DATA EMBEDDING AND EXTRACTION IN SCRAMBLED VIDEO USING MACHINE LEARNING

by

Afaf Eltayeb Mohamedelbagir Ahmed

A Thesis Presented to the Faculty of the
American University of Sharjah
College of Engineering
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in
Computer Engineering

Sharjah, United Arab Emirates

December 2020

Declaration of Authorship

I declare that this thesis is my own work and, to the best of my knowledge and belief, it does not contain material published or written by a third party, except where permission has been obtained and/or appropriately cited through full and accurate referencing.

Signature.....Afaf Eltayeb.....

Date.....21 December 2020.....

Approval Signatures

We, the undersigned, approve the Master's Thesis of Afaf Eltayeb Mohamedelbagir Ahmed.

Title: Data Embedding and Extraction in Scrambled Video using Machine Learning.

Date of Defense: 14 December 2020

Name, Title and Affiliation	**Signature**
Dr. Tamer Shanableh Professor, Department of Computer Science and Engineering Thesis Advisor	
Dr. Gerassimos Barlas Professor, Department of Computer Science and Engineering Thesis Committee Member	
Dr. Usman Tariq Assistant Professor, Department of Electrical Engineering Thesis Committee Member	
Dr. Fadi Aloul Head Department of Computer Science and Engineering	
Dr. Lotfi Romdhane Associate Dean for Graduate Studies and Research College of Engineering	
Dr. Sirin Tekinay Dean College of Engineering	
Dr. Mohamed El-Tarhuni Vice Provost for Graduate Studies Office of Graduate Studies	

Acknowledgements

Foremost, I would like to express all the gratitude and thank to Great Allah for his uncounted blessings, from which being who i am today.

I would like to express my deepest appreciation to my thesis advisor; Dr. Tamer Shanableh, for his guidance, dedication, scientific and moral support. Without the persistent feedback and support he provided me with, this work would not have been possible.

Besides, I would like also to thank my thesis committee members; Dr. Gerassimos Barlas and Dr. Usman Tariq for their valuable comments and advices.

At last, my sincere thank to the American University of Sharjah and the Department of Computer Engineering for granting me Graduate Teaching Assistantship (GTA) to pursue my studies, and to all the individuals in the university, from the founders and the academic members to the forgotten workers who work day and night to provide us with a healthy, well-prepared environment suitable for scientific production.

Dedication

To my dear parents, whose affection, sacrifice and persistent encouragement make me achieve such success and honour. . . To them for being my role models in this life.

To my grandmother, for her love, care and prayers day and night.

To my uncles, who have always been like a father to me.

To my lovely sisters, for always being there for me.

To my family . . . To my friends.

Abstract

Data embedding in videos and images has various important applications such as digital rights management (DRM), content authentication, copyright protection, error resiliency and concealment as well as law enforcement. With the high possibility of illegal access and unauthorized content manipulation in shared storage platforms such as cloud data centers and with the risk of encountering different types of attacks during network transmission, videos and other sensitive data are usually transmitted and stored in an encrypted form. Accordingly, the need for data hiding techniques that operate directly on the encrypted video domain has emerged. This work proposes a novel data hiding scheme in encrypted video streams where scrambling and data embedding are performed simultaneously at the encoder side by rotating the motion vectors of the cover video. Then a machine learning solution is proposed at the decoder side to classify the motion vectors to rotated/ unrotated, extract the hidden information bits and reconstruct the original cover video. A sequence-dependent approach is applied where the first part of the video is used for training and model generation. The proposed system is composed of two phases: firstly, the training phase where the model is trained to distinguish between the correctly reconstructed macroblocks and the macroblocks reconstructed using rotated motion vectors. Secondly, the testing phase in which the trained model is applied to identify which of the candidate macroblocks are the ones associated with the true motion vectors. Once the true motion vectors are identified, they are compared to the ones received in the bit stream and thus the embedded bits are extracted, and the video is reconstructed. Experiments are conducted on a number of well-known video sequences after compressing them once with the Moving Pictures Expert Group-2 video codec standard and then with the High-Efficiency Video Coding standard. A detailed analysis is provided based on the macroblock type, the number of motion vectors and the type of the encoding sequence. Lastly, the proposed solution is evaluated in terms of classification accuracy, embedding capacity and reconstruction quality.

Keywords: ***data embedding, data extraction, scrambled video, machine learning, sequence-dependent approach.***

Table of Contents

List of Figures

List of Tables

Abbreviations

B-Frame - Bi-directional Frame

BAC - Binary Arithmetic Coding

CABAC - Context Adaptive Binary Arithmetic Coding

CAVLC - Context Adaptive Variable Length Coding

CU - Coding Unit

DCT - Discrete Cosine Transform

EBER - Extracted Bit Error Rate

GOP - Group Of Pictures

HEVC - High Efficiency Video Coding

HR - Hiding Ratio

I-Frame - Intra Frame

IPM - Intra Prediction Mode

LSB - Least Significant Bit

MB - Macro Block

MC - Motion Compensation

ME - Motion Estimation

MPEG - Moving Picture Experts Group

MSB - Most Significant Bit

MSE - Mean Square Error

MV - Motion Vector

MVD - Motion Vector Difference

OOB - Out Of Bag

P-Frame - Predicted Frame

PE - Prediction Error

PSNR - Peak Signal to Noise Ratio

QS - Quantization Scale

RBF - Radial Basis Function

RDH - Reversible Data Hiding

RF - Random Forest

RRBE - Reserve Room Before Encryption

SAD - Sum of Absolute Difference

SVM - Support Vector Machine

VLC - Variable Length Coding

Chapter 1. Introduction

1.1 Introduction

Data embedding is the technique by which some external message bits are hidden into a host content before being transmitted. Different types of interactive media have been used as hosts or carriers for hiding messages such as texts, audios, images as well as videos [1]. However, the size of a video and its large amount of redundant data make it the most eligible multimedia type for data embedding [1, 2]. Up to date, data hiding has been employed for various applications such as watermarking, copyright protection, content authentication, error resiliency and concealment, military services and law enforcement [2, 3].

As mentioned in [3], data embedding process may lead to visual distortion and excessive bitrate in the host compressed video. Such distortions and increased bitrate are required to be minimal while maximizing message payload. Consequently, various data hiding techniques have emerged and are competing to achieve these requirements.

Regardless of the purpose of data concealment and depending on the phase at which message bits are embedded, data hiding in digital videos can be classified into three categories: intra-hiding, pre-hiding and post-hiding [2]. The term intra-hiding is used when the message bits are embedded in the compressed domain and in turn is divided into different categories based on the compression stage such as motion vectors, intra-prediction modes, transform coefficients and quantization scales. Pre-hiding methods are mainly focused on transform and spatial domains of the raw video while in post-hiding techniques, message bits are embedded into the compressed bit stream [2].

Different data embedding techniques are evaluated according to three essential factors: hiding capacity, visual quality of the host video and robustness [1, 2]. Higher hiding capacity means that more message bits can be hided in the digital video, but this might result in higher distortion in the quality of the carrier video and larger increase in its size. A data embedding technique is considered to be robust if the receiver can correctly extract the embedded message bits from the received bit stream without encountering any errors. In general, Higher embedding capacity and robustness may

lead to lower imperceptibility of the compressed video. Thus, a good trade-off between these three factors is required to achieve the best performance in data embedding applications [2].

Recently, due to the rapid evolution in internet technology and mobile applications, there has been an increasing demand on the high computational power and large-scale storage provided by cloud data centres. Since cloud resources can be exposed to various types of attacks and are vulnerable to untrustworthy system administrators, privacy protection and security related issues have been a major concern for cloud service providers. In an attempt to address this issue, it is preferable to store videos and other sensitive data in encrypted form. Sometimes, a data manager may need to embed some additional data into video such as authentication data and ownership information to be able to verify its integrity and prevent unauthorized access. However, since the data hider in most cases may not have the encryption key, embedding has to be performed in the encrypted domain without knowing the original video content [4, 5]. Consequently, there has been a growing research recently on data hiding techniques that perform directly in the encrypted domain.

Likewise, a wide area of the ongoing research has been focusing on developing machine learning approaches and applying them to different research and industrial applications. One of the fields where machine learning has gained a considerable interest is image and video processing. Two types of training and model generation in video applications have been mentioned in the literature, namely, video-dependent approach where the first part of the video is used for training and model generation as in [6, 7], and video-independent approach where training and model generation is performed on different video sequences than the ones being tested [8].

1.2 Motivation and Problem Statement

Data hiding in videos is of high importance for various applications such as digital rights management (DRM), copyright protection, content authentication and error resiliency and concealment. With the high risk of unauthorized access in shared storage environments such as cloud and possibility of encountering different hacking

attacks during network transmission, videos are normally stored and transmitted in an encrypted form. As a result, many techniques have been proposed in the literature for hiding data in the encrypted video domain. However, most of these techniques perform video encryption and data hiding separately, i.e. they firstly encrypt the video, then embed the desired information into the encrypted video afterwards. Moreover, to the best of our knowledge, none of the existing techniques applied machine learning approaches to data extraction and video recovery.

Therefore, this work aims to contribute to the existing literature by proposing the following solutions:

- A simultaneous video scrambling and data embedding approach by rotating the motion vectors of the cover video at the encoder according to the information bits to be hidden.
- A machine learning approach to distinguish the rotated motion vectors from unrotated ones and consequently being able to extract the message bits and recover the original cover video at the decoder.

1.3 Thesis Outline

The rest of the thesis is organized as follows: Chapter 2 provides an overview of the existing literature on data embedding and extraction in scrambled videos and images and a brief background on the basic concepts related to data embedding and video codec standards as well as machine learning. Chapter 3 describes the proposed system model. Experiments and results are presented in Chapter 4 before the report is finally concluded in Chapter 5.

Chapter 2. Literature Review

In comparison with the traditional data hiding methods in the plain video domain, data embedding in encrypted videos has become more desirable as it copes with real-life applications and scenarios especially with the emergence of cloud computing technology. Machine learning techniques are also becoming more popular in most scientific researches. In this chapter, we firstly introduce the three major factors that should be considered by any data embedding scheme whether in encrypted or plain domain. Then, the basic concepts of the block-based video coding standards such as MPEG2 and HEVC are explained. After that, some existing methodologies of hiding information in encrypted videos as well as encrypted images are reviewed. Finally, the theoretical concepts of the main machine learning approaches used in this work are introduced.

2.1 Performance Assessment

Data embedding process may lead to visual distortion and excessive bitrate in the host compressed video. Such distortions and increased bitrate are required to be minimal while maximizing message Payload [3]. Consequently, various data hiding techniques have emerged and are competing to achieve optimal performance. Three main factors must be considered in any data hiding scheme: visual quality of the host video, robustness of the embedded message and hiding capacity [2].

2.1.1 Video quality. The process of concealing data in a video can result in either slight or severe degradation in the visual quality of the host video. Different assessment metrics have been utilized to evaluate the effect on visual quality such as the Peak Signal to Noise Ratio (PSNR), which reflects the difference between the embedded frames and the original frames and is calculated using the following formula [1, 2]:

$$\begin{aligned} \text{PSNR} &= 10\log_{10}\left(\frac{MAX_A^2}{\text{MSE}}\right) \quad \text{(dB)} \\ \textit{where, } \text{MSE} &= \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}\left[A(i,j)-B(i,j)\right]^2}{a\times b} \end{aligned} \tag{2.1}$$

where A and B denote the original and embedded frames respectively, a and b denote the size of each frame in the video and MAX is the largest pixel value in the original frame.

2.1.2 Embedding capacity. Embedding capacity is defined as the number of message bits that can be embedded into the host video. Hiding ratio (HR) is the measure used to evaluate the performance of data hiding methods in terms of embedding capacity, and is calculated using the following formula [2]:

$$HR = \frac{Size\ of\ the\ embedded\ message}{Size\ of\ the\ video} \times 100\% \tag{2.2}$$

2.1.3 Robustness or data extraction accuracy. A data hiding scheme is said to be robust if it can extract the whole message correctly and error-free at the receiver side. Robustness measures two aspects; the method's strength against network attacks and transmission errors and the reversibility of the data hiding scheme itself. The extracted-bit error rate is one measure used in the literature to evaluate the performance of the different data concealment methods in terms of data extraction ability, and is defined by the following equation [2]:

$$\text{EBER} = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}[M(i,j) \oplus \bar{M}(i,j)]}{a \times b} \tag{2.3}$$

where $M(i,j)$ and $\bar{M}(i,j)$ denote the original and extracted message respectively, $a \times b$ is the number of the embedded bits.

