Turk J Elec Eng & Comp Sci, Vol.18, No.2, 2010, © TÜBİTAK
doi:10.3906/elk-0907-140

Artificial neural network based chaotic generator for cryptology

İlker DALKIRAN, Kenan DANIŞMAN
Department of Electrical and Electronics Engineering, Faculty of Engineering Erciyes University, 38039, Kayseri-TURKEY
e-meil: ilkerd@erciyes.edu.tr, danismak@erciyes.edu.tr

Abstract

Chaotic systems are sensitive to initial conditions, system parameters and topological transitivity and these properties are also remarkable for cryptanalysts. Noise like behavior of chaotic systems is the main reason of using these systems in cryptology. However some properties of chaotic systems such as synchronization, fewness of parameters etc. cause serious problems for cryptology. In this paper, to overcome disadvantages of chaotic systems, the dynamics of Chua's circuit namely x, y and z were modeled using Artificial Neural Network (ANN). ANNs have some distinctive capabilities like learning from experiences, generalizing from a few data and nonlinear relationship between inputs and outputs. The proposed ANN was trained in different structures using different learning algorithms. To train the ANN, 24 different sets including the initial conditions of Chua's circuit were used and each set consisted of about 1800 input-output data. The experimental results showed that a feed-forward Multi Layer Perceptron (MLP), trained with Bayesian Regulation backpropagation algorithm, was found as the suitable network structure. As a case study, a message was first encrypted and then decrypted by the chaotic dynamics obtained from the proposed ANN and a comparison was made between the proposed ANN and the numerical solution of Chua's circuit about encrypted and decrypted messages.

Key Words: *Artificial neural network, chaos, cryptology.*

1. Introduction

Cryptology was as significant as weapons during the World War II and the Cold War. There were lots of studies to develop robust crypto-systems and to use them in communications. These studies have continued up to now. Today some of those crypto-systems called as "classical crypto-systems" are improved and still being used. [1-3]. Chaotic crypto-systems can be introduced as an alternative way to classical crypto-systems [4-10]. During the last two decades a growing amount of research effort has been dedicated to the investigation of chaos from different disciplines. Chaos is defined as stochastic behavior occurring in deterministic system [11]. It is known that chaotic systems are non-periodical and also sensitive to initial conditions, system parameters

and topological transitivity [12]. These properties are remarkable for cryptanalysts. The main reason of using chaotic systems in cryptography is especially noise like non-periodic dynamics of these systems.

Chaotic crypto-systems have two main sub-categories: continuous and discrete time chaotic crypto systems. In [13], Chua' s circuit based chaotic masking technique which can be considered as an application of continuous time analogue chaotic systems was presented as an example of secure communication systems. The system has two parts: a generator of chaotic dynamics (transmitter) and a receiver. The receiver has generally the same circuit topology with transmitter. A chaotic dynamic, produced by the transmitter, is used as an input for the receiver to synchronize the common dynamics of both systems. After synchronization, the message, coded with the chaotic dynamic, is transmitted from the transmitter and recovered using the same chaotic dynamic at the receiver [7, 8, 10, 14].

Simultaneously discrete time chaotic systems have been offered as an alternative way to encrypt messages. Iterative equations such as Logistic map or Henon map are used to generate chaotic dynamics. Then generated chaotic dynamics are separated into pieces. The starting and the finishing points of pieces are known by both the transmitter and the receiver. Each piece represents an alpha-numerical symbol in the alphabet, like a look-up table, and each character of the message is replaced with these pieces. The coded message is sent from the transmitter then the original message is recovered at the receiver using the same chaotic dynamics which are produced by the same iterative equations [4, 6, 9, 15].

It is known that chaotic circuits are sensitive to initial conditions. Initial conditions are static loads on capacitors and inductor in analogue circuitry as well. So anyone can not intentionally define the same static loads to generate the same chaotic dynamics in different times on the hardware. Additionally, communication channels are also non-ideal which cause synchronization problems between two analogue chaotic circuits [16, 17]. Up to now, both continuous and discrete time chaos based cryptographic systems are presented and in both of those chaos based cryptographic systems, the major weakness is fewness of system parameters. The system parameters represent the secret keys for a chaos based cryptographic algorithms. In cryptology, it is accepted that the cryptographic algorithm is known by everybody [1-3]. So the security of any crypto-system depends on only secret keys.

The development of ANNs comes from simulating intelligent tasks which are performed by human brain. They are most widely used by soft computing techniques that have the capability to capture and model complex input/output relationships of any system. The advantages of ANNs are the ability to generalize results obtained from known situations to unforeseen situations, the fast response time in operational phase, the high degree of structural parallelism, reliability and efficiency. If a set of input-output data pairs which belongs to a problem is available, ANNs can learn and exhibit good performance. For these reasons, application of ANNs has emerged as a promising area of research, since their adaptive behaviors have the potential of conveniently modeling strongly nonlinear characteristics [18-20].

In this paper, an ANN based chaotic generator is proposed to overcome the weaknesses of chaotic crypto-systems. The ANN model, presented in this work, was trained in MATLAB to generate chaotic dynamics $\hat{x}$, $\hat{y}$ and $\hat{z}$ which are similar to chaotic dynamics x, y and z generated by the numerical solution of Chua's circuit. As an application of chaotic cryptology, a message was encrypted and decrypted by the dynamics obtained from the proposed ANN model and a comparison was made between the proposed model and the numerical solution of Chua's circuit about encrypted and decrypted messages.

This paper is organized as follows: In Section 2, Chua's circuit configuration is described. ANN is

explained briefly and the proposed model is described in Section 3. In Section 4, the results of encryption and decryption processes are presented and in Section 5 concluding remarks are discussed.

2. Chua's circuit

Chua's circuit has a simple structure and easily produces chaotic dynamics with suitable parameters. Thus this circuit has attracted the interest of many researchers. Chua's circuit was described as "the first real physical system where chaos is observed in laboratory, confirmed by computer simulation, proven mathematically by two independent methods" [12, 21].

Figure 1 shows Chua's circuit. The circuit consists of three components that can store energy (C_1, C_2 and L), two linear resistors (R and R_S) and a nonlinear resistor (N_R) that is called Chua's diode. Using Kirchhoff's current and voltage laws, the obtained differential equations are shown in Equation (1) [22].

