International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 7 Issue 1|| January 2018 || PP. 15-21

An Efficient Security System Using Neural Networks

[1*]Amit Kore, [2]Preeyonuj Boruah, [3]Kartik Damania, [4]Vaidehi Deshmukh, [5]Sneha Jadhav
Information Technology, AISSMS Institute of Information Tecchnology, India
Corresponding Author: Amit Kore

***Abstract**: Cryptography is a process of protecting information and data from unauthorized access. The goal of any cryptographic system is the access of information among the intended user without any leakage of information to others who may have unauthorized access to it. As technology advances the methods of traditional cryptography becomes more and more complex. There is a high rate of exhaustion of resources while maintaining this type of traditional security systems. Neural Networks provide an alternate way for crafting a security system whose complexity is far less than that of the current cryptographic systems and has been proven to be productive. Both the communicating neural networks receive an identical input vector, generate an output bit and are based on the output bit. The two networks and their weight vectors exhibit novel phenomenon, where the networks synchronize to a state with identical time-dependent weights. Our approach is based on the application of natural noise sources that can include atmospheric noise generating data that we can use to teach our system to approximate the input noise with the aim of generating an output non-linear function. This approach provides the potential for generating an unlimited number of unique Pseudo Random Number Generator (PRNG) that can be used on a one to one basis. We can use the PRNG generated to encrypt the input data and create a cipher text that has a high cryptographic strength.*

***Keywords:** Neural Networks; PRNG; Natural Noise; synchronize*

Date of Submission: 17-11-2017 Date of acceptance:24-12-2017

I. Introduction

Most companies today handle and store data that is sensitive in nature and can wreak havoc if they fall into the wrong hands. So utmost importance is paid to keep everything safe and secure. In the pursuit of perfect security, the traditional security system are upgraded constantly. With constant evolution of technology, frequent upgradation is pretty much increasing the complexity of the operations performed. A time will come when it will reach a saturation point and adding more operations would cause wastage of infrastructural resources. Previous work in this field [1] presented the idea of evolutionary computing being used to generate ciphers using natural noise as input data stream. They were working with a system called Eureqa, designed by Nutonian Inc. [2]. The system was 'seeded' with natural noise sources which was procured from data based on atmospheric noise generated by radio emissions due to lightening. The focus of this system is to 'force' the system to output a result (a nonlinear function) that is an approximation to the input noise.In this paper, we present a system which uses artificial neural networks to generate cipher keys. We exhibit ow a simple neural network can support a cryptographic system that scores high on a cryptographic strength scale.This paper has 3 sections. State-of-the-art is given in section 2. Section 3 provides the basics of the neural networks that can be used for the generation of ciphers. Section 4 states the algorithm of the security system. The final section concludes the paper.

II. State-of-The-Art

Neural networks in the field of cryptography is relatively new. Neural cryptography is being rigorously researched for the purpose of encryption and cryptanalysis. Recent works in this field include:

2.1. Synchronization of neural networks

Einat Klein et.al. [3] proposed a system that uses two neural networks trained on their mutual outputs. They synchronize using a time based weight vector. This makes the neural networks a medium of generating secret keys that don't need key to be transmitted across public channels. They assess the attacks that can be made on the system, but the system successfully prevents them. R.M. Jogdand et.al. [4] proposed a method that provides a way to synchronize two neural networks using random initial weight vectors. The final synaptic weight

obtained after synchronization is used as a secret key. In this method, in each iteration we get a different key that is why randomness is more which in turn leads to more security

2.2. Neural Cryptography with feedback

Feedback is a core concept in the field of neural networks.Normal neural network cryptographic algorithms follow a simple feed forward network to generate pseudo random numbers which theoretically could be cracked after a large but finite string of data. Hence Andreas Ruttor et.al. [5] suggest a feedback mechanism. Synchronization by mutual learning has been applied to cryptography. They tested the system against ensemble of attackers, i.e. they took an ensemble of many networks. This creates a higher chance of any one of the networks finding the secret key.But when feedback was introduced into the system, the chances of this attack being successful decreases exponentially.

