monitor the effectiveness of our achievements, we deployed a pre-trained deep learning model with acceptable accuracy to distinguish and examine the outcomes. The model has pre-trained using Adam optimizer and Sparse Categorical Cross entropy error function.This model has a good level of accuracy in recognizing Mnist and Fashion Mnist data

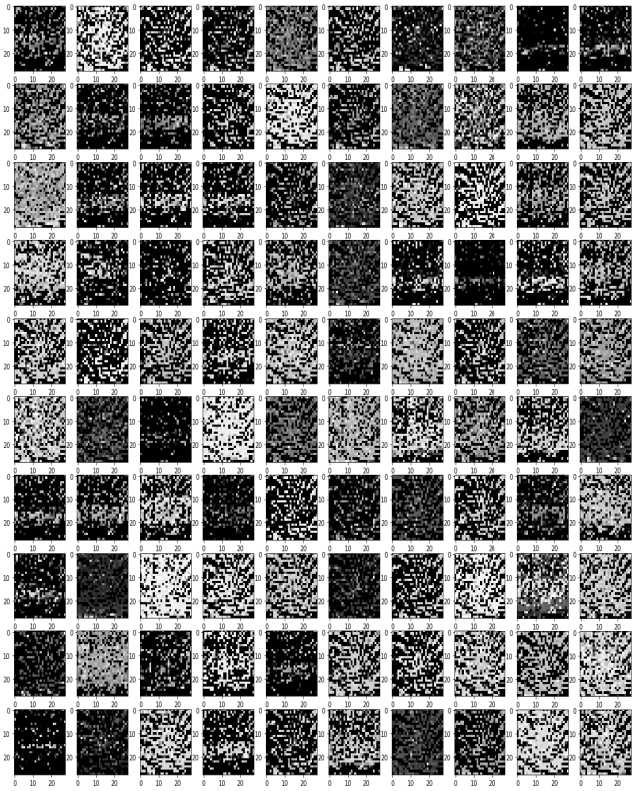


FIGURE 10. Fashion Mnist ciphered images with one round CML.

sets. It has a precision of (98.05 %) for Fashion Mnist and (99.00 %) percent for Mnist, and its design is fairly simple. It is also suitable for implementation as an experimental investigation. Figure 16 represents the architecture of this pre-trained model.

We examined the model's prediction performance on the original MNIST and Fashion MNIST test sets first and then used it to monitor the effectiveness of our predicted encrypted images. Figure 17 and 18 demonstrate the visual results of the quantitative prediction analysis.

B. QUANTITATIVE RESULTS AND COMPARISON

The first factor we saw was that when the number of rounds increased, so the effectiveness of the deep learning attack decreased. But this degradation is relative to several parameters and it differs from one permutation alternative to another.

Discrete chaos permutation patterns, for example, are more strong to attacks than continuous chaotic, and coupled map lattice is more secure and robust than the Gray code based permutation technique.

As a result, we find that discrete chaos is more resistant to our attack when the number of rounds increases, followed by some little resistance from continuous chaos; the scientific interpretation of the resistance is the discrete generation of permutation patterns, which makes the attack more difficult by more efficiently destroying the correlation between the swappable atoms.

TABLE 12. Related literature works Comparison.

Literature related works	Attack type	Uniform Distribution	KPA (Cipher/Plain) Pairs	Computational complexity	Number of Rounds	Recovered information	Contribution
Shujun Li at al[43]	Black box	Required	$O(n(MN)^2)$	$O(Log_L(MN))$	It has not been addressed	50 % of the key	General quantitative concept
Chengqing Li at al [44]	Black box	Required	$O(32(MN))$ and $O(16.n_0.(MN))$	$O(Log_L(MN).MN)$ and $O(MN)$	It has not been addressed	More than 50 % of the Key	Optimal quantitative concept
Alireza Jolfaei at al [46]	Black box	Required	$Log_L(MN)$	$O(n(MN))$	It has not been addressed	100 % of the key	Recovering completely the key
Leo Yu Zhang at al [54]	Black box	Required	$Log_L(MN)$	$O(n(MN))$	It has not been addressed	100 % of the key	Concept of composite representation
Our work	Black box	Not Required	More pairs for more best results	Depends on the trained model and parameters	Taken into consideration	Plain image	Concept of non uniform distribution

FIGURE 11. Fashion Mnist corresponding predicted images of one round CML.

