International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:8, 2014

DWT Based Image Steganalysis

Indradip Banerjee, Souvik Bhattacharyya, Gautam Sanyal

***Abstract*—**'Steganalysis' is one of the challenging and attractive interests for the researchers with the development of information hiding techniques. It is the procedure to detect the hidden information from the stego created by known steganographic algorithm. In this paper, a novel feature based image steganalysis technique is proposed. Various statistical moments have been used along with some similarity metric. The proposed steganalysis technique has been designed based on transformation in four wavelet domains, which include Haar, Daubechies, Symlets and Biorthogonal. Each domain is being subjected to various classifiers, namely K-nearest-neighbor, K* Classifier, Locally weighted learning, Naive Bayes classifier, Neural networks, Decision trees and Support vector machines. The experiments are performed on a large set of pictures which are available freely in image database. The system also predicts the different message length definitions.

***Keywords*—**Steganalysis, Moments, Wavelet Domain, KNN, K*, LWL, Naive Bayes Classifier, Neural networks, Decision trees, SVM.

I. Introduction

INFORMATION HIDING [1] has become the center of attention in the ground of research at this instant. This is the science and art of hiding a message signal in a host signal, such as text, audio, video and images without any undetectable distortion of the host signal. Steganography [2] is one of the most important information hiding applications. Steganography techniques attempt to hide the existence of the message itself, so that an observer or eavesdropper does not know that the information is even there or not.

The steganography is now become a challenging situation because the process of reverse engineering is becoming a best research area for the researchers. To achieve secure and undetectable communication, the stego-objects which containing a secret message and cover-objects when there is no secret message are indistinguishable. In this respect the technique of steganalysis is the process that aims to distinguish between cover-objects and stego-objects. Steganalysis can be implemented in two types of attacks, one is passive and another is active style [3]. A passive attack merely examines the message and tries to determine if it potentially contains a hidden message. On the other hand the active attack can alter messages deliberately, even though there may not see any trace of a hidden message, in order to halt any secret communication that can yet be occurring. This paper concerned with passive attack Steganalysis which can investigate embedded message length or hidden message location or secret key used in embedding. It should be noted that although there has been quite some effort in the steganalysis of digital images [4]-[7].

Indradip Banerjee and Gautam Sanyal are with the National Institute of Technology, Durgapur, India (e-mail: indradip.banerjee@yahoo.com, nitgsanyal@gmail.com).

Souvik Bhattacharyya, University Institute of Technology, The University of Burdwan, Burdwan (e-mail: souvik.bha@gmail.com).

Steganalysis is the science of detecting hidden information. The main objective of Steganalysis is to break steganography and the detection of stego is the goal of steganalysis. Steganalysis deals with three important categories (Fig. 1) [8]:

(a) Visual attacks: With a support of a computer system or inspection by human naked eye it has disclosed the presence of hidden information, which helps to separate the cover and stego.
(b) Statistical attacks: These types of attacks are more powerful and successful, because they disclose the smallest variation between the cover and stego. Statistical attacks can be further divided into two types i.e. (i) Passive attack and (ii) Active attack.
(c) Structural attacks: The format of the data files changes based on the data to be embedded. Identifying these characteristic structure changes of stego can help to find out the presence of information.

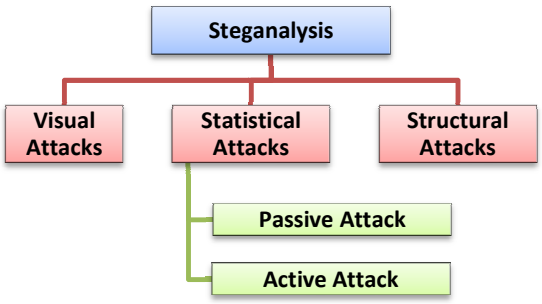


Fig. 1 Types of Steganalysis

There are four grounds of research i.e. text, image, audio and video related to Steganalysis. The details discussion of image steganalysis methods are given in Fig. 2.

A. Image Based Steganalysis

All the steganalysis algorithms work on the Steganographic algorithms establishing statistical differences between cover and stego image. Steganalysis can be consider as a two-class pattern classification problem which intend to determine whether a testing medium is cover or stego.

Targeted Steganalysis: It works on a specific type of stego-system and sometimes inadequate on image formats. Image statistics can change after embedding which is one of the parameter to find out the embedding algorithm. The results from most targeted steganalysis techniques are very accurate but the techniques are not able to extend for other embedding algorithms.

Blind Steganalysis: This can work on every embedding technique and all image formats. This algorithm can make difference of pure and stego images through the statistical properties and distinguishes between them. The machine learning process is done by training the on a large image database. Blind techniques are usually less accurate but more expandable.

Semi-blind Steganalysis: This system can work on a specific range of diverse stego-systems. The range of the stego-systems can only be depending on the specific domain they embed on, i.e. spatial or transform domain.

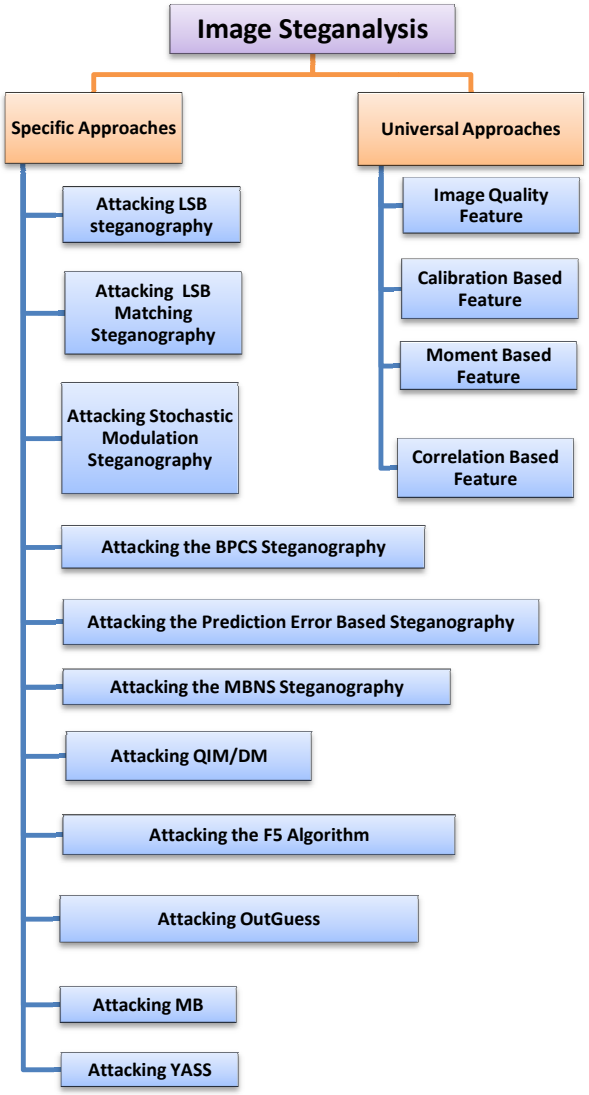


Fig. 2 Classification of Image Steganalysis through various approaches

1) Specific Approaches of Image Based Steganalysis

Some specific Steganalysis methods for attacking the steganographic schemes are established here.

a) Attacking LSB Steganography

One of the most important spatial domain related steganography techniques is LSB steganography, where extensive amount of work have been carried out by the researchers in the initial stage of the development of steganalysis and many steganalytic methods toward LSB steganography have been proved most successful, such as Chi-square statistical attack [9], [10], RS analysis [11], sample pair analysis (SPA) analysis [12], weighted stego (WS) analysis [13] and structural steganalysis [14], [15], etc.

b) Attacking LSB Matching Steganography

It is related to frequency domain technique [16]. Equal trend of the frequency of occurrence of PoVs no longer exists for LSB matching steganography. LSB matching, or more general ±k steganography, may be modeled in this context.

c) Attacking Stochastic Modulation Steganography

Stochastic modulation steganography include stego-noise with a specific probability distribution into the cover image to embed secret message bits. In [17] it has shown that the horizontal pixel difference histogram of a natural image can be modeled as a generalized Gaussian distribution (GGD). A quantitative approach to steganalyse stochastic modulation steganography was presented in [18], [19].

d) Attacking the BPCS Steganography

In BPCS steganography, the random binary patterns of data-blocks are used and the complexities of the data-blocks follow a Gaussian distribution [20]. Binary patterns of the image blocks are not random and do not follow a Gaussian distribution for high significant bit-planes.

e) Attacking the Prediction Error Based Steganography

Zhang et al. [21] proposed a technique for attacking PVD steganography based on observing the histogram of the prediction errors.

f) Attacking the MBNS Steganography

Any abnormality between a cover image and its MBNS stego image through the histogram of pixel values and the histogram of pixel prediction errors has been observed [22]. For any base value some small symbols are generated than large symbols in the process of converting binary data to symbols. The attacking algorithm works through this way.

g) Attacking QIM/DM

QIM/DM has been formulated into two sub-issues by Sullivan et al. [23]. One is to distinguish the standard QIM stego objects from the plain-quantized cover objects and another is to differentiate the DM stego objects from the un-quantized cover objects.

h) Attacking the F5 Algorithm

Some crucial characteristics of the histogram of DCT coefficients are preserved by the F5 algorithm. But F5 does modify the shape of the histogram of DCT coefficients. This drawback is employed by Fridrich et al. [24] to launch an attack against F5.

i) Attacking OutGuess

Fridrich et al. [25] found a new path to detect OutGuess quantitatively by measuring the discontinuity along the boundaries of 8x8 JPEG grids.

j) Attacking MB

MB steganography uses a generalized Cauchy distribution model to control the data embedding operation. Bohme and Westfeld [26] observed that the histogram of the DCT coefficients in a natural image is not always conforming the distribution. There exist more outlier high precision bins in the histogram in a cover image than in a stego image.

k) Attacking YASS

The locations of the H-blocks of YASS are determined by a key, which is not available to Wendy. Therefore, it may not be straightforward for Wendy to observe the embedding artifacts. Li et al. [27] proposed a method for attacking the YASS.

