Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 12, No. 3, 2024, pp. 585-598
DOI:10.22034/cmde.2023.59134.2512

Convolutional neural network-based high capacity predictor estimation for reversible data embedding in cloud network

Prasad C. N* **and R Suchithra**
Research Scholar, Department of Computer Science, Chirashree Institute Of Research and Development(CIRD), University of Mysore, Karnataka, India.

Abstract

This paper proposes a reversible data embedding algorithm in encrypted images of cloud storage where the embedding was performed by detecting a predictor that provides a maximum embedding rate. Initially, the scheme generates trail data which are embedded using the prediction error expansion in the encrypted training images to obtain the embedding rate of a predictor. The process is repeated for different predictors from which the predictor that offers the maximum embedding rate is estimated. Using the estimated predictor as the label the Convolutional neural network (CNN) model is trained with the encrypted images. The trained CNN model is used to estimate the best predictor that provides the maximum embedding rate. The estimation of the best predictor from the test image does not use the trail data embedding process. The evaluation of proposed reversible data hiding uses the datasets namely BossBase and BOWS-2 with the metrics such as embedding rate, SSIM, and PSNR. The proposed predictor classification was evaluated with the metrics such as classification accuracy, recall, and precision. The predictor classification provides an accuracy, recall, and precision of 92.63%, 91.73%, and 90.13% respectively. The reversible data hiding using the proposed predictor selection approach provides an embedding rate of 1.955 bpp with a PSNR and SSIM of 55.58dB and 0.9913 respectively.

Keywords. Reversible data hiding, Image encryption, Prediction error expansion, Convolutional neural network, Embedding capacity.
2010 Mathematics Subject Classification. 68T07, 68P25, 68P99.

1. Introduction

Due to the increased capacity in storage, cloud computing [28] has attracted several users and researchers. However, there exist several challenges in cloud storage that include authentication, confidentiality, and integrity. In order to handle these challenges data encryption [13], and data hiding [22] are used. Data encryption is used to preserve the plain image content of the image by modifying the plain image content to the cipher image. The data embedding modifies the pixel content for holding the secret content. The commonly used reversible data hiding approach is derived from the schemes such as integer transform [25], histogram shifting [21], difference expansion [7], and prediction error expansion [8]. In the non-reversible approach [31], the original image cannot be reconstructed after the extraction of the hidden data, but in the reversible data hiding approach [27] both the hidden data and the carrier image can be reconstructed without any loss. For preserving, the privacy content of the image the user will typically encrypt the image before uploading it to the cloud. Partial or full encryption can be used to prevent the unauthorized person from accessing the actual image content. Block-based MSB plain rearrangement approach [4] estimates the highly compressible bit streams from the MSB planes of the image. Parametric binary tree labeling [29] was proposed by the author Yi et al. where the data is embedded in small blocks that are encrypted. Two data embedding schemes were proposed by Zhang et al. [30], where the images are encrypted based on the homomorphic and probabilistic properties using a public cryptosystem. The first approach is reversible, whereas the second approach is non-reversible. In the reversible approach, the histogram is initially shrunk and the homomorphic cryptosystem is used to encrypt the

Received: 07 November 2023 ; Accepted: 06 January 2024.
∗ Corresponding author. Email: prasad_achar@sju.edu.in.

resultant image. In the non-reversible approach, the image is directly encrypted and the embedding is performed using multilayer wet paper coding.

A sufficient amount of carrier for embedding the data was provided by the patch-level representation [3], where the original image is converted to sparse coefficients based on a dictionary. However, this approach requires an additional reversible data-hiding mechanism for embedding the residual errors. The process of preserving the privacy content of the image can be classified into two broad categories namely Vacating a room after encryption (VRAE) and reserving a room before encryption (RRBE). In the RRBE approach [9, 14], the owner of the image needs to preprocess the image such that encryption needs to be done after leaving some space for data embedding. But in the VRAE approach [12, 17, 18] the owner can encrypt the image without any other pre-processing and can upload it to the cloud. The cloud can modify the encrypted image to hide the data. A higher payload can be achieved in the RRBE approach than in the VRAE approach, but it requires an additional preprocessing step that creates a burden for the image owner.

The RRBE scheme was first introduced by Ma et al. [9]. This approach was derived from the histogram shifting-based reversible data hiding, where some vacant is created for embedding a few pixel LSBs. The vacant space is kept as it is while the remaining regions are encrypted and uploaded to the cloud. The vacant region is used to hide the secret data and this approach provides an embedding rate of 0.5 bpp. The predictor used in [14] is modified by Puyang et al. [17] where the authors used a second MSB plane along with the first MSB plane for data embedding. This approach provides an average payload of 1.35 bpp. The bit planes are used iteratively from MSB to LSB by Puteaux et al. [16] which provides an average embedding rate of 1.84 bpp. A median edge detector-based predictor was introduced by Hu et al. [5] which is derived from the context adaptive predictor. The performance of the reversible data hiding based on predictor error expansion proposed by Thodi et al. [24] provides a better performance when compared to a different expansion scheme introduced by Tian et al. [6]. The prediction error obtained from the histogram is shifted right and left resulting in several zero and peak points. Thus the combination of histogram shifting and prediction error expansion [26] provides a better capacity. Li et al. [8] grouped the image pixel into two areas namely flat area and rough area, where the flat area offers an embedding capacity of 2 bits per pixel and the rough area offers an embedding rate of 1 bit per pixel. The average of four adjacent pixels is used to estimate the prediction value [20] which improves the quality of the marked image. One-dimensional (1D) prediction error histogram and two-dimensional (2D) prediction error histogram (PEH) were introduced by Ou et al. [11], where the 2D approach provides a better correlation between the prediction errors. The author Zhang et al. [10] combined 1D-PEH and 2D-PEH to obtain an efficient map for data embedding.

The contribution of the paper is as follows. (i). This paper proposes a CNN model-based best predictor detection algorithm that provides a maximum embedding rate (ii). The algorithm uses a trail data generation approach to estimate the predictor that provides a maximum embedding rate (iii). The trained model was deployed in a cloud so that it detects the best predictor for embedding the data without the need for trial data embedding. (iv). Finally, the evaluation of the predictor classification was evaluated using the metrics namely classification accuracy, recall, and precision. The evaluation of the data embedding algorithm was also done using the metrics such as embedding rate, PSNR, and SSIM with the datasets namely BossBase and BOWS-2.