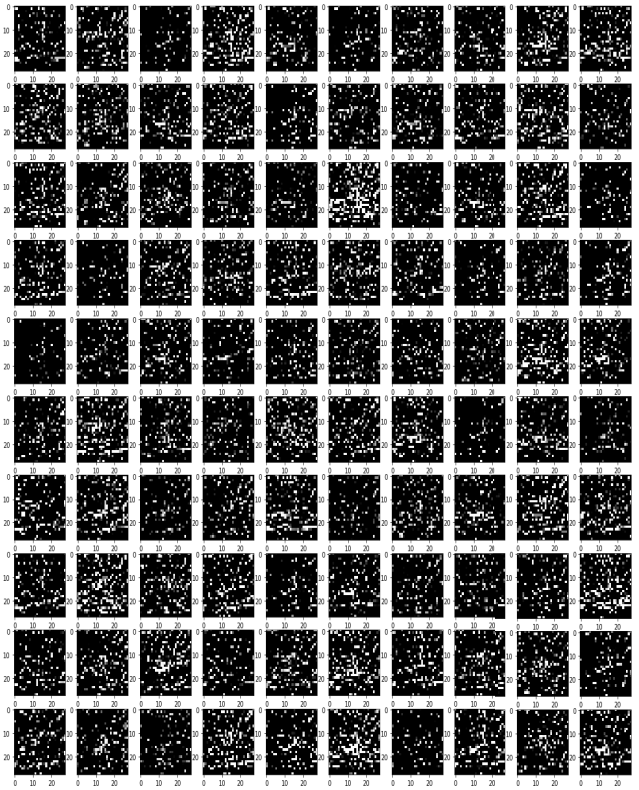


FIGURE 12. Mnist ciphered images with one round CML.

Despite a large number of rounds, CML and GCBPM are very weak in this test.

What should also be mentioned is another very important parameter, which is the redundancy of atoms (pixels) in the plain text with a non-uniform distribution. Despite the non-uniform distribution of colors in the Mnit and fashion mnist databases, the attack findings vary dramatically; it inherently comes down to the number of duplicated atoms, which is highly visible intuitively.

It should also be highlighted that our work is sensitive to the kind of data. For example, if we use an image with all of the intensities and a total number of pixels equal to zero or any other value between zero and 255, the model cannot learn anything as well as the loss function, which is represented by the limitation of differential cryptanalysis in the case of zero difference.

C. RELATED LITERATURE WORKS COMPARISON

Shujun Li at al[43] established the quantitative cryptanalysis concept for recovering the permutation key for use in the

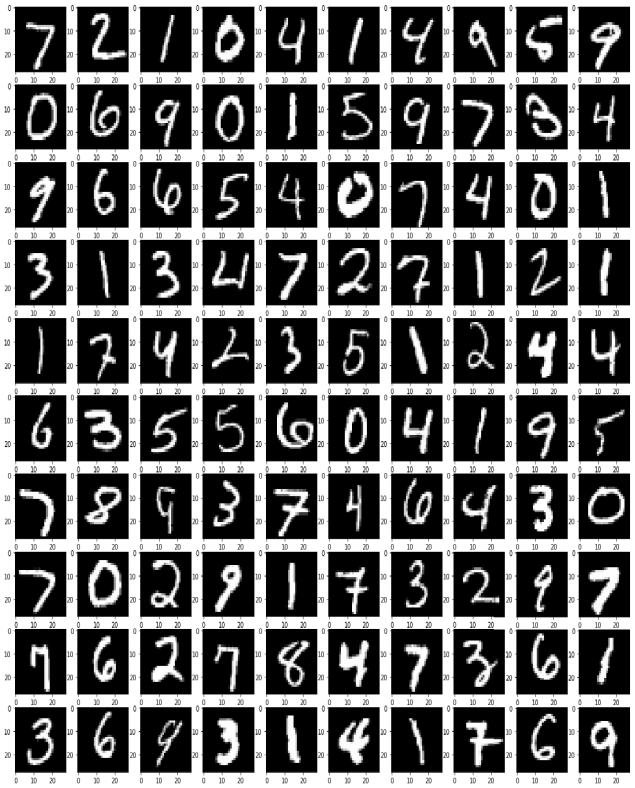


FIGURE 13. Mnist corresponding predicted images of one round CML.

