Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-18959-6

A fusion of machine learning and cryptography for fast data encryption through the encoding of high and moderate plaintext information blocks

Arslan Shafique[1] · Abid Mehmood[2] · Moatsum Alawida[2] · Mourad Elhadef[2] · Mujeeb Ur Rehman[3]

Received: 19 July 2022 / Revised: 26 February 2024 / Accepted: 13 March 2024

Abstract
Within the domain of image encryption, an intrinsic trade-off emerges between computational complexity and the integrity of data transmission security. Protecting digital images often requires extensive mathematical operations for robust security. However, this computational burden makes real-time applications unfeasible. The proposed research addresses this challenge by leveraging machine learning algorithms to optimize efficiency while maintaining high security. This methodology involves categorizing image pixel blocks into three classes: high-information, moderate-information, and low-information blocks using a support vector machine (SVM). Encryption is selectively applied to high and moderate information blocks, leaving low-information blocks untouched, significantly reducing computational time. To evaluate the proposed methodology, parameters like precision, recall, and F1-score are used for the machine learning component, and security is assessed using metrics like correlation, peak signal-to-noise ratio, mean square error, entropy, energy, and contrast. The results are exceptional, with accuracy, entropy, correlation, and energy values all at 97.4%, 7.9991, 0.0001, and 0.0153, respectively. Furthermore, this encryption scheme is highly efficient, completed in less than one second, as validated by a MATLAB tool. These findings emphasize the potential for efficient and secure image encryption, crucial for secure data transmission in rea-time applications.

Keyword Data security · Computational time · Machine learning · Internet of things

✉ Arslan Shafique
arslanshafique762@gmail.com

1 School of Biomedical Engineering, University of Glasgow, Glasgow, UK

2 Department of Computer Sciences, Abu Dhabi University, Abu Dhabi, United Arab Emirates

3 Cyber Technology Institute, School of Computer Science and Informatics, De Montfort University, Leicester, UK

Published online: 04 April 2024

1 Introduction

The increasing use of sensitive medical, military and defense images in the Internet of Things (IoT) has resulted in a substantial surge in the volume of data transmitted through the Internet infrastructure [1, 2]. Given that the Internet is inherently insecure, there is a heightened risk of intruders attempting to compromise such sensitive data, potentially containing confidential or classified information. Intruders may exploit vulnerabilities in unsecured communication methods, emphasizing the imperative need to safeguard this data against unauthorized access [3, 4]. Cryptography plays a pivotal role in achieving this protection, employing sophisticated mathematical algorithms to encrypt data prior to transmission [5, 6]. This becomes especially critical in the context of IoT, where securing the integrity and confidentiality of data is paramount across various sectors.

Encryption serves as a crucial layer of security for unencrypted data by transforming the transmitted information into a seemingly random or unintelligible form to potential adversaries [7]. Notable encryption algorithms widely used for this purpose include DES (Data Encryption Standard), AES (Advanced Encryption Standard), IDEA (International Data Encryption Algorithm), and RSA (Rivest, Shamir, and Adleman) [8–11]. While these algorithms demonstrate efficiency in terms of performance, they encounter computational challenges, particularly when dealing with large datasets. In the context of the IoT, where massive amounts of data are routinely transmitted and security is paramount, the efficiency and computational demands of encryption algorithms become integral considerations for ensuring effective and timely data protection.

Similarly, encryption techniques rooted in chaos and employing intricate mathematical structures may demand a substantial amount of computational time [12, 13]. While chaotic systems provide robust security for digital data [14], their inherent high time consumption becomes a critical concern in the context of the IoT, where real-time processing is paramount [15]. In IoT applications, where massive amounts of data are transmitted, the tradeoff between security and computational efficiency becomes particularly pronounced. The challenge lies in addressing this time-intensive characteristic of chaotic systems to ensure their practicality in real-time scenarios.

For digital images in IoT, which inherently involve large datasets, managing computational complexity becomes imperative. Plentiful redundant data, indicative of minimal or no pixel changes in plain text images, should be selectively excluded during the encryption process to optimize processing time [16]. This optimization is crucial for IoT devices operating in resource-constrained environments where computational efficiency is key. Figure 1 illustrates a plaintext image with multiple regions (Area, B Area, and C Area). Notably, encrypting areas with minimal pixel changes, like Area 'A', becomes unnecessary and contributes to an increased overall encryption computational time for the encryption scheme. The intersection of chaotic systems, cryptographic techniques, and the unique demands of IoT underscores the need for tailored solutions that strike a balance between security and computational efficiency, ensuring the seamless integration of encryption strategies into real-time IoT applications.

Reducing the time complexity of encryption algorithms is crucial to their applicability in real-time applications, and leveraging machine learning (ML) is an effective strategy to achieve this goal [17]. The intersection of machine learning and cryptography is particularly significant within the landscape of the IoT, where real-time processing is a fundamental requirement [18]. Both ML and cryptography share the need for processing large amounts of data, and as the volume of generated data exceeds ten billion bytes daily, the integration of ML approaches into cryptography becomes increasingly crucial within the IoT framework [19].

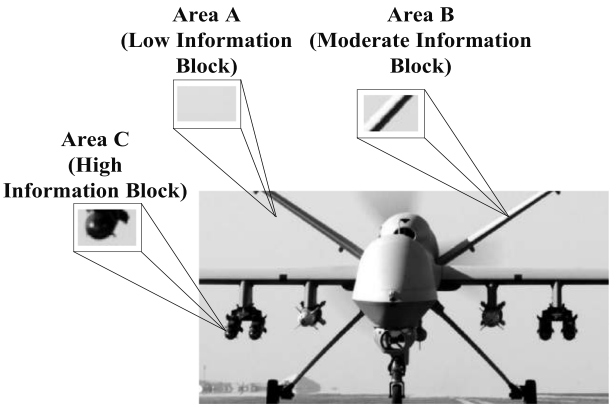


Fig. 1 Pixel change in different areas (A,B and C)

In the context of IoT, where devices continuously generate and exchange massive datasets, machine learning's ability to automate the development of analytical models becomes indispensable [20]. By continually adapting and learning from substantial data input, ML-driven encryption solutions have the potential to enhance the efficiency and security of real-time IoT applications. This symbiotic relationship between ML, cryptography, and the evolving demands of IoT underscores the necessity for advanced and adaptive encryption strategies tailored for the dynamic and data-intensive nature of IoT environments.

The design of encryption algorithms based on chaotic maps with the integration of ML techniques may be more beneficial. Moreover, protecting only relevant information, referred to as an area of interest (ROI), using techniques such as bit-plane extraction may reduce the computational cost significantly. To preserve the highest level of security with minimum computational cost, our contributions to this work are as follows:

- In this research, a new approach to image encryption that addresses the trade-off between computational complexity and robust data security is proposed. By leveraging machine learning algorithms such as support vector machines (SVM), the methodology enhances encryption computational time efficiency.
- A selective encryption strategy leveraging machine learning algorithms is presented. By encrypting only high and moderate information blocks, the proposed methodology significantly reduces computational time, making it suitable for real-time applications.
- A detailed mathematical model is presented to illustrate the bit-plane extraction methodology from plaintext images. This model demonstrates how the application of the bit-plane encryption technique can reduce the computational cost without compromising the security of the encrypted data.
- To gauge the effectiveness of the proposed methodology, two sets of parameters are considered. The first group includes F1-score, accuracy, recall, and precision, which refer to machine learning metrics. The second group involves correlation, energy, contrast, and entropy. Such metrics are associated with cryptographic aspects. Additionally, the efficiency of the proposed approach is demonstrated through a comparative analysis between the proposed work and existing techniques.

These contributions are important for enhancing the overall security of digital data. Simultaneously, they show the significance of ensuring the suitability of the proposed model for real-time applications. In a landscape where fast data processing is required, the proposed

enhancements not only enhance data security but also show the efficiency of the proposed model in meeting the demands of real-time scenarios.

2 Related work

To secure digital data, a number of researchers have used various encryption approaches, such as chaos-based encryption, bit-plane extraction-based technology, and machine learning-based encryption [21–24]. Due to its exceptional properties, such as sensitivity to initial conditions and the ability to generate random numbers, chaos has been shown to be a suitable option for encrypting digital images.

In [25], Lu et al. proposed a chaotic S-box-based image encryption scheme in which a method provided in [26] is analyzed and cryptanalysis is done on the image encryption scheme. Several significant contributions are made in light of the results of the cryptanalysis, including the development of (a) a new chaotic map; (b) a substitution box-(S-box); and (c) an improved image encryption scheme. Although only a single S-box has been employed by the authors in the presented scheme, it compromises the security of the overall encryption scheme. A single S-box was used in an encryption system, which caused two main vulnerabilities, one relating to security strength and one relating to computational complexity. Because the information from the plaintext image can be visualized in the encrypted image, a single S-box is insufficient to adequately conceal an image pixel in the encrypted image. Furthermore, for a 256×256 image, the single S-box takes nearly 15 seconds to substitute the pixel values with the specific S-box values. A detailed explanation of the drawbacks of using a single S-box is given in [27]. Moreover, a solution to the limitations of using a single S-box is also provided in [27], in which multiple S-boxes are included and merged with the chaotic logistic map [28]. Although the authors have successfully proposed a solution and produced better statistical results as compared to the single S-box, the information can still be visualized in the enciphered images which corresponds to the improper concealment of the image pixels. Afterward, Ahmed et al. [29] process the idea proposed in [27] and improve the methodology to encrypt the digital images using multiple S-boxes. The information in the generated enciphered images was properly concealed, but the computational complexity was a bit too high to be acceptable. To overcome the computational time issue, Shafique et al. [30] proposed an encryption scheme in which discrete wavelet transform (DWT) is considered. The purpose of using DWT was to reduce the computational cost by encrypting only a specific frequency band, i.e., low-frequency band (LL-subband), which contains most of the plaintext information.

