https://doi.org/ 10.33472/AFJBS.6.9.2024.2139-2159

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Medical Image Security using POB based Artificial Intelligence Techniques for Reversible Data Hiding

Jagadevi N Kalshetty[1], Dr. Piyush Kumar Pareek[2]
Nitte Meenakshi Institute of Technology,
NREA UOM, Bengaluru, India
jagadevi.n.kalshetty@nmit.ac.in

Article History

Volume 6,Issue 9, 2024

Received: 26-03-2024

Accepted : 28-04-2024

doi: 10.33472/AFJBS.6.9.2024.2139-2149

Abstract

Thanks to its intuitive design, the picture hosting platform is quickly gaining users, although some worry about their privacy. Unfortunately, visual privacy protection solutions are typically permanent in their attempts to strike a compromise between privacy and usability. There is a lot of activity in the area of information security related to reversible data hiding (RDH). A cover medium can contain a secret in RDH. Applications that require deformation-free cover recovery in addition to hidden secret recovery highlight the importance of RDH over alternative data-hiding strategies. This paper presents a permutation ordered binary (POB) authentication and hiding scheme for medical images. The scheme consists of three steps: image recovering the original information with hidden information. Its purpose is to enhance the embedding ability of data in encrypted medical images and to protect medical image security. Two encrypted shares are created after compressing and encrypting the two new share pictures. With the use of the POB algorithm, secret data was embedded and authentication bits were attached to each of the two encrypted shares. During the recovery phase, the work presents a framework for ensemble models that utilises machine learning and deep learning to efficiently recover. Cover picture recovery is guaranteed by the majority vote of the various trained models. This study use the flamingo optimisation algorithm (FOA) to fine-tune the ensemble models' hyper-parameters. At last, the original medical image is restored after extracting the embedded hidden message. The results of the studies show that the strategy suggested in this study is superior when it comes to data embedding and recovery information. Excellent security has been achieved throughout the entire process, according to the experiments and analysis, using the proposed system. As data was being concealed, it managed to achieve high embedding capacity, PSNR, rate, and low SSIM.

Keywords: Reversible data hiding; Permutation ordered binary; Flamingo optimization algorithm; Double hiding; Recovery Information; Data Security.

Introduction

Enabling the embedding of sensitive data into various types of media, rescindable data hiding (RDH) has grown into a compelling approach. The military, healthcare, national government, and copyrighted media are all part of this [1]. embedding techniques—histogram shifting [2], difference expansion, and lossless compression—have traditionally been the focus of research in the plaintext domain. Improving the carrier pictures' visual quality while increasing the embedding rate is the main goal of these techniques [3]. To sum up, RDH technology is a powerful tool for securely embedding sensitive data in a way that can be reversed, all without harming the original carrier [4]. Its usefulness is practically limitless, and it shines in industries where protecting sensitive information is of the utmost importance [5]. Nevertheless, in order to create an RDH-EI approach that is appropriate for information-sensitive circumstances, it is essential to utilise encryption methods for content-sensitive scenarios [6]. Using an encryption key, these methods convert the original picture into an unreadable variant, therefore securing the image content.

Particularly in delicate industries where data security is of the utmost importance, RDH in encrypted images (RDH-EI) has several advantages [7]. Although RDH technology has its limitations in certain data hider circumstances, it is highly recommended for instances where there is absolutely no room for picture loss due to its inherent reversibility [8]. With the use of encryption methods, we can create an RDH-EI approach for encrypted domains that is appropriate for situations when the material is sensitive. The conventional RDH-EI model isn't very useful in situations when there are more than one data hider since it only accounts for one data provider and one data hider [9-10]. Nevertheless, the ability to reverse the process allows for flawless restoration of the initial carrier while secretly extracting data. Because of its quality, RDH technology is perfect for uses where there can be no room for picture loss, such as medical imaging in healthcare, government photographs in court decisions, and military satellite photography [11]. The problem of how to securely embed and retrieve sensitive information from encrypted pictures is efficiently addressed by RDH-EI, a subset of RDH-ED. As a carrier, encrypted picture data is used in this method [12]. Incorporating the data within the encrypted picture prevents the carrier image from losing any pixels when extracted. In order to safeguard patients' private information, data-hiding technology is frequently employed in telemedicine applications [13]. This method involves affixing data information that must be sent to the carrier picture; it is a technological term. The lossless photos under different tampering assaults is one of numerous medical image-based data-hiding techniques that exist; nonetheless, this scheme is limited to simple authentication and still has issues when it comes to carrying extra data [14].

For the purposes of neighbourhood and watermark refining, Liu et al. [15] employed a POB digital method in conjunction with lossless recovery and picture tamper detection. While the strategy did work under some conditions, the experiment showed that recovering large-scale tampering was still not without its challenges. A solution to these issues was put up by Li et al. [16] in their proposal for medical picture authentication recovery using the POB method. Both theoretical and experimental studies have shown that this approach permits In order to address these shortcomings, this study suggests a twofold POB system-based method for medical picture concealment and authentication. Not only does the technique manage medical picture authentication, it may also communicate extra secret data while doing so. To avoid tampering with the high-bit sensitive pixels in the ROI, the original

medical image's region of significance (ROS) is first segmented into non-importance (RONI). Then, to create two shares, the two regions are cross-reorganized and separated. After that, we used a hyperchaotic Lorenz encryption scheme to secure the aircraft after repeatedly placing the compressed data within it, making sure it had the same size as the original photographs. The last piece of the data concealment and authentication puzzle was the usage of the double POB number mechanism. Once the authentication process is complete, the hidden message may be deciphered and the original medical picture can be restored. Even if data is altered, the original picture can be restored in its entirety by inserting the data again.

This research mostly contributes to the following areas:

- The study employed a dual point-of-brow digital system to accomplish data concealing and medical picture authentication at the same time. Data concealment has a huge embedding capacity and is reversible; authentication is pixel-level, which to even little manipulation and fulfils the requirements of medical imaging.
- Using compressed data repeated filling, the study suggests a strategy for recovering medical images tampered with. It is possible to immediately use the data from the untampered area for tampering recovery in the event that authentication fails in the authentication stage because of tampering.
- A FOA model optimises the parameters before a machine learning and deep learning model recovers the data.

What follows is the outline for the remainder of the article. The relevant prior works to the proposed system are detailed in Section 2. Section 3 explains in detail how medical photos may be authenticated and hidden using a double POB approach. Section 4 offerings the findings and analysis of the experiments. In Section 5, this paper comes to a close.

2. Related works

In order to improve telemedicine, Qu et al. [17] optional a secure multi-party computing system that incorporates a high-capacity reversible data hiding (RDH) technique. In a three-by-three picture block, the pixel values are split into embedded pixels (EPs) and sampled pixels (SPs). After two steps, secret data is embedded into the prediction errors of the two types of pixels. To begin, the prediction error of EPs is calculated by having two edge servers use the four SPs in the block to make EP predictions. The secret data is integrated into EPs using the histogram shift approach and addition operations. Step two involves using the same approach as step one to embed secret data, and it involves calculating the prediction error of SPs. On the UCID dataset, the suggested approach obtains an average embedding rate of 0.47 bpp, which is greater than classical and state-of-the-art techniques, according to the experimental findings. Additionally, current attacks are not able to penetrate the suggested methodology.

