COMP90055 Research Project Final Report: A steganographic performance comparison of the C_4S algorithm with selected SSIS algorithms

Student Name: Qingyang Feng (980940)

Principal Supervisor: Prof. Udaya Parampalli

Co-Supervisor: Dr. Peter Eze

The University of Melbourne

August 2023

Abstract

This report presents a study of the steganographic performance of the Constant Correlation Compression Coding Scheme (C_4S) [5] in comparison with two other Spread Spectrum Image Steganography (SSIS) algorithms, that of Kumar et al. [9] and that of Naseem et al. [11]. By using the evaluation framework proposed by Eze et al. [4], the performance of the three algorithms is evaluated in terms of distortion, steganographic capacity, accuracy of watermark extraction, effect on machine-based diagnostic systems, ability for tamper detected and robustness against attacks.

DECLARATION

I certify that

– this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.

– where necessary I have received clearance for this research from the University's Ethics Committee and have submitted all required data to the School

– the thesis is 5,945 words in length (excluding text in images, table, bibliographies and appendices).

Signed by: Qingyang Feng 05/08/2023

CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. Introduction

With the significant rise in the adoption of e-health technologies in recent years, medical information such as medical images and Electronic Medical Records (EMR) are often stored in a digital form and transmitted electronically among different parties. This sensitive medical data not only could be breached during transmission but could also be tampered without authorization by different parties. While data encryption techniques can offer some protection during transmission, once the receiver decrypts the data, it is no longer protected. To complement data encryption, information hiding (IH) techniques are often used to provide extra security.

Watermarking and steganography are both IH techniques to hide information in cover data such as audio, videos and images. However, they are different in their purposes. Steganography is used to conceal the existence of a message in the cover data and the cover data itself is of less or no importance. In contrast, watermarking adds value to the cover data that can prove ownership or identify the source of the cover [12]. In the context of medical data, watermarking can be used to embed patient information into medical images, which can then be transmitted securely. Only individuals with the corresponding key can extract the embedded patient data from the transmitted image, ensuring the privacy and security of the sensitive information [12].

Three general challenges of IH techniques are robustness, imperceptibility and capacity. Robustness refers to the ability to detect the watermark after the algorithm is applied despite various unintended modifications or attacks that may occur during transmission, storage, or processing of the cover media [1]; imperceptibility refers to the resistance against steganalytic attacks as well as how well the diagnostic qualities are preserved [12]; and capacity refers to *the maximum data in bits that can be hidden into a cover without degrading it to a level that an adversary can easily detect* [5]. These three factors are interrelated and there are trade-offs among them based on the specific requirements of the application.

Steganography techniques can be categorized by the way information is embedded in the cover. Some common classes of IH techniques are the Least Significant Bit (LSB) method which replaces the least significant bit of each pixel in an image with a bit from the secret message; Transform Domain techniques which embeds the secret message in the transform domain of an image, such as the frequency domain; and spread spectrum techniques. [7].

Among these methods, spread spectrum is of increasing importance as it is more secure than other methods [5]. Spread spectrum (SS) is a modulation technology initially developed to provide secure communication. It reduces and spreads the energy of the modulating information across various channels over a much larger transmission bandwidth with a noisy pseudorandom sequence [12]. The embedded information is ensured to not have a distinguishable peak in comparison to other noise signals in the

wideband so that it is hard for an attacker to detect, intercept or jam the embedded information [12].

In a generalized SS steganography system, the secret message is embedded into the original cover with a pseudorandom noise (PN) sequence. The watermarked data is then transmitted and the receiver uses the same PN sequence to extract the hidden message [12]. A commonly used embedding function is the additive embedding function [12]:

$$Y_{ij} = X_{ij} + \alpha W_{ij}(-1)^{S_i} \tag{1}$$

where Y_{ij} is the individual elements of a watermarked data, X_{ij} is the individual elements of the original cover data, W_{ij} is the pseudorandom noise signal, α is the embedding strength and $S_i = \{0, 1\}$ is the individual elements of the message to be embedded. At the receiving end, the message embedded in the cover is normally retrieved by applying a linear correlation between Y_{ij} and W_{ij}. The extracted message S_i' is given by [12],

$$S_i' = \begin{cases} 0 & corr(Y_{ij}, W_{ij}) > 0 \\ 1 & corr(Y_{ij}, W_{ij}) < 0 \end{cases} \tag{2}$$

where

$$corr(Y, W) = p = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} Y_{ij} W_{ij} [5] \tag{3}$$

Based on the generalized SS steganography system, various adaptation SSIS algorithms were developed [9],[10], [11]. However, these algorithms suffer from the inherent problem of SS steganography, low data-carrying capacity. In SS steganography, a single bit of S_i is embedded by many bits of W_{ij} and thus the information is spreaded across a sub-block W_i. Although by embedding a single bit of information within a large area of the cover, SS watermarking achieves higher security, the capacity is small [12].

To improve SS steganography, a novel IH technique Constant Correlation Compression Coding Scheme (C_4S) [5] was proposed. It offers flexibility in cover integrity checks, image authentication, and robustness with an increased capacity. To assess the performance of the C_4S algorithm, a steganographic evaluation is necessary.

The purpose of this work is to follow the path of previous works on the C_4S algorithm and study the steganographic performance of the C_4S algorithm with a focus on its application to medical images. Two other Spread Spectrum Image Steganography (SSIS) algorithms are selected, that of Kumar et al. [9] and Naseem et al. [11], to make a comparison with the C_4S algorithm in terms of distortion, steganographic capacity, effect on machine-based diagnostic systems, ability for tamper detection as well as robustness to attack.

In Section II, the related works are reviewed. In Section III the methodology for evaluating the chosen SSIS algorithms are described. The results are presented and discussed in Section IV and a conclusion is summarised in Section V.

II. Literature Review

A. Spreading Sequence

At the core of the design of a SS embedding algorithm is the choice of spreading sequences (W) with appropriate correlation properties and a proper embedding strength (α) so that the robustness, imperceptibility and capacity of the algorithm are optimized [12]. These sequences help spread the embedded data across a wide frequency band, making it hard for unauthorized users to detect the hidden information. Some important properties of spreading sequences are: some level of security to prevent easy forgery by unauthorized users; easy reproduction at the receiver for extraction of the transmitted message from the sender; little or no interference with co-existence of other users within the same spectrum. Some commonly used spreading sequences in SSIS include:

- Pseudo Noise (PN) sequence: a sequence of pseudorandom binary numbers generated using a linear-feedback shift register (LFSR).
- Gold code: a binary sequence generated by combining two or more maximal-length LFSRs with different feedback taps and initial states.
- Kasami code: Kasami codes are a type of sequence derived from combinatorial designs known as Kasami sequence sets.
- Chaos sequence: a scrambled sequence generated by utilizing an initial condition, it can be deterministically retrieved if the same initial condition is supplied and its behaviour is hard to predict by analytical methods without the knowledge of the initial condition.

