A New Blowfish Using A Neural Network

[1] **Raghad Abdul Hadi Abdul Qader** , [2] **Auday H. Saeed AL-Wattar**
[1, 2] University of Mosul
Raghad.csp43@student.uomosul.edu.iq

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 16 April 2021

Abstract: The Blowfish algorithm is one of the most popular, but it requires significant computational power with many details that make them prey to many attackers. In this paper, the original Blowfish algorithm was optimized by reducing the size of the "s-box" and then designing and implementing it on a neural network (NN). The input to the neural network is text either the(plaintext or ciphertext) and the output obtained from the network is the equivalent text and the key used in both encryption and decryption are the initial weights of the neural network that are trained using the backpropagation network. The proposed neural network (NN) model was designed and simulated by a computer. Simulation results demonstrate reduced complexity of the algorithm, implementation speed, and convergence of results achieved through the NN-based optimized Blowfish algorithm encryption system with optimized Blowfish results.

KEYWORDS: Blowfish, Symmetric block cipher, Neural networks, Back propagation.

1. INTRODUCTION

Computer security is the general name for protecting data and thwart hackers [3]. Cryptography is one of the methods used to secure and guarantee the confidentiality of data [2]. Cryptography Technique symmetric encryption key and asymmetric encryption key [1] Symmetric encryption key is fast. In cryptography, the symmetric key, a single key shared by both parties for encryption and decryption [4].

Symmetric cryptography is classified into two groups, stream cipher, and block cipher [5]. The stream cipher encrypts and decrypts one byte at a time. Block cipher encrypts and decrypts a block simultaneously [6]. There are several symmetric algorithms, such as DES, TRIPLE DES, BLOWFISH, AES, RC4, and RC6 [8]. Block codes are mostly based on Shannon's idea, gave two properties that a good cryptosystem should have to hinder statistical analysis: diffusion and confusion. The strength of the blowfish algorithm rests on the subkey generation and its primary confusion and diffusion design [7]. Blowfish is a symmetric master cipher system based on the Feistel network. Bruce Schneider developed the algorithm. It is a 64-bit block size cipher, and the full version uses 16 rounds to complete the block cipher and uses a large number of subkeys. variable-length key from (32- bits to 448- bits) [9]. This article aims to make some modifications to remove these complex mathematical calculations and details that increase the load of the algorithm in applications by using the neural network in Blowfish design and implementation. An artificial neural network is a computational technology developed to replicate the human brain's solution. This biological neuron simulation aims to obtain the smart characteristics of these cells. They are networks of neural computational elements that respond to input stimuli. One of the most critical capabilities of an artificial neural network is learning from dynamic environments to build widespread solutions by approximating basic mapping between input and output [11, 12, 13, 14]. Neural networks enter various application areas to solve many information processing problems. They succeeded in classification patterns, pattern recognition, system identification, and prediction [16]. Recently, robust studies were obtained on various methods of encryption based on neural networks. A form of encryption is called Neural encryption. The (ANN)-based encryption method is very efficient and offers more security [10]. Neural networks (NN) were used to build and apply the Blowfish algorithm. This paper is organized into six sections as the following section 1 includes the introduction, Section 2 represents the related work, Section 3 includes the proposed method, Section 4 system implementation, section 5 the results and discussion, and section 6 the conclusion.

Blowfish:

Blowfish is a block cipher [17], which means that it splits the text into fixed-length blocks during encryption and decryption. The proposed system enables the conversion of plaintext into ciphertext and back. The block length of blowfish is 64 bits; Plaintext that is no more than eight bytes must be padded. Data encryption begins with a 64-bit block element of plaintext that converts to 64-bit ciphertext. Blowfish uses 18 each of (32 bit) Permutation arrays known as (P-Boxes) from p1, p2..., p18, and four Substitution boxes are known as (S-Box) each of 32-bit size and having (4×256) entries each. These keys have been previously calculated.