In order to encrypt photos with a large payload, Gao et al. [18] suggested a hierarchical reversible data concealing scheme. There are two categories of original picture pixels: those that are predicted using multiple linear regressions (MLR) and those that are not. By utilising adaptive adjustment of anticipated data and multiple bits prediction (MBP), the MLR predictable pixels are transformed into a two-level label map of pixels. Using a straightforward one-bit prediction method, the label map of pixels (LMP) is generated from

the non-MLR predictable pixels. The encrypted photos incorporate the extra data generated by compressing three LMPs. The suggested method achieves a high embedding rate because of the strong connection between pixels in the visible area of the picture. The UCID dataset yielded an average payload of 2.9885 bpp, the BOWS-2 dataset 3.759 bpp, and the BOSS dataset 3.883 bpp. Experimental results show that the state-of-the-art RDHEI methods in payload on most images, and comparisons of performance show that the proposed method has made great improvements.

In order to attain greater EC, Fu et al. [19] announced a new RDHEI that is pixel prediction based. It adaptively combines "L"-shaped block embedding (LBE) with improved binary-block embedding (IBBE). To be more precise, eight binary prediction-error bit-planes (PEBPs) are created after the original picture is forecasted to obtain the prediction-error image. The nett EC to encode each PEBP is then calculated using the suggested LBE and IBBE. To further guarantee that the intra-block -based PEBPs stay unaltered, a novel bit-plane selection encryption is also created. Data hiding eventually finishes data embedding adaptively based on PEBP coding. To take security to the next level, we encrypt the carrier picture and secret data using the 4-D hyperchaotic system. Our technique outperforms several state-of-the-art schemes, according to extensive experimental data.

A new approach to compressible encrypted domain reversible for OpenEXR pictures has been proposed by Kikuchi and Imaizumi [20]. The suggested technique has three primary benefits. In order to implement block-by-block encryption, it first incorporates an encryption-then-compression method. Because of this, tagged encrypted pictures are able to be compressed very efficiently. We use an RDH technique that is based on prediction error expansion to incorporate a payload in the encrypted domain. Because of this, our approach has a great concealment capability. We can decode a tagged encrypted image without data extraction, and the system also permits customisable restoration patterns. The result will be a marked picture in this instance. Data concealment and encryption both work while keeping the original image's dynamic range intact. Both compression efficiency and marked-image quality are enhanced by this. To demonstrate the efficacy of our suggested strategy, we conduct an experiment to assess the JPEG XT compression efficiency, marked-image quality, and concealing capability.

To record the in a both ends, Liu et al., [21] introduced a unique bi-directional block encoding (BDBE) approach. This is the first of its kind. When compared to both conventional and cutting-edge encoding techniques, this technology enables the encoding of pictures with reduced file sizes. Our high-capacity RDH-EI method is based on the BDBE approach. Here, the content owner uses pixel prediction to provide space for data embedding, and then uses BDBE to encode the prediction mistakes. The data that has been encoded is then sent to a data hider after being encrypted with a secure stream cypher like the Advanced Encryption Standard. In order to store, handle, or process sensitive information, the data hider might insert it inside the encrypted picture. An authorised recipient can reliably get both the embedded data upon receiving the data. The experimental findings show that our RDH-EI scheme outperforms many state-of-the-art systems in capacity.

A privacy-preserving method has been proposed by Shi et al., [22] that utilises a neural network to conceal reversible medical data within plaintext encrypted photos. In order to get accurate results in classification and segmentation, we first build a combined model of

a neural network using U-Net and AlexNet. In order to prevent visible disclosure of patient privacy and undue attention caused by ciphertext encryption, we then use plaintext encryption to hide the ROI. We secure and include private data into the region of non-interest (RONI) using a skipping concealment technique. The suggested method is notable because it may be separated, which increases both flexibility and security. An enhanced plaintext encryption approach is used to encrypt the ROI after a combined neural network model separates the medical picture into ROI and RONI. This is done for the benefit of the content owners. Without gaining access to the original picture data, data hiders insert hidden bits into RONI via skipping hiding. Depending on their level of power, recipients can access the data they need. Our system successfully reduces the risk of loss or mismatch by addressing the difficulty of keeping medical photos and diagnostic information separately. Without lowering the quality of ROI images, it reduces the danger of privacy exposure in diagnostic data. Our experimental results show that our scheme is very accurate (F1-score, recall, accuracy, and precision all exceeded 0.982), has great efficiency and low complexity (processing time of 3.19 s), and is fully reversible (restored image:: 1). We also have strong plaintext encryption (plaintext-encrypted stego-image: average PSNR: 43 dB, SSIM: 0.99, NC: 0.99), high complexity. In addition, the present privacy protection concerns are well addressed by our system.

One such method is PVPBC-RDHEI, which was suggested by Zhang et al., [23] and is based on pixel value preprocessing and block classification. It allows for reversible data hiding in encrypted pictures. To ensure that the correlation between adjacent pixels in the encrypted picture block is maintained, the content owner preprocesses the original image's pixel values before encrypting the image using a mixture of bitwise exclusive-or and block permutation. Using a top-down count of the pixel block's continuously constant value bit planes, the data hider sorts all the blocks into six categories upon receiving the encrypted picture. It then uses Huffman coding to create associated indications for each of these types. As a result, storage space for data may be made available in every pixel block of the image. Based on the various keys, the receiver can either retrieve the original image or extract the supplementary data individually. The suggested technique guarantees security and reversibility and has a far higher embedding rate than state-of-the-art schemes, according to the experimental data.

2.1. Problem Statement:

The problem at hand is the need to enhance security measures for medical images while simultaneously optimizing data embedding and recovery processes. Current approaches often sacrifice usability for privacy or vice versa, leading to inefficiencies in both areas. Additionally, existing encryption methods may not adequately address the unique challenges posed by medical image security. There is a growing demand for a comprehensive solution that can securely embed data within medical images while ensuring efficient recovery and preserving image integrity.

3. Proposed model

The hiding process has been shown in figure 1.

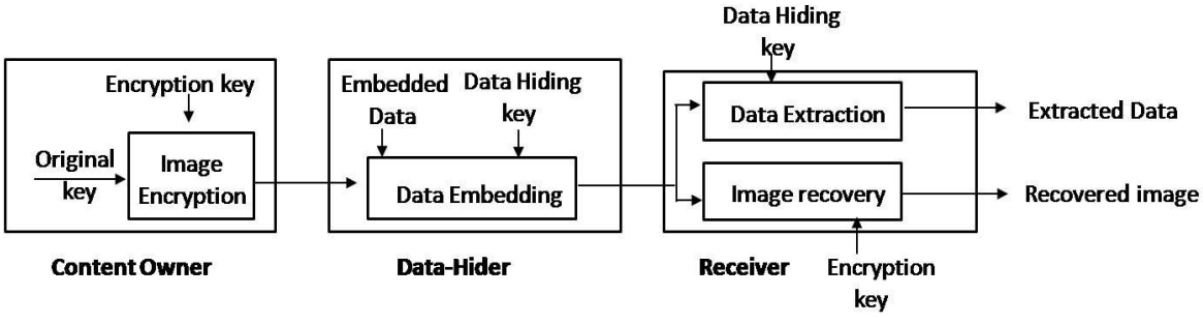

Figure 1. Encryption & Data Hiding Process

3.1. Dataset Detail

(1) White blood cell (WBC) dataset: This dataset records measures of breast cancer patients and was retrieved from the UCI machine learning repository. Among its 699 samples, there are nine characteristics. You can get this dataset, which has two classes, from the provided URL.

$$(Prognostic)\ https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin$$

The second dataset is the PID dataset, which was originally sourced from the National Institute of Diabetes, Digestive Diseases, and kidney diseases. With 768 samples and eight qualities, it's perfect. To access this dataset, go to https://www.kaggle.com/uciml/pimaindians-diabetes-database/version/1. We split the two datasets in half, using 70% for training and 30% for testing, in order to conduct our experiments.