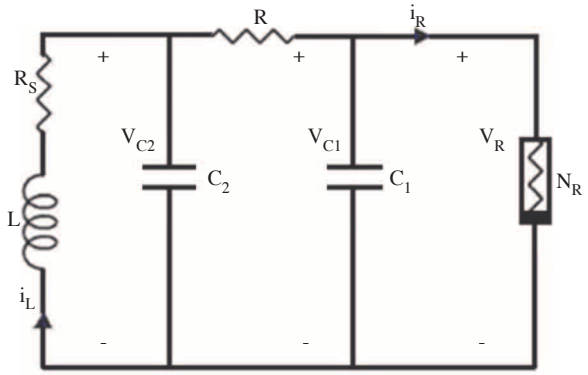

Figure 1. Chaotic Chua's circuit.

The mathematical model, $f(V_{C1})$ and characteristic of nonlinear Chua's diode are given in Equation (2) and Figure 2 respectively. The G_a and G_b are the slopes in the inner and outer regions respectively, and $\pm$ B_P denote the breakpoints.

$$\begin{aligned} C_1\tfrac{dV_{C1}}{dt} &= \tfrac{1}{R}(V_{C2}-V_{C1}) - f(V_{C1}) \\ C_2\tfrac{dV_{C2}}{dt} &= i_L - \tfrac{1}{R}(V_{C2}-V_{C1}) \\ L\tfrac{di_L}{dt} &= -V_{C2} - i_L.R_S \end{aligned} \tag{1}$$

$$i_R = f(V_{C1}) = G_b V_{C1} + \frac{1}{2}(G_a - G_b) * (|V_{C1}+B_P| - |V_{C1}-B_P|) \tag{2}$$

Equation (1) and (2) can be transformed into dimensionless forms which are given in Equation (4) and (5), by introducing the following variables.

$$\begin{aligned} x &= \tfrac{V_{C1}}{B_P} \qquad y = \tfrac{V_{C2}}{B_P} \qquad z = \tfrac{Ri_L}{B_P} \\ \alpha &= \tfrac{C_2}{C_1} \qquad \beta = \tfrac{C_2R^2}{L} \qquad \tau = \tfrac{t}{RC_2} \\ a &= G_aR \qquad b = G_bR \qquad c = B_P \end{aligned} \tag{3}$$

The dimensionless state equations given in Equation (4) were solved by MATLAB differential equation solver, namely "*ode 45*". The parameters and initial conditions were given by: $\alpha = 10$, $\beta = 14.87, a$= -1.27, b= -0.68, c= 1, $x_0 = 0$, $y_0 = 0$ and $z_0 = 1$. The obtained chaotic dynamics, named as x, y and z respectively, and double scroll are shown in Figure 3(a) and Figure 3(b).

$$\begin{aligned} \frac{dx}{d\tau} &= \alpha(y - x - f(x)) \\ \frac{dy}{d\tau} &= x - y + z \\ \frac{dz}{d\tau} &= -\beta y \end{aligned} \tag{4}$$

$$f(x) = bx + \frac{1}{2}(a - b) * (|x + c| - |x - c|) \tag{5}$$

The chaotic dynamics x, y and z are given in Figure 3(a). They represent V_{C1}, V_{C2} and i_L which are shown in Figure 1 respectively. The double scroll given in Figure 3(b) demonstrates relation between chaotic dynamics x and y.

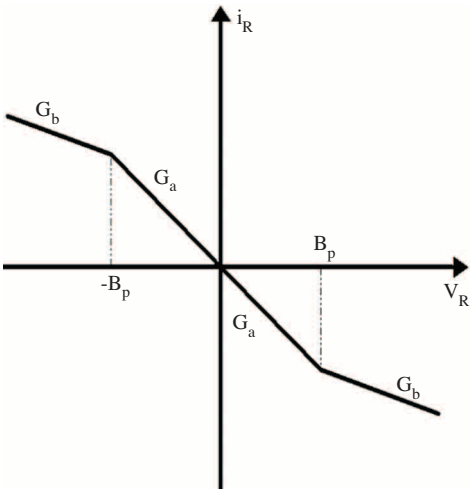

Figure 2. Characteristic of Chua's diode.

3. Artificial neural network

Because of distinctive advantages of human brain, such as powerful thought ability and solving complex problems, ANNs are based on the mechanism of the biologically inspired brain model. ANNs are most widely used soft computing techniques that have the capability to capture and model complex input/output relationships. ANNs have a wide application area such as medical, chemistry, electronics and automotive, etc. Some capabilities of ANNs like learning from experiences, generalizing from a few data and nonlinear relationship between inputs and outputs, come into prominence [22, 23-25].

In order to design an ANN system, three characteristics must be identified: (i) the neuron model, which is the single processing element; (ii) the network architecture, which defines the connections of the processing elements; (iii) the learning algorithm, which evaluates the weights of the connections.

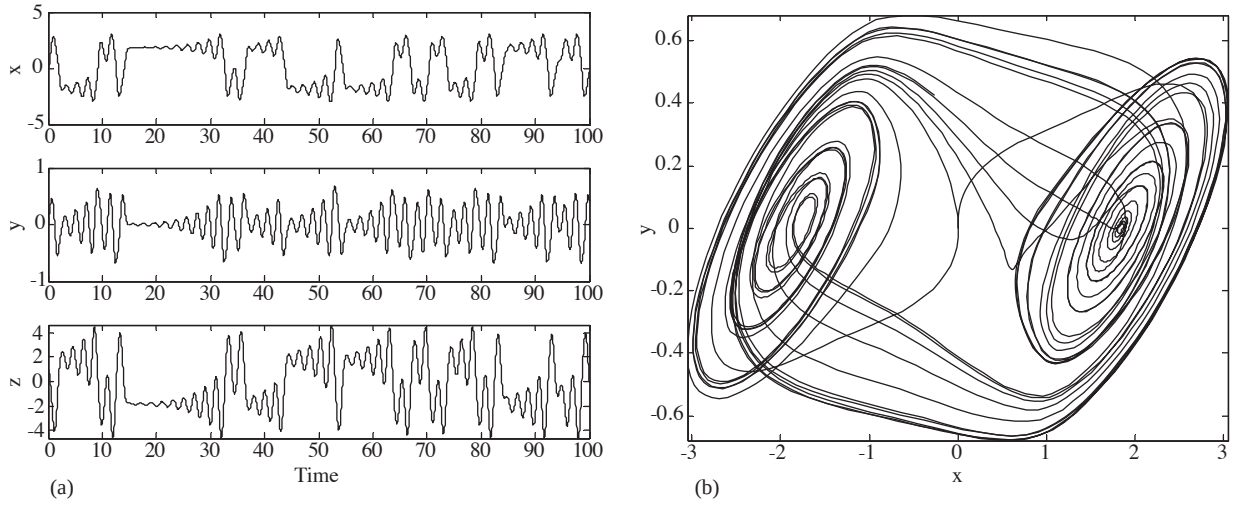

Figure 3. The dynamics (a) and the double scroll attractor of chaotic Chua's circuit (b) obtained by the numerical solution.