B. Kumar SS

Kumar et. al [9] proposed an SS steganography algorithm that uses PN sequence to embed information in the discrete wavelet transform domain (DWT). DWT decomposes an image into four components: a lower resolution approximation image (LL) and three detail components, horizontal (HL), vertical (LH), and diagonal (HH). This process can be iteratively applied to achieve multiple level wavelet decompositions. To embed the watermark in the cover image, "Haar" Wavelet transform is performed on the cover image and the second level subband coefficients are extracted: ccA (approximation coefficient), ccH (horizontal detail coefficient), ccV (vertical detail coefficient), ccD (diagonal detail coefficient). The watermark bits are embedded in the vertical and horizontal subband coefficients with different PN

sequence pairs (PN_h and PN_v): For each watermark bit, if the bit = 0, add the PN sequence to ccH and ccV by equation 4,

$$\begin{aligned} ccH &= ccH + k * PN_h \\ ccV &= ccV + k * PN_v \end{aligned} \tag{4}$$

where PN_h and PN_v are different for each watermark bit and k is the embedding strength.

The stego image is retrieved by applying the inverse "Haar" wavelet transform with second level horizontal and vertical detail coefficients being updated.

To extract the embedded information, the second level horizontal and vertical subband detail coefficients are obtained from the stego image. The same PN sequences used for embedding the watermark bits are generated and the correlation coefficients between the PN sequences and the two detail coefficients are computed. For each watermark bit:

$$\begin{aligned} corr_H &= \text{correlation between } PN_h \text{ and } ccH \\ corr_V &= \text{correlation between } PN_v \text{ and } ccV \\ avg_{corr} &= (corr_H + corr_V)/2 \end{aligned} \tag{5}$$

If avg_{corr} is greater than the mean of the average correlation values over all watermark bits, the corresponding watermark bit is "0", otherwise it is "1".

The algorithm was tested on 8-bit grayscale CT scans of size 512×512. Binary watermark images of medical information such as doctor's signature and telemedicine original center are used as watermarks. Different combinations of embedding strengths and watermarke sizes were experimented with. Imperceptibility was measured by Peak Signal-to-Noise Ratio (PSNR), which is explained later in this section, and the accuracy of watermark extraction was measured by the correlation between the original watermark image and the extracted watermark image. An embedding rate of 0.002bpp with a correlation of 0.149 was achieved at a PSNR of 41.412dB and a highest embedding of 0.0122bpp with a correlation of 0.461 was achieved at a PSNR of 30.138dB.

C. Naseem SS

Naseem et. al [11] proposed an SSIS method using chaotic sequence and Residue Number System (RNS). The region of interest (ROI) part of the image is residued: for a pixel with value X and pixel depth of 8 (the largest value of signal is $2^8 - 1 = 255$), the residue of the pixel can be computed as follows:

$$\begin{aligned} x_1 &= X \mod m_1 \\ x_2 &= X \mod m_2 \end{aligned} \tag{6}$$

where $m_1 = 17$, $m_2 = 15$.

If $x_1 = 16$, then $x_1 = x_2, x_2 = 15$; If $X = 255$, then $x_1 = 15, x_2 = 15$. The residue of the pixel can then be presented by 8 bits: $(\text{binary}(x_1), \text{binary}(x_2))$. By taking the residues of the region, the ROI is kept secure as residues are more sensitive to changes.

The watermark is embedded in the region of non-interest (RONI): a chaotic sequence S is generated to specify the embedding position of the watermark bits in the RONI. The watermark is spread by a chip-rate cr by repeating each watermark bit for cr times, modulated by a binary chaotic sequence $C = \{c_i | c_i \in \{-1, 1\}\}$ and embedded into the positions of RONI specified by S:

$$X_i' = x_i + \alpha \times m_i \times c_i \tag{7}$$

where x_i is the original pixel value, m_i is the message bit and α is the embedding strength.

A hash value is then computed from the residued ROI combined with the watermarked RONI. The hash value is used as an authentication watermark and is embedded in RONI by LSB substitution to obtain the stego image.

To extract the watermark, the same chaotic sequence S is generated. The hash value is extracted from the corresponding positions from RONI and is compared with the computed hash value of the stego image. If the two hash values are different, the image is tampered with. Otherwise, the watermark bit m_j can be extracted from specified positions by:

$$p_j = \sum_{i=j\times cr}^{(j+1)\times cr-1} x_i' \times c_i \tag{8}$$

Where c_i is the same chaotic modulating sequence used for embedding and x_i' is the watermarked pixel value. If the $p_j \geq 0$, $m_j = 0$; otherwise $m_j = 1$.

The original pixels can be recovered by applying Chinese Remainder Theorem (CRT):

$$X = \left[\sum_{i=1}^{k} M_i |r_i L_i|_{m_i} \right] \mod M \tag{9}$$

where $M = 255$, $k = 2$, $m_1 = 17$, $m_2 = 15$,

$$M_i = \frac{M}{m_i}$$

and

$$|L_i M_i|_{m_i} = 1,$$

where L_i is the multiplicative inverse of M_i with respect to m_i.

The method was tested on 8-bit grayscale ultrasound images of size 194×259 and a 8-bit grayscale logo image of size 50×50 is used as the watermark. The experiment was focused on the robustness

against different attacks on the watermarked image such as gaussian noise attack, speckle noise attack, rotation, cropping and tampering some bits. Robustness is measured by the normalized correlation (N_c) between the original and extracted watermark. The N_c values under all attempted attacks were good and the watermark was easily detectable under various attacks.

D. Constant Correlation Compression Coding Scheme (C_4S)

One problem related to the generalized SS is the Host Signal Interference (HSI) [6], i.e. the correlation p in (3) is not always zero for a sub-block without watermarking which leads to false positives for watermark detection. To eliminate HSI and ensure accurate watermark detection, a method called constant correlation watermarking based on the general additive SS watermarking is introduced in [6]. The method chooses a constant correlation value p and dynamically computes the embedding strength α for each sub-block depending on the message bit S_i to be embedded [6]:

$$\alpha_{0,1} = \frac{pmn - \sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}W_{ij}(-1)^{S_i}}{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}^2} \tag{10}$$

Watermark detection can be applied by performing a linear correlation between Y and W. A correlation of p will be detected for a 0-bit and $-p$ for a 1-bit. Any deviation from p or $-p$ beyond an acceptable error range ϵ indicates that the sub-block is not watermarked or has been tampered with. Therefore, if all sub-blocks are watermarked, data tampering can be detected. Such a method simultaneously provides tamper detection and watermark detection with increased accuracy [6].