2.3. Cryptography based on delayed chaotic neural networks

aTraditional cryptographic methods following Kerckhoff and Shannon's Principle have reached their limits in terms of algorithmic complexity and chaos. Wenwu Yu et.al. [6] uses chaotic neural networks to generate binary sequences to mask the plaintext by switching of chaotic neural maps and permutations hence providing an alternate way of encryption.

2.4. Cryptography with Artificial Intelligence

Jonathan Blackledge et.al. [7] proposed a paper which gives a way which creates a PRNG through evolutionary and ANN individually and then prove how ANN can be a better solution for cryptography. They provide a method that forces the ANN to approximate the natural noise source which creates a non-linear function creating random numbers.

2.5. Evolutionary computing for Cryptography

Blackledge et.al. [8] demonstrate use of evolutionary computing to create a PRNG which provides a practical solution for large database of random numbers. They proposed the idea: Neural network to approximate the environment noise, thus creating a PRNG in the process. The complexity passes all the required tests such as Lyapunov exponent, maximum entropy, high cycle length, key diffusion characteristics etc.

2.6. Interacting Neural Networks

R. Metzler et.al. [9] gives several scenarios of interacting neural networks which are trained either in an identical or in a competitive way. They proposed an analytical solution that shows the system relaxes to a stationary state which yields a good performance of the system for small learning rates. Using perceptrons in the limit of infinite system size, they derived exact equations of motion for the dynamics of order parameters which describe the properties of the system. The performance of the algorithm is insensitive to the size of the history window used for the decision.

III. Artificial Neural Networks

Artificial neural network is a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.ANNs are composed of multiple nodes, which imitate biological neurons of human brain. The neurons are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its activation or node value. Each link is associated with weight. ANNs are capable of learning, which takes place by altering weight values.A neural network is characterized by its pattern of connections between the neurons, its method of determining the weights on the connections and its activation function.The presence of a hidden unit together with nonlinear activation function, gives it the ability to solve many more problems than can be solved by a net with only input and output units. On the other hand, it is more difficult to train a net with hidden unitsNeural networks are often classified as single layer or multilayer. In determining the number of layers, the input units are not counted as a layer because they perform no computation. Equivalently, the number of layers in the net can be defined to be the number of layers weighted interconnect links between the slabs of neurons. This view is motivated by the fact that the weights contains extremely important information. The basic operation of an artificial neuron involves summing its weighted input signal and applying an output, or activation function. For the input units, this function is the identity function. Typically, the same activation function is used for all the neurons in any particular layer of a neural net, although this is not required. In most cases, a non-linear activation function is used. In order to achieve the advantages of multilayer nets, compared with the limited capabilities of single layer nets, non-linear functions are required (since the results of feeding a signal through

two or more layers of linear processing elements i.e. elements with linear activation functions- are no different from what can be obtained using a single layer.)This paper emphasizes on the use of neural networks in the field of security system. The kind of neural network used in this paper is Radial Basis Function Neural Network (RBFN). An RBFN performs classification by measuring the input's similarity to examples from the training set. Each RBFN neuron stores a "prototype", which is just one of the examples from the training set. When classifying a new input, each neuron computes the Euclidean distance between the input and its prototype.The input vector is the n-dimensional vector that is being classified. The entire input vector is shown to each of the RBF neurons. Each RBF neuron stores a "prototype" vector which is just one of the vectors from the training set. Each RBF neuron compares the input vector to its prototype, and outputs a value between 0 and 1 which is a measure of similarity. If the input is equal to the prototype, then the output of that RBF neuron will be 1. As the distance between the input and prototype grows, the response falls off exponentially towards 0. The shape of the RBF neuron's response is a bell curve, as illustrated in the network architecture diagram.