Blowfish encryption worked as follows, it divides a block (64 bits) into two equal blocks of size (32 bits) each, and the left block is (XOR) with the first array P1, and thus the obtained result is fed into a function called (F-function). Inside the (F- function), operations are performed that convert (32 bit) blocks to other (32 bit) blocks.

Thus, the (32 bit) entries are (XOR) with the right half, and the result obtained is swapped as the left half for the next round. The Feistel structure of Blowfish algorithm with (16 rounds) of encryption.

Blowfish decryption is the same as encryption, except that P1, P2,...., P18 are used in the inverse order.

The Blowfish algorithm has been extensively analyzed, slowly gaining acceptance as a robust cryptography algorithm [18]. In this paper, we used a neural network to development and implemented the Blowfish algorithm by designing a network to simulate the design of Blowfish.

2. Related work:

Saravana and Shanmugam improved the complexity and security of the Blowfish algorithm by proposing a modified Feistel network with the G function of the Blowfish algorithm [22]. Researchers Bahubali Akiwate and Veena Desai, in their research [23] in 2013, showed promising results in terms of computational requirements, time, several ciphertexts, and plaintext for cryptanalytic. Performance can improve by increasing the number of samples used in neural network training and changing neuronal weights. The authors proposed in [24] an artificial neural network based on a cipher key. A plaintext message consists of bits. Then the bits are transferred to the recipient. The paper proposed a backpropagation network for the proposed approach. In 2011, introduced the Siddeeq. Y development of Advanced Encryption Standard (AES) algorithm to stand against types of attacks using a multi-layered neural network with feed-forward (for encryption and decryption processes). The results show the proximity from NN-based AES with that of the standard (AES)[25].

3. The Proposed method:

3.1 Optimized Blowfish encryption algorithm:

The proposed system designed 64 bits optimized blowfish algorithm depending on the neural network. The architecture of the proposed algorithm is the same as the original one. The modifying was made for the (S-Box) of the F- function. For the original algorithm, the F-function's 32 bits input is divided into four S-boxes (8 bits for each S-Box), while for the proposed method, these 32 inputs is divided into two S-box with 16 bits each. By using the neural system, the proposed method converted the plaintext to ciphertext Effectively. The results showed that the optimized algorithm flexibly showed high security [19].

3.2 Dataset file Proposed:

In the proposed system, a program was written in Matlab (2018a) to create a data set of (3000) for use in training and validation. This data set is stored in a file (data2.txt) consisting of two columns, the first column represents the input data and the second column represents the target data with the ability to flip the columns in order to create a two-way system that accepts the input as (plaintext or ciphertext) and outputs the equivalent text.

3.3 Design of NN –Based Blowfish cryptosyst:

In this part of the research, the neural network was designed using (Object-Oriented Programming) for the "Matlab 2018a" language for ease and flexibility in dealing with and changing the network, which enables the designed system to work on medium performance computers, and this system enables it to process the encryption and decryption in a "Bidirectional system." Which were performed on plain text or ciphertext with good performance and low error. The encrypting algorithm that must be implemented on the neural network is the optimized blowfish. The neural network provides a useful method for controlling nonlinear systems by using the nonlinear activation function. The nonlinear approximation feature of the network makes it more useful. We have used Multilayer Perceptron (MLP), a forward neural network, made up of a number of units (neurons). They can be arranged in layers. The input is taken by the first layer and the output is produced by the last layer. Also, the middle layer is referred to as the hidden layers. MLP has simulation capability [15]. The MLP is designed here as a sequential training. Weight updates are performed after each training is presented. The error function is calculated after each input form and the weights are adjusted and The calculated error function is (root mean square error). MLP takes parameters for a "Bidirectional system" encrypting and decrypting process:

- The input vector is the(plaintext or ciphertext).
- The target from the neural network will be the (plaintext or ciphertext) output from the optimized Blowfish algorithm.
- The initial weights will be the key of the encryption and decryption process.

3.4 Designed network features:

- The network is considered "Supervised Learning", that is, it contains the input and target output[26].
- Its weight is an array with random values .
- The value was installed the learning rate (0.01).