The remaining sections of the paper are constructed as below. Section 2 depicts the proposed reversible data embedding algorithm, section 3 discusses the experimental results and analysis of the work. Finally, section 4 concludes the work.

2. Proposed methods

The proposed reversible data embedding in a cloud network includes three major processes such as (2.1) three-level encryption and decryption (2.2) CNN training for predictor class estimation (2.3) Reversible data embedding using classified predictor.

2.1. **Three-level encryption and decryption.** The image that is to be uploaded to the cloud is encrypted by the user using the three-level encryption algorithm that uses the key 'K'. Let $u \times v$ be the size of the image I_{i}. Initially, the image is subdivided into 7×7 blocks. Therefore the number of non-overlapping blocks is expressed as

$$N_b = \frac{uv}{49}. \tag{2.1}$$

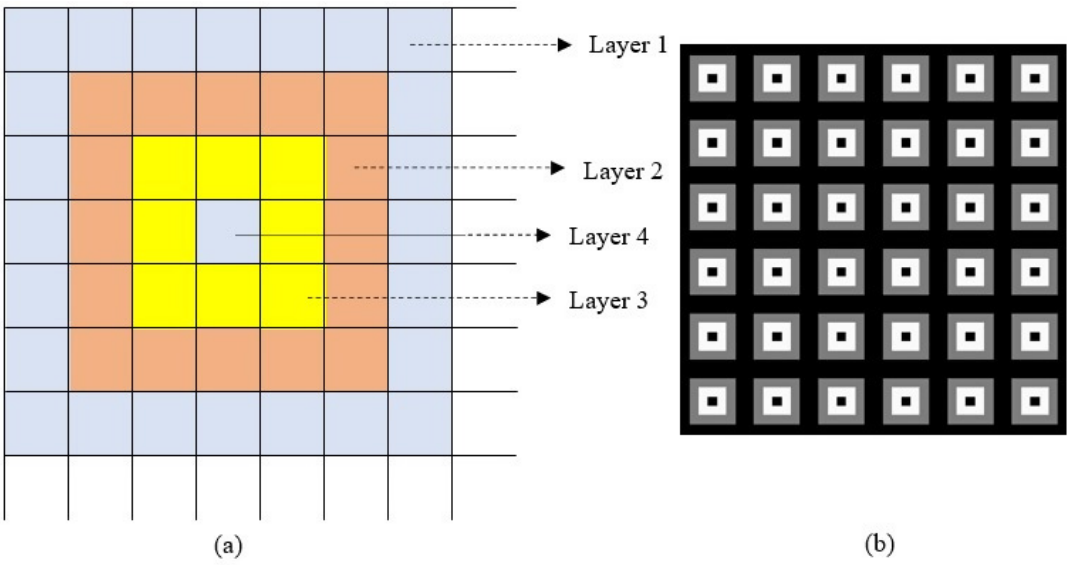

FIGURE 1. Representation of four layers in a 7×7 sub-block (a) Layers in Single sub-block (b) Layers in multiple sub-blocks.

The sub-block contains four layers namely layer 1, layer 2, layer 3, and layer 4 as illustrated in Figure 1(a). Layer 1 and Layer 4 of the complete image are encrypted by a global encryption algorithm. In a 7×7 sub-image, the number of pixels constitutes layer 1 and layer 4 is 25 which is subjected to global encryption. The key K is grouped into three keys namely as $K = \{K_1, K_2, K_3\}$. The number of pixels that constitute layer 2 and layer 3 is 24. Therefore the total number of pixels in the image I_{i} that constitute layer 1 and layer 4 is

$$N_{14} = 25N_b = \frac{25uv}{49}. \tag{2.2}$$

Figure 1(b) illustrates the different layers that are estimated throughout the image. The position of layer 2 and Layer 3 between the blocks are scrambled using the key K_2. The total number of pixels in the image I_{i} that constitute layer 2 and layer 3 is

$$N_{23} = 24N_b = \frac{24uv}{49}. \tag{2.3}$$

Thus the total number of pixels in the image and the number of pixels in layers are related by

$$N_T = N_{14} + N_{23}. \tag{2.4}$$

Using the key K_1, N_14 number of random integer sequences are generated. Based on the generated random sequence the pixels in layer 1 and layer 4 are scrambled to obtain the global encrypted layers for layers 1 and 4. Similarly using the key K_2, N_23 number of random sequences is generated which is used to scramble the position of layers 2 and 3 in the blocks with other blocks. The key K_3 is used to generate N_{23} number of random sequences such that the random sequence is between 1 to 8. The random sequence that is generated using the key K_3 is used to rotate the layers as depicted in Figure 2. For example, if $R_3 = 1$, the pixels in layers 2 and 3 are retained with the same position shown in Figure 2(a). If $R_3 = 4$, the pixels in layers 2 and 3 are changed (rotated) as illustrated in Figure 2(d). Since layer 2 and layer 3 undergo only the rotational change, these two layers are used to embed the data.

2.2. **CNN training for predictor class estimation.** Figure 3 depicts the block diagram for the estimation of the predictor class for training the Convolutional neural network. Let $I_1, I_2, \ldots I_{N_t}$ represents N_2 number of training

