# A Novel Image Watermarking Scheme Using Extreme Learning Machine

**Anurag Mishra**[1] **Amita Goel**[2] **Rampal Singh**[3] **Girija Chetty**[4] **Lavneet Singh**[4]

*1. Department of Electronics, Deendayal Upadhyay College, University of Delhi, New Delhi, India*
***anurag_cse2003@yahoo.com***
*2. Department of Information Technology, Maharaja Agrasen Institute of Technology, GGSIP University, New Delhi, India*
*3. Department of Computer Science, Deendayal Upadhyay College, University of Delhi, New Delhi, India*
*4. Faculty of Information Sciences and Engineering, University of Canberra, Australia.*

***Abstract*** — In this paper, a novel digital image watermarking algorithm based on a fast neural network known as Extreme Learning Machine (ELM) for two grayscale images is proposed. The ELM algorithm is very fast and completes its training in milliseconds unlike its other counterparts such as BPN. The proposed watermarking algorithm trains the ELM by using low frequency coefficients of the grayscale host image in transform domain. The trained ELM produces a sequence of 1024 real numbers, normalized as per N(0, 1) as an output. This sequence is used as watermark to be embedded within the host image using Cox's formula to obtain the signed image. The visual quality of the signed images is evaluated by PSNR. High PSNR values indicate that the quality of signed images is quite good. The computed high value of SIM (X, X*) establishes that the extraction process is quite successful and overall the algorithm finds good practical applications, especially in situations that warrant meeting time constraints.

***Keywords:*** Digital Image Watermarking, Extreme Learning Machine (ELM), Watermark Extraction, SIM(X, X*), Real time applications

## I. INTRODUCTION

Different image processing applications require implementation within the specified time constraints. One such application in which the embedding and extraction of external data should be accomplished with the minimum time complexity is digital watermarking of images and video sequences. Besides this, the embedding and extraction processes need to be optimized one and should not result in loss of visual quality after embedding. A number of research groups have employed different soft computing tools to optimize these processes to develop a robust watermarking package [1-4]. On the other hand, it remains an interesting avenue to meet timelines for these processes without compromising visual quality and robustness. Such an algorithm may be successfully applied in case of videos at a later stage. Artificial Neural Networks (ANNs) are successfully employed to embed and extract watermarks. However traditional neural network training methods based on gradient descent optimization such as a BPN suffer from various drawbacks such as long training time, multiple local minima etc [1]. Recently, E. G. B. Huang et al. developed a fast algorithm for training ANNs, popularly known as the extreme learning machine (ELM). The benefits of this approach are that it has only one tunable parameter, namely, the number of hidden neurons, and its training process consists of only a single step, thereby reducing time up to a large extent. The training of this machine is reported to have been completed within milliseconds with a reasonably good accuracy [5].

In the present work, two standard grayscale host images – Lena and Baboon are used in transform domain to embed a watermark sequence obtained as a result of training the ELM. It is found that the visual quality of the signed images is good as indicated by computed PSNR values. The extracted watermarks are also subject to SIM(X, X*) computation. High values of SIM parameter in case of both these images indicate successful recovery process.
The paper is organized as follows. Section II of this paper gives mathematical review of Extreme Learning Machine (ELM). Section III presents the experimental details of proposed watermarking algorithm while section IV discusses the observed results and their analysis. Finally, the paper is concluded in section V.

## II. REVIEW OF EXTREME LEARNING MACHINE MODEL

The Extreme Learning Machine [5, 6, 7] is a Single hidden Layer Feed forward Neural Network (SLFN) architecture. Unlike traditional approaches such as Back Propagation (BP) algorithms which may face difficulties in manual tuning control parameters and local minima, the results obtained after ELM computation are extremely fast, have good accuracy and has a solution of a system of linear equations. For a given network architecture, ELM does not have any control parameters like stopping criteria, learning rate, learning epochs etc., and thus, the implementation of this network is very simple. The main concept behind this algorithm is that the input weights (linking the input layer to the hidden layer) and the hidden layer biases are randomly chosen based on some continuous probability distribution function such as uniform probability distribution in our simulation model and the output weights (linking the hidden layer to the output layer) are then analytically calculated using a simple generalized inverse method known as Moore – Penrose generalized pseudo inverse [9].