Based on the constant correlation watermarking method [6], a Constant Correlation Compression Coding Scheme (C_4S) [5] is proposed to improve the capacity of SS watermarking. In C_4S, a group of cr binary bits rather than a single bit is embedded into each sub-block. This is achieved by choosing 2^{cr} different constant correlation values instead of one p value. Suppose $cr = 3$, then there are $2^3 = 8$ different binary bit groups and 8 distinct values will be chosen to embed these binary bit groups, e.g. $p = (-8, -6, -4, -2, 2, 4, 6, 8)$. Accordingly, correlations of the chosen values will be detected for the corresponding bit group and any deviation from it beyond a fault tolerance ϵ indicates data tampering or absence of watermarking.

E. Evaluation Methods

Some commonly used parameters for measuring the performance of steganography algorithms are:

- **Peak Signal-to-Noise Ratio (PSNR)**: measures the statistical imperceptibility of the watermark. PSNR is expressed in decibels (dB) and is computed by the ratio of the maximum value of a signal by the distorting noise introduced by embedded information.

$$PSNR = 20 * \log_{10}(\frac{B}{\sqrt{MSE}})$$

where B is the largest value of signal i.e. $2^n - 1$, where n is pixel depth and MSE is the Mean Squared error between the cover image and the stego image. The greater PSNR is the less imperceptible the degradation is. An effective IH algorithm commonly requires a PSNR of 40dB or greater.

- **Structural Similarity Index (SSIM)**: measures the visual imperceptibility of the watermark.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy}) + C_2}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \in [0, 1]$$

  where μ_x, μ_y and σ_x^2, σ_y^2 are the corresponding mean and variance of the images x and y, σ_{xy} is the covariance of x and y, $C_1 = (k_1 L)^2$, $C_2 = (k_2 L)^2$, where $L =$ largest value of signal, and $k_1 = 0.01$, $k_2 = 0.03$.

  The greater SSIM is, the more similar the two images are.

- **Bit Error Rate (BER)**: measures the accuracy of the extraction of the watermark. It is computed by the ratio of the number of bits that are wrongly extracted by the total number of embed bits:

$$BER = \frac{B_{error}}{B_{total}}$$

As machine-based automatic diagnostic algorithms are used in medicine, it has also become critical to evaluate the effect of data embedding in medical images on the performance of these diagnostic algorithms when evaluating medical image steganography algorithms. Eze et al. [2] studied the effect of C_4S on image classification with Support Vector Machine (SVM) where features are extracted from the ROI of medical images and fed into the models. Four textural features are extracted: contrast, energy, homogeneity and entropy. The extracted features are found to be modified by the application of C_4S but the modification effect is very low on certain features and both positive and negative effects are evident in the accuracy of detecting the positive class.

Eze et al. [3] also studied the effect of C_4S on the performance of deep learning models where the raw data are directly inputted into the neural networks. Different basic CNN and MobileNetV2 CNN models are trained with original data or watermarked data or a combination of original and watermarked data. The basic CNN model trained on combined data yielded the highest accuracy in predicting whether a blood smear image has malaria or not.

Eze et al. [4] also highlighted the limitations of existing evaluation frameworks for IH algorithms, which are restricted to signal processing parameters and fail to assess the suitability of these algorithms for preserving the diagnostic features of medical images. A novel evaluation framework was proposed to allow a dynamic selection of IH security algorithms for use in teleradiology. Eight criteria are selected in the framework including integrity score, privacy score, distortion score, location change score, dispersion

score, SVM accuracy, attack response, average score. Three IH algorithms, C_4S [5], that of Zain et. al [13] and LSB replacement steganographic algorithm are implemented and tested as an illustration of the framework. This framework is chosen to evaluate the algorithms implemented in this research and is further discussed in the next section.

III. METHODS

Three algorithms are implemented in this research, that of Kumar et al. [9], that of Naseem et al. [11] and C_4S with gold code. Randomly generated binary bit sequences are used as the watermarks for all algorithms.

The dataset used to evaluate the selected algorithms is the Pneumonia Chest X-ray dataset [8]. It contains over 3000 chest X-ray images of varied sizes and are either diagnosed with pneumonia or are normal. A subset of 1000 images are used in this research among which 500 are normal X-ray images and 500 are pneumonia-diagnosed X-ray images.

To choose the optimal parameters for each selected algorithm, they are first experimented on 100 X-ray images and are chosen by computing the associated PSNR, SSIM and BER of the watermarked images and extracted watermarks.

The parameter tuned algorithms are then evaluated with the framework proposed by Eze et al.[4]. The evaluation criteria included in the framework are:

1) **Integrity score** I_{sc}: measures the percentage of sub-blocks with attacks being correctly detected by the $ItegrityChecker$.

$$I_{sc} = \begin{cases} 0, & \text{if } IntegrityChecker = NULL \\ \frac{B}{A} \times 100, & \text{otherwise} \end{cases} \tag{11}$$

where A is the total number of sub-blocks under attack in ROI and B is the number of detected sub-blocks under attack.

2) **Privacy Score** P_{sc}: measures privacy by the accuracy of watermark extraction and the ability to hide the information in cover with out easy detection by unauthorised party.

$$P_{sc} = \frac{(100 - BER) + Capacity}{2} \tag{12}$$

where BER is the previously described bit error rate and

$$Capacity = \begin{cases} 100, & \text{if } C \geq M \\ \frac{C}{M} \times 100, & \text{otherwise} \end{cases} \tag{13}$$

where C is the number of bits embedded into the data with an acceptable distortion and M is the total number of bits to be embedded.

3) **Distortion Score** DT_{sc}: measures the relative distortion introduce by hidden information of a certain algorithm among all considering algorithms.

$$DT_{sc} = \begin{cases} 0, & \text{if } D_i < 40dB \\ 100, & \text{if } D_i = \infty \\ \frac{D_i}{\sum_{i=1}^{N} D_i} \times 100, & \text{otherwise} \end{cases} \tag{14}$$

where D_i is the PSNR distortion of algorithm i out of N evaluating algorithms. As previously described, a higher distortion score indicates a lower distortion.

4) **SVM Accuracy**: measures the classification accuracy of a Support Vector Machine (SVM) trained on biomarkers extracted from the images watermarked by each of the three algorithms. SVM is a supervised machine learning algorithm used for classification tasks. Its primary objective is to find the optimal hyperplane that best separates different classes of data points in a high-dimensional feature space. In the case of this work, there are two classes: positive, indicating the x-ray has pneumonia, and negative, indicating the x-ray is normal. Accuracy is defined as:

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{15}$$

where TP is ture positive, TN ture negative, FN is false negative, and FP is false positive. Accuracy measures the percentage of images that are correctly classified.