- Connecting it consists of two stages:
  The first stage is the Forward stage, the flow of inputs towards the output.The second stage is the Back propagation stage, any flow or error return from the output to the input direction [27][28].
- Data entering the neural network after changing it (Normalization).
- The network consists of several layers(Multi-layer).[29][30][31].
- The non linear activation function of each neuron will be (leaky ReLU).
- Change the weights to reduce error produce by from neural network[32].
- The condition for stopping the network is to obtain the target output[33]:Target output=Actual output.
- The number of epoch each(25) was specified to approach the correct solution with the lowest possible error.

3.5 Network phases:

The Back propagation network has two phases[34][35][36]:

- **Learning phase**: The network is trained on the given texts with feed forward and Back propagation.
  Forward Feed: is the flow of data from the input cells toward the output cells.
  Back propagation: return error from the output layer toward the input to obtain the ideal weights.
- **The guessing phase**: that is, the network test. Have you trained on the given texts or not, and did we get the ideal weight or not, and it only has a feed forward.

3.6 Back propagation network architecture used in encryption and decryption:

The Backward propagation network that was used in encryption and decryption consists of layers as shown in Figure 1:

- **Input layer:** It includes 64 nodes each neuron represents one bit from plaintext or ciphertext and 4 nodes (to know whether the entered text is plaintext or ciphertext).
- **Hidden layer:** It includes three layers and the number of nodes in the hidden layers. The following formula was adopted in choosing the number of nodes:
  Number of hidden nodes= 2/3 + input nodes × output nodes
- **Output layer:** It includes 64 nodes each neuron represents one bit from plaintext or ciphertext.
-

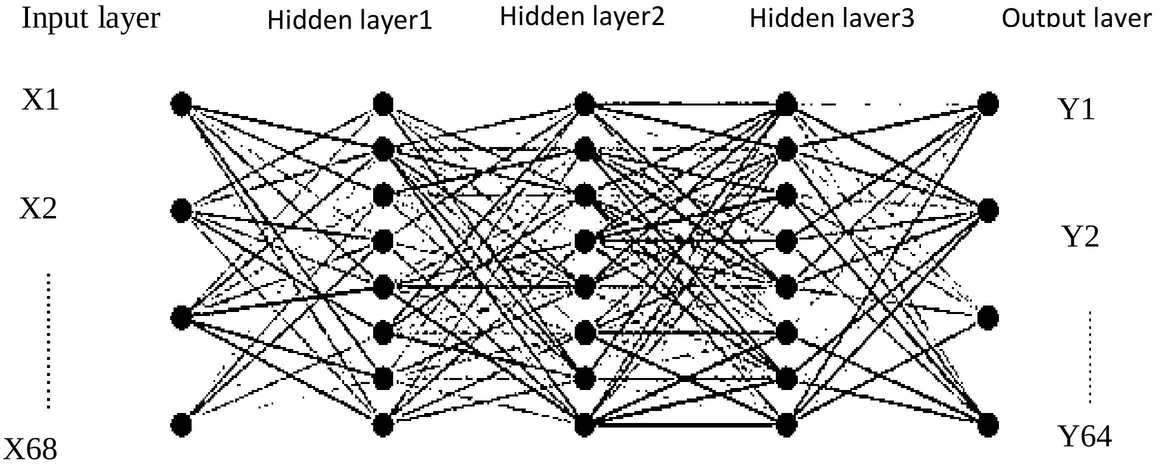


Figure 1: The proposed Blowfish artificial neural network model for encryption and decryption

3.7 Points of note when designing a network:

- The complexity of the Blowfish using the neural network makes it vulnerable to what is called overfitting, That is, the network performs well with the training set, but not as well with the test set To solve this problem, first reduce the number of nodes in each hidden layer, provided that the number of nodes in a single layer does not exceed twice the input nodes[20].
- L2 regularization: is make reducing overfitting and thus improving neural network performance. L2 regularization works by adding a term to the error function used by the training algorithm. This additional term penalizes the weight values. The according to the following equation:
  $$L2 = E + \sum w^2 \times \lambda$$
  where E is the error ,W is the weight and λ is the parameter of regularization and it can be adjusted, the larger weight values will be more punishing if the λ value is large in relation to the λ value of the

smaller the effect of regulation is smaller and this will lead to lowering the values of the weight matrix [21].