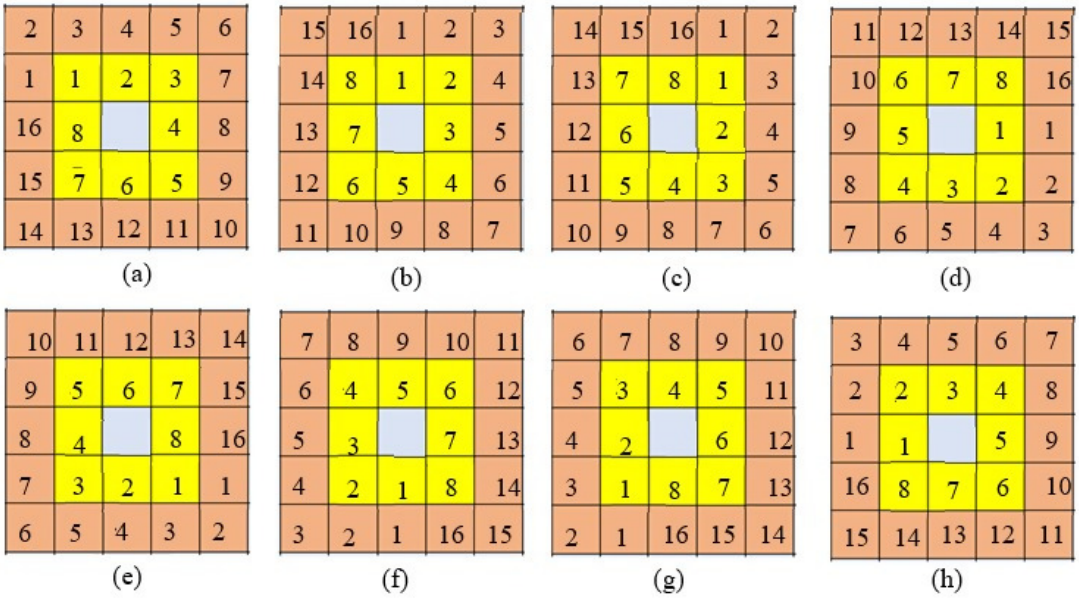

FIGURE 2. Rotation of layers 2 and 3 for different values of R_3 (a) $R_3 = 1$ (b) $R_3 = 2$, (c) $R_3 = 3$ (d) $R_3 = 4$ (e) $R_3 = 5$ (f) $R_3 = 6$ (g) $R_3 = 7$ (h) $R_3 = 8$.

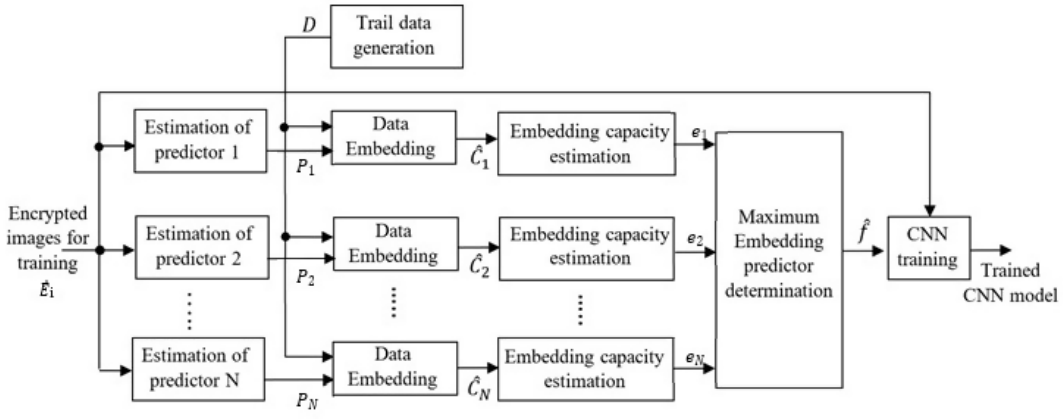

FIGURE 3. Estimation of predictor class for training the Convolutional neural network.

images generally represented as

$$I_{\mathrm{i}} = [I_1, I_2, \ldots I_{N_t}] \quad \mathrm{i} = 1, 2, \ldots N_t. \tag{2.5}$$

The images I_i is encrypted to obtain the encrypted image E_i. Let $P_1, P_2, \ldots . P_N$ represents N different predictor generally represented as

$$P_{\mathrm{j}} = [P_1, P_2, \ldots P_N] \quad \mathrm{j} = 1, 2, \ldots N. \tag{2.6}$$

Let D represent the trail data used in the embedding process. The data D is embedded in the image E_i using the predictor P_j to obtain the marked encrypted image. Let $C_{\mathrm{i}} = \{C_1, C_2 \ldots C_N\}$ be the marked encrypted image. The

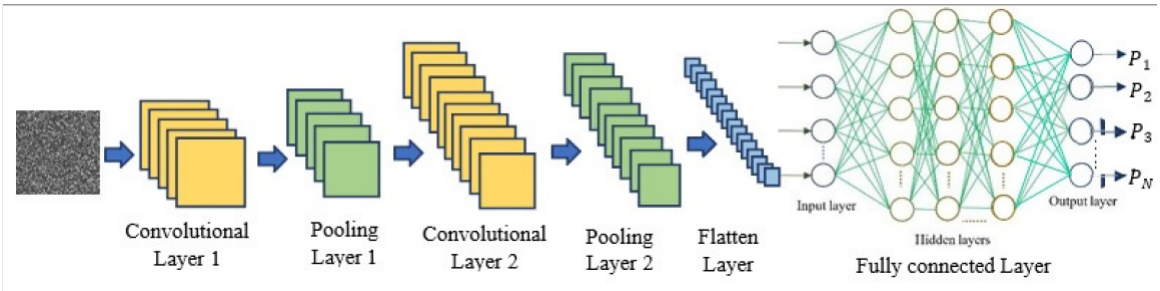

FIGURE 4. Architecture of Convolutional neural network in training the best predictor class.

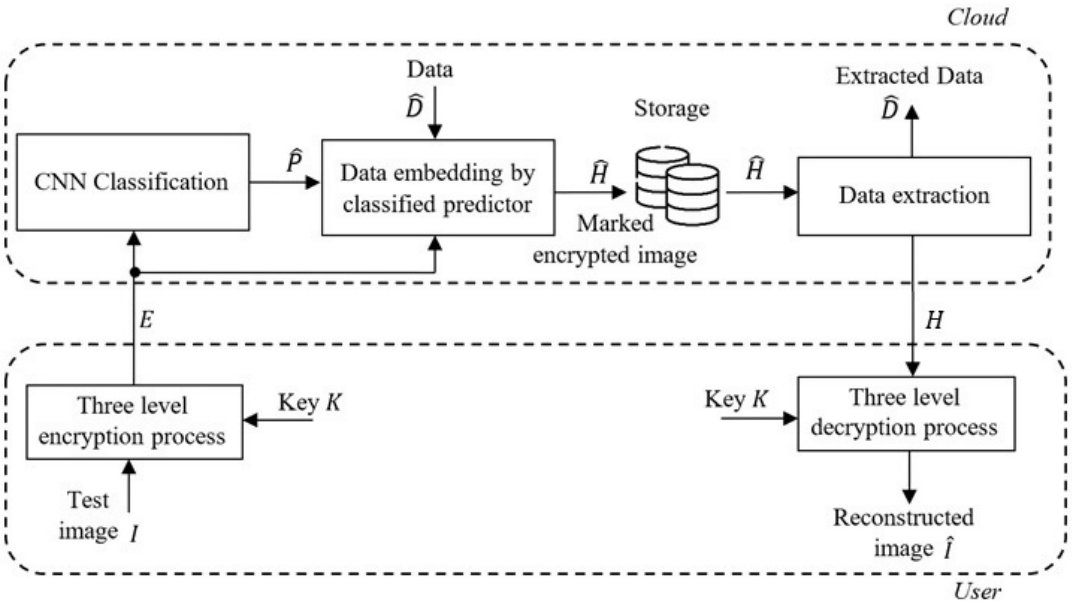

FIGURE 5. Block diagram representation of data embedding and extraction process in the cloud.