**(i) Mathematics of ELM Model**

Given a series of training samples $(x_i, y_i)_{i=1,2,\ldots,N}$ and $\hat{N}$ the number of hidden neurons where $x_i = (x_{i1},\ldots,x_{in}) \in \Re^n$ and $y_i = (y_{i1},\ldots,y_{im}) \in \Re^m$, the actual outputs of the single-hidden-layer feed forward neural network (SLFN) with activation function $g(x)$ for these $N$ training data is mathematically modelled as

$$\sum_{k=1}^{\hat{N}} \beta_k g(\langle w_k, x_i \rangle + b_k) = o_i, \quad \forall\, i = 1,\ldots, N \quad (1)$$

where $w_k = (w_{k1},\ldots,w_{kn})$ is a weight vector connecting the $k^{th}$ hidden neuron, $\beta_k = (\beta_{k1},\ldots,\beta_{km})$ is the weight vector connecting the $k^{th}$ hidden neuron and output neurons and $b_k$ is the threshold bias of the $k^{th}$ hidden neuron. The weight vectors $w_k$ are randomly chosen. The term $\langle w_k, x_i \rangle$ denotes the inner product of the vectors $w_k$ and $x_i$ and $g$ is the activation function.

The above $N$ equations can be written as

$$H\beta = O \quad (2)$$

and in practical applications $\hat{N}$ is usually much less than the number $N$ of training samples and $H\beta \neq Y$ ,where

$$H = \begin{bmatrix} g(\langle w_1, x_1\rangle + b_1) & . & . & . & g(\langle w_{\hat{N}}, x_1\rangle + b_{\hat{N}}) \\ . & & & & . \\ . & & . \; . \; . & & . \\ . & & & & . \\ g(\langle w_1, x_N\rangle + b_1) & . & . & . & g(\langle w_{\hat{N}}, x_N\rangle + b_{\hat{N}}) \end{bmatrix}_{N\times\hat{N}}$$

$$\beta = \begin{bmatrix} \beta_1 \\ . \\ . \\ . \\ \beta_{\hat{N}} \end{bmatrix}_{\hat{N}\times m}, \quad O = \begin{bmatrix} o_1 \\ . \\ . \\ . \\ o_N \end{bmatrix}_{N\times m}$$

and

$$Y = \begin{bmatrix} y_i \\ . \\ . \\ . \\ y_N \end{bmatrix}_{N\times m} \quad (3)$$

The matrix $H$ is called the hidden layer output matrix. For fixed input weights $w_k = (w_{k1},\ldots,w_{kn})$ and hidden layer biases $b_k$, we get the least-squares solution $\hat{\beta}$ of the linear system of equation $H\beta = Y$ with minimum norm of output weights $\beta$, which gives a good generalization performance. The resulting $\hat{\beta}$ is given by $\hat{\beta} = H^{+}Y$ where matrix $H^{+}$ is the Moore-Penrose generalized inverse of matrix $H$ [9]. The above algorithm may be summarized as follows:

(ii) **The ELM Algorithm**

Given a training set

$S = \{(x_i, y_i) \in \Re^{m+n}, y_i \in \Re^m\}_{i=1}^{N} \sum$ , for activation function $g(x)$ and the number of hidden neurons $\hat{N}$ ;

**Step1:** For $k = 1,...,\hat{N}$ randomly assign the input weight vector $w_k \in \Re^n$ and bias $b_k \in \Re$.

**Step2:** Determine the hidden layer output matrix $H$.

**Step3:** Calculate $H^+$ .

**Step4:** Calculate the output weights matrix $\hat{\beta}$ by $\hat{\beta} = H^+ Y$ .

Many activation functions can be used for ELM computation. In the present case, Sigmoid activation function is used to train the ELM.

**(iii) Computing the Moore-Penrose Generalized Inverse of a matrix**

Definition 1.1: A matrix $G$ of order $\hat{N} \times N$ is the Moore-Penrose generalized inverse of real matrix $A$ of

order if $N \times \hat{N}$ $AGA = A, GAG = G$ and $AG, GA$ are symmetric matrices.

Several methods, for example orthogonal projection, orthogonalization method, iterative methods and singular value decomposition (SVD) methods exist to calculate the Moore-Penrose generalized inverse of a real matrix. In ELM algorithm, the SVD method is used to calculate the Moore-Penrose generalized inverse of H. Unlike other learning methods, ELM is very well suited for both differential and non – differential activation functions.