4. The System implementation:

4.1 The training phases:

To learning the proposed neural network on the practical side of this research, we used dataset (data2.text) pairs of plaintexts – ciphertext in the learning process. The texts, plaintext, and ciphertext are converted into binary and then transformed (Normalization). The neural network takes text (plaintext or ciphertext) because it is a "Bi-directional system "and trains itself with the target output. After 2000 iterations, with a set of plaintexts and the same initial weights. The model structure is called (series-parallel model). The trained network (1 to32) and (33 to 64) is stored and used in the operational phase.

4.2 The operation phases:

The operation phases will be the feed-forward multilayer neural network using the final weight to produce the (encryption or decryption) processes.

4.3 Network construction and training flowchart:

The flowchart is shown in Figure (2) a and b were used in network construction and training:

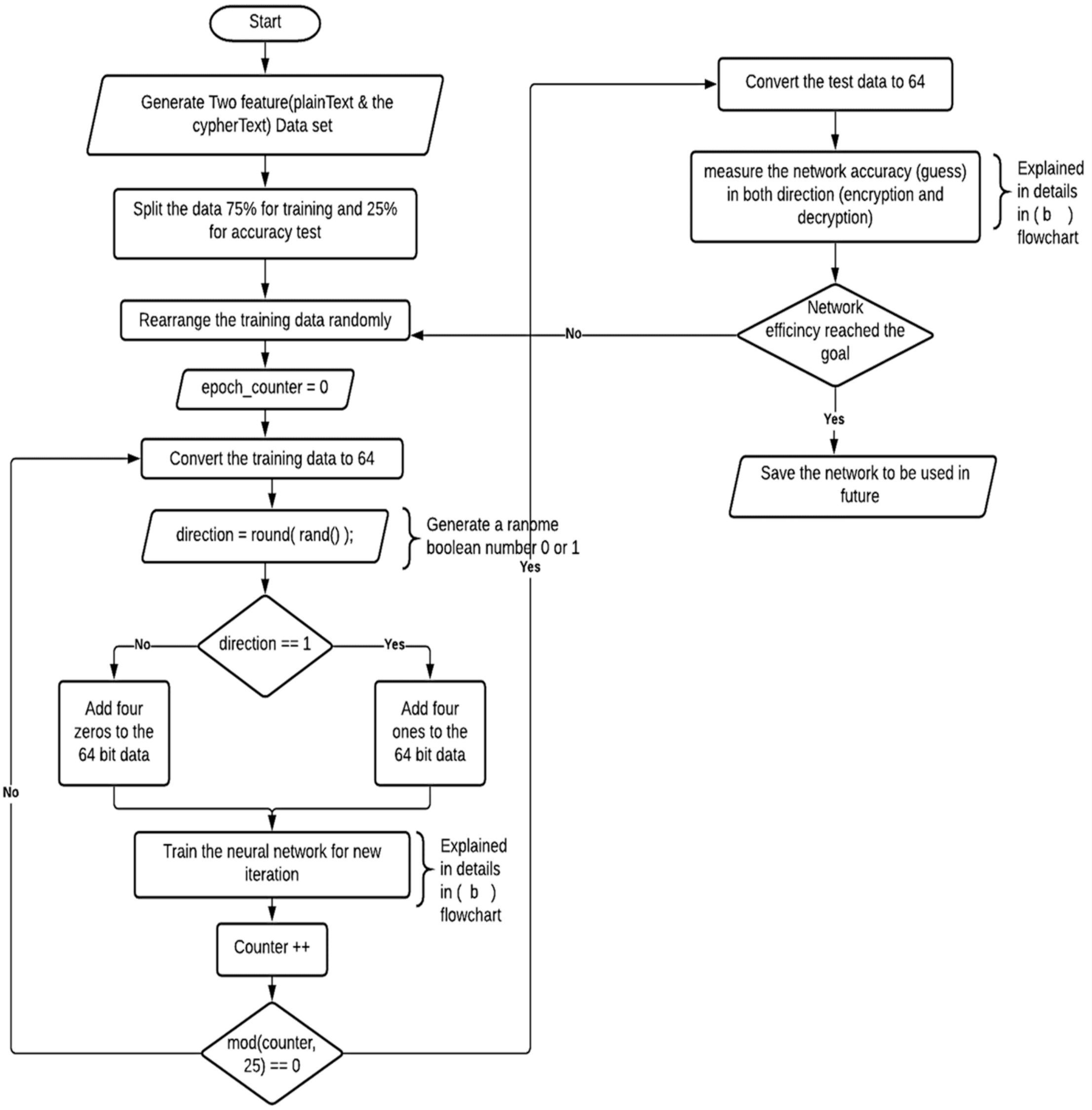

Start
Generate Two feature(plainText & the cypherText) Data set
Split the data 75% for training and 25% for accuracy test
Rearrange the training data randomly
epoch_counter = 0
Convert the training data to 64
direction = round(rand());
Generate a ranbme boolean number 0 or 1
direction == 1
No
Yes
Add four zeros to the 64 bit data
Add four ones to the 64 bit data
Train the neural network for new iteration
Explained in details in (b) flowchart
Counter ++
mod(counter, 25) == 0
No
Yes
Convert the test data to 64
measure the network accuracy (guess) in both direction (encryption and decryption)
Explained in details in (b) flowchart
Network efficincy reached the goal
No
Yes
Save the network to be used in future

Figure (2: a) Network training flowchart

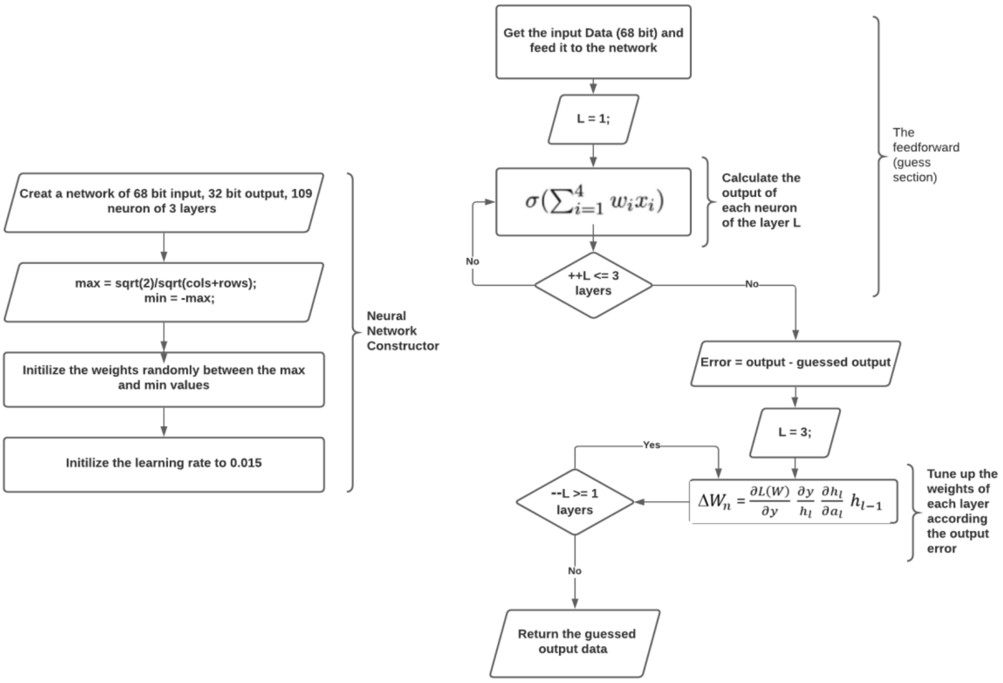
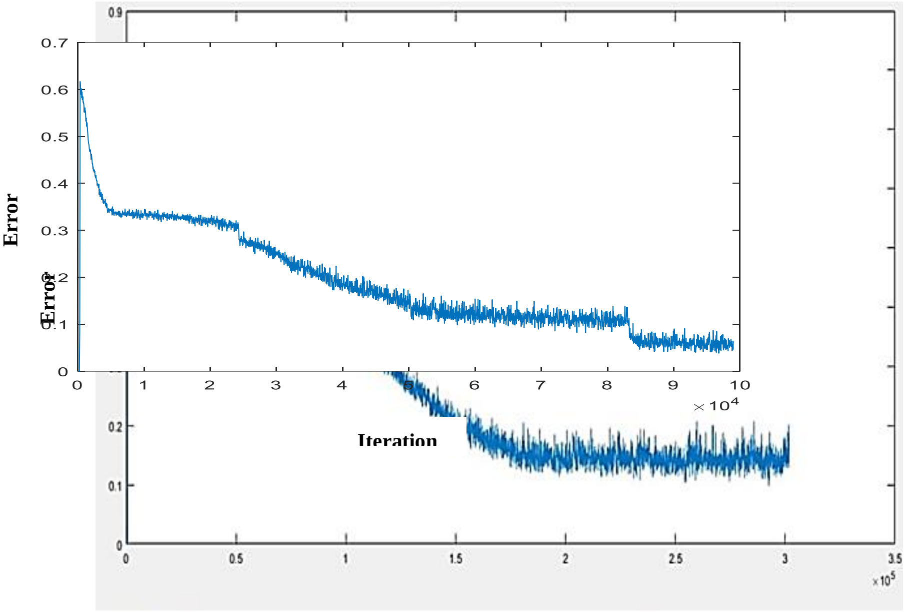