2.2 Block based Video Coding Schemes

The block-based video coding standards such as MPEG2, H.264/AVC and HEVC are based on a video compression procedure that exploits the high level of spatial and temporal redundancy in the original video sequences [9]. At first, the temporal correlation is reduced by computing the similarities between consecutive frames and transmitting the difference between them rather than the actual frame, resulting in fewer transmitted bits due to the small values representing the differences. A process called 'Mo-

tion Estimation' (ME) is performed between each two successive frames on a 16×16-block basis known as 'Macroblocks' (MBs) in MPEG2 standard, to find the best match location of a MB relative to the reference frame, which results in two values (horizontal and vertical components) per a macroblock called 'Motion vector' (MV) indicating the best match location. For HEVC, the motion estimation process can take place for even smaller sub-block dimensions known as 'Coding Units' (CUs) [10]. Then, a 'Motion Compensation' (MC) process takes place which copies the best match MB/ CU in the previous frame to the location of the current MB/ CU, in order to subtract each MB/ CU from its best match MB/ CU of the reference frame to further compress the video. After that, each frame is subtracted from its previous motion compensated frame producing what is known as 'Prediction Error' (PE). The spatial redundancy is reduced by further processing the prediction errors by applying the discrete cosine transformation (DCT), after which a scalar quantization is performed by dividing the DCT coefficients by specific 'Quantization Scales' (QSs) that fall in the range [1,31] for MPEG2 and [0,51] for HEVC. Eventually, Variable Length Coding (VLC) is performed and the output video bit stream is produced [9, 10].

Three types of frames exist in both MPEG2 and HEVC video standards, namely, I-frames, P-frames and B-frames and two coding schemes are defined, namely, Intra-frame coding and Inter-frame Coding. I-frames are intra coded, which means no subtraction from reference frames is performed, and the DCT, quantization and VLC are applied to the original frame. Both P-frames and B-frames are inter-coded where the DCT, quantization and VLC are applied to their prediction errors. However, the former is forward predicted (subtracted from the previous frame), while the latter can be forward predicted, backward predicted (subtracted from the future frame) or both (bi-directional). For I-frames all the MBs/ CUs are intra-coded while in P-frames the coding decision is performed on MB/ CU-basis according to some criteria, meaning that a macroblock/ coding unit in a P-frame can be either intra-coded or forward-predicted. Likewise, a MB/ CU in a B-frame can be intra-coded, forward-predicted, backward-predicted or bi-directional-predicted [9, 11].

2.3 Data Embedding in Encrypted Domain

In [4], a novel scheme was proposed to embed secret data in the encrypted H.264/AVC bit stream. Since encrypting the whole video bit stream is not practical due to computational cost and real time constraints, only sensitive portions of data were encrypted. Typically, the codewords of motion vector differences, the codewords of residual coefficients and the codewords of intra-prediction modes were encrypted using standard stream ciphers with encryption keys in combination with the Exp-Golomb entropy coding and Context-adaptive variable-length coding (CAVLC). The proposed encryption scheme guaranteed both cryptographic security via the used secure cipher and perceptual security via the chosen codewords to be encrypted. Additional data were then embedded in the encrypted video without knowing its original content by substituting the CAVLC codewords of Levels in P-frames only in such a way that the new codeword has the same length as the original one, thus video size is preserved while degradation in visual quality is minimal. Data can then be extracted either from the encrypted or decrypted domain depending on the application [4].

A similar approach was used in [5] to hide data in the encrypted compressed domain of H.264/AVC as well. However, while only the codewords of Levels with suffix-length of 2 or 3 were used for data hiding in [4] and the codewords with suffix-length equal to 1 were not used due to the unavailability of two codewords with the same size, [5] has shown a greater exploitation of the embedding space by proposing a paired codeword substitution scheme to include the case of the unity suffix-length. In addition, a multiple-based notational system was adopted for data hiding when suffix-length of the codewords is greater than 2 as opposed to the single codeword substitution used in [4], thus resulting in higher embedding capacity.

In [12], a two-dimensional histogram shifting scheme was used to embed data in the non-zero quantized AC coefficients of P-frames. Unlike [4] and [5], encryption and embedding are performed during the encoding process, not in the bit stream. The motion vector difference (MVD), the intra-prediction modes (IPM) and the sign bits of the coefficients of residual blocks are first encrypted via secure stream ciphers. Then, the range of non-zero residuals is divided into coefficient pairs and a specific mapping

referred to as 2D histogram shifting is applied allowing one or two bits of data to be embedded per a coefficient pair.

Partial scrambling is done in [13, 14] where only certain portions referred to as "privacy areas" are scrambled to prevent general users from accessing them, while the remaining of video is transmitted normally. Examples of privacy regions are people's faces, vehicles' licenses and trademarks. In [13], the motion vector differences, the quantized coefficients and the intra-coding modes of only 4×4 blocks associated with the privacy regions are scrambled, then data embedding takes place to hide the original unscrambled data in the bit stream such that only the authorized users can extract them and recover the scrambled regions. After that, a second phase of scrambling is applied by changing the sign bits of coding data to introduce more noise. The main idea behind modifying only the 4×4 intra-prediction modes is their many details compared to the smooth regions of the 16×16 intra-predicted blocks, thus the effect of scrambling on visual quality is more severe. Besides, avoiding the increase in file size that can be caused when too many data that show weak response to scrambling are modified is another motivation. Video confidential data is then embedded into the non-zero AC quantized coefficients during the encoding process. First, the coordinates of the privacy areas of each frame are hided in the beginning of the video in order for the authorized decoder to be able to locate them, followed by the original MVDs, IPMs and quantized coefficients.

A scheme for data embedding in partially encrypted H.264/AVC videos using CABAC bin-string substitution was proposed in [15]. Compared to CAVLC-based encryption, binary arithmetic coding (BAC) is extremely sensitive to errors and modifying even a single bit can influence the compliance of the whole bit stream. Therefore, selective encryption was applied to CABAC bin-strings which are the inputs to the binary arithmetic coder, rather than applying it to the final bit stream. Typically, the sign bits of the non-zero quantized coefficients of I and P frames and the sign bits of MVDs were encrypted by XORing with random binary sequences generated by standard stream ciphers. After that, data embedding takes place by substituting bin-strings of absolute values of motion vector differences (abs-MVD) and bin-strings of absolute values of non-zero QTC levels (abs-level) which are grouped into two groups C_0 and C_1 corre-

sponding to an embedded bit “0” and “1” respectively. Only bin-strings of abs-level in P-frames were used for hiding data while keeping those of I-frames unchanged to avoid distortion propagation. Data extraction was then performed in both encrypted and decrypted domain considering different application scenarios. Results have shown that both encryption and data hiding schemes have preserved the bitrate and the degradation in video quality resulted from data concealment is quite small.

The work in [16] has also used CABAC bin-string substitution to hide information in encrypted H.264/AVC videos. However, besides sign bits of MVDs and sign bits of luma QTCs, the sign bits of chroma QTCs and the sign bits of intra-prediction modes (IPM) were also encrypted, providing more scrambling and enhancing the security level of the video. Compared to the single bin-string substitution used in [15] where only one bit of data can be embedded per bin-string of abs-level, higher hiding capacity was obtained as a result of embedding two bits per bin-string when abs-level is greater than 17 since there are more than two bin-string that have the same length [16]. In addition, while only the bin-strings associated with abs-MVD greater than 32 were utilised to hide information bits in [15], the embedding space was extended by using all bin-strings of abs-MVD that are greater than 8 [16].

A separable reversible data hiding scheme for encrypted high efficiency video coding (HEVC) video is proposed in [17]. First, the signs of motion vector differences and the signs of residual coefficients are encrypted by applying XOR operation with encryption keys generated by standard stream cipher RC4. Also, the amplitudes of motion vector differences are encrypted by swapping their vertical and horizontal components when the key bit equals 1 and remain unchanged otherwise. After that, data are embedded into the amplitudes of nonzero AC coefficients. The proposed scheme utilizes TUs of all sizes supported by HEVC to embed data, thus higher payload can be achieved.

Likewise, data embedding in encrypted images has got a considerable research interest in the last years, especially in the reversible data hiding (RDH) domain. Since the traditional RDH methods cannot guarantee reversibility in the encrypted domain due to the huge change in the entropy of the original image after being encrypted, most researchers have focused recently on exploring new schemes that emphasise reversibility in encrypted domain. Both [18] and [19] applied homomorphic encryption on image

prior to data embedding. Their idea behind using this encryption is that there is no need to vacate room for the additional data before encryption. A pixel value ordering (PVO) technique in joint with prediction error histogram shifting was used in [18] to conceal additional information. The encrypted image is divided into non-overlapping equal-sized blocks. Then each block is sorted in an ascending order and a prediction error is obtained by subtracting the two top values, after which the value of the prediction error is used to modify the top-valued pixel either to embed a bit or otherwise shifted by one. This work has guaranteed both separability and reversibility meaning that both the hidden bits and the original image can be recovered losslessly. Also, it does not cause data expansion which means that the size of image remains the same after encryption. A Paillier homomorphic encryption was applied to the original image pixels after being divided into three components each, using energy transfer equation [19]. After that, each encrypted pixel is used to hide a single bit, resulting in higher embedding capacity than [18] which embeds a maximum of one bit per block of pixels. However, data expansion exists in [19] due to the Paillier encryption where the size of the encrypted data is square of the size of the original data.

Qin and Zhang [20] proposed an algorithm for image privacy protection by encrypting all pixels' bits except the 4^{th} least significant bit with a stream cipher. Then data hiding is achieved by first splitting the encrypted image into multiple non-overlapping blocks of equal size and the pixels of each block are then grouped into two groups G_0 and G_1 according to the data-embedding key. Then a single bit can be embedded per block by modifying only the three LSBs of all pixels in one group while keeping the other group unchanged based on the value of the bit to be embedded. This approach proved to have a satisfactory visual quality after decrypting the image since only a slight modification is done per block of pixels. At the receiver side, knowing that a high spatial correlation exists between the pixels of an image, an adaptive function is used to identify which one of the two groups per block is the modified one by assessing the level of smoothness for both groups. The group that shows lower correlation is judged to be the modified one and thus a bit of 0 or 1 is extracted, resulting in high accuracy of message extraction.

Other researchers made use of prediction errors to achieve reversibility of data hiding [21,22]. In [21], each block of image is divided into two groups of pixels; sample pixels SP and non-sample pixels NSP where the former is used to predict the latter. The NSPs are then replaced by their prediction errors and encrypted along with the SPs via strong encryption keys. Then, according to some predefined threshold range, the prediction errors that fall within this range are used to embed external bits through a difference expansion technique, otherwise, histogram shifting takes place. A location map, which is a binary array indicating the locations of prediction errors used to conceal the message bits, is transmitted along with the image in order for the receiver to be able to identify the embedding locations and extract the bits correctly. Whereas in [22], embedding is achieved by directly replacing some bits of the candidate prediction error by the external bits and synchronization is done by reserving one more bit to serve as an embedding flag that holds a value of zero if the Prediction error is used for hiding data and a value of one otherwise. In addition, A block permutation process was further applied after encryption by changing the original positions of blocks inside an image in order to improve the security. This approach results in higher payload capacity and security than [21], however, data extraction can only take place in the encrypted domain.

Some schemes have proposed an RRBE solution which reserves room in the original image before encryption for later data embedding. The idea behind this approach is to fully exploit the strong spatial correlation between neighbouring regions and heavily compress them to reserve room for data hiding. A sparse coding scheme is adopted by [23], where the host image is firstly divided into patches with different levels of smoothness within them. Then the smooth patches that have low residual differences are chosen and expressed by sparse coefficients, while the corresponding residual differences are compressed and hided into the other non-smooth patches, thus vacating a room for hiding additional information. In [24], binary block embedding (BBE) is utilized to hide data in binary images, by embedding the information of the LSB planes of an image into MSB planes, thus vacating the LSBs for subsequent data hiding. First, the binary image is divided into non-overlapping blocks which are then categorized into five categories depending on their proportion of 0's and 1's, and the first two or three bits of each block are then replaced by two or three-bit labels indicating their category.