## III. EXPERIMENTAL DETAILS

### A. Generating and Embedding the Watermark

In this experiment, two grayscale images – Lena and Baboon of size 256*256 pixels each are taken as host images. The image object is divided into 8*8 pixel blocks and DCT of all such blocks is computed to transform the blocks into frequency domain. Zigzag scanning of each block is done to select first 21 AC coefficients barring the DC coefficient. Thus, a dataset of size 1024*21 is created which holds 21 selected coefficients from each of 1024 blocks in all. From this dataset, the mean of each row is computed and placed at the first column position. This results in creation of another dataset of size 1024 x 22 wherein the mean values (labels) are fixed in column 1. This is used as the training dataset for the ELM to be used in regression mode. Once the ELM algorithm is trained, it gives as its output the predicted values for the mean calculated in the form of a vector of size 1024 * 1. The predicted output values $f(\tilde{x})$ corresponding to each block are kept in a separate data file. This output vector is used as watermark to be embedded within the host image using Cox's formula given in Eqn. 4.

$$v'_i = v_i(1.0 + \alpha \times f(\tilde{x})) \tag{4}$$

where $f(\tilde{x})$ is output of ELM after training, $v_i$ are DCT coefficients and $v_i$' are coefficients of the signed image.. The parameter α is known as embedding strength and is optimized to be 0.5 for all our practical calculations. The computed watermark is embedded into each block and the inverse DCT of each block is taken to retrieve the signed image. Listing 1 gives watermark embedding algorithm.

**Listing 1: Watermark Embedding Algorithm**

1. Apply 8*8 size block code on host image
2. Convert the host image in transform domain by taking block wise DCT
3. Apply zigzag scanning to all AC coefficients of each block and select first 21 coefficients from each block barring the DC coefficient. Thus develop a dataset of size 1024 * 21 using these coefficients
4. Compute the mean of all 21 coefficients for each row and place it in first column as label. Thus, recreate a dataset of size 1024 * 22
5. Train the ELM in regression mode by supplying this dataset to the machine. As a result, the ELM produces an output vector of size 1024 * 1 which is used as watermark to be embedded within the host image using Cox's formula
6. Take Inverse DCT (IDCT) to obtain signed image

The block diagram of the embedding procedure is shown in Fig. 1.

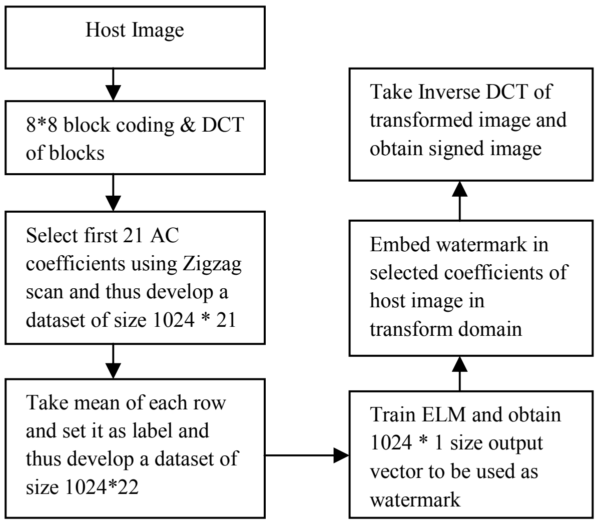

Host Image
8*8 block coding & DCT of blocks
Select first 21 AC coefficients using Zigzag scan and thus develop a dataset of size 1024 * 21
Take mean of each row and set it as label and thus develop a dataset of size 1024*22
Train ELM and obtain 1024 * 1 size output vector to be used as watermark
Embed watermark in selected coefficients of host image in transform domain
Take Inverse DCT of transformed image and obtain signed image

**Fig. 1:** Watermark Embedding Procedure

**B. Extracting the Watermark:**

In the extraction procedure, first of all, 8*8 block wise DCT of both host and signed images are computed and the coefficients of the original image which are used in embedding process are subtracted from the respective coefficients of the signed image. In this manner, both the original and recovered watermark sequences X and X* are known. A statistical similarity correlation check is performed over X and X* as given by Eqn. 5. Listing 2 depicts the watermark extraction algorithm.

$$\text{SIM}(X,\ X^*) = \sum_{i=1}^{n}(X.X^*) / \sum_{i=1}^{n}\sqrt{(X, X^*)} \qquad (5)$$

**Listing 2: Watermark Extraction Algorithm**

1. Divide both the original and watermarked images into 8*8 size blocks
2. Compute DCT of all blocks of both images
3. Subtract only those computed coefficients of the original image from the respective coefficients of signed image which are used in embedding process and thus recover the watermark X*
4. Compute the SIM(X, X*) correlation parameter

The block diagram for extraction process is shown in Fig. 2.

