World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:9, 2008

Support Vector Machine based Intelligent Watermark Decoding for Anticipated Attack

Syed Fahad Tahir, Asifullah Khan, Abdul Majid, and Anwar M. Mirza

***Abstract*—**In this paper, we present an innovative scheme of blindly extracting message bits from an image distorted by an attack. *Support Vector Machine (SVM)* is used to nonlinearly classify the bits of the embedded message. Traditionally, a hard decoder is used with the assumption that the underlying modeling of the Discrete Cosine Transform (DCT) coefficients does not appreciably change. In case of an attack, the distribution of the image coefficients is heavily altered. The distribution of the sufficient statistics at the receiving end corresponding to the antipodal signals overlap and a simple hard decoder fails to classify them properly. We are considering message retrieval of antipodal signal as a binary classification problem. Machine learning techniques like SVM is used to retrieve the message, when certain specific class of attacks is most probable. In order to validate SVM based decoding scheme, we have taken Gaussian noise as a test case. We generate a data set using 125 images and 25 different keys. Polynomial kernel of SVM has achieved 100 percent accuracy on test data.

***Keywords*—**Bit Correct Ratio (BCR), Grid Search, Intelligent Decoding, Jackknife Technique, Support Vector Machine (SVM), Watermarking.

I. Introduction

WATERMARKING is the method of invisibly altering data to embed a message. It is an effective way to counter problems like unauthorized handling, copying and reuse of information [1]. Digital watermarking is performed upon a range of digital media, like text, audio, images, movies and 3D models. Applications of watermarking include ownership assertion, authentication, broadcast monitoring, and integrity control [2]. Generally, a watermarking scheme is designed in view of its application. Watermarking demands a fine balance between two of its most important requirements: robustness and imperceptibility. There is no such generic watermarking scheme that can resist all sorts of attacks. But practically it is not required either, as different applications of watermarking might need to consider only a specific set of conceivable attacks. In this work Gaussian noise attack is taken as a test case.

Recently, machine learning techniques have been applied in watermark detection and decoding [3]-[5]. However, these techniques do not consider the hostile situation, where a watermarked image can be distorted severely. In this work, we are considering message retrieval as a binary classification problem. SVM based learning techniques are thus used to retrieve an embedded message, when certain specific class of unintentional attacks are most probable. The effectiveness of the developed intelligent models is restricted to the conceivable attack scenario.

Work related to intelligent decoding and watermarking is discussed in Section II. Section III describes our proposed methodology. Results and discussion are presented in Section IV. At the end, conclusion and future work is discussed.

II. Related Theory and Research Work

Intelligent systems are being used in several different fields. In watermarking, it is being applied successfully at different watermark stages such as embedding, detection and decoding. Recently, Zhang et al [8], in their work have used Independent Component Analysis to extract watermark correctly without using the original image. Accuracy of the watermarking extraction depends on the key and the statistical independence between the original image and the watermark. Kırbız et al [9], have introduced an audio watermark decoding scheme. This scheme performs SVM based supervised learning followed by a blind decoding. Venkataramani et al [9], have illustrated the application of SVMs as discriminative models for the refined search spaces. They have shown that SVMs can be used for continuous speech recognition. A. Khan et al [11] in their work have used the idea of perceptually shaping the watermark with respect to both the conceivable attack and cover image at the embedding stage using Genetic Programming. Our proposed method is closely related to the work of A. Khan [6]. He has used an idea of automatically modifying the decoder structure in accordance to the cover image and conceivable attack using Genetic Programming. Information pertaining to watermarked cover coefficients and conceivable attack is utilized.

A watermarked data can be attacked in different ways. However, each application usually has to deal with a particular set of distortions. Cox et al. [6] and Barni et al. [2] have described some of the attacks as well as their countermeasures. Some of the attacks are addition of Gaussian and Non Gaussian noise, signal processing attacks like D/A conversion, color reduction, linear filtering attacks like high pass and low pass filtering, lossy compression, geometric

Syed Fahad Tahir is with Faculty of Computer Sciences & Engineering, Ghulam Ishaq Khan (GIK) Institute of Engineering Science & Technology, Topi, Pakistan (e-mail: fahad_290@yahoo.com).

