A Novel Optimized Graph-Based Transform Watermarking Technique to Address Security Issues in Real-Time Application

Chirag Sharma, Amandeep Bagga, Bhupesh Kumar Singh, and Mohammad Shabaz

Department of Computer Science and Engineering, Lovely Professional University, Phagwara, India
Department of Computer Application, Lovely Professional University, Phagwara, India
Arba Minch University, Arba Minch, Ethiopia

Correspondence should be addressed to Chirag Sharma; chiragsharma1510@gmail.com

Received 16 February 2021; Revised 1 March 2021; Accepted 20 March 2021; Published 9 April 2021

Copyright © 2021 Chirag Sharma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The widespread of the Internet has led to the distribution of multimedia data, i.e., images, videos, and audios everywhere. The access to multimedia has become possible thus maintaining the confidentiality of the need of an hour. The illegal distribution and copying of multimedia data such as videos and images has lead to the design and development of watermarking technique. Many of the illegal users are getting the access of the real-time videos from various sources such as NETFLIX, Amazon Prime, and various OTT Platforms, and thus, illegal distribution is done. These videos are available in compressed domain. A watermark is a digital data that can be used like text, audio, and image and can be embedded in the multimedia, videos and images, so that it can be detected or extracted later so as to identify illegal use of multimedia media data [1]. An efficient watermarking scheme is characterized by four properties [2]:

(a) Imperceptibility: this property calculates the degree of invisibility of the hidden signal to the user, so the watermark should be targeted in those locations where watermark signal is unnoticeable

(b) Robustness: this property verifies the ability of watermark to survive attacks such as cropping, resizing, and additional noise

(c) Capacity: it identifies the total amount of data that can be hidden in the multimedia data

(d) Security: it identifies the addition of cipher to watermark that can be embedded in multimedia data

1. Introduction

The widespread of the Internet has led to the distribution of multimedia data, i.e., images, videos, and audios everywhere. The access to multimedia has become possible thus maintaining the confidentiality of the need of an hour. The illegal distribution and copying of multimedia data such as videos and images has lead to the design and development of watermarking technique. Many of the illegal users are getting the access of the real-time videos from various sources such as NETFLIX, Amazon Prime, and various OTT Platforms, and thus, illegal distribution is done. These videos are available in compressed domain. A watermark is a digital data that can be used like text, audio, and image and can be embedded in the multimedia, videos and images, so that it can be detected or extracted later so as to identify illegal use of multimedia media data [1]. An efficient watermarking scheme is characterized by four properties [2]:

(a) Imperceptibility: this property calculates the degree of invisibility of the hidden signal to the user, so the watermark should be targeted in those locations where watermark signal is unnoticeable

(b) Robustness: this property verifies the ability of watermark to survive attacks such as cropping, resizing, and additional noise

(c) Capacity: it identifies the total amount of data that can be hidden in the multimedia data

(d) Security: it identifies the addition of cipher to watermark that can be embedded in multimedia data
Real-time videos are the most part utilized interactive media information accessible now days. The illegal use by unauthorized personnel has propelled researchers to develop various video watermarking techniques [3]. The video watermarking techniques have many constraints regarding extraction of frames [4], selection of frames [5], types of videos [1], and codes of video [6]. There are 2 types of videos where watermark can be embedded, raw videos and compressed videos [7]. Raw videos are the videos that can be obtained using TV capture card or satellite that has huge bit rate and quality is never compromised in those, whereas compressed videos are processed in compressed domain where the bit rate is less and this watermarking scheme is faster, and it allows low computation cost for processing the watermark [8]. The increasing use of real-time video-based applications such as videos on demand [9], set top box, and online streaming such as Netflix and Amazon Prime are demanding high compression to meet the requirements of bandwidth for certain users along with maintaining best quality of videos [6]. For the real-time videos to be available with best codec, different video codecs have been mentioned in [6] so as to meet requirement of users in today’s world. Some of them are H.264/AVC (advanced video codec), XVID, and WMV, but since H.264 is the codec used nowadays, so the proposed watermarking scheme will be applicable to the same. The major challenge in watermarking in real-time videos is the selection of extracted frames and finding the best suitable frames from the video to embed watermark, and many frame selection mechanisms have been proposed in [10]. Our proposed technique will give optimal results in comparison to these. After the frames are selected, the next challenge is to select areas of frame to be watermarked. The research has been done of various techniques by various researchers [11]. The major challenge is the addition of watermark in the frames of the video poses the loss of quality [12] by affecting the video bit rate of the video [13], so the research is being carried out to focus on real-time requirements such as downscaling; quality loss of the video after watermark is added to the video, and our proposed work intends to solve these issues. Video watermarking techniques have been classified into following types based on domain [6]. In spatial or pixel domain methods, watermark is embedded directly on pixel positions; the best example of this method is LSB (Least Significant Bit) modification, although this is the fast method but lacks in achieving imperceptibility [6]. In frequency domain methods, host signal is converted to frequency signal [14]. The techniques are mentioned in Figure 1. The proposed technique deals with a transform named as graph-based transform that transforms signals in the form of graphs (V, E). The technique is coupled with SVD. The validity of watermarking scheme is dependent upon values of peak signal-to-noise ratio (PSNR) after applying attacks such as rotation, Gaussian noise, and compression attack on the proposed watermarking scheme. Further sections explain the research work carried out in this area followed by the proposed technique.

2. Related Work

A number of research studies related to watermark embedding using DWT as a transform domain has been carried out by Chaudhary et al. [3] which embeds watermark on the selected frame. High frequency subbands are selected on the areas where watermark embedding is present [15]. The research frequency domain achieves more level of robustness compared to spatial domain but is not efficient to handle various attack scenarios. The combination of DWT and SVD was also applied by various researchers [16]. The improvement of results based on values of PSNR was obtained against image processing attack scenarios. The particle swarm optimization methods [13] based on Dither modulation optimizes the results of PSNR by optimizing various factors (optimal frequency set) for embedding watermark. Applying optimization algorithm on combination of DWT and SVD has also produced high values of PSNR. The same has been listed in the work done by Mishra et al. [17]. Here, combination of DWT and SVD is carried out with firefly algorithm where multiple scaling factors were optimized using firefly algorithm. Cheung [10] proposed the scheme, depth map coding, using graph-based transform and transform domain scarification to counter the problems faced by discrete cosine transform method considering its inability to code block that contains arbitrarily shaped edges. The proposed scheme Agarwal et al. [18] also extended work by using hybrid fuzzy-BPN architecture to embed binary watermark in gray scale images, and the proposed scheme achieves good results in real-time scenarios. Rajpal et al. [19] extended the work done by Agarwal et al. [18] by applying novel fuzzy frame selection technique using bidirectional extreme learning machine on videos. The proposed scheme by [20] addresses fast mechanism to select frames. Masoumi and Amiri [4] proposed a method of video watermarking by applying BWT (biorthogonal wavelet transform) and SVD by using artificial bee optimization algorithm. Tabassum and Islam [3] proposed a method of identical frame extraction, and using frame difference method, identification and selection of appropriate frames is done to embed watermark inside them. The method is improved by Masoumi and Amiri [4] where motioned part of video is detected using the technique proposed in [21] by applying 3D wavelet transformation coefficients of different subbands which are selected for embedding. Venugopala et al. [11] listed key areas of feature selection for watermarking embedding by adjusting the pixel values and using scene change detection mechanism. Research carried out by Kumar and Rewani [8] also featured secured watermarking technique by encrypting watermark using rail fence cipher before embedding to randomly selected frames, and Mansouri and Amiri [21] have also proposed secured video watermarking in compressed domain by proposing new content-based key to select particular regions for embedding watermark. Araghi and Manaf [22] proposed a secured watermarking scheme by employing DWT with 2D SVD, thus eliminating the challenges of dependencies of watermarked image with cover images along with the use of 2 level-authentication systems to be applied on watermark extraction. This method increases the dimensionality of the image space, which increases the efficiency of the algorithm and makes it easier to implement in real-time application. Egilmez et al. [23] applied a concept of graph-based transform on video compression considering graph-based transform designs and edge adaptive design for video encoding and decoding. Gu
et al. [24] proposed a large-scale optimization technique to solve large-scale global optimization problems by introducing hybrid grey wolf optimization with genetic algorithm. Cao and Wang [25] proposed a secure watermarking technique based on hyperchaotic encryption on binary watermark to add additional security mechanism on the watermark. Sharma et al. [26] conducted the research by using graph-based transform, singular-valued decomposition, and hyperchaotic encryption on selected frames of video. The results were good, but the addition of optimization can still improve it.