3) Dataset for fundoscopy. Classification of diabetic retinopathy is typically done using the Fundoscopy dataset. In total, there are 3,662 training photos and 1,928 test images in the APTOS 2019 BD [24] dataset. 'Normal' through 'Mild/Moderate/Severe/Proliferative' are the five categories that are indicated on each photograph.

4) Dataset on skin lesions. Classification of skin diseases is accomplished using the Skin Lesion dataset. There are 10,015 training photos and 1,512 testing images in the dataset, which is sourced from the ISIC-2018 Challenge [25]. Five levels, ranging from "NV" to "MEL/BCC/BKL/DF/VASC/SCC/UNK," are indicated on each image.

3.2. Pre-processing using Data compression and data filtering

At this point, the entire dataset is split up into two distinct sets, which are referred to as ShareA and ShareB respectively. The initial step involves partitioning ShareA into 8 × 8 blocks that do not overlap. Subsequently, the Huffman coding technique is employed to compress each individual block. After that, the beaten data of each block is saved in C, where the length of the compressed data is n. After that, the compressed data is repeatedly filled in a Share picture that is 512 × 512 in size, which is the same size as the original medical image. In order to ease image recovery, the Share image needs to be filled repeatedly (512 × 512)/n times. Data compression procedures are carried out in the same manner for ShareB. Using a concrete example, the study provides an illustration: if the initial size of the medical image is 512 × 512, then it has a the extracted ROS is 352 × 352, it has 123,904 pixels. Subsequently,

the retrieved ROS pixels are reduced by roughly half. Following this, ShareA is compressed in of pixels after compression is around 30,621. To obtain an image with dimensions of 512 × 512, it is necessary to fill the compressed data with a minimum of ⌊262, 144/30, 621⌋, which is equivalent to eight times.

3.3. Double Hiding

The pre processed image is by the image's owner.

3.3.1. Image Encryption

The completion of the data density and data filling processes results in the creation of two new Share pictures, which are referred to as Share A′ and Share B′. The hyperchaotic Lorentz system creates random numbers of the same size as the image in order to encrypt the two new Share images. The key is saved as K1, and the same key is utilised for both encryption and decryption at the same time. After that, the random numbers that were generated are utilised for XOR encryption, which results in the acquisition of two encrypted Share A′ and encrypted Share B′ images that are of the same size as the initial medical image. Subsequently,the embedding of the secret and the embedding of the authentication information based two encrypted Shares.

3.3.2. Data embedding

Presented in this section of the paper is the data-hiding procedure that was submitted for consideration. embedding function are both included in this. In the next subsections, these functions are broken out in greater detail.

A. Embedding Function

In the medical field, digital images are typically saved in the DICOM format. DICOM facilitates the interchange of medical images in a way that is less dependent on the manufacturer of the imaging equipment. Images often have an uint16 data type, which indicates that each pixel contains a positive integer between 0 and 65536, inclusive.

Included in the cover image of the proposed data-hiding technique are the patient's photo and the watermark logo. The DICOM file's image serves as the cover image, concealing the sensitive data within. In order to explain the suggested embedding process, let's pretend the cover image is 512 × 512 in size. For this reason, the embedding process does not alter every pixel in the image; rather, it picks out specific blocks of the image and inserts the data into them in order to reduce distortion as much as possible. The payload size determines the minimum number of blocks needed.

B. Extracting Function

The recipient side is responsible for applying the extraction procedure to the conventional DICOM file. Just like embedding, data extraction is a rather straightforward procedure. The DICOM file is initially read to obtain the cover image. Afterwards, 10-by-10-dimensional pieces are used to reshape the cover image. The compression field is used for pre-processing in every 10 × 10 block.

3.3.3. Double Embedding Based on POB

Sreekumar and Sundar were the ones who first suggested using POB numbers [27]. Here, n is an integer more than or equal to r and r is not negative, and the system is represented as POB (n, r). A string has n bits and r 1s, where n is the sum of bits in this context. The range of possible POB numbers includes P(A) is $0, 1, \cdots \binom{n}{r} - 1$, where $A = a_{j-1}a_{j-2} \cdots a_0$.

The subsequent can be used to find P(A):

$$P(A) = \sum_{j0}^{n-1} a_j \binom{j}{v_j} \quad (1)$$

Where $v_j = \sum_{j=0}^{n-1} a_j$.

Using the aforementioned formula, one may determine the POB value from its binary representation and vice versa.

An example of a 10-bit binary string with a POB value that can be determined using Equation (2) is 1011001010, which is represented as POB (10, 5).

$$\begin{aligned} lP(A) &= a_0 \times \binom{0}{0} + a_1 \times \binom{1}{1} + \cdots + a_8 \times \binom{8}{4} + a_9 \times \binom{9}{5} \\ &= 0 + 1 + 0 + \binom{3}{2} + 0 + 0 + \binom{6}{3} + \binom{7}{4} + 0 + \binom{9}{5} \\ &= 185 \end{aligned} \quad (2)$$

It is possible to convert the gotten POB value of 185 to the matching 8-bit binary, which is 10111001. Some data-hiding studies are severely affected by this finding. While concealing data, it can re-encrypt it and compress it without losing any information.

In the same way, 010111011 is POB (9, 6), and the associated POB charge is 10, while 010101101 is POB (9, 5), representing a POB worth of 28. The paper's 8-bit pictures plus the two-bit embedded secret and authentication bits allow us to transform the 10-bit POB (10, r) string used in each operation to an 8-bit can be anything from 0 to 251.

Encrypted Shares can have sensitive messages (such as patients' identities, conditions, medical histories, etc.) embedded within them using the POB algorithm, which also synchronises the re-encryption process. To ensure that sensitive areas' information remains intact, the secret message is ′ before moving on to encrypted ShareA′. This is because encrypted ShareB′ contains less important pixel information. This is the process for embedding a secret message using POB tangible steps:

Step 1: In order to find out if the secret message is too long, it first iterates through the pixels of encrypted ShareB′. When that doesn't work, the secret message is inserted directly, and the process continues with stages 2 and 3 until all the information is embedded. Step 4 is executed following the completion of steps 2 and 3, in the event that it surpasses the capacity.

Step 2: To change each pixel of encrypted Share B′ into a ten-bit binary number, the secret message w0, w1 ∈ {0, 1} is chosen and embedded into each pixel.

Step 3: Encrypted Share B's secret information is compressed from using the POB number system, and new secret information-containing pixels are generated at the same time.

Step 4: The embedding process is carried out in encrypted ShareA′ using the procedures in steps 2 and 3, in the event that encrypted Share B′ does not have sufficient capacity.

After the secret message is implanted, the encrypted picture with the secret message is obtained. Then, the authentication bit is embedded in both encrypted images with the secret information using the POB. The following are the concrete steps for embedding authentication information based on POB:

Step 1: Two authentication bits are computed for each 8-bit pixel in the containing the secret message and subsequently linked to that pixel.

Step 2: The first one is the number of 1s eight pixels, which is the first field in the authentication bit. The second field is the number of 1s in the last four digits, which is 1 when the number is odd and 0 when it is even.

Step 3: And lastly, using the POB number system, the ten-bit pixels that include verification bits are transformed into values, and Share A" and Share B" are produced.

Both the capability for data concealing and the security of images are enhanced by the double embedding based on the POB method. The reason behind this is that the approach has the ability to automatically re-encrypt the pixels that are now being processed. This destroys the local correlation that was kept by the cryptosystem's design. Additionally, once the encrypted image is embedded, it becomes more difficult to comprehend.

