Asian Journal of Electrical Sciences
ISSN: 2249-6297 (P) Vol.7, No.2, 2018, pp.19-23

DOI: https://doi.org/10.51983/ajes-2018.7.2.2281

Application of Neural Network Based Data Security in to LFC of a Two Area Power System

T. Rathimala[1], R. Francis[2] and M. Kamarasan[3]
[1&3]Assistant Professor, Department of Computer and Information Science,
Annamalai University, Annamalai Nagar, Tamil Nadu, India
[2]Assistant Professor, Government Engineering College, Bargur, Tamil Nadu, India
E-Mail: rathimalat@gmail.com, smkrasan@yahoo.com, francis.electrical@gmail.com
(Received 20 June 2018; Revised 19 July 2018; Accepted 3 August 2018; Available online 9 August 2018)

Abstract **- The paper is concentrated with the study of design of a data security system based on neural networks. This data security method added Load-Frequency Control of reheat interconnected two area power systems problems with non-linearity. The neural network control is incorporated to load-frequency control in power systems. Elman Recurrent neural network is involves forecasting controller and system's output. The system was simulated and the frequency variations in area 1 and area 2 and tie-line power variations for 1% step-load disturbance in area 1 were obtained. The comparison due to frequency variations and tie-line power deviations for the two area interconnected thermal power system. The result data of different keys were taken as test data, encrypted, decrypted and compared with the original data. The results have ensured better its advantages over conventional techniques.**
Keywords: **Back Propagation Neural Networks, Data Security, Cryptography, Load-Frequency Control, Integral Controller**

I. INTRODUCTION

We start of the computer research, the automated tools for secure files and other important information stored there on. The general tools design to protect data and to hackers is computer security [7]. Based on this, the optimalised controller is designed; Load-frequency control is one of the major expectations providing flexible and good operation in multi-area power systems. In large power systems, differentiations in frequency carry over to serious large scale stability problems for good operation, constant frequency and active power balance must be made. The changing frequency is a common basis, any disturbance in active power demand or generation reflected throughout the system by a varying in frequency [8, 9]. The design of LFC system is control the power-generation and active-power at the tie lines. In an LFC, proportional-integral (PI) controllers are generally used, but it is not easy to obtain gain of the PI controller [10]. The inputs of NN controller added dynamic power system is that state variables and disturbance vector provided. Back propagation algorithm has been developed to the continuous-time dynamics as system learning rule. By far the most important automated tool for network and communications security is encryption. Two major techniques used in encryption are symmetric and asymmetric encryption. In symmetric encryption, two parties share a single encryption-decryption key [11]. The encrypts of the sender has original message (P), which is addressed to as plain-text, using a key (K) to generate apparently random nonsense, addressed to as cipher-text (C),

$$C = \text{Encrypt}\ (K, P). \tag{1}$$

Once the cipher-text generated, it may be transferred. Upon receipt, the cipher text can be transferred back to the original plain text by applying a decryption algorithm and the same key that was used for encryption, which can be expressed as follows:

$$P = \text{Decrypt}\ (K, C). \tag{2}$$

In an asymmetric encryption, two keys are used; one key is encryption and another paired key for decryption. One of the keys is private by the key pair generated by party and the other is made public. The encryption is that the message encrypted with the public key. It is the corresponding private key only decrypted. In this paper we present an encryption system based on neural networks (NNs). A neural network is used to make a highly-efficiency encryption by a permanently change the key. The generic NN algorithm is an important problem, as it consists on the application and design of the system. We have used a multi-layer topology. In present paper, General Regression-Neural Network (GRNN), a simple, one-parameter neural network model, are presented for the encryption-and-decryption process. Neural networks said to a very powerful and frame-work for presenting non-linear mapping from several input variables to several output variables. The process to determining the values of these parameters on the a data set is referred to as learning or training, the data set is generally referred to as a training set. A neural network can suitable selection for the functional forms used for encryption and decryption operations.