The structure of ANNs consists of simple processing elements or neurons which have weighted inputs and bias, summation and activation functions and output. Each neuron receives input signals from nearby neurons or external sources and gives an output signal which is propagated to other neurons or constitutes a part of the network output. In each neuron, the link among its inputs and output is provided by an activation function. The activation function is a non-decreasing function, such as hard limiting threshold function (sign function), linear or semi-linear function and smoothly limiting threshold function (sigmoid function) [26]. The block diagram of a neuron is shown in Figure 4.

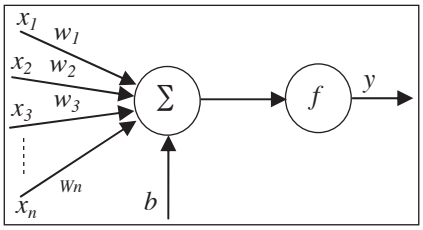

Figure 4. A simple neuron.

In Figure 4, x_i is the input, w_i is the weight, where *i=1, 2, 3 ... n,* b is the bias value and y is the output of neuron. The calculation of output y is given in Equation (6). f represents the activation function of the neuron.

$$y_j = f(\sum_i w_i x_i + b) \tag{6}$$

ANNs have many different architectures [18, 26, 27]. Multilayer perceptron (MLP) is the most used class of feed-forward networks for constructing non-linear transfer functions among inputs and one or more outputs [18, 26, 27]. A typical MLP neural network consists of inputs, outputs and one or more hidden layers with a predefined number of neurons. The neurons in the input layer only act as buffers for distributing the input signals to neurons in the hidden layer. During the training (learning) process, biases in each neuron and weights

between the neurons are adjusted by learning algorithm according to some criterions. These criterions are maximum allowable number of epochs, time and error (The mean square error, (mse) between the target output and the measured value for all the training set falls below a predetermined threshold) [28].

It is possible to prove that this architecture is able to perform any transformation when it is trained with the back-propagation (BP) learning algorithm for determining the weights of connections. The BP algorithm is based on a learning rule by which the weights are evoluted in order to minimize the mean of squared differences between the desired and actual values of the output neurons, namely:

$$E = \frac{1}{n}\sum_{j=1}^{n}(y_{dj} - y_j)^2 \tag{7}$$

where n is the number of neurons, y_{dj} is the desired value of the output neuron j and y_j is the actual output value of that neuron. Each weight w_{ji} is adjusted by adding an increment Δw_{ji} to it. Δw_{ji} is selected to reduce E as rapidly as possible. The adjustment is carried out over several training iterations until a satisfactorily small value of E is obtained or a given number of iteration, is reached. How Δw_{ji} is computed depends on the training algorithm adopted.

The standard BP algorithm suffers from a few drawbacks such as the risk to converge in local minima and a long computational time [18-20, 23-26]. To improve performance, five different types, high performance BP training algorithms, which use different optimization techniques, were used in this study. These algorithms are Levenberg–Marquardt (LM) [23, 24], Broyden–Fletcher–Goldfarb–Shanno (BFGS) [29], Bayesian Regularization (BR) [27, 30], Conjugate Gradients (CGs) [18] and Resilient Back-propagation (RP) [18] algorithms. These algorithms are briefly explained below.

Levenberg–Marquardt (LM) method: The LM algorithm is designed to approach second-order training speed without having to compute the Hessian matrix. The method combines the best features of the Gauss-Newton method and the steepest-descent method, but avoids many of their limitations.

Broydon–Fletcher–Goldfarb–Shanno (BFGS) method: This method uses an update formula derived from the quasi-Newton update of Hessian. It requires the storage of the approximate Hessian matrix and has more computation than conjugate gradient algorithms at each iteration, but usually converges in less iteration.

Bayesian Regularization (BR) method: This method is the modification of the Levenberg-Marquardt training algorithm to produce a well-generalized network. It minimizes a linear combination of squared errors and weights. This algorithm can train any network as long as its weight, inputs, and activation functions have derivative functions.

Conjugate Gradients (CGs) method: This algorithm is a second-order method, which restricts each step direction to be conjugate to all previous step directions. This restriction simplifies the computation greatly because it is no longer necessary to store or calculate the Hessian or its inverse. There are a number of versions of CGs (Polak–Ribiere, Fletcher–Reeves, and Powell–Beale). Powell–Beale's version of CGs is used in this article.

Scaled Conjugate Gradient (SCG) method: Each of the conjugate gradient algorithms requires a line search at each iteration. This line search is computationally expensive, since it requires that the network response to all training inputs be computed several times for each search. This method combines the model-trust region approach (used in the Levenberg-Marquardt algorithm), with the conjugate gradient approach to avoid the time-consuming line search.

4. The structure of ANN based chaotic generator

The block diagram of the proposed ANN model is given in Figure 5. As shown in the figure, three initial conditions and time variable were applied to the inputs and three chaotic dynamics $\hat{x}$, $\hat{y}$ and $\hat{z}$ were obtained from the outputs of the ANN. For the training and test phases of the ANN, approximately 1800 input-output data pairs which belong to 24 different initial condition sets were obtained from Equation (4). A quarter of those 1800 data pairs were sorted to use in the test phase and the rest of data were used in the training phase.