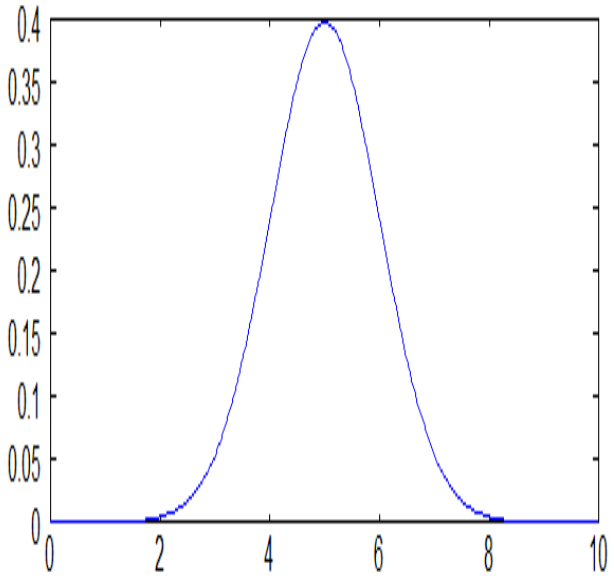

Fig. 1: Bell curve for the activation function

The neuron's response value is also called its "activation" value.The prototype vector is also often called the neuron's "center", since it's the value at the center of the bell curve.The output of the network consists of a set of nodes, one per category that we are trying to classify. Each output node computes a sort of score for the associated category. Typically, a classification decision is made by assigning the input to the category with the highest score.The score is computed by taking a weighted sum of the activation values from every RBF neuron. By weighted sum it means that an output node associates a weight value with each of the RBF neurons, and multiplies the neuron's activation by this weight before adding it to the total response.Because each output node is computing the score for a different category, every output node has its own set of weights. The output node will typically give a positive weight to the RBF neurons that belong to its category, and a negative weight to the others.There are different possible choices of similarity functions, but the most popular is based on the Gaussian. Below is the equation for a Gaussian with a one-dimensional input.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Where x is the input, mu is the mean, and sigma is the standard deviation.