For homogeneous blocks (all zeros or all ones), only the category label is maintained and for non-homogeneous blocks, additional structural information namely, the number of minority bits and their positions are also self-embedded in the MSBs of the block for image recovery purpose, while the rest LSBs are reserved for later replacement by message bits.

2.4 Machine Learning

In order to identify whether a received motion vector is the original motion vector for a macroblock or a rotated version of it and consequently be able to extract the message bits and reconstruct the video at the decoder. Three supervised machine learning classification techniques are utilized, namely, Random Forests (RF), Support Vector Machine (SVM) and polynomial classifier. However, only the two main and mostly-used approaches; RF and SVM, are explained in details in the following subsections.

2.4.1 Random forests. Random forests is one of the ensemble supervised learning approaches that employ a group of single trained classifiers and combine their individual predictions to provide a more accurate trustworthy prediction compared to the individual ones. As the name 'Forests' reveals, it uses decision trees as its base classifier while 'Random' refers to the randomization property it has; the random sampling of the input data into multiple train and test sets while growing the trees and the random selection of feature variables for each sub-tree as well [25]. The process of growing the individual trees in a random forest for an N-record dataset with M feature variables can be summarized by:

- Each tree is trained with a different training set, which is a random subset of the original dataset sampled randomly with replacement using bootstrap sampling.
- Each tree is trained with a different subset of m predictors, where $m << M$ and is kept constant for all trees.
- For each tree, the records that are not used in its training set are kept out-of-bag (OOB) and used for testing to obtain unbiased classification accuracy and prevent overfitting, thus no need for cross validation.

- When a new feature vector is applied to the generated RF model, each tree in the forest votes for the class, then the final decision will be the class with the largest number of votes.
- RF can evaluate the importance of the feature variables and identify the most influential on prediction, which can be useful in the case of data with very high dimensionality [26].

2.4.2 Support vector machine. Support vector machine is another supervised learning approach used to classify the input vectors into two classes or multiple classes. The main idea behind SVM is to transform the input feature vectors into higher-dimensional space where the two classes or multiple classes can be linearly separable by a high-dimensional surface called a hyper-plane [27]. Different mapping functions are used for transforming the input features into the high-dimensional space, known as kernel functions. Some commonly used kernels are linear, RBF, sigmoid and polynomial.

Figure 2.1 illustrates an example of a hyper-plane in 2D and 3D representations, where the white dots and black dots are the two classes to be separated by the hyper-plane. The middle line is called the line of margin and the two parallel lines of either side of it are the hyper-plane edges. The perpendicular distances between the marginal line and hyper-plane edges are known as the margins. The main goal of SVM is to maximize these margins to obtain better separation between the two classes and thus more accurate classification. The equation of such a hyper-plane can be given by [28]:

$$aX + bY = C \tag{2.4}$$

where C is known as the cost function or complexity parameter and is defined as the sum of all distances of the points that are on the wrong side of the hyper-plane [28].

This chapter has reviewed a wide range of existing solutions in the topic of data hiding in scrambled videos and images. In this work, we extended the work mentioned in the literature by adopting a novel scheme for simultaneous data embedding and video scrambling at the encoder by motion vector rotation and proposing a machine learning approach to recover the original video and extract the embedded message bits at the

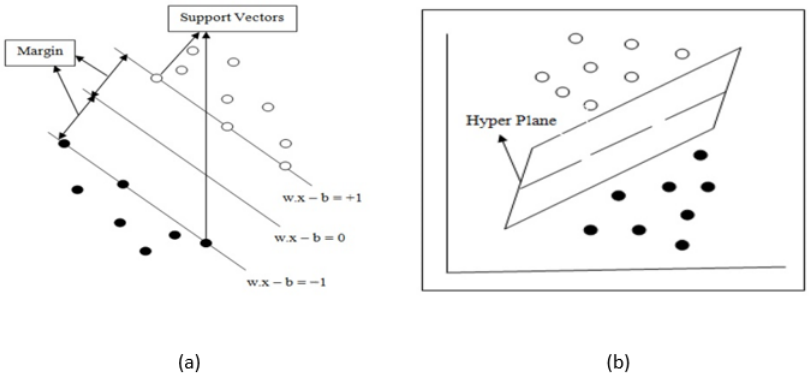

Figure 2.1: A hyper-plane in (a) 2D and (b) 3D [28].

decoder. The proposed methodology and the conducted experiments are illustrated in the following chapters.

Chapter 3. Methodology

In this chapter, the overall data embedding and extraction system is described in details. We use a video-dependent approach, where the first 10% of the video is used for training the model which will be used later to classify the motion vectors of the rest 90% of the video that is used for data embedding to either rotated or unrotated. Since model generation process needs the actual class of each training sample, no data embedding is applied to the first 10% of the video. The decoder upon receiving the motion vectors in the bit stream knows that the MVs of the first 10% of the frames are the true MVs, therefore it uses them as a ground truth to train the model. After that, the decoder applies the trained model to the rest of the video which contains the embedded rotated MVs to classify them. This approach assumes that no major scene change appears in the test video sequence which can be due to using small-sized videos with limited number of frames. However, for larger video sizes, i.e. higher number of frames, where scene changes are expected, the training can be repeated periodically every N frames, e.g. 100 frames, to cater for scene changes. Figure 3.1 illustrates the overall system of training, classification and data extraction. The detailed model training approach is illustrated in the following section.

3.1 Training Phase

For a received uni-directional macroblock (i.e. a MB with only one MV; either forward or backward), its corresponding MV is rotated using 4 angles: 0° , 90° , 180° and 270° , then each of the 4 rotated MVs is applied to motion compensation process with the reference frame providing 4 motion compensated MBs which are then added to the prediction error to reconstruct 4 different MBs. In the case of a bi-directional macroblock (i.e. a MB with 2 MVs; forward MV and backward MV), each MV is rotated using the four angles, resulting in 16 candidate reconstructed MBs due to the 16 possible combinations of the two motion vectors. Features are then extracted from each reconstructed MB and its associated MV(s), resulting in 4 or 16 feature vectors per a MB. The feature vector corresponding to rotation angle of 0° (or 0°,0° for bi-directional

MBs) belongs to the unrotated class so is marked as 1, while the rest 3 or 15 feature vectors belong to the rotated class so are marked as 0. This process is repeated for all the received macroblocks in the first 10% of the video except the intra MBs since they have no MVs, and the MBs with (0,0) MVs since they cannot be rotated. The obtained feature matrix is then used to build the model. The motion vectors' rotation, the feature variables used, and the machine learning approaches utilised are described in the following subsections respectively.

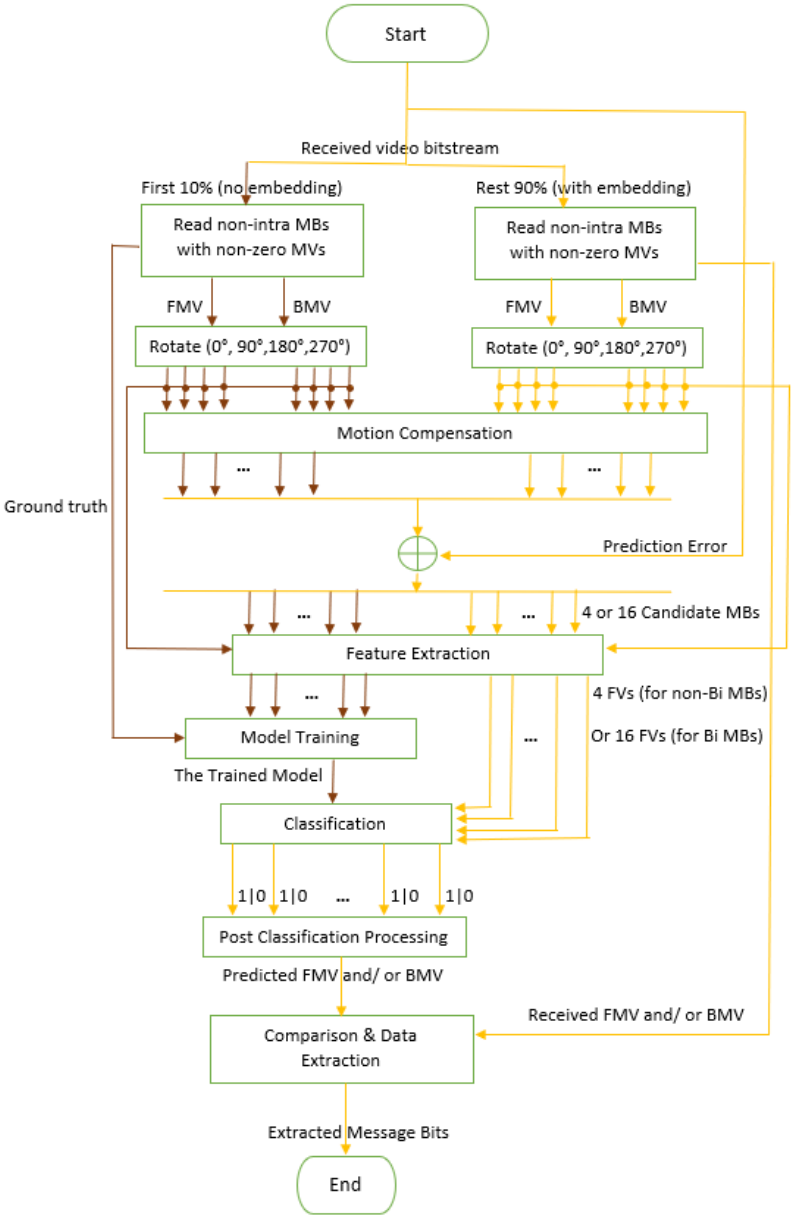

Figure 3.1: Flow chart of the complete training and data extraction system.

3.1.1 Motion vector rotation. For each eligible MB (i.e. non-intra and has non-zero MV), its motion vector(s) is/ are rotated using four different angles as shown in the following equations:

$$\begin{aligned} MV_x^0 &= MV_x \\ MV_y^0 &= MV_y \end{aligned} \tag{3.1}$$

$$\begin{aligned} MV_x^{90} &= -1 * MV_y \\ MV_y^{90} &= MV_x \end{aligned} \tag{3.2}$$

$$\begin{aligned} MV_x^{180} &= -1 * MV_x \\ MV_y^{180} &= -1 * MV_y \end{aligned} \tag{3.3}$$

$$\begin{aligned} MV_x^{270} &= MV_y \\ MV_y^{270} &= -1 * MV_x \end{aligned} \tag{3.4}$$

where MV_x and MV_y are the horizontal and vertical components of the original motion vector respectively, while MV_x^i and MV_y^i are the horizontal and vertical components of the motion vector rotated by an angle of i$^\circ$.

3.1.2 Feature variables. The feature set used to represent each candidate reconstructed MB and its associated MV(s) consists of 11 feature variables, 7 of them are extracted from the macroblock domain and the other 4 are extracted from the motion vector domain. Table 3.1 and Table 3.2 describe them respectively.

3.1.3 Model generation. After extracting the features and obtaining the complete training dataset, we applied three machine learning approaches: the support vector machine (SVM), the random forests (RF) and the polynomial classifier. For SVM model, a RBF kernel was used which is given by the following equation [29]:

$$K_{RBF}(x_i, x_j) = exp(-\gamma \left\| x_i - x_j \right\|^2) \tag{3.5}$$

where x_i and x_j are the i^{th} and j^{th} feature vectors.

For random forests, we set the number of grown trees to 128. Each tree is then trained with a subset of the full feature set mentioned in Table 3.1 and Table 3.2. In polynomial classifier, the feature vectors are expanded into higher dimensionality using polynomial expansion, and the classification decision is then determined from the following equation that combines the expanded feature variables with the polynomial weights [30] :

$$\begin{aligned} f(w,x) = \alpha_o + \sum_{k=1}^{r} \sum_{j=1}^{l} w_{kj} x_j^k + \sum_{j=1}^{r} w_{rl+j} (x_1 + x_2 + ... + x_l)^j \\ + \sum_{j=2}^{r} (w_j^T x)(x_1 + x_2 + ... + x_l)^{j-1}, l, r \geq 2 \end{aligned} \tag{3.6}$$

where r is the polynomial order, w are the polynomial weights to be estimated and x is the feature vector with l feature variables. The polynomial weights are estimated using least-squares error minimization. The order of polynomial expansion used is five.