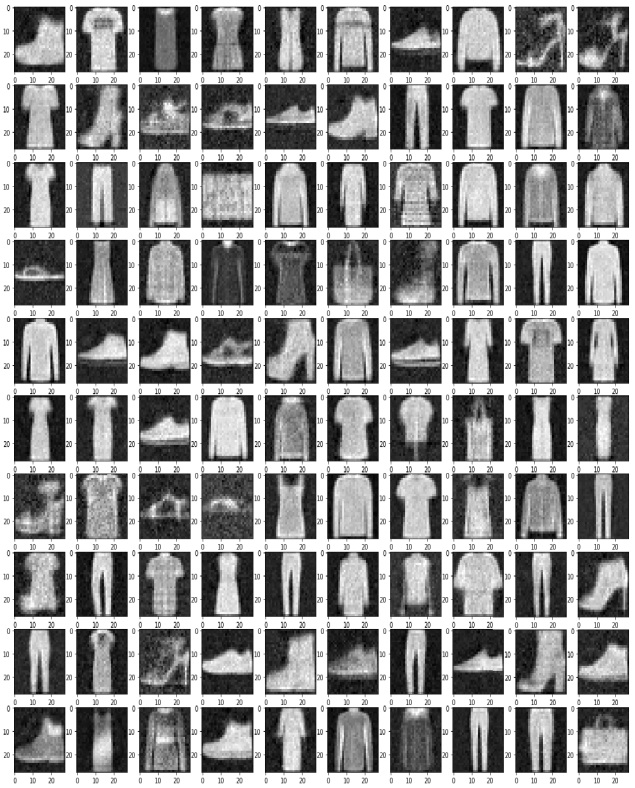


FIGURE 14. Mnist corresponding predicted images of 16 rounds Discrete chaos.

encryption of images encrypted only by permutation. This approach to expressing his efforts is based on the black box attack employing KPA pairs, which is primarily based on the uniform distribution of colors in the pairs involved. This attack was successful in retrieving 50% of the permutation key that was tested. However, their findings are entirely based on considerable computational and storage resources. Chengqing Li at al[44], inspired by the work of [43], optimized the findings of [43] by reducing the number of KPA pairs required to work as well as the computational and spatial complexity. Their approach can be expanded to perform any attack as a permutation cipher only with KPA pairs that provide uniform distribution; he effectively enhanced the attack and retrieved more than half of the permutation key used in the test. Alireza Jolfaei *et al.* [46] improved previous research and showed that the right permutation mapping is totally retrieved by a KPA attack and selected in all permutation-only image ciphers, independent of cipher construction. He is also reduced the number of KPA pairs required to work as well as the computational and complexity. Leo Yu Zhang *et al.* [54] build on previous studies by addressing how to balance storage cost and computational complexity while performing the KPA strategy. focuses on these two issues. He also bridged the assessed KPA gap between artificial noise-like images that fully match the theoretical model and the equivalent real images using a novel idea of the composite representation. Despite their relevance, all of these efforts have one important weakness: the uniform distribution of colors in KPA pairs. Among the restrictions, we may mention that all of these studies are based on the black box attack with classical research approaches and various optimization methods, and none of these works discussed the scenario of applying many rounds of permutation. Because of the unavailability of uniform distribution in real data, their reuse in a process of evaluating the permutation approach remains a concern in real-world practical scenarios. Table 12 provides a comparison between all of these works. In our contribution, we examined the problem from a different perspective than that taken by the previous research. As the main viewpoint, we considered the absence of uniform color distribution as a focus point. We also discussed this topic in terms of the number of rounds and key generation strategies. We consider that our technique has the first advantage of being easily reusable. This technique can be used to test the strength of image encryption algorithms during the deployment phase or to select the best permutation strategy during the development phase.

For further illustration, designers can use the same dataset used in our study to produce images encrypted by different algorithms that are under test evaluation and then train our model on these generated data with different rounds, algorithm parameters, and configurations in order to select the best permutation algorithm to use, as well as its best choice of the optimal algorithms' parameters and the adequate number of rounds.

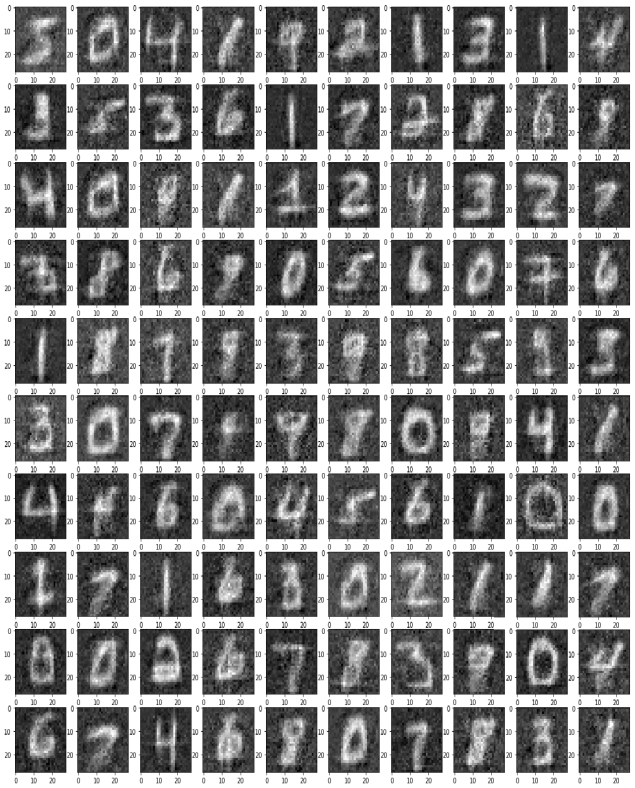


FIGURE 15. Mnist corresponding predicted images of 16 rounds Discrete chaos.