II. MODELING OF TWO AREA INTERCONNECTED POWER SYSTEM

The mathematical modeling of the two area power system, is given by the below set of the state variable second order differential equations as

$$\Delta F_1(s) = \frac{Kps_1}{1 + sTps_1}\left[\Delta Pg_1(s) + Kc_1.\Delta Pc_1(s) - \Delta Pd_1(s) - \Delta Ptie(s)\right] \tag{2.1}$$

$$\Delta Pg_1(s) = \frac{1+sKr_1Tr_1}{1+sTr_1}\Delta Pt_1'(s) \quad (2.2)$$

$$\Delta Pt_1'(s) = \frac{1}{1+sTt_1}\Delta Xe_1(s) \quad (2.3)$$

$$\Delta Xe_1(s) = \frac{1}{1+sTg_1}\left[\Delta Pc_1(s) - \frac{1}{R_1}\Delta F_1(s)\right] \quad (2.4)$$

$$\Delta Ptie(s) = \frac{2\pi.T_{12}}{s}\left[\Delta F_1(s) - \Delta F_2(s)\right] \quad (2.5)$$

$$\Delta F_2(s) = \frac{Kps_2}{1+sTps_2}\left[\Delta Pg_2(s) + Kc_2.\Delta Pc_2(s) - \Delta Pd_2(s) - a_{12}.\Delta Ptie(s)\right] \quad (2.6)$$

$$\Delta Pg_2(s) = \frac{1+sKr_2Tr_2}{1+sTr_2}\Delta Pt_2'(s) \quad (2.7)$$

$$\Delta Pt_2'(s) = \frac{1}{1+sTt_2}\Delta Xe_2(s) \quad (2.8)$$

$$\Delta Xe_2(s) = \frac{1}{1+sTg_2}\left[\Delta Pc_2(s) - \frac{1}{R_2}\Delta F_2(s)\right] \quad (2.9)$$

The system state space differential equations are

$$\dot{X} = Ax + Bu + \Gamma d \quad (2.10)$$

where, x, u and d are the state, control and disturbance vectors.

System Control input vector

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \Delta P_{c1} \\ \Delta P_{c2} \end{bmatrix} \quad (2.11)$$

Disturbance vector $d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} \Delta P_{D1} \\ \Delta P_{D2} \end{bmatrix}$ (2.12)

Where

Augmented system matrix $\overline{A} = \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix}$ (2.13)

Augmented control input matrix $\overline{B} = \begin{bmatrix} 0 & B \end{bmatrix}$ (2.14)

Augmented disturbance matrix $\overline{\Gamma} = \begin{bmatrix} 0 & \Gamma \end{bmatrix}$ (2.15)

Augmented output matrix $\overline{C} = \begin{bmatrix} 0 & C \end{bmatrix}$ (2.16)

The Two state vectors $\int ACE_1$ and $\int ACE_2$ are in combined augmented form.

$$\int ACE_i = \int \beta_i \Delta f_i + \Delta P_{tie} \qquad i=1, 2 \quad (2.17)$$

Substituting the value we get,

$$\begin{bmatrix} \bar{Y} \\ \dot{\bar{X}} \end{bmatrix} = \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix}\begin{bmatrix} \int ACE.dt \\ \bar{X} \end{bmatrix} + \begin{bmatrix} 0 \\ B \end{bmatrix} U \quad (2.18)$$

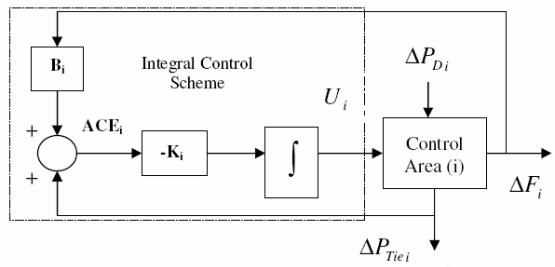

Fig.1 Conventional PI Control Scheme

III. THE NEURAL NETWORK CONTROL SCHEME

In the control scheme, neural network is selected to make the real-time dynamic model of the power system. This connection with the present controller output u(r), the tie-line power variation dptie(r) and the frequency deviation df(r), the neural network scheme is used to predict the next stage's frequency deviation df(r+1), thus calculate the ACE, Elman network is a typical dynamic recurrent neural network. Its feedback consists of a group of connected modules and is used to record the implicit memory. Meanwhile, the feedback, along with the network input, acts as the import to hidden units in the next moment. This is used to recurrent-neural network with dynamic-memory and thus the capacity to predict future output, which is quite fitful to power system load-frequency control.Fig.4 The Elman neural network basic structure in the Automatic generation control. The network structure is shown in Fig.1 α (0≤α≤1) is the feedback link gain. The external inputs to the network controller output u (r) €R, the tie-line power deviation dPtie (r) €R and frequency deviation df (r) €R. The network output is the predicted frequency deviation for the next moment df (r+1) €R, in which r is the sampling instant. Let the hidden layer output be x (r+1) € R5, then