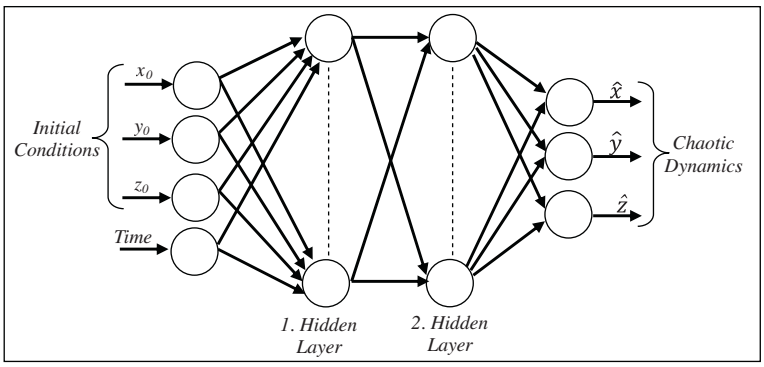

Figure 5. The block diagram of trained ANN model.

In this study, the aim is to find the best ANN structure to produce chaotic dynamics. It is achieved by changing the number of hidden layers, the number of neurons in the hidden layers and the transfer functions in the neurons. After the training process, the ANN models were tested with sorted test data and the results are given in Table 1. As seen from the table, the suitable network structure is 4x20x19x3 trained with Bayesian Regularization algorithm. This means that the number of neurons is 4 for the input layer, 20 for the first hidden layer, 19 for the second hidden layer and 3 for the output layer. The input and output layer neurons have linear activation functions and the first and the second hidden layer neurons have hyperbolic tangent sigmoid and hyperbolic logarithmic sigmoid activation functions, respectively. The normalized error convergence curves in the learning algorithms used in the analysis for 1000 epochs are graphically shown in Figure 6. As the learning proceeds, the mean square error progressively decreases and finally attains a steady state minimum value as shown in the figure. The test errors and the correlation coefficients for each of chaotic dynamics $\hat{x}$, $\hat{y}$ and $\hat{z}$are also given in Table 2. The test and training errors and the correlation coefficients were calculated using the differences between chaotic dynamics x, y, z and $\hat{x}$, $\hat{y}$, $\hat{z}$ and these dynamics with different initial values are shown in Figure 7.

Table 1. Training and test errors for ANN models trained with different algorithms.

Training Algorithm	Number of Neurons		Train Error (mse)	Test Error (mse)
	1. Hidden Layer	2. Hidden Layer		
Bayesian Regularization (BR)	20	19	0.4320	0.0567
Levenberg–Marquardt (LM)	20	20	0.5345	0.0612
Broyden–Fletcher–Goldfarb– Shanno (BFGS)	20	20	1.441	0.1218
Conjugate Gradient Backpropagation With Powell-Beale Restarts (CGB)	16	15	1.7304	0.1435
Scaled Conjugate Gradient Back propagation (SCG)	19	19	1.7544	0.1452

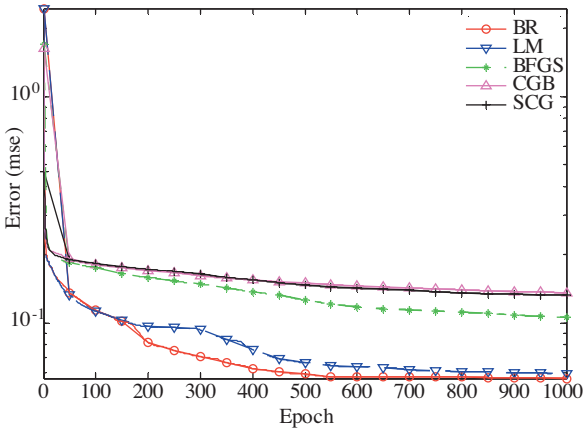

Figure 6. Learning (convergence) characteristics of the ANN for different learning algorithms used in the analysis for 1000 epoch.