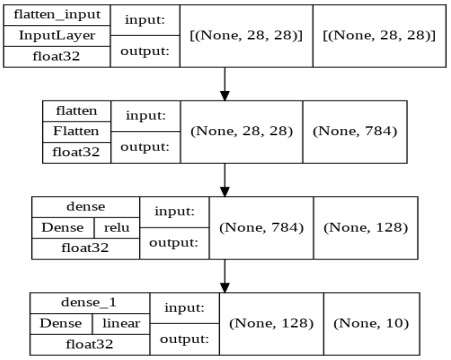


FIGURE 16. The measure model architecture.

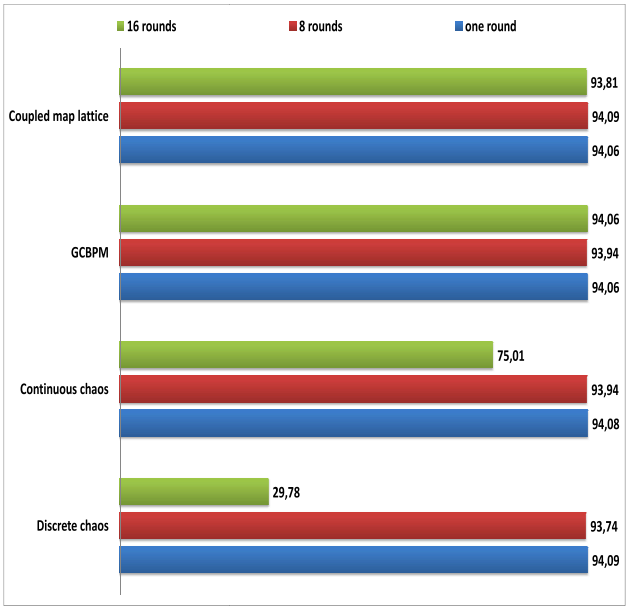


FIGURE 17. Quantitative results for Mnist dataset.

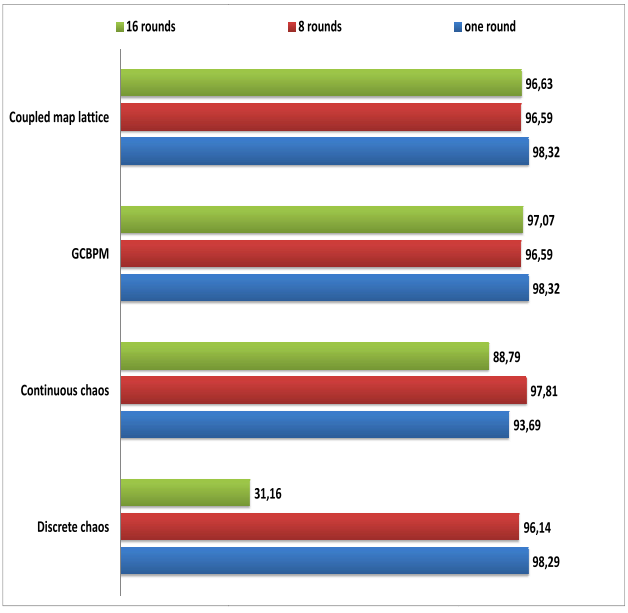


FIGURE 18. Quantitative results for fashion mnist dataset.

It can also aid in decision-making by encrypting images with multiple rounds and then testing the results provided by the models to customize the number of rounds and compare many methods.

The notion of reuse refers to two aspects. The first is to provide the model as we have illustrated in this article with the same datasets, model architecture, and parameters to test the desired permutation algorithm. The second aspect is to use the encryption algorithm to be evaluated to generate encrypted images of the size of 28×28 from other datasets and reuse the same model architecture. However, if designers want to test ciphers in a larger space with large images, we recommend focusing on our convolution and deconvolution strategies, which are used to build deep learning models with different more suitable layers. to the needs of the designer, as long as the hyper-parameters are preserved.

Despite that, our approach necessitates preprocessing dataset procedures and a large number of computational resources, as well as computation time and a vast number of experiments in the search for the desired model, highlighting the technique's limitations.

VII. CONCLUSION

In this research, our findings provide an innovative methodology for leveraging deep learning to identify decryptors

on symmetric permutation primitives. Our approaches are applicable to any number of (non-zero) input variations. At its heart, we adopt frequent dissimilarities to solve the challenge of discriminating in two-dimensional space.

The presented research is intended to be used separately from the operational mode of cryptography implementations. It should be used a prior, such that, during the design stage of cipher architecture, it can be used to examine the strongest permutation mechanism to be used.

Otherwise, it can be implemented to assess and compare different permutation patterns algorithms with a scientific hypothesis. However, the time required to calculate those assessments is a significant factor influencing their utility.