$$X(r+1) = f(W\,x(r) + W\,u(r) + W\,dPtie(r) + W\,df(r)) \quad (2.31)$$

$$x(r) = x(r) + \alpha x(r-1) \quad (2.32)$$

$$df(r+1) = g(W\,x(r+1)) \quad (2.33)$$

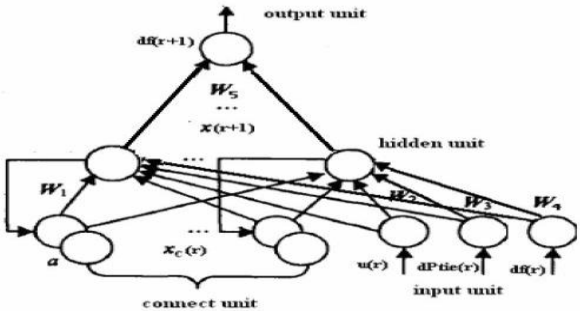

Fig. 2 The Elman-Neural Network diagram in the Automatic Generation Control

WhereW1, W2, W3, W4 and W5 are the hidden units of weight matrix routed to the hidden units, input units to hidden units and hidden units to the output unit respectively. f (•) and g (•) are the non-linear vector function of the hidden layer neural cell with activation function of and output layer neural unit-cell; xc (r+1) represents the state at r+1 moment. Here, x(r+1) is the total state of the power system dynamic.

IV. THE CONTROL ALGORITHM

The control algorithm based on neural network scheme predictive method can be summarized as follows

1. Set the initial values of the desired frequency deviation df(r), desired ACE(r) and desired ACEN(r) to 0.

2. Forecast the frequency deviation df(r+1) at the (r+1) moment using recurrent neural network as shown in Fig. 1, resulting the forecasting of ACE(r+1).

V. DESIGN OF THE PROPOSED SYSTEM

The encryption function has the following 3 sub-functions, explained in the following sub-sections

1. The keys are created.
2. The input message is broken into blocks equal to the number of keys and processed, one block at a time, as input to the neural network.
3. The neural network (GRNN) is used to encrypt each block of data producing the encrypted message.

A. Creating the Keys

Selection of a fine increasing Knapsack and it to be encrypt the data. The number of keys selected equal to *N*, so that:

1. The sum of the numbers (keys) must be less than 2^x, where $x=2^N$, and
2. The numbers cannot be equal ($K_1 \neq K_2 \neq \ldots. \neq = K_n \neq 0$). To keep our processing is simple, we have select objects of the using numbers: 27, 14, 68, so that K_1 =68, K_2 =14 and K_3 =27 will be our keys.

B. Breaking Data of the Input

Suppose the M is a set of N-bit initial unipolar data [4],

$$i.e.: Mi = \{0, 1\}, 0 \leq I \leq N-1. \quad (3)$$

In the current model, a 3-bit plain text were input (N = 3) and an 8-bit cipher text were output (2^N). Now, we need expect to encrypt, for example 011010110. First, we break it down into blocks (N =3) the length of our Knapsack, thus: 011 010 110. In the first block (011), this is 1's in the second and third places. Thus, we take the second and third keys are adding them to each other: 0+14+68=82. Repeating the procedure for all the input data blocks give results the encrypted of 01001000 as 82, as seen bellow Table I. The encrypted transmitted to the receivers.

TABLE I TRAINING DATA FULL PATTERN

Training Input			**Output**							
P_3	P_2	P_1	C1	C2	C3	C4	C5	C6	C7	C8
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0
0	1	0	0	0	0	0	1	1	1	0
0	1	1	0	1	0	1	0	0	1	0
1	0	0	0	0	0	1	1	0	1	1
1	0	1	0	1	0	1	1	1	1	1
1	1	0	0	0	1	0	1	0	0	1
1	1	1	0	1	1	0	1	1	0	1

C. The Proposed Neural Network-Based Encryption System

Neural network is a global network. A neural network is a Input layers, hidden layers, output layers structure (artificial neurons). Each layer has many classifications. This method involved supervised method and unsupervised methods In this paper, Backprobagation neural network (Bpnn), a simple, one-parameter neural network model, is proposed for encryption and decryption. Backprobagation neural network (Bpnn) was developed by Specht [6] and is a simple yet very effective local approximation based on a neural network, in the sense of estimating a probability distribution function. The main advantages of Bpnn are as follows

1. Fast learning.
2. Convergence to backprobagation surface for large numbers of samples.
3. Effective use with sparse data.
4. The capacity to handle non-stationary data.