Asifullah Khan is with Department of Information and Computer Sciences, Pakistan Institute of Engineering and Applied Sciences, Nilore, Islamabad, Pakistan (e-mail: asif@pieas.edu.pk, abdulmajiid@gmail.com).

Anwar M. Mirza is with Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan (e-mail: anwar.m.mirza@gmail.com).

International Science Index, Computer and Information Engineering Vol:2, No:9, 2008 waset.org/Publication/13275

distortions etc. Keeping in view the expected distortions, different approaches like redundant embedding, selection of perceptually significant coefficients, spread spectrum modulation, and inverting distortion in the detection phase are investigated to make a watermark system reliable.

A. Support Vector Machine

SVM has emerged in recent years as a popular approach to the classification of data. SVM is margin-based classifier with good generalization capabilities. It is the method of creating functions from a set of labeled training data. The function can be either a classification function or a general regression function. SVM finds an optimal separating hyper-plane between data points of different classes in a high dimensional space. Support vectors are the points that form the decision boundary between classes. SVM decoding models are based on the *Structural Risk Minimization* (SRM) principle from statistical learning theory. Different SVM classification models have been selected due to their high discrimination power and low generalization error.

SVM decoding models can be developed by using different kernel functions. Two issues are catered in order to optimize SVM models, i.e. selection of suitable kernel function and its associated parameters (model selection) [12]-[14]. In optimal kernel selection, first, SVM are tested on various kernels for improved classification performance and minimum training error [15]. In kernel model selection, mostly iterative search is applied in order to optimize the parameters within a specified range [13]. In the current work, we are optimizing different SVM kernel functions in the decoding of message bits in watermarking.

B. SVM Classification Models

SVM performs pattern classification between two classes to find a decision surface that has maximum distance to the closest points in the training set. SVM views the classification problem as a quadratic optimization problem and avoids the *curse of dimensionality* by placing an upper bound on the margin between classes.

For a linearly separable data, such a hyper-plane is determined by maximizing the distance between the support vectors. Consider n training pairs (x_i, y_i), where $x_i \in R^N$ and $y_i \in [1.-1]$, the decision surface is defined as:

$$f\left(x\right) = \sum_{i=1}^{n} \alpha_i y_i x_i^T .x + b \tag{1}$$

where the coefficient $\alpha_i > 0$ is the Langrange multiplier in an optimization problem. A vector x_i that corresponds to $\alpha_i > 0$ is called a support vector. *f(x)* is independent of the dimensions of the feature space and the sign of *f(x)* gives the membership class of *x*. In case of Linear-SVM, the kernel function is simply the dot product of two points in the input space.

In our case, to find an optimal hyper-plane for non-separable patterns, the solution of the following optimization problem is sought.

$$\Phi(w,\xi) = \frac{1}{2} w^T w + C\sum_{i=1}^{N} \xi_i \ , \tag{2}$$

subject to the condition $y_i\left(w^T\Phi(x_i)+b\right) \geq 1-\xi_i,\ \xi_i \geq 0.$ where $C > 0$ is the penalty parameter of the error term $\sum_{i=1}^{N}\xi_i$ and Φ(x) is nonlinear mapping. The weight vector w minimizes the cost function term $w^T w$.

For nonlinear data, we have to map the data from the low dimension N to higher dimension M through Φ(x) such that $\Phi : R^N \rightarrow F^M, M >> N$. Each point Φ(x) in the new space is subject to Mercer's theorem in which different kernel functions are defined as: $K(x_i,x_j) = \Phi(x_i).\Phi(x_j)$. In this way, we can construct the nonlinear decision surface $f(x)$ in terms of $\alpha_i > 0$ and kernel function $K(x_i,x_j)$ as:

$$f(x) = \sum_{i=1}^{N_S}\alpha_i y_i K(x_i,x) + b = \sum_{i=1}^{N_S}\alpha_i y_i \Phi(x_i)\cdot\Phi(x_j) + b \tag{3}$$

where, *Ns* is the number of support vectors.