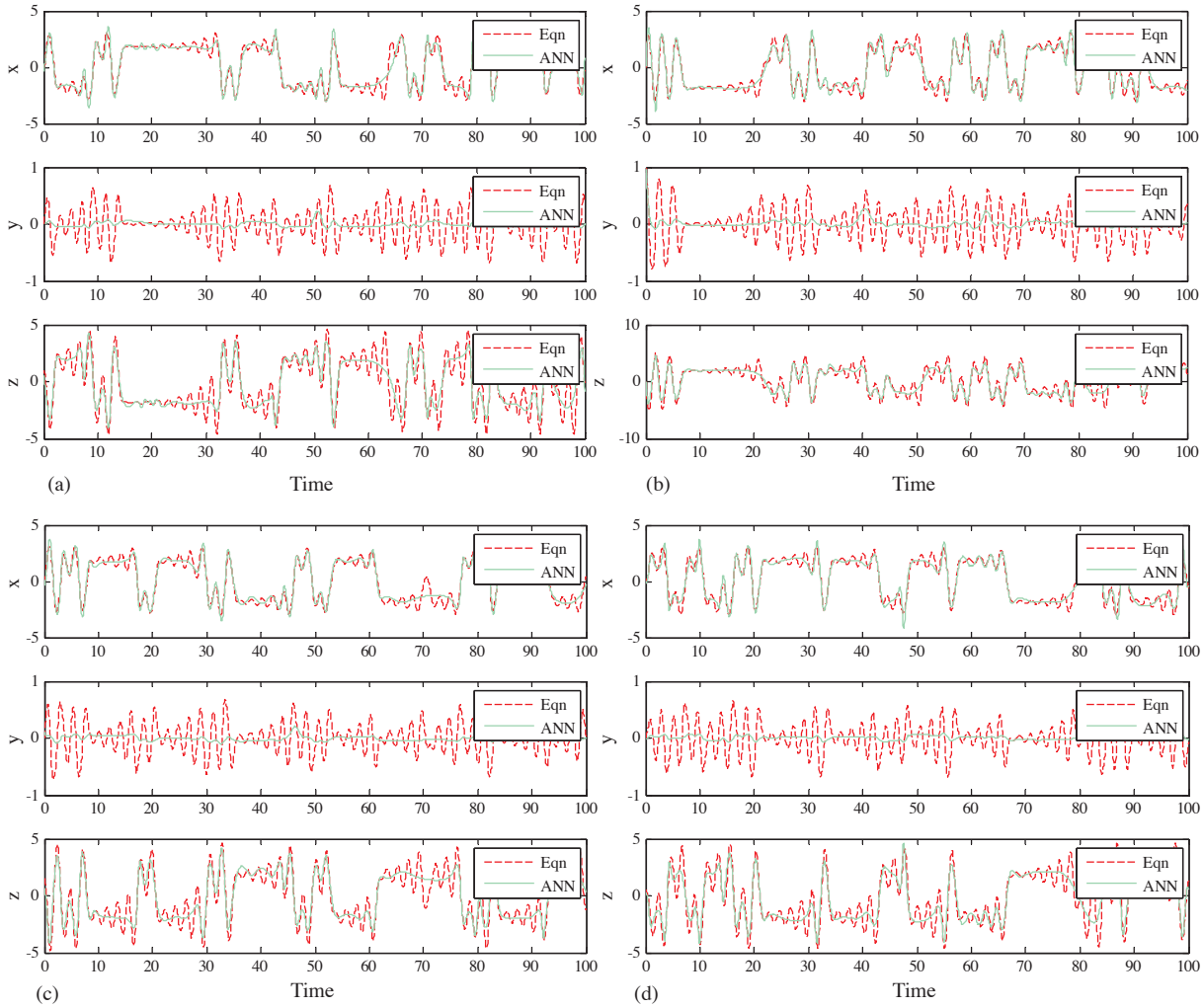

Figure 7. Three chaotic dynamics produced by both the ANN and the numerical solution of Chua's circuit for different initial condition sets.

Table 2. Correlation coefficients and test errors for each chaotic dynamics produced by the proposed ANN.

Training Algorithm	Test Error (mse)			Correlation Coefficient		
	$\hat{x}$	$\hat{y}$	$\hat{z}$	$\hat{x}$	$\hat{y}$	$\hat{z}$
Bayesian Regularization (BR)	0.2303	1.7195	0.9189	0.9632	0.2224	0.8976
Levenberg–Marquardt (LM)	0.3413	1.8243	1.1738	0.9449	0.3227	0.8671
Broyden–Fletcher–Goldfarb– Shanno (BFGS)	0.9592	2.8157	2.4834	0.8360	0.1847	0.6891
Conjugate Gradient Back propagation With Powell-Beale Restarts (CGB)	1.7067	2.8824	3.3284	0.6815	0.0595	0.5443
Scaled Conjugate Gradient Back propagation (SCG)	1.6759	2.8892	3.2990	0.6884	0.0779	0.5499

It is important to note that the selection criteria of hidden layer neuron numbers depends on some factors, such as ANN type, training set characteristics and type of application. This topic is still under special attention of artificial intelligence researchers today.

4.1. A Case Study: Encryption and Decryption Using ANN Based Chaotic Generator

Cryptology is a discipline that covers many different studies which are about overcoming security and identification holes on communication systems. Cryptography is a process which consists of two parts that are called as encryption and decryption processes. The encryption process can be defined as converting the original message which is named as plain-text to an inscrutable form which is named as cipher-text by an algorithm with secret keys. Decryption process is the inverse form of encryption process. The block diagram of a crypto-system is given in Figure 8. In this figure, a plain-text is encrypted using the secret keys in the transmitter and then the cipher-text is transmitted throughout an unsecure channel. At the receiver, the cipher-text is decrypted by the decryption algorithm using the secret keys and the plain-text is obtained again. Cryptanalysts can also monitor the unsecure channel. In general, obtaining plain-text from cipher-text without secret key has major priority. Learning the decryption algorithm, learning the key, sometimes learning that the same message is sent again, part of the key, time the message was sent are also important. Unknown decryption algorithm is the worst situation for cryptanalysts. Cryptanalysts try to decrypt all cipher-texts without the secret keys and/or

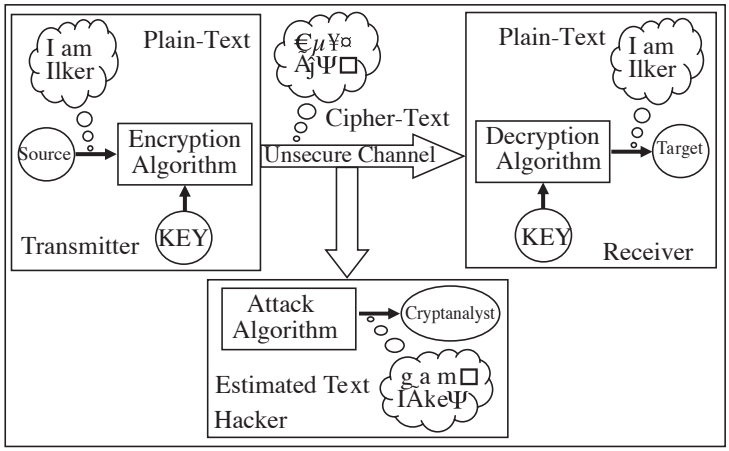

Figure 8. The block diagram of a crypto-system.

the decryption algorithm. They use different attack algorithms and achieve the estimated texts which is only similar to the plain text but not exactly the same. If a cryptanalyst achieves the plain-text without having the secret keys and/or the decryption algorithm, it is easily said that the crypto-system is failed [1-3].

As a case study, a plain-text was encrypted and then obtained cipher-text was decrypted by using the chaotic dynamics both z and $\hat{z}$. It is accepted that the initial conditions which were used in the training phase of the ANN model and the system parameters are known by both the transmitter and the receiver. The block diagram of the process is given in Figure 9.

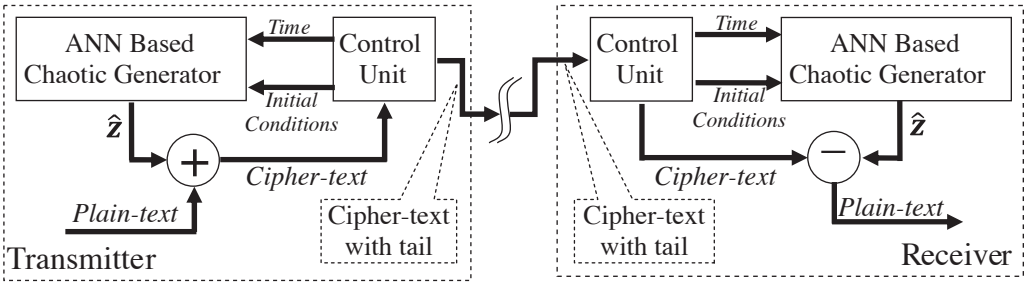

Figure 9. The block diagram of ANN based chaotic crypto-system.