3.4. Histogram shifting process

The interpolation errors (e) can now be defined as (Eq. 3).

$$e = P - P' \quad (3)$$

The results of the above calculation are the interpolated pixel values. After that, we divide the histogram of the error sequence in half. As a matter of convenience, we will refer to the numbers '0' and above as the right half, and negative values as the left half. In this context, "RM" and "LM" represent the limits, respectively, of the right and left sides. The symbols 'RN' and 'LN' represent the right and left half's minimum values, correspondingly. Data embedding is followed by evaluation of the new additive interpolation error. Bit '0' indicates RM while bit '1' represents RM + 1 if the anticipated error is RM. The left side can be treated in the same way. This procedure was carried out until all of the data had been incorporated.

Problems with overflow and underflow occur, as in previous RDH algorithms, when the values of the pixels along the natural boundary change from 255 to 256 or 10 to 1. This is avoided by incorporating the data solely into estimation errors for pixels having values ranging from 1 to 254. Nevertheless, uncertainties persist because the pixel values can change from 1 to 0 or 254 to 255 while the data is being embedded. Peripheral pixels are the name given to these pixels. In order to identify real or faux border pixels in a tagged image, the extraction method employs a boundary map. The image's border map is strategically positioned and embedded using LSB replacement by making optimal use of the image's marginal region. There is enough room in the marginal region to accommodate the relatively short boundary map, even with a high embedding rate; hence, many parameters are

embedded in the marginal area. These features dictate the procedure for recovering the initial picture and extracting data.

3.5. Recovering the Original Image besides Hidden Evidence

Here, we take the data-encrypted picture E' key K as inputs and attempt to produce an output that contains both the original picture and the secret information that was hidden inside. Please be informed that the receiver receives the data-encrypted picture E', the decryption key K, and the size of the processing block for the image A × A. You can't get back the original image or the secret info without these specifics.

To start the recovery procedure, the receiver divides the image E' into TN nonoverlapping chunks with dimensions A × A. The suggested RDH method for picture restoration employs ML models). Consequently, the machine learning models need to be trained before we can process the nonoverlapping blocks of the picture E'. The non-overlapping blocks at picture position i in E' are denoted as NBi. That means

$$E' = \bigcup_{i=1}^{TN} NB_i \quad (4)$$

The receiver is tasked with retrieving the concealed data unit from each distinct block in NB_i. The receiver does this by processing each block independently. To clarify, the Fibonacci transform function is applied to a block NB_i, which represents a collection of potential values that can be included in a block of size $A \times A$ pixels. The range of possible data units contained length A is $\{0,1,2,\ldots,j-1\}$, historical value of the Fibonacci transform (see Equation (5)). We have a garbled version of the block NB_i for every charge in the set $\{0,1,2,\ldots,j-1\}$. Let,

$$ZZ = \{0,1,2,\ldots,j-1\} \quad (5)$$

and the function is $F(x,y)$. Thus,

$$F(NB_i, ZZ_l) = NB_{i,ZZ_l} \quad (6)$$

where i designates the site of the block NB in E', ZZ_{l} reflects from the set ZZ with $0 \leq l \leq j-1$, besides NB_{i,ZZ_l} is the knotted block generated by transient NB_{i} and ZZ_{l} as the advices to the function.

To illustrate, the range of possible values for the Fibonacci transform within these blocks is {0,1,2,...,15}, presuming that the block size A is 8. So, we take this dataset and apply the Fibonacci transform to every block, resulting in 16 unique iterations of the same block.

Decrypting each of the subsequent $NB_{\mathrm{i,l}}$ blocks and to the decrypted form of the $NB_{\mathrm{i,l}}$ be designated as $DB_{\mathrm{i,l}}$. The function of the collaborative tactic is to identify the original chunk after the diverse $DB_{\mathrm{i,l}}$ blocks. The The ensemble method involves a presumption of certainty for a block when two or more models indicate a preference for or classification of that block. $DB_{\mathrm{i,l}}$ as the unique block. When it comes unfortunate to it, CNN models take in blocks directly to extract their own them, while SVM and KNN models use the blocks' feature vectors to do classification.

In order to determine the initial block, the RDH system takes a voting perspective into account. If most of models agree that a given block is unique, then it will be considered

original. In other words, if two out of three models agree that the block is unique, then it is considered original. In the worst-case situation, the block that the models have the least confidence in classifying as an encrypted block is used.

Now, block be RB_{i}. Thus,

$$RB_i \in DB_{i,l} \quad (7)$$

Once the original block RB_i the picture E's concealed information in the can be extracted after recovery. In order to extract the secret data from every 〚block of E', we adhere to the procedures outlined below.

1. Encrypt the recovered block RB_i using the encoding key. This block, say EB_{i}.

2. Pass EB_{i} the function, with j being its period, and every data unit set $ZZ = \{0,1,2,\ldots,j-1\}$ as inputs. Equation (8) is used in this context.

$$F(EB_i, ZZ_l) = NEW_{i,l} \quad (8)$$

3. Checked for the equality among the knotted block $NEW_{\mathrm{i,l}}$ and NB_{i}.

4. Once a match is found, we end the procedure. The value l that corresponds to that block denotes the data that is buried within it. NB_{i}.

The next step is to restore each original block and its concealed data units so that we can recreate the original cover twin and secret info stream.

3.5.1. Training of the Algorithm

The five feature vectors used SVM besides KNN models are listed below. In order for the model to be able to differentiate between original and encrypted blocks, training must be conducted. What makes them:

1. F1 (Entropy): The degree of unpredictability is clear as the entropy of the copy. The entropy of image is advanced than that of the unique image. For this reason, entropy is a key component in the block categorisation process.

2. F2 (Standard Deviation): Data or measured values are said to be out of whack if their standard deviation is higher than the mean. In this case, the standard deviation checks how far individual pixels are from the average. A more encrypted block is indicated by a greater value.

3. F3 (Smoothness): You may learn about the relationship between neighbouring image pixels by looking at the smoothness metric. Therefore, in comparison to the original blocks, the encrypted blocks' smoothness value is high.

4. The difference in grey level, also known as F4 (Alteration in Gray-Level), is the alteration between the highest and lowest grey levels generated from the histogram.

Fifthly, the average gradient (F5) takes into account the variation in each neighbouring pixel. Because of this, it has the potential to be a helpful attribute for block categorisation.

3.5.2. Flamingo for Tuning the Hyperparameters of the Classifier

Flamingo optimisation, a substitute for the conventional Adam optimizer in ML classifiers, finds the global optimal search space. When it comes to adjusting the classifier, the flamingo optimisation is the way to go because of its strong features. To get the best answer, the deep classifier makes use of flamingo optimisation by adjusting the hyperparameters. The deep classifier's bias and weights are fine-tuned as they are learnable parameters.

Inspiration: The optimisation process is considerably aided by the gregarious, flock-living, scavenging, and nomadic behaviour of flamingos. At first glance, flamingos' feeding habits can be classified as either gregarious, bill-scanning, or claw locomotive. Under conditions of resource scarcity, this behaviour enables foraging and subsequent global optimisation. The flamingo's traits are included into the deep classifier. Below, we mathematically explain the stages involved in the flamingo optimisation and modify the classifier to find the optimum answer.

(i) Scavenging behaviour

Behaviour indicative of social interaction: at first, the flamingo that finds the food will signal to the other flamingos to move to a spot where they can reach it. The mathematical expression of food abundance in the kth dimension is as follows: if the flamingo seeks the optimal solution, which is the location with the most food, yd_{k}.

(ii) Fitness function

After analysing each flamingo's fitness function, the one that offers the best candidate solution is chosen as the fittest. One can find the fitness function by,

$$f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (9)$$

The global solution to that corresponds to the maximal value of fitness.