Biomarkers are extracted from medical images and are used as features for SVM classification. The biomarkers considered in this work, as suggested by [2], are contrast, energy, homogeneity and entropy extracted from ROI of the chest X-rays. As suggested in [2], the rectangular center three quarters of the chest X-ray image is taken as the ROI, e.g. if the cover image is of size 800×800, the ROI is the 600×600 rectangle with the upper left corner position of the ROI being $(100, 100)$. An example of the separation of ROI and RONI is illustrated in figure 1.

Values of the contrast, energy and homogeneity features are computed by Gray Level Co-occurrence Matrix (GLCM). GLCM is a statistical method used to examine the textures of the image by analyzing the spatial distribution of grey-level pixel intensities in an image. GLCM is defined by equation 16:

$$GLCM_{d,\theta} = G_{d,\theta}[i, j] = C_{ij} \tag{16}$$

where GLCM is computed by counting how many times a pixel of grayscale value i occurs adjacent to a pixel of grayscale value j at a distance d in a direction specified by θ. A distance

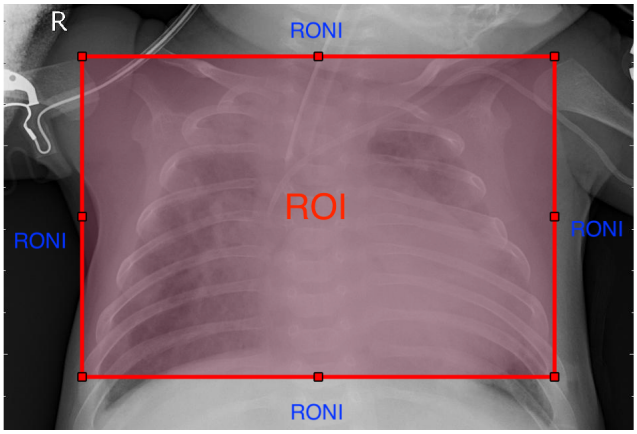


Figure 1: The center three quarters are taken as ROI and the rest of image is RONI.

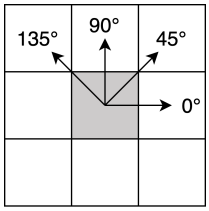


Figure 2: Computation of GLCM with $d = 1$ and $\theta = 0°, 45°, 90°, 135°$

$d = 1$ and directions $\theta = 0°, 45°, 90°, 135°$ are considered in this work and are illustrated in figure 2.

The three biomarkers calculated from GLCM are:

- **Contrast**: measures the local intensity contrast between pixels and their neighbouring pixels.

$$C_t = \sum_i \sum_j (i-j)^2 G_{d,\theta}[i,j] \tag{17}$$

- **Homogeneity**: measures the proximity of element distribution in the GLCM to the diagonal of GLCM.

$$C_h = \sum_i \sum_j \frac{G_{d,\theta}[i,j]}{1+|i-j|} \in [0,1] \tag{18}$$

  Homogeneity also known as Angular Second Moment (ASM). The higher ASM value is the more similar the local pixels are. It is ranged between 0 and 1, where 0 indicates high variation in pixel intensities and 1 indicates high homogeneity, i.e. uniformity in pixel intensities.

- **Energy**: measures the sum of squared elements in the GLCM.

$$C_e = \sqrt{\sum_i \sum_j (G_{d,\theta}[i,j])^2} \in [0,1] \tag{19}$$

Energy is also known as Angular First Moment (AFM). Both homogeneity and energy provide information about the uniformity of the image. Homogeneity emphasizes how closely packed the GLCM elements are around the main diagonal i.e. the local patterns, while energy focuses on the overall sum of squared elements i.e. the overall pattern.

The other biomarker used is:

- **Entropy**: measures the amount of randomness in the pixel intensities of the image:

$$H(X) = \sum_i p(i) \log p(i) \tag{20}$$

where $p(i)$ is the probability of the occurrence of the pixel intensity level i in the image. Entropy gives texture information about the image as well. A low entropy value indicates that the image is composed of only a few different pixel values while a high entropy value suggests that the pixel values are more spread out across a wide range of values, indicating a higher complexity of the image.

5) **Location Change Score** LC_{sc}: measures the percentage deviation in value of a biomarker caused by data embedding.

$$LC_{sc} = 100 - \frac{|Y - X|}{Y} \times 100 \tag{21}$$

where X is the original biomarker value and Y is the biomarker value obtained after information hiding.

6) **Dispersion Score** D_{sc}: measures the tendency of the algorithm to alter disease classification by machine based systems or humans. It is a binary measurement of value 0 or 1. Suppose a biomarker extracted from the original images before watermarking has values between X_1 and X_2 for images with class X, then the Dispersion Score is calculated as:

$$D_{sc} = \begin{cases} 0, & \text{if } Y < X_1 \text{ or } Y > X_2 \\ 100, & \text{otherwise} \end{cases} \tag{22}$$

where Y is value of the biomarker after applying the stenography algorithm.

7) **Attack Response**: measures the ability to extract the hidden information after attack. It is defined as:

$$AttackResponse = 100 - BER \tag{23}$$

8) **Average Score**: the average score of the considered criteria

$$AverageScore = \frac{\text{sum of individual criteria scores}}{\text{number of criteria considered}} \tag{24}$$

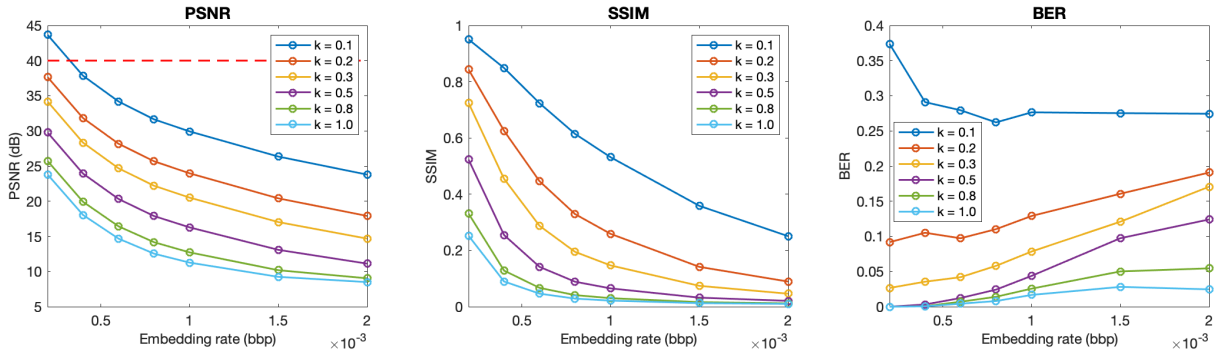


Figure 3: Average PSNR/SSIM/BER over 100 X-ray images watermarked by KumarSS with different values of embedding rate and embedding strength (k).

IV. Results and Discussion

The code for the implementation of the algorithms and the experiments is available at https://github.com/claire-ef/COMP90055.