3. Proposed Work

The contribution to the research is done in 3 phases.

   (i) Frame extraction and selection
   (ii) Embedding watermarking procedure
   (iii) Watermark extraction procedure
   (iv) Performance evaluation

3.1. Frame Extraction and Selection. The first phase in the proposed work is finding the suitable number of frames from extracted frames. The process of finding suitable frames in real time is carried out using scene change detection. The proposed work has applied frame selection technique on the video based on compressed domain which is used in various real-time applications such as broadcasting. The watermark cannot be embedded in all extracted frames because adding watermark will significantly increase the bit rate of video and thus will affect the processing time and increase complexity of code. So, frame selection is a very crucial step to decide which frames are selected from the set of extracted frames. Frame selection will depend on motion and motionless frames, and the selection criteria for a frame depend upon variations in the video. The significant changes in the video will have higher number of frames getting selected. Frame selection using a key has been conducted in most of the studies, but it does not serve as the best method as frame selection [27]. In the proposed work, an advanced approach with temporal sampling is done on Sobel–Feldman algorithm which can be used to identify the frame from the set of similar and different frames of video. The comparison of adjacent frames is done with one another to find frame difference which is calculated in equation (1). The value of the frame difference will decide whether the frame will be considered as the part of the same group or a different group. If the difference is large, then it will be considered as part of a different group. The parameter of decision will be taken as threshold which is calculated in equation (2); if frame difference is higher than threshold, the next frame will be the part of next group. The temporal sampling is performed which enhances the process of frame selection that gives better results compared to [3]. After different frames have been taken as part of different groups, the random key generator is used to select one similar image out of similar images. The work gives better results in comparison to Gu et al. [28] where random frame selection was carried out [29]. Frame difference can be represented as histogram difference which can be expressed as

\[ F_k = \sum_{K=1}^{T} H_k(m) - H_k(m+1), \tag{1} \]

where \( F_k \) is representing frame difference, \( H_k \) is the histogram value of \( k^{th} \) frame of level \( m \), and \( T \) is the number of levels of the histogram. The proposed work focuses on the value of histogram difference; the selection of frames is carried out on the basis of difference represented in equation (1), and the frames with less difference compared to threshold cannot be chosen, but the frames representing the large histograms difference value are selected in the next group. The grouping of similar images is based on scene change detection. The threshold is maintained to detect intensity histogram difference to calculate sudden transition amongst frames (in order to find larger frame difference). This scenario is expressed as

\[ M_b = \mu + \alpha \sigma, \tag{2} \]

where \( M_b \) = threshold value, \( s \) and \( \mu \) are the mean and standard deviation value of selected frame intensity histogram difference, And \( \alpha \) is the factor that varies from 2 to 6, and in our research, we have taken this value as 2.5. The temporal sampling has also enhanced the process of frame selection. The frame selection is always followed by frame extraction which can be done randomly by extracting frames after the timeline of certain seconds. The process is

![Figure 1: Types of watermarking techniques.](image-url)
carried by applying a timer on the given video for selection of frames that runs till the end of the video. The criteria of frame selection depend on the comparison of $F_k$ with $M_0$. The algorithm is given Algorithm 1.

The frame selection procedure is represented in Figure 2, where the number of frames gets extracted from the video. The grouping is done based on frame difference. The single frame is selected from a group, and after that the watermark embedding process is carried out. Algorithm 1 provides frame selection procedure which is the crucial part of research.

3.2. Watermark Embedding Procedure. The next step after selection of frames is to embed the encrypted watermark. The watermark is not applied directly on the frame; the transformation of the frame has to be done using a transform. The embedding of watermark on selected frames imposes the challenge of quality loss. Every selected frame has its own property, so the target is to embed the watermark so as to obtain high values of PSNR. The watermark is not embedded directly as the image is required to be transformed before embedding is done. Mostly research is done on DWT and DCT for transforming the frame [30], and the proposed technique focuses on transforming the frame into graph-based transform followed by SVD. The embedding factor to embed a watermark is calculated using equation (11). The combined technique of GBT-SVD does provide good values of the embedding factor which results in good values of PSNR but the same can be improved by adding optimization to the embedding factor, where the target value of PSNR is further improved, which results in less loss of quality after the watermark is embedded to the selected frame. The performance of the proposed technique is always tested when attacks are applied to the watermarked frame. High values of PSNR always indicate the good performance of the technique. Most of the research on embedding of watermark is carried out using discrete wavelet transform and discrete cosine transform. However, the proposed technique presented in the paper is hybrid combination of GBT-SVD-GWO-GA. The target is to get higher values of PSNR after embedding of watermark. The same can be addressed in the following sections. The proposed work applies graph-based transform followed by singular value decomposition on selected frames. Graph-based transform (GBT) is a transform the uses signal in the form of graph and produces better results in terms of adapting in signal structure of an image. This section introduces novel embedding technique based on the combination of GBT and SVD, further optimized by hybrid algorithm of grey wolf optimization and genetic algorithm. The additional security feature given in Section 3.2.3 based on hyperchaotic encryption adds to the functionality of proposed technique. The embedding of encrypted watermark is proposed in further sections.