We do not claim that deep-learning tools will eventually replace classical cryptanalysis. However, we believe that our findings demonstrate that deep learning models are able to be trained to do cryptanalysis at a level that is attractive to cryptographers and that deep learning approaches can be a helpful addition to the arsenal of cryptographic assessors. The interpretability of deep learning-based black box attacks is still a problem. The fact that a neural model is a black box tells us very little concerning the real weakness of the studied cryptosystem. This opens the doors for the possibility of future studies to better answer this question.

In future research, we will look at deep learning-based cryptanalysis for video and sound encryption as well as other multimedia encryption systems. We will also attempt to solve and optimize conventional classical cryptanalysis challenges using artificial intelligence techniques and tools.

REFERENCES

[1] M. F. Idris, J. S. Teh, J. L. S. Yan, and W.-Z. Yeoh, "A deep learning approach for active S-box prediction of lightweight generalized feistel block ciphers," *IEEE Access*, vol. 9, pp. 104205–104216, 2021.

[2] E. Amirhossein, R. Francesco, and P. Paolo, "Reducing the cost of machine learning differential attacks using bit selection and aPartial ML-Distinguisher," School Comput. Sci. IT, Cryptol. ePrint Arch., Cork, Irland, Tech. Rep. 1479, 2021.

[3] R. L. Rivest, "Cryptography and machine learning," in *Advances in Cryptology—ASIACRYPT*. Berlin, Germany: Springer, 1993, pp. 427–439.

[4] A. Gohr, "Improving attacks on round-reduced Speck32/64 using deep learning," in *Advances in Cryptology*. Cham, Switzerland: Springer, 2019, pp. 150–179.

[5] B. Hou, Y. Li, H. Zhao, and B. Wu, "Linear attack on round-reduced DES using deep learning," in *Computer Security—ESORICS*. Cham, Switzerland: Springer, 2020, pp. 131–145.

[6] B. Zahednejad and J. Li, "An improved integral distinguisher scheme based on neural networks," Inst. Artif. Intell. Blockchain, Guangzhou, China, Tech. Rep. 4735, 2020.

[7] G. Liu, J. Lu, H. Li, P. Tang, and W. Qiu, "Preimage attacks against lightweight scheme Xoodyak based on deep learning," in *Advances in Information and Communication*, Vancouver, BC, Canada. Cham, Switzerland: Springer, 2021, pp. 637–648.

[8] A. Benamira, D. Gerault, T. Peyrin, and Q. Q. Tan, "A deeper look at machine learning-based cryptanalysis," in *Advances in Cryptology* (Lecture Notes Comput. Science). Cham, Switzerland: Springer, 2021, pp. 805–835.

[9] A. Jain and G. Mishra, "Analysis of lightweight block cipher few on the basis of neural network," in *Harmony Search and Nature Inspired Optimization Algorithms*. Singapore: Springer, Aug. 2018, pp. 1041–1047.

[10] R. Focardi and F. L. Luccio, "Neural cryptanalysis of classical ciphers," in *Proc. Italian Conf. Theor. Comput. Sci.*, Urbino, Italy, 2018, pp. 104–115.

[11] A. Gomez, S. Huang, I. Zhang, B. Li, M. Osama, and L. Kaiser, "Unsupervised cipher cracking using discrete GANs," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.

[12] C. Tan and Q. Ji, "An approach to identifying cryptographic algorithm from ciphertext," in *Proc. 8th IEEE Int. Conf. Commun. Softw. Netw. (ICCSN)*, Beijing, China, Jun. 2016, pp. 19–23.

[13] W. Zhang, Y. Zhao, and S. Fan, "Cryptosystem identification scheme based on ASCII code statistics," *Secur. Commun. Netw.*, vol. 2020, pp. 1–10, Dec. 2020.

[14] A. Baksi, J. Breier, Y. Chen, and X. Dong, "Machine learning assisted differential distinguishers for lightweight ciphers (extended version)," Cryptol. ePrint Arch., Nanyang Technol. Univ., Singapore, Tech. Rep. 571, Dec. 2020.

[15] J. So, "Deep learning-based cryptanalysis of lightweight block ciphers," *Secur. Commun. Netw.*, vol. 2020, pp. 1–11, Jul. 2020.

[16] G. Mishra, S. V. S. S. N. V. G. K. Murthy, and S. K. Pal, "Neural network based analysis of lightweight block cipher present," in *Harmony Search and Nature Inspired Optimization Algorithms*. Singapore: Springer, Aug. 2018, pp. 969–978.

[17] E. M. Rogers, "Claude Shannon's cryptography research during world war II and the mathematical theory of communication," in *Proc. IEEE Int. Carnahan Conf. Security Technol. (CCST)*, Albuquerque, NM, USA, Oct. 1994, pp. 1–5.