In [31], Leng et al. explored the advancements in cancelable palmprint coding frameworks to reduce computational complexity and storage costs. The extension of one-dimensional frameworks to two dimensions is introduced, accompanied by the implementation of measures such as perpendicular orientation transposition and multi-orientation score level fusion. The research extensively compares the performance of PalmHash Code and PalmPhasor Code. In [32], Leng et al. enhanced accuracy by extracting features from left and right palmprints using two-dimensional discrete cosine transform (2DDCT) within a dual-source space. Normalization in this space mitigates disturbances caused by coefficients with large absolute values, eliminating the need for complex pre-masking. The utilization of discrimination power analysis (DPA) further improves accuracy by preserving more discriminative coefficients. In [33], Leng et al. proposed the Conjugate 2DPalmHash Code (CTDPHC) as a cancelable multi-modal biometric, constructed from 2DPalmHash Codes (2DPHCs)

of palmprint and palmvein. The study compares and discusses different fusion rules at the score level, fine-tunes transposition orientation ranges of 2DPHCs, and demonstrates that CTDPHC achieves higher verification accuracy and stronger anti-counterfeit ability without increasing computational complexity or storage costs compared to 2DPHC.

In [34], Pourasad et al. devised a technique for encrypting digital images using random sequences created using chaos theory. Additionally, DWT is used to transform the pixel values into the frequency domain. Li et al. [35] used an imitating jigsaw method to propose a chaos based image encryption scheme (PEP). Three distinct processes are performed on a digital image: pre-processing, encryption, and finally, post-processing. All of these procedures are carried out sequentially, resulting in a significant computational cost. To further enhance the security of his suggested encryption technique, he incorporates the diffusion operation. According to the statistical data mentioned in his report, PEP is not suited for real-time applications. Fractional-order chaos and neural network-based image encryption is proposed in [36]. The diffusion is performed using substitution boxes which can lead to high computational complexity. Moreover, according to Shannon's criteria of diffusion and confusion, the lack of any cryptosystem consists of only permutation or diffusion can be non-resists against several cyberattacks such as entropy attack, ciphertext-only attack, and chosen-plaintext attack [37–40]. Lin et al. [41] proposed a modified version of AES is based on chaos theory, in which the concept of random key generation is also introduced. Kumar et al. [42] used the zig-zag scan method to secure digital images. For shuffling purposes, which corresponds to confusion, one-dimensional chaos for the generation of random sequences. Additionally, a substitution operation that corresponds to the diffusion is performed using a zig-zag methodology. Although the authors have successfully satisfied the criteria of Shannon's theory, due to the implementation of such operations in sequential order, it may take a bit more time, which makes the encryption algorithm unsuitable for real-time applications.

The literature reveals that the majority of existing schemes make a trade-off between security and computational complexity. In light of the vulnerabilities discussed in this section, a time-efficient image encryption is proposed in this paper while preserving a suitable level of security. The proposed work employs machine learning, chaos theory, and a bit-plane extraction method.

The rest of the paper is structured as follows: Section 3 contains the preliminaries, which describe bit-plane extraction and the chaotic map utilized in the proposed work. The proposed method and its experimental validation are discussed in detail in Section 5 and Section 6 respectively. While Section 7 concludes the research work.

3 Preliminaries

For the security of the digital images, the following components are used to reduce the computational complexity and manipulate the image pixels in the proposed work:

1. SVM
2. Modified pulse-coupled spiking neurons circuit-based chaotic map (MPSNC)
3. Bit-plane extraction

3.1 Support Vector Machine (SVM)

The SVM is a classifier that classifies the data points into specific classes and finds an ideal hyperplane with the maximum margin [43]. The proposed machine learning-based model

classifies the data samples into two classes, i.e., whether the block of a plaintext image should be encrypted or not. For classification purposes, The data points may be separated using the (1).

$$P = X * A + c \tag{1}$$

Where P and X represent the hyperplane and the angle of the hyperplane in N-dimensional space respectively. While input and the threshold are represented by A and c respectively. The threshold value shows the distance of the hyperplane from the origin. Equations (2) and (3) define the canonical hyperplane for positive and negative class samples, respectively. The positive class corresponds to the plaintext block that should be passed through the encryption algorithm, while the negative class corresponds to the block of original pixels that should not be encrypted using the proposed encryption algorithm.

$$P_{+} = X * A + c = +1 \tag{2}$$

$$P_{-} = X * A + c = -1 \tag{3}$$

The data points that fall above (2) are classified as positive, while those that are below (3) are classified as negative. Support vectors are the data points that lie on (2) and (3). The shape of the hyperplane and the location of the support vectors on it have an impact on the design of the SVM classifier. Figure 2 illustrates the SVM structure in detail.

The purpose of SVM is to maximize the margin, which is accomplished by decreasing the factor w as mentioned in (4).

$$margin = \frac{2}{||X||} \tag{4}$$

3.2 Chaotic map used in the proposed work

Modified pulsed-coupled spiking neurons circuit (MPSNC) is proposed by Zhang et. al. [44] and it is used in the proposed encryption scheme. MPSNC is selected due to its degree of security strength and advantages such as large key space, sensitivity to initial conditions, and permutation entropy. MPSNC is selected as the encryption approach for the proposed scheme owing to its high level of security and several benefits, including a wide key space, sensitivity to initial conditions, and permutation entropy. These benefits are discussed in further depth

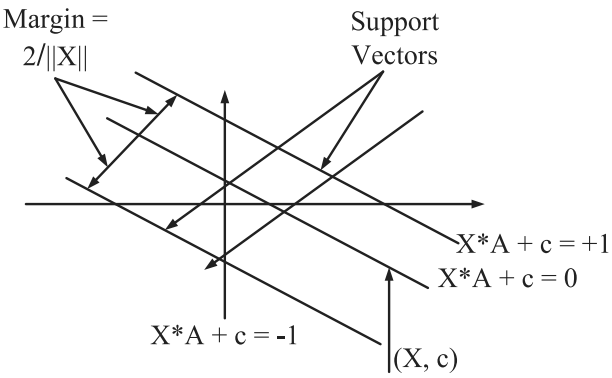


Fig. 2 SVM structure

in the subsections. The MPSNC is controlled by six unique sensitive variables and has a large key space that assists in brute force attack resistance. All such benefits demonstrate the effectiveness of MPSNC in the image encryption domain. The selected chaotic map is defined in (5), (6) and (7).

$$y(j+1) = G(y(j)) = \begin{cases} G_1(y(j)) \; mod \; \mathrm{M} & \text{for odd j} \\ G_2(y(j)) \; mod \; \mathrm{M} & \text{for even j} \end{cases} \tag{5}$$

$$G_1(y(j)) = \begin{cases} \frac{-L_{12} sin(2\pi s y(j)) + L_{11}(y_n(j) - \frac{1}{4})}{K} + y(j), \\ \quad 0 \le y_n(j) < \frac{1}{2} \\ \frac{-L_{12} sin(2\pi s y(j)) - L_{11}(y_n(j) - \frac{1}{4})}{K} + y(j), \\ \quad \frac{1}{2} \le y_n(j) < 1 \end{cases} \tag{6}$$

$$G_2(y(j)) = \begin{cases} \frac{-L_{22} sin(2\pi s y(j)) + L_{21}(y_n(j) - \frac{1}{4})}{K} + y(j), \\ \quad 0 \le y_n(j) < \frac{1}{2} \\ \frac{-L_{22} sin(2\pi s y(j)) - L_{21}(y_n(j) - \frac{1}{4})}{K} + y(j), \\ \quad \frac{1}{2} \le y_n(j) < 1 \end{cases} \tag{7}$$

Where M is the lowest denominate of the fraction form of s, $y \in [0, M)$, $y_n(j) = y(j) \mathrm{mod} 1$, and $\frac{|L_{11}|}{4} + L_{12} < 1$, $\frac{|L_{21}|}{4} + |L_{22}| < 1$.

With appropriate control parameters and initial conditions, the chaotic behavior of MPSNC may be evaluated using the relationship between the parameters given in [44]. MPSNC is composed of six independent parameters denoted by the letters L_{11}, L_{12}, L_{21}, L_{22}, s and K. In this section, the effectiveness of the MSSNC is evaluated using the sensitivity, permutation entropy, Lyapunov, and bifurcation diagrams by setting the constant values, i.e.,. L_{11}=1.6, L_{12} = 1.9, L_{21}=2.3, L_{22}=0.3, s=0.1, and K=0.2. While it is possible to access a large and flexible range of key spaces, only two parameters, such as L_{12} and $y(1)$ are considered to concisely describe the proposed technique. The detail analysis to show the strength of MSSNC are given in [44].

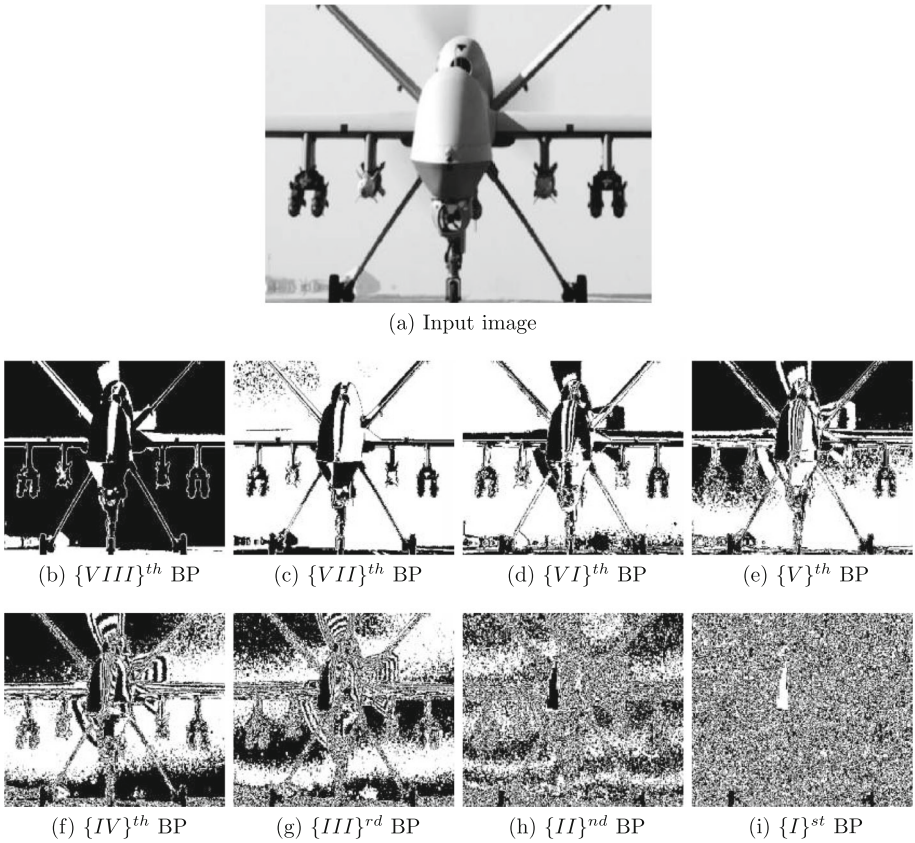
3.3 Bit-plane extraction

The bit-plane (BP) extraction technique is used to extract bit planes from plaintext images. Depending on the number of bits in a single pixel of an image, various plaintext images may have a varied number of bit-planes [45]. For instance, an eight-bit plaintext image is composed of eight eight-bit planes. A two-bit image, likewise, includes two bit-planes. In the proposed work, 8-bit grayscale images are used. Figure 3 illustrates the plaintext image and its corresponding eight bit-planes ($\{VIII\}^{th}$ BP, $\{VII\}^{th}$ BP, …, $\{I\}^{st}$ BP).