embedding rate e_i is then evaluated on the image E_i that corresponds to the predictor P_j. The predictor $\hat{f}$ that offers the maximum embedding is estimated using the relation

$$\hat{f} = \arg\max_{P_i} (e_j) . \tag{2.7}$$

Using the predictor class $\hat{f}$ as the label and the image E_i as the training image, the CNN is trained. The architecture of the CNN is depicted in Figure 4. The CNN architecture [19, 23] here uses two sections of convolutional layer and max pooling layer. The size of convolutional layer 1 and max-pooling layer 1 are $128 \times 128 \times 8$ and $64 \times 64 \times 8$ respectively. The size of the convolutional layer 2 and max-pooling layer 2 has the size of $64 \times 64 \times 16$ and $32 \times 32 \times 16$ respectively. The two sections of the convolutional layer and max-pooling layers are followed by a flattening layer and a fully convolutional layer. The fully convolutional layer has 16384 input neurons followed by three hidden layers each having 16386 neurons followed by the output layer with N classes.

2.3. **Reversible data embedding using classified predictor.** Figure 5 depicts the representation of data embedding and extraction process in the cloud network. Initially, the test image I (the image that is being uploaded to the cloud by the user) is first encrypted using the three-level encryption process using the key K. The encrypted image E

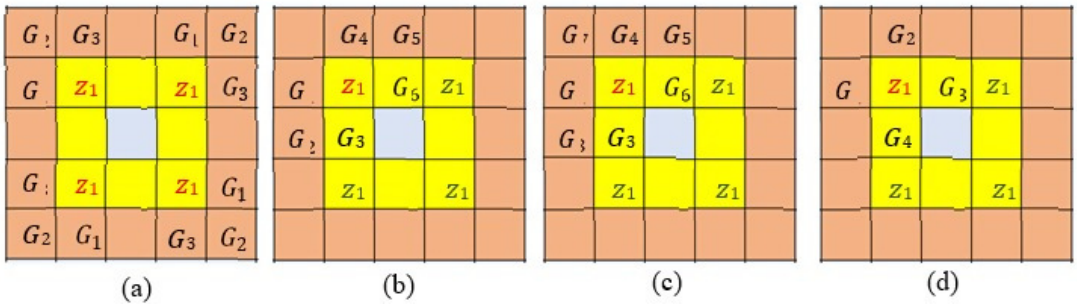

FIGURE 6. Representation of four different types of predictor (a) Predictor 1 (P_1), (b) Predictor 2 (P_2), (c) Predictor 3 (P_3), (c) Predictor 4 (P_4) (The predictor represented as Z_1 in green color can be estimated as represented by Z_1 in red color).

is then uploaded to the cloud by the owner. The trained CNN model that was deployed in the cloud will estimate the best predictor that provides a maximum embedding rate. Let $\hat{P}$ be the best predictor classified by the CNN classifier. The data $\hat{D}$ is then embedded on the encrypted image E using the predictor $\hat{P}$, to obtain the marked encrypted image $\hat{H}$ which is then stored in the cloud. In the extraction phase, the data $\hat{D}$ is extracted from the marked encrypted image $\hat{H}$ and also the encrypted image E is reconstructed. Let the reconstructed image be represented as H. Using the same encryption key K, the three-level decryption process is applied to the reconstructed encrypted image H to reconstruct the actual image I.

The predictor $\hat{P}$ can be estimated from the neighborhood of the encrypted image E using any one of the following predictors P_1, P_2, P_3 and P_4 as $\hat{P} = P_1, P_2, P_3, P_4$.

(i). Predictor 1

$$P_1 = \begin{cases} \min(G_3, G_2)\, G_1 \geq \max(G_3, G_2), \\ \max(G_3, G_2)\, G_1 \leq \min(G_3, G_2), \\ G_3 + G_2 - G_1 \text{ otherwise}. \end{cases} \tag{2.8}$$

(ii). Predictor 2

$$P_2 = \begin{cases} \min(\alpha_1, \alpha_2) \quad \alpha_1 \geq \alpha_2, \\ \max(\alpha_1, \alpha_2) \text{ otherwise}. \end{cases} \tag{2.9}$$

Where $\alpha_1 = \frac{(G_1+G_2+G_3)}{3}\Gamma$ and $\alpha_2 = \frac{(G_4+G_5+G_6)}{3}\Gamma$.

(iii). Predictor 3

$$P_3 = \begin{cases} \min(\alpha_1, \alpha_2)\, \alpha_1 \geq \max(G_7, \alpha_2), \\ \max(\alpha_1, \alpha_2)\, \alpha_1 \leq \min(G_7, \alpha_2), \\ G_7 + \alpha_1 - \alpha_2 \text{ otherwise}. \end{cases} \tag{2.10}$$

(iv). Predictor 4

$$P_4 = \frac{(G_1 + G_2 + G_3 + G_4)}{4}\Gamma. \tag{2.11}$$

The neighborhood of Z_1 is used to estimate the predictor $\hat{P} = P_1$, $\hat{P} = P_2$, $\hat{P} = P_3$ and $\hat{P} = P_4$ is illustrated in Figures 6(a), 6(b), 6(c), and 6(d) respectively. The embedding and extraction are done as traditional prediction error

expansion [8] embed and extracts the data. The prediction error between the original pixel Z_1 and $\hat{P}$ can be estimated as

$$r = z_1 - \hat{P}. \tag{2.12}$$

The prediction error r is expanded to obtain the

$$\hat{r} = \begin{cases} 2r - \Delta \text{ if } r \in (-\infty, -\Delta), \\ 2r + \Delta \text{ if } r \in (\Delta + \infty), \\ 2r + \widehat{D} \text{ if } r \in (-\Delta, \Delta). \end{cases} \tag{2.13}$$

Where $\hat{D}$ is the data that is to be embedded and the parameter Δ decides the embedding capacity. The marked encrypted pixel can be estimated from the predicted value and the expanded error using the relation,

$$\widehat{H} = \hat{P} + \hat{r}. \tag{2.14}$$

The next section discusses the experimental results of the work.