Bpnn uses a standard statistical formula for calculating the conditional mean *Y* of the scalar random variable *y* given a measurement *X* of a vector random variable *x*. The vector random variable *x* corresponds to the input of the network and the random variable *y* corresponds to the network's output. Apart from being used as a static regression technique, GRNN can be used when the data statistics changes over time, is the use derived recently by specifying a time constant and a threshold. In order to build a Bpnn

1. Set the number of input, pattern, and output layer (Processing Elements, or PEs).
2. Choose the pattern unit.
3. Choose the time constant and the reset factor.
4. Set the radius of influence.

This is a simple clustering mechanism which assigns an input vector to a cluster if the cluster center is the cluster center nearest to the input vector AND closer than the radius of influence. Otherwise, the input vector is assigned as the center of a new cluster (if possible). The implementation of RNN allows an exponentially decaying _ of the following form.

D. Building and Training the Neural Network

The problem we face is a computational problem, so we will use a multi-layer GRNN. The Bpnn used for encryption consists of three layers. Each layer consists of a number of neurons, depending on the case to be solved. In the encryption process, the input message is divided into 3-bit data sets, while 8-bit sets are produced after the encryption process. The basic GRNN architecture is shown in Figure 1. Each layer consists of the following.

1. An input layer consisting of 3 nodes, which represents the *N*-bit blocks.
2. A pattern layer of 8 nodes.
3. An output layer of 8 nodes, used to define the decrypted output message.

The other parameters of the GRNN are as follows
1. Time constant=1000.0.
2. Reset Factor=0.000.
3. Radius of influence=0.050.
4. Scale=1.000.
5. *E*xponent=0.500.

A value of 0.5 for *E* is suitable under most circumstances. A value of 0.0 for *E* allows to use a constant value for _ (equal to 5).

In order to study the behavior of the NNs, two training sets are used
1. A full pattern of inputs, which consists of all the possible inputs, or
2. A half of the pattern mentioned above.

E. Data Security Based On Neural Networks

At the beginning of learning the encryption key, was GRNN fed with the valid states as shown in Table I for the full pattern and Table II for half the input pattern.

The training set is repeatedly presented to the network and the weight values are adjusted until the overall error is below a re-determined level of tolerance. After the weight matrix is constructed, the network is tested for encryption.

TABLE II THE HALF-FULL PATTERN OF TRAINING DATA SETS

Training Input			**Output**							
P_3	P_2	P_1	C1	C2	C3	C4	C5	C6	C7	C8
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0
0	1	0	0	0	0	0	1	1	1	0
0	1	1	0	1	0	1	0	0	1	0
1	0	0	0	0	0	1	1	0	1	1
1	0	1	0	1	0	1	1	1	1	1
1	1	0	0	0	1	0	1	0	0	1
1	1	1	0	1	1	0	1	1	0	1

VI. RESULTS AND DISCUSSION

In order to evaluate the discussed mechanism, the encryption steps of typical digital data are shown below. We have tested the behavior of the neural network described in the previous section, and found that: 1. the neural network works reliably when using the full pattern and absolutely no errors are found in the outputs, as shown in Table III. 2. The network's performance is poorer when using a half or other part of the full input patterns. Due to numerous errors found in the outputs, the network fails to encrypt the input data, as shown in Table IV.

TABLE III THE ENCRYPTION RESULTS (FULL PATTERN OF INPUTS)

Test Input Data			**Output**							
P_3	P_2	P_1	C1	C2	C3	C4	C5	C6	C7	C8
0	0	0	0.000000	0.018002	0.000323	0.017730	0.035752	0.035062	0.035107	0.018053
0	0	1	0.000000	0.000000	0.982032	0.000324	0.017732	0.017732	0.035752	0.964971
0	1	0	0.000000	0.982031	0.000323	0.017733	0.035753	0.964971	0.018050	0.018055
0	1	1	0.000000	0.018002	0.017732	00.00032	0.982373	0.018055	0.947277	0.018055
1	0	0	0.000000	0.982030	0.017732	0.982031	0.017733	0.052370	0.018055	0.018054
1	0	1	0.000000	0.018002	0.017697	0.964385	0.982405	0.018313	0.964702	0.982084
1	1	0	0.000000	0.982030	0.017697	0.964384	0.982405	0.981720	0.964390	0.982084
1	1	1	0.000000	0.018003	0.964453	0.017627	0.999676	0.034962	0.034905	0.982084