Get the input Data (68 bit) and feed it to the network
L = 1;
$\sigma(\sum_{i=1}^{4} w_i x_i)$
Calculate the output of each neuron of the layer L
The feedforward (guess section)
No
++L <= 3 layers
No
Creat a network of 68 bit input, 32 bit output, 109 neuron of 3 layers
max = sqrt(2)/sqrt(cols+rows);
min = -max;
Initilize the weights randomly between the max and min values
Initilize the learning rate to 0.015
Neural Network Constructor
Error = output - guessed output
L = 3;
Yes
--L >= 1 layers
$\Delta W_n = \frac{\partial L(W)}{\partial y} \frac{\partial y}{h_l} \frac{\partial h_l}{\partial a_l} h_{l-1}$
Tune up the weights of each layer according the output error
No
Return the guessed output data

Figure (2: b). Neural Network constructor

5. The Results and Discussion:

The training results for the 64-bit proposed model were the output text of the NN based optimized Blowfish in Bidirectional both encrypting and decrypting system was identical to the optimized Blowfish encryption and decryption process in term of security. The neural network was validated using RMSE (root mean square error), the square root of SME, which illustrates several iterations during training. Figure 3 shows training the neural network using 64-bit.

Figure 3: Training NN of 64 bits



Iteration

The training results in batch form into (1 to 32) bits and (33 to 64) bits running in parallel; this will reduce t raining time and closer to the target vector. Figures 4 and 5 show the training process.