As seen in Fig. 3, the highest bit-plane ($8th - BP$) holds the bulk of the information in the plaintext image, while the lowest bit-plane ($1st - BP$) contains the least information in the plaintext image. In the proposed work, only those BPs with a sufficient amount of information in the plaintext image are considered for encryption. The amount of information of input image in each BP can be calculated using (8) [22].

$$BP_m = \frac{2^{m-1}}{\sum_{m=0}^{7} 2^{m-1}} \tag{8}$$

The parentage amount of data included in each BP can be calculated using (8) and is displayed in Table 1. As shown in Table 1, the four most important bit planes (BP_8, BP_7,



(a) Input image

(b) $\{VIII\}^{th}$ BP (c) $\{VII\}^{th}$ BP (d) $\{VI\}^{th}$ BP (e) $\{V\}^{th}$ BP

(f) $\{IV\}^{th}$ BP (g) $\{III\}^{rd}$ BP (h) $\{II\}^{nd}$ BP (i) $\{I\}^{st}$ BP

Fig. 3 Extracted BPs form input image

BP_6, BP_5) contain more than 90% of plaintext information. Therefore, it is unnecessary to encrypt the least significant bit planes(BP_4, BP_3, BP_2, BP_1), as this may increase the computing time required for encryption.

Table 1 Amount of information in individual BP (%)

Position of BP_m	Amount present in each BP
BP_1	0.30
BP_2	0.79
BP_3	1.42
BP_4	3.12
BP_5	6.25
BP_6	12.23
BP_7	25.7
BP_8	50.20

4 Overview of the proposed IMLC

The proposed research is divided into two key modules: (a) machine learning and (b) encryption module. The machine learning modules are based on SVM while the proposed encryption module consists of a few primary components, including bit-plane extraction and a chaotic map, as seen in Fig. 4.

The plaintext image is divided into blocks to allow for the implementation of machine learning algorithms on each block. The objective of applying machine learning to each block of plaintext image is to reduce the computational complexity of the proposed encryption procedure. This is accomplished by disregarding block pixels that carry only a tiny amount of information. The information threshold is determined by extracting statistical and texture data from each pixel block. After extracting the features, the proposed machine learning module identifies which blocks should be encrypted.

The data set for the machine learning module is constructed by assigning relevant feature values to each feature. Additionally, several machine learning algorithms, such as decision tree (DT), naive Bayes (NB), K-Nearest Neighbour (K-NN), random forest (RF), and SVM are evaluated in order to choose the optimal option for the proposed work.

The proposed encryption module starts by extracting bit-planes (BPs) from each pixel block. The extraction of bit-planes from each block is intended to further minimize computational complexity. Only the first four bit-planes (BP_8, BP_7, BP_6, and BP_5) are encrypted, since they contain over 95% of the plaintext image as shown in Table 1. After extracting BPs, a chaotic map is incorporated to create confusion and diffusion in the plaintext blocks. Confusion and diffusion are both necessary to meet Shannon's criterion [37]. According to him, any encryption technique that incorporates these two operations can be considered a secure encryption scheme. To create confusion, a chaotic map is used to generate various random sequences. The plaintext blocks' pixels are scrambled based on the random sequences generated using a chaotic map and given the term "pre-ciphered blocks". Additionally, for the diffusion process, a Random Image (RI) of the same size as the plaintext image is created. To construct the final encrypted picture, this RI is XORed with the pre-ciphered blocks. The details of each step of the proposed encryption algorithm are given in the Section 5.

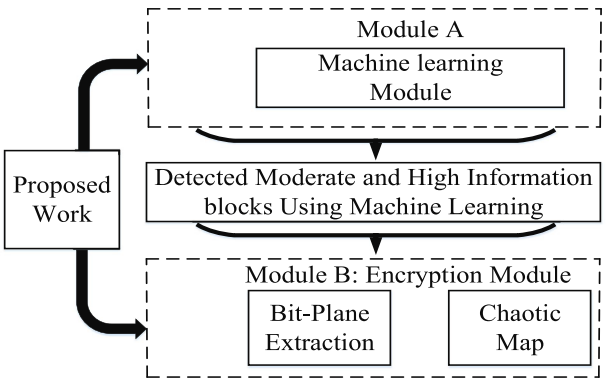


Fig. 4 Generalized block diagram of the proposed research

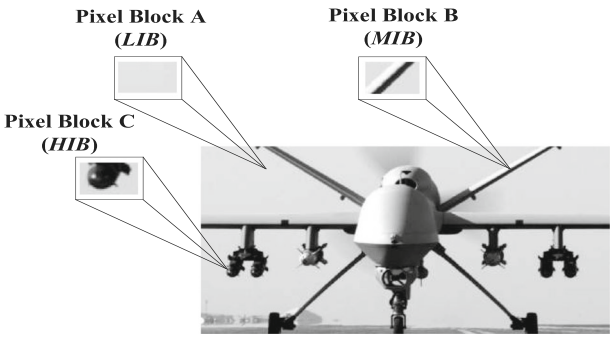


Fig. 5 Visualization of information blocks (HIB, MIB, and LIB)

5 Proposed encryption scheme

Machine learning, bit-plane extraction, and chaos theory are utilized in the proposed work. The main aim of the proposed work is to develop such an encryption algorithm which should be time efficient and provide a suitable level of security to the digital images. Before applying the proposed machine learning and encrypt modules on the plaintext image, the original image is divided into blocks of size 8×8 as follows:

5.1 Machine learning module

The ML module is applied to the image pixel blocks generated by the algorithm ref algo1. The purpose of using machine learning is to categorize pixel blocks based on the information they contain. Image pixel blocks are classified as high information blocks (HIB), moderate information blocks (MIB), and low information blocks (LIB). Based on the intervals listed in Table 2, these three classifications are formed. Figure 5 provides an illustration of the visualization of such a block. The LIB and MIB blocks contain little and considerable information about the plaintext image, respectively. Whereas, the HIB block contains the high information content of the plaintext image.

Algorithm 1 Divide image into blocks.

```
Start
Input Plaintext image (PI)
Calculate size of PI: [row col] = size (PI)
Define size of a blockR
    for r = 1:row
        for c = 1:col
Divide image with the required number of pixels: PI/64
        end
    end
End
```

Several machine learning algorithms, including SVM, RF, DT, K-NN, and NB, are implemented in the suggested work in order to choose the optimal alternative. The dataset used in this research is created by incorporating security parameters as features The details of the features extracted from each block are given in Table 2.

Table 2 Intervals define corresponding to each feature

Parameters	Mathematical equations	Relationship with strong security (S.S)	Intervals for amount of information
Entropy	$Ent = -\sum O(p_i) log_2 en(c_i)\ O(p_i)$	Entropy $\propto$S.S	[7.9999 7.9000] $\rightarrow LIB$
	is probability of occurrence		[7.2930 6.4490] $\rightarrow MIB$
			[6.4230 6.0010] $\rightarrow HIB$
Energy	$Energy = \sum O(a,b)^2$	Energy $\propto \frac{1}{S.S}$	[0.0100 0.0150] $\rightarrow LIB$
	O(a,b) is an original image		[0.0151 0.0201] $\rightarrow MIB$
			[0.0201 0.0349] $\rightarrow HIB$
Correlation	$Co = \frac{\frac{1}{L}\sum_{j=1}^{L}(x_i - En(a))(y_i - En(b))}{\sigma_a \sigma_b}$	Correlation $\propto \frac{1}{S.S}$	[-0.0012 0.0308]$\rightarrow LIB$
	$\sigma_a = \sqrt{VARa}, \sigma_b = \sqrt{VAR_b}$		[0.0001 0.0011] $\rightarrow MIB$
	$VAR(a) = \frac{1}{L}\sum_{j=1}^{L}(a_i - E(a))^2$		[0.0000 0.4500] $\rightarrow HIB$
	$VAR(b) = \frac{1}{L}\sum_{j=1}^{L}(b_i - E(b))^2$		
	L: Total pixels,		
	E(a) and E(b) is Encrypted		
	image in horizontal		
	and vertical direction		
Contrast	$Cont = \sum \lvert a-b \rvert^2 O(a,b)\Big]$	Contrast $\propto$S.S	[10.5000 9.5000] $\rightarrow LIB$
	O(a,b) is gray-level		[09.2500 8.7500] $\rightarrow MIB$
	co-occurrence matrices		[08.5000 7.5000] $\rightarrow HIB$
Homogeneity	$\sum_a \sum_b \frac{O(a,b)}{1+\lvert a-b \rvert}\Big]$	Homogeneity $\propto \frac{1}{S.S}$	[0.4122 0.4418]$\rightarrow LIB$
	O(a,b) is gray-level		[0.4521 0.4821] $\rightarrow MIB$
	co-occurrence matrices		[0.5367 0.6125] $\rightarrow HIB$