C. SVM Kernels

In SVM, there are two types of kernel functions, i.e. local (Gaussian) kernels and global (linear, polynomial, sigmoidal) kernels. The measurement of local kernels is based on a distance function while the performance of global kernels depends on the dot product of data samples. Linear, polynomials and radial basis functions are mathematically defined as:

$K(x_i,x_j) = x_i{}^T.x_j$ (Linear kernel with parameter C)

$K(x_i,x_j) = \exp(-\gamma\|x_i - x_j\|^2)$ (RBF with kernel parameter γ, C)

$K(x_i,x_j) = [\gamma < x_i, x_j > +r\ ]^d$ (Polynomials kernel with parameters *γ, r, d* and C)

Linear, RBF and polynomial kernels have one, two, and four adjustable parameters respectively. All these kernels share one common cost parameter *C*, which represents the constraint violation of the data points occurring on the wrong side of the SVM boundary. The parameter *γ* in the RBF shows the width of Gaussian functions.

In order to obtain optimal values of these parameters, there is no general rule about the selection of grid range and step size [18]. In the present work to select the optimal parameters of kernel functions, grid search method is used. To achieve the generalization capability of watermark decoding model, 4-fold cross-validation is used [13].

III. Proposed Intelligent Watermark Decoding Scheme

We use SVM to intelligently decode the message after an attack. In the proposed scheme, the decoding is carried out using 22 features. We evaluate the performance of our decoding scheme in terms of Bit Correct Ratio BCR [16]. BCR is an important performance characteristic of any decoding module. BCR of decoded message is computed as in (4)

$$BCR\left(M,M'\right)=\frac{\sum_{i=1}^{L_m}\overline{\left(m_i\oplus m_i'\right)}}{L_m} \qquad (4)$$

where *M* represents the original, while *M'* represents the decoded message, L_m is the length of the message and $\oplus$ represents exclusive-OR operation.

Even a small margin of improvement in BCR, can heavily effect the performance of a watermarking system in the context of distortion introduced, both due to malicious as well as non malicious attacks. It should be noted that $(1-BCR)$ represents bit incorrect ratio. The basic architecture of our proposed scheme is shown in Fig. 1.

In our work, we compare the performance of our decoding scheme with Hernandez scheme. In the scheme, watermark is embedded in DCT domain in the form +1 and -1 bits using the Watson's perceptual model. If no noise or attack is added, watermark forms two non-overlapping Gaussian distributions as represented in Fig. 2. Previously Hernandez et al [7] have used a simple threshold decoder to decode the message from the watermarked image in the form of +1 and -1 bits.

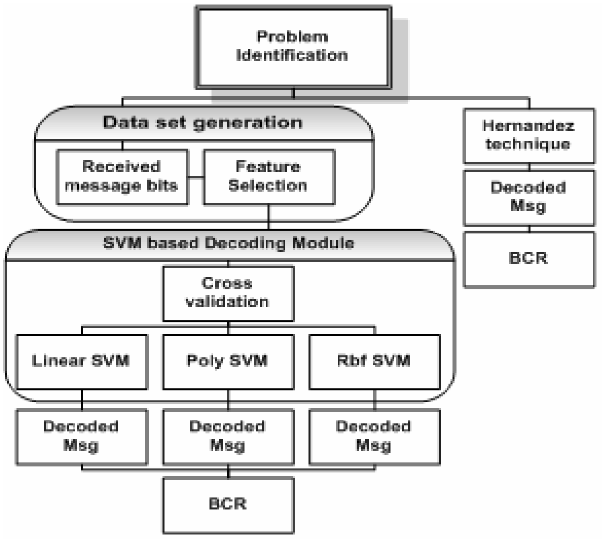

Fig. 1 Overall work Flow Diagram

In case of added noise, the distribution takes the form as represented in Fig. 3. In that case, a normal threshold decoder may not decode the message well. We introduce a novel idea of decoding the message using intelligent technique. Our idea is that a non-separable message in lower dimensional space may be separable if we take it to higher dimensional space. We use SVM to decode the embedded message. In this way, we convert decoding problem into a pattern classification and machine learning problem.