In the transmitter, software based control unit decides the values of initial conditions and the ANN based chaotic generator produces the outputs according to those initial conditions. A gray scaled picture was chosen as a plain-text. A gray scaled picture, in MATLAB, is in matrix form and has pixels which have integer values and vary between 0 and 255. The chaotic dynamic $\hat{z}$, shown in Figure 7(a), was chosen to encrypt the plain-text. Because of varying the amplitude of the chaotic dynamic $\hat{z}$ between -5 and +5, the gray scaled picture was converted to a column vector and normalized in that interval. The cipher-text is obtained by adding normalized data to the chaotic dynamic $\hat{z}$. A tail which consists of initial conditions, image dimensions, and normalization parameters, is tied to the cipher-text by the control unit and then the cipher-text with tail is transmitted. In the decryption phase, the tail is firstly discriminated from the cipher-text and the initial conditions are extracted from the tail by the control unit. Then, the initial conditions are applied to the ANN based chaotic generator and the chaotic dynamic $\hat{z}$ is produced by the generator. Finally the plain-text is obtained by subtracting the chaotic dynamic $\hat{z}$ from the cipher-text. The size of cipher-text is equal to the size of plain-text. But tying the tail to the cipher-text enlarges the size of encrypted data. The size of tail is only 8 bytes.

To make a comparison between the ANN based chaotic generator and the numerical solution of Chua' s circuit, the same operations are also executed for the chaotic dynamic z instead of $\hat{z}$ to encrypt and decrypt a plain-text.

The image, chosen as plain-text, is shown in Figure 10. This image was encrypted using the chaotic dynamics z and $\hat{z}$ and obtained cipher-texts are shown in Figure 11(a) and 11(b) respectively. Those cipher-texts are in scalar vectors and can be sent to receiver in these forms. If a cryptanalyst knows that the scalar vectors in channel are images and converts those scalar vectors into image forms, he can only see black blocks as shown in Figure 11(c) and 11(d). As mentioned before, the normalized image was encrypted with the chaotic dynamics z and $\hat{z}$. If a cryptanalyst knows the normalization parameters and then denormalizes the black blocks, he can obtain the images which were encrypted using the chaotic dynamics z and $\hat{z}$, that are shown in Figure 11(e) and 11(f) respectively. The images encrypted and then decrypted using the chaotic dynamics z and $\hat{z}$ are the same and they are given in Figure 11(g) and 11(h).

Figure 10. The image chosen as plain-text.

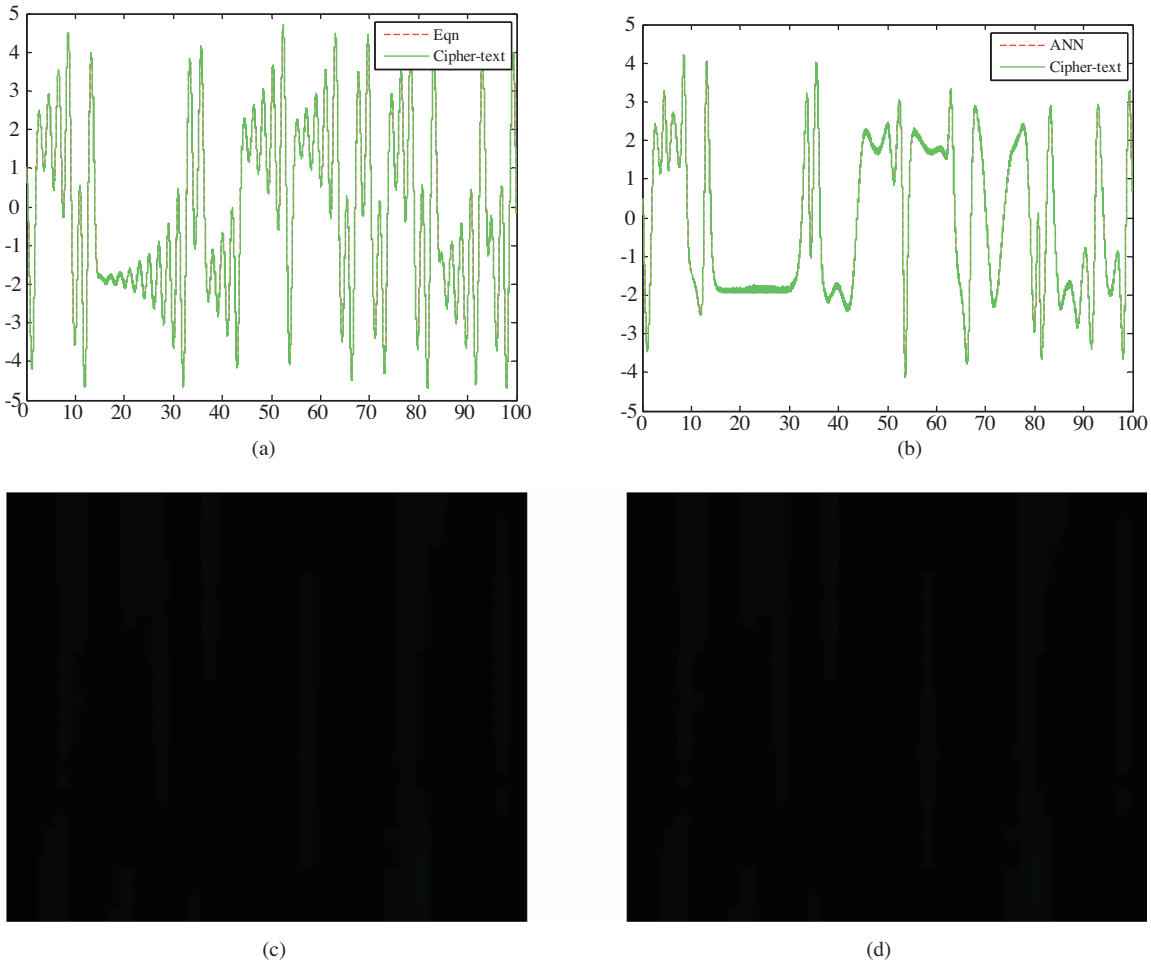

Figure 11. Cipher-texts encrypted with the chaotic dynamics (a) z and (b) $\hat{z}$. The cipher-texts in image forms encrypted with the chaotic dynamics (c) z and (d) $\hat{z}$. Denormalized cipher-texts encrypted with the chaotic dynamics (e) z and (f) $\hat{z}$. Images, decrypted with the chaotic dynamics (g) z and (h) $\hat{z}$.