3.2.1. Embedding Technique. Graph-based transform is a newly formed transform that is represented as $G = \{V, E, s\}$, where V and E are the vertices and edges of the graph and s represents the frame signal for graph G:

$$M(i, j) = \begin{cases} 
    \sum m_{i,j}, & \text{if } i = j, \\
    0, & \text{otherwise}, 
\end{cases}$$

(3)

where $m_{i,j}$ represents the weight of the edge. The degree matrix $D \in N \times N$ is a diagonal matrix:

$$K(i, j) = \begin{cases} 
    \sum m_{i,j}, & \text{if } i = j, \\
    0, & \text{otherwise}. 
\end{cases}$$

(4)

Then, the Laplacian graph matrix $L$ would be defined as

$$L = K - M,$$

(5)

where the operator $L$ is also known as Kirchhoff operator, which is represented as adjacency matrix $A$. Eigenvalue decomposition is done to a set of real nonnegative eigenvalues which are represented by $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_N\}$, and orthogonal eigenvectors are represented by $V = \{v_1, \ldots, v_N\}$, derived as

$$L = VAV^T.$$  

(6)

Decorrelation of the signal defined on the graph is done using eigenvectors:

$$C = V^Ts,$$  

(7)

$$A = E_A * S_A * (R_A)^T = \sum_{i=1}^{r} E_i * S_i * (R_i)^T,$$  

(8)

$$E_A = [e_1, e_2, e_3, \ldots, e_N],$$

$$R_A = [r_1, r_2, \ldots, r_N],$$

$$S_A = \begin{bmatrix}
S_1 & \cdots & N \\
\vdots & \ddots & \vdots \\
0 & \cdots & S_N
\end{bmatrix},$$

(9)

$$A = ESR^T,$$  

(10)

$$K' (i, j) = K (i, j) + aW (i, j).$$  

(11)

3.2.2. Incremental Optimization Method for Embedding Factor. The embedding factor in equation (11) is optimized using an incremental approach to grey wolf optimization. The major part in embedding is to take care of PSNR after embedding of watermark. The embedding factor of GBT and SVD calculated in equation (10) was taken as 0.02, and the quality loss is addressed after applying certain attacks. The proposed scheme in the given research achieves good results in terms of PSNR by optimizing the embedding factor by using the combined approach of grey wolf optimization and genetic algorithm. The grey wolf optimization model [11] optimizes the factor by finding the fitness value from the position of 3 wolves $\alpha$, $\beta$, and $\delta$ which provides the best solution, 2nd best solution, and 3rd best solution. The approach is to find a prey which is referred as PSNR in the watermarking scheme, thus in order to get the fittest
solution. The whole process is the search of prey, gradually updating the position so as to capture the prey faster. $a$ is the leader of the pack and rest of the wolves follow it. The position of grey wolves is updated by using formula given in equation (11) and the following equation:

\[
\overrightarrow{M} = |\overrightarrow{L} + \beta \overrightarrow{S_p(t)} - \delta \overrightarrow{S}(t)|, \tag{12}
\]

\[
\overrightarrow{S}(t + 1) = \overrightarrow{S_p(t)} - \overrightarrow{F} \times \overrightarrow{M}, \tag{13}
\]

where $\overrightarrow{M}$ = distance between grey wolf and a prey, $\overrightarrow{S}$ = coefficient vector, $t$ = number of iterations, $\overrightarrow{S_p}$ = position vector of prey, $\overrightarrow{S}$ = position vector of wolf, and $\overrightarrow{F}$ = balance convergence coefficient between prey and wolf. The calculation of coefficient vectors $\overrightarrow{F}$, $\overrightarrow{L}$ is done by equation (13) and the following equation:

\[
\overrightarrow{F} = 2 \overrightarrow{d} \times \overrightarrow{q} - \overrightarrow{d}, \tag{14}
\]

\[
\overrightarrow{L} = 2 \overrightarrow{q}, \tag{15}
\]

where $\overrightarrow{d}$ linearly decreases from 2 to 0, $\overrightarrow{q}$ and $\overrightarrow{d}$ are random vectors, and $\overrightarrow{F}$ decreases with decrease in $\overrightarrow{d}$. It means the wolf is very much near to the prey to be attacked. The grey wolves will update the positions by moving within search space to find optimal solution. The location updating is done by following equations (16)–(22):

\[
\overrightarrow{M}_a = |\overrightarrow{L} 1 \times \overrightarrow{S}_a - \overrightarrow{S}|, \tag{16}
\]

\[
\overrightarrow{M}_\beta = |\overrightarrow{L} 2 \times \overrightarrow{S}_\beta - \overrightarrow{S}|, \tag{17}
\]

\[
\overrightarrow{M}_\delta = |\overrightarrow{L} 3 \times \overrightarrow{S}_\delta - \overrightarrow{S}|, \tag{18}
\]

\[
\overrightarrow{S}(t + 1) = \overrightarrow{S}_a - \overrightarrow{F}_i \times \overrightarrow{M}_a, \tag{19}
\]

\[
\overrightarrow{S} 2 = \overrightarrow{S}_\beta - \overrightarrow{F} 2 \times \overrightarrow{M}_\beta, \tag{20}
\]

\[
\overrightarrow{S} 3 = \overrightarrow{S}_\delta - \overrightarrow{F} 3 \times \overrightarrow{M}_\delta, \tag{21}
\]

\[
\overrightarrow{S} (t + 1) = \frac{\overrightarrow{S} 1 + \overrightarrow{S} 2 + \overrightarrow{S} 3}{3}, \tag{22}
\]

where $\overrightarrow{M}_a$, $\overrightarrow{M}_\beta$, $\overrightarrow{M}_\delta$ = distance amongst wolves, $a$, $\beta$, and $\delta$ are wolves and the prey, $\overrightarrow{S} 1$, $\overrightarrow{S} 2$, and $\overrightarrow{S} 3$ represent parameters that determine position w.r.t to $a$, $\beta$, and $\delta$ wolves, and $\overrightarrow{S}(t + 1)$ = the positional vector after updating grey wolf.

Steps of GWO model of optimization:

1. First step is to generate the initial set of population randomly and initialize parameters $\overrightarrow{a}$, $\overrightarrow{F}$, $\overrightarrow{L}$.
2. Second step is to calculate the fitness value of the grey wolf individual and save the highest individual with highest fitness values $\overrightarrow{S}_a$, $\overrightarrow{S}_\beta$, $\overrightarrow{S}_\delta$.
3. Third step is to update the position parameter value as per equations (15)–(21) so as to obtain the next generation population and further updating value of $\overrightarrow{a}$, $\overrightarrow{F}$, $\overrightarrow{L}$.
4. Calculate fitness value of each individual of grey wolf and update $\overrightarrow{S}_a$, $\overrightarrow{S}_\beta$, $\overrightarrow{S}_\delta$.
5. Repeat Steps 2–4 until maximum iterations are reached and optimal solution is obtained.

Improved algorithm of global exploration improves convergence speed. The value of $a$ improves efficiency of algorithm. The value of $a$ does not impact any local convergence, thus solving large-scale multimodel problems will be hard. So, the improvement in the value of $a$ will improve optimal solution and the same is listed as
where $a_{\text{max}}$ = initial value of parameter $a$, $a_{\text{min}}$ = end value of parameter $a$, $i$ = iteration index, $i_{\text{max}}$ = maximum iterations, and $g$ = nonlinear adjustment coefficient.