3. Experimental results

The proposed approach uses two datasets namely Bossbase [1] and Bows-2 [2] dataset each having 10000 8-bit grayscale images where each image has a size of 512 × 512. A few of the sample images from BOWS-2 and Bossbase dataset is provided in Figure 7. Out of 10000 images, two class of data is constructed, such that in Class 1, 60% of images (6000 images) are used in training and 40% of images (4000 images) are used in testing. In Class 2, 70% of images (7000 images) are used in training and 30% of images (3000 images) are used for testing. The algorithm was implemented using MATLAB 2018a. The performance of CNN in predictor classification can be evaluated using the metrics such as accuracy, recall, and precision which can be evaluated as follows.

$$\text{Accuracy } = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}, \tag{3.1}$$

$$\text{Recall } = \frac{t_p}{t_p + f_n}, \tag{3.2}$$

$$\text{Precision } = \frac{t_p}{t_p + f_p}. \tag{3.3}$$

The performance of the reversible data embedding algorithm can be evaluated with the metrics such as peak signal-to-noise ratio ($PSNR$), embedding rate (ER), and structural similarity index measurement ($SSIM$) as follows,

$$PSNR = 10 \log_{10} \frac{255^2}{s}, \tag{3.4}$$

where s is the mean square error estimated by

$$\text{s} = \frac{1}{u \times v} \sum_{x=1}^{u} \sum_{y=1}^{v} [E(x, y) - \hat{H}(x, y)], \tag{3.5}$$

where $E(x, y)$ is the encrypted image and $\hat{H}(x, y)$ is the marked encrypted image. The embedding rate (ER) can be estimated using the relation

$$\text{ER } = \frac{L}{u \times v}, \tag{3.6}$$

where $u \times v$ represents the size of the image and L represents embedding capacity. $SSIM$ can be estimated using the relation,

$$\text{SSI}(E, \widehat{H}) = \frac{\left(2\sigma_E \widehat{H} + \alpha_1\right)\left(2\mu E \mu \widehat{H} + \alpha_2\right)}{\left(\sigma^2 E + \sigma^2 \widehat{H} + \alpha_1\right)\left(\mu^2 E + \mu^2 \widehat{H} + \alpha_2\right)}, \tag{3.7}$$

FIGURE 7. Sample images from BOWS-2 dataset and Bossbase dataset.

TABLE 1. Performance evaluation of Predictor classification for the two datasets.

Metrics	BossBase dataset		BOWS-2 dataset	
	Class 1	Class 2	Class 1	Class 2
Accuracy (%)	88.36	92.63	87.42	91.24
Recall (%)	87.12	91.73	88.19	90.63
Precision (%)	89.74	90.13	88.18	89.17

μE represent the mean of the encrypted image E, $\mu\hat{H}$ represents the mean of the marked encrypted image $\hat{H}$, σ^2 represents the variance of E, $\sigma^2\hat{H}$ represents the variance of $\hat{H}$, $\sigma E\hat{H}$ represents the covariance of E and $\hat{H}$, α_1 and α_2 are constants.

Figure 8 depicts the experimental results from the BossBase dataset, where the level 1 encrypted image is depicted in Figure 8(b). It shows the encryption of layer 4 and layer 1. The level 2 encrypted image also scrambles layers 2 and 3 of one neighborhood with other neighbourhood as illustrated in Figure 8(c). The level 3 encrypted image also rotates the layers 2 and 3 along with level 1 and 2 encryption as illustrated in Figure 8(d). Figure 8(e) shows the marked encrypted image where the predictor used in embedding the data is decided by the classification result provided by the CNN classifier. Figure 9 depicts the experimental results from the BOWS-2 dataset.

Table 1 shows the performance of the proposed predictor classification process. As the number of training images is increased the performance increases in both the BossBase and BOWS-2 dataset. With 60% of training images (Class 1), the proposed predictor classification process provides an accuracy of 88.36% and 87.42% respectively. With 70% of training images (Class 2), the accuracy increases by 4.27% and 3.82% respectively. The comparison of accuracy, recall, and precision for the two datasets with two classes is depicted in Figure 10. The maximum performance is obtained for the BossBase dataset with Class 2 testing. The accuracy, recall, and precision were estimated as 92.63%, 91.73%, and 90.13% respectively.

Figure 11 depicts the ROC curve obtained while training the CNN for predictor classification on the datasets BOWS-2 and BossBase. The maximum area under the ROC (AUC) is obtained while training the CNN using the BossBase dataset when 70% of the image is used in training. In the class 1 category, the AUC for Bows-2 and BossBase datasets is estimated as 0.8192 and 0.8654 respectively. However, increasing the training images to 70% increases the AUC for BOWS-2 and BossBase datasets as 0.9179 and 0.9476 respectively. Figures 12(a) and 12(b) show the confusion matrix obtained during the testing phase of CNN for BossBase and Bows-2 dataset each with