A. Parameter Tuning

1) Kumar SS: The Kumar spread spectrum algorithm is implemented in MATLAB. Different embedding rate ranging $[0.0002, 0.0004, 0.0006, 0.0008, 0.001, 0.0015, 0.002]$ bit per pixel (bpp) and different embedding strength $k = [0.1, 0.2, 0.3, 0.5, 0.8, 1]$ were experimented. The average PSNR/SSIM/BER over 100 images are reported in figures 3.

It is noticed that both PSNR and SSIM decrease rapidly with the increase of either embedding rate or embedding strength. A PSNR of value over 40dB is only achieved when $k = 0.1$ at an embedding rate of 0.0002 bpp with a high BER. For $k > 0.1$, BER increases with the increase of embedding rate and decreases with the increase of embedding strength. However, no significant improvement in the accuracy of watermark extraction $(1 - BER)$ is observed with the increase of embedding strength. Therefore, for comparison with other algorithms, the embedding strength is chosen to be 0.2 and the embedding rate is chosen to be $0.0002bpp$.

2) Naseem SS: The Naseem spread spectrum algorithm is implemented in Java. The same rectangular center three quarters as defined in figure 1 is used as the ROI. The logistic map used for the generation of chaotic sequences is $X_{n+1} = rX_n(1 - X_N)$ with $x_0 = 0.25$ and $r = 3.58$. The embedding strength is chosen to be 6. Different chip rates cr are experimented and the computed embedding rate and average PSNR/SSIM/BER over 100 images are reported in table I.

Watermarks of different lengths are used for different chip rates and image sizes to utilize all the pixels in RONI and the corresponding embedding rate is computed. PSNR and SSIM is computed on the cover

Chip Rate (cr)	1	2	4	8	16	32	64
Embedding Rate (bpp)	0.0761	0.0380	0.0190	0.0095	0.0048	0.0024	0.0012
PSNR (dB)	43.7737	43.7748	43.7769	43.7790	43.7730	43.7736	43.7718
SSIM	0.996	0.996	0.996	0.995	0.996	0.995	0.996
BER	0.48998	0.07722	0.07309	0.06352	0.04838	0.01949	0.00279

Table I: Embedding rate and average PSNR/SSIM/BER over 100 X-ray images watermarked by NaseemSS with different chip rates (cr).

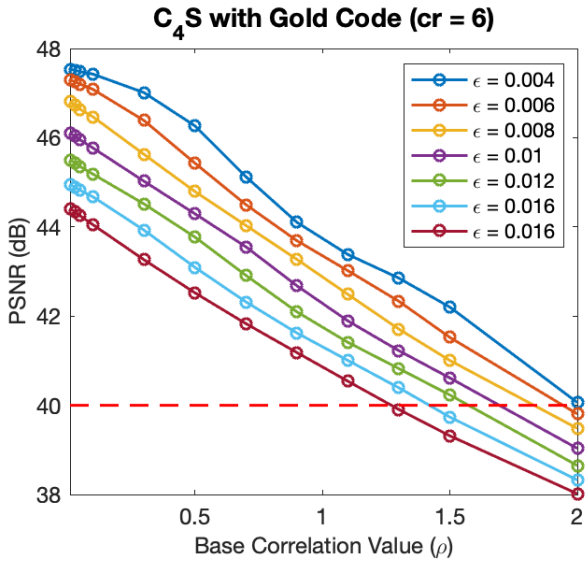


Figure 4: Average PSNR over 100 X-ray images watermarked by C_4S with different values of BCV (ρ) and fault tolarence (ϵ).

image and the stego image after the residued ROI is recovered.

It is noticed that both PSNR and SSIM remain constantly high for all different chip rates and when $cr \geq 2$, BER remains relatively low and no significant decrease is observed with the increase of chip rate. Therefore, for comparison with other SSIS algorithms, the chip rate is chosen to be 2.

3) C_4S: The C_4S algorithm is implemented with Java. Sub-blocks of size 8×8 are used. The gold code is generated from $x^6 + x^5 + 1$ and $x^6 + x^5 + x^4 + x + 1$ preferred pair. To choose an appropriate base correlation value (BCV), denoted by ρ, and a fault tolerance (ϵ), a grid search is performed while keeping the compression rate constant (cr=6). Different combinations of ρ and ϵ are experimented and the average PSNR/SSIM/BER over 100 images are computed. It is observed that only when $\rho >= 0.01$ and $\epsilon >= 0.004$, zero BER can be achieved. The PSNR obtained by different combinations of ρ and ϵ is plotted in figure 4.

Based on figure 4, $\rho = 0.01$ and $\epsilon = 0.004$ is chosen. With the chosen BCV (ρ) and fault tolerance (ϵ), different compression rates $cr = 2, 3, ..., 11, 12$ are also experimented. Zero BER is achieved for all

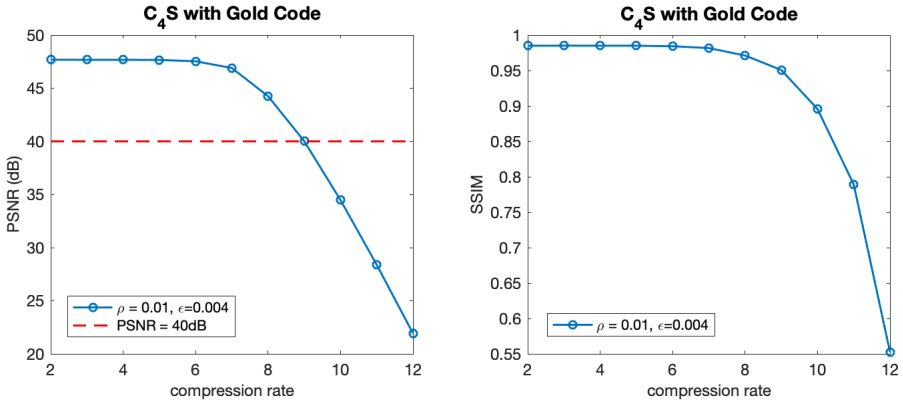


Figure 5: Average PSNR/SSIM over 100 X-ray images watermarked by C_4S with $\rho = 0.01$, $\epsilon = 0.004$ and different compression rates (cr).