(e) (f)

(g) (h)

Figure 11. Contunied.

Let's suppose that a cryptanalyst determines that the encryption and the decryption processes are based on chaos and illegally obtains the initial condition set which was used in the encryption process. In this case, even if he produces the chaotic dynamic z, using the numerical solution of Chua's circuit with that initial condition set and decrypts the cipher-text which was encrypted with the $\hat{z}$ given in Figure 11(b), he can obtain an estimated-text given in Figure 12.

5. Results and discussion

In analogue chaotic circuits, it is too difficult to regenerate the same chaotic dynamics at different times due to the fact that the same initial conditions cannot be intentionally defined. Therefore, to produce the same chaotic dynamics, analogue chaotic circuits have to synchronize with each other. This is a serious problem for cryptology. The numerical solution of an analogue chaotic circuit for regenerating the same chaotic dynamics at different times can be offered as an alternative way to overcome the disadvantage of analogue chaotic circuits. Although the same initial conditions can be intentionally defined by using the numerical solution of an analogue chaotic circuit, another problem occurs. That is named as fewness of system parameters. The system parameters

of the numerical solution of Chua's circuit are α, β, a, b, c and three initial condition values, x_0, y_0 and z_0 as seen from Equation (4) and (5).

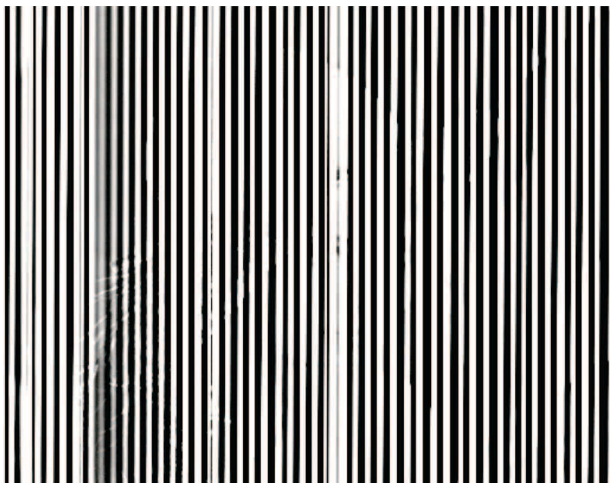

Figure 12. The estimated-text.

In this study, different chaotic dynamics were produced by using the numerical solution of Chua's circuit for different initial conditions to train the proposed ANN. Different ANN structures which were obtained by changing the number of hidden layers, the number of neurons in the hidden layers and the transfer functions in neurons were trained and tested with different learning algorithms, to obtain better performance and faster convergence with simpler structure. Five learning algorithms, BR, LM, BFGS, CGB, and SCG exhibited better performance among them. Figure 6 clearly shows that the minimum training error belongs to BR algorithm. As it is clearly seen from Figure 6, the next solution which is closer to BR is obtained by LM algorithm. Among the neural models presented here, the worst result is obtained using the SCG algorithm for this particular application. Table 1 shows the training and test errors from the complete learning algorithms used in the analysis for the different network configuration mentioned above. The training and test mean square errors of the network, trained with BR, are 0.0567 and 0.4320 respectively.

In the test phase, it is expected that ANN should produce the outputs with minimum error. Theoretically, the test error and the correlation coefficients should be 0 and 1 respectively. However, the test errors are greater than 0 and the correlation coefficients are less than 1 in practice. The test errors and the correlation coefficients are given in Table 2 for each chaotic dynamic. Figure 7 shows that, although ANN can not sufficiently learn the chaotic dynamic y, it exhibits a good performance for the chaotic dynamics x and z.

A gray scaled image was encrypted and decrypted using the chaotic dynamic, z and $\hat{z}$. The results of encryption and decryption processes are given in Figure 11. After the decryption process, the image is obtained without any deformation. The correlation coefficients related to encrypted and decrypted images using z-z, $\hat{z}-\hat{z}$, $z-\hat{z}$ and $\hat{z}-z$ pairs are given in Table 3. The smaller correlation coefficient between two data is, the less similarity is. So it can be said that the first two rows of the table correspond to the expectations and the ANN based chaotic generator is as successful as the numerical solution of Chua's circuit in both encryption and decryption processes. On the other hand, the last two rows of the table show that the chaotic dynamics z and $\hat{z}$ cannot be used instead of each other to decrypt the cipher-text. The situation, in which a plain-text was encrypted using the chaotic dynamic $\hat{z}$ and decrypted using the chaotic dynamic z, is verified by Figure 12.

Table 3. Correlation coefficients of encrypted and decrypted images.

	Correlation Coefficient Between Plain-text and Cipher-text	Correlation Coefficient Between Original Plain-text and Decrypted Plain-text
Encrypted using z and decrypted using z	0.1416	1.0000
Encrypted using $\hat{z}$ and decrypted using $\hat{z}$	0.1620	1.0000
Encrypted using z and decrypted using $\hat{z}$	0.1416	0.0434
Encrypted using $\hat{z}$ and decrypted using z	0.1620	0.0384

The test process of the ANN takes only a few seconds after finding the most suitable network configuration and the learning algorithm. The real-time calculation time is in the order of microseconds. Therefore the neural model is very fast after being trained and it requires no complicated mathematical functions. As a comparison, a performance test between the proposed ANN model and the numerical solution of Chua's circuit was performed in MATLAB on a server with AMD Opteron Dual Core 2216HE 3.01HGz CPU and 7.2 GB RAM. Elapsed times of encryption and decryption processes were measured for different sizes of a gray scaled picture. The results are given in Table 4. As seen from this table, the proposed ANN model produces the outputs faster than the numerical solution of Chua's circuit.

Table 4. The encryption and decryption times of both the ANN and the numerical solution of Chua's circuit.