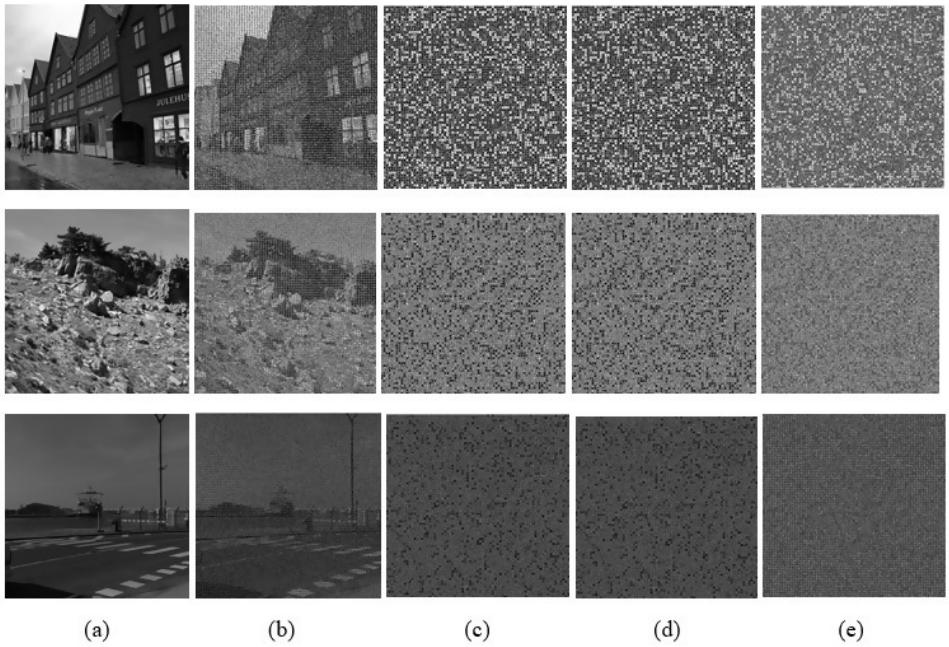

FIGURE 8. Experimental results from BossBase dataset: (a) Original image, (b) Level-1 encrypted image, (c) Level-2 encrypted image, (d) Level-3 encrypted image, (e) Marked encrypted image.

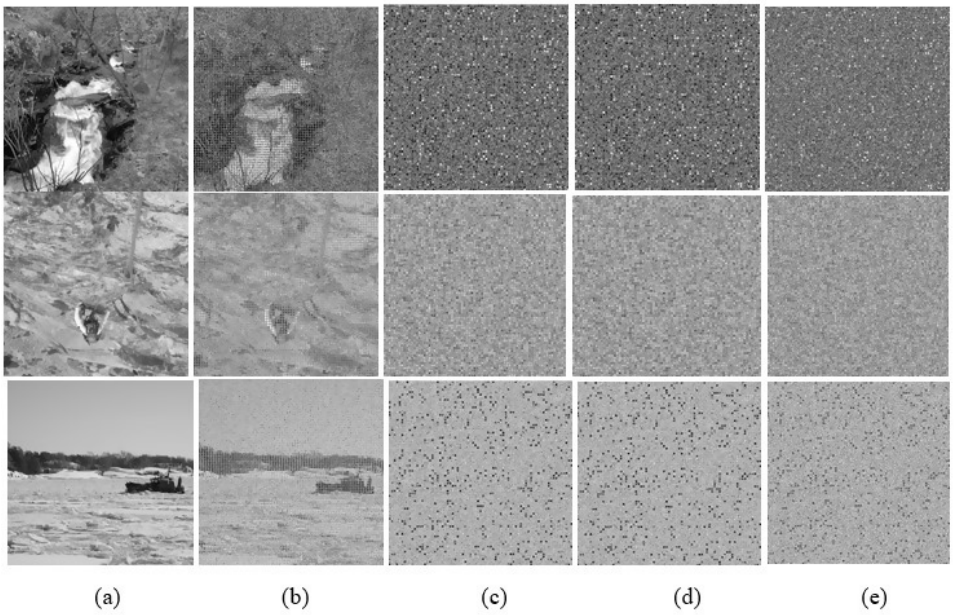

FIGURE 9. Experimental results from BOWS-2 dataset: (a) Original image, (b) Level-1 encrypted image, (c) Level-2 encrypted image, (d) Level-3 encrypted image, (e) Marked encrypted image.

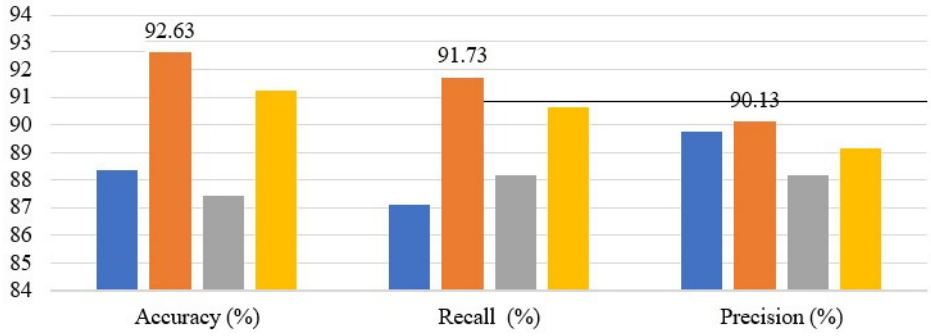

FIGURE 10. Graphical comparison of accuracy, recall, and precision.

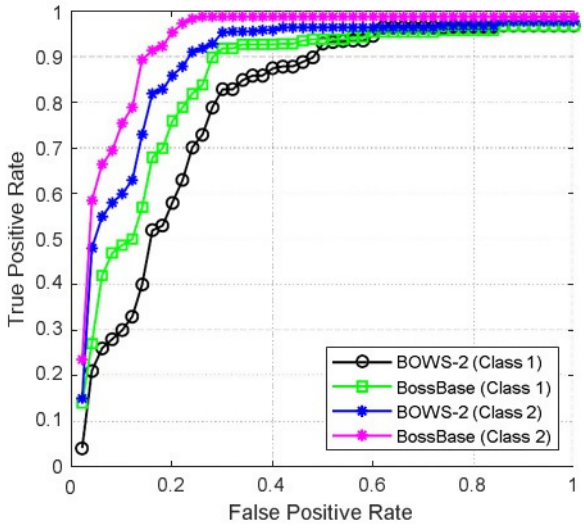

FIGURE 11. ROC curve for the datasets BOWS-2 and BossBase.

3000 test images. Figures 12(c) and 12(d) show the confusion matrix obtained during the testing phase of CNN for BossBase and Bows-2 dataset each with 4000 test images.

Table 2 shows the comparison of evaluation metrics PSNR, SSIM, and ER for the proposed method with traditional approaches. The proposed approach provides a higher embedding rate when compared to traditional schemes. On average, the proposed scheme provides an embedding rate of 1.955bpp with a PSNR and SSIM of 55.58dB and 0.9913 which is higher than the schemes proposed by Chen et al. [4] and Pauline et al. [15]. Figure 13 shows the comparison of PSNR and SSIM for different embedding rate

Figures14(a) and 14(b) shows the comparison of PSNR between the encrypted image and the marked encrypted image for different embedding rate for the image Figures 7(e) and 7(j) respectively. As the embedding rate is increased the PSNR gradually reduces. The proposed approach provides a high PSNR for different embedding rates between 0.2 bpp and 2bpp when compared to other traditional schemes.