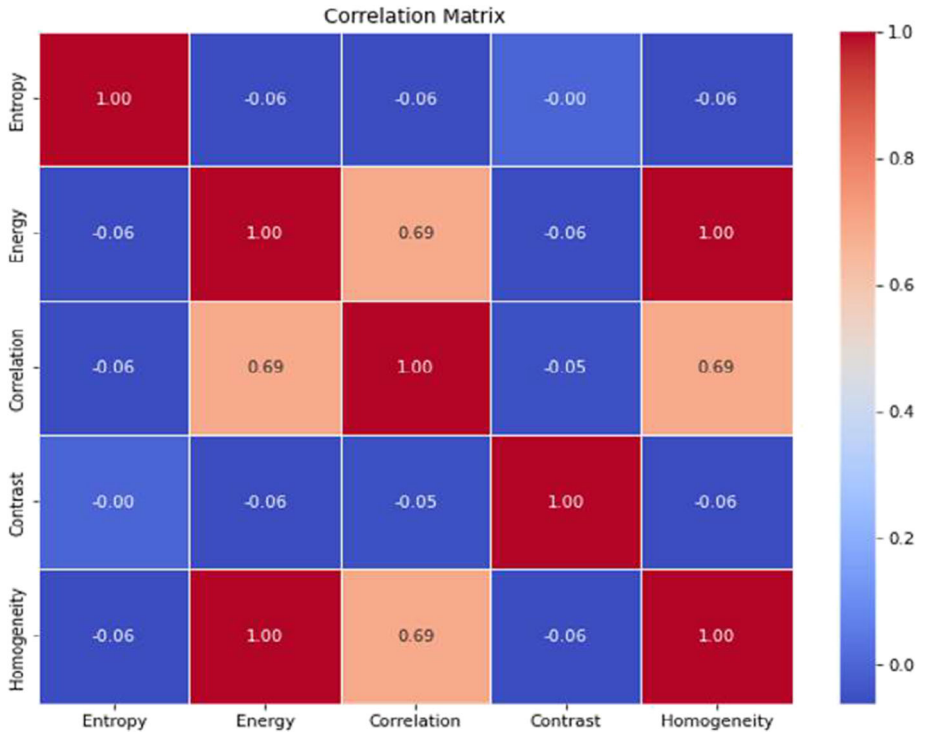


Fig. 6 Correlation heatmap of selected parameters

According to the feature intervals given in Table 2, if the pixel block's entropy value falls between $Ent \in [7.9997.9000]$ and $Ent \in [7.29306.4490]$, it will have a high (HIB) and moderate (MIB) level of information, respectively. Therefore, they should be encrypted to secure it from eavesdroppers. Similarly, if the entropy value of a pixel block falls within the range $Ent \in [7.2936.449]$, it indicates that the block has very low information LIB and can be transmitted without an encryption module to reduce the computing time required for encryption. The level of information present in each pixel block also depends on other features such as energy, correlation, contrast, and homogeneity. To identify any redundant features, a correlation analysis between such features is performed. In the analysis, a threshold value of 0.4 is set to evaluate the correlation between the parameters, namely entropy, energy, correlation, contrast, and homogeneity. The resulting correlation matrix can be visualized in the heatmap plot as shown in Fig. 6, which reveals that all the parameters exhibited correlation coefficients below the specified threshold. This shows that each parameter has its own independent significance. Moreover, the absence of strong correlations among the parameters reinforces the careful selection and non-redundant nature of the features considered in this research.

A generalized proposed algorithm, which is implemented in Python for the classification of each block, is given in Algorithm 2.

Algorithm 2 Machine learning module

```
Start
Input Pixel block
load dataset
Separate feature columns: x = dataset.iloc[:,1:5].values
drop labeled classes: y = dataset.iloc[:,5].values
import label encoder: from sklearn.preprocessing import LabelEncoder
    for r = 1:row
        for c = 1:col
            (labelencoder)_y = LabelEncoder()
            y = (labelencoder)_y.fit-transform(y)
        end
    end
Separate testing and training data:
        X_train, y_train = 80%
        X_test, y_test = 20%
Import classification algorithm (SVM): from sklearn.svm import SVC
Start data training
        svm = SVC(kernel = 'poly')
        svm.fit(x_train,y_train)
End of data training
prediction = svm.predict (x_test)
An upcoming block:
    if [Ent: 6.364, Energy: 0.0153, Co: 0.0237, Cont: 9.5692, Homogeneity: 0.4683]
            Block should be encrypted
    end
    if [Ent: 7.836, Energy: 0.0157, Co: 0.0009, Cont: 9.1635, Homogeneity: 0.4408]
            Block should be encrypted
    end
    if [Ent: 6.954, Energy: 0.0168, Co: 0.0337, Cont: 9.5692, Homogeneity: 0.3682]
            Encryption not necessary
    end
End
```

5.2 Encryption module

The block diagram of the proposed encryption technique is shown in the green dotted box of Fig. 7. In the first phase, a block of pixels is split into its eight bit-planes, and each bit-plane is rotated at several angles, such as 90^0, 180^0, and 270^0. The chaotic map is used in the second stage to generate six random sequences by selecting initial values for parameters such as L_{11}, L_{12}, L_{21}, L_{22}, s, and K. The confusion operation is then performed on the bit-planes that have been rotated. In the confusion operation, generated random sequences are employed to permute the bit-values of the bit-planes to make decryption more challenging for the eavesdroppers. In the final stage, combine all of the modified bit-planes into a single pixel block. Using the chaotic map, a two-dimensional noisy image is produced in order to execute the diffusion operation using XOR on the substituted images in order to get the final encrypted image.

In order to recover the original plaintext image, each stage of the encryption process must be performed in reverse order. The above mentioned encryption steps are described in detail in the following sections.

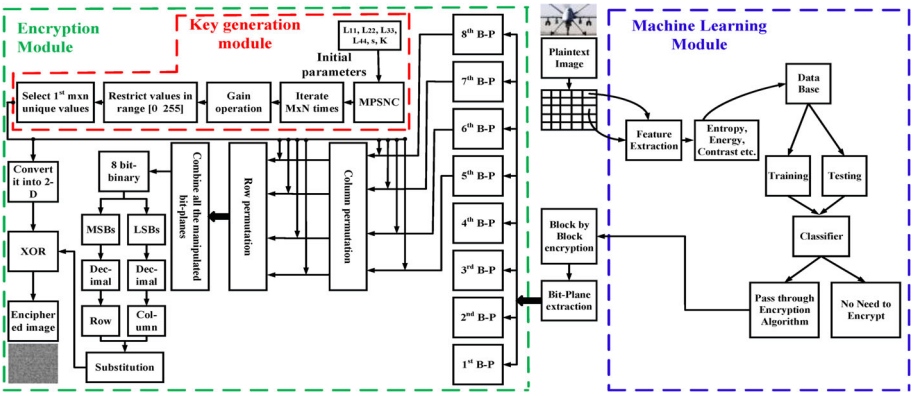


Fig. 7 Flow diagram of the proposed work

5.2.1 Bit-plane rotation

consider a block A of size 3×3 as en example. Let's assume that the block A belongs to the class HIB. Therefore it must be passed through the encryption process.

$$\text{Block A} = (HIP)_A = \begin{bmatrix} 105 & 156 & 202 \\ 253 & 84 & 95 \\ 116 & 39 & 48 \end{bmatrix}$$

Binary version of $(HIP)_A$ is:

$$\text{Binary version} = \begin{bmatrix} 01101001 & 10011100 & 11001010 \\ 11111101 & 01010100 & 01011111 \\ 01110100 & 00100111 & 00110000 \end{bmatrix}$$

Extract the binary bit-planes from the binary version of the HIP_A. Each bit of the pixel value makes its corresponding bit-plane, i.e., the $8th$ bit of each pixel value corresponds to the $8th$ bit-plane. Likewise, the $7th$ bit of each pixel value creates the $7th$ bit-plane, and so forth. All the bit-planes of $(HIP)_A$ are given below:

$$\{VIII\}^{th}BP = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \{VII\}^{th}BP = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \{VI\}^{th}BP = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \{V\}^{th}BP = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\{IV\}^{th}BP = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \{III\}^{rd}BP = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \{II\}^{nd}BP = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \{I\}^{st}BP = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Rotate each bit-plane at quadrangle angles as follows:

$$\{VIII\}^{th}BP(\text{Rotated at } 90^0) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \{VII\}^{th}BP(\text{Rotated at } 180^0) = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\{VI\}^{th}BP(\text{Rotated at } 270^0) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \{V\}^{th}BP(\text{Rotated at } 90^0) = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\{IV\}^{th}BP(\text{Rotated at } 180^0) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \{III\}^{rd}BP(\text{Rotated at } 270^0) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\{II\}^{nd}BP(\text{Rotated at } 90^0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \{I\}^{st}BP(\text{Rotated at } 180^0) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The rotated version of bit-planes is scrambled according to the random sequence generated using the chaotic map in the subsection.

5.2.2 Chaotic map used for key generation

The chaotic map is used in the proposed work for two main purposes; Generation of random sequence and noisy image. These can be generated using Algorithm 3

Algorithm 3 Random sequence and noisy image Key generation.