Algorithm	Embedding Rate (bpp)	PSNR (dB)	SSIM	BER (%)
KumarSS [9]	0.0002	37.4867	0.9281	7.4560
NaseemSS [11]	0.0380	44.0699	0.9922	6.6982
C_4S [5]	0.1250	43.6326	0.9614	0

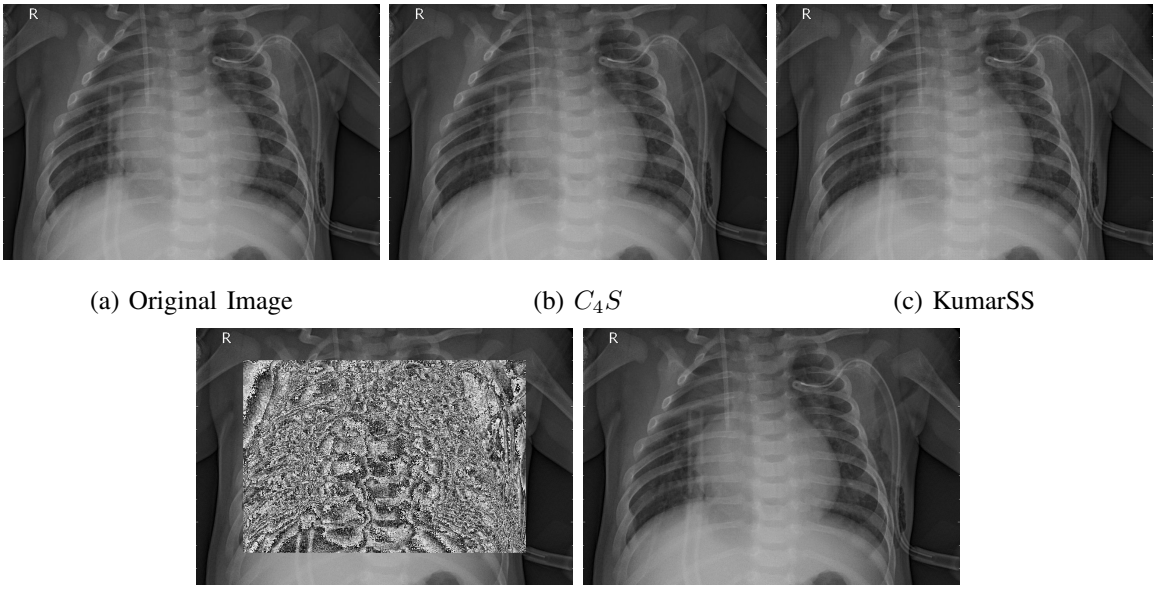
Table II: Embedding rate and average PSNR/SSIM/BER over 1000 X-ray images watermarked by different algorithms.

compression rates. PSNR and SSIM obtained with different cr are plotted in figure 5. It can be observed that when $cr \leq 8$, both PSNR ($\geq 44dB$) and SSIM (≥ 0.98) remained high and when $cr > 8$, there is a significant drop in both of the quality measures. Therefore, cr is chosen to be 8 for comparison with other algorithms.

B. Watermarking Results

The algorithms are then performed on the 1000 images after the parameters are tuned. Randomly generated binary bit sequences of different lengths are used as watermarks. The length of the watermark is decided by the capacity of the algorithm and the size of the image by embedding as many bit as possible allowable by the SSIS algorithm. The watermarked images obtained by different SSIS algorithms are displayed in figure 6. Figure 6a shows the original cover image, figure 6b shows the image watermarked by C_4S, figure 6c shows the image watermarked by $KumarSS$ [9], figure 6d shows the image watermarked by $NaseemSS$ and figure 6e shows the watermarked image 6d with ROI recovered. The embedding rate and the average PSNR/SSIM/BER over 1000 chest x-ray images are reported in table II. The values of PSNR and SSIM for $NaseemSS$ is calculated on the watermarked images with recovered ROI pixels.

Among the three SSIS algorithms, $KumarSS$ yielded the lowest PSNR ($37.4867dB$)/ SSIM (0.9281) and the highest BER (7.4560%) at a very low embedding rate of 0.0002 bpp. Figure 7 shows a zoomed-in part of an image watermarked by $KumarSS$ and there are grid patterns visible on the watermarked



(a) Original Image (b) C_4S (c) KumarSS

(d) NaseemSS (e) NaseemSS with recovered ROI

Figure 6: Images watermarked by different algorithms.

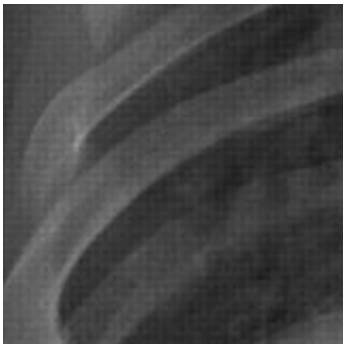


Figure 7: Gird patterns on the image watermarked by $KumarSS$

image.

Compared to $KumarSS$, $NaseemSS$ produced better PSNR ($44.0699dB$) and SSIM (0.9922) at a much higher embedding rate of 0.0380 bpp. It also has a slightly lower BER compared to $NaseemSS$. By performing the reversible Residue Number System (RNS) scheme, the ROI is kept secure and is not recognisable to the naked eye while it is still able to be recovered so that the diagnostic qualities are preserved.

The C_4S algorithm resulted in the highest embedding rate (0.1250 bpp) by embedding 8 bits in each 8×8 sub-block at a high PSNR ($43.6326dB$) and SSIM (0.9614). It is also the only SSIS algorithm among the three that achieved zero BER which makes it suitable for embedding text-based watermarks. Unlike image-based watermarks, text-based watermark are more sensitive to bit errors. For example, the

Algorithm	$KumarSS$	$NaseemSS$	C_4S
$contrast_{1,0^\circ}$	47.51	0.00	27.01
$contrast_{1,45^\circ}$	50.93	0.00	14.82
$contrast_{1,90^\circ}$	45.73	0.00	26.53
$contrast_{1,135^\circ}$	51.35	0.00	16.17
$homogeneity_{1,0^\circ}$	-31.73	0.00	-14.88
$homogeneity_{1,45^\circ}$	-40.50	0.00	-8.38
$homogeneity_{1,90^\circ}$	-30.58	0.00	-15.28
$homogeneity_{1,135^\circ}$	-40.83	0.00	-9.59
$energy_{1,0^\circ}$	-74.33	0.00	-34.32
$energy_{1,45^\circ}$	-84.48	0.00	-18.58
$energy_{1,90^\circ}$	-69.42	0.00	-33.79
$energy_{1,135^\circ}$	-85.00	0.00	-20.58
$entropy$	0.32	0.00	0.06

Table III: Location Change $\left(\frac{Y-X}{Y} \times 100\right)$ of biomarkers after watermarking.

ASCII code for character '1' is 00110001 and with a single bit error, the watermark will be mistakened as character '5' with its ASCII code being 00110101. This can be troublesome when using Electronic Health Records (EMR) as watermarks for medical applications.

C. Change in Biomarkers and SVM Classification

The values for the biomarkers are recomputed from the ROI after the images are watermarked. The location change, $\frac{Y-X}{Y} \times 100$ (where X is the original biomarker value and Y is the recomputed value after watermarking), is then calculated and the average location change over 1000 image for each biomarker is reported in table III.