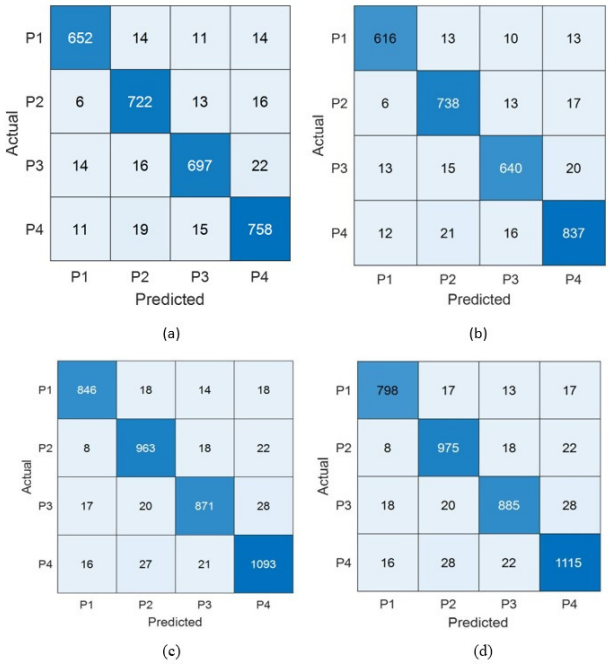

FIGURE 12. Confusion matrix obtained during the testing phase of CNN: (a) BossBase dataset (Class 1), (b) Bows-2 (Class 1), (c) BossBase dataset (Class 2), (d) Bows-2 dataset (Class 2).

TABLE 2. Comparison of PSNR, SSIM, and ER for the proposed method with traditional schemes.

Scheme	PSNR (dB)	SSIM	ER (bpp)
Puyang et al.[14]	60.84	0.9988	1.42
Puteaux et al.[16]	55.63	0.9909	1.81
Yi et al.[29]	51.36	0.9871	1.92
Chen et al.[4]	53.47	0.9892	1.93
Pauline et al.[15]	57.81	0.9921	1.70
Proposed (BossBase-Class 1)	55.97	0.9916	1.95
Proposed (Bows-2 Class 1)	55.36	0.9912	1.96
Proposed (BossBase-Class 2)	55.17	0.9911	1.96
Proposed (Bows-2 Class 2)	55.82	0.9915	1.95

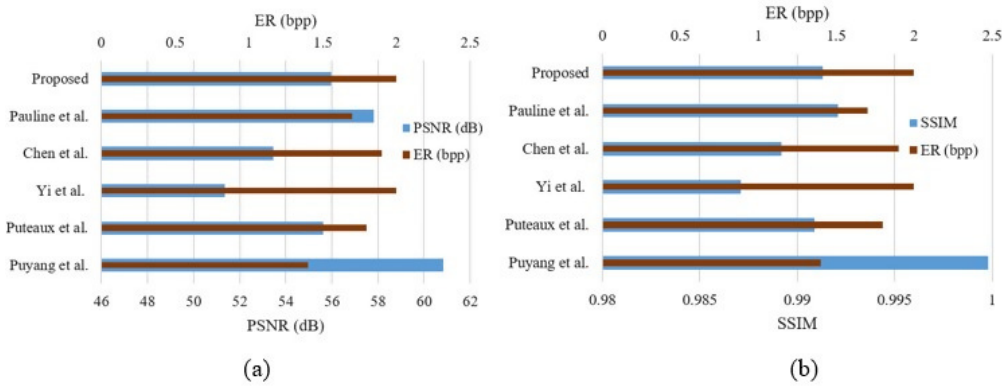

FIGURE 13. Comparison of PSNR, SSIM, and ER for the proposed method with traditional approaches: (a) Variation of PSNR with ER, (b) Variation of SSIM with ER.

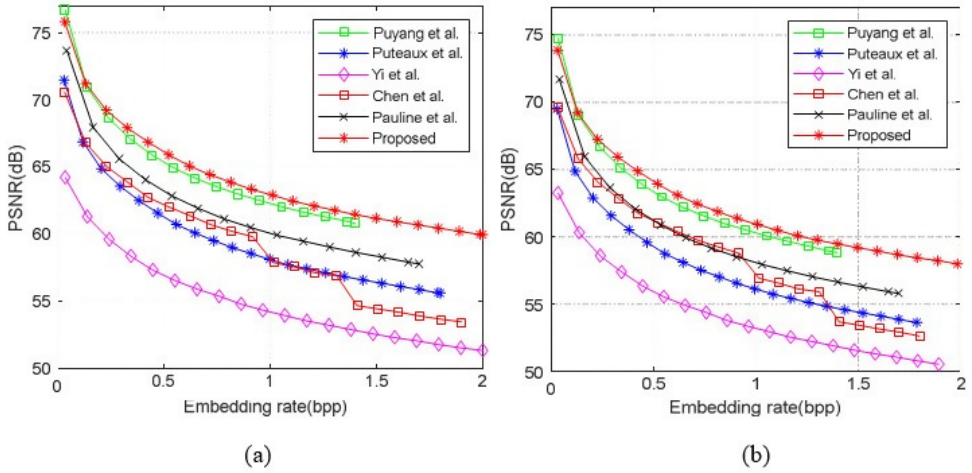

FIGURE 14. Comparison of PSNR for different embedding for the sample images from BossBase and Bows-2 dataset (a).

4. Conclusion

This paper proposed a CNN-based high-capacity predictor estimation for reversible data embedding in the cloud network. Initially, this approach estimates the best predictor using the CNN where the labels for the CNN training are created by the trial embedding approach. The predictor that offers the maximum embedding rate is considered as the class of the corresponding encrypted image. The images that are being uploaded to the cloud are encrypted by the user using a three-level encryption process. In the testing phase, the encrypted images that are uploaded to the cloud by the user are provided as the test image to the trained CNN model for estimating the predictor class that provides a maximum embedding capacity. The data is then embedded using the classified predictor. The performance of the classification algorithm was evaluated using the metrics such as accuracy, sensitivity, and specificity. The use of CNN in predictor classification provides an accuracy, recall, and precision of 92.63%, 91.73%, and 90.13% respectively for the BossBase dataset. Further, the PSNR, SSIM and embedding capacity of the reversible data embedding algorithm were evaluated using the datasets namely BossBase and BOWS-2. The proposed approach provides an average PSNR and SSIM of 55.58dB and 0.9913 respectively with an embedding rate of 1.955bpp

References

[1] P. Bas, T. Filler, and T. Pevný, *Break Our Steganographic System": The Ins and Outs of Organizing BOSS*, Information Hiding, *6958* (2011), 59–70.