Algorithm of The Security System

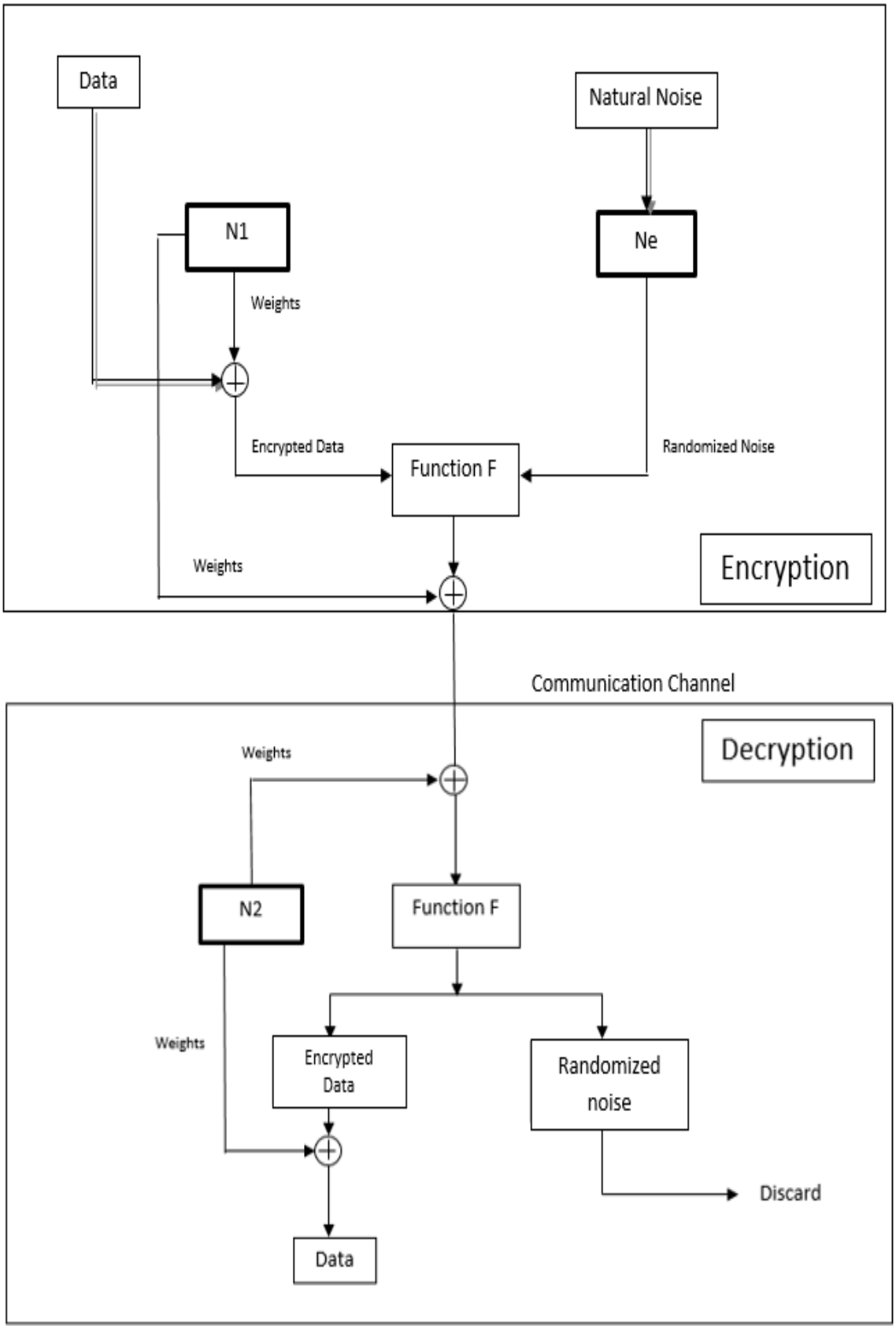

Fig.2: System Architecture

4.1. Training Module

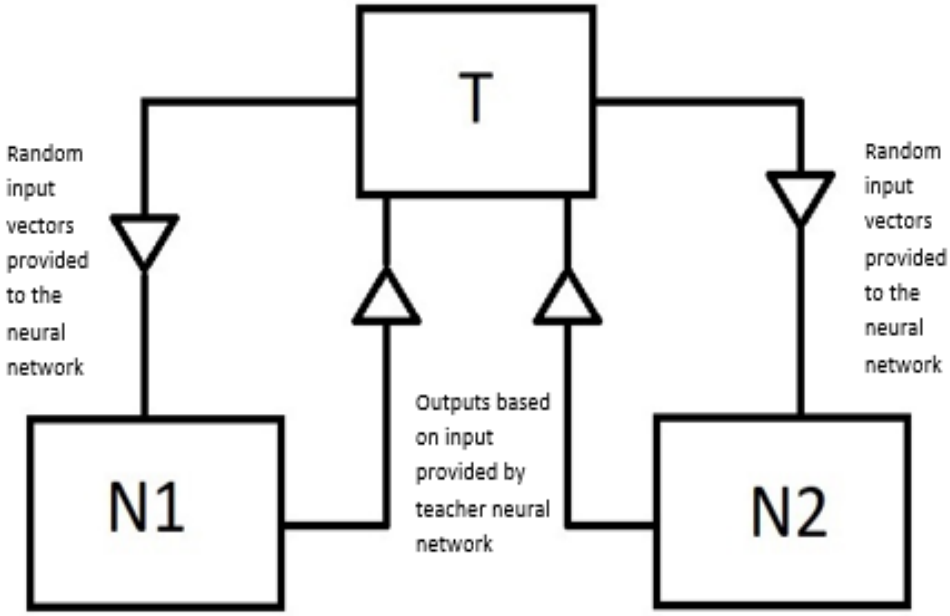

Fig.3: Training Module

Given, neural network N1 and N2. They are initialized with random weights and have 4 input and output nodes.
Step 1: Teacher network T feed randomly initialized input vector into the neural networks N1 and N2.
Step 2: The neural networks N1 and N2 process the input provided by teacher network T and output a vector.
Step 3: If the neural networks produce the same output, move ahead with the training.
Step 4: If the neural networks do not produce the same output, then Weights of neural networks N1 and N2 are changed according to the formulas given below.

$$w_{n1} = w_{n1} + \eta . y_i . x_i$$
$$w_{n2} = w_{n2} + \eta . y_i . x_i$$

Here, wn1 = weight vector of neural network N1
wn2 = weight vector of neural network N2.
η = learning rate of the neural networks.
xi = input vector from teacher network T.
yi = the expected output.