(iii) Bill-scanning behaviour

In order to find food, flamingos submerge their beaks in the water. Once they catch something, they swallow it upside down, which allows them to filter out any extra water and waste. In flamingos scan more thoroughly, with a scanning radius that changes depending on the circumstances. The location of the flamingo at position l in the kth dimension is denoted as 〖yl〗_k. There is a chance of small mistakes occurring during information interchange, however they can be overcome by using ordinary normal distribution. The mathematical expression for the extreme distance that the flamingos can cover is

$$A_1 = M_1 \times yd_k + \lambda_1 = y_{lk} \quad (10)$$

Assuming initially that the skimming is done at its extreme distance and that M_1 is a uniformly distributed random value, the variance in the flamingos' scanning range is described by

$$A_2 = M_2 \times |M_1 \times yd_k + \lambda_1 + y_{lk}| \quad (11)$$

M_2 is a random value that shadows a typical unchanging distribution, l1 and l2 are accidental statistics in the variety [−1,1].

(iv) Claw locomotive behaviour

Wherever there is an claws will go. The flamingos' range is represented by 〖yd〗_k, and the area with abundant food is $\lambda_l \times yd_k$ which food in the nth iteration is given by

$$s_{lk}^{n} = \lambda_1 \times ys_k^n + M_2 \times |M_1 \times yd_k + \lambda_1 + y_{lk}| \quad (12)$$

Mathematically, the flamingo's position is updated with different locations and is articulated as

$$y_{lk}^{n+1} = \frac{(y_{lk}^{n} + \lambda_1 \times ys_k^n + M_2 \times |M_1 \times yd_k + \lambda_1 + y_{lk}|)}{R} \quad (13)$$

Here, y_{lk}^{n+1} is the the lth dimension in the kth iteration of (n + 1), and y_lk^(n+1) is the lth flamingo with the kth repetition. As of the nth iteration, the optimal solution is represented by 〖ys〗_k^n. A random number is assigned by the diffusion factor R, which is equal to R(q), according to the chi- of q degrees of freedom. To get a better global optimal solution, we use the diffusion factor to expand the search area. Both M1 and M2 are normally distributed random variables with values between 0 and 1.

(v) Emigrating behaviour

All flamingos are given a fitness value depending on their ability to emigrate, with the migratory flamingo being the fittest. The flamingos will move to a new location in quest of food when the food supply in their current area is low. Here is a mathematical representation of the flamingo's migratory behaviour:$y_{lk}^{n+1} = y_{lk}^{n} + \sigma(ys_k^n - y_{lk}^n)$ (14)

where y_{lk}^{n+1} characterizes the position of the lth in the n + 1 iteration. Correspondingly, y_{lk}^{n} represents the site of the lth the kth the nth repetition. ys_k^n is the most effective method for improving people's fitness levels. The ideal answer can be found by simulating individual behaviour and a huge search space using.

(vi) Termination

The procedure is ended once the maximum number of iterations has been reached, as the optimal solution has been determined. Algorithm 1 presents the overall procedure for proposed model.

Algorithm 1: Overall Proposed Model

```
procedure POB_Authentication_And_Hiding_Scheme():
   // Preprocessing Phase
   Cover_Share_1, Cover_Share_2 = Compress_And_Encrypt(Original_Image)

   // Embedding Phase
   Embed_Data_And_Attach_Bits(Cover_Share_1)
   Embed_Data_And_Attach_Bits(Cover_Share_2)

   // Recovery Phase
   Trained_Models = Train_Ensemble_Models()
   Recovered_Image = Recover_Image(Trained_Models)

   // Hyperparameter Fine-Tuning
   Fine_Tune_Hyperparameters(Trained_Models)
```

```
    // Data Extraction
    Hidden_Message = Extract_Hidden_Message(Restored_Image)

    // Evaluation Phase
    Evaluate_Performance()

    // Results Analysis
    Analyze_Results()

    // Conclusion
    Conclude_Effectiveness()

function Compress_And_Encrypt(image):
    // Compress the image
    compressed_image = Compress(image)
    // Encrypt the compressed image
    encrypted_image = Encrypt(compressed_image)
    return encrypted_image

function Embed_Data_And_Attach_Bits(image):
    // Apply POB algorithm to embed secret data and attach authentication bits
    // Secret data embedding process
    // Authentication bits attachment process
    return modified_image

function Train_Ensemble_Models():
    // Train ensemble models using machine learning and deep learning techniques
    // Return the trained models
    return trained_models

function Recover_Image(models):
    // Recover the original image using trained ensemble models
    // Apply majority voting among trained models for cover picture recovery
    return recovered_image

function Fine_Tune_Hyperparameters(models):
    // Use Flamingo Optimization Algorithm (FOA) to fine-tune hyperparameters of the ensemble models
    // Hyperparameter tuning process
    return tuned_models

function Extract_Hidden_Message(image):
    // Extract the embedded hidden message from the recovered image
    // Hidden message extraction process
    return hidden_message

function Evaluate_Performance():
    // Evaluate performance metrics such as embedding capacity, PSNR, rate, SSIM, and security measures
    // Performance evaluation process
```

```
function Analyze_Results():
   // Analyze experimental results to determine effectiveness and superiority of the scheme
   // Results analysis process

function Conclude_Effectiveness():
   // Conclude that the proposed scheme achieves high embedding capacity, PSNR, rate, and low SSIM
   // Verify superiority in data embedding and recovery
   // Ensure robust security throughout the process
```

4. Results and Discussion

A PC with an Intel(R) Core running at 2.40 GHz and 8.00 $GB\ of\ RAM$ was used to conduct the experimental study. This OS was designed for 64-bit architecture. A software package called MATLAB, version R2019a, was used to conduct the simulations.

4.1. Validation analysis of Proposed model

All the four datasets are considered for validation, where different images are tested for PSNR and SSIM model with existing techniques that is shown in Table 1 and 2.

Table 1 The hiddenness PSNR (dB) of four test imageries

Images	IBBE [19]	BDBE [21]	Proposed
Breast cancer	49.8664	45.8265	43.7356
Skin cancer	49.8901	44.8262	42.7351
Fundoscopy	49.9898	45.4018	44.9874
PID	50.0839	47.7204	45.7654

In above Table 1 represent that the PSNR of four test images. In the analysis of Breast cancer 49.8664 cancer, the IBBE (improved binary-block embedding (IBBE) [19] reached PSNR as 45.8265correspondingly. Then the Skin cancer, the IBBE [19] reached PSNR as 49.8901 and BDBE(Bi-directional block encoding (BDBE)) [21] as 44.8262 and then proposed model reached as 42.7351 correspondingly. Then the Fundoscopy cancer, the IBBE [19] reached PSNR as 49.9898 and BDBE [21] as 45.4018 and then proposed model reached as 44.9874 correspondingly. Then the PID cancer, the IBBE [19] reached PSNR as 50.0839 and BDBE [21] as 47.7204 and then proposed model reached as 45.7654 correspondingly.