After extracting the feature vectors associated with each set of 4 or 16 candidate MBs, we ranked them as follows:

- For all feature variables in Table 3.1, the 4 or 16 candidate MBs are ranked in an ascending order, where the MB having the lowest value is ranked as 1 and the MB having the highest value is ranked as 4 or 16. When two or more MBs in the candidates set have the same value for that feature variable, they are given the same ranking value.
- For all feature variables in Table 3.2, since they are phase differences; first, the whole angular range is divided into 8 parts of 45°-wide each, and each part is assigned a unique ranking number between 0 and 7, then a direct mapping is performed between the actual phase difference values and the ranking values.

3.2 Data Embedding and Extraction

As mentioned previously, only the rest 90% of the video is used to embed the message bits. Data embedding takes place at the encoder by rotating the motion vectors in the final bit stream, not during the encoding process in order not to affect the motion compensation process. Again, intra MBs and MBs with (0,0) MVs are not used for data hiding. A random binary number generator is used to generate the to-be-embedded sequence, and two bits are embedded per MV by rotating it according to the value of the two bits as follows:

- To embed '00' the MV is rotated by 0° according to Equation 3.1.
- To embed '01' the MV is rotated by 90° according to Equation 3.2.
- To embed '10' the MV is rotated by 180° according to Equation 3.3.
- To embed '11' the MV is rotated by 270° according to Equation 3.4.

By this embedding approach, 2 bits are embedded in a uni-directional MB with a single MV and 4 bits are embedded in a bi-directional MB with two motion vectors.

The decoder upon receiving the second part of the video knows that the received motion vectors could be rotated, however, the original values are unknown. Therefore, in order to obtain the actual values of the received MVs, the same steps explained in the training phase in the previous section are applied to the received possibly rotated MVs. First, the motion vector of a received uni-directional MB is rotated using the 4 rotation angles described in Subsection 3.1.1 producing 4 candidate MVs which are

Table 3.1: List of features extracted from macroblock domain

Feature ID	Feature Description
1	The entropy of pixel values in a reconstructed candidate MB.
2	The average of pixel values in a reconstructed candidate MB.
3	The variance of pixel values in a reconstructed candidate MB.
4	The Sum of absolute edge values of a reconstructed candidate MB using 'Sobel' edge detection.
5	The pixel violation counts in a reconstructed candidate MB (number of pixels with values less than 0 or greater than 255).
6	The sum of absolute values of the prediction error obtained by subtracting a candidate reconstructed forward MB (FMB) from the co-located MB in the reference frame.
7	Same as feature ID 6 above but applied to the backward MB (BMB).

then applied to motion compensation process with the reference frame resulting in 4 motion compensated MBs which in turn are added to the PE producing 4 candidate reconstructed MBs. If the received MB is bi-directional, each of its 2 MVs is rotated using the 4 rotation angles and thus 16 candidate MBs are reconstructed in this case. Then, feature extraction is performed on the 4 or 16 candidate MBs and their corresponding MVs. However, in this case the decoder does not know the class of each feature vector. Therefore, the classification models explained in the previous section are used to decide which of the candidate reconstructed MBs corresponds to the true unrotated MV(s). After that, the MV classified as the true unrotated MV by the classifier is compared against the received MV and the hidden message bits are then extracted using the following formulas:

$$\text{B}its_{embedded} = \begin{cases} 00 & \text{if } |arctan(MV_y/MV_x) - arctan(\bar{M}V_y/\bar{M}V_x)| = 0^\circ \\ 01 & \text{if } |arctan(MV_y/MV_x) - arctan(\bar{M}V_y/\bar{M}V_x)| = 90^\circ \\ 10 & \text{if } |arctan(MV_y/MV_x) - arctan(\bar{M}V_y/\bar{M}V_x)| = 180^\circ \\ 11 & \text{if } |arctan(MV_y/MV_x) - arctan(\bar{M}V_y/\bar{M}V_x)| = 270^\circ \end{cases} \tag{3.7}$$

where MV_x and MV_y indicate the horizontal and vertical components of the received MV respectively, while $\bar{M}V_x$ and $\bar{M}V_y$ are the horizontal and vertical components of the classified MV.

Table 3.2: List of features extracted from motion vector domain

Feature ID	Feature Description
8	The absolute difference between the phase of the candidate MV and the phase of the top MB of the same frame.
9	The absolute difference between the phase of the candidate MV and the phase of the left MB of the same frame.
10	The absolute difference between the phase of the candidate forward MV (FMV) and the phase of the co-located MV of the reference frame.
11	Same as feature ID 10 above but applied to the backward MV (BMV).

3.3 Post Classification Processing

In some cases, the classifier may result in different types of errors. Three types of errors are defined in this work. For a candidate set of 4 reconstructed macroblocks (or 16 reconstructed MBs for a bi-directional MB), if the classification model classifies two or more of them as true unrotated, a Type I is encountered. If none of the candidate set members is classified as the true unrotated, a Type II error is encountered. For both Type I and Type II errors, a post classification process is performed on the corresponding candidate set by computing the sum of absolute differences (SAD) between each candidate MB in the wrongly classified set and the top and left MBs of the same frame. The MB that produces the minimum SAD value is considered as the true MB corresponding to the true unrotated MV(s) . For Type III error in which the classifier identifies only one of the 4 or 16 candidate MBs as the true MB, however, it is not actually the true MB, no further processing is performed. Figure 3.2 illustrates the three different types of errors and Equation 3.8 describes the SAD between a candidate MB and the top and left MBs of the same image.

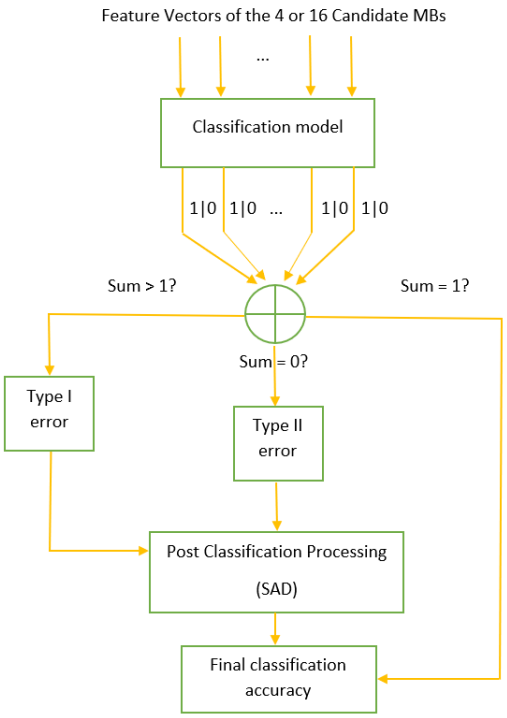

Figure 3.2: The three different error types.

$$SAD = \sum_{j=1}^{16} |p_c(1,j) - p_t(16,j)| + \sum_{i=1}^{16} |p_c(i,1) - p_l(i,16)| \tag{3.8}$$

where $p_c(i,j)$, $p_t(i,j)$ and $p_l(i,j)$ denote the pixel values in the i^{th} row and j^{th} column of the candidate MB, the top MB and the left MB respectively.

3.4 Combined Encoder-Decoder Solution

In the previous solution where the classification model is built only at the decoder, some of the MVs can be wrongly classified which leads to inaccurate message extraction and also affects the quality of the decoded frames due to the incorrect counter-rotation of the MVs as will be illustrated in the experimental results. Therefore, a second solution is proposed where the model generation and MVs' classification take place also at the encoder side. The algorithm of this solution is illustrated in Figure 3.3. First, the encoder uses the same frames that will be used by the decoder to build the classification model, i.e. the first 10 frames in the sequence, which makes the encoder and the decoder have the same classification model without adding the extra overhead of transferring the model parameters in the bit stream. Once the model is generated, for each non-intra MB with a non-zero MV in the rest 90% of the frames sequence, the following procedure is followed: the encoder rotates its MV(s) according to the bits to be embedded. Then, the same steps of rotating the MVs, reconstructing the candidate MBs, and extracting the feature variables that are followed by the decoder are executed by the encoder. After that, the 4 or 16 computed feature vectors for a MB are applied to the encoder's classification model and the result of classification is compared against the ground truth (the true un-rotated MV(s)). If the MB is classified correctly, data embedding takes place by rotating its MV(s) according to the message bits. Otherwise, no bits are embedded and the MB is forced to be either intra-coded or inter-coded with zero MV in order to enable the decoder to distinguish the MBs with hidden bits from those with no hidden bits.

Compared to the previous decoder-only solution, this solution results in 100%-accurate classification at the decoder side, thus error-free message extraction and correct reconstruction without affecting the quality of the decoded video frames. On the

other hand, restricting the embedding process at the encoder side only to the correctly-classified MBs results in reducing the embedding rate. Also, a slight increase can be witnessed in the video bitrate due to forcing the misclassified MBs at the encoder to intra-coding or zero-MV-inter-coding.

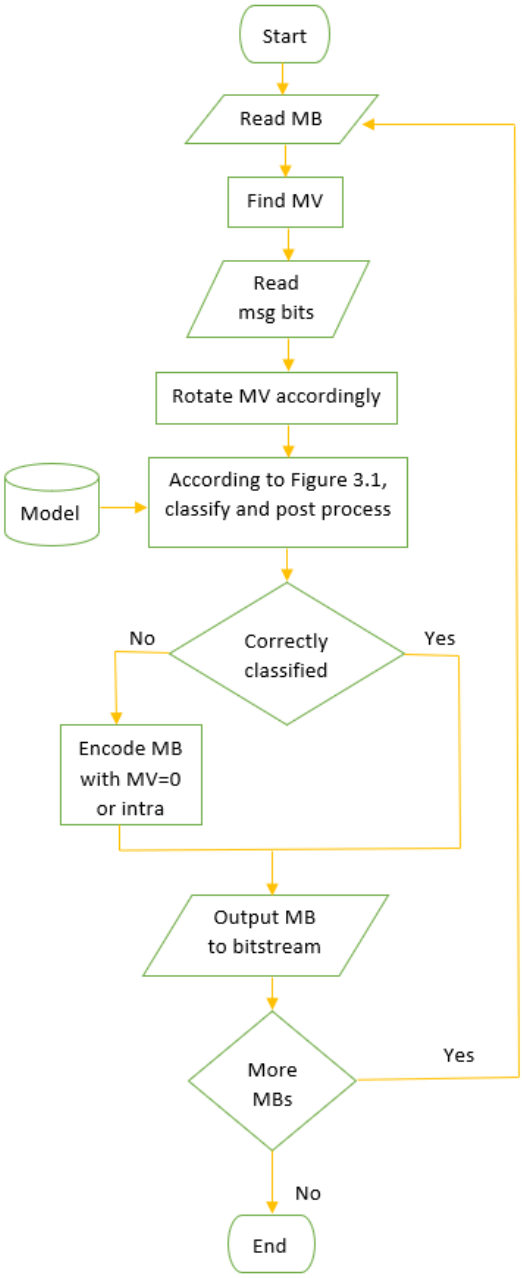

Figure 3.3: Data embedding in selective MBs at the encoder's side.

Chapter 4. Experimental Setup and Results

In this chapter, we present the results obtained by implementing the proposed system described in Chapter 3. We conducted the experiments using 100 frames of each of the 9 well-known video sequences shown in Table 4.1. As shown in the table, the highest spatial resolution we had for a video sequence was 1280×768 which was completely enough for demonstrating and evaluating our proposed work, as using videos with greater spatial resolutions will only result in increasing the embedding rate further due to the higher number of MBs they contain. First, all sequences were compressed using MPEG2 video coding standard. Each sequence was encoded three times using a different quantization scale in each. The values of QS used are: 5, 15 and 25. The solution was evaluated on two types of encoding sequences: IPPP... sequence (with only P frames) and IBPBP... sequence (with P and B frames). For IBPBP... sequence, the proposed system was evaluated for the three different MB types separately, namely, the forward-predicted MBs of P frames, the mono-directional MBs of B frames and the bi-directional MBs of B frames. A video-dependent training was used where the first 10 frames were used for training the model. Three classifiers were trained per sequence per QS value, namely, Random Forests, SVM with RBF kernel and fifth order polynomial. The results for MPEG2 are presented in Section 4.1. Then, all the experiments were repeated using HEVC video coding standard. For HEVC, the proposed solution was tested only for IPPP... encoding sequence and the quantization scale values used to compress the different test video sequences were 22, 27, 32 and 37. The results of HEVC are reported in Section 4.2. After that, a comparison between our proposed solution (machine learning followed by SAD as post processing) and only using SAD for classification is conducted in Section 4.3, along with comparing it with existing solutions. At last, the sequence-independent training approach was conducted and examined for completeness, and the classification results are shown in Section 4.4.