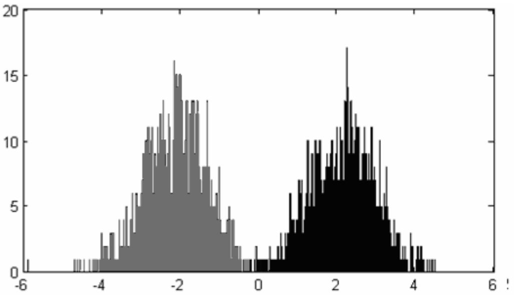

Fig. 2 Distribution of sufficient statistics corresponding to bit +1 and -1 without attack

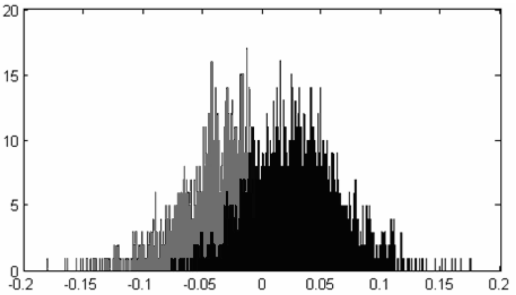

Fig. 3 Distribution of sufficient statistics corresponding to bit +1 and -1 after attack

A. Data Set Generation

To perform experiments and analyze the performance of our purposed scheme, we generated a data set of 16000 bits of received messages. For this purpose first we use 5 different images each of size N = 256 x 256. Then we embed a message of size 128 bit in each image. The whole process is repeated 25 times, each time embedding key is changed. In this way, 125 messages are embedded in 5 different types of images. Gaussian attack with σ =10 is applied on each image. Finally, watermark is extracted from the attacked image. In this way, we form a data set representing 125 different messages in 5 different types of images attacked with Gaussian noise, forming 16000 bits. It can be considered a generalized data set representing most of the general properties of different type of images. Table I shows different parameters of our data set.

TABLE I
PARAMETERS OF DATA SET

Type of Images	Gray Scale
Number of images	5
Name of Images	Baboon, Lena, Trees, Boat & couple.
Size of images	256 *256
Size of Message	128 bits
Number of keys	25
Type of Attack	Gaussian Attack
Severity of attack	σ =10

B. Features Extraction

When the watermarked image is attacked, the message within the image is also corrupted with the same noise. We extract features for each bit of message, in two ways. In the first method, we combine all the statistical coefficients r$_i$ corresponding to a single bit in message and then sum them the number of times they are repeated.

$$r_i \overset{\Delta}{=} \sum_{\boldsymbol{k} \in G_i} \frac{\left|Y[\boldsymbol{k}]+\alpha[\boldsymbol{k}]S[\boldsymbol{k}]\right|^{c[\boldsymbol{k}]}-\left|Y[\boldsymbol{k}]-\alpha[\boldsymbol{k}]S[\boldsymbol{k}]\right|^{c[\boldsymbol{k}]}}{\sigma[\boldsymbol{k}]^{c[\boldsymbol{k}]}} \tag{5}$$

In this case, we get a numerical value corresponding to each bit. In the second method, we do not combine all the statistical coefficients to get the single numeric value; rather we keep all r_i as features of the bit and add r$_i$ of same channel only. In this way, we get 22 features corresponding to each bit. We train SVM classification models for 22 features. The corresponding actual message bit is used as the target in training phase.

Each element in training data set consists of a pair of input pattern and the corresponding target values. In the current work, the input pattern comprises 22 features. These are statistical coefficients r$_i$ of 22 channels representing message bits in the image, repetition of any bit is added in corresponding channels. The target value consists of bits of actual message embedded in the image. These target values of training dataset are used to make the SVM model learn the behavior of bits when distorted by certain conceivable noise.

C. Data Sampling Techniques

In this work, we applied 4-fold jackknife technique in cross validation. In this technique, we use 25 percent data for training and remaining 75 percent data is used for testing. To improve the statistical significance of our results, we repeat the process four times so that all the data can be used for training and testing. We perform *Grid Search* to obtain the optimal classification model. Based on highest average BCR we select the optimal model.