4.2. Encryption Module

Given, Neural network N1 with weight vector w_n1. A random neural network Ne. Input vector D of size 8 bytes
Step 1: The size of D is of 4 bytes and appended with zeroes to make it of 8 bytes.

$$s1 = D \oplus w_{n1}$$

Here, weights of N1 are of 8 bytes. The last 4 bytes of s1 is discarded as it contains the weights of N1.
Step 2: The randomized noise is taken of 4 bytes.

$$s2 = F(s1, Ne) = (s1 + Ne) \ll 3$$

They are simply concatenated and shifted left by 3 bytes. Then s2 which is of size 8 bytes.
Step 3: Final encryption is done by XORing the weights of neural network N1 and s2.

$$f = s2 \oplus w_{n1}$$

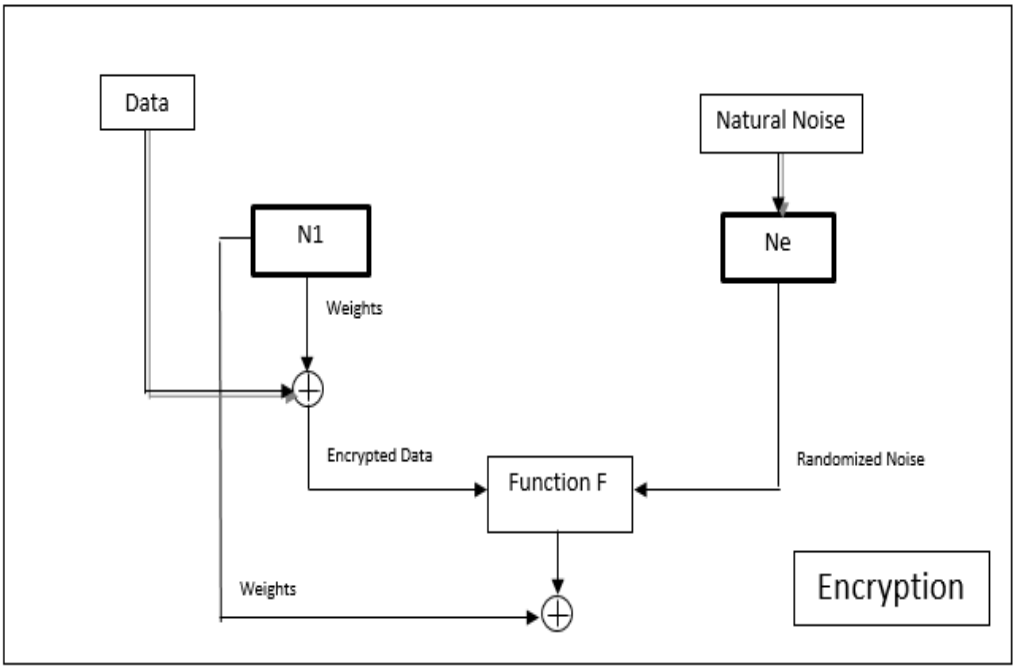

Fig.4: Encryption Module

4.3. Decryption Module

Given, Neural network N2 with weight vector w_{n2}. Encrypted data f coming in through communication channel.

Step 1: The input data f (8 bytes) is XORed with w_{n2}.

$$d2 = w_{n2} \oplus f$$

Step 2: d2 is fed into the function F which separates the data. First right shift and then dissociate in the middle.

$$d1 = F(d2) = (Ed + Ne) \gg 3$$

Output is the encrypted data d1.