[18] S. K. Abd-El-Hafiz, S. H. Abdelhaleem, and A. G. Radwan, "Novel permutation measures for image encryption algorithms," *Opt. Lasers Eng.*, vol. 85, pp. 72–83, Oct. 2016.

[19] A.-V. Diaconu, "Circular inter–intra pixels bit-level permutation and chaos-based image encryption," *Inf. Sci.*, vols. 355–356, pp. 314–327, Aug. 2016.

[20] C. Fu, B.-B. Lin, Y.-S. Miao, X. Liu, and J.-J. Chen, "A novel chaos-based bit-level permutation scheme for digital image encryption," *Opt. Commun.*, vol. 284, no. 23, pp. 5415–5423, 2011.

[21] G. Ye, "Image scrambling encryption algorithm of pixel bit based on chaos map," *Pattern Recognit. Lett.*, vol. 31, no. 5, pp. 347–354, 2010.

[22] Z.-L. Zhu, W. Zhang, K.-W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Inf. Sci.*, vol. 181, no. 6, pp. 1171–1186, 2011.

[23] N. Bourbakis and A. Dollas, "SCAN-based compression-encryption-hiding for video on demand," *IEEE Multimedia*, vol. 10, no. 3, pp. 79–87, Jul. 2003.

[24] M. Zanin and A. N. Pisarchik, "Gray code permutation algorithm for high-dimensional data encryption," *Inf. Sci.*, vol. 270, no. 20, pp. 288–297, 2014.

[25] X. Zhang, "Lossy compression and iterative reconstruction for encrypted image," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 53–58, Mar. 2011.

[26] F. Huang, J. Huang, and Y.-Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2777–2789, Dec. 2016.

[27] S. M. M. Rahman, M. A. Hossain, H. Mouftah, A. E. Saddik, and E. Okamoto, "Chaos-cryptography based privacy preservation technique for video surveillance," *Multimedia Syst.*, vol. 18, no. 2, pp. 145–155, 2012.

[28] F. Dufaux and T. Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1168–1174, Aug. 2008.

[29] C. Wang, H.-B. Yu, and M. Zheng, "A DCT-based MPEG-2 transparent scrambling algorithm," *IEEE Trans. Consum. Electron.*, vol. 49, no. 4, pp. 1208–1213, Nov. 2003.

[30] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 118–129, Mar. 2003.

[31] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Trans. Signal Process.*, vol. 48, no. 8, pp. 2439–2451, Aug. 2000.

[32] H. Sohn, W. De Neve, and Y. M. Ro, "Privacy protection in video surveillance systems: Analysis of subband-adaptive scrambling in JPEG XR," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 2, pp. 170–177, Feb. 2011.

[33] J. Zhou, X. Liu, O. C. Au, and Y. Y. Tang, "Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 1, pp. 39–50, Nov. 2014.