D. Grid Search

The optimal values of these parameters are adjusted by using grid search [13]. In Grid search, keeping first parameter constant, we change the value of second parameter. Then the same process is repeated for new input value of the first parameter. In this way, the optimal values of both parameters are found at which the SVM models behave optimally. Suitable grid range and step size is estimated for SVM kernels.

Poly SVM has four adjustable parameters; *d, r, γ* and *C*. However, to simplify problem, the values of degree and coefficient are fixed at d = 3, r = 1. The optimum values of γ and C are then selected. In case of Poly-SVM, a grid range of C = [2^{-2} to 2^2] with step ΔC = 0.4, γ = [2^{-2} to 2^8] with step size Δγ = 0.4 are used. In case of RBF SVM, the range of grid, and step size of C and γ are selected as C = [2^{-2} to 2^2], ΔC = 0.4, γ = [2^{-2} to 2^8], Δγ = 0.4. The optimal value of C parameter for linear kernel is obtained by adjusting the grid range of C = [2^{-2} to 2^5] with ΔC = 0.4. Fig. 4 shows the performance of RBF-SVM model using grid search for optimizing the values of C and gamma parameters.

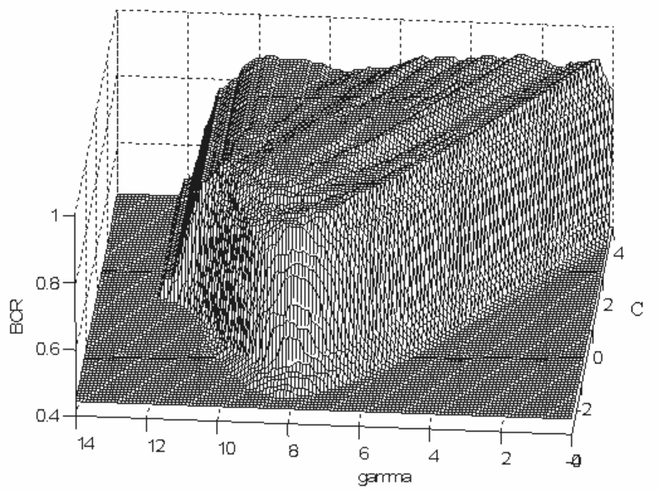

Fig. 4 Dependency of performance of SVM model on C and γ

E. SVM Classification Models Optimization

In our work, we obtain various SVM classification models by training different SVM kernel functions such as Linear, polynomial and RBF. These SVM classification models are first trained on the training data using the above sampling technique. The classification error in SVM models is due to those data points, which occur on the wrong side of the SVM boundary. Decoding performance of these models is optimized using grid search technique. In order to obtain optimal values of these parameters, we try various values of grid range and step size as described in section D

In our proposed scheme, once the SVM model is trained, it can be used to test the performance on same or entirely different data. The testing data, consisting of novel data samples, is given as an input to the trained SVM model. The results obtained by using SVM models are used to estimate the decoding performance in terms of BCR. For example, in case of RBF kernel, its γ and C parameters are optimized on training data and its decoding performance is evaluated on testing data.

IV. Results and Discussion

Our main objective is to form a model, general enough to perform well even for a new data. We perform our cross-validation experiments for 22 features data set. We divide whole data set into four equal sets. One set is used for training and remaining 3 sets for testing. We repeat this process for all the four sets. We then calculate the average BCR.

The experimental results are obtained using Pentium IV machine (2.4 GHz, 512 Mb RAM). Linear RBF and Poly-SVM are used separately and their results are compared. To develop SVM models, we use 'LIBSVM', a toolbox for support vector machines, which has basic functions for the creation and simulation of SVM model [18]. We use a MATLAB 7 [20] interface of LIBSVM for the creation and simulation of SVM models.

As a reference, we compute BCR message decoded by Hernandez scheme [7]. Data is given in the same fashion as it is used in our SVM models for training\testing phase in cross-validation. Results of Hernandez technique is given in Table II. This table shows that average BCR value 0.984 for 16000 message bits.