Figure 4: Training process for bits (1 to 32)bit

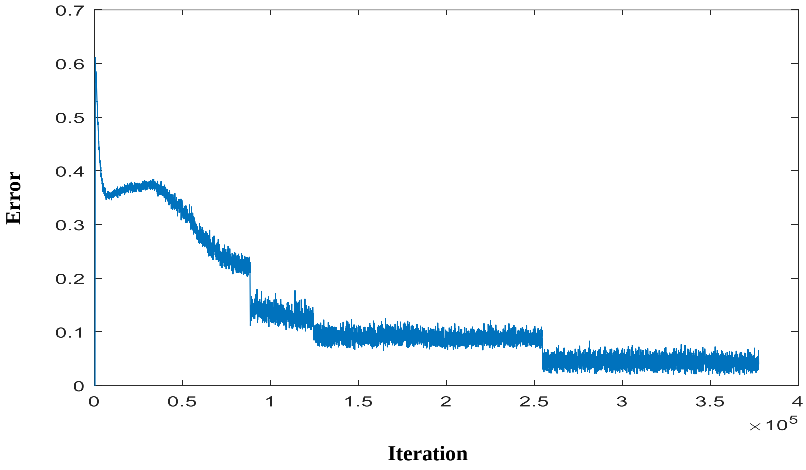
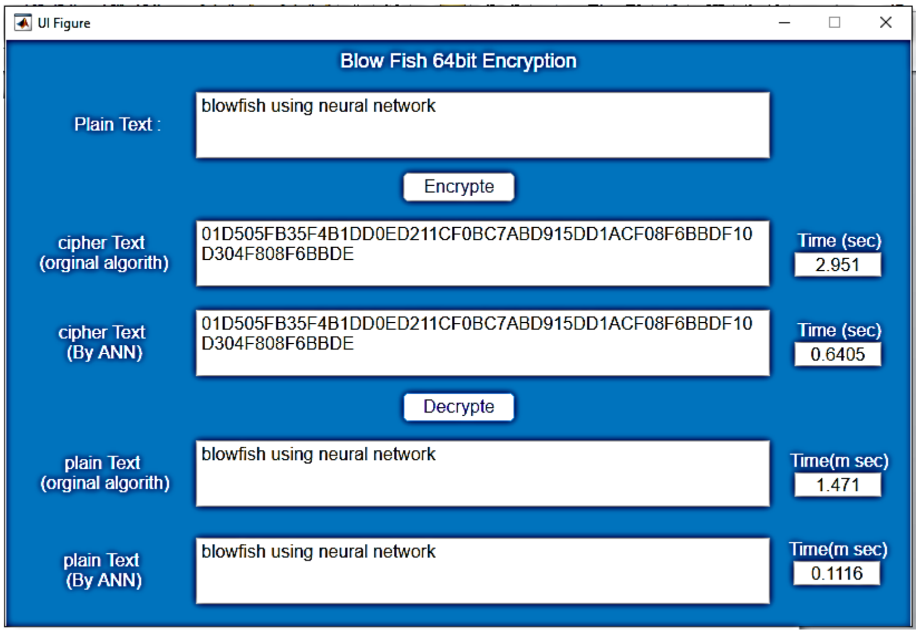


Figure 5: Training process for bits (33 to 64) bit

The program was implemented, and the interface that illustrated the encryption and decryption process is shown in figure 6. The results were high-speed, with the results converging between the optimized Blowfish using the neural network and the optimized Blowfish.



Blow Fish 64bit Encryption

Plain Text : blowfish using neural network

cipher Text (orginal algorith): 01D505FB35F4B1DD0ED211CF0BC7ABD915DD1ACF08F6BBDF10D304F808F6BBDE

Time (sec) 2.951

cipher Text (By ANN): 01D505FB35F4B1DD0ED211CF0BC7ABD915DD1ACF08F6BBDF10D304F808F6BBDE

Time (sec) 0.6405

plain Text (orginal algorith): blowfish using neural network

Time(m sec) 1.471

plain Text (By ANN): blowfish using neural network

Time(m sec) 0.1116

Figure 6: implementation Blowfish-based Neural Network

Figures 7 and 8 illustrate the comparison process between the same texts that were encrypted and decrypted d using the traditional method of the Blowfish algorithm and the proposed method that relied on neural networks NN, as it appears that the proposed method showed excellent security that matches the security of the original algorithm in addition to the speed of implementation of the algorithm