The value of $A$ from equation (12) entirely depends upon value of $a$, where higher value of $A$ expands searching criteria of a better way. If $|A| < 1$, then the range is limited to a certain extent. The population evaluation directly impacts the efficiency of an optimized algorithm. To calculate the same, assume current population fitness is represented by $F_i$. Selection of individuals is done by the highest fitness values.

The value of $K$ represents total fitness and $P_i$ represents probability of selected individual given in equations (24) and (25), respectively:

$$K = \sum_{i=1}^{M-1} F_i \quad (i = 1, 2, \ldots, M - 1),$$

$$P_i = \frac{F_i}{\sum_{i=1}^{M-1} F_i} \quad (i = 1, 2, \ldots, M - 1).$$

Cumulative fitness value $K_i$ is calculated till children are consistent like parent, and the same is illustrated in equation (25):
\[ K_i = \frac{\sum_{j=0}^{d} F_{ij}}{K} \]  \hspace{1cm} (26)

To solve large-scale problems, the grey wolf optimization algorithm is combined with genetic algorithm, where entire population \( P \) is divided into \( t \times d \) subpopulations \( P_i, (i=1,\ldots,t \) and \( j=1,\ldots,d \). Crossover is applied to increase diversity of population. Linear crossover is applied in the proposed work. A random number is generated for each individual \( x_i \) for crossover population. The crossover probability is represented by \( P_c \) and \( c11 \) and \( c21 \) represent children generated from parents \( p11 \) and \( p22 \), where \( z \) represents the random number that ranges from 0 to 1. The crossover operation is derived from equations (27) and (28):

\[ c_i^1 = z p_i^1 + (1-z) p_i^2 \]  \hspace{1cm} (i = 1,2,\ldots,M),  \hspace{1cm} (27)

\[ c_i^2 = z p_i^2 + (1-z) p_i^1 \]  \hspace{1cm} (i = 1,2,\ldots,M).  \hspace{1cm} (28)

Mutation operator is applied on elite individuals in population. The individuals are represented as \( x_i \) with mutation probability \( P_m \). Selection of gene \( x_i \) is done with probability \( P_m \), and generation of new individuals \( x_{i1} \) is based on 2 criteria:

\[ \begin{bmatrix}
    x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\
    \vdots & \ddots & \vdots \\
    x_{p,1} & x_{p,2} & \cdots & x_{p,d}
\end{bmatrix} \]

\[ \begin{bmatrix}
    x_{1,1} & \cdots & x_{1,5} \\
    \vdots & \ddots & \vdots \\
    x_{5,1} & \cdots & x_{5,5}
\end{bmatrix} \]

\[ \begin{bmatrix}
    x_{(p-4),1} & \cdots & x_{(p-4),5} \\
    \vdots & \ddots & \vdots \\
    x_{p,1} & \cdots & x_{p,5}
\end{bmatrix} \]

\[ \begin{bmatrix}
    x_{(p-4),1} & \cdots & x_{(p-4),d} \\
    \vdots & \ddots & \vdots \\
    x_{p,1} & \cdots & x_{p,d}
\end{bmatrix} \]

\( x_{i1}^1 = \left\{ \begin{array}{ll}
  z + (u-l), & i = g, \\
  x_i, & i > g, i < g,
\end{array} \right\} \]

where \( l \) and \( u \) are lower and upper bounds to generate new individuals and \( z \) is a random number between 0 to 1. The hybrid optimization model is represented as follows.

The proposed model of GWO-GA optimizes the embedding factor to target high values of PSNR. The results obtained by using the incremental approach not only gets high values of PSNR alone but also gets high values of PSNR when different attacks such as Gaussian noise, sharpening, rotation, blurring, and JPEG compression are applied to watermarked frames. The algorithm of the incremental approach by GWO-GA is given in Algorithm 2. The approach starts with application of grey wolf optimization to find best search agents followed by genetic algorithm to further optimize to get high values of PSNR as a target function. The same is implemented in this research and optimization is applied to the embedding factor which is done using combined transform of GBT-SVD. Although the same transform is used to embed the watermark, using the optimization model provides additional feature to target high values and provides better results. The working of the hybrid grey wolf optimization and genetic algorithm (GWO-GA) is done in 2 phases where optimization of the embedding factor is done by using grey wolf optimization followed genetic algorithm. The areas to find best fitness parameter (PSNR) is done by calculating fitness of each search agent, updating the fitness value after every iteration. The process is continued till the best fitness value is calculated followed by the incremental approach of genetic algorithm which also searches for the best fitness value (PSNR), and the higher value of PSNR is directly proportional to quality. The process of selecting subpopulation is followed by mutation if the high PSNR value is not found out which is followed by crossover; then, convergence is done to get the best fitness value. Algorithm 2 provides the optimized embedding factor. The algorithm works better when compared with alone of grey wolf optimization and genetic algorithm. The embedding of watermark is done using graph-based transform (GBT) and singular-valued decomposition (SVD). During the embedding process, the embedding factor taken in equation (11) is optimized using this incremental approach so as to aim the higher value of PSNR, and the results obtained in Section 4 are a proof of this technique being robust, and the same approach is being tested on 6 videos and higher values of PSNR is obtained in all.

3.2.3. Encryption of Watermark before Embedding. The watermark embedded on selected frames is encrypted using hyperchaotic encryption to add security feature to the proposed technique [25]:

\[
\begin{align*}
    x &= a (y - x) + w \\
    y &= c x - y - x z \\
    z &= x y - b z \\
    w &= -y z + r w 
\end{align*}
\]

The value of \( x, y, z, \) and \( w \) calculated from the above equation will be used for encrypting the watermark image to be used in a frame. The standard values of \( a, b, \) and \( c \) were taken as per [25]. The second step is to convert \( R, S \) into \( x, y \) for column and row, respectively, to encrypt the watermarked frame:

\[
\begin{align*}
    X &= \text{mod} \left( \text{floor} \left( R + 100 \right) \times 105, i \right) + 1, \\
    Y &= \text{mod} \left( \text{floor} \left( S + 100 \right) \times 105, j \right) + 1.
\end{align*}
\]

The 3rd step is to interchange the coefficients of \( m^{th} \) row and \( x \left( m^{th} \right) \) row of image \( W \left( m = 1, 2, \ldots, \ldots, i \right) \) and \( N = 1, 2, \ldots, \ldots, j \):
Input: embedding factor
Output: optimized embedding factor for watermark embedding on alpha value

Begin
(1) Initialize parameters $a$, $F$, and $L$ and population size $N$, $P_c$, and $P_m$
(2) Initialize current population $P$
(3) $i=0$
(4) While $i < i_{\text{max}}$
(5) {
(6) Find the fitness value of all search agents
(7) $S_\alpha$ = Best Search agent for embedding factor
$S_\beta$ = 2nd Best Search agent for embedding factor
$S_\delta$ = 3rd Best Search agent for embedding factor
(8) for $i=1$ to $N$
(9) {
(10) Update search agent positions given in equation (21).
(11) }
(12) Assign $P_{1} \leftarrow P$ except $S_\alpha$
(13) for $i=1$ to $N-1$
(14) {
(15) Get new value of $P_2$ by the selection on new value of $P_1$.
(16) }
(17) Assign $P \leftarrow \text{new} P_2$, $S_\alpha$
(18) Generate subpopulation $P_s$ using partitioning on $P$.
(19) Select the individuals using crossover probability $P_c$.
(20) for $i=1$ to $N \times P_c$
(21) {
(22) Perform crossover on each search agent by equations (27) and (28).
(23) }
(24) Generate $S'_\alpha$, $S'_\beta$, and $S'_\delta$ by equation (29) on mutation probability $P_m$.
(25) }
(26) Return $S_\alpha$.
End

Algorithm 2: Hybrid GWO-GA approach for embedding factor.

\[ W(m,:) = W(x(m,:),:) \]  \hspace{1cm} (33)

\[ W1 = W, \]  \hspace{1cm} (34)

\[ W1(:,n) = W1(:,y(n)), \]  \hspace{1cm} (35)

where $W1$ is the encrypted watermark image to be embedded.