Start (Random sequence (R_{seq}))
Input Initial conditions and control parameters: L_{11}=1.6, L_{21}=2.3, L_{22}=0.3, s=0.1, and K=0.2.
for n = 1:10,000;
y(j+1) = G(y(j)) = $\begin{cases} G_1(y(j))\ mod\ \text{M} & \text{for odd j} \\ G_2(y(j))\ mod\ \text{M} & \text{for even j} \end{cases}$
Iterate (5)
Q(j) = (y(j+1))× 999; ▷ amplification of generated values
F(j) = floor(Q(j)) ▷ Convert fractional values into integers
Mod(j) = mod(F(j),256) ▷ Restrict values in range [0 255]
end
R = unique(Mod, 'stable') ▷ select first 256 unique values
End (R_{seq})
Start (Noisy image (N_{im}))
Repeat all the steps performed to generate random sequences (R)
N_{im} = reshape (R,256,256) ▷ Noisy image of size 256 × 256
End (N_{im})

Let the R_{seq} for the pixel permutation of $(HIP)_A$ is: $R_{seq} = \Big[R_1 = 3, R_2 = 1, R_3 = 2, R_4 = 4, R_5 = 7, R_6 = 5, R_7 = 8, R_8 = 6 \text{ and } R_9 = 9\Big]$. The individual values present in the rotated versions of bit-planes are converted into a row matrix as given below:

$\{\text{VIII}\}^{th}$ BP = [0 1 0 1 0 0 1 0 0], $\{\text{VII}\}^{th}$ BP = [1 0 1 1 1 1 1 0 0],$\{\text{VI}\}^{th}$ BP = [1 1 1 0 0 1 0 0 1], $\{\text{V}\}^{th}$ BP = [0 1 1 1 1 0 0 1 1], $\{\text{IV}\}^{th}$ BP = [1 1 1 1 0 1 0 0 0], $\{\text{III}\}^{rd}$ BP = [1 1 1 1 1 1 0 1 0], $\{\text{II}\}^{nd}$ BP = [0 0 0 0 0 1 1 1 0], $\{\text{I}\}^{st}$ BP = [1 0 0 1 0 1 0 1 0].

The permuted bit-planes ($P_{BP}s$) of row matrix bit-planes are: $P_{BP}^{VIII^{th}}$ = [0 0 1 1 1 0 0 0 0], $P_{BP}^{VII^{th}}$ = [1 1 0 1 1 1 0 1 0], $P_{BP}^{VI^{th}}$ = [1 1 1 0 0 0 0 1 1], $P_{BP}^{V^{th}}$ = [1 0 1 1 0 1 1 0 1], $P_{BP}^{IV^{th}}$ = [1 1 1 1 0 0 0 1 0], $P_{BP}^{III^{rd}}$ = [1 1 1 1 0 1 1 1 0], $P_{BP}^{II^{nd}}$ = [0 0 0 0 1 0 1 1 0], $P_{BP}^{I^{st}}$ = [0 1 0 1 0 0 1 1 0].

The planes $P_{BP}^{VIII^{th}}, P_{BP}^{VII^{th}}, \cdots, P_{BP}^{I^{st}}$ are known as manipulated bit-planes. The process of the combination of such bit-planes is given in Section 5.2.3.

5.2.3 Combination of manipulated bit-planes

Convert P_{BPS} into two dimensional matrices ($P_{BP}^{VIII^{th}}, P_{BP}^{VII^{th}}, \cdots, P_{BP}^{I^{st}}$) and combine them using (9) and save it in a variable called S_{box}. So, a substitution box (Sbox) can be applied to it to create diffusion.

$$P_{BP}^{VIII^{th}} = \begin{bmatrix} 0\ 0\ 1 \\ 1\ 1\ 0 \\ 0\ 0\ 0 \end{bmatrix}, P_{BP}^{VII^{th}} = \begin{bmatrix} 1\ 1\ 0 \\ 1\ 1\ 1 \\ 0\ 1\ 0 \end{bmatrix}, P_{BP}^{VI^{th}} = \begin{bmatrix} 1\ 1\ 1 \\ 0\ 0\ 0 \\ 0\ 1\ 1 \end{bmatrix}, P_{BP}^{V^{th}} = \begin{bmatrix} 1\ 0\ 1 \\ 1\ 0\ 1 \\ 1\ 0\ 1 \end{bmatrix}$$

$$P_{BP}^{IV^{th}} = \begin{bmatrix} 1\ 1\ 1 \\ 1\ 0\ 0 \\ 0\ 1\ 0 \end{bmatrix}, P_{BP}^{III^{rd}} = \begin{bmatrix} 1\ 1\ 1 \\ 1\ 0\ 1 \\ 1\ 1\ 0 \end{bmatrix}, P_{BP}^{II^{nd}} = \begin{bmatrix} 0\ 0\ 0 \\ 0\ 1\ 0 \\ 1\ 1\ 0 \end{bmatrix}, P_{BP}^{I^{st}} = \begin{bmatrix} 0\ 1\ 0 \\ 1\ 0\ 0 \\ 1\ 1\ 0 \end{bmatrix}$$

$$S_{box} = P_{BP}^{VIII^{th}} \times 2^7 + P_{BP}^{VII^{th}} \times 2^6 + P_{BP}^{VI^{th}} \times 2^5 + P_{BP}^{V^{th}} \times 2^4 + P_{BP}^{IV^{th}} \times 2^3 + P_{BP}^{III^{rd}} \times 2^2 + P_{BP}^{II^{nd}} \times 2^1 + P_{BP}^{I^{st}} \times 2^0 \tag{9}$$

$$S_{box} = \begin{bmatrix} 01111100\ 01101101\ 10111100 \\ 11011101\ 11000010\ 01010100 \\ 00010111\ 01101111\ 00110000 \end{bmatrix}, \rightarrow Sbox_{dec} = \begin{bmatrix} 124\ 109\ 188 \\ 221\ 194\ 84 \\ 23\ 111\ 48 \end{bmatrix}$$

$Sbox_{dec}$ is a pre-ciphered image pixel block that is fed into the substitution box in which the the pixel values of $Sbox_{dec}$ will be substituted with the S-boxes given in [46–48]

5.2.4 XOR operation and Substitution box algorithm

The decimal values in $Sbox_{dec}$ are substituted with the multiple S-boxes. The multiple S-boxes algorithm was proposed by Amir et al. in 2014 [27]. A few modifications in the substitution algorithm proposed in [27] are made that create more randomness in image pixels. Algorithm 4 shows the modified multiple S-boxes process.

After applying Algorithm 4, the resulted substitution image is:

$$\text{Substituted block}(S_{blk}) = \begin{bmatrix} 150\ 123\ 139 \\ 236\ 111\ 20 \\ 31\ 167\ 231 \end{bmatrix}$$

In the last, apply XOR operation on the substituted image with N_{im} ($CB = S_{blk} \bigoplus N_{im}$) to get the final ciphertext image block. Let assume the N_{im} is:

$$N_{im} = \begin{bmatrix} 132\ 101\ 196 \\ 36\ 131\ 69 \\ 91\ 138\ 155 \end{bmatrix}$$

$$CB = \begin{bmatrix} 150\ 123\ 139 \\ 236\ 111\ 20 \\ 31\ 167\ 231 \end{bmatrix} \bigoplus \begin{bmatrix} 132\ 101\ 196 \\ 36\ 131\ 69 \\ 91\ 138\ 155 \end{bmatrix} \rightarrow \begin{bmatrix} 18\ 30\ 79 \\ 200\ 136\ 81 \\ 68\ 45\ 124 \end{bmatrix}$$

Algorithm 4 Modified multiple S-boxes process.

Start
Input $Sbox_{dec}$
[row col] =size($Sbox_{dec}$)
$Sbox_{dec}$ = reshape($Sbox_{dec}$,1,row × col) ▷ Convert $Sbox_{dec}$ into row vector
for col = 1 : row × col
P_{scramb} (:, R_{seq}(col)) = $Sbox_{dec}$ (:,col) ▷ Image pixel Scrambling
end
Set counter: N = 1;
for i=1:row
for i=1:col
bin = dec2bin(P_{scramb}(i,j),8) ▷ Convert P_{scramb} into its 8 bit binary
LSBs = [bin(1) bin(2) bin(3) bin(4)] ▷ Extract Least Significant Bits (MSBs)
MSBs = [bin(5) bin(6) bin(7) bin(8)] ▷ Extract Most significant bits (MSBs)
LSBsrow = bin2dec(LSBs) ▷ Corresponds to the row number of S-box
MSBscol = bin2dec(MSBs) ▷ Corresponds to the column number of S-box
if R ==1
SB_{im}(i,j) = S_1(LSBsrow, MSBscol)
elseif R ==2
SB_{im}(i,j) = S_2(LSBsrow, MSBscol)
elseif R ==3
SB_{im}(i,j) = S_3(LSBsrow, MSBscol)
elseif R ==4
SB_{im}(i,j) = S_4(LSBsrow, MSBscol)
elseif R ==5
SB_{im}(i,j) = S_4(LSBsrow, MSBscol)
end
N = N+1
end
end
End

It can be seen that the pixel values in block $(\text{HIP})_A$ are completely different from the pixel values in the original block CB. This means that the plaintext block CB is properly concealed. Similarly, all pixel blocks (HIP and MIP) are encrypted using the suggested encryption scheme to produce a partial ciphertext image (PCI). In Fig. 8(a-f) and (g-l), a few plaintext images and their corresponding $PCIs$ are shown, respectively where it can be seen that all the meaningful information is completely encrypted. Moreover, the pixel blocks with the least amount of information are not changed.

To produce the Final Ciphertext Image (FCI), $anXORoperation$ (FCI = PCI $\bigoplus$ N_{im}) is used to break the correlation between the unmodified pixel blocks. Figure 8(m-r) depicts the $FCIs$, in which it can be observed that the correlation between all the pixel values in $FCIs$ is entirely broken.

6 Statistical results and analysis

This section evaluates the proposed machine learning model based on numerous parameters, including precision, recall, and F1-score. In addition, security metrics such as entropy, correlation, energy, contrast, histogram analysis, key space analysis, and key sensitivity analysis are used to assess the FCIs' security.