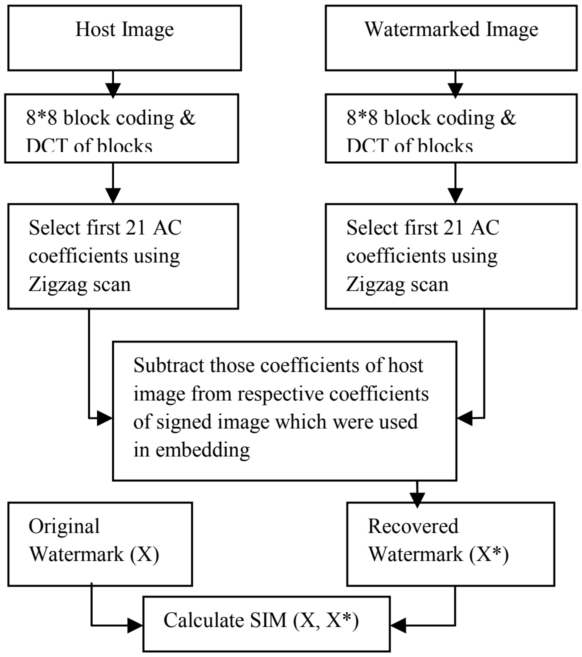

Host Image
Watermarked Image
8*8 block coding & DCT of blocks
8*8 block coding & DCT of blocks
Select first 21 AC coefficients using Zigzag scan
Select first 21 AC coefficients using Zigzag scan
Subtract those coefficients of host image from respective coefficients of signed image which were used in embedding
Original Watermark (X)
Recovered Watermark (X*)
Calculate SIM (X, X*)

**Fig. 2:** Watermark Extraction Procedure

IV. RESULTS AND DISCUSSION

Fig. 3 depicts two grayscale host images – Lena and Baboon. As indicated in section III(A), the output column vector of ELM is first normalized to N(0, 1) and subsequently embedded within these images to obtain signed images depicted in Fig. 4(a) and (b) using formula given in Eqn. 4. Their respective RMSE and PSNR values are mentioned above these images. High computed PSNR values indicate that the visual quality of these images is very good. Fig. 5(a) and (b) respectively show the SIM plots for the watermarks recovered from signed images of Fig. 4(a) and (b). Their computed values are