The encryption of a watermark image is represented as $W(i,j)$ where image size is represented as $m \times n$. The first step is generating the sequence of $R$ and $S$ using the Lorenz system. The security feature added here adds to the security feature by encrypting watermark before being embedded, thus making the technique more secure. Real-time applications such as broadcasting face security issues and copyright protection. The proposed technique combined with hyperchaotic encryption adds to security feature and also adds to copyright protection. In previous research done in [31], rail fence cipher is used as a security feature that is applied to binary image watermark. The watermark was ciphered with rail fence cipher so that exact retrieval of the watermark from any unauthorized person is not easy. Mostly research is carried out using transposition cipher [19] as adding higher security mechanisms such as AES and DES will increase the complexity of code. The research carried out by [25] is secure and provides better results compared to transposition ciphers. Hyperchaotic encryption is not adding to the complexity of code. Algorithm 3 provides embedding of the watermark inside selected frames using the optimized embedding factor provided by Algorithms 2 and 4 and defines the extraction of watermark. Figure 3 explains the updated embedding factor using a hybrid algorithm.

3.3 Watermark Extraction Procedure. This section describes the watermark extraction procedure so as to recover the watermark from the watermarked video. The extraction of a watermarked from the watermarked video is a reverse process of embedding when the watermark was embedded with the help of equation (10). The extraction of frames is followed by applying GBT and SVD and the extraction is calculated as per equation (32). This is followed by inverse GBT and inverse SVD, then
decryption is done using a key, and then the watermark is recovered. Figure 4 represents the watermarking embedding procedure, and Figure 5 represents the watermark extraction procedure:

\[ W(i,j) = \frac{(K'(i,j) - K(i,j))}{\alpha} \]  

where \( W(i,j) \) = extracted watermark, \( K'(i,j) \) = watermarked frame, and \( K(i,j) \) = selected frame.

3.4. Performance Evaluation. The performance evaluation of the watermarking technique is typically calculated in terms of quality parameters of the video and robustness against various attack scenarios such as Gaussian noise, sharpening, rotation, blurring, and JPEG compression. The parameters are as follows.

PSNR (peak signal-to-noise ratio): PSNR is a major quality parameter that differentiates the original and watermark frame on the basis of the mean square error. It is calculated by the following equation:

\[ \text{PSNR} = 10 \log_{10} \left( \frac{L^2}{MSE} \right) \]
Initialize grey wolf parameters- search agents, iterations, number of search agents, alpha, beta and delta positions $a, F, I$

Calculate fitness value of each search agents using equation
$S_\alpha$ = best search agent,
$S_\beta$ = 2nd best search agent,
$S_\delta$ = 3rd best search agent

Find the value of $S_\alpha, S_\beta, S_\delta$

$Y$

$i < i_{\text{max}}$

$N$

Finalized fitness value

Initialize population as $N_s$

Evaluate fitness value

Update the position of $S_\alpha, S_\beta, S_\delta$ using equation

Update $a, F, L$

Calculate fitness value using 3

Update $S_\alpha, S_\beta, S_\delta$

$i = i + 1$

Converge

$Y$

Return best fitness value

Stop

$N$

Select sub population $N_p$ such that $N_s >> N_p$

Mating pools

Crossover

Mutation

Replacement

Figure 3: Optimizing embedding factor using hybrid GWO-GA.
\[
\text{MSE} = \frac{1}{G \times H} \sum_{i=0}^{G-1} \sum_{j=0}^{H-1} (AI(i, j) - EI(i, j))^2 , \tag{37}
\]

where \(G\) and \(H\) = rows and columns of the image, \(AI(i, j)\) = selected frame, and \(EI(i, j)\) = watermarked frame.

\[
\text{PSNR} = 10 \log_{10} \frac{(255)^2}{\text{MSE}} , \tag{38}
\]

\[
\text{Average PSNR} = \frac{\sum_{i=0}^{n} \text{PSNR}_i}{n} . \tag{39}
\]

The average PSNR is the sum of PSNR of all selected frames divided by the number of frames. The objective of the proposed technique is obtaining high values of PSNR as embedding of the watermark which causes quality loss. Higher values of PSNR indicate the efficiency of the technique.

To find the similarity between the selected frame and watermarked frame, the quality parameters’ NC (normalized correlation), SSIM (structural similarity index measure), and BER (bit error rate) are calculated by the following equation:

\[
\text{NC} = \frac{\sum_{i=1}^{G} \sum_{j=1}^{H} A(i, j)E(i, j)}{\sum_{i=1}^{G} \sum_{j=1}^{H} E(i, j)^2} , \tag{40}
\]

\[
\text{SSIM}(m, n) = \frac{(2P_mP_n)(2K_{mn})}{(P_m^2 + P_n^2)(K_m^2 + K_n^2)} , \tag{41}
\]

\[
\text{BER} = \frac{1}{\text{PSNR}} . \tag{42}
\]

The numerical values of NC, SSIM, and BER lie in the range of [0, 1]. While SSIM and NC measure the similarity, so high values of them are preferred, and BER is inversely proportional to PSNR, so lower values indicate the efficiency of the technique.

4. Experimental Results

The results were calculated in MATLAB 2018b using i5 processor. The frame selection and embedding time is
dependent on the type of processor used. The compiled results are dependent upon watermark embedding and frame selection.