[34] Q. Zhang, L. Liu, and X. Wei, "Improved algorithm for image encryption based on DNA encoding and multi-chaotic maps," *Int. J. Electron. Commun.*, vol. 68, no. 3, pp. 186–192, Mar. 2014.
[35] A. G. Radwan, S. H. AbdElHaleem, and S. K. Abd-El-Hafiz, "Symmetric encryption algorithms using chaotic and non-chaotic generators: A review," *J. Adv. Res.*, vol. 7, no. 2, pp. 193–208, 2016.
[36] Q. Zhang and X. Wei, "A novel couple images encryption algorithm based on DNA subsequence operation and chaotic system," *Optik*, vol. 124, no. 23, pp. 6276–6281, Dec. 2013.
[37] X. Wang and L. Yang, "A novel chaotic image encryption algorithm based on water wave motion and water drop diffusion models," *Opt. Commun.*, vol. 285, no. 20, pp. 4033–4042, Sep. 2012.
[38] M. Bertilsson, E. F. Brickell, and I. Ingemarsson, "Cryptanalysis of video encryption based on space-filling curves," in *Advances in Cryptology—EUROCRYPT*, Houthalen, Belgium. Berlin, Germany: Springer, 1989, pp. 403–411.
[39] R. F. Graf, *More Scrambling & Descrambling Techniques, Video Scrambling & Descrambling: For Satellite & Cable TV*, 2nd ed. London, U.K.: Newnes, 1999, ch. 5, pp. 39–43.
[40] Y. Matias and A. Shamir, "A video scrambling technique based on space filling curves (extended abstract)," in *Advances in Cryptology—CRYPTO*, Athens, GA, USA. Berlin, Germany: Springer, 1988, pp. 398–417.
[41] W. Li, Y. Yan, and N. Yu, "Breaking row-column shuffle based image cipher," in *Proc. 20th ACM Int. Conf. Multimedia (MM)*, Nara, Japan, 2012, pp. 1097–1100.
[42] C. Li, D. Lin, and J. Lü, "Cryptanalyzing an image-scrambling encryption algorithm of pixel bits," *IEEE Multimedia*, vol. 24, no. 3, pp. 64–71, Mar. 2017.
[43] S. Li, C. Li, G. Chen, N. G. Bourbakis, and K.-T. Lo, "A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Image Commun.*, vol. 23, no. 3, pp. 212–223, 2008.
[44] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 235–246, Feb. 2016.
[45] X.-Y. Zhao, G. Chen, D. Zhang, X.-H. Wang, and G.-C. Dong, "Decryption of pure-position permutation algorithms," *J. Zhejiang Univ. Sci.*, vol. 5, no. 7, pp. 803–809, 2004.
[46] C. Li and K.-T. Lo, "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Process.*, vol. 91, no. 4, pp. 949–954, 2011.
[47] T. Bianchi, A. Piva, and M. Barni, "Efficient linear filtering of encrypted signals via composite representation," in *Proc. 16th Int. Conf. Digit. Signal Process.*, Santorini, Greece, Jul. 2009, pp. 1–6.
[48] T. Bianchi, T. Veugen, A. Piva, and M. Barni, "Processing in the encrypted domain using a composite signal representation: Pros and cons," in *Proc. 1st IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, London, U.K., Dec. 2009, pp. 176–180.
[49] Z. Hua and Y. Zhou, "Image encryption using 2D logistic-adjusted-sine map," *Inf. Sci.*, vol. 339, pp. 237–253, Apr. 2016.
[50] T. Stutz and A. Uhl, "A survey of H.264 AVC/SVC encryption," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 325–339, Mar. 2012.
[51] L. Y. Zhang, X. Hu, Y. Liu, K.-W. Wong, and J. Gan, "A chaotic image encryption scheme owning temp-value feedback," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 10, pp. 3653–3659, Oct. 2014.
[52] Q. Zhang, H. Zhong, L. T. Yang, Z. Chen, and F. Bu, "PPHOCFS: Privacy preserving high-order CFS algorithm on the cloud for clustering multimedia data," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 4s, pp. 1–15, Nov. 2016.
[53] L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai, "On the security analysis of an image scrambling encryption of pixel bit and its improved scheme based on self-correlation encryption," *Commun. Nonlinear Sci. Numer. Simulat.*, vol. 17, no. 8, pp. 3303–3327, 2012.
[54] L. Y. Zhang, Y. Liu, C. Wang, J. Zhou, Y. Zhang, and G. Chen, "Improved known-plaintext attack to permutation-only multimedia ciphers," *Inf. Sci.*, vols. 430–431, pp. 228–239, Mar. 2018.
[55] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[56] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[57] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, CA, USA, 2015, pp. 1–15.
[59] M. Ishikawa, "Structural learning with forgetting," *Neural Netw.*, vol. 9, no. 3, pp. 509–521, Apr. 1996.
[60] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 281–305, Feb. 2012.
[61] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward Neural Networks," in *Proc. AISTATS*, Sardinia, Italy, 2010, pp. 1–8.
[62] X.-Y. Wang, Y.-Q. Zhang, and X.-M. Bao, "A novel chaotic image encryption scheme using DNA sequence operations," *Opt. Lasers Eng.*, vol. 73, pp. 53–61, Oct. 2015.
[63] J.-X. Chen, Z.-L. Zhu, C. Fu, H. Yu, and L. Zhang, "An efficient image encryption scheme using Gray code based permutation approach," *Opt. Lasers Eng.*, vol. 67, pp. 191–204, Apr. 2015.
[64] T. R. Lee, J. S. Teh, N. Jamil, J. L. S. Yan, and J. Chen, "Lightweight block cipher security evaluation based on machine learning classifiers and active S-Boxes," *IEEE Access*, vol. 9, pp. 134052–134064, 2021.

ZAKARIA TOLBA was born in Oum El Bouaghi, Algeria. He received the B.Eng. degree in parallel and distributed systems and the M.Sc. degree in distributed software architectures from Oum El Bouaghi University, Algeria, in 2010 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Mathematics, Larbi Tebessi University, Algeria, in the areas of machine learning applications in cryptography, security, privacy, and cryptanalysis.

From 2011 to 2015, he worked as an Engineer and a System Administrator in the military sector. Since 2017, he has been the Head of the Center for Information and Communication Systems, Networks, Distance Education, and Videoconferencing, Om El Bouaghi University.