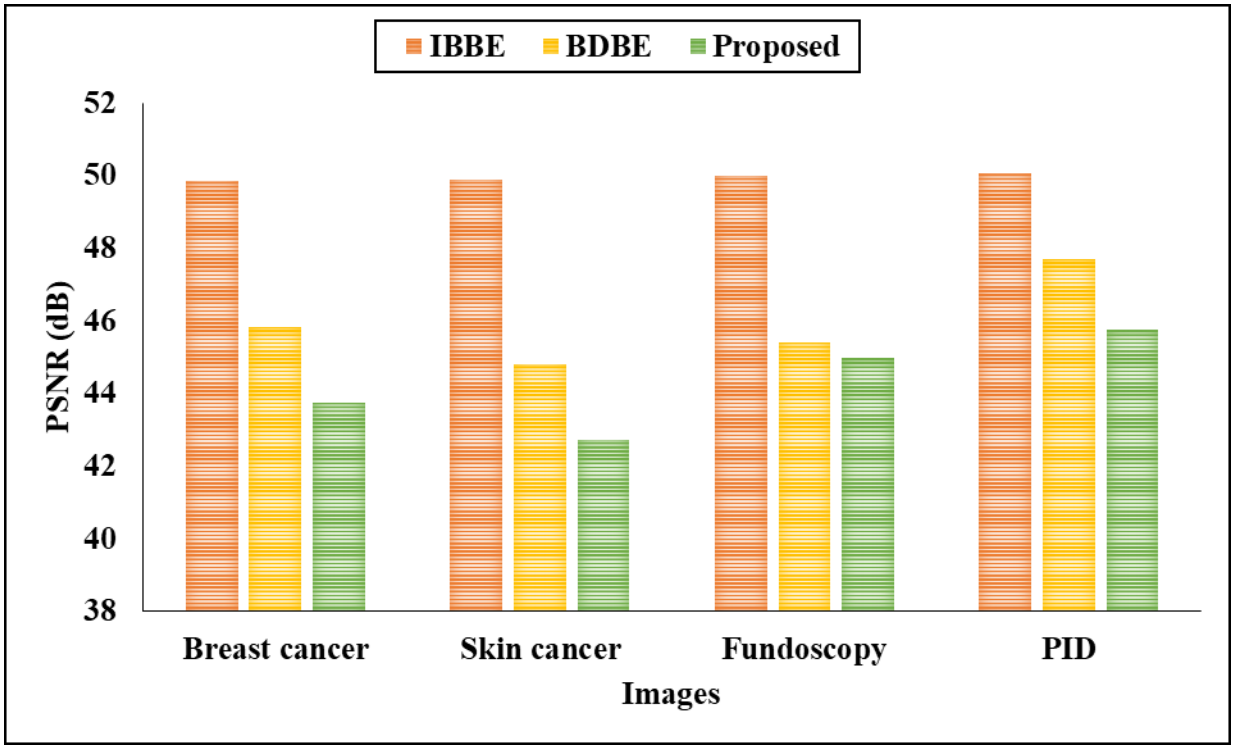

Figure 2: PSNR Analysis

Table 2 The hiddenness SSIM of four test imageries.

Images	IBBE [19]	BDBE [21]	Proposed
Breast cancer	0.9854	0.9771	0.9827
Skin cancer	0.9957	0.9957	0.9893
Fundoscopy	0.9870	0.9872	0.9872
PID	0.9859	0.9857	0.9859

In above Table 2 The SSIM of images. In the analysis of Breast cancer the IBBE [19] reached the SSIM as 0.9854 and BDBE [21] as 0.9771 and lastly proposed model reached as 0.9827 correspondingly. Then the Skin cancer the IBBE [19] reached the SSIM as 0.9957 and BDBE [21] as 0.9957 and lastly proposed model reached as 0.9893 correspondingly. Then the Fundoscopy the IBBE [19] reached the SSIM as 0.9870 and BDBE [21] as 0.9872 and lastly proposed model reached as 0.9872 correspondingly. PID the IBBE [19] reached the SSIM as 0.9859 0.9857 the IBBE [19] reached the SSIM as 0.9859 correspondingly.

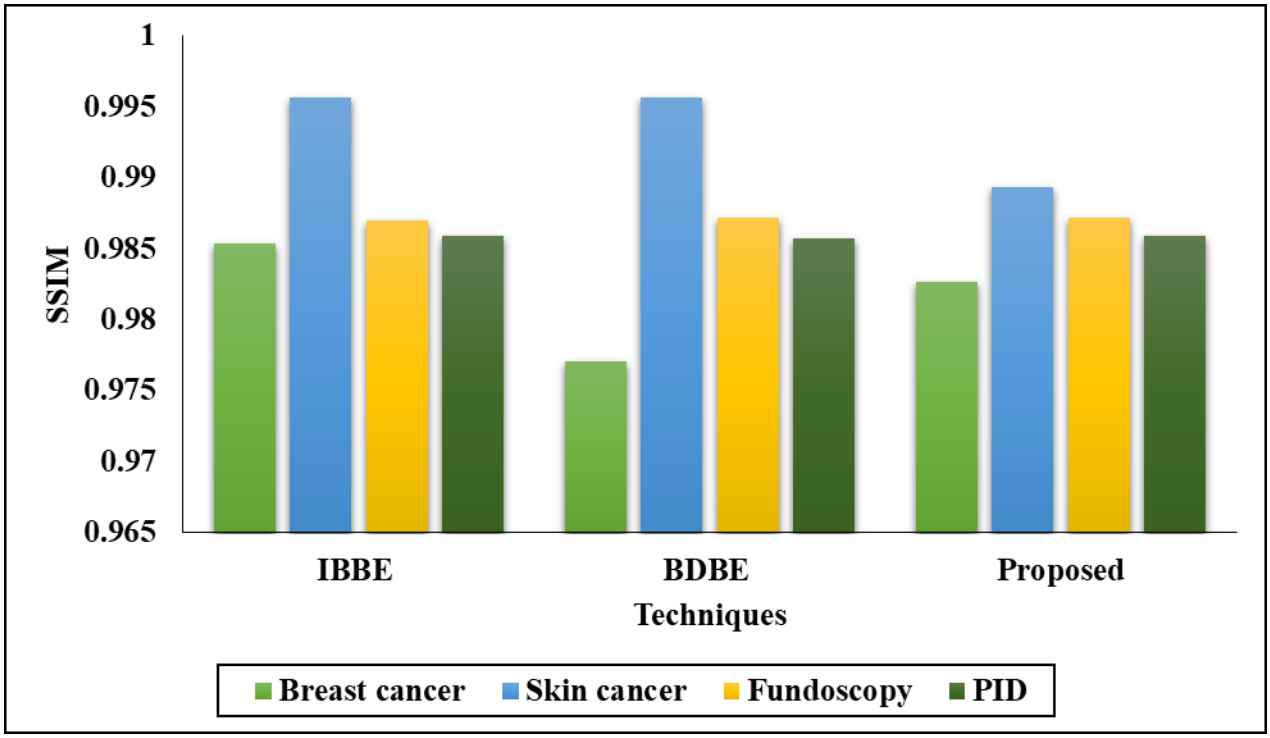

Figure 3: SSIM Analysis

Table 3 Comparison of NC values among existing with projected arrangement under JPEG Compression attacks.