2) Universal Approaches

Specific steganalytic methods require knowing the details of the targeted steganographic methods, whereas the universal steganalysis [28] requires less or even no such priori information. A universal steganalytic approach usually takes a learning based strategy i.e. a training stage and a testing stage. In the following, some typical universal steganalytic features has been discussed.

a) Image Quality Feature

The statistical evidence absent by steganography may be captured by a group of IQMs and then broken for detection [29]. Steganographic schemes may add or less cause some forms of degradation to the image. In order to seek sensitive specific quality measures, consistent and monotonic steganographic artifacts and distortions, the analysis of variance (ANOVA) technique is exploited.

b) Calibration Based Feature

Fridrich et al. [30] applied the feature-based classification together with calibration to devise and blind detector specific to JPEG images. Calibration means that some parameters of the cover image may be recovered by using the stego image. So, the calibration process increases the features' sensitivity to the embedding amendment of various images. Applying calibration to the Markov process based features described in [31] and reducing their dimension and Pevny et al. merged the resulting feature sets to produce a 274-dimensional feature vector [32].

c) Moment Based Feature

The impact of steganography to a cover image can be regarded as introducing some stego-noise. Lyu and Farid [33] used the assumption that the PDF of the frequency domain subband coefficients and the prediction error of the subband coefficients altered after data embedding. A three-level wavelet decomposition[13], the first four PDF moments, i.e., mean, variance, skewness, and kurtosis, of the subband coefficients at each high-pass orientation of each level are utilize as one set of features.

d) Correlation Based Feature

Correlation in an image has changed during data embedding. Here the correlation is mainly referred to the inter-pixel dependency for a spatial image and the intra-block or inter-block DCT coefficient dependency for a JPEG image. Sullivan et al. [34] modeled the inter-pixel dependency by Markov chain and depicted it by a gray-level co-occurrence matrix (GLCM) in practice.

A study of Steganalysis and development of a steganalyzer with the help of some image features like central moment, invariant moment, Zernike moment, standard deviation and spam features of wavelet domain has been recycled in this literature. This paper has been organized as following sections: Section II describes some review works of image steganalysis, various methods for image feature selection, various machine learning procedure. Section III describes the proposed methodology. Section IV describes the algorithm of steganalyzer. Experimental Results and analyses of the method has been discussed in Section V and Section VI draws the conclusion.

II. Associated Work

Image steganalysis can be grouped into two broad categories which have discussed earlier, namely specific steganalysis approach and universal steganalysis approach. The *specific steganalysis techniques* [35] are designed for a targeted embedding technique and worked by first analysing the embedding operation and next step is to identify some features of the cover image that become modified as a result of the embedding process. The design of specific steganalysis techniques requires detailed knowledge of the steganographic embedding process and results a very accurate decisions when they are used against the particular steganographic technique. *A universal steganalytic approach* [36] usually adopts a learning based strategy involving training as well as a testing stage. In this process, a feature extraction step is required which is used in both training and testing stage. This feature extraction step is used to map an input image from a high-dimensional image space to a low-dimensional feature space. The training stage results a trained classifier. Out of many effective classifiers, like Fisher linear discriminant (FLD) [37], support vector machine (SVM) [38], *k*-nearest neighbour (KNN) [39], neural network (NN) [40], etc., anyone can be chosen. Decision boundaries are created by the classifier to separate the feature space into positive regions and negative regions with the help of the generated feature vectors extracted from the training images. In the testing stage, with the help of the trained classifier with a specific decision boundary, an image can be classified according to its feature vector's domination in the feature space. If the feature vector identifies a region where the classifier is labelled as positive, the testing image is classified as a positive class or the stego image. Otherwise, it is classified as a negative class or the cover image.

A. Image Features Selection Methods

Some methods are for image Feature Selection, these are given below:

1) Central Moments

In probability theory and statistics, central moments [41] form one set of values by which the properties of a probability distribution can be usefully characterized. Central moments are used in preference to ordinary moments because then the values' higher order quantities relate only to the spread and shape of the distribution, rather than to its location. There are two ways of viewing moments, one based on statistics and one based on arbitrary functions such as *f(x)* or $f(x, y)$. As a result moments can be defined in more than one way. Moments are the statistical expectation of certain power functions of a random variable. The most common moment is the mean which is just the expected value of a random variable: where *f*(*x*) is the probability density function of continuous random variable *X*. More generally, moments of order $p = 1,2,3, \dots, n$ can be calculated as $m_p = \mathrm{E}\left[\mathrm{X}^p\right]$. These are sometimes referred to as the raw moments. There are other kinds of moments that are often useful. One of these is the central moments $m_p = \mathrm{E}\left[(\mathrm{X} - m)^p\right]$. The best known central moment is the second, which is known as the variance. Two less common statistical measures, skewness and kurtosis, are based on the third and fourth central moments. The use of expectation assumes that the PDF is known. Moments are easily extended to two or more dimensions. For example: where $f(x, y)$ is the joint PDF.

2) Invariant Moments:

In many applications such as shape recognition, it is useful to generate shape features which are independent of parameters which cannot be controlled in an image [41]. Such features are called invariant features. M.K. Hu [41] derived a transformation of the normalized central moments to make the resulting moments rotation invariant. Equation (1) represents the mathematical formulation of Invariant moments.

$$
\begin{aligned}
&p + q = 2\\
&\boldsymbol{\delta_1} = \zeta_{20} + \zeta_{02}\\
&\boldsymbol{\delta_2} = (\zeta_{20} + \zeta_{02})^2 + 4{\zeta_{11}}^2\\
&p + q = 3\\
&\boldsymbol{\delta_3} = (\zeta_{30} - 3\zeta_{12})^2 + (\zeta_{03} - 3\zeta_{21})^2\\
&\boldsymbol{\delta_4} = (\zeta_{30} + \zeta_{12})^2 + (\zeta_{03} + \zeta_{21})^2\\
&\boldsymbol{\delta_5} = (\zeta_{30} - 3\zeta_{12})(\zeta_{30} + \zeta_{12})[(\zeta_{03} + \zeta_{12})^2 - 3(\zeta_{21} + \zeta_{03})^2]\\
&\qquad + (\zeta_{03} - 3\zeta_{21})(\zeta_{03} + \zeta_{21})[(\zeta_{30} + \zeta_{21})^2 - 3(\zeta_{12} + \zeta_{30})^2]\\
&\boldsymbol{\delta_6} = (\zeta_{20} - \zeta_{02})[(\zeta_{30} + \zeta_{12})^2 - (\zeta_{21} + \zeta_{03})^2] + 4\zeta_{11}(\zeta_{30} + \zeta_{12})(\zeta_{03} + \zeta_{21})\\
&\boldsymbol{\delta_7} = (3\zeta_{21} - \zeta_{03})(\zeta_{30} + \zeta_{12})[(\zeta_{30} + \zeta_{12})^2 - 3(\zeta_{21} + \zeta_{03})^2] +\\
&\qquad (\zeta_{30} - 3\zeta_{12})(\zeta_{21} + \zeta_{03})[(\zeta_{03} + \zeta_{21})^2 - 3(\zeta_{30} + \zeta_{12})^2] \quad (1)
\end{aligned}
$$

3) Zernike Moments

For an image (or other) function $f(r, \theta)$, the Zernike moments [42] are given as follows (2):

$$
\begin{aligned}
A_{nl} &= < f, V_{nl} >\\
&\approx \frac{n+1}{\pi} \iint_{r \le 1} f(r, \theta) V_{nl}^*(r, \theta) r \, dr \, d\theta\\
&\approx \frac{n+1}{\pi} \Delta x \, \Delta y \, \sum_i \sum_j f(x_i, y_j) \, V_{nl}^*(x_i, y_j) \qquad (2)
\end{aligned}
$$

where $x = r\cos(\theta)$ and $y = r \sin(\theta)$, and the function *f* must be rescaled so that it is contained in the unit circle.