TABLE II
RESULTS OF HERNANDEZ SCHEME FOR DATA USED IN CROSS VALIDATION

Training Data (bits)	BCR	Avg. BCR	Test Data (bits)	BCR	Avg. BCR
4000	0.9830	0.984	12000	0.98433	0.984
4000	0.9805		12000	0.98517	
4000	0.98775		12000	0.98275	
4000	0.98475		12000	0.98375	

Table III and Table IV show the results of SVM models trained and tested using 22 features Data set. We observe very promising results. Table III shows that Poly SVM has average BCR is 1 for training data at the optimized values of $\gamma = 194$ and C = [0.4 - 2].

Table IV shows results of Linear, Poly, and RBF kernels on testing data. Though Linear and RBF SVM do not show high improvement in performance, but in case of Poly-SVM, we have achieved 100 percent accuracy. BCR is 1 that means all the 12000 bits of test data are predicted correctly. In case of distorted watermark, this is indeed a high improvement. Overall performance of our models compared to Hernandez scheme is shown in Fig. 5. This figure shows the order of performance of different classification models on test data as follows:

Poly SVM > Linear SVM > RBF SVM > Hernandez

In this figure the order of performance of different classification models on training data is as follows:

Poly SVM > RBF SVM > Linear SVM > Hernandez

TABLE III
SVM RESULTS ON TRAINING DATA IN CROSS VALIDATION

Type of SVM	C	Gamma Γ	Train Data (bits)	BCR	Avg. BCR
Linear	48.503	-	4000	0.9852	0.9853
	48.503	-	4000	0.9832	
	111.43	-	4000	0.9868	
	111.43	-	4000	0.9860	
Poly	0.4 to 2	194	4000	1	1
	0.4 to 2	194	4000	0.9998	
	0.4 to 2	194	4000	1	
	0.4 to 2	194	4000	1	
RBF	0.75786	5.2768	4000	0.9850	0.9877
	1.3195	6.9644	4000	0.9875	
	1	1.7411	4000	0.9868	
	2.2974	9.1896	4000	0.9915	

TABLE IV
SVMS RESULTS ON TESTING DATA IN CROSS VALIDATION

Type of SVM	C	Gamma Γ	Test Data (bits)	BCR	Avg. BCR
Linear	48.503	-	12000	0.9855	0.9855
	48.503	-	12000	0.98617	
	111.43	-	12000	0.9850	
	111.43	-	12000	0.98525	
Poly	0.4 to 2	194	12000	1	1
	0.4 to 2	194	12000	1	
	0.4 to 2	194	12000	1	
	0.4 to 2	194	12000	1	
RBF	0.75786	5.2768	12000	0.98483	0.9843
	1.3195	6.9644	12000	0.98475	
	1	1.7411	12000	0.98333	
	2.2974	9.1896	12000	0.98325	

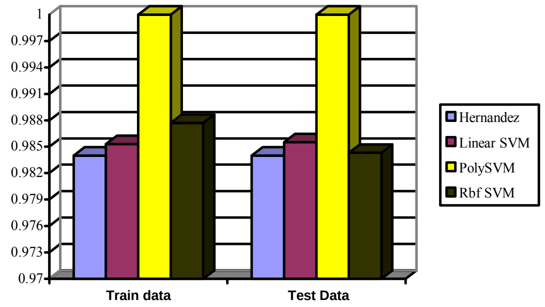

Fig. 5 Performance of cross validation on train & test data using 22 feature after Grid Search

We observe a better performance of our proposed scheme. This scheme is capable of intelligent decoding of watermark in any sort of hostile environment with a number of possible attacks on an adversary's disposal.

We achieved 100% accuracy by using cross-validation data sampling scheme. In the current work, our results are better than the Hernandez [7] scheme.

V. CONCLUSION

We have been able to validate the exploitation of machine learning concepts for decoding purposes. The experimental results in this paper have demonstrated that SVM based decoding is quite effective in a hostile environment. Proposed SVM models have successfully predicted the distorted bit of the watermark in the image. SVM models have performed better than the Hernandez scheme on both training and testing dataset. We have used this technique for image watermarking, but this methodology can also be applied for audio and video watermarking.