There are 13 biomarkers in total, four for each of the three biomarkers computed from GLCM as four directions are considered and one for entropy.

There is no location change in all the biomarkers for $NaseemSS$ watermarked images as reversible RNS scheme is applied and there is no change in values for pixels of the ROI after watermarking. Compared to C_4S, $KumarSS$ produced more location change. For both C_4S and $KumarSS$, homogeneity and energy decreased and contrast increased after watermarking which indicates that the uniformity of the images has changed and more complexity was introduced into the image by data embedding. It is also noticed that both algorithms resulted in a greater location change in energy than in homogeneity. This suggests that the watermarking have had a greater impact on the overall complexity and uniformity of the image, while the localized uniformity might have been affected to a less degree.

The dispersion score for each biomarker is calculated according to equation 22 and is reported in table IV. It is observed that compared to the effect of $KumarSS$, there is a smaller shift in classes introduced

Algorithm	$KumarSS$	$NaseemSS$	C_4S
$contrast_{1,0^\circ}$	98.5	100.0	99.8
$contrast_{1,45^\circ}$	92.5	100.0	99.8
$contrast_{1,90^\circ}$	98.0	100.0	99.1
$contrast_{1,135^\circ}$	92.5	100.0	99.8
$homogeneity_{1,0^\circ}$	99.3	100.0	99.8
$homogeneity_{1,45^\circ}$	86.7	100.0	99.8
$homogeneity_{1,90^\circ}$	98.6	100.0	98.9
$homogeneity_{1,135^\circ}$	86.7	100.0	99.8
$energy_{1,0^\circ}$	69.4	100.0	99.0
$energy_{1,45^\circ}$	70.1	100.0	99.7
$energy_{1,90^\circ}$	79.5	100.0	98.7
$energy_{1,135^\circ}$	69.8	100.0	99.7
$entropy$	99.7	100.0	99.8

Table IV: Dispersion Score of biomarkers after watermarking.

Algorithm	Accuracy (%)	Specificity (%)	Recall (%)	Precision (%)
Original	99.00	98.67	98.68	99.33
KumarSS [9]	93.67	96.67	96.45	90.67
NaseemSS [11]	99.00	98.67	98.68	99.33
C_4S [5]	66.33	32.67	59.76	100.00

Table V: Accuray, specificity, precision and recall of the SVM models trained on images watermarked by different SSIS algorithms.

by C_4S.

SVM models are trained with Python. Four SVM models are trained: one on the original unwatermarked images, three on the images watermarked by the three different SSIS algorithms. For $NaseemSS$, watermarked images with recovered ROI is used. Polynomial kernel function is used to fit the SVM models. 700 randomly selected images in the dataset are watermarked and used as train set and the rest 300 unwatermarked images are used as test set to test the accuracy of the classification. The accuracy, specificity, precision and recall of each model is reported in table V.

Accuracy is as defined in the previous section. Specificity measures the percentage of people who are normal are correctly predicted as normal.

$$specificity = \frac{TN}{TN + FP} \tag{25}$$

Precision measures the percentage of people who are predicted as having pneumonia are actually diagnosed with pneumonia.

$$precision = \frac{TP}{TP + FP} \tag{26}$$

Recall measures the percentage of people who are diagnosed with pneumonia are correctly predicted to have pneumonia.

$$recall = \frac{TP}{TP + FN} \tag{27}$$

It can be observed from table V that after applying the watermark, SVM performance does not change for $NaseemSS$ as the ROI pixels are not altered after watermarking. KumarSS results in a decrease in all four measurements but the performance is still good. For C_4S, watermarking largely affected the performance of SVM classification with a low specificity (32.67%) and recall (59.76%) and a very high precision (100.00%) indicating that only about one third of the people who are normal are correctly predicted as normal, about two thirds of the people who are diagnosed with pneumonia are correctly detected and all of the people who are predicted to have pneumonia do have pneumonia. With a low recall, it is suggested that C_4S should be applied cautiously when false negative can have severe consequences. Suppose a patient has pneumonia, the probability of them being mistakenly diagnosed as normal by the SVM model trained on C_4S watermarked images is over 40% which can delay necessary treatment and intervention.

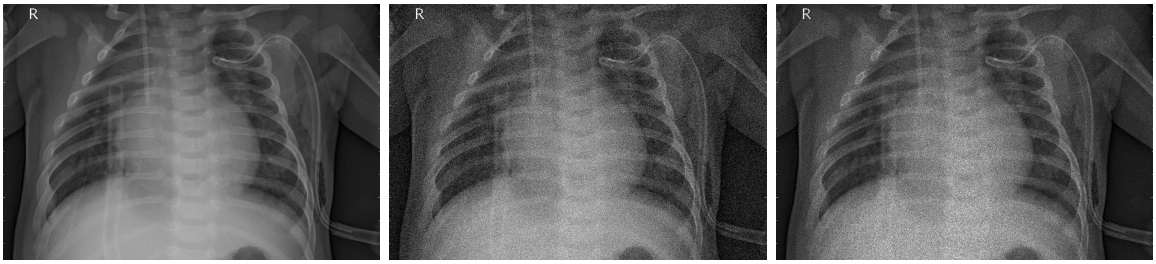
D. Tamper Detection and Robustness under Attacks

Four different attacks were considered in this work, which includes:

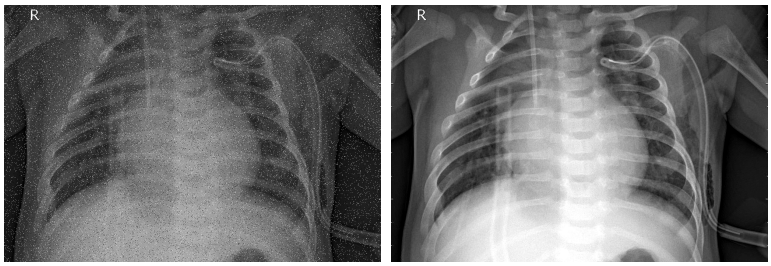
- Guassian Noise: adds Gaussian white noise with mean of 0 and variance of 0.01 to image.
- Speckle Noise: adds multiplicative noise with equation $J = I + n * I$, where I is the image to be attacked and n is uniformly distributed random noise with mean 0 and variance 0.05.
- Salt and Pepper Noise: adds salt and pepper noise with noise density 0.05 to image. Salt and pepper noise are random and isolated pixel intensity values that are much higher (salt) or much lower (pepper) than the surrounding pixels.
- Contrast Adjustment: increases the contrast of the image by saturating the bottom and the top 1% of all pixel values.

The four different kinds of attacks are performed on images after the watermarks are embedded. The effect of the attacks on the images are illustrated in figure 8.