4.1 MPEG2 Results

With MPEG2 encoder, we tested our solution on 2 different encoding sequences; IPPP... sequence (with N=100, GOP=100, M=1, i.e. only one I frame followed by a

Table 4.1: Video test sequences and their resolutions.

	Sequence Name	Width × Height	frames/sec
1	RaceHorses	416 × 240	30
2	BlowingBubbles	416 × 240	50
3	BasketballPass	416 × 240	50
4	RaceHorses	832 × 480	30
5	BQMall	832 × 480	60
6	BasketballDrill	832 × 480	50
7	ParkScene	1280 × 768	24
8	Kimono1	1280 × 768	24
9	Cactus	1280 × 768	50

sequence of 99 predicted frames, no bi-directional frames) and IBPBP... sequence (with N=100, GOP=100, M=2, i.e. only one I frame followed by a sequence of bi-directional and predicted frames, with one B frame between each two P frames). For each encoding sequence: each sequence was compressed using the three aforementioned quantization scales; 5, 15 and 25 and for each of them, three classifiers were trained which are: RF, SVM with RBF kernel and fifth order polynomial classifiers. Each classifier was trained on the first 10% of each video, i.e. first 10 frames, and then used to classify the MVs of the rest 90% of the same video, i.e. last 90 frames. In the cases when the classifier resulted in Type I or Type II error where more than one MB or no MB respectively in a candidates set were/ was classified as the correct MB, post processing was applied to the corresponding MBs to identify the correct MB by the one having the lowest SAD with the top and left MBs of the same frame.

4.1.1 Classification and post processing. Firstly, we present the classification results obtained using IBPBP... sequence. We tested our solution on the three different MB types that exist in an IBPBP... sequence: forward-predicted MBs in P frames (P MBs), mono-directional (forward or backward) predicted MBs in B frames (mono-B MBs) and bi-directional (both forward and backward) predicted MBs in B frames (bi-B MBs). Table 4.2 to Table 4.10 illustrate the classification results obtained by RF, RBF-SVM and 5^{th} order plynomial on a MB-type basis for each quantization scale separately. The results are all presented as an average over the 9 test sequences in order to make them readable and easy to comprehend. Then, the results for each classifier

averaged over the 3 quantization scales are summarized in Tables 4.11, 4.12 and 4.13 for RF, SVM and polynomial classifiers respectively. The term 'Model accuracy' refers to the accuracy obtained by the classifier, while the term 'Final accuracy' refers to the accuracy obtained after applying the post processing using SAD to the output of the classifier. Beside Type I and Type II errors explained above, the classification error in which only one MB in a candidates set is classified as the true MB while it is not is also presented. From the results we can observe the following:

- Observing the results for the different QS values obtained by the same classifier, we can conclude that as we increase the quantization scale value, the classification accuracy decreases. This can be explained as follows: when using small quantization scale values for compression, most of the DCT coefficients of the prediction errors maintain non-zero values after quantization, providing more information to the classifier to base its decision on. On the other hand, when using high values of quantization scales, the resulting quantized DCT coefficients are mostly zeros, thus less information is available for the classifier which affects its accuracy.
- Observing the average results for the different classifiers, RF outperformed 5^{th} order polynomial which in turn performed better than RBF-SVM with final accuracy of 96.79% , 96.58% and 91.21% respectively, averaged over all QS values and all MB types.
- In all cases (for all classifiers and all QS values), the classification accuracies for P MBs are lower than their mono-B MBs counterparts. This can be due to the fact that mono-B MBs use reference frames that are one frame apart, while P MBs use reference frames that are 2 frames apart.
- For RF and RBF-SVM classifiers, the percentage of Type II error is greater than the percentage of Type I error, which means that the classifier tends to classify a training instance as '0' more than as '1', which can be due to the imbalance data, even with applying the down-sampling technique to the majority class (class 0) which reduces the percentage of Type II error, but cannot eliminate it completely. For Polynomial classifier, the expansion of feature variables had a positive impact on Type II error as the percentages of Type II error have decreased.

Table 4.2: RF results for QS 5 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	93.85	94.67	95.96	94.83
% of Type I error	1.74	1.01	1.01	1.25
% of Type II error	1.87	2.08	1.01	1.65
% of Classification error	2.54	2.23	2.02	2.26
Fixed % of Type I error	68.06	81.67	95.09	81.61
Fixed % of Type II error	54.35	48.05	93.22	65.21
Final accuracy	**96.05**	**96.53**	**97.85**	**96.81**

Secondly, we present the classification results obtained using IPPP... sequence. Table 4.14 and Table 4.15 show the results obtained by RF classifier and SVM classifier with RBF kernel respectively for the different QS values. Then, the results of both classifiers are averaged over the 3 quantization scales and summarized in Table 4.16. We can observe that the conclusions that we derived from IBPBP... sequence results are supported by IPPP... sequence results. That is; increasing the QS value used for compression results in decreasing the classification accuracy, RF classifier produced better classification results than SVM classifier, and the percentages of Type II error are higher than the percentages of Type I error.

At last, when comparing the classification accuracies obtained by the best classifier (RF) for IBPBP... sequence against IPPP... sequence, we can see that the overall average classification accuracy using P frames only is 96.34% which is slightly lower than 96.79% in the case of using both P and B frames. This can be attributed to the fact that with IBP... sequences, having bi-directional MBs results in improving the classification accuracy since the classifier in this case has more information to base its decision on, as the most influential features such as the MVs' phase differences and the sum of absolute values of the prediction errors are now extracted in both directions; referencing to the previous and future frames.

4.1.2 Features selection. In this subsection, we test the effect of features selection by applying it to Random Forests. The idea is that: While first training the different RF classifiers using all the feature variables of Tables 3.1 and 3.2, the importance of each feature variable in predicting the correct classification of a test feature vector was computed and recorded. After that, only the features having an importance greater

Table 4.3: RF results for QS 15 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	93.99	95.76	95.87	95.21
% of Type I error	1.63	1.01	1.01	1.22
% of Type II error	1.80	1.08	1.01	1.30
% of Classification error	2.57	2.14	2.10	2.27
Fixed % of Type I error	67.47	90.01	88.67	82.05
Fixed % of Type II error	52.89	35.54	54.86	47.76
Final accuracy	**96.05**	**97.06**	**97.32**	**96.81**

Table 4.4: RF results for QS 25 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	94.24	95.52	95.83	95.19
% of Type I error	1.36	1.00	1.03	1.13
% of Type II error	1.84	1.38	1.02	1.41
% of Classification error	2.56	2.09	2.12	2.26
Fixed % of Type I error	66.54	73.61	72.69	70.95
Fixed % of Type II error	48.75	44.88	69.31	54.31
Final accuracy	**96.08**	**96.89**	**97.27**	**96.75**

Table 4.5: SVM results for QS 5 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	89.38	95.04	75.86	86.76
% of Type I error	2.14	1.22	2.26	1.87
% of Type II error	5.87	1.50	19.03	8.80
% of Classification error	2.60	2.23	2.85	2.56
Fixed % of Type I error	65.13	53.76	67.88	62.26
Fixed % of Type II error	60.71	62.47	41.60	54.93
Final accuracy	**95.39**	**96.66**	**85.77**	**92.61**

Table 4.6: SVM results for QS 15 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	90.70	95.38	73.55	86.54
% of Type I error	2.12	1.10	2.64	1.95
% of Type II error	4.52	1.41	20.52	8.82
% of Classification error	2.66	2.11	3.29	2.69
Fixed % of Type I error	68.14	40.42	60.54	56.37
Fixed % of Type II error	61.82	57.48	36.59	51.96
Final accuracy	**95.62**	**96.65**	**83.04**	**91.77**

Table 4.7: SVM results for QS 25 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	88.77	95.47	69.98	84.74
% of Type I error	1.71	1.05	2.71	1.82
% of Type II error	6.88	1.39	23.66	10.64
% of Classification error	2.63	2.08	3.65	2.79
Fixed % of Type I error	62.75	40.73	57.51	53.66
Fixed % of Type II error	55.63	57.53	34.45	49.20
Final accuracy	**95.13**	**96.71**	**80.06**	**90.63**

Table 4.8: Polynomial results for QS 5 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	93.05	95.66	93.18	93.96
% of Type I error	2.43	1.18	3.72	2.44
% of Type II error	1.00	1.00	1.00	1.00
% of Classification error	3.52	2.17	2.09	2.59
Fixed % of Type I error	62.36	55.35	81.24	66.32
Fixed % of Type II error	100	100	88.89	96.29
Final accuracy	**95.63**	**97.30**	**96.55**	**96.49**

Table 4.9: Polynomial results for QS 15 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	93.58	95.73	93.21	94.17
% of Type I error	1.85	1.15	3.65	2.22
% of Type II error	1.00	1.00	1.01	1.00
% of Classification error	3.57	2.12	2.13	2.61
Fixed % of Type I error	61.69	46.33	64.81	57.61
Fixed % of Type II error	100	100	81.48	93.83
Final accuracy	**95.74**	**97.27**	**96.01**	**96.34**

Table 4.10: Polynomial results for QS 25 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	92.83	95.15	92.59	93.52
% of Type I error	2.56	1.50	4.37	2.81
% of Type II error	1.00	1.01	1.00	1.00
% of Classification error	3.60	2.34	2.03	2.66
Fixed % of Type I error	67.32	49.40	88.44	68.39
Fixed % of Type II error	100	93.06	100	97.69
Final accuracy	**95.54**	**96.86**	**96.56**	**96.32**

Table 4.11: RF results for MPEG2 IBP...sequence averaged over all QS values.

	P	mono-B	bi-B
Model accuracy	94.03	95.32	95.89
% of Type I error	1.58	1.01	1.02
% of Type II error	1.84	1.51	1.01
% of Classification error	2.56	2.16	2.08
Fixed % of Type I error	67.36	81.76	85.48
Fixed % of Type II error	52.00	42.82	72.46
Final accuracy	**96.06**	**96.83**	**97.48**

Table 4.12: SVM results for MPEG2 IBP...sequence averaged over all QS values.

	P	mono-B	bi-B
Model accuracy	89.62	95.29	73.13
% of Type I error	1.99	1.13	2.54
% of Type II error	5.76	1.43	21.07
% of Classification error	2.63	2.14	3.26
Fixed % of Type I error	65.34	44.97	61.98
Fixed % of Type II error	59.39	59.16	37.55
Final accuracy	**94.34**	**96.67**	**82.62**

Table 4.13: Polynomial results for MPEG2 IBP...sequence averaged over all QS values.

	P	mono-B	bi-B
Model accuracy	93.15	95.51	92.99
% of Type I error	2.28	1.28	3.91
% of Type II error	1.00	1.00	1.01
% of Classification error	3.56	2.21	2.09
Fixed % of Type I error	63.79	50.36	78.16
Fixed % of Type II error	100	97.69	90.12
Final accuracy	**95.64**	**97.15**	**96.95**

Table 4.14: RF results for MPEG2 using an IPP... sequence.

	QS 5	QS 15	QS 25
Model accuracy	94.30	92.00	91.40
% of Type I error	1.21	2.34	2.39
% of Type II error	1.82	2.46	2.57
% of Classification error	2.67	3.20	3.64
Fixed % of Type I error	79.21	89.63	88.38
Fixed % of Type II error	81.45	93.14	92.68
Final accuracy	**96.74**	**96.39**	**95.89**

Table 4.15: SVM results for MPEG2 using an IPP... sequence.

	QS 5	QS 15	QS 25
Model accuracy	93.20	90.70	89.40
% of Type I error	1.42	2.71	2.87
% of Type II error	2.03	2.98	3.16
% of Classification error	3.35	3.61	4.57
Fixed % of Type I error	78.96	87.26	88.02
Fixed % of Type II error	84.21	92.53	93.23
Final accuracy	**96.03**	**95.82**	**94.87**

Table 4.16: Comparison of RF and SVM for MPEG2 using an IPP... sequence.

	RF	SVM
Model accuracy	92.57	91.10
% of Type I error	1.98	2.33
% of Type II error	2.28	2.72
% of Classification error	3.17	3.84
Fixed % of Type I error	85.74	84.75
Fixed % of Type II error	89.09	89.99
Final accuracy	**96.34**	**95.57**

than a specific threshold were selected to train the RF classifiers. This was repeated for each sequence for each QS value and the effect of feature selection was tested for each MB type separately using IBPBP... sequence. Tables 4.17, 4.18 and 4.19 illustrate the results of RF with feature selection for QS 5, 15 and 25 respectively, averaged over all the 9 test sequences. A comparison between RF with feature selection and RF without feature selection is shown in Table 4.20 with the results being averaged over all QS values. From the table, we can clearly observe that feature selection did not improve the classification results of RF, as the average classification accuracies with and without features selection are almost similar.

Although features selection did not add a value to RF classification results, exploring the most influential features that contributed the most to the obtained classification accuracies is a must. Therefore, we plot the histograms of the selected feature variables per MB type in Figure 4.1. In general, we can observe that the most frequently used features for all MB types are the phase differences between a candidate MB and its top and left MBs of the same frame and the reference MB in the reference frame, and the sum of absolute values of the prediction errors obtained by subtracting a candidate

Table 4.17: RF with features selection for QS 5 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	94.04	95.77	95.87	95.23
% of Type I error	1.57	1.01	1.02	1.20
% of Type II error	1.80	1.08	1.02	1.30
% of Classification error	2.59	2.14	2.09	2.27
Fixed % of Type I error	60.21	89.68	76.43	75.44
Fixed % of Type II error	47.51	35.20	78.02	53.58
Final accuracy	**95.93**	**97.05**	**97.43**	**96.80**

Table 4.18: RF with features selection for QS 15 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	94.28	95.54	95.80	95.21
% of Type I error	1.34	1.01	1.03	1.13
% of Type II error	1.82	1.35	1.02	1.39
% of Classification error	2.56	2.10	2.15	2.27
Fixed % of Type I error	73.33	83.07	75.00	77.13
Fixed % of Type II error	49.51	52.83	58.44	53.59
Final accuracy	**96.17**	**97.08**	**97.16**	**96.80**

MB from the co-located MB in the reference frame, which correspond to the IDs 1-4, 10 and 11. Except that features with IDs 4 and 11 have a value of zero for P MBs, as they correspond to backward prediction which is not applicable for P MBs. Additionally, feature with ID 9, which is the pixel variance of the reconstructed candidate MB, appears to have high importance in the case of bi-directional B MBs.

Table 4.19: RF with features selection for QS 25 for MPEG2 IBP...sequence.

	P	mono-B	bi-B	Average
Model accuracy	93.88	94.59	95.96	94.81
% of Type I error	1.70	1.02	1.02	1.25
% of Type II error	1.88	2.12	1.01	1.67
% of Classification error	2.54	2.27	2.02	2.28
Fixed % of Type I error	72.44	59.89	95.12	75.82
Fixed % of Type II error	53.31	52.60	92.52	66.14
Final accuracy	**96.09**	**96.31**	**97.85**	**96.75**

Table 4.20: RF with and without features selection averaged over all QS values for MPEG2 IBP...sequence.

	with features selection			without features selection		
MB Type	P	mono-B	bi-B	P	mono-B	bi-B
Model accuracy	94.07	95.30	95.88	94.03	95.32	95.89
% of Type I error	1.54	1.01	1.02	1.58	1.01	1.02
% of Type II error	1.83	1.52	1.02	1.84	1.51	1.01
% of Classification error	2.56	2.17	2.09	2.56	2.16	2.08
Fixed % of Type I error	68.66	77.55	82.18	67.36	81.76	85.48
Fixed % of Type II error	50.11	46.88	76.33	52.00	42.82	72.46
Final accuracy	**96.06**	**96.82**	**97.48**	**96.06**	**96.83**	**97.48**

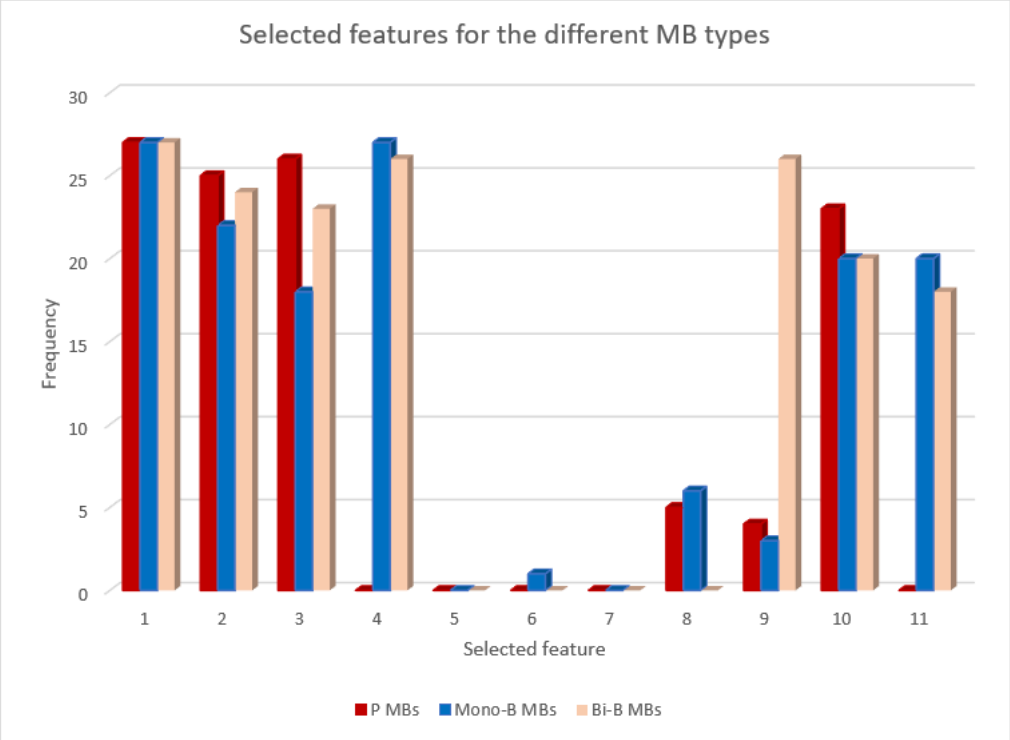

Figure 4.1: Selected features for the three different MB types.

4.1.3 Embedding capacity. The embedding capacity for each sequence measured in bits/MB was computed by the following formula:

$$\text{Embedding capacity} = \frac{number\,of\,embedded\,bits}{total\,testing\,MBs} \quad Bits/MB \tag{4.1}$$

The embedding capacities for all test sequences for the different quantization scales are reported in Tables 4.21 and 4.22 for IBP... sequence and IPP... sequence respectively. We can observe that as the quantization scale value for IPP... sequence increases, the embedding capacity decreases. This can be explained as follows: when motion estimation is performed between a MB and its reference frame, if the reference frame is quantized using high quantization scale value, the chance to find a best match location for the MB in its previous frame becomes lower, so the encoder may decide either to consider the co-located MB as a best match which means the MV becomes (0,0) or to encode the MB as an intra MB. Another reason could be that when a macroblock is quantized with a high QS value, there is a good chance that all the quantized DCT coefficients become zeros and thus the encoder decides to skip this macroblock. However, for IBP... sequence there is no constant behaviour of the embedding rate with increasing the QS value, this can be due to that the encoder's decision on the MB type to be mono-directional or bi-directional is irrelevant to the increase in the quantization scale value used for compression.

Tables 4.23 and 4.24 provide a comparison between the embedding rates measured in bits/MB and Kbits/second respectively for each test sequence when encoded using an IPP... sequence and an IBP... sequence averaged over the 3 QS values. From the tables, we can clearly observe the advantage of having B frames in the encoding sequence over only having P frames, as the bi-directional MBs of B frames can embed 4 bits (2 bits per each of their 2 MVs) compared to only 2 bits per the forward-predicted MBs of P frames, resulting in an overall average of 1.68 bits/MB for IBP... sequence compared to 1.07 bits/MB for IPP... sequence.

4.1.4 Effect of missclassification on the decoded video quality. As mentioned previously in Chapter 2, a good data hiding scheme should maintain an acceptable level of quality for the host video. In other words, embedding the bits should not severely affect the visual content of the cover video. In our approach, for the percentage of macroblocks that are wrongly classified (1 - accuracy), the decoder rotates them back wrongly which may affect the quality of the reconstructed video at the decoder. To examine the effect of misclassification on the PSNR of the reconstructed video, we modi-

Table 4.21: Embedding capacity in bits/MB for MPEG2 IBP... sequence.

Sequence ID	QS 5	QS 15	QS 25
1	2.18	2.15	2.06
2	1.36	1.41	1.42
3	1.23	1.43	1.44
4	2.28	2.22	2.15
5	1.22	1.33	1.40
6	1.18	1.36	1.33
7	1.86	1.75	1.68
8	2.36	2.23	2.15
9	1.29	1.39	1.47

Table 4.22: Embedding capacity in bits/MB for MPEG2 IPP... sequence.

Sequence ID	QS 5	QS 15	QS 25
1	1.65	1.54	1.40
2	1.03	0.86	0.71
3	0.97	0.85	0.74
4	1.70	1.64	1.56
5	0.86	0.84	0.80
6	1.04	1.00	0.84
7	1.27	1.50	1.42
8	1.55	0.74	0.69
9	0.85	0.48	0.44

fied MPEG2 decoder to simulate the misclassification effect by randomly rotating a percentage of motion vectors equals to (1 - accuracy) before motion compensation takes place. As a result, a similar percentage of macroblocks was wrongly reconstructed. Then, the PSNRs of the resulting decoded frames were measured and compared to the PSNRs of the normally decoded frames (without wrong rotation) and hence the PSNR differences were observed. We tested this for IBPBP... sequence twice; once with no intermediate I frames (N=100, GOP=100, M=2, i.e. only one I frame, followed by a sequence of bi-directional and predicted frames, with one B frame between each two P frames) and then with inserting an intermediate I frame every 12 frames in the sequence (N=100, GOP=12, M=2, i.e. an I frame every 12 frames, followed by a sequence of bi-directional and predicted frames, with one B frame between each two P frames). The results are shown in Table 4.25 for each test sequences averaged over all the 3 quantization scales. Referring to the table, the overall PSNR drop averaged over all sequences

Table 4.23: A comparison between the embedding capacity of IPPP... and IBPBP... sequences for MPEG2 in bits/MB, averaged over all QS values .

Sequence ID	IPP... sequence	IBP... sequence
1	1.53	2.13
2	0.87	1.40
3	0.85	1.37
4	1.63	2.22
5	0.83	1.32
6	0.96	1.29
7	1.40	1.76
8	0.99	2.25
9	0.59	1.38
Average	**1.07**	**1.68**

Table 4.24: A comparison between the embedding capacity of IPPP... and IBPBP... sequences for MPEG2 in Kbits/sec, averaged over all QS values .

Sequence ID	IPP... sequence	IBP... sequence
1	14.32	19.94
2	13.52	21.79
3	13.31	21.32
4	68.60	93.10
5	70.00	110.60
6	67.20	90.30
7	120.27	151.85
8	85.54	193.46
9	105.85	248.17

and QS values is 6.43 dB without intermediate I frames, whereas inserting periodic I frames mitigated the effect of error propagation on the quality of the decoded video resulting in decreasing the PSNR drop to 4.98 dB. Therefore, we can conclude that having periodic I frames with our embedding approach ensures better quality of the decoded video, and if we increase the frequency of the intermediate I frames further, i.e. an I frame every 8 frames or 4 frames, a better quality will be obtained for the decoded video. On the other hand, increasing the frequency of the periodic I frames will increase the number of intra MBs in the sequence at the expense of inter MBs, which will reduce the embedding capacity.

As a result, to totally eliminate the effect of misclassification on the quality of the decoded video as well as on the correctness of message bits extraction, a combined

Table 4.25: PSNR drop [in dB] in MPEG2 IBP... decoded sequence as a result of misclassification at the decoder, averaged over all QS values.

Sequence ID	With I frames	Without I frames
1	6.03	8.94
2	1.82	3.12
3	3.59	5.10
4	6.09	7.18
5	6.38	7.99
6	4.25	6.06
7	4.76	5.22
8	6.21	7.35
9	5.71	6.87
Average	**4.98**	**6.43**

encoder-decoder solution was proposed as will be illustrated and examined in the following subsection.

4.1.5 Combined encoder-decoder solution. To eliminate the drop in PSNR of the reconstructed video at the decoder, this combined encoder-decoder solution was proposed. As mentioned in Chapter 3, this solution implies that the training and classification processes are also performed at the encoder, so that the MBs which are wrongly classified by the encoder's classifier are not used to hide message bits, instead, their coding modes are set to intra coding or inter coding with no motion compensation (zero MV) to make the decoder able to distinguish them from the MBs having hidden bits. In this way, the classification accuracy at the decoder will be 100%, the message bits will thus be 100%-error-free extracted and all the MVs will be correctly counter-rotated and hence eliminating the drop in PSNR of the reconstructed video at the decoder. However, excluding the misclassified MBs from the embedding process at the encoder reduces the embedding rate, and manipulating the coding modes of those MBs accordingly leads to an increase in the video bitrate. We evaluated this solution in terms of bitrate increase and PSNR drop at the encoder side (referenced to the original video before manipulating the coding modes of misclassified MBs) as well as the embedding capacity. Table 4.26 depicts the results for each test sequence averaged over all QS values for IBPBP... sequence. As shown in the table, the average PSNR drop at the encoder compared to the original video for all video sequences is 0.12 dB which is a slight acceptable drop,

Table 4.26: PSNR drop, bitrate increase and embedding capacity at MPEG2 encoder for IBP... sequence as a result of the combined encoder-decoder solution, averaged over all QS values.

Sequence ID	PSNR drop [dB]	Bitrate increase [%]	Embedding rate [bits/MB]
1	0.09	6.63	2.06
2	0.06	2.72	1.32
3	0.26	8.15	1.28
4	0.14	12.87	2.15
5	0.12	5.93	1.24
6	0.11	5.90	1.23
7	0.05	1.88	1.69
8	0.17	8.38	2.18
9	0.08	2.96	1.31
Average	**0.12**	**6.16**	**1.61**

and the drop in embedding rate from 1.68 to 1.61 bits/MB is also slight, with an advantage that embedding with this rate of 1.61 bits/MB will result in extracting them all correctly and error-free at the decoder as opposed to embedding with a higher rate of 1.68 bits/MB at the encoder but having 4% of them on average (1-accuracy) wrongly extracted at the decoder. The average increase in the bitrate at the encoder compared to the original video is 6.16%.

4.2 HEVC Results

In this section, we present the results obtained from repeating the experiments on HEVC-encoded videos. The experiments for HEVC were conducted using only P frames (IPPP... encoding sequence with N=100, GOP=100, M=1). For simplification, the size of the coding units (CUs) was fixed to 16×16 in all the experiments. For each video sequence, four quantization scale values were used for compression: 22, 27, 32 and 37 and for each of them, RF classifier was trained on the first 10% of the video, i.e. first 10 frames, and then used to classify the motion vectors of the rest 90% of the same video, i.e. last 90 frames of the video sequence.

4.2.1 Classification and post processing. Classification accuracy and the different types of errors obtained by RF for IPP... sequence are presented in Table 4.27 for all the 4 QS values as an average over all the 9 test video sequences. As expected, with

Table 4.27: RF results for HEVC using an IPPP... sequence.

	QS 22	QS 27	QS 32	QS 37
Model accuracy	93.83	92.93	93.30	92.92
% of Type I error	1.04	1.04	1.07	1.05
% of Type II error	2.92	3.70	3.19	3.68
% of Classification error	2.21	2.33	2.44	2.35
Fixed % of Type I error	83.72	89.43	89.70	83.28
Fixed % of Type II error	73.12	73.91	68.80	60.00
Final accuracy	**96.83**	**96.59**	**96.45**	**96.00**

the increase in QS value from 22 to 37, the classification accuracy decreases, with the smallest QS value having the highest average classification accuracy of 96.83% and the largest QS value having the lowest average classification accuracy of 96.00%. When comparing these results of HEVC with their counterparts in MPEG2, we find a compatible behaviour; an overall average final accuracy of 96.47% for HEVC compared to 96.34% for MPEG2.

4.2.2 Embedding capacity. Table 4.28 illustrates the embedding capacities for each test video sequence for every quantization scale value. Again as expected, as the value of QS used to compress a video sequence increases, the embedding capacity for that sequence decreases accordingly. The table shows an overall embedding capacity of 0.96 bits/MB (averaged over all test sequences and all QS values) which is found to be lower than its counterpart in MPEG2 video codec standard; 1.07 bits/MB. This can be due to the high efficiency of HEVC compression mechanism that may result in higher percentage of skipped macroblocks, thus lower availability of macroblocks to be used for data embedding.

4.2.3 Effect of missclassification on the decoded video quality. We tested the effect of misclassification at the decoder on the PSNR of the reconstructed HEVC video using the same encoding scheme of IPPP... sequence. The experiments were conducted twice; once with no intermediate I frames (N=100, GOP=100, M=1, i.e. only one I frame, followed by a sequence of only P frames) and then with a periodic I frame every 12 frames (N=100, GOP=12, M=1, i.e. an I frame every 12 frames, followed by a sequence of P frames). The results are shown in Table 4.29 for each

Table 4.28: Embedding capacity in bits/MB for HEVC using an IPPP... sequence.

Sequence ID	QS 22	QS 27	QS 32	QS 37	Avg per seq.
1	1.53	1.45	1.39	1.35	1.43
2	0.84	0.76	0.69	0.64	0.73
3	0.81	0.75	0.71	0.67	0.74
4	1.63	1.60	1.55	1.49	1.57
5	0.80	0.78	0.76	0.75	0.77
6	0.46	0.44	0.42	0.41	0.43
7	0.99	0.90	0.83	0.78	0.88
8	1.50	1.46	1.44	1.41	1.45
9	0.71	0.69	0.66	0.64	0.68
Average	**1.03**	**0.98**	**0.94**	**0.90**	**0.96**

Table 4.29: PSNR drop [in dB] in HEVC IPP... decoded sequence as a result of misclassification at the decoder, averaged over the 4 QS values.

Sequence ID	With I frames	Without I frames
1	6.19	8.82
2	3.22	4.01
3	6.79	7.01
4	6.39	6.91
5	5.73	6.15
6	5.99	6.30
7	7.90	8.37
8	7.07	7.61
9	7.58	8.03
Average	**6.32**	**7.02**

test sequence averaged over all the 4 QS values. Referring to the table, the overall PSNR drop averaged over all sequences and QS values is 7.02 dB without intermediate I frames, whereas inserting periodic I frames mitigated the effect of error propagation on the quality of the decoded video resulting in decreasing the PSNR drop to 6.32 dB. Again, we can conclude that having periodic I frames results in better quality of the decoded video, however, the embedding capacity will drop below the 0.96 bits/MB in this case.

4.2.4 Combined encoder-decoder solution. As mentioned in MPEG2 section, this solution is evaluated in terms of bitrate increase and PSNR drop at the encoder side as well as the embedding capacity. Table 4.30 illustrates the results for each

Table 4.30: PSNR drop, bitrate increase and embedding capacity at HEVC encoder as a result of the combined encoder-decoder solution, averaged over all 4 QS values.

Sequence ID	PSNR drop [dB]	Bitrate increase [%]	Embedding rate [bits/MB]
1	0.11	5.99	1.39
2	0.08	2.84	0.68
3	0.24	7.73	0.68
4	0.16	11.60	1.51
5	0.14	5.94	0.72
6	0.11	5.66	0.38
7	0.07	1.80	0.83
8	0.21	8.66	1.41
9	0.13	2.75	0.62
Average	**0.14**	**5.89**	**0.91**

video sequence averaged over all the 4 QS values for HEVC IPPP... encoding sequence. From the table, the average PSNR drop at the HEVC encoder compared to the original video for all video sequences is 0.14 dB and the embedding rate has dropped slightly from 0.96 to 0.91 bits/MB due to excluding the wrongly-classified MBs at the encoder from the data embedding process. In addition, the average increase in video bitrate obtained at the encoder was 5.89%. At the decoder side, a classification accuracy of 100% was obtained, accordingly the embedded message bits were 100% accurately extracted and no drop in the PSNR of the decoded video was encountered. Overall, the results reported for HEVC coding scheme have shown similar behaviour to their MPEG2 counterparts.

4.3 Comparison with Post-Processing-Only Classification and with Existing Solutions

To the far of our knowledge, none of the previous work in data embedding in video streams used motion vector rotation to embed the message bits in the host video. Therefore, to show the superiority of our novel work, we compare it with only using SAD as a decision criteria, where no feature extraction is performed at the decoder and hence no model training is conducted. In this case, when the decoder receives the motion vectors in the bit stream, it only performs the rotation using the aforementioned four rotation angles, and then motion compensation is performed and 4 or 16 candidate

macroblocks are reconstructed for each received macroblock. After which, the sum of absolute difference is calculated between each candidate macroblock and its top and left macroblocks of the same frame. Then, the macroblock with the lowest SAD value among the 4 or 16 candidates is classified by the decoder as the true macroblock corresponding to the true unrotated motion vector(s). To show the superiority of our machine learning approach over only using SAD, we provide a comparison between the two approaches in Table 4.32 by reporting the average classification accuracies over all test sequences for the 3 QS values using MPEG2 IPPP... sequence. Overall, our approach (machine learning followed by SAD as post processing) outperformed the classification using only SAD by an average of 20.5% for all QS values. For detailed reference, Table 4.31 reports the accuracies obtained by using SAD only for classification for each test sequence for every QS value. Despite having a combined encoder-decoder solution that provides a 100% classification accuracy, the comparison of using only SAD is performed with our first solution (decoder-only solution) for two reasons:
1- Unify the base of comparison to make it fair; by comparing the classification accuracy of the two approaches while having the same exact embedding rates.
2- Demonstrate that our proposed work provides better classification accuracy than only using SAD, even in its worst case scenario.

However, in terms of the drop in the PSNR of the reconstructed video at the decoder and the embedding capacity, we compare our best proposed solution (the combined encoder-decoder solution) with some of the previous data embedding solutions that exist in the related literature as illustrated in Table 4.33. As a summary, Figure 4.2 depicts that our proposed solution is superior to all the other mentioned previous solutions in terms of both the embedding rate and the PSNR drop at the decoder. That is; the highest embedding rate and the lowest PSNR drop in the decoded video. The proposed solution has an average embedding capacity of 1.61 bits/MB which is approximately three times the average embedding capacity of all the mentioned existing solutions together. Moreover, the proposed solution has a zero drop in the PSNR of the reconstructed video at the decoder compared to an average of 0.3 dB for all the existing solutions combined.

Table 4.31: Classification accuracy [in %] at the decoder for MPEG2 IPPP.. sequence when only using SAD for classification.

Sequence ID	QS 5	QS 15	QS 25
1	75.95	75.65	77.78
2	56.00	67.14	63.08
3	78.40	75.82	71.33
4	82.47	79.17	78.82
5	83.13	77.21	75.58
6	89.57	84.96	81.25
7	60.84	60.00	60.01
8	93.32	85.75	84.32
9	83.58	76.16	71.02

Table 4.32: Comparison of the proposed solution to only using SAD for classification using MPEG2 IPPP.. sequence.

QS	Machine Learning with SAD [%]	Only SAD [%]
5	96.74	78.14
15	96.39	75.76
25	95.89	73.68

4.4 Sequence-Independent Approach

As described and experimented in all the previous sections, the whole proposed work is designed and evaluated based on the sequence-dependent approach. That is; training the machine learning models on the first portion of a video, then testing the generated models on the rest of the same video. The proposed solutions proved to perform well with video-dependent approach and showed superiority over many other existing solutions. However, for completeness and full exploration of the available options, in this section we examine the video-independent approach by training the

Table 4.33: Comparison of the proposed solution with existing solutions in terms of PSNR drop and embedding capacity at the decoder side.