Total of 6 cif-encoded videos have been taken and frame selection mechanism is dependent on scene change detection, and using the frame selection algorithm mentioned in Section 3.1, the frames were selected from the videos. Akiyo did not have sufficient scene change detection, so the watermarking technique could not be applied on that, as the value of $F_k$ (frame difference) was not greater than $M_b$ (threshold), so no significant frames got selected from the video, and rest of the videos have significant frames selected as per frame selection algorithm in Section 3.1. Table 1 represents number of frames getting selected from the input video dataset. Pure storage video has got maximum frames selected. Some videos indicated major changes due to which more frames got selected from them. The datasets of the videos were obtained from Figures 6a–6i signify some selected frames from the dataset of videos. Along with these videos 2 binary watermarks and their encrypted versions are shown. The compressed domain videos taken in the research are the same type of videos used in broadcast application; to remove unauthorized access to these videos, the given videos are embedded with encrypted watermark that addresses the issues faced by real-time application. The encrypted watermark not only addresses security issues but also adds to copyright protection so as to achieve ownership identification. The experimental results were divided into certain phases. Starting with taking the input video in avi or mp4 file, this phase is followed by frame extraction. The relevant frames were selected from extracted frame which is followed by the embedding of watermark. Embedding of watermark is done using the proposed method followed by testing the validity of the technique by applying certain attacks. Higher values of PSNR and lower values of BER implicate the proposed technique to be efficient which leads to less loss in quality of output video. Every video will have different properties which mean frame selection in every video will be different. The same is demonstrated in this research, where different videos have different numbers of frames getting selected. Figures 7a–7e represent the watermarked frame after the watermarking process along with the binary watermark. Figures 8a–8d represent the plot of various videos and their respective quality parameters against no attack.

### 4.1. Experimental Tests for Quality Check

The experimental tests have been carried out using 2 binary watermarks, and the results are calculated by applying the watermark on the selected frame of the video. There are 2 types of calculation of results done: with attacks and without attacks. The following sections describe the performance of proposed techniques against both scenarios.

The value of quality parameters is taken as per comparison with original and watermarked frames. The higher value of PSNR, NC, and SSIM and lower value of BER suggests the proposed scheme is efficient when the watermark is imparted to the original host video. Table 2 compiles results of embedding of watermark 1 on selected frames. Tables 1 and 2 indicate there is not much of a difference when both watermarks are applied on the same set of videos. Binary watermark 1 is taken for carrying out the research work, and all the attacks in further sections are applied on the same watermark.

The embedding technique requires iterations to the process of combined transform of GBT-SVD with GWO and GA. To formulate results, we have taken iterations’ value as 10. Since the method is entirely dependent on processor speed, so taking the large value of iteration will be time-consuming, although the higher value of iteration will give better results. Table 3 compiles the results of watermarked frames for all videos. The results of average values of Figures 7a–7e and Figures 9a–9e depict the average values of quality parameters PSNR, SSIM, NC, and BER on selected frames from 5 videos. The high values of PSNR, SSIM, and NC and the less value of BER indicate less quality loss after embedding of both types of watermark. It means the visual quality is good overall. It is hard to distinguish between the original and watermarked frame. The results obtained in the proposed technique supersede the results obtained in [4, 8, 20, 21, 28, 32, 33]. Figures 10a–10d represent plots of average values of performance parameters using watermark 2.

### 4.2. Experimental Tests for Time Complexity

Tables 4 and 5 compile the processing time (in seconds) required to carry out frame selection, embedding, and extraction of both watermark for given set of videos. Figure 11 represents plots for comparison of quality parameters of both watermarks. The time is entirely based on processor requirements. The total time consumed depends on selection of frames from the video. Pure storage video has got 5 frames selected and the time for every frame varies from 20 to 35 seconds for every frame. Embedding time is dependent on the number of selected frames and iterations in which optimization is applied. As a standard, 10 iterations have been taken to obtain good value of fitness, i.e., PSNR, and more iteration will lead to higher value of embedding time. Total time consumed for every video is mentioned in Tables 4 and 5, and plots of the same are represented in Figure 12. The value of embedding time is similar for both watermark 1 and watermark 2. The value of embedding time is directly proportional to the number of selected frames. Total of 5 frames got selected from pure storage video; thus, total embedding time is the highest for the same video.

Plots in Figures 13a and 13b represent the embedding time for both watermarks. It can be seen in plots that there is

### Table 1: Comparison of videos in terms of frame selection.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Video name</th>
<th>No. of frames selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Akiyo</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Coastguard</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Foreman</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>News</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>Bowing</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>Pure storage</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S. No</th>
<th>Video name</th>
<th>No. of frames selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Akiyo</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Coastguard</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Foreman</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>News</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>Bowing</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>Pure storage</td>
<td>5</td>
</tr>
</tbody>
</table>

---

**Mathematical Problems in Engineering**
less difference in the values of quality parameters when both watermarks are compared.

The analysis was done using watermark 2 and results were obtained which are highlighted in Table 5.

4.3. Processing Attacks. The robustness of the proposed technique is tested against various attack scenarios such as Gaussian noise, sharpening, rotation, blurring, and JPEG compression. Series of experiments have been conducted to attack every watermarked frame to measure quality loss. The robustness of the technique entirely depends upon the values of PSNR, SSIM, NC, and BER. The proposed technique is designed to have real-time requirement in mind which is tested against these signal processing attacks. The processing attacks have been applied with some variance of the value of an attack to check robustness of the scheme. The detailed description is given as follows.

4.3.1. Gaussian Noise Attack. In Gaussian noise attack, random Gaussian sequence real values \([0, 0.0001, 0.001, 0.01]\) are added to the all frames of the watermarked video using watermark 1. The value 0 indicates no attack is applied. The results for 0 value were obtained from Table 2.
It can be identified from plots in Figures 14a–14d that values of quality parameters reduce after increasing Gaussian noise variance, although difference is less and negligible, indicating the proposed technique performs better in case of Gaussian noise.

It can be seen from plots in Figures 14a–14d as average PSNR, NC, and SSIM decrease with increase in the attack value and BER increases with increase in the attack value. The higher value of PSNR, NC, and SSIM justifies the robustness of the proposed technique. Figures 14a–14d compile plots of real-time testing by applying this attack with different values of Gaussian noise for all quality parameters using values \(0, 0.0001, 0.001, 0.01\) for all 5 videos taken in research work. PSNR, SSIM, NC, and BER values of all the videos are similar. The value of NA in graphs means no attack is applied and higher values of PSNR, NC, and SSIM will be obtained for NA. Lower values of BER are obtained for NA. Table 6 represents the value of quality parameters of watermarked frames after Gaussian noise with 0.0001 variance is applied to all. Results indicate the proposed technique to be robust against this attack.

4.3.2. Sharpening Attack. In sharpening attack, random sequence real values \(0, 0.0001, 0.001, 0.01\) are added to all frames of the watermarked video using watermark 1. It can be seen from plots in Figures 15a–15d as average PSNR, NC, and SSIM decrease with increase in the attack value, and BER increases with increase in the attack value. The higher value of PSNR, NC, and SSIM justifies the robustness of the proposed technique against sharpening attack. The value of sharpening attack for 0.0001 and 0.001 is approximately the same. Sharpening attack has negligible difference on the watermarked frame, and the higher value of NC indicates the robustness of the proposed technique. Figures 15a–15d compile results of sharpening attack for all quality parameters using values \(0, 0.0001, 0.001, 0.01\) for all 5 videos taken in research work. PSNR, SSIM, NC, and BER values of all the videos are similar. The value of NA in graphs means no attack is applied and higher values of PSNR, NC, and SSIM will be obtained for NA. Lower values of BER are obtained for NA. The value 0 indicates no attack is applied. The results for 0 value were obtained from Table 2. The sharpening attack is applied to highlight details of the image. The result
Table 2: Results after embedding of watermark 1 on selected frames.