**(a)**

**(b)**

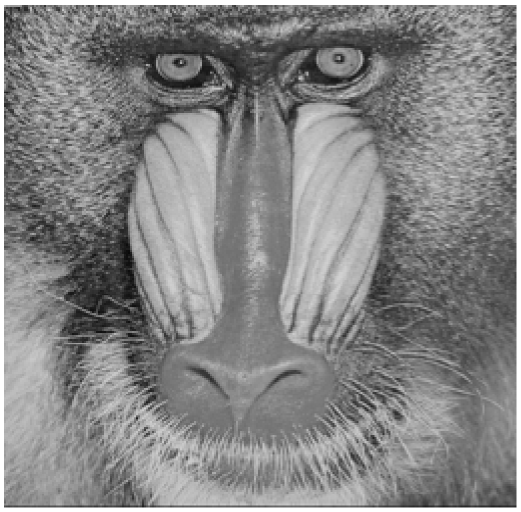
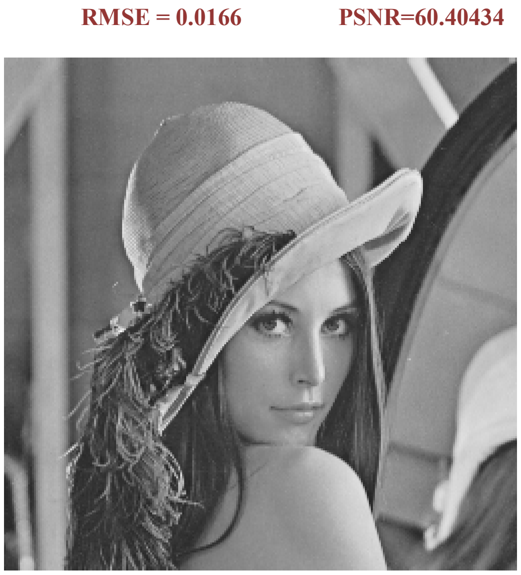
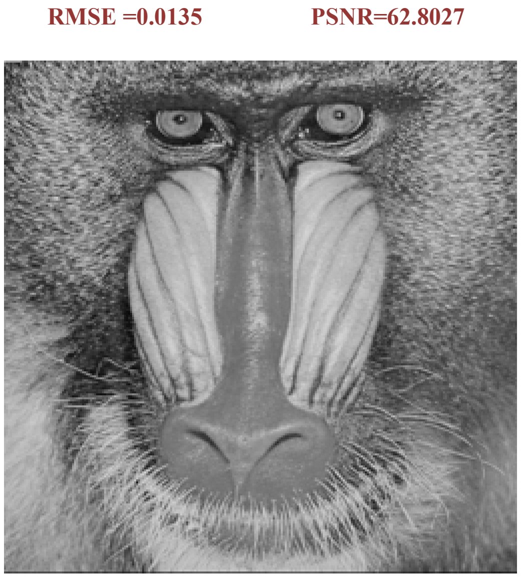


**Fig 3:** Original host Images – (a) Lena and (b) Baboon



(a)



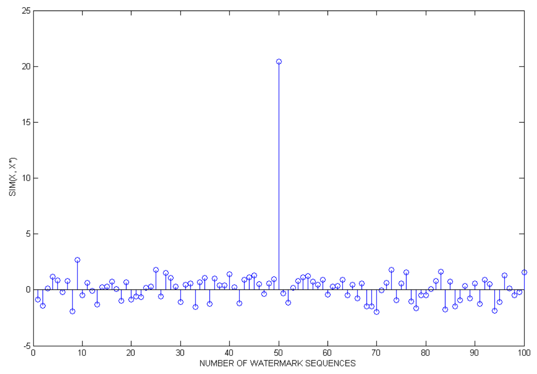
(b)

**Fig 4:** Signed Images – (a) Lena and (b) Baboon

respectively SIM = 20.4956 and SIM = 26.34663. As it is evident, the SIM values are also high, thereby indicating a very successful recovery process by using Cox's algorithm.

Table 1 depicts the ELM training time, time consumed in embedding and extraction processes from Fig. 4(a) as a function of number of hidden neurons. It can be noted that ELM training time ranges from 15.6 ms to 109 ms as a function of number of hidden neurons (N). This indicates a fast training process unlike training of a gradient descent based BPN which usually gets trapped in multiple local minimas and thus waste time. This feature makes this algorithm suitable for real time image processing applications in general and image and video watermarking in particular. In this case, the

**(a)**



**(b)**

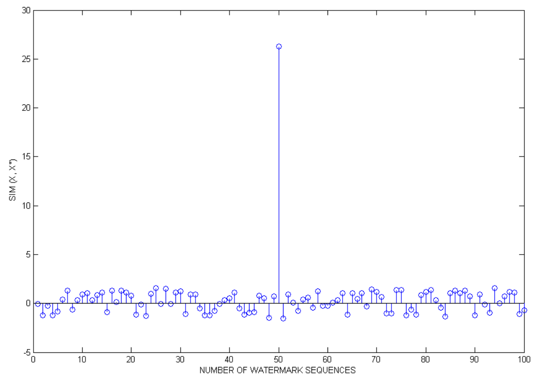


**Fig. 5:** SIM plots for watermarks extracted from images depicted in Fig. 4(a) and (b) respectively

**Table 1:** Three different computed time spans for ELM based watermarking of Lena image

| Number of Hidden Neurons (N) | ELM Training Time (Sec) | Embedding Time (Sec) | Extraction Time (Sec) |
|---|---|---|---|
| 20 | 0.0156 | 1.6563 | 0.0156 |
| 40 | 0.0313 | 1.7031 | 0.0156 |
| 60 | 0.0625 | 1.8281 | 0.0156 |
| 80 | 0.0781 | 1.7656 | 0.0156 |
| 100 | 0.1094 | 1.7500 | 0.0313 |