Image	Technique	20	30	40	50	60	70	80
Breast cancer	IBBE [19]	0.7304	0.6728	0.6659	0.7166	0.7350	0.7650	0.9124
	BDBE [21]	0.7797	0.8840	0.9558	0.9824	0.9945	0.9979	0.9994
	Proposed	0.7812	0.8952	0.9623	0.9854	0.9956	0.9981	0.9999
Skin cancer	IBBE [19]	0.8041	0.6544	0.7097	0.6659	0.7857	0.8341	0.9539
	BDBE [21]	0.7696	0.8976	0.9568	0.9774	0.9860	0.9890	0.9920
	Proposed	0.7969	0.9134	0.9674	0.9845	0.9885	0.9901	0.9975
Fundoscopy	IBBE [19]	0.3917	0.6774	0.7212	0.6959	0.7419	0.8111	0.9286
	BDBE [21]	0.7987	0.9253	0.9742	0.9927	0.9970	0.9984	0.9997
	Proposed	0.8065	0.9562	0.9876	0.9986	0.9990	0.9992	0.9998
PID	IBBE [19]	0.5991	0.6567	0.7211	0.6959	0.7512	0.7719	0.9355
	BDBE [21]	0.8632	0.9467	0.9774	0.9884	0.9948	0.9979	0.9988
	Proposed	0.8976	0.9531	0.9865	0.9895	0.9965	0.9993	1

Under JPEG compression attacks, Table 3 demonstrations the judgement of NC values between the existing strategy and the suggested one.The following values were obtained for the NC rates: 20 NC rate = 0.7304, 30 NC value = 0.6728, 30 NC value = 0.6659, 30 NC value = 0.7166, 30 NC value = 0.7350, 30 NC value = 0.7650, and 30 NC value = 0.9124, according to the IBBE [19] model. After that, the BDBE [21] model got to 20NC rate = 0.7797, 30NC value = 0.8840, 30NC value = 0.9558, 30NC value = 0.9824, 0.9945, 30NC

value = 0.9979, and 30NC value = 0.9994 in that order. Next, the proposed model achieved the following 20NC rates: 0.7812, 0.8952, 0.9623, 0.9854, 0..9956, and 0.9981, 0.9999, respectively. Next, the IBBE [19] model arrived to the following 20NC rates: 0.8041, 0.6544, 0.7097, 0.6659, 0.7857, 0.8341, and 0.9539, in that order. The BDBE [21] model then arrived at the following 20NC rates: 0.7696, 0.8976, 0.9568, 0.9774, 0.9860, 0.9890, and 0.9920, in that order. The proposed model obtained the following values at 20 and 30 NC rates: 0.7969, 0.9134, 0.9674, 0.9845, 0.9885, and 0.9901. The 20NC rate was then 0.3917, 0.6774, 0.7212, and 0.6959 according to the IBBE [19] model, and the 40NC value was 0.7419, 0.8111, and 0.9286. Afterwards, the BDBE [21] model achieved the following 30NC rates: 0.7987, 0.9253, 0.9742, 0.9927, 0.9970, 0.9984, and 0.9997, in that order. Afterwards, the estimated values for the following parameters are given: 0.8065, 0.9562, 0.9876, 0.9986, and 0.9998 for the 30 NC. After that, the IBBE [19] model yielded the following results: 20 NC rate = 0.5991; 30 NC value = 0.7512; 30 NC value = 0.9719; and 30 NC value = 0.9355, in that order. Next, the BDBE [21] model arrived to the following 20NC rates: 0.8632, 0.9467, 0.9774, 0.9884; 30NC values: 0.9948, 0..9979, and 0.9988, respectively. This leads to the following suggested values: 0.8976, 0.9865, 0.9895, 0.9965, and 0.9993 for the 30 NC.