<table>
<thead>
<tr>
<th>Video</th>
<th>PSNR (db)</th>
<th>SSIM</th>
<th>NC</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>52.5768</td>
<td>0.99993</td>
<td>0.9999</td>
<td>0.01902</td>
</tr>
<tr>
<td>Foreman</td>
<td>53.05685</td>
<td>0.99976</td>
<td>0.9999</td>
<td>0.01885</td>
</tr>
<tr>
<td>News</td>
<td>54.0078</td>
<td>0.99993</td>
<td>0.9999</td>
<td>0.01852</td>
</tr>
<tr>
<td>Bowing</td>
<td>53.289375</td>
<td>0.9996925</td>
<td>0.9999</td>
<td>0.01876</td>
</tr>
<tr>
<td>Pure storage</td>
<td>54.91922</td>
<td>0.999894</td>
<td>0.9999</td>
<td>0.01829</td>
</tr>
</tbody>
</table>

Table 3: Results after embedding of watermark 2 on selected frames.

<table>
<thead>
<tr>
<th>Video</th>
<th>PSNR (db)</th>
<th>SSIM</th>
<th>NC</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>55.0202</td>
<td>0.99977</td>
<td>0.99998</td>
<td>0.01817</td>
</tr>
<tr>
<td>Foreman</td>
<td>56.3271</td>
<td>0.999765</td>
<td>0.999985</td>
<td>0.01776</td>
</tr>
<tr>
<td>News</td>
<td>55.10973</td>
<td>0.99997</td>
<td>0.99999</td>
<td>0.01814</td>
</tr>
<tr>
<td>Bowing</td>
<td>53.49083</td>
<td>0.99971</td>
<td>0.999978</td>
<td>0.01869</td>
</tr>
<tr>
<td>Pure storage</td>
<td>55.41838</td>
<td>0.999845</td>
<td>0.999983</td>
<td>0.01807</td>
</tr>
</tbody>
</table>

Figure 8: Plots of PSNR, SSIM, NC, and BER of input videos against no attack. (a) Average PSNR vs. no attack using Watermark 1. (b) Average SSIM vs. no attack using Watermark 1. (c) Average NC vs. no attack using Watermark 1. (d) Average BER vs. no attack using Watermark 1.
Figure 9: (a–e) Watermarked frames from the video. (f) Selected binary watermark.

Figure 10: Continued.
calculation of the watermarked frame is represented in Table 7.

4.3.3. Rotation Attack. In rotation attack, a watermarked frame is rotated with an angle of \([0, 1, 90, 180]\) using watermark 1. The value 0 indicates no attack is applied. The results for 0 value were obtained from Table 2. It can be seen from plots in Figures 16a–16d as average PSNR, NC, and SSIM deteriorate with increase in the attack value and BER increases with increase in the attack value. PSNR, SSIM, NC, and BER values of all the videos are similar. The value of NA in graphs means no attack is applied and higher values of PSNR, NC, and SSIM will be obtained for NA. Lower values of BER are obtained for NA. The result calculation of these watermarked frames after rotation attack is represented in Table 8.

4.3.4. Blurring Attack. In blurring attack, random sequence real values \([0, 2, 0.5, 3.05, 4.05]\) are added to all frames of the watermarked video using watermark 1. The value 0 indicates no attack is applied. The results for 0 value were obtained from Table 2. It can be seen from plots in Figures 17a–17d as average PSNR, NC, and SSIM decrease with increase in the attack value and BER increases with increase in the attack value. The higher value of PSNR, NC, and SSIM justifies the robustness of the proposed technique against blurring attack. Plots in Figures 17a–17d compile results of blurring attack for all quality parameters using values \([0, 2, 0.5, 3.05, 4.05]\) for all 5 videos taken in research work. PSNR, SSIM,
NC, and BER values of all the videos are similar. The value of NA in graphs means no attack which is applied, and higher values of PSNR, NC, and SSIM will be obtained for NA. Lower values of BER are obtained for NA. Table 9 represents calculation of quality parameters after blurring attack.

4.3.5. JPEG Compression Attack. In JPEG Compression attack, values \{0, 98, 96, 94\} are taken, that is, applied to the all-watermarked video. The value 0 indicates no attack is applied. The results for 0 value were obtained from Table 2. It can be seen from plots in Figures 18a–18d as average PSNR, NC, and SSIM decrease with decrease in value of compression attack value and BER increases with decrease in attack value. Higher value of PSNR, NC, and SSIM justifies the robustness of the proposed technique against compression attack. Plots in Figures 18a–18d compile results of compression attack for all quality parameters using values \{0,
Figure 13: Plot of embedding time taken for watermarks 1 and 2. (a) Average embedding time for watermark 1. (b) Average embedding time for watermark 2.

Figure 14: Continued.
Figure 14: Plot of PSNR, NC, SSIM, and BER w.r.t Gaussian noise variance using watermark 1. (a) Average comparison of PSNR vs. Gaussian noise variance. (b) Average comparison of NC vs. Gaussian noise variance. (c) Average comparison of SSIM vs. Gaussian noise variance. (d) Average comparison of BER vs. Gaussian noise variance.

Table 6: Results after applying Gaussian noise attack on watermarked frames using watermark 1 using value 0.0001.

<table>
<thead>
<tr>
<th>Video</th>
<th>PSNR (db)</th>
<th>SSIM</th>
<th>NC</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>39.7634</td>
<td>0.9993</td>
<td>0.99904</td>
<td>0.025149</td>
</tr>
<tr>
<td>Foreman</td>
<td>39.8525</td>
<td>0.976145</td>
<td>0.999045</td>
<td>0.025093</td>
</tr>
<tr>
<td>News</td>
<td>39.8598</td>
<td>0.96966</td>
<td>0.99921</td>
<td>0.025088</td>
</tr>
<tr>
<td>Bowing</td>
<td>39.83645</td>
<td>0.959358</td>
<td>0.998818</td>
<td>0.025103</td>
</tr>
<tr>
<td>Pure storage</td>
<td>39.78444</td>
<td>0.961182</td>
<td>0.998698</td>
<td>0.025136</td>
</tr>
</tbody>
</table>

Figure 15: Continued.
Figure 15: Plot of PSNR, NC, SSIM, and BER w.r.t sharpening attack using watermark 1. (a) Average comparison of PSNR vs. sharpening attack variance. (b) Average comparison of NC vs. sharpening attack variance. (c) Average comparison of SSIM vs. sharpening attack variance. (d) Average comparison of BER vs. sharpening attack variance.

Table 7: Results after applying sharpening noise attack on watermarked frames using watermark 1 for value 0.0001.