4) Standard Deviation

This feature measures the mean of the deviation of the image elements before and after reconstruction. Standard deviation [43] is an important statistical measure that provides the mean or average estimation of the deviation of a data set from the calculated average. It is often applied on data sets with random predictability such as a set of age of people or the set comprising of marks obtained by a candidate in a public examination etc.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2} \qquad (3)$$

where σ represents the standard deviation, x_i represents the numbers and μ is the mean value of that numbers. Equation (3) represents the mathematical formulations of the standard deviation.

5) Spam Features

The Subtractive Pixel Adjacency Model (SPAM) [44] is one kind of image features which will used for computation of image steganalysis. The transition probabilities along eight directions are computed at first. The transition probability and the differences are always computed beside the undistinguishable direction. It can be further calculated only on the horizontal direction as the other directions are obtained in a similar manner. All direction-specific quantities will be denoted by a superscript $\{\leftarrow, \rightarrow, \downarrow, \uparrow, \nwarrow, \searrow, \swarrow, \nearrow\}$ showing the direction of the calculation. The features are calculated and computed the difference array D. The horizontal direction that is the left-to-right direction is:

$$\mathrm{D}_{i,j}^{\rightarrow} = \mathrm{I}_{i,j} - \mathrm{I}_{i,j+1} \qquad (4)$$

where $i \in \{1, \dots, m\}$, $j \in \{1, \dots, n-1\}$

A. Wavelet Domain Methods

Now followings Wavelet Domains methods are used for analysis.

1) Haar Wavelet Domain

In mathematics, the Haar wavelet [45] is a sequence of rescaled "square-shaped" functions which together form a wavelet family or basis. Wavelet analysis is similar to Fourier analysis in that it allows a target function over an interval to be represented in terms of an orthonormal function basis. The Haar sequence is now recognised as the first known wavelet basis and extensively used as a teaching example. The Haar sequence was proposed by Alfréd Haar [45]. Haar used these functions to give an example of a countable orthonormal

system for the space of square integrable functions on the real line. The study of wavelets, and even the term "wavelet", did not come until much later. As a special case of the Daubechies wavelet, the Haar wavelet is also known as *D2*. The Haar wavelet is also the simplest possible wavelet. The technical disadvantage of the Haar wavelet is that it is not continuous and therefore not differentiable. This property can, however, be an advantage for the analysis of signals with sudden transitions, such as monitoring of tool failure in machines or monitoring of the heart rate in the ECG machine. Mainly it is used for analysis of properties, features or events that have sudden transitory properties.

$$\psi_{n,k}(t) = 2^{\frac{n}{2}}\psi(2^n t - k) \quad (5)$$

where ψ is denoted as a Haar function, *n* and *k* are the pair of integer denotes the position of the matrices of the integer Z. *t* is belongs to R. R is the real number.

2) Daubechies Wavelet Domain

Named after Ingrid Daubechies, the Daubechies wavelets [46] are a family of orthogonal wavelets defining a discrete wavelet transform and characterized by a maximal number of vanishing moments for some given support. With each wavelet type of this class, there is a scaling function (called the father wavelet) which generates an orthogonal multiresolution analysis. In general the Daubechies wavelets are chosen to have the highest number *A* of vanishing moments, (this does not imply the best smoothness) for given support width *N=2A* and among the *2A−1* possible solutions, the one is chosen whose scaling filter has external phase. The wavelet transform is also easy to put into practice using the fast wavelet transform. Daubechies wavelets are widely used in solving a broad range of problems, e.g. self-similarity properties of a signal or fractal problems, signal discontinuities, etc.

$$a(Z) = 2^{1-A}(1+Z)^A p(Z) \quad (6)$$

where *p* having the real coefficient, A is the approximation order of the orthogonal discrete wavelet transform.

3) Symlets Wavelet Domain

The symlets [47] are nearly symmetrical wavelets proposed by Daubechies after the modifications of *db* family. But the properties of the both the wavelet families are more or less similar with nature. Important efficient denoising application is used among wavelet family and the Symlet wavelet is more efficient among them. Like Daubechies wavelets the Symlet wavelets are used in practice and these are selected even number of wavelets. Symlet wavelets are used when the applied signal performs better and SNR of reconstructed or denoised signal is improved.

$$\sum_{n=0}^{N-1}|y[n]|^2 = \sum_{k=0}^{N-1}\sum_{l}|H(k+lN)|^2|F(k+lN)|^2 \quad (7)$$

where *y[n]* represents the transform coefficients. *N* represents the number of transform coefficients, *k* represents the set of frequencies and *l* represents the set of integers.

4) Biorthogonal Wavelet Domain

The biorthogonal weblet has been introduced after the modifications of *db*. The biorthogonal wavelet offers a mathematical framework for describing functions at different level of resolution [48]. This wavelet is closely related to human perception.

$$\psi = \sum_{n\in Z} a_n \tilde{a}_{n+2m} = 2.\delta_{m,0} \quad (8)$$

where a and $\tilde{a}$ are the scaling sequence of the coefficients. The scaling sequences can satisfy the biorthogonality condition.

A. Methods for Classifications

Now the followings are some classifiers have been used in this contribution.

1) Ibk (K-Nearest-Neighbor)

In pattern recognition, the k-nearest neighbour algorithm (KNN) [49] is a method for classifying objects based on closest training examples in the feature space. KNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest neighbour algorithm is amongst the simplest of all machine learning algorithms. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common amongst its k-nearest neighbours (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbour. The same method can be used for regression, by simply assigning the property value for the object to be the average of the values of its k-nearest neighbours. It can be useful to weight the contributions of the neighbours, so that the nearer neighbours contribute more to the average than the more distant ones. The k-nearest neighbour algorithm is sensitive to the local structure of the data. Nearest neighbour rules in effect compute the decision boundary in an implicit manner. It is also possible to compute the decision boundary itself explicitly and to do so in an efficient manner so that the computational complexity is a function of the boundary complexity.

2) Kstar (K* Classifier)

Instance-based learners can classify an instance by comparing through pre-classified examples. The essential hypothesis is that parallel occurrences which will be parallel classifications. K* [50] is an instance-based classifier, that is the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function. It differs from other instance-based learners in that it uses an entropy-based distance function. K* is not only a distance function, it also general non-zero and non-symmetric function. Though probably the counter-

intuitive does not interfere with the development of the K* algorithm.

3) LWL (Locally Weighted Learning)

It is a lazy learning method which can comply processing of training data until a query needs to be answered [51]. Generally it involves storing the training data in memory and finding significant data in the database to response a specific query. This type of learning is furthermore called memory-based learning. Relevance is dignified using a distance function, with adjacent points having high relevance. One form of lazy learning finds a set of nearest neighbours and selects or votes on the extrapolations completed by each of the stored points. A single global model is used to fit all of the training data, observed by most learning methods. But the answer of the query is known during processing of training data, training query specific local models is promising in lazy learning.

4) NaiveBayes (Naive Bayes Classifier)

Bayes classifier [52] is one of the artless probabilistic classifier, which is perfectly established on applying Bayes' theorem. It is basically comes from Bayesian statistics with strong independence assumptions. A more colourful term for the underlying probability model would be "independent feature model". A naive Bayes classifier adopts that the presence or absence of an actual feature of a class is distinct to the presence or absence of any other feature. Liable on the defined nature of the probability model, naive Bayes classifiers can be trained very proficiently in a supervised learning setting. Many of the practical applications, parameter appraisal for naive Bayes models uses the method of maximum likelihood. On the former confrontations, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

5) Neural Networks (Multi-Layer Perceptron's)

A multilayer perceptron (MLP) [53] is an artificial neural network model that maps sets of input data onto a set of outputs. MLP consists of multiple layers of nodes in a directed graph and each layer fully connected to the next one. Excluding the input nodes, every node is a neuron with a nonlinear activation function. MLP utilizes a supervised learning technique called back propagation for training the network. Any successful pattern classification methodology depends heavily on the particular choice of the features used by that classifier .The Back-Propagation is the best known and widely used learning algorithm in training multilayer feed forward neural networks. The feed forward neural net refer to the network consisting of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computation nodes and an output layer of computation nodes. The input signal propagates through the network in a forward direction, from left to right and on a layer-by-layer basis.

6) J48 (Decision Trees)

A decision tree [54] is a decision support tool which practices a tree-like graph or decisions model mechanism and their possible significances, including chance incident outcomes, resource costs and efficacy. It is solitary approach to display an algorithm. Decision trees are commonly used in operations research, specifically in decision analysis. The decision tree most likely to reach the goal which can leads to identify the strategic resolution.