Localised tamper detection is available for C_4S by determining whether each sub-block is tampered with or not. $NaseemSS$ is only able to tell whether the image is tampered with but it can not locate where the tampering was performed. There is no tamper detection available for $KumarSS$. The integrity scores for C_4S under different attacks (as defined in equation 11) are recorded in table VI. From the



(a) Original watermarked image (b) Adding Gaussian Noise (c) Adding Speckle Noise



(d) Adding Salt and Pepper Noise (e) Contrast Adjustment

Figure 8: Watermarked images under different attacks.

Attack	Gaussian Noise	Speckle Noise	Salt and Pepper Noise	Contrast Adjustment
Integrity Score	82.57	85.59	87.80	54.91

Table VI: Integrity Score of C_4S under different attacks.

results presented in table VI, it is suggested that C_4S's tamper detection is more robust under noise based attacks and is more vulnerable under contrast adjustment attack.

The BER of the extracted watermark under different attacks are reported in table VII. The watermark extraction of both $KumarSS$ and $NaseemSS$ are more robust against contrast adjustment attack than noise based attacks while C_4S is more robust against noise based attacks than contrast adjustment attacks. However, the watermark extraction of C_4S is more vulnerable to all the experimented attacks compared to $KumarSS$ and $NaseemSS$.

	Gaussian Noise	Speckle Noise	Salt and Pepper Noise	Contrast Adjustment
$KumarSS$	27.93	30.27	31.84	6.67
$NaseemSS$	38.58	28.04	10.73	6.70
C_4S	51.02	53.76	53.34	62.24

Table VII: BER(%) of the extracted watermark under attacks.

Criteria	$KumarSS$	$NaseemSS$	C_4S
Integrity Score (I_{sc})	0	1	77.72
Privacy Score (P_{sc})	46.35	61.85	100
Distortion Score (DT_{sc})	0 (37.49)	35.20	34.85
Location Change Score (LC_{sc})	49.79	100	81.54
Dispersion Score (D_{sc})	87.79	100	99.52
SVM Accuracy	93.67	99.00	66.33
Attack Response	75.82	78.99	44.91
Average Score	50.49	68.01	72.12

Table VIII: Values of Evaluation Criteria for the three selected SSIS

E. Average Score

The values for the eight criteria is displayed in table VIII. Integrity score is not available for $KumarSS$ as there is no integrity checker implemented; for $Naseem$, an integrity score of 1 is assigned, indicating only global tamper detection is available; for C_4S the average integrity score over the four types of attacks is taken as the final integrity score. The average location change scores and dispersion scores over the 13 biomarkers are taken as the final values for LC_{sc} and D_{sc} respectively. The average values of attack response over the four attacks are taken as the final values of attack response for each algorithm. The average score is computed on all other seven criteria.

Among the three algorithms, $KumarSS$ has the lowest average score. Compared to the two other algorithms, $KumarSS$ has a low capacity with high distortion. Although zero BER can not be achieved, the accuracy of watermark extraction is good with a BER of 7.46% and despite its large location change and dispersion, its effect on SVM classification is small.

$NaseemSS$ has a higher average score than $KumarSS$. It has a higher capacity with less perceptible degradation in the image. Due to its reversible RNS residue scheme, the location change score, dispersion score and SVM accuracy is high. It is also the most robust against attacks among the three algorithms.

C_4S yielded the highest average score. It achieved zero BER at a high embedding rate with low distortion when no attack is presented. However, despite its high location change score and dispersion score, C_4S implemented in this work has a significant negative effect on SVM classification accuracy. It is also less robust against noise-based attacks and contrast adjustment attack.

V. Conclusion

This work compared the steganographic performance of the C_4S algorithm with two other SSIS algorithms, that of Kumar et al. ($KumarSS$) [9] and that of Naseem et al. ($NaseemSS$) [11]. Results of the experiments showed that C_4S has a much higher embedding capacity (three times more than

$NaseemSS$ and 600 times more than $KumarSS$) with an effective PSNR of 43.63dB. When the watermarked image is not tampered with, zero BER can be achieved by C_4S. The ability of localised tamper detection is tested under contrast adjustment attacks and noise addition attacks. An average of 77.72% of the tampered sub-blocks can be detected. The robustness of watermark extraction under contrast adjustment attacks and noise addition attacks is also experimented. C_4S is found to be less robust against these attacks compared to the two other algorithms. C_4S is also found to have a more significant negative effect on SVM classification accuracy than the two other algorithms.

REFERENCES

[1] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan kaufmann, 2007.

[2] P Eze, U Parampalli, R Evans, and D Liu. Evaluation of the effect of steganography on medical image classification accuracy. *J. Appl. Bioinform. Comput. Biol.*, 9(4):2, 2020.

[3] Peter Eze and Udaya Parampalli. Deep learning evaluation of a steganographic algorithm. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1999–2005. IEEE, 2021.

[4] Peter Eze, Udaya Parampalli, Robin Evans, and Dongxi Liu. A new evaluation method for medical image information hiding techniques. In *2020 42nd annual international conference of the IEEE engineering in Medicine & Biology Society (EMBC)*, pages 6119–6122. IEEE, 2020.

[5] Peter U Eze, Udaya Parampalli, Robin J Evans, and Dongxi Liu. Spread spectrum steganographic capacity improvement for medical image security in teleradiology. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1–4. IEEE, 2018.

[6] Peter U Eze, P Udaya, and Robin J Evans. Medical image watermark and tamper detection using constant correlation spread spectrum watermarking. *International Journal of Computer and Information Engineering*, 12(3):124–131, 2018.

[7] Neil F Johnson and Stefan Katzenbeisser. A survey of steganographic techniques. In *Information hiding*, pages 43–78, 2000.

[8] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *cell*, 172(5):1122–1131, 2018.

[9] Basant Kumar, Harsh Vikram Singh, Surya Pal Singh, Anand Mohan, et al. Secure spread-spectrum watermarking for telemedicine applications. *Journal of Information Security*, 2(02):91, 2011.

[10] Hirak Kumar Maity and Santi Prasad Maity. Joint robust and reversible watermarking for medical images. *Procedia technology*, 6:275–282, 2012.

[11] Muhammad Tahir Naseem, Ijaz Mansoor Qureshi, Muhammad Zeeshan Muzaffar, and Atta ur Rahman 0001. Spread spectrum based invertible watermarking for medical images using rns and chaos. *Int. Arab J. Inf. Technol.*, 13(2):223–231, 2016.

[12] U Eze Peter, Udaya Parampalli, C Iwuchukwu Uchechi, and Onuekwusi Nnaemeka. Challenges and prospects of blind spread spectrum medical image watermarking. In *2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON)*, pages 10–18. IEEE, 2017.

[13] Jasni M Zain, Abdul RM Fauzi, and Azian A Aziz. Clinical evaluation of watermarked medical images. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5459–5462. IEEE, 2006.