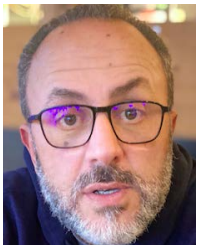
MAKHLOUF DERDOUR received the Engineering degree in computer sciences from the University of Constantine, Algeria, in 2004, the Magister degree in computer sciences from the University of Tebessa, and the Ph.D. degree in computer networks from the University of Pau and Pays de l'Adour (UPPA), France, in 2012. He is currently a Full Professor at the Computer Science Department, University of Oum El Bouaghi, Algeria. His research interests include software architecture, multimedia applications, adaptation and self-adaptation of applications, design and modeling of systems, and systems security. He is a General Chair of the International Conference on Pattern Recognition and Intelligent Systems (PAIS).

MOHAMED AMINE FERRAG (Senior Member, IEEE) received the bachelor's, master's, Ph.D., and Habilitation degrees in computer science from Badji Mokhtar—Annaba University, Annaba, Algeria, in June 2008, June 2010, June 2014, and April 2019, respectively.

Since October 2014, he has been a Senior Lecturer with the Department of Computer Science, Guelma University, Guelma, Algeria. Since July 2019, he has been a Visiting Senior Researcher with the NAU-Lincoln Joint Research Center of Intelligent Engineering, Nanjing Agricultural University, Nanjing, China. His current H-index is 24, i10-index is 42, and 3388 citations in Google Scholar Citation. His research interests include wireless network security, network coding security, and applied cryptography. He has published over 90 papers in international journals and conferences in the above areas. He has been conducting several research projects with international collaborations on these topics.

He is featured in Stanford University's list of the world's Top 2 % scientists for the years 2019 and 2020. Some of his research findings are published in top-cited journals, such as the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, IEEE ACCESS, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, *Sensors* (MDPI), *Journal of Information Security and Applications* (Elsevier), *Transactions on Emerging Telecommunications Technologies* (Wiley), *Telecommunication Systems* (Springer), *International Journal of Communication Systems* (Wiley), *Sustainable Cities and Society* (Elsevier), and *Journal of Network and Computer Applications* (Elsevier). He is also serving on various editorial positions, such as an Editorial Board Member in journals (indexed SCI and Scopus), such as *ICT Express* (JCR IF 4.317), *IET Networks* (Citescore 4.1), *International Journal of Internet Technology and Secured Transactions* (Citescore 1.0), *Security and Communication Networks* (JCR IF 1.791), and *Journal of Sensor and Actuator Networks* (Citescore 6.2). He reviewed more than 1160 articles (verified by publons) for top-cited journals, including *Nature*, IEEE TRANSACTIONS, Elsevier, Springer, and Wiley journals. He was a recipient of the 2021 IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT Best Paper Award.

S. M. MUYEEN (Senior Member, IEEE) received the B.Sc.Eng. degree in electrical and electronic engineering from the Rajshahi University of Engineering and Technology (RUET), Bangladesh, formerly known as the Rajshahi Institute of Technology, in 2000, and the M.Eng. and Ph.D. degrees in electrical and electronic engineering from the Kitami Institute of Technology, Japan, in 2005 and 2008, respectively. He is currently working as a Full Professor with the Electrical Engineering Department, Qatar University. He has been a keynote speaker and an invited speaker at many international conferences, workshops, and universities. He has published more than 250 papers in different journals and international conferences. He has published seven books as an author or editor. His research interests include power system stability and control, electrical machine, FACTS, energy storage systems (ESS), renewable energy, and HVDC systems. He is a fellow of Engineers Australia. He is serving as an Editor/Associate Editor for many prestigious journals from IEEE, IET, and other publishers, including IEEE TRANSACTIONS ON ENERGY CONVERSION, IEEE POWER ENGINEERING LETTERS, *IET Renewable Power Generation*, and *IET Generation, Transmission & Distribution*.



MOHAMED BENBOUZID (Fellow, IEEE) received the B.Sc. degree in electrical engineering from the University of Batna, Batna, Algeria, in 1990, the M.Sc. and Ph.D. degrees in electrical and computer engineering from the National Polytechnic Institute of Grenoble, Grenoble, France, in 1991 and 1994, respectively, and the Habilitation à Diriger des Recherches degree from the University of Picardie "Jules Verne," Amiens, France, in 2000. After receiving the Ph.D. degree, he joined the Professional Institute of Amiens, University of Picardie "Jules Verne," where he was an Associate Professor of electrical and computer engineering. Since September 2004, he has been with the University of Brest, Brest, France, where he is currently a Full Professor of electrical engineering. He is also a Distinguished Professor and a 1000 Talent Expert at the Shanghai Maritime University, Shanghai, China. His main research interests include analysis, design, and control of electric machines, variable-speed drives for traction, propulsion, and renewable energy applications, and fault diagnosis of electric machines. He is a fellow of the IET. He is the Editor-in-Chief of the *International Journal on Energy Conversion* and the *Applied Sciences* (MDPI) Section on Electrical, Electronics and Communications Engineering. He is a Subject Editor of the *IET Renewable Power Generation*.