TABLE IV THE ENCRYPTION RESULTS (HALF-FULL PATTERN OF INPUTS)

Test Input Data			**Outputs**							
*P*3	P2	P1	*C*1	*C*2	*C*3	*C*4	*C*5	*C*6	*C*7	*C*8
0	0	0	0.000000	0.000000	0.072323	0.197732	0.465575	0.195062	0.393109	0.269805
0	1	0	0.000000	0.000000	0.197733	0.072324	0.802373	0.534597	0.606279	0.269057
1	0	0	0.000000	0.000000	0.196862	0.534438	0.803407	0.072231	0.607703	0.731084
1	1	0	0.000000	0.000000	0.534453	0.196629	0.927678	0.196496	0.392907	0.731084

The remaining of next test has been made to research of the number of hidden units on the model's turned to convergence of problem. It provides the differentiation of errors as a function of neuron values in the hidden encryption layer. The errors reduce rapidly to zero at 8 hidden neurons, which represents that the neurons in the hidden layer must be equal to output-layer neurons. During training period error change for the encryption at the receiver point, the decryption process will reverse of the encryption process.

In order to decrypt the cipher data correctly,

1. The receiver must use the same key numbers to decrypt the data
2. The data reached its intended receiver, as only the receiver knows the correct key numbers necessary to remove the encryption.

The message must have been authentic, because only the sender has the key numbers needed to encrypt the message so that receiver's key numbers will decrypt it correctly.

VII. CONCLUSION

In this paper, neural network is proposed to model the load frequency control of a two-area power system. The control strategy is tested under load perturbation. The convergence of program results is better controller performance with neural controller. Then, it was try to design an encryption system based on neural networks of the Bpnn method is different to the secret keys. The proposed NN has been tested for various numbers of training iterations and for different numbers of hidden neurons, input data. The simulation results have shown a very good result, with relatively better performance than the traditional encryption methods.

APPENDIX

Data for the interconnected two area thermal power system [2, 10]
Rating of each area=2000 MW
Base power=2000 MVA
f = 60 Hz
R_1=R_2= 2.4 Hz/pu MW
T_{g1} = T_{g2}=0.08 sec
T_{t1} = T_{t2}=0.3 sec.
T_{p1} =T_{p2}= 20 sec
K_{p1} = $K_{p\ 2}$=120 Hz/pu MW
B_1 =B_2= 0.425 pu MW/Hz
T_{12} = 0.545 MW/Hz
ΔP_{d1}=0.01pu MW/HZ
a_{12} = -1.
Krfb=1.8.
Tdi=0.
Tri=0.
Kri=0.

REFERENCES

[1] W. Stallings, *Data and Computer Communications*, Prentice Hall of India, 2002.
[2] A. Tanenbaum, *Computer Networks*, Prentice Hall International, Inc, 1996.
[3] W. Stallings, *Cryptography and Network Security Principles and Practice*, Pearson Edition Asia, 2002.
[4] R. Hossein, M. Anoloni and M. Samee, *Neural Network in Network Security, ACIT 2003*, Proceeding, pp. 274-281, 2004.
[5] R. Schalkoff, *Artificial Neural Networks*, McGraw-Hill, 1999.
[6] D. F. Specht, *IEEE Trans. Neural Networks*, Vol. 2, No. 6, pp. 568, 1991.
[7] H. Shayeghi, H.A. Shayanfar and A. Jalili, "Load frequency control strategies state of the art survey for the researcher", *Energy conservation and Management*, Vol. 50, No. 2, pp. 344-353, 2009.
[8] S. Velusami and I. A. Chidambaram, "Decentralized biased dual mode controller for LFC of interconnected power systems considering GDB and GRC nonlinearities", *Energy Conversion and Management*, Vol. 48, No. 1, pp. 1691-1702, 2007.
[9] F. Beau fays, Y. Abdel-Magid and B. Widrow, "Application of neural networks to load frequency control in power systems", *IEEE Transaction on Neural Networks*, Vol. 7, No. 1, pp. 183-194, 1994.
[10] Hadi saadat, *Power System Analysis*, Tata McGraw-Hill edition, 2002.
[11] C. S. Chang and W. Fu, "Area load frequency control using fuzzy gain scheduling of PI controllers", *Electric Power System Research*, Vol. 42, pp. 145-152, 1997.