ACKNOWLEDGEMENTS

The authors acknowledge the support of Dr. Ajmal Bangash, Assistant Professor, GIK Institute during the course of this work. The author, S. Fahad Tahir, is also grateful to

National Engineering and Scientific Commission (NESCOM), Government of Pakistan, for the award of fellowship for MS.

REFERENCES

[1] I.J. Cox, M.L. Miller, J.A. Bloom, Digital Watermarking and Fundamentals, Morgan Kaufmann, San Francisco, 2002.

[2] Piva, M. Barni, F. Bartolini, V. Cappellini, "DCT-based watermark recovering without resorting to the uncorrupted original image," Proc Int. Conf. Image Processing, Oct. 1997, vol. 1 pp. 520-523.

[3] S. Lyu and H. Farid, Detecting hidden messages using high-order statistics and support vector machines, 5th international workshop on Information Hiding, Noordwijkerhout, The Netherlands, 2002.

[4] Y. Fu, R. Shen and H. Lu, Optimal watermark detection based on support vector machines, Proc. of International Symposium on Neural Networks, Dalian, China, August 19-21, 2004, pp.552-557.

[5] P.T. Yu, H.H. Tsai, J.S. Lin, Digital watermarking based on neural networks for color images, Signal Processing, Elsevier Science, 81, 663-671, 2001.

[6] Asifullah Khan "A Novel Approach to decoding: Exploiting anticipated attack information using Genetic programming." International Journal of knowledge based and Intelligent engineering Systems 9(2006) pp. 1-10.

[7] J.R. Hernandez, M. Amado, F. Perez-Gonzalez, DCT-Domain watermarking techniques for still images: Detector performance analysis and a new structure, IEEE Trans. Image Process. 9 (1) (2000) 55–68.

[8] Zhang Li1, SamKwong2, Marian Choy2, Wei-wei Xiao 1, Ji Zhen1, and Ji-hong Zhang1, An Intelligent Watermark Detection Decoder Based on Independent Component Analysis, DOCIS Documents in Computing and Information Science 2003.

[9] S. Kırbız, Y. Yaslan, B. Günsel, Robust Audio Watermark Decoding By Nonlinear Classification, Multimedia Signal Processing and Pattern Recognition Lab, 2005.

[10] Veera Venkataramani, Shantanu Chakrabartty, and William Byrne. Support Vector Machines For Segmental Minimum Bayes Risk Decoding Of Continuous Speech, 2003 IEEE Automatic Speech Recognition and Understanding Workshop.

[11] Asifullah Khan, Anwar M. Mirza, Genetic perceptual shaping: Utilizing cover image and conceivable attack information during watermark embedding, Sep.2005. Elsevier. Journal of Information Fusion.

[12] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing Multiple Parameters for Support Vector Machines, Machine Learning," vol. 46, No. 1-3, pp. 131-159, 2002.

[13] C.W. Hsu, C.C. Chang, and C.J. Lin, "A practical guide to support vector machines," Technical report, *Department of Computer Science & Information Engineering, National* Taiwan University, 2003.

[14] C. Staelin, "Parameter selection for support vector machines," Technical report, *HP Labs*, Israel, 2002.

[15] B. Moghaddam, and M.H. Yang, "Learning Gender with support faces," in IEEE Transaction on *Pattern Analysis and Machine Learning*, vol. 24, 2002.

[16] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification," *John Wiley & Sons, Inc.*, New York, 2nd edition, 2001.

[17] Hsiang-Cheh Huang, Lakhmi C. Jain, Jeng-Shyang Pan, Intelligent Watermarking Techniques, World Scientific Pub Co Inc, 2004.

[18] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[19] A. Majid, "Optimization and combination of classifiers using Genetic Programming," PhD Thesis, *Faculty of Computer Science, GIK institute*, Pakistan. Dec. 2005.

[20] MATLAB 7.0, Mathworks, http://www.mathworks.com.