Step 3: d1 is XORed with the weights of neural network N2.

$$d = Ed \oplus w_{n2}$$

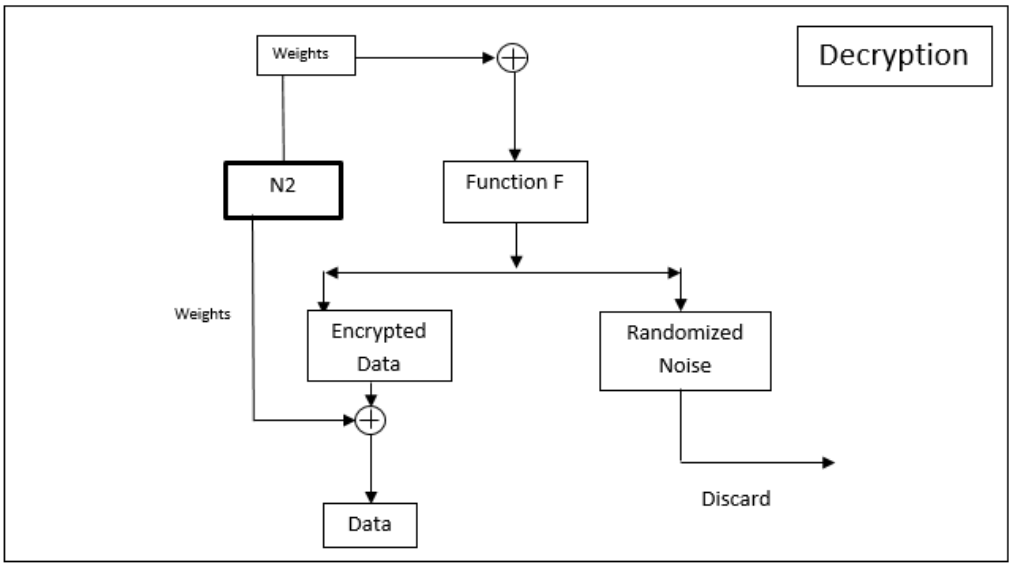

Fig.5: Decryption Module

IV. Conclusion

With the use of the proposed system, we intend to reduce the amount of resources usually used for securing any form of data. The use of neural network in achieving our objectives is exemplary. We can provide a layer of abstraction that is much less complex than the current traditional cryptographic methods. This method provides a way of designing algorithms for generating pseudo-random (chaotic) sequences using truly random strings to evolve an iterator that is taken to be an approximation to these sequences.

Acknowledgements

The authors acknowledge Professor Pritesh Patil, Head ofInformation Technology Dept., AISSMS Institute of Information Technology, Pune, for his support in this work. The authors are thankful to authors / editors / publishers whose articles are cited and included as references in this manuscript.

References

[1]. J. Blackledge, S. Bezobrazov, P. Tobin and F. Zamora, Cryptography using Evolutionary Computing, Signals and Systems Conference, 24th IET Irish, ISSC 2013.
[2]. Eureqa, A software tool for detecting equations and hidden mathematical relationships in your data, Cornell Creative Machine Labs, USA, 20 \3, http://creativemachines.comell.edu/eureqa.
[3]. E. Klein, R. Mislovaty, I. Kantor, A. Ruttor And W. Kinzel, Synchronization of neural networks by mutual learning and its application to cryptography, Advances in Neural Information Processing Systems 17, NIPS 2004, December 13-18, 2004.
[4]. R. M.Jogdand and Sahana S.Bisalapur, Design of an Efficent Neural Key Generation, International Journal of Artificial Intelligence and Applications, Vol.2, No.1, January 2011
[5]. Andreas Ruttor and Wolfgang Kinzel,Neural Cryptography With Feedback, Phys. Rev. E 69, 2004
[6]. Wenwu Yu, Jinde Cao, Cryptography based on Delayed Chaotic Neural Networks,American Physical Society, Phys. Rev. E 69, 2004
[7]. Jonathan Blackledge and Sergei Bezobrazov, Cryptography using Artificial Intelligence, International Joint Conference on Neural Networks, July 2015
[8]. J. Blackledge S. Bezobrazov P. Tobin F. Zamora, Cryptography using Evolutionary Computing, International School of Scientific Computing, June 2013
[9]. R. Metzler , W. Kinzel and I. Kanter, Interacting Neural Networks, Phys. Rev. E 62, 2555, September, 2000

International Journal of Engineering Science Invention (IJESI) is UGC approved Journal with Sl. No. 3822, Journal no. 43302.

Amit Kore."An Efficient Security System Using Neural Networks." International Journal of Engineering Science Invention(IJESI), vol.07, no. 01, 2018, pp. 15-21.