embedding time encompasses both ELM training as well as actual embedding using Eqn. 4 while extraction time only involves extraction process. It may be noted that the embedding time and the time of extraction are also within the expected range. Table 2 compiles the same time spans for the second image Baboon. In this case also, the training time ranges from 15.6 ms to 109 ms which is also identical to the results obtained in case of Lena image. The other two time spans are also on the similar lines. In case of both these images, it is pertinent to mention that with increase in the number of hidden neurons (N), the computed values of PSNR and SIM(X, X*) are by and large constant. Thus, it is only the ELM training time and therefore, to some extent the embedding time that vary with N.

**Table 2:** Three different computed time spans for ELM based watermarking of Baboon image

| Number of Hidden Neurons (N) | ELM Training Time (Sec) | Embedding Time (Sec) | Extraction Time (Sec) |
|---|---|---|---|
| 20 | 0.0156 | 1.7188 | 0.0156 |
| 40 | 0.0313 | 1.6563 | 0.0156 |
| 60 | 0.0469 | 1.6875 | 0.0313 |
| 80 | 0.1094 | 1.7344 | 0.0156 |
| 100 | 0.1094 | 1.8438 | 0.0156 |

The small time span (~ msec) obtained for ELM training makes it a fit candidate for optimizing image watermarking procedures under real time constraints without compromising the visual quality of the signed image. This work may further be extended to watermarking of video sequences as it compulsorily requires embedding and extraction of watermarks within the given timelines

## V. CONCLUSIONS

A novel image watermarking algorithm based on training of a fast neural network, known as Extreme Learning Machine (ELM) is proposed in this paper. To the best of our knowledge, we have used this machine for image watermarking in regression mode for the first time. Two grayscale images – Lena and Baboon are embedded with the output of the trained ELM within the selected low frequency coefficients obtained after 8x8 block coding followed by DCT of the blocks. Visual quality of signed images is examined by PSNR. High PSNR values of signed images clearly indicate that embedding process is well optimized and the visual quality after embedding is quite good. Watermark extraction is performed using Cox's algorithm and SIM plots between the original and recovered watermarks are also obtained. High SIM values indicate that watermark extraction is also successful. Overall, the ELM training, embedding and extraction processes are well optimized and the algorithm finds good practical applications, especially in situations that require fulfilling time constraints such as video watermarking.

## VI. REFERENCES

[1] Charu Agarwal and Anurag Mishra, "A Novel Image Watermarking Technique using Fuzzy-BP Network", (2010) Proceedings of 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 102-105

[2] Rajesh Mehta, Anurag Mishra, Rampal Singh, Navin Rajpal, "Digital Image Watermarking in DCT Domain Using Finite Newton Support Vector Regression" (2010) Proceedings of 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp 123 – 126

[3] Mukesh C. Motwani and Fredrick C Harris, Jr, "Fuzzy Perceptual Watermarking for Ownership Verification" (2009) Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV'09), Las Vegas, Nevada, July 13-16

[4] Der-Chyuan Lou, Ming-Chiang Hu, and Jiang-Lung Liu, "Healthcare Image Watermarking Scheme Based on Human Visual Model and Back-Propagation Network", (2008) Journal of C.C.I.T, vol.37, no.1 , pp. 151-162

[5] M-B. Lin, G-B Huang, P. Saratchandran and N. Sudararajan, *"Fully complex extreme learning machine"*, Neurocomputing, (2005), vol (68), pp 306 – 314

[6] G -B Huang, Q -Y Zhu and C K Siew, "*Extreme Learning Machine: Theory and Applications"*, (2006), Neurocomputing, vol (70), pp 489-501

[7] G -B Huang, Q-Y Zhu and C K Siew, "*Real-Time Learning Capability of Neural Networks"*, (2006), IEEE Transactions on Neural Networks, vol 17(4), pp 863-878

[8] G-B Huang (2004), The Matlab code for ELM is available on: http://www.ntu.edu.sg/home/egbhuang

[9] D. Serre (2002), *"Matrices: Theory and Applications"*, Springer Verlag, New York Inc