<table>
<thead>
<tr>
<th>Video</th>
<th>PSNR (db)</th>
<th>SSIM</th>
<th>NC</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>50.054</td>
<td>0.99887</td>
<td>0.99994</td>
<td>0.01902</td>
</tr>
<tr>
<td>Foreman</td>
<td>49.6947</td>
<td>0.998375</td>
<td>0.99993</td>
<td>0.020126</td>
</tr>
<tr>
<td>News</td>
<td>49.78257</td>
<td>0.99844</td>
<td>0.999933</td>
<td>0.020087</td>
</tr>
<tr>
<td>Bowing</td>
<td>49.9124</td>
<td>0.99752</td>
<td>0.99992</td>
<td>0.020036</td>
</tr>
<tr>
<td>Pure storage</td>
<td>50.07826</td>
<td>0.997752</td>
<td>0.99991</td>
<td>0.019976</td>
</tr>
</tbody>
</table>

Figure 16: Continued.
Figure 16: Plot of PSNR, NC, SSIM, and BER against rotation attack using watermark 1. (a) Average comparison of PSNR vs. rotation attack variance. (b) Average comparison of NC vs. rotation attack variance. (c) Average comparison of SSIM vs. rotation attack variance. (d) Average comparison of BER vs. rotation attack variance.

Table 8: Results after applying rotation attack on watermarked frames using watermark 1.

<table>
<thead>
<tr>
<th>Video</th>
<th>PSNR (db)</th>
<th>SSIM</th>
<th>NC</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>8.3042</td>
<td>0.09946</td>
<td>−0.399</td>
<td>0.12042</td>
</tr>
<tr>
<td>Foreman</td>
<td>8.1601</td>
<td>0.31619</td>
<td>−0.17586</td>
<td>0.12255</td>
</tr>
<tr>
<td>News</td>
<td>8.488167</td>
<td>0.231913</td>
<td>−0.10994</td>
<td>0.117817</td>
</tr>
<tr>
<td>Bowing</td>
<td>11.10585</td>
<td>0.221873</td>
<td>0.067068</td>
<td>0.090324</td>
</tr>
<tr>
<td>Pure storage</td>
<td>11.98104</td>
<td>0.477626</td>
<td>0.132206</td>
<td>0.08598</td>
</tr>
</tbody>
</table>

Figure 17: Continued.
Figure 17: Plot of PSNR, NC, SSIM, and BER w.r.t. blurring attack variance using watermark 1. (a) Average comparison of PSNR vs. blurring attack variance. (b) Average comparison of NC vs. blurring attack variance. (c) Average comparison of SSIM vs. blurring attack variance. (d) Average comparison of BER vs. blurring attack variance.

Table 9: Results after applying blurring attack on watermarked frames using watermark 1.

<table>
<thead>
<tr>
<th>Video</th>
<th>PSNR (db)</th>
<th>SSIM</th>
<th>NC</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>39.5519</td>
<td>0.99313</td>
<td>0.99897</td>
<td>0.025283</td>
</tr>
<tr>
<td>Foreman</td>
<td>43.4012</td>
<td>0.99725</td>
<td>0.9996</td>
<td>0.023055</td>
</tr>
<tr>
<td>News</td>
<td>36.00073</td>
<td>0.992993</td>
<td>0.998083</td>
<td>0.027777</td>
</tr>
<tr>
<td>Bowing</td>
<td>40.97863</td>
<td>0.993015</td>
<td>0.99908</td>
<td>0.024403</td>
</tr>
<tr>
<td>Pure storage</td>
<td>37.82222</td>
<td>0.987412</td>
<td>0.997874</td>
<td>0.026453</td>
</tr>
</tbody>
</table>

Figure 18: Continued.
Table 10: Results after applying JPEG compression attack on watermarked frames using watermark 1.

<table>
<thead>
<tr>
<th>Video</th>
<th>PSNR (db)</th>
<th>SSIM</th>
<th>NC</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>46.5895</td>
<td>0.9968</td>
<td>0.99982</td>
<td>0.021464</td>
</tr>
<tr>
<td>Foreman</td>
<td>44.71645</td>
<td>0.995785</td>
<td>0.999695</td>
<td>0.022371</td>
</tr>
<tr>
<td>News</td>
<td>42.8397</td>
<td>0.993483</td>
<td>0.9996</td>
<td>0.023344</td>
</tr>
<tr>
<td>Bowing</td>
<td>46.08708</td>
<td>0.994165</td>
<td>0.999745</td>
<td>0.021699</td>
</tr>
<tr>
<td>Pure storage</td>
<td>46.21462</td>
<td>0.995892</td>
<td>0.999716</td>
<td>0.021671</td>
</tr>
</tbody>
</table>

Figure 18: Plot of PSNR, NC, SSIM, and BER w.r.t JPEG compression attack variance using watermark 1. (a) Average comparison of PSNR vs. JPEG compression attack variance. (b) Average comparison of NC vs. JPEG compression attack variance. (c) Average comparison of SSIM vs. JPEG compression attack variance. (d) Average comparison of BER vs. JPEG compression variance.

Figure 19: Continued.
98, 96, 94} for all 5 videos taken in research work. PSNR, SSIM, NC, and BER values of all the videos are similar. The value of NA in graphs means no attack is applied and higher values of PSNR, NC, and SSIM will be obtained for NA. Lower values of BER are obtained for NA.

Result calculation of all parameters after JPEG compression is done using Table 10. Figures 19(a–d) describe plots of all the attacks applied on input dataset.

The results compiled from plots in Figures 19(a–d) suggest that the proposed technique achieves good results when all 5 attacks are applied on watermarked video. Higher value of quality parameters is achieved of all videos. The proposed novel technique is robust against all attacks applied and achieves higher values against various attack scenarios.

4.4. Analysis of Compressed and Uncompressed Data. The proposed technique was applied on both compressed and uncompressed video, but most of the analysis was done on uncompressed video as it takes less time to process using optimization algorithm. The same was analyzed with plots in
FIGURES 20A AND 20B. ALTHOUGH PSNR OF UNCOMPRESSED VIDEO WAS MORE, OTHER FACTORS SUCH AS SSIM, NC, AND BER WERE ALMOST SIMILAR.

5. CONCLUSION

In this paper, (1) frame selection was implemented using scene change detection along with temporal sampling and (2) approach was done to embed the encrypted watermark using hybrid combination of graph-based transform along with singular value decomposition (SVD) combined with optimized approach of grey wolf optimization and genetic algorithm which covers various areas such as broadcasting and copyright protection. The proposed technique focuses on real-time requirements of the video such as quality loss after embedding of the watermark. The quality parameters PSNR, SSIM, NC, and BER are used to evaluate effectiveness of the scheme against various attack scenarios. The higher values of PSNR, SSIM, and NC and lower BER values indicate that the proposed embedding approach achieves good visual quality and robustness. The embedding watermarking technique was found to be robust against practical video processing attacks: Gaussian noise, sharpening, rotation, blurring, and JPEG compression. Various OTT platforms do cover compressed domain of videos and our proposed technique covers all aspects of security for broadcasting application, since the watermark was encrypted with hyperchaotic encryption and successful recovery of the watermark after extraction suggests the efficiency of watermarking. The efficiency of the technique can be improved with advanced optimization methods and machine learning models which will be addressed in future work.

DATA AVAILABILITY

The data used to support the findings of the study are available from the corresponding author upon request.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES