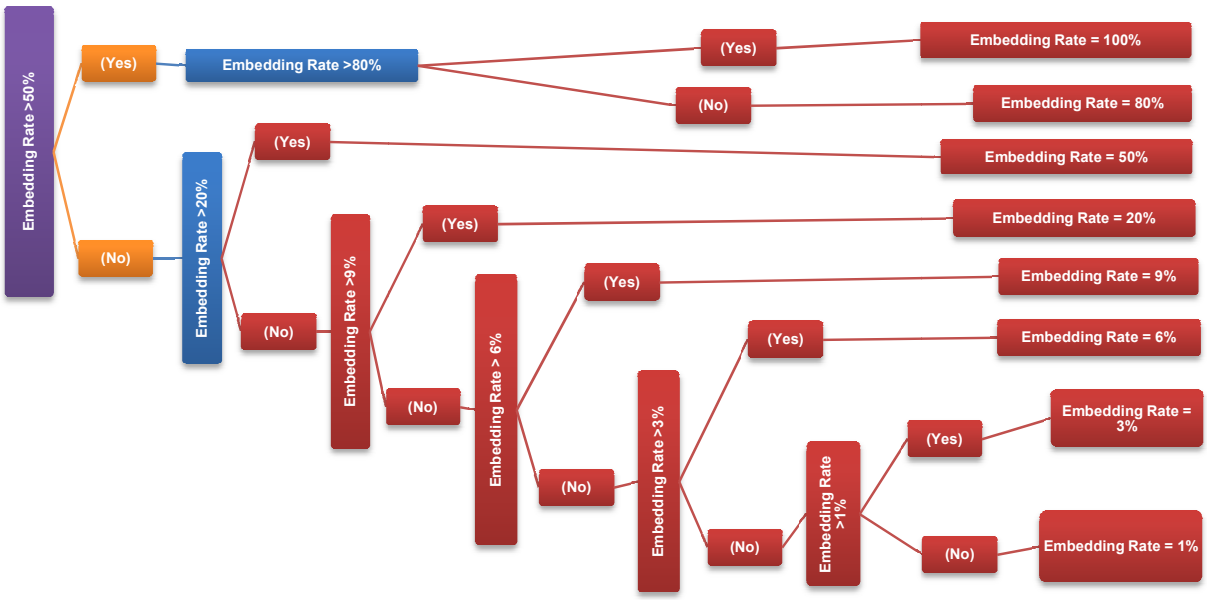


Fig. 3 Proposed Image Steganalyzer

7) SVM (Support Vector Machines)

In machine learning, support vector machines [55] are supervised learning models with associated learning algorithms which can analyse data and recognize patterns. It is mostly used for classification and regression analysis. The basic SVM takes a set of input data and predicts it, for each given input, which of two possible classes forms the input, making it a non-probabilistic binary linear classifier. SVM training algorithm builds a model that assigns new examples into one category or the other. The support vector method is based on an efficient multidimensional function optimization, which tries to minimize the empirical risk, which is the training set error. The set of vectors is said to be optimally separated if it is separated without error and the distance between the closest vectors to the hyper plane is maximal.

III. Proposed Method

Fig. 3 exemplified the entire method. At the beginning of the work the image is converted to DWT domain. After that some image steganography tools like S-Tool [56], PQ [57], Model Base [58], F5 [59] has been used for creating stego DWT images. After embedding the inverse DWT procedure has been prepared to construct the image. Various lengths of characters have been used for embedding in cover image and produce the corresponding stego images. In this contribution the authors have used Haar [45], Daubechies [46], Symlets [47] and Biorthogonal [48] methods of Discrete Wavelet Domain. In the next step extract some image features like central moments, invariant moments, Zernike moments and also extract all spam features of the images. Then store all the features in a tabular form and put it to the database. Then form a training dataset from the database to predict the embedding length. Use the binary classifier to classify the capacity of message prediction. Here authors have used KNN [49], K* [50], LWL [51], Naive Bayes Classifier [52], Neural networks [53], Decision trees [54] and SVM Classifier [55] for classification. Performance accuracy that means the correctly classified instances has been measured through these classifiers. Through this process an unknown image can be classified and predict the secret hidden message length. Finally obtained the correctly classified instances for the corresponding stego and try to compare and contrast various embedding techniques in four wavelet domains using seven different classifiers.

IV. Algorithm of Proposed Method

INPUT: 1000 Images of dimension 512x512 has been taken where 70% used for training and evaluation purpose and 30% images for testing.

Step 1. Select an image

Step 2. Perform transformation in Haar [45], Daubechies [46], Symlets [47] and Biorthogonal [48] Wavelet Domain

Step 3. Extract Features

1. Central Moments (up-to 7th order)
2. Invariant Moments (up-to 7th order)
3. Zernike Moments (up-to 2nd order)
4. Standard Deviation
5. Spam features

Step 4. Embed message of variable length using Steganographic Methods: - S-Tool [56], PQ [57], Model Base [58], and F5 [59].

Step 5. Repeat Step 2 for the output of step 3.

Step 6. Continue step 3 and 4 until left with a new image.

Step 7. Store the results of step 2 and 4.

Step 8. Create Sample Dataset for the image results.

Step 9. Create training dataset for recognizing stego in variable length of message.

Step 10. Apply Classifier KNN [49], K* [50], LWL [51], Naive Bayes Classifier [52], Neural networks [53], Decision trees [54] and SVM Classifier [55] to measure performance using the correctly classified instances for various Stego and various message length. The correctly classified instances are taken as the performance of the Classifier against unknown images.

Step 11. End

TABLE I
PERFORMANCE (% OF CORRECTLY CLASSIFIED INSTANCES) MEASURE OF THE PROPOSED IMAGE STEGANALYZER FOR VARIOUS EMBEDDING TOOLS USING THE HAAR WAVELET DOMAIN

Steganography Tools	Embedding Rate	Ibk (K-nearest-neighbor)	Kstar (K* Classifier)	LWL (Locally weighted learning)	NaiveBayes (Naive Bayes classifier)	Neural networks (multi-layer perceptrons)	J48 (Decision trees)	SVM (Support vector machines)
S Tools	0.01	100	100	83	75	85	100	91
	0.03	100	100	83	75	85	100	91
	0.06	100	100	83	75	85	100	91
	0.09	100	100	83	75	85	100	91
	0.20	100	100	83	75	85	100	91
	0.50	100	97	83	74	85	100	91
	0.80	100	96	83	72	85	100	91
	1.00	100	96	83	70	85	100	91
PQ	0.01	100	48	100	100	100	100	48
	0.03	100	48	100	100	100	100	48
	0.06	100	48	100	100	100	100	48
	0.09	100	48	100	100	100	100	48
	0.20	100	100	100	100	100	100	48
	0.50	100	100	100	100	100	100	48
	0.80	100	100	100	100	100	100	48
	1.00	100	100	100	100	100	100	48
MB	0.01	100	48	100	100	100	100	51
	0.03	100	48	100	100	100	100	51
	0.06	100	48	100	100	100	100	51
	0.09	100	48	100	100	100	100	51
	0.20	100	100	100	100	100	100	51
	0.50	100	100	100	100	100	100	51
	0.80	100	100	100	100	100	100	51
	1.00	100	100	100	100	100	100	51
F5	0.01	73	56	80	76	70	78	43
	0.03	73	56	80	76	70	78	43
	0.06	73	56	80	76	70	78	43
	0.09	73	56	80	76	70	78	43
	0.20	73	56	80	76	70	78	43
	0.50	73	56	80	76	70	78	43
	0.80	75	56	78	77	71	80	43
	1.00	75	56	78	78	71	81	43

TABLE II
PERFORMANCE (% OF CORRECTLY CLASSIFIED INSTANCES) MEASURE OF THE PROPOSED IMAGE STEGANALYZER FOR VARIOUS EMBEDDING TOOLS USING THE DAUBECHIES WAVELET DOMAIN

Steganography Tools	Embedding Rate	Ibk (K-nearest-neighbor)	Kstar (K* Classifier)	LWL (Locally weighted learning)	NaiveBayes (Naive Bayes classifier)	Neural networks (multi-layer perceptrons)	J48 (Decision trees)	SVM (Support vector machines)
S Tools	0.01	99	100	83	75	85	100	91
	0.03	99	100	83	75	85	100	91
	0.06	100	100	83	75	85	100	91
	0.09	100	100	83	75	85	100	91
	0.20	100	100	83	75	85	100	91
	0.50	100	97	83	74	85	100	91
	0.80	100	96	82	72	85	100	91
	1.00	100	96	82	70	85	100	91
PQ	0.01	100	48	96	98	91	96	48
	0.03	96	48	96	98	91	96	48
	0.06	96	48	96	98	91	96	48
	0.09	96	48	96	98	91	96	48
	0.20	100	48	100	100	100	100	48
	0.50	100	48	100	100	100	100	48
	0.80	100	48	100	100	100	100	48
	1.00	100	48	100	100	100	100	48
MB	0.01	100	100	96	98	91	96	51
	0.03	96	100	96	98	91	96	51
	0.06	96	100	96	98	91	96	51
	0.09	96	100	96	98	91	96	51
	0.20	100	100	100	100	100	100	51
	0.50	100	48	100	100	100	100	51
	0.80	100	48	100	100	100	100	51
	1.00	100	48	100	100	100	100	51
F5	0.01	73	56	80	76	70	78	43
	0.03	73	56	80	76	70	78	43
	0.06	73	56	80	76	70	78	43
	0.09	73	56	80	76	70	78	43
	0.20	73	56	80	76	70	78	43
	0.50	73	56	80	76	70	78	43
	0.80	75	56	78	77	71	80	43
	1.00	75	56	78	78	71	81	43

TABLE III
PERFORMANCE (% OF CORRECTLY CLASSIFIED INSTANCES) MEASURE OF THE PROPOSED IMAGE STEGANALYZER FOR VARIOUS EMBEDDING TOOLS USING THE SYMLETS WAVELET DOMAIN

Steganography Tools	Embedding Rate	Ibk (K-nearest-neighbor)	Kstar (K* Classifier)	LWL (Locally weighted learning)	NaiveBayes (Naive Bayes classifier)	Neural networks (multi-layer perceptrons)	J48 (Decision trees)	SVM (Support vector machines)
S Tools	0.01	100	100	83	75	85	100	91
	0.03	100	100	83	75	85	100	91
	0.06	100	100	83	75	85	100	91
	0.09	100	100	83	75	85	100	91
	0.20	100	100	83	75	85	100	91
	0.50	100	97	83	74	85	100	91
	0.80	100	96	83	72	85	100	91
	1.00	100	96	83	70	85	100	91
PQ	0.01	100	48	100	100	100	100	48
	0.03	100	48	100	100	100	100	48
	0.06	100	48	100	100	100	100	48
	0.09	100	48	100	100	100	100	48
	0.20	100	48	100	100	100	100	48
	0.50	100	48	100	100	100	100	48
	0.80	100	48	100	100	100	100	48
	1.00	100	48	100	100	100	100	48
MB	0.01	100	100	100	100	100	100	51
	0.03	100	100	100	100	100	100	51
	0.06	100	100	100	100	100	100	51
	0.09	100	100	100	100	100	100	51
	0.20	100	100	100	100	100	100	51
	0.50	100	48	100	100	100	100	51
	0.80	100	48	100	100	100	100	51
	1.00	100	48	100	100	100	100	51
F5	0.01	73	56	80	76	70	78	43
	0.03	73	56	80	76	70	78	43
	0.06	73	56	80	76	70	78	43
	0.09	73	56	80	76	70	78	43
	0.20	73	56	80	76	70	78	43
	0.50	73	56	80	76	70	78	43
	0.80	75	56	78	77	71	80	43
	1.00	75	56	78	78	71	81	43

TABLE IV
PERFORMANCE (% OF CORRECTLY CLASSIFIED INSTANCES) MEASURE OF THE PROPOSED IMAGE STEGANALYZER FOR VARIOUS EMBEDDING TOOLS USING THE BIORTHOGONAL WAVELET DOMAIN

Steganography Tools	Embedding Rate	Ibk (K-nearest-neighbor)	Kstar (K* Classifier)	LWL (Locally weighted learning)	NaiveBayes (Naive Bayes classifier)	Neural networks (multi-layer perceptrons)	J48 (Decision trees)	SVM (Support vector machines)
S Tools	0.01	100	100	83	75	85	100	91
	0.03	100	100	83	75	85	100	91
	0.06	100	100	83	75	85	100	91
	0.09	100	100	83	75	85	100	91
	0.20	100	100	83	75	85	100	91
	0.50	100	97	83	74	85	100	91
	0.80	100	96	83	72	85	100	91
	1.00	100	96	83	70	85	100	91
PQ	0.01	100	48	100	100	100	100	48
	0.03	100	48	100	100	100	100	48
	0.06	100	48	100	100	100	100	48
	0.09	100	48	100	100	100	100	48
	0.20	100	48	100	100	100	100	48
	0.50	100	48	100	100	100	100	48
	0.80	100	48	100	100	100	100	48
	1.00	100	48	100	100	100	100	48
MB	0.01	100	100	100	100	100	100	51
	0.03	100	100	100	100	100	100	51
	0.06	100	100	100	100	100	100	51
	0.09	100	100	100	100	100	100	51
	0.20	100	100	100	100	100	100	51
	0.50	100	100	100	100	100	100	51
	0.80	100	48	100	100	100	100	51
	1.00	100	48	100	100	100	100	51
F5	0.01	73	56	80	76	70	78	43
	0.03	73	56	80	76	70	78	43
	0.06	73	56	80	76	70	78	43
	0.09	73	56	80	76	70	78	43
	0.20	73	56	80	76	70	78	43
	0.50	73	56	80	76	70	78	43
	0.80	75	56	78	77	71	80	43
	1.00	75	56	78	78	71	81	43

TABLE V
COMPARISON OF PROPOSED NOVEL IMAGE STEGANALYZER WITH OTHER EXISTING METHODS IN HAAR WAVELET DOMAIN

Software	Emb. Rate	BSM	FBS	WBS	Ibk (K-nearest-neighbor)	Kstar (K* Classifier)	LWL (Locally weighted learning)	NaiveBayes (Naive Bayes classifier)	Neural networks (multi-layer perceptrons)	J48 (Decision trees)	SVM (Support vector machines)
S Tools	0.05	68	97	56	100	100	83	75	85	100	91
	0.10	78	99	65	100	100	83	75	85	100	91
	0.20	87	99	75	100	100	83	75	85	100	91
	0.40	92	99	87	100	97	83	74	85	100	91
	0.60	93	99	91	100	96	83	72	85	100	91
PQ	0.05	75	85	76	100	48	100	100	100	100	48
	0.10	75	85	76	100a	48	100	100	100	100	48
	0.20	75	85	75	100	100	100	100	100	100	48
	0.40	76	86	79	100	100	100	100	100	100	48
	0.60	NA	NA	NA	100	100	100	100	100	100	48
MB	0.05	75	85	76	100	48	100	100	100	100	51
	0.10	75	85	76	100	48	100	100	100	100	51
	0.20	75	85	75	100	100	100	100	100	100	51
	0.40	76	86	79	100	100	100	100	100	100	51
	0.60	NA	NA	NA	100	100	100	100	100	100	51
F5	0.05	51	71	51	73	56	80	76	70	78	43
	0.10	51	77	52	73	56	80	76	70	78	43
	0.20	51	85	55	73	56	80	76	70	78	43
	0.40	53	93	61	73	56	80	76	70	78	43
	0.60	NA	NA	NA	75	56	78	77	71	80	43

TABLE VI
COMPARISON OF PROPOSED NOVEL IMAGE STEGANALYZER WITH OTHER EXISTING METHODS IN DAUBECHIES WAVELET DOMAIN

Software	Emb. Rate	BSM	FBS	WBS	Ibk (K-nearest-neighbor)	Kstar (K* Classifier)	LWL (Locally weighted learning)	NaiveBayes (Naive Bayes classifier)	Neural networks (multi-layer perceptrons)	J48 (Decision trees)	SVM (Support vector machines)
S Tools	0.05	68	97	56	100	100	83	75	85	100	91
	0.10	78	99	65	100	100	83	75	85	100	91
	0.20	87	99	75	100	100	83	75	85	100	91
	0.40	92	99	87	100	97	83	74	85	100	91
	0.60	93	99	91	100	96	82	72	85	100	91
PQ	0.05	75	85	76	96	48	96	98	91	96	48
	0.10	75	85	76	96	48	96	98	91	96	48
	0.20	75	85	75	100	48	100	100	100	100	48
	0.40	76	86	79	100	48	100	100	100	100	48
	0.60	NA	NA	NA	100	48	100	100	100	100	48
MB	0.05	75	85	76	96	100	96	98	91	96	51
	0.10	75	85	76	96	100	96	98	91	96	51
	0.20	75	85	75	100	100	100	100	100	100	51
	0.40	76	86	79	100	48	100	100	100	100	51
	0.60	NA	NA	NA	100	48	100	100	100	100	51
F5	0.05	51	71	51	73	56	80	76	70	78	43
	0.10	51	77	52	73	56	80	76	70	78	43
	0.20	51	85	55	73	56	80	76	70	78	43
	0.40	53	93	61	73	56	80	76	70	78	43
	0.60	NA	NA	NA	75	56	78	77	71	80	43

TABLE VII
COMPARISON OF PROPOSED NOVEL IMAGE STEGANALYZER WITH OTHER EXISTING METHODS IN SYMLETS WAVELET DOMAIN

Software	Emb. Rate	BSM	FBS	WBS	Ibk (K-nearest-neighbor)	Kstar (K* Classifier)	LWL (Locally weighted learning)	NaiveBayes (Naive Bayes classifier)	Neural networks (multi-layer perceptrons)	J48 (Decision trees)	SVM (Support vector machines)
S Tools	0.05	68	97	56	100	100	83	75	85	100	91
	0.10	78	99	65	100	100	83	75	85	100	91
	0.20	87	99	75	100	100	83	75	85	100	91
	0.40	92	99	87	100	97	83	74	85	100	91
	0.60	93	99	91	100	96	83	72	85	100	91
PQ	0.05	75	85	76	100	48	100	100	100	100	48
	0.10	75	85	76	100	48	100	100	100	100	48
	0.20	75	85	75	100	48	100	100	100	100	48
	0.40	76	86	79	100	48	100	100	100	100	48
	0.60	NA	NA	NA	100	48	100	100	100	100	48
MB	0.05	75	85	76	100	100	100	100	100	100	51
	0.10	75	85	76	100	100	100	100	100	100	51
	0.20	75	85	75	100	100	100	100	100	100	51
	0.40	76	86	79	100	48	100	100	100	100	51
	0.60	NA	NA	NA	100	48	100	100	100	100	51
F5	0.05	51	71	51	73	56	80	76	70	78	43
	0.10	51	77	52	73	56	80	76	70	78	43
	0.20	51	85	55	73	56	80	76	70	78	43
	0.40	53	93	61	73	56	80	76	70	78	43
	0.60	NA	NA	NA	75	56	78	77	71	80	43

TABLE VIII
COMPARISON OF PROPOSED NOVEL IMAGE STEGANALYZER WITH OTHER EXISTING METHODS IN BIORTHOGONAL WAVELET DOMAIN

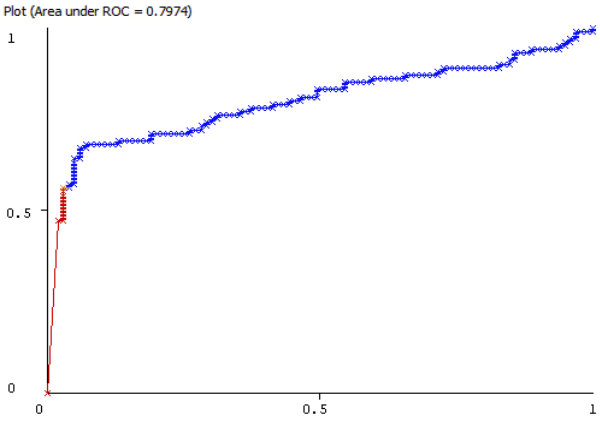
Software	Emb. Rate	BSM	FBS	WBS	Ibk (K-nearest-neighbor)	Kstar (K* Classifier)	LWL (Locally weighted learning)	NaiveBayes (Naive Bayes classifier)	Neural networks (multi-layer perceptrons)	J48 (Decision trees)	SVM (Support vector machines)
S Tools	0.05	68	97	56	100	100	83	75	85	100	91
	0.10	78	99	65	100	100	83	75	85	100	91
	0.20	87	99	75	100	100	83	75	85	100	91
	0.40	92	99	87	100	97	83	74	85	100	91
	0.60	93	99	91	100	96	83	72	85	100	91
PQ	0.05	75	85	76	100	48	100	100	100	100	48
	0.10	75	85	76	100	48	100	100	100	100	48
	0.20	75	85	75	100	48	100	100	100	100	48
	0.40	76	86	79	100	48	100	100	100	100	48
	0.60	NA	NA	NA	100	48	100	100	100	100	48
MB	0.05	75	85	76	100	100	100	100	100	100	51
	0.10	75	85	76	100	100	100	100	100	100	51
	0.20	75	85	75	100	100	100	100	100	100	51
	0.40	76	86	79	100	100	100	100	100	100	51
	0.60	NA	NA	NA	100	48	100	100	100	100	51
F5	0.05	51	71	51	73	56	80	76	70	78	43
	0.10	51	77	52	73	56	80	76	70	78	43
	0.20	51	85	55	73	56	80	76	70	78	43
	0.40	53	93	61	73	56	80	76	70	78	43
	0.60	NA	NA	NA	75	56	78	77	71	80	43

V. EXPERIMENTAL RESULTS OF PROPOSED METHOD

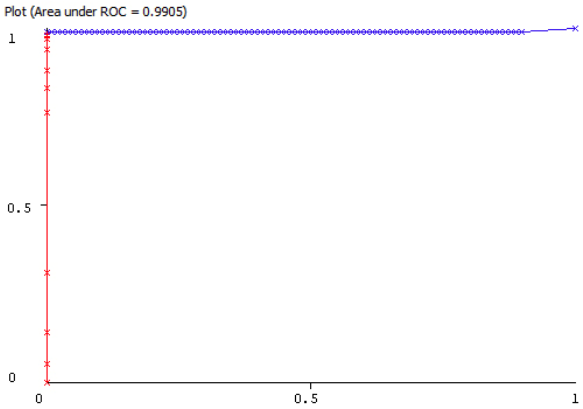
The Steganalyzer has been designed using a training set that is obtained by the application of different embedding tools like S-Tool [56], PQ [57], Model Base [58], and F5 [59] etc. in four different wavelet domain namely Haar, Daubechies, Symlets and Biorthogonal. In the experimental set up 70% images were used for training while 30% images were used for testing. The experiments were performed on a large data set of 1000 images obtained from publicly available websites. The image data set is categorized with respect to different features of the image to determine their potential impact on steganalysis performance. After embedding secret message into the cover image with the various embedding rates of 0.01, 0.03, 0.06, 0.09, 0.20, 0.50, 0.80, 1.00 in all four of the wavelet domains various stego images has been created. From the tables, it can be observed that with the introduction of even a small amount of hidden data the values of statistical as well as similarity based features of the images change. This change is different for different wavelet domains that have been employed. In the conclusion we try to compare and contrast the different wavelet domains that have been used. In order to calculate the performance of seven classifiers at different embedding rate and to show the relationship between the false-positive rate and the detection rate of steganographic data embedding for the four different embedding methods. The correctly classified instances performance accuracy are calculated for the classifiers by first designing a classifier and then testing the data unseen to the classifier against the trained classifier. The testing accuracy is calculated by the steganalysis algorithm. Above tables respectively shows the Performance accuracy of KNN [49], K* [50], LWL [51], Naive Bayes Classifier [52], Neural networks [53], Decision trees [54] and SVM Classifier [55] at different embedding rate for S-Tool [56], PQ [57], Model Base [58], and F5 [59]. Performance comparison of different classifier at various embedding rate has also been shown. A comparative study amongst various existing image steganalysis method with proposed novel steganalysis method has been shown in Tables V-VIII. Table IX describes the performance analysis of the Classifiers used in this steganalysis procedure along with the others methods like BSM, FBS and WBS. Fig. 4 shows some of the ROC of proposed work.

TABLE IX
PERFORMANCE COMPARISON OF PROPOSED NOVEL IMAGE STEGANALYZER WITH OTHER EXISTING METHODS

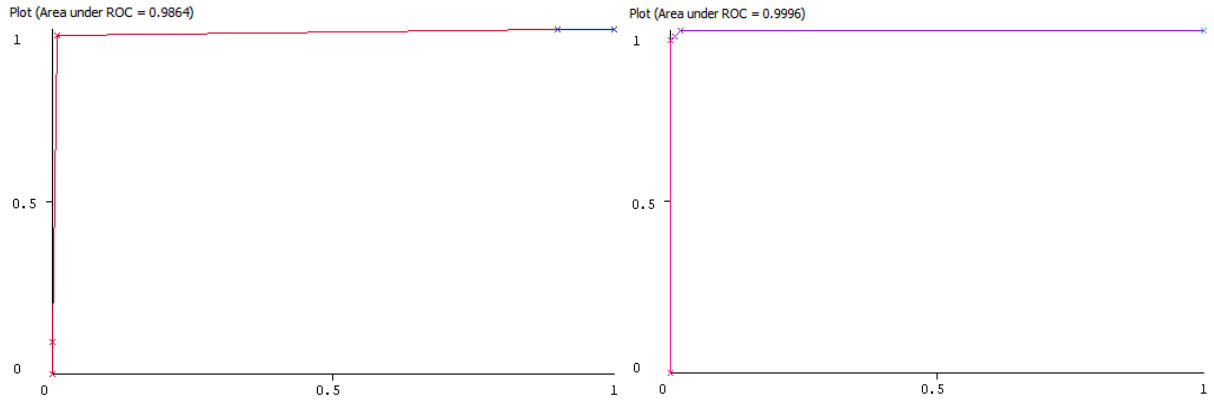
Wavelet Domain	Soft ware	Very Good (90% - 100%)	Good (70% - 89%)	Moderate (50% - 69%)
Haar Wavelet Domain	S Tools	Ibk (K-nearest-neighbor), Kstar (K* Classifier), J48 (Decision trees), **FBS**	LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), SVM (Support vector machines), **BSM, WBS**	
	PQ	Ibk (K-nearest-neighbor), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees)	**BSM, FBS, WBS**	Kstar (K* Classifier)
	MB	Ibk (K-nearest-neighbor), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees)	**BSM, FBS, WBS**	
	F5		Ibk (K-nearest-neighbor), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees)	Kstar (K* Classifier), **BSM, WBS**
Daubechies Wavelet Domain	S Tools	Ibk (K-nearest-neighbor), Kstar (K* Classifier), J48 (Decision trees), SVM (Support vector machines), **FBS**	LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), **BSM, WBS**	
	PQ	Ibk (K-nearest-neighbor), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees)	**BSM, FBS, WBS**	
	MB	Ibk (K-nearest-neighbor), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees)	Kstar (K* Classifier), **BSM, FBS, WBS**	
	F5		Ibk (K-nearest-neighbor), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees), **FBS**	
Symlets Wavelet Domain	S Tools	Ibk (K-nearest-neighbor), Kstar (K* Classifier), J48 (Decision trees), SVM (Support vector machines), **FBS**	LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), **BSM, WBS**	
	PQ	Ibk (K-nearest-neighbor), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees)	**BSM, FBS, WBS**	
	MB	Ibk (K-nearest-neighbor), Kstar (K* Classifier), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees)	**BSM, FBS, WBS**	
	F5		Ibk (K-nearest-neighbor), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees), **FBS**	**WBS**
Biorthogonal Wavelet Domain	S Tools	Ibk (K-nearest-neighbor), Kstar (K* Classifier), J48 (Decision trees), SVM (Support vector machines), **FBS**	LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), **BSM, WBS**	Kstar (K* Classifier), SVM (Support vector machines)
	PQ	Ibk (K-nearest-neighbor), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees)	**BSM, FBS, WBS**	SVM (Support vector machines)
	MB	Ibk (K-nearest-neighbor), Kstar (K* Classifier), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees)	**BSM, FBS, WBS**	
	F5		Ibk (K-nearest-neighbor), LWL (Locally weighted learning), NaiveBayes (Naive Bayes classifier), Neural networks (multi-layer perceptrons), J48 (Decision trees), **FBS**	Kstar (K* Classifier), **BSM, WBS**



(a)



(b)



(c) (d)

Fig 4 ROC of Proposed Image Steganalyzer (1% of embedding capacity has been used here) (a) In Biorthogonal Wavelet Domain using F5 Steganography tool and NaiveBayes classifier. (b) In Daubechies Wavelet Domain using PQ Steganography tool and locally weighted learning classifier. (c) In Haar Wavelet Domain using PQ Steganography tool and K* classifier. (d) In Biorthogonal Wavelet Domain using S-Tools Steganography tool and Decision trees classifier

VI. Conclusion

In this paper a novel feature based image steganalysis technique in wavelet domain is proposed and tested which has been designed based on moments and spam features based image distortion measurement. The de-noised version of the image object is taken as a measure of the cover image. Next step is to use statistical, invariant and other similarity based features to measure the distortion which in turn is used to design a classifier to determine the presence of hidden information in an image. The design of the image steganalyzer using seven different classifiers in each of the four wavelet domains is useful as it comes in handy to detect the presence of even small amount of data. Comparison of the results that we have obtained in our report with other existing wavelet based steganalysis (WBS) technique goes on to show that the proposed image steganalysis technique shows much higher detection rate than existing steganalyzers operating in the same domain. In the concluding part a table has been provided that compares the proposed image steganalysis technique in wavelet domain with other existing image steganalysis techniques namely Binary Similarity Based Image Steganalysis Technique-BSM, Feature Based Image Steganalysis Technique – FBS and Wavelet Based Steganalysis Technique – WBS. Increases of accuracy level along with the detection power are the aim of our future extension.

References

[1] Abbas Cheddad, Joan Condell, Kevin Curran, Paul Mc Kevitt, Digital image steganography: Survey and analysis of current methods Signal Processing 90, 2010,pp. 727– 752.

[2] Indradip Banerjee, Souvik Bhattacharyya, Gautam Sanyal, "Robust image steganography with pixel factor mapping (PFM) technique", Computing for Sustainable Global Development (INDIACom), 2014 International Conference on 5-7 March 2014. Page(s): 692 - 698, Print ISBN: 978-93-80544-10-6. Publisher: IEEE Xplore Digital Library.

[3] Jim Bartel. "Steganalysis: An Overview" Security Essentials Bootcamp Style (Security 401). Global Information Assurance Certification Paper.

[4] Avcıbaş, İ., Memon N., Sankur B., "Steganalysis using image quality metrics", IEEE Trans. on Image Process., January 2003.

[5] Gojan, J., M. Goljan, R. Du, "Reliable detection of LSB steganography in color and grayscale images", Proc., of the ACM Workshop on Mult. And Secur., Ottawa, CA, pp. 27-30, October 5, 2001.

[6] Johnson, N.F., S. Jajodia, "Steganalysis of images created using current steganography software", in David Aucsmith (Ed.): Information Hiding, LNCS 1525, pp. 32-47. Springer-Verlag Berlin Heidelberg, 1998.

[7] Westfeld, A. Pfitzmann, "Attacks on steganographic systems", in Information Hiding, LNCS 1768, pp. 61-66, Springer-Verlag Heidelberg, 1999.

[8] Philip Bateman, Hans Georg Schaathun. "Image Steganography and Steganalysis" Thesis for the Degree of Master of Science in Security Technologies & Applications at Department of Computing, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey, United Kingdom.4th August 2008.

[9] A. Westfeld, A. P¨tzmann. Attacks on steganographic systems - breaking the steganographic utilities ezstego, jsteg, steganos, and s-tools-and some lessons learned. In Proceedings of the 3rd Information Hiding Workshop, volume 1768 of LNCS, pages 61-76. Springer, 1999.

[10] Niels Provos, Peter Honeyman. Detecting steganographic content on the internet. In Proceedings of NDSS'02: Network and Distributed System Security Symposium, pages 1-13. Internet Society, 2002.

[11] Jessica Fridrich, Miroslav Goljan, Rui Du. Reliable detection of lsb steganography in color and grayscale images. In Proceedings of 2001 ACM workshop on Multimedia and security: new challenges, pages 27-30. ACM Press, 2001.

[12] S. Dumitrescu, X. L. Wu, Z. Wang. Detection of lsb steganography via sample pair analysis. IEEE Transactions on Signal Processing, 51(7):1995-2007, 2003.

[13] J. Fridrich, M. Goljan. On estimation of secret message length in lsb steganography in spatial domain. In IS&T/SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents VI, volume 5306, pages 23-34. SPIE, 2004.

[14] D. Ker. Fourth-order structural steganalysis and analysis of cover assumptions. In IS&T/SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents VIII, volume 6072, pages 1-14. SPIE, 2006.

[15] A. D. Ker. A general framework for the structural steganalysis of lsb replacement. In Proceedings of the 7th Information Hiding Workshop, volume 3727 of LNCS, pages 296-311. Springer, 2005.

[16] J. Harmsen, W. Pearlman. Steganalysis of additive noise modelable information hiding. In IS&T/SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents V, volume 5020, pages 131-142. SPIE, 2003.

[17] Jinggang Huang, David Mumford. Statistics of natural images and models. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 541-547, 1999.

[18] J. H. He, J. W. Huang, G. P. Qiu. A new approach to estimating hidden message length in stochastic modulation steganography. In Proceedings of the 4th Internation Workshop on Digital Watermarking, volume 3710 of LNCS, pages 1-14. Springer, 2005.
[19] J. H. He, J. W. Huang. Steganalysis of stochastic modulation steganography. Science in China Series: F-Information Sciences, 49(3):273-285, 2006.
[20] M. Niimi, R. O. Eason, H. Noda, E. Kawaguchi. Intensity histogram steganalysis in bpcs steganography. In IS&T/SPIE Electronic Imaging: Security and Watermarking of Multimedia Contents III, volume 4314, pages 555-564. SPIE, 2001.
[21] X. P. Zhang, S. Z. Wang. Vulnerability of pixel-value di®erencing steganography to histogram analysis and modi¯cation for enhanced security. Pattern Recognition Letters, 25(3):331-339, 2004.
[22] Bin Li, Yanmei Fang, Jiwu Huang. Steganalysis of multiple-base notational system steganography. IEEE Signal Processing Letters, 15:493-496, 2008.
[23] K. Sullivan, Z. Bi, U. Madhow, S. Chandrasekaran, B. S. Manjunath. Steganalysis of quantization index modulation data hiding. In Proceedings of 2004 IEEE International Conference on Image Processing, volume 2, pages 1165-1168, 2004.
[24] J. Fridrich, M. Goljan, D. Hogea. Steganalysis of jpeg images: Breaking the f5 algorithm. In Proceedings of the 5th Information Hiding Workshop, volume 2578 of LNCS, pages 310-323. Springer, 2002.
[25] J. Fridrich, M. Goljan, D. Hogea. Attacking the outguess. In Proceedings of 2002 ACM Workshop on Multimedia and Security, pages 3-6. ACM Press, 2002.
[26] R. BÄohme, A. Westfeld. Breaking cauchy model-based jpeg steganography with first order statistics. In Proceedings of the 9th European Symposium On Research in Computer Security, volume 3193 of LNCS, pages 125-140. Springer, 2004.
[27] Bin Li, Yun Q. Shi, Jiwu Huang. Steganalysis of yass. In Proceedings of the 10th ACM workshop on Multimedia and security (MM&Sec'08), pages 139-148. ACM Press, 2008.
[28] X. Y. Luo, D. S.Wang, P.Wang, F. L. Liu. A review on blind detection for image steganography. Signal Processing, 88(9):2138-2157, 2008.
[29] I. Avcibas, N. Memon, B. Sankur. Steganalysis using image quality metrics. IEEE Transactions on Image Processing, 12(2):221-229, 2003.
[30] J. Fridrich. Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes. In Proceedings of the 6th Information Hiding Workshop, volume 3200 of LNCS, pages 67-81. Springer, 2004.
[31] Y. Q. Shi, C. Chen, W. Chen. A morkov process based approach to effective attacking jpeg steganography. In Proceedings of the 8th Information Hiding Workshop, volume 4437 of LNCS, pages 249-264. Springer, 2006.
[32] Tomas Pevny, Jessica Fridrich. Merging markov and dct features for multi-class jpeg steganalysis. In Proceedings of SPIE: Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, volume 6505, pages 3-14. SPIE, 2007.
[33] Lyu Siwei, H. Farid. Detecting hidden message using higher-order statistics and support vector machines. In Proceedings of the 5th Information Hiding Workshop, volume 2578 of LNCS, pages 131-142. Springer, 2002.
[34] K. Sullivan, U. Madhow, S. Chandrasekaran, B. S. Manjunath. Steganalysis for markov cover data with applications to images. IEEE Transactions on Information Forensics and Security, 1(2):275-287, 2006.
[35] Zhuo Li, Kuijun Lu, Xianting Zeng, Xuezeng Pan. "A Blind Steganalytic Scheme Based on DCT and Spatial Domain for JPEG Images", Journal of Multimedia, p.200-207, VOL. 5, NO. 3, JUNE 2010.
[36] X. Y. Luo, D. S.Wang, P.Wang, F. L. Liu. A review on blind detection for image steganography. Signal Processing, 88(9):2138-2157, 2008.
[37] McLachlan, Geoffrey J. Discriminant analysis and statistical pattern recognition. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1992. xvi+526 pp. ISBN: 0-471-61531-5
[38] Corinna Cortes, Vladimir Vapnik."Support-Vector Networks". Machine Leaming, 20, 273-297 (1995) @ 1995 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.
[39] Bremner D, Demaine E, Erickson J, Iacono J, Langerman S, Morin P, Toussaint G (2005). "Output-sensitive algorithms for computing nearest-neighbor decision boundaries". Discrete and Computational Geometry 33 (4): 593–604.
[40] Bhadeshia H. K. D. H. (1999). "Neural Networks in Materials Science". ISIJ International 39 (10): 966–979.
[41] M. K. Hu, "Visual pattern recognition by moment invariants," IRE Trans. Information Theory, vol. 8, pp. 179–187, 1962.
[42] Hongli Tian, Huiqiang Yan, Hongdong Zhao. "A Fast and Accurate Approach to the Computation of Zernike Moments" Applied Informatics and Communication Communications in Computer and Information Science. Volume 228, 2011, pp 46-53
[43] Paul S. Dwyer. "The mean and standard deviation of the distribution of group assembly sums". Psychometrika December 1964, Volume 29, Issue 4, pp 397-408
[44] T. Pevny, P. Bas, J. Fridrich Steganalysis by subtractive pixel adjacency matrix. Steganalysis by subtractive pixel adjacency matrix, Princeton, NJ, September 7-8, 2009.
[45] Alfréd Haar, "Zur Theorie der orthogonalen Funktionensysteme", Mathematische Annalen 69 (3): 331–371, (1910), doi: 10.1007/BF01456326.
[46] Daubechies, Ingrid. Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics (1992).
[47] Mahesh S. Chavan, Nikos Mastorakis, Manjusha N. Chavan, M.S. Gaikwad. "Implementation of SYMLET Wavelets to Removal of Gaussian Additive Noise from Speech Signal" Recent Researches in Communications, Automation, Signal Processing, Nanotechnology, Astronomy and Nuclear Physics. P 37-41.
[48] O. Prakash, R. Srivastava; A. Khare. "Biorthogonal wavelet transform based image fusion using absolute maximum fusion rule" IEEE Conference on Information & Communication Technologies (ICT), 2013.
[49] V. Suresh Babu, P. Viswanath. Rough-fuzzy weighted k-nearest leader classifier for large data sets. Pattern Recognition, 42(2009):1719–1731, 2009.
[50] John G. Cleary, Leonard E. Trigg: K*: An Instance-based Learner Using an Entropic Distance Measure. In: 12th International Conference on Machine Learning, 108-114, 1995.
[51] Christopher G. Atkeson, Andrew W. Moore, Stefan Schaal, "LocallyWeighted Learning" Artificial Intelligence Review 11: 11–73, 1997. Kluwer Academic Publishers. Printed in the Netherlands.
[52] Caruana, R.; Niculescu-Mizil, A. (2006). "An empirical comparison of supervised learning algorithms". Proceedings of the 23rd international conference on Machine learning. CiteSeerX: 10.1.1.122.5901.
[53] Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961.
[54] Deng,H.; Runger, G.; Tuv, E. (2011). "Bias of importance measures for multi-valued attributes and solutions". Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN).
[55] P. Pudil, J. Novovicova, J. Kittler, "Floating Search Methods in Feature Selection," Pattern Recognition Letters,vol. 15, no. 11, pp. 1119 - 1125, November 1994.
[56] A.Brown,S-ToolsVersion4.0,Copyright©1996, http://members.tripod.com/steganography/stego/s-tools4
[57] J. Fridrich, M. Goljan, D. Soukal, Perturbed quantization steganography with wet paper codes," ACM Multimedia Workshop, Magdeburg, Germany, September 20-21, 2004.
[58] P. Sallee, Model-based steganography," International Workshop on Digital Watermarking, Seoul, Korea.2003
[59] Wetfeld, F5a steganographic algorithm: High capacity despite better steganalysis," 4th International Workshop on Information Hiding. , 2001.

Indradip Banerjee is a Research Scholar at National Institute of Technology, Durgapur, West Bengal, India. He received his MCA degree from IGNOU in 2009, PGDCA from IGNOU in 2008, MMM from Annamalai University in 2005 and BCA (Hons.) from The University of Burdwan in 2003. He is registered and pursuing his PhD. in Engineering at Computer Science and Engineering Department, National Institute of Technology, Durgapur, West Bengal, India. His areas of interest are Biometric Information Security, Steganography, Cryptography, Text Steganography, Image Steganography, Quantum Steganography and Steganalysis. He has published 24 research papers in International and National Journals / Conferences.

Souvik Bhattacharyya received his B.E. degree in Computer Science and Technology from B.E. College, Shibpur, India and M.Tech degree in Computer Science and Engineering from National Institute of Technology, Durgapur, India. He has received Ph.D (Engg.) from National Institute of Technology, Durgapur, India. Currently he is working as an Assistant Professor and In-Charge in Computer Science and Engineering Department at University Institute of Technology, The University of Burdwan. His areas of interest are Natural Language Processing, Network Security and Image Processing. He has published nearly 56 papers in International and National Journals / Conferences.

Gautam Sanyal has received his B.E and M.Tech degree National Institute of Technology (NIT), Durgapur, India. He has received Ph.D (Engg.) from Jadavpur University, Kolkata, India, in the area of Robot Vision. He possesses an experience of more than 25 years in the field of teaching and research. He has published nearly 150 papers in International and National Journals / Conferences. Two Ph.Ds (Engg) have already been awarded under his guidance. At present he is guiding six Ph.Ds scholars in the field of Steganography, Cellular Network, High Performance Computing and Computer Vision. He has guided over 10 PG and 100 UG thesis. His research interests include Natural Language Processing, Stochastic modeling of network traffic, High Performance Computing, Computer Vision. He is presently working as a Professor in the department of Computer Science and Engineering and also holding the post of Dean (Students' Welfare) at National Institute of Technology, Durgapur, India.