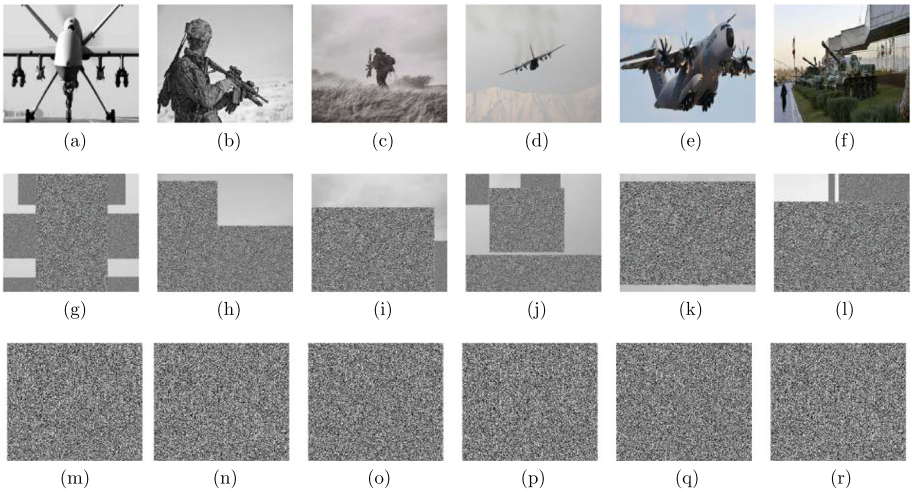


Fig. 8 Plaintext images and their corresponding $PCIs$ and $FCIs$: (a-f) Plaintext images, (g-l) $PCIs$ and (m-r) $FCIs$

6.1 Results and analysis of the proposed machine learning model

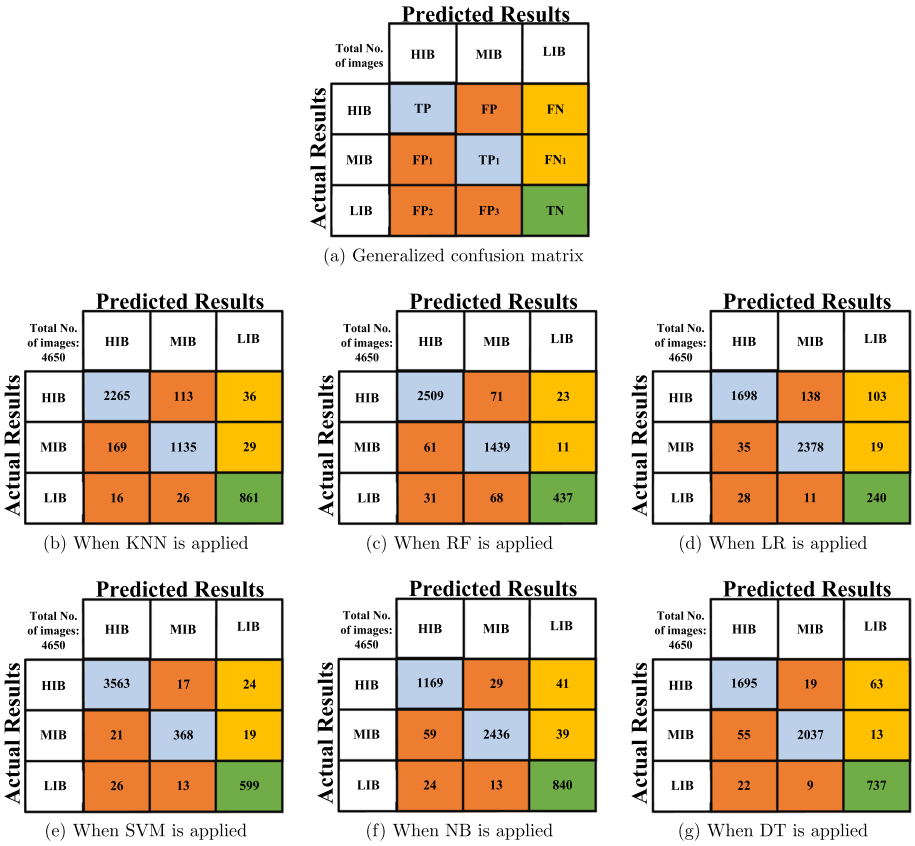
The proposed model is implemented in Python 3.7. For the experimental results and analysis of the proposed machine learning model, a computer with 8GB of RAM, an 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, and Windows 11 are used. To find out the statistics of the performance parameters to evaluate the machine learning model, a confusion matrix can be considered.

6.1.1 Confusion matrix

A confusion matrix is a two-dimensional (2-D) table that can be used to calculate the values of parameters such as F1-score, accuracy, precision, and recall. The generalized confusion matrix of the proposed work is given in Fig. 9(a). Whereas the statistical values of the parameters given in Fig. 9(a) are given in Fig. 9(b-g) when K-Nearest Neighbours (KNN), Random Forest (RF), Linear Regression (LR), SVM, Naive Byes (NB) and Decision Tree (DT) are incorporated respectively.

The terms TP, TN, FP, and FN given in Fig. 9(a) are defined below.

- TP: The image pixel block is HIB or MIB, and the predicted result is YES (pixel block should be encrypted).
- TN: The image pixel block is LIB, and the predicted result is NO (pixel block should not be encrypted)
- FP: The image pixel block is LIB, and the predicted result is YES (pixel block should be encrypted).
- FN: The image pixel block is HIB or MIB, and the predicted result is NO (pixel block should be encrypted).



(a) Generalized confusion matrix

(b) When KNN is applied

(c) When RF is applied

(d) When LR is applied

(e) When SVM is applied

(f) When NB is applied

(g) When DT is applied

Fig. 9 Generated confusion matrices when different ML algorithms are incorporated

The mathematical expressions of the parameters such as F1-score, accuracy, precision and recall are given in Table 3.

The statistical results for the proposed machine learning model including SVM, DT, RF, NB, and KNN are shown in Table 4. All values are expressed in percentages. In addition, a comparison with existing work is presented in Table 4 to demonstrate that the suggested work is superior than existing approaches.

Table 3 Performance measuring parameters

Parameters	Mathematical expressions
Accuracy	$\frac{TP+TP_1+TN}{TP+FP+FN+FP_1+TP_2+FN_1+FP_2+FP_3+TN}$
precision	$\frac{TP+TP_1}{TP+TP_1+FP+FP_1+FP_2+FP_3}$
Recall	$\frac{TP+TP_1}{TP+TP_1+FN+FN_1}$
F1-Score	$2 \times \left[\frac{Precision \times Recall}{Precision+Recall}\right]$

Table 4 Performance analysis of the proposed machine learning model

Models	KNN	RF	LR	SVM	NB	DT
Accuracy analysis						
[49]	94.3	96.1	95.5	93.7	91.6	90.9
[50]	92.6	89.9	90.1	92.4	89.9	91.6
[51]	84.6	86.4	87.1	89.6	87.8	88.8
Proposed	91.6	94.3	92.8	**97.4**	95.5	96.1
Precision analysis						
[49]	90.6	94.6	91.6	94.8	96.7	94.8
[50])	91.8	96.4	98.6	94.6	91.6	95.3
[51])	94.6	97.6	94.6	**97.8**	94.6	96.6
Proposed	91.2	94.4	95.0	97.8	96.6	97.2
Recall analysis						
[49]	94.6	95.6	94.3	91.8	94.6	97.8
[50]	94.6	94.6	97.6	91.6	94.6	89.6
[51]	94.6	95.8	97.6	91.6	97.6	94.6
Proposed	98.1	99.1	97.0	**98.9**	97.8	98.0
F-1 score analysis						
[49]	95.5	95.0	91.9	93.2	95.6	96.4
[50]	93.1	95.4	98.0	93.0	93.0	92.3
[51]	94.6	96.6	96.0	94.5	96.0	95.5
Proposed	94.5	96.6	95.9	**98.3**	97.1	97.5

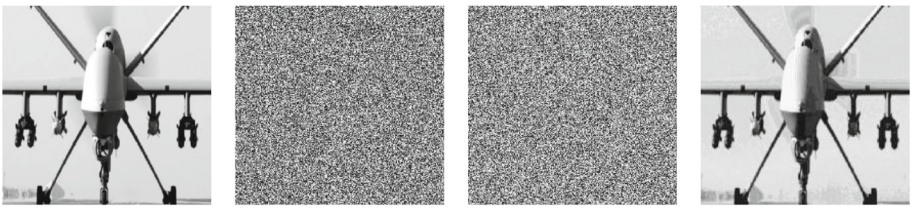
The bold entries highlight the best results

6.2 Results and analysis of the proposed encryption scheme

The proposed encryption scheme is based on the proposed machine learning model, which is employed to reduce the computing time required for encryption. In addition, a bit-plane extraction technique is used to further reduce processing time, making the proposed encryption system appropriate for real-time applications. Several statistical and security analyses, including histogram analysis, cropping and noise attack analysis, key sensitivity and key space analysis, correlation, entropy, contrast, energy mean square error, and peak signal-to-noise ratio, are performed to evaluate the performance of the proposed encryption scheme. The statistical values of such parameters are measured in the following three cases:

- **Case(a):** When only HIB and MIB are encrypted using the proposed encryption scheme.
- **Case(b)**: The entire plaintext image is encrypted.
- **Case(c):** When XOR operation is performed between PCI and N_{im} to generate FCI.

In case (a), the focus is only on encrypting HIBs and MIBs to enhance encryption computational efficiency and the security of blocks containing high- and moderate-information plaintext images. Conversely, in case (b), the entire plaintext image is encrypted, including all blocks such as HIBs, MIBs, and LIBs, using the entire proposed encryption process. Although this ensures robust security, it increases the computational time. However, in case (c), the LIBs are encrypted only using the XOR operation, while the HIBs and MIBs are encrypted through the proposed encryption process. This strategy results in a negligible increase in computational time while ensuring robust security.



(a) Original image (b) E_n (c) N_v (d) Decrypted image

Fig. 10 Noise attack analysis

6.2.1 Noise and cropping attack analysis

In a noise attack, the adversary often injects random noise into the ciphertext image [52]. In contrast, in cropping attack analysis, the attacker removes a portion of the ciphertext image in order to make the decryption failure [53]. The noisy version of the ciphertext image is generated using (10).

$$N_v = E_n \oplus Noise \qquad Noise \in \mathbb{Z} \tag{10}$$

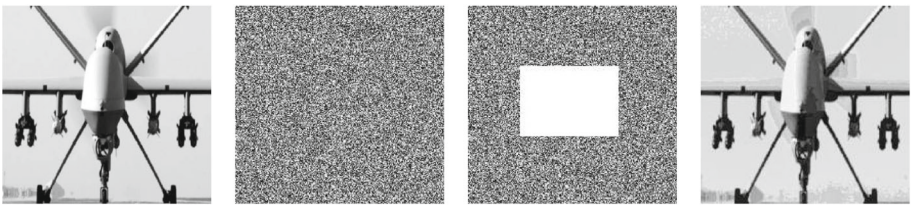
Where N_v is a noisy version of the ciphertext image (E_n) and $\oplus$ is the XOR operator.