[2] P. Bas and T. Furon, *Image database of BOWS-2*, Available: http://bows2.eclille.fr.

[3] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo, *High Capacity Reversible Data Hiding in Encrypted Images by Patch-Level Sparse Representation*, IEEE Transactions on Cybernetics, *46*(5) (2016), 1132–1143.

[4] K. Chen and C. C. Chang, *High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based MSB plane rearrangement*, Journal of Visual Communication and Image Representation, *58* (2019), 334–344.

[5] Y. Hu, H. -K. Lee, and J. Li, *DE-Based Reversible Data Hiding With Improved Overflow Location Map*, IEEE Transactions on Circuits and Systems for Video Technology, *19*(2) (2009), 250–260.

[6] Jun Tian, *Reversible data embedding using a difference expansion*, IEEE Transactions on Circuits and Systems for Video Technology, *13*(8) (2003), 890–896.

[7] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. -G. Choo, *A Novel Difference Expansion Transform for Reversible Data Embedding*, IEEE Transactions on Information Forensics and Security, *3*(3) (2008), 456–465.

[8] X. Li, B. Yang, and T. Zeng, *Efficient Reversible Watermarking Based on Adaptive Prediction-Error Expansion and Pixel Selection*, IEEE Transactions on Image Processing, *20*(12) (2011), 3524–3533.

[9] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, *Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption*, IEEE Transactions on Information Forensics and Security, *8*(3) (2013), 553–562.

[10] B. Ou, X. Li, W. Zhang, and Y. Zhao, *Improving Pairwise PEE via Hybrid-Dimensional Histogram Generation and Adaptive Mapping Selection*, IEEE Transactions on Circuits and Systems for Video Technology, *29*(7) (2019), 2176–2190.

[11] B. Ou, X. Li, Y. Zhao, R. Ni, and Y. -Q. Shi, *Pairwise Prediction-Error Expansion for Efficient Reversible Data Hiding*, IEEE Transactions on Image Processing, *22*(12) (2013), 5010–5021.

[12] W. Puech, M. Chaumont, and O. Strauss, *A Reversible Data Hiding Method for Encrypted Images*, SPIE Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents, San Jose, CA, USA, (2008), 1–9.

[13] P. Puteaux, S. Ong, K. Wong, and W. Puech, *A survey of reversible data hiding in encrypted images – The first 12 years*, Journal of Visual Communication and Image Representation, *77* (2021), 103085.

[14] P. Puteaux and W. Puech, *An Efficient MSB Prediction-Based Method for High-Capacity Reversible Data Hiding in Encrypted Images*, IEEE Transactions on Information Forensics and Security, *13*(7) (2018), 1670–1681.

[15] P. Puteaux and W. Puech, *A Recursive Reversible Data Hiding in Encrypted Images Method With a Very High Payload*, IEEE Transactions on Multimedia, *23* (2021), 636–650.

[16] P. Puteaux and W. Puech, *EPE-based Huge-Capacity Reversible Data Hiding in Encrypted Images*, 018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China, (2018), 1–7.

[17] Y. Puyang, Z. Yin, and Z. Qian, *Reversible Data Hiding in Encrypted Images with Two-MSB Prediction*, 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China , (2018), 1–7.
[18] Z. Qian, X. Zhang, and S. Wang, *Reversible Data Hiding in Encrypted JPEG Bitstream*, IEEE Transactions on Multimedia, *16*(5) (2014), 1486–1491.
[19] A. Rosewelt and A. Renjit, *Semantic analysis-based relevant data retrieval model using feature selection, summarization and CNN*, Soft Computing, *24* (2020), 16983–17000.
[20] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, *Reversible Watermarking Algorithm Using Sorting and Prediction*, IEEE Transactions on Circuits and Systems for Video Technology, *19*(7) (2009), 989–999.
[21] N. K. Sao, C. T. Luyen, and P. V. At, *Efficient reversible data hiding using block histogram shifting with invariant peak points*, J. Inf. Hiding Multim. Signal Process., *38*(1) (2022), 78–97.
[22] K. P. S. Shijin and D. D. Edwin, *Simulated attack based feature region selection for efficient digital image watermarking*, 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), Nagercoil, India, (2012), 1128–1132.
[23] M. Suchetha, N. S. Ganesh, R. Raman, and D. E. Dhas, *Region of interest-based predictive algorithm for subretinal hemorrhage detection using faster R-CNN*, Soft Computing, *25* (2021), 15255–15268.
[24] D. M. Thodi and J. J. Rodriguez, *Expansion Embedding Techniques for Reversible Watermarking*, IEEE Transactions on Image Processing, *16*(3) (2007), 721–730.
[25] M. Y. Valandar, M. J. Barani, P. Ayubi, and M. Aghazadeh, *An integer wavelet transform image steganography method based on 3D sine chaotic map*, Multimed Tools Appl, *78* (2019), 9971–9989.
[26] W. Weiqing, Y. Junyong, W. Tongqing, and W. Weifu, *A high capacity reversible data hiding scheme based on right-left shift*, Signal Processing, *150* (2018), 102–115.
[27] C.-H. Yang and M.-H. Tsai, *Improving histogram-based reversible data hiding by interleaving predictions*, IET Image Processing, *4*(4) (2010), 223–234.
[28] M. Yesilyurt and Y. Yalman, *New approach for ensuring cloud computing security: using data hiding methods*, Sādhanā, *41* (2016), 1289–1298.
[29] S. Yi and Y. Zhou, *Separable and Reversible Data Hiding in Encrypted Images Using Parametric Binary Tree Labeling*, IEEE Transactions on Multimedia, *20*(1) (2019), 51–64.
[30] X. Zhang, J. Long, Z. Wang, and H. Cheng, *Lossless and Reversible Data Hiding in Encrypted Images With Public-Key Cryptography*, IEEE Transactions on Circuits and Systems for Video Technology, *26*(9) (2016), 1622–1631.
[31] A. Zulehner and R. Wille, *Make it reversible: Efficient embedding of non-reversible functions*, Design, Automation & Test in Europe Conference & Exhibition (DATE), (2017), 458–463.