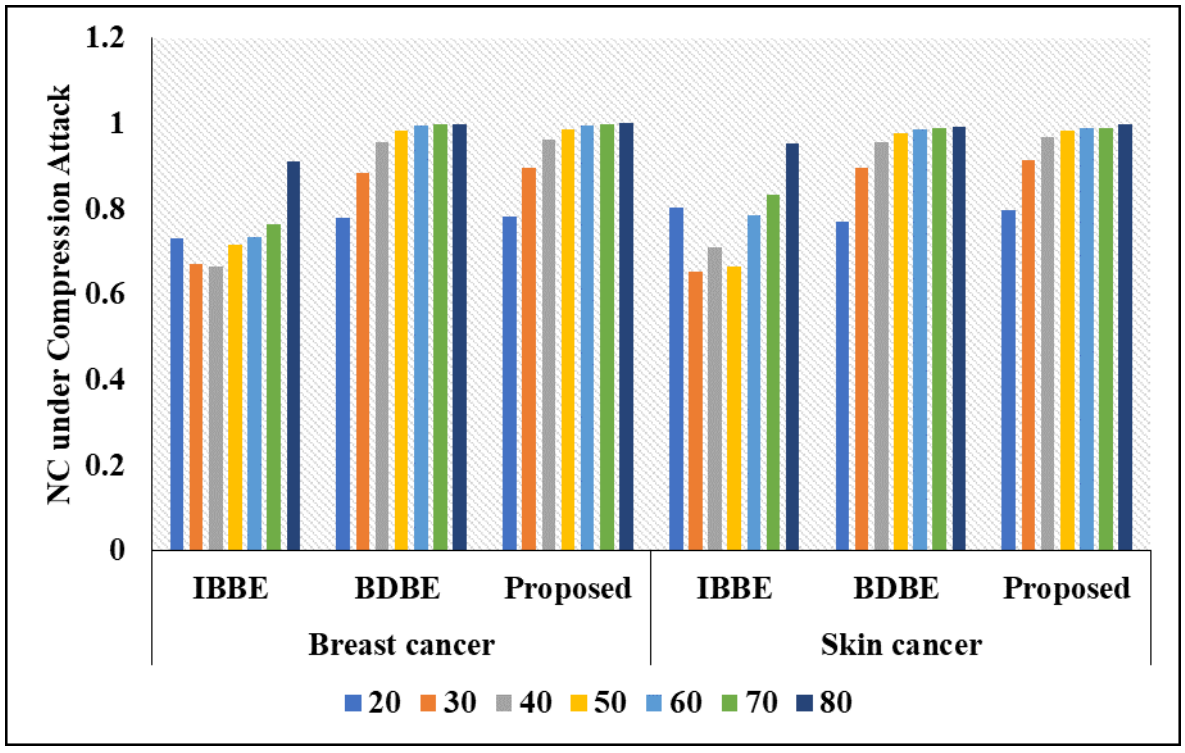

Figure 4: NC values on two datasets

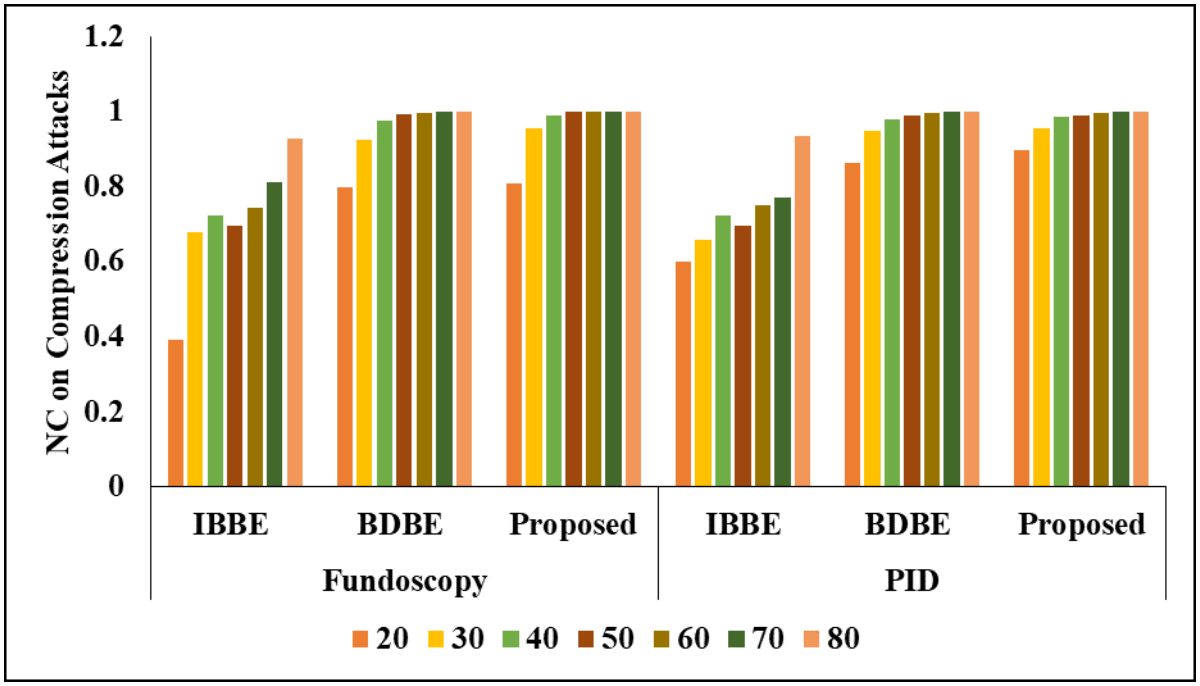

Figure 5: Validation analysis of NC on proposed model

5. Conclusion

Techniques for the safe transfer of sensitive patient medical data are necessary for e-health applications. The recommended RDH mechanism uses deep learning for recovery and double embedding for encryption, making it possible to conceal patient electronic health records in the cover picture. This research proposes a safe patient data transfer system. This research presents a novel method for medical picture authentication and concealment that makes use of a double POB system. A POB number scheme is utilised in the process to integrate the secret bits. After the authentication step of recovery is complete, the finish recovering the original image, provided that the authentication step was not tampered with. The recovery process begins with the data that has been entered in many times in the event that authentication fails. Deep learning and machine learning are implemented using FOA-tuned parameters. Without compromising the image's perceptual quality, the experimental findings demonstrated that the approach presented in this study substantially increases the embedding capability. Data embedding and lossless recovery are two areas where it excels in comparison to competing approaches. Crucial for their use, the bit method also successfully protects very sensitive medical images. The time overhead is marginally larger when employing a double POB scheme since compression and re-encryption are executed simultaneously. Reducing the time cost of employing the double POB purpose will be our primary focus in future development.

References

[1] Chen, K., Guan, Q., Zhang, W., & Yu, N. (2022). Reversible data hiding in encrypted images based on binary symmetric channel model and polar code. IEEE Transactions on Dependable and Secure Computing.

[2] Manikandan, V. M., & Zhang, Y. D. (2022). An adaptive pixel mapping based approach for reversible data hiding in encrypted images. Signal Processing: Image Communication, 105, 116690.

[3] Yu, C., Zhang, X., Qin, C., & Tang, Z. (2023). Reversible data hiding in encrypted images with secret sharing and hybrid coding. IEEE Transactions on Circuits and Systems for Video Technology.

[4] Tsai, C. S., Zhang, Y. S., & Weng, C. Y. (2022). Separable reversible data hiding in encrypted images based on paillier cryptosystem. Multimedia Tools and Applications, 81(13), 18807-18827.

[5] Meng, L., Liu, L., Wang, X., & Tian, G. (2022). Reversible data hiding in encrypted images based on IWT and chaotic system. Multimedia Tools and Applications, 81(12), 16833-16861.

[6] Wang, Y., Xiong, G., & He, W. (2023). High-capacity reversible data hiding in encrypted images based on pixel-value-ordering and histogram shifting. Expert Systems with Applications, 211, 118600.

[7] Hua, Z., Wang, Y., Yi, S., Zheng, Y., Liu, X., Chen, Y., & Zhang, X. (2022). Matrix-based secret sharing for reversible data hiding in encrypted images. IEEE Transactions on Dependable and Secure Computing.

[8] Weng, C. Y., & Yang, C. H. (2023). Reversible data hiding in encrypted image using multiple data-hiders sharing algorithm. Entropy, 25(2), 209.

[9] Qiu, Y., Ying, Q., Yang, Y., Zeng, H., Li, S., & Qian, Z. (2022). High-capacity framework for reversible data hiding in encrypted image using pixel prediction and entropy encoding. IEEE Transactions on Circuits and Systems for Video Technology, 32(9), 5874-5887.

[10] Hua, Z., Wang, Y., Yi, S., Zhou, Y., & Jia, X. (2022). Reversible data hiding in encrypted images using cipher-feedback secret sharing. IEEE Transactions on Circuits and Systems for Video Technology, 32(8), 4968-4982.

[11] Yang, C. H., Weng, C. Y., & Chen, J. Y. (2022). High-fidelity reversible data hiding in encrypted image based on difference-preserving encryption. Soft Computing, 26(4), 1727-1742.

[12] Gao, K., Horng, J. H., & Chang, C. C. (2022). High-capacity reversible data hiding in encrypted images based on adaptive block encoding. Journal of Visual Communication and Image Representation, 84, 103481.

[13] Gunapriya, B., Rajesh, T., Thirumalraj, A., & B, M. (2023). LW-CNN-based extraction with optimized encoder-decoder model for detection of diabetic retinopathy. Journal of Autonomous Intelligence, 7(3). doi:http://dx.doi.org/10.32629/jai.v7i3.1095

[14] Yu, M., Yao, H., & Qin, C. (2022). Reversible data hiding in encrypted images without additional information transmission. Signal Processing: Image Communication, 105, 116696.

[15] Liu, Y.; You, Z.; Gao, T. Lossless image hierarchical recovery based on POB number system. Signal Process. 2020, 167, 107293.

[16]. Li, Q.; Fu, Y.; Zhang, Z.; Fofanah, A.J.; Gao, T. Medical images lossless recovery based on POB number system and image compression. Multimed. Tools Appl. 2022, 81, 11415–11440.

[17] Qu, L., Li, M., & Chen, P. (2024). Reversible data hiding in encrypted image with secure multi-party for telemedicine applications. Biomedical Signal Processing and Control, 93, 106209.

[18] Gao, H., Zhang, X., & Gao, T. (2024). Hierarchical reversible data hiding in encrypted images based on multiple linear regressions and multiple bits prediction. Multimedia Tools and Applications, 83(3), 8757-8783.

[19] Fu, Z., Chai, X., Tang, Z., He, X., Gan, Z., & Cao, G. (2024). Adaptive embedding combining LBE and IBBE for high-capacity reversible data hiding in encrypted images. Signal Processing, 216, 109299.

[20] KIKUCHI, N., & IMAIZUMI, S. (2024). Reversible Data Hiding for OpenEXR Images in Compressible Encrypted Domain. Bulletin of the Society of Photography and Imaging of Japan, 34(1), 1-8.

[21] Liu, X., Hua, Z., Yi, S., Zhang, Y., & Zhou, Y. (2024). Bi-directional Block Encoding for Reversible Data Hiding over Encrypted Images. ACM Transactions on Multimedia Computing, Communications and Applications, 20(5), 1-23.

[22] Shi, H., Zhou, Z., Qin, J., Sun, H., & Ren, Y. (2024). A separable privacy-preserving technique based on reversible medical data hiding in plaintext encrypted images using neural network. Multimedia Tools and Applications, 1-26.

[23] Zhang, T., Zhang, J., Zou, Y., & Zhang, Y. (2024, January). High Capacity Reversible Data Hiding in Encrypted Images Based on Pixel Value Preprocessing and Block Classification. In International Conference on Multimedia Modeling (pp. 14-27). Cham: Springer Nature Switzerland.

[24] S. D. Karthik, Maggie, "APTOS 2019 Blindness Detection," 2019, https://kaggle.com/competitions/aptos2019-blindness-detection.

[25] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," Sci. Data, vol. 5, no. 1, pp. 1–9, 2018.

[26] M. Mustra, K. Delac, and M. Grgic, ''Overview of the DICOM standard,'' in Proc. 50th Int. Symp. (ELMAR), vol. 1, Sep. 2008, pp. 39–44

[27] Sreekumar, A.; Sundar, S.B. An efficient secret sharing scheme for n out of n scheme using POB-number system. Hack 2009, 1, 33–37