Solution	Embedding rate [bits/MB]	PSNR drop [dB]
Proposed	**1.61**	**0**
Ref. [13]	0.75	0.51
Ref. [4]	1.06	0.20
Ref. [16]	0.05	0.20
Ref. [15]	0.08	0.17

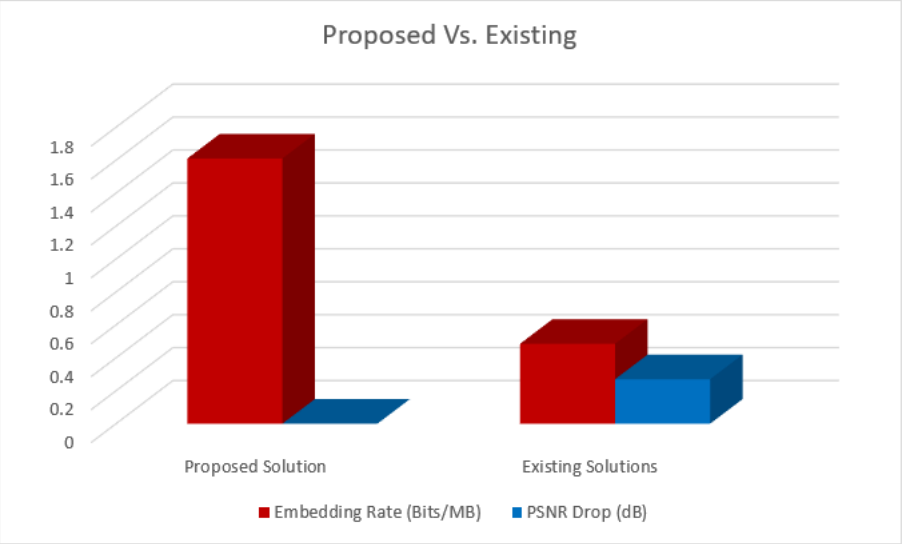

Figure 4.2: Comparison of the proposed solution with existing solutions.

classifiers on some video sequences and testing them on a completely different video sequence that could be with different spatial and temporal activities. The experiments were conducted in a round-robin fashion by training the classifier each time using a different combination of 8 video sequences and testing it on the 9^{th} video sequence, resulting in having each of the 9 video sequences to be tested once. The sequences used for experiments were compressed with HEVC encoder using an IPPP... encoding sequence and QS value of 27. The training and testing processes took place using Random Forests. Table 4.34 illustrates the classification accuracy obtained for each test video sequence.

From the table, the average classification accuracy for all the 9 test video sequences using sequence-independent training is found to be 79.28%, which is much lower than the classification accuracies obtained by the different types of classifiers when we used the sequence-dependent training approach. For example, the obtained overall accuracy when using Random Forests was 92.93% on average. This could be due to the fact that in sequence-independent approach, the sequences used for training the Random Forests classifier have different spatial activities and different visual patterns than the tested video sequences.

Table 4.34: Classification accuracy using RF for each video sequence using sequence-independent training.

Sequence ID	Classification accuracy [%]
1	82.24
2	78.77
3	80.96
4	81.46
5	82.00
6	78.44
7	78.22
8	74.14
9	77.32

Consequently, we can conclude that the proposed motion vector rotation solution for data embedding performs well with the video-dependent approach, but not with the video-independent approach.

Chapter 5. Conclusion and Future Work

Data embedding in video plays an important role in different applications including digital rights management, content authentication, law enforcement and error resiliency and concealment. Recently, videos and other sensitive data are preferred to be transmitted and stored in an encrypted form to avoid unauthorized access and possible attacks.

In this work, we proposed a novel data embedding scheme that jointly serves both data hiding and video scrambling at the encoder through motion vector rotation. A machine learning solution was then proposed to distinguish between the true motion vectors and the rotated ones, and consequently extract the message bits and reconstruct the cover video at the decoder.

Experiments were conducted on well-known video sequences compressed using MPEG2 video standard, while examining different quantization scales and various machine learning techniques with and without feature selection. A detailed analysis was provided based on the type of the encoding sequence, the macroblock type and the number of motion vectors. The experiments were then repeated for HEVC video codec standard to validate the generality and applicability of our proposed solution over different video coding standards. Results have shown that much better classification accuracy was obtained by the proposed machine learning approach compared to using SAD only as a decision criteria. Moreover, better classification results were attained by random forests compared to support vector machine and polynomial classifiers. Applying feature selection did not improve the classification accuracy obtained by the models, however, it helped to identify the most influential features which found to be the phase differences between a candidate MB and its surrounding MBs as well as the sum of absolute values of prediction errors. Testing our approach for different quantization scale values showed that as the value of the quantization scale used for compression increases, the classifier accuracy decreases and also does the embedding rate. Two different encoding sequences were used; a sequence with only P frames and a sequence with incorporating both P and B frames. It was clearly shown that including B frames leads to a considerable increase in embedding rate due to the ability of bi-

directional MBs to hide 4 bits instead of only 2 bits as in P MBs. It was also concluded that the number of candidate MBs is irrelevant to the classification accuracy as all the three MB types; P MBs, mono-B MBs and bi-B MBs showed very similar classification accuracies.

The effect of misclassification at the decoder on the quality of the reconstructed video was investigated and reported. Then, to eliminate this effect, a combined encoder-decoder solution was proposed and evaluated. This solution proved to guarantee the following at the decoder: a 100%-classification accuracy, a completely error-free message extraction and correct reconstruction of the video without any drop in its quality. On the other hand, it led to a slight increase in the video bitrate at the encoder.

At last, when comparing with some previous data embedding solutions that exist in the related literature, our proposed solution outperformed in terms of both data embedding capacity and reconstruction quality.

As for future work, since the current solution provides the maximum performance in terms of classification accuracy (100%) and reconstruction quality (zero drop in PSNR) but slightly decreases the embedding capacity, more effort can be put in the future to further increase the embedding rate. This can be achieved in multiple ways. For example, by applying the motion estimation on smaller blocks, that is; on a sub-MB-basis instead of a MB-basis, which will result in more MVs and thus more embedded data. Another idea is to conduct the experiments with increasing the number of B frames in the video sequence, i.e. an IBBPBBP... sequence with M=3, this may increase the availability of bi-directional MBs in the video sequence and thus increase the embedding capacity.

References

[1] R. J. Mstafa and K. M. Elleithy, "Compressed and raw video steganography techniques: a comprehensive survey and analysis," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 749–21 786, 2017.

[2] Y. Liu, S. Liu, Y. Wang, H. Zhao, and S. Liu, "Video steganography: A review," *Neurocomputing*, vol. 335, pp. 238–250, 2019.

[3] T. Shanableh, "Matrix encoding for data hiding using multilayer video coding and transcoding solutions," *Signal Processing: Image Communication*, vol. 27, no. 9, pp. 1025–1034, 2012.

[4] D. Xu, R. Wang, and Y. Q. Shi, "Data hiding in encrypted H.264/AVC video streams by codeword substitution," *IEEE transactions on information forensics and security*, vol. 9, no. 4, pp. 596–606, 2014.

[5] ——, "An improved scheme for data hiding in encrypted H.264/AVC videos," *Journal of Visual Communication and Image Representation*, vol. 36, pp. 229–242, 2016.

[6] E. Peixoto, T. Shanableh, and E. Izquierdo, "H.264/AVC to HEVC video transcoder based on dynamic thresholding and content modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 1, pp. 99–112, 2013.

[7] M. Hassan and T. Shanableh, "Predicting split decisions of coding units in HEVC video compression using machine learning techniques," *Multimedia Tools and Applications*, vol. 78, no. 23, pp. 32 735–32 754, 2019.

[8] G. Correa, P. A. Assuncao, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC encoding decisions using data mining," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 660–673, 2014.

[9] F. S. Rovati, D. Pau, E. Piccinelli, L. Pezzoni, and J.-M. Bard, "An innovative, high quality and search window independent motion estimation algorithm and architecture for MPEG-2 encoding," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, pp. 697–705, 2000.

[10] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (HEVC)," in *Integrated Circuit and Systems, Algorithms and Architectures*. Springer, 2014, vol. 39, pp. 49–90.

[11] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[12] D. Xu, Y. Zhu, R. Wang, J. Fu, and K. Chen, "Two-dimensional histogram modification for reversible data hiding in partially encrypted H.264/AVC videos," in *International Workshop on Digital Watermarking*. Springer, 2016, pp. 393–406.

[13] P.-C. Su, T.-F. Tsai, and Y.-C. Chien, "Partial frame content scrambling in H.264/AVC by information hiding," *Multimedia Tools and Applications*, vol. 76, no. 5, pp. 7473–7496, 2017.

[14] Y. Wang, F. Kurugollu *et al.*, "Privacy region protection for H.264/AVC with enhanced scrambling effect and a low bitrate overhead," *Signal Processing: Image Communication*, vol. 35, pp. 71–84, 2015.

[15] D. Xu and R. Wang, "Context adaptive binary arithmetic coding-based data hiding in partially encrypted H.264/AVC videos," *Journal of Electronic Imaging*, vol. 24, no. 3, p. 033028, 2015.

[16] D. Xu, R. Wang, and Y. Zhu, "Tunable data hiding in partially encrypted H.264/AVC videos," *Journal of Visual Communication and Image Representation*, vol. 45, pp. 34–45, 2017.

[17] M. Long, F. Peng, and H.-y. Li, "Separable reversible data hiding and encryption for HEVC video," *Journal of Real-Time Image Processing*, vol. 14, no. 1, pp. 171–182, 2018.

[18] D. Xiao, Y. Xiang, H. Zheng, and Y. Wang, "Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism," *Journal of Visual Communication and Image Representation*, vol. 45, pp. 1–10, 2017.

[19] X. Wu, B. Chen, and J. Weng, "Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 58–64, 2016.

[20] C. Qin and X. Zhang, "Effective reversible data hiding in encrypted image with privacy protection for image content," *Journal of Visual Communication and Image Representation*, vol. 31, pp. 154–164, 2015.

[21] D. Xu and R. Wang, "Separable and error-free reversible data hiding in encrypted images," *Signal Processing*, vol. 123, pp. 9–21, 2016.

[22] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2011.

[23] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo, "High capacity reversible data hiding in encrypted images by patch-level sparse representation," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1132–1143, 2015.

[24] S. Yi and Y. Zhou, "Binary-block embedding for reversible data hiding in encrypted images," *Signal Processing*, vol. 133, pp. 40–51, 2017.

[25] V. Y. Kulkarni and P. K. Sinha, "Pruning of random forest classifiers: A survey and future directions," in *2012 International Conference on Data Science & Engineering (ICDSE)*. IEEE, 2012, pp. 64–68.

[26] F. Livingston, "Implementation of breiman's random forest machine learning algorithm," *ECE591Q Machine Learning Journal Paper*, pp. 1–13, 2005.

[27] V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, and M. Chica-Rivas, "Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines," *Ore Geology Reviews*, vol. 71, pp. 804–818, 2015.

[28] A. Pradhan, "Support vector machine-a survey," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 8, pp. 82–85, 2012.

[29] K. S. Durgesh and B. Lekha, "Data classification using support vector machine," *Journal of Theoretical and Applied Information Technology*, vol. 12, no. 1, pp. 1–7, 2010.

[30] K.-A. Toh, Q.-L. Tran, and D. Srinivasan, "Benchmarking a reduced multivariate polynomial pattern classifier," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 740–755, 2004.

Vita

Afaf Eltayeb Mohamedelbagir Ahmed was born in 1994, in Hail, Kingdom of Saudi Arabia. She started her elementary education in Hail, then she moved back to her country of origin, Sudan, and continued her secondary education in Asmaa Abd-Alraheem private secondary school in Khartoum. She was one of the top 20 students in Sudan in the Sudanese Certificate exam. She joined the University of Khartoum, the Faculty of Electrical and Electronic Engineering in 2011 for her Bachelor degree, and graduated in 2016 with a first class. After graduation, she worked as a teaching assistant in her undergraduate university for two semesters, then started working at Nile Centre for Technology and Research as a network and systems administrator after which she moved to Sudatel Company for Mobile Networks and worked as a Radio Network Optimization engineer.

In spring 2019, she joined the American University of Sharjah for pursuing her Master's degree in Computer Engineering beside working as a teaching and research assistant meanwhile. Afaf also co-authored one paper during her Master's study, titled "Predictive Data Embedding in Scrambled Video by Rotating Motion Vectors".