File Size (in pixels) Plain-Text	File Size (KB)		Encryption Time (seconds)		Decryption Time (seconds)	
	Plain-Text	Cipher-Text with tail	Chua's Circuit	ANN	Chua's Circuit	ANN
64x64	65	$\simeq$ 65	1.109	0.453	1.263	0.577
128x128	257	$\simeq$ 257	1.988	0.531	2.217	0.709
256x256	1025	$\simeq$ 1025	4.668	1.631	5.020	2.068
512x512	4097	$\simeq$ 4097	16.385	4.535	18.103	6.104

Conclusion

In this paper, an ANN based chaotic generator is proposed to overcome the weaknesses of chaotic cryptosystems. The ANN model, presented in this work, was trained in MATLAB to generate the chaotic dynamics which are similar to the chaotic dynamics generated by the numerical solution of Chua's circuit. It is found that the 4x20x19x3 multilayer perceptron network, trained with Bayesian Regularization (BR) algorithm, is the most suitable structure to produce chaotic dynamics with minimum error. If a cryptanalyst wants to generate chaotic dynamics which are the same as the ANN model, he must exactly know the ANN structure, weights and biases. In other words, to produce same chaotic dynamics a cryptanalyst must determine the number of neurons, transfer functions in each neuron, the values of 517 weights and 42 biases. Also, the proposed ANN model does not have any synchronization problem. In addition, the difference between the chaotic dynamics z and $\hat{z}$ can be considered as an advantage of the ANN based chaotic generator. Eventually the major weaknesses of analogue circuit and the numerical solution of chaotic circuit are eliminated with the proposed model.

References

[1] A.J. Menezes, P.C. Van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, New York, CRC Press, 1996.

[2] B. Schneier, Applied Cryptography, New York, John Wiley & Sons Inc., 1996.

[3] H.C.A. Van Tilborg, Fundamentals of Cryptology, New York, Kluwer Academic Publishers, 2000.

[4] T. Chien, T. Liao, "Design of secure digital communication systems using chaotic modulation, cryptography and chaotic synchronization", Chaos, Solitons and Fractals, Vol. 24, pp. 241–255, 2005.

[5] R. He, P.G. Vaidya, "Implementation of chaotic cryptography with chaotic synchronization", Physical Review E, Vol. 57(2), pp. 1532-1535, 1998.

[6] H. Gao, Y. Zhang, S. Liang, D. Li, "A new chaotic algorithm for image encryption", Chaos, Solitons and Fractals, Vol. 29, pp. 393–399, 2006.

[7] O. Morgul, E. Solak, M. Akgul, "Observer based chaotic message transmission", Int. J. Bifurcation Chaos, Vol. 13(4), pp. 1003-1017, 2003.

[8] A.A. Pacha, N. Hadj-Said, B. Belmeki, A. Belgoraf, "Chaotic behavior for the secrete key of cryptographic system", Chaos, Solitons and Fractals, Vol. 23, pp. 1549–1552, 2005.

[9] N.K. Pareek, V. Patidar, K.K. Sud, "Short communication cryptography using multiple one-dimensional chaotic maps", Communications in Nonlinear Science and Numerical Simulation, Vol. 10, pp. 715–723, 2005.

[10] L.M. Pecora, T.L. Carroll, "Synchronization in chaotic systems", Phys. Rev. Lett., Vol. 64, pp. 821-824, 1990.

[11] I.N. Stewart, God Does It Play With the Dice? The New Mathematics of Chaos, London, Penguin, 1997.

[12] E. Bilotta, P. Pantano, F. Stranges, "A gallery of Chua attractors Part-I", Int. J. Bifurcation Chaos, Vol. 17(1), pp. 1-60, 2007.

[13] L. Kocarev, K. Halle, K. Eckert, L. Chua, "Experimental demonstration of secure communications via chaotic synchronization", Int. J. Bifurcation Chaos, Vol. 2, pp. 709-713, 1992.

[14] Y. Z. Yin, "Experimental demonstration of chaotic synchronization in the modified Chua' s oscillators", Int. J. Bifurcation Chaos, Vol. 7(6), pp. 1401-1410, 1997.

[15] N.K. Pareek, V. Patidar K. K. Sud, "Discrete chaotic cryptography using external key", Physics Letters A, Vol. 309, pp. 75–82, 2003.

[16] J. Kawata, Y. Nishio, H. Dedieu, A. Ushida, "Performance Comparison of Communication Systems Using Chaos Synchronization", IEICE Transactions on Fundamentals, Vol. E82-A(7), pp. 1322-1328, 1999.

[17] G. Kolumban, J. Schweizer, J. Ennitis, H. Dedieu, B. Vizvari, "Performance evaluation and comparison of chaos communication schemes", 4th International Workshop on Nonlinear Dynamics of Electronic Systems (NDES'96), Vol. 1, pp.105-110, 1996.

[18] M.T. Hagan, H.B. Demuth, M. Beale, Neural Network Design, Boston, PWS Publishing Company, 1995.

[19] A. Maren, C. Harston, R. Pap, Handbook of Neural Computing Applications, New York, Academic Press, 1990.

[20] P. Vas, Artificial Intelligence Based Electrical Machines and Drivers, New York, Oxford University Press, 1999.

[21] T. Matsumoto, L.O. Chua, K. Ayaki, "Reality of chaos in the double scroll circuit: a computer assisted proof, IEEE Trans. Circuit Syst., Vol. 35, pp. 909-925, 1988.

[22] L.O. Chua, C.W. Wu, "A universal circuit for studying and generating chaos", IEEE Trans. on Circuits and Sys.-I: Fundamental Theory and Applications, Vol. 40, pp. 732-745, 1993.

[23] M.T. Hagan, M. Menhaj, "Training feedforward networks with the marquardt algorithm", IEEE Neural Networks, Vol. 5(6), pp. 989–993, 1994.

[24] F.M. Ham, I. Kostanic, Principles of Neurocomputing for Science and Engineering, New York, Mc-Graw Hill, 2001.

[25] R. Rojas, Neural Networks A Systematic Introduction, New York, Springer Verlag, 1996.

[26] S. Haykin, Neural Networks: A Comprehensive Foundation, Second Ed., New Jersey, Prentice Hall, 1999.

[27] D.J.C. MacKay, "Bayesian interpolation", Neural Comput., Vol. 4, pp. 415–447, 1992.

[28] K. Danisman, I. Dalkiran, F.V. Celebi, "Design of a high precision temperature measurement system based on artificial neural network for different thermocouple types, Measurement, Vol. 39, pp. 695–700, 2006.

[29] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, New Jersey, Prentice-Hall, 1983.

[30] F.D. Foresee, M.T. Hagan, "Gauss–Newton approximation to Bayesian regularization, Proc. of Int. Conference on Neural Networks ICNN'97, pp. 1930–1935, 1997.