By adding noise to the ciphertext image using the XOR operation, the resistivity against the noise attack of the proposed encryption scheme is evaluated. Figure 10 shows the noise attack analysis. Figure 10(b), (c), and (d) are the E_n, N_v, and the decrypted version form that is recovered from N_v, respectively. It can be seen that the decrypted image is not an exact replica of the plaintext image due to the presence of noise in E_n, but the information of the original image can be clearly seen, demonstrating that the proposed encryption scheme is resistant to noise attack.

To demonstrate resilience against cropping attacks, a portion of the ciphertext image is cropped, and the original image is decrypted from the cropped version of the ciphertext image. Figure 11(d) depicts the resultant decrypted image, in which the information contained in the plaintext image is clearly visible.

6.2.2 Computational time analysis

When creating an image encryption technique, there are often two significant factors to consider: (a) security strength and (b) computational complexity [54, 55]. The suggested encryption scheme is implemented on a system with the specifications listed in Section 6 in order to fulfill the criteria of robust encryption methods.



(a) Original image (b) E_n (c) Cropped image (d) Decrypted image

Fig. 11 Cropping attack analysis

Table 5 Computational time analysis (sec)

Encryption schemes	Cases	Aeroplane	Tank	Drone	Land
Proposed	Case(a)	**0.1352**	0.1364	0.1410	0.1366
	Case(b)	0.5130	0.5690	0.5333	0.6103
	Case(c)	0.1353	0.1365	0.1413	0.1368
Existing encryption	[34]	2.2665	2.2663	2.9432	2.8479
schemes	[35]	4.3365	4.1960	4.3786	4.3499
	[36]	3.9365	3.9948	4.1131	4.0990
	[42]	6.3541	6.5697	6.1598	6.3799
	[43]	7.1368	7.9846	7.1198	8.0019

The bold entry highlights the best result

The comparison of computational times for cases (a), (b), and (c) is shown in Table 5, where it can be observed that the computational time needed for case 'a' is somewhat less than the time necessary to encrypt the plaintext image in case 'c'. In addition, a comparison of the proposed encryption scheme (case (c)) with the existing scheme is provided in Table 5, which demonstrates that the proposed encryption scheme outperforms the existing ones in terms of computational complexity.

6.2.3 Peak Signal to noise ratio and Mean Square Error

The peak signal-to-noise ratio (PSNR) can be used to measure the encryption quality by computing the distortion in the recovered image compared with the plaintext image. The higher the value of PSNR, the higher the difference between the pixel values of the decrypted and plaintext image. Therefore, for a secure encryption scheme, PSNR values should be low. PSNR can be calculated for any image using (11).

$$PSNR = log_{10}(\frac{Max_O}{\sqrt{MSE}}) \tag{11}$$

Where Max_O is the maximum pixel value in the plaintext image.

Table 6 displays the various PSNR values. These values are calculated between the various plaintext-ciphertext image pairs. According to the values displayed in Table 6, the cipher-

Table 6 PSNR analysis

Encryption schemes	Cases	Aeroplane	Tank	Drone	Land
Proposed	Case(a)	12.6545	13.4558	12.1964	12.3789
	Case(b)	**10.1168**	10.67467	10.37678	10.1998
	Case(c)	10.1337	10.9978	10.3331	10.7916
Existing encryption	[34]	15.6497	17.3468	19.1667	2.1978
schemes	[35]	18.6789	19.6487	18.3167	19.9976
	[36]	21.3487	22.0976	23.1037	22.8896
	[42]	22.1578	21.678	21.6687	22.3448
	[43]	22.3187	21.678	21.3238	20.7872

The bold entry highlights the best result

Table 7 MSE analysis

Encryption schemes	Cases	Aeroplane	Tank	Drone	Land
Proposed	Case(a)	101.9682	103.2478	102.7678	103.2258
	Case(b)	**106.5782**	107.648	106.6687	107.989
	Case(c)	105.5685	106.1368	105.9986	105.6783
Existing encryption schemes	[34]	89.33878	89.8587	91.5389	90.5578
	[35]	92.9656	90.5578	91.9687	91.4598
	[36]	93.6768	91.198	93.6789	93.9989
	[42]	91.2268	93.0378	94.6787	95.1587
	[43]	90.6157	89.3382	90.3216	89.3335

The bold entry highlights the best result

text image created by the proposed encryption algorithm is fully distinct from the plaintext information.

In addition to PSNR, mean square error (MSE) may be used to quantify the difference between plaintext and ciphertext images. It may be determined mathematically as:

$$MSE = \frac{1}{AB}\sum_{i=1}^{A}\sum_{j=1}^{B}(P(i,j) - C(i,j))^2 \tag{12}$$

Where AB represents the size of the picture and $P(i,j)$and $C(i,j)$represent plaintext and ciphertext images, respectively.

In Table 7, the MSE values for several pairs of plaintext-ciphertext pictures are shown. The fact that the MSE values for case 'a' are less than those for case 'c' shows that the ciphertext image created in case 'c' is more random than the one generated in case 'a'. Moreover, a comparison with the existing encryption schemes shows that the proposed encryption scheme can generate more random ciphertext images than the existing ones.

6.2.4 Other security analysis

A few security parameters such as correlation, entropy, contrast, and energy, which are used in the proposed machine learning model, can be used to gauge the performance of an encryption scheme. The mathematical equations of such parameters are mentioned in Table 2.

Several values corresponding to the different ciphertext images are given in Table 8. It can be seen from the statistical values given in Table 8 that the proposed encryption scheme offers high security in case 'a' as compared to the other two cases. Moreover, the analysis of the existing schemes is also performed to show the effectiveness of the proposed research work. The statistical values given in Table 8 for case 'c' are much better than the existing scheme, which indicates that the proposed encryption scheme can perform better in terms of correlation, entropy, contrast, homogeneity, and energy.

7 Conclusion and discussion

The proposed work introduces a new encryption technique tailored for real-time applications within IoT systems. Merging machine learning with cryptographic elements like chaos and

Table 8 Statistical security analysis

Entropy analysis						
Proposed	Case(a)	7.9981	7.9972	7.9983	7.9965	7.9978
	Case(b)	7.9995	7.9993	7.9995	7.9994	7.9992
	Case(c)	**7.9998**	7.9997	7.9998	7.9998	7.9997
Existing encryption	[34]	7.9786	7.9863	7.9876	7.9915	7.9933
schemes	[35]	7.9915	7.9933	7.9919	7.9933	7.9915
	[36]	7.9934	7.9916	7.9975	7.9919	7.9976
Energy analysis						
Proposed	Case(a)	0.0160	0.0162	0.0159	0.0160	0.0163
	Case(b)	0.0159	0.0160	0.0158	0.0159	0.0160
	Case(c)	**0.0157**	0.0156	0.0157	0.0157	0.0156
Existing encryption	[34]	0.0167	0.0161	0.0167	0.0159	0.0162
schemes	[35]	0.0167	0.0166	0.0164	0.0162	0.0160
	[36]	0.0161	0.0162	0.0167	0.0168	0.0162
Contrast analysis						
Proposed	Case(a)	9.7698	9.8764	9.7886	9.8862	9.7793
	Case(b)	9.8893	10.0312	9.9986	9.9998	10.0031
	Case(c)	**10.6432**	10.9874	10.3667	10.8974	10.3794
Existing encryption	[34]	9.5568	9.8987	9.3468	9.5664	9.7487
schemes	[35]	9.8878	9.68780	9.5778	9.5176	9.6686
	[36]	9.8683	9.3598	9.4987	9.88784	9.5369
Correlation analysis						
Proposed	Case(a)	0.0068	0.0069	0.0089	-0.0064	0.0046
	Case(b)	-0.0013	0.0011	0.0031	0.0011	-0.0036
	Case(c)	**0.0001**	-0.0003	0.0004	0-0.0007	0.0001
Existing encryption	[34]	-0.0034	0.0068	0.0069	0.0098	0.0067
schemes	[35]	0.0012	0.0061	0.025	0.0012	-0.0035
	[36]	0.003	0.0022	0.0011	0.0013	-0.0034
Homogeneity analysis						
Proposed	Case(a)	0.4229	0.4669	0.4378	0.3996	0.4013
	Case(b)	0.4136	0.4168	0.39998	0.4236	0.4663
	Case(c)	**0.3679**	0.3996	0.3736	0.3996	0.9895
Existing encryption	[34]	0.4578	0.5099	0.5067	0.5178	0.5279
schemes	[35]	0.5075	0.4898	0.5368	0.5186	0.5067
	[36]	0.5269	0.5376	0.5167	0.5299	0.5178

The bold entries highlight the best results

bit-plane extraction, the proposed method prioritizes robust security for digital data while strategically minimizing computational time, aligning with the time-sensitive demands of IoT environments. In recognition of the diverse nature of digital images and the prevalence of redundant data, the encryption system strategically classifies image blocks into high, moderate, and low information categories using machine learning algorithms, ensuring selective encryption for optimal efficiency.

To enhance computational speed, the plaintext image is segmented into blocks, allowing the proposed machine learning model to classify them based on varying information content. High and moderate information blocks are then selectively encrypted, while low information blocks remain unaltered, significantly reducing processing time. The integration of security features, such as entropy, correlation, contrast, energy, and homogeneity, into the classification process further contributes to the encryption system's effectiveness.

Beyond computational considerations, the proposed encryption system ensures a high level of security for digital images, which is crucial in the IoT landscape. Pixel modifications employing confusion and diffusion techniques add an extra layer of security. The machine learning model's evaluation includes parameters such as accuracy, precision, recall, and F1-score, showcasing an impressive accuracy level of 97.4%. Rigorous comparisons with previous models underscore the proposed machine learning model's superiority. Security tests, encompassing entropy, correlation, energy, PSNR, and MSE, affirm the robustness of the encryption scheme.

The IoT-centric design of this encryption approach, attuned to the intricacies of real-time data processing within IoT applications, positions it as a promising solution for secure data transmission in the dynamically evolving IoT landscape.

Acknowledgements ADU authors acknowledge financial support from Abu Dhabi University's Office of Research and Grant Programs.

Funding statement Abu Dhabi University's Office of Sponsored Programs in the United Emirates (Grant Number: 19300752) funded this endeavor.

Data Availability Data will be made available on reasonable request.

Declarations

Conflicts of interest This manuscript is not submitted or under review in any other journal for publication.

References

1. Somaraj S, Hussain MA (2014) Securing medical images by image encryption using key image. Int J Comput Appl 104(3):30–34
2. Shafique A, Hazzazi MM, Alharbi AR, Hussain I (2021) Integration of spatial and frequency domain encryption for digital images. IEEE Access 9:149943–149954
3. Tang Z, Song J, Zhang X, Sun R (2016) Multiple-image encryption with bit-plane decomposition and chaotic maps. Opt Lasers Eng 80:1–11
4. Al-Maadeed TA, Hussain I, Anees A, Mustafa MT (2021) A image encryption algorithm based on chaotic lorenz system and novel primitive polynomial s-boxes. Multimed Tools Appl :1–22
5. Hosny KM, Kamal ST, Darwish MM (2022) A color image encryption technique using block scrambling and chaos. Multimed Tools Appl 81(1):505–525

6. Wang M, Wang X, Zhao T, Zhang C, Xia Z, Yao N (2021) Spatiotemporal chaos in improved cross coupled map lattice and its application in a bit-level image encryption scheme. Inf Sci 544:1–24
7. Anees A, Hussain I (2019) A novel method to identify initial values of chaotic maps in cybersecurity. Symmetry 11(2):140
8. Daemen J, Rijmen V (2001) Reijndael: The advanced encryption standard. Dr Dobbs J Softw Tools Prof Programm 26(3):137–139
9. Standard DE et al (1999) Data encryption standard. Federal Inf Process Standards Publ 112
10. Basu S (2011) International data encryption algorithm (idea)-a typical illustration. J Global Res Comput Sci 2(7):116–118
11. Barrett P (1986) Implementing the rivest shamir and adleman public key encryption algorithm on a standard digital signal processor. In: Conference on the theory and application of cryptographic techniques. Springer, pp 311–323
12. Jamal SS, Anees A, Ahmad M, Khan MF, Hussain I (2019) Construction of cryptographic s-boxes based on mobius transformation and chaotic tent-sine system. IEEE Access 7:173273–173285
13. Hussain I, Ahmed F, Khokhar UM, Anees A (2018) Applied cryptography and noise resistant data security.
14. Zhu S, Zhu C (2021) Security analysis and improvement of an image encryption cryptosystem based on bit plane extraction and multi chaos. Entropy 23(5):505
15. Hua Z, Zhu Z, Yi S, Zhang Z, Huang H (2021) Cross-plane colour image encryption using a two-dimensional logistic tent modular map. Inf Sci 546:1063–1083
16. Chen J-X, Zhu Z-L, Fu C, Zhang L-B, Zhang Y (2015) An efficient image encryption scheme using lookup table-based confusion and diffusion. Nonlinear Dyn 81(3):1151–1166
17. Choudhury A (2002) Fast machine learning algorithms for large data. PhD thesis, University of Southampton
18. Shafique A, Mehmood A, Elhadef M (2021) Survey of security protocols and vulnerabilities in unmanned aerial vehicles. IEEE Access 9:46927–46948
19. Wueller D, Fageth R (2008) Statistic analysis of millions of digital photos. In: Digital Photography IV, vol. 6817. SPIE, pp 186–194
20. Hoffman M, Shahriari B, Freitas N (2014) On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In: Artificial Intelligence and Statistics. PMLR, pp 365–374
21. Hussain I, Anees A, Aslam M, Ahmed R, Siddiqui N (2018) A noise resistant symmetric key cryptosystem based on s8 s-boxes and chaotic maps. Eur Phys J Plus 133(4):1–23
22. Rehman MU, Shafique A, Khan KH, Khalid S, Alotaibi AA, Althobaiti T, Ramzan N, Ahmad J, Shah SA, Abbasi QH (2022) Novel privacy preserving non-invasive sensing-based diagnoses of pneumonia disease leveraging deep network model. Sensors 22(2):461
23. Manikandan V, Masilamani V (2018) Reversible data hiding scheme during encryption using machine learning. Procedia Comput Sci 133:348–356
24. Hussain I, Anees A, Algarni A (2018) A novel algorithm for thermal image encryption. J Integr Neurosci 17(3–4):447–461
25. Lu Q, Zhu C, Deng X (2020) An efficient image encryption scheme based on the lss chaotic map and single s-box. IEEE Access 8:25664–25678
26. Wang X, Çavuşoğlu Ü, Kacar S, Akgul A, Pham V-T, Jafari S, Alsaadi FE, Nguyen XQ (2019) S-box based image encryption application using a chaotic system without equilibrium. Appl Sci 9(4):781
27. Anees A, Siddiqui AM, Ahmed F (2014) Chaotic substitution for highly autocorrelated data in encryption algorithm. Commun Nonlinear Sci Numer Simul 19(9):3106–3118
28. Pareek NK, Patidar V, Sud KK (2006) Image encryption using chaotic logistic map. Image Vis Comput 24(9):926–934
29. Ahmad J, Hwang SO (2015) Chaos-based diffusion for highly autocorrelated data in encryption algorithms. Nonlinear Dyn 82(4):1839–1850
30. Shafique A, Ahmed F (2020) Image encryption using dynamic s-box substitution in the wavelet domain. Wireless Pers Commun 115(3):2243–2268
31. Leng L, Zhang J (2013) Palmhash code vs. palmphasor code. Neurocomputing 108:1–12
32. Leng L, Li M, Kim C, Bi X (2017) Dual-source discrimination power analysis for multi-instance contactless palmprint recognition. Multimed Tools Appl 76:333–354
33. Leng L, Li M, Teoh ABJ (2013) Conjugate 2dpalmhash code for secure palm-print-vein verification. In: 2013 6th International congress on image and signal processing (CISP), vol. 3. IEEE, pp 1705–1710
34. Pourasad Y, Ranjbarzadeh R, Mardani A (2021) A new algorithm for digital image encryption based on chaos theory. Entropy 23(3):341
35. Li Z, Peng C, Tan W, Li L (2021) An effective chaos-based image encryption scheme using imitating jigsaw method. Complexity 2021

36. Musanna F, Dangwal D, Kumar S (2021) Novel image encryption algorithm using fractional chaos and cellular neural network. J Ambient Intell Hum Comp 1–22
37. Shannon CE (1949) Communication theory of secrecy systems. Bell Syst Tech J 28(4):656–715
38. Ye G, Pan C, Huang X, Zhao Z, He J (2018) A chaotic image encryption algorithm based on information entropy. Int J Bifurcation Chaos 28(01):1850010
39. Sirichotedumrong W, Kiya H (2020) Visual security evaluation of learnable image encryption methods against ciphertext-only attacks. In: 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, pp 1304–1309
40. Chen J, Chen L, Zhou Y (2020) Universal chosen-ciphertext attack for a family of image encryption schemes. IEEE Trans Multimed 23:2372–2385
41. Lin C-H, Hu G-H, Chan C-Y, Yan J-J (2021) Chaos-based synchronized dynamic keys and their application to image encryption with an improved aes algorithm. Appl Sci 11(3):1329
42. Kumar CM, Vidhya R, Brindha M (2022) An efficient chaos based image encryption algorithm using enhanced thorp shuffle and chaotic convolution function. Appl Intell 52(3):2556–2585
43. Hao P-Y, Chiang J-H, Lin Y-H (2009) A new maximal-margin spherical-structured multi-class support vector machine. Appl Intell 30(2):98–111
44. Ge R, Zhang L, Zhang T, Li S, Ma Y (2015) A modified spiking neuron circuit with memory threshold and its application in image encryption. In: 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS). IEEE, pp 872–877
45. Shafique A, Shahid J (2018) Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps. Eur Phys J Plus 133(8):1–16
46. Khan M, Asghar Z (2018) A novel construction of substitution box for image encryption applications with gingerbreadman chaotic map and s8 permutation. Neural Comput Appl 29(4):993–999
47. Khan M, Shah T, Batool SI (2016) Construction of s-box based on chaotic boolean functions and its application in image encryption. Neural Comput Appl 27(3):677–685
48. Ramzan M, Shah T, Hazzazi MM, Aljaedi A, Alharbi AR (2021) Construction of s-boxes using different maps over elliptic curves for image encryption. IEEE Access 9:157106–157123
49. Maniyath SR, Thanikaiselvan V (2020) An efficient image encryption using deep neural network and chaotic map. Microprocess Microsyst 77:103134
50. Ravanna C, Keshavamurthy C (2019) A novel priority based document image encryption with mixed chaotic systems using machine learning approach. Facta Univ Electron Energ 32(1):147–177
51. Xiao H-P, Zhang G-J (2006) An image encryption scheme based on chaotic systems. In: 2006 International Conference on Machine Learning and Cybernetics. IEEE, pp 2707–2711
52. Shafique A, Ahmed J, Rehman MU, Hazzazi MM (2021) Noise-resistant image encryption scheme for medical images in the chaos and wavelet domain. IEEE Access 9:59108–59130
53. Anees A, Hussain I, Algarni A, Aslam M (2018) A robust watermarking scheme for online multimedia copyright protection using new chaotic map. Secur Commun Netw 2018
54. Shafique A, Ahmed J (2021) Dynamic substitution based encryption algorithm for highly correlated data. Multidimens Syst Signal Process 32:91–114
55. Shafique A, Ahmed J, Boulila W, Ghandorh H, Ahmad J, Rehman MU (2020) Detecting the security level of various cryptosystems using machine learning models. IEEE Access 9:9383–9393