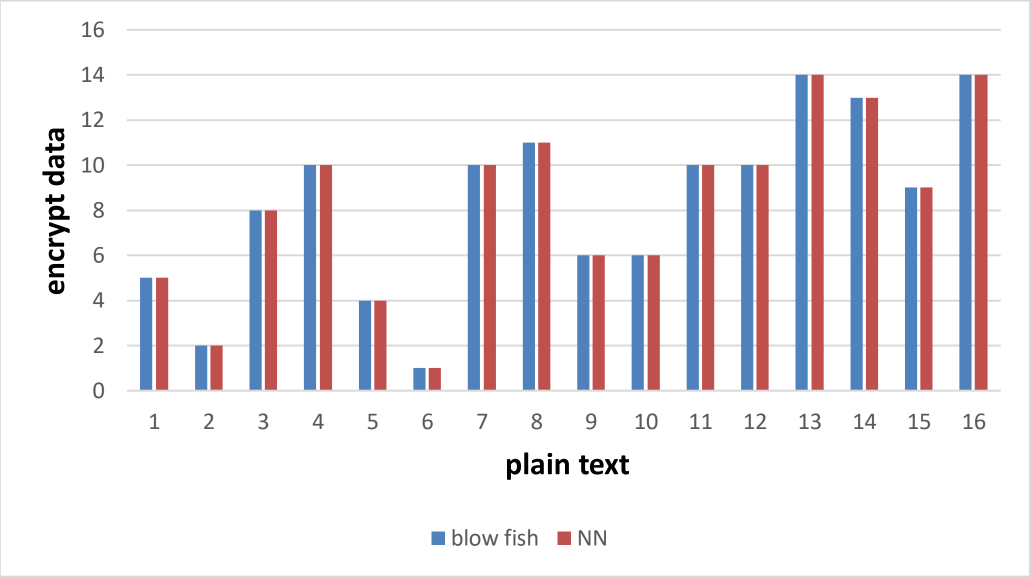


Figure 7: The encryption process for both tradition and NN based Blowfish algorithm

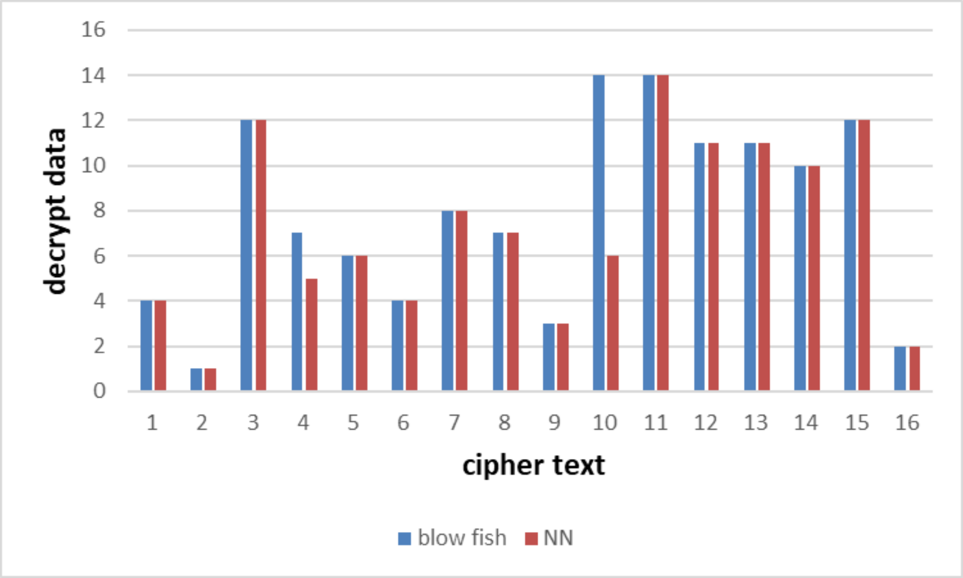


Figure8: The encryption process for both tradition and NN based Blowfish algorithm

6. Conclusion:

It is possible to replace the traditional encryption methods and adopt methods based on artificial intelligence and neural networks, where strong security results are obtained in addition to high speed and accuracy and make the attacker's task more difficult.

The use of a single system (bi-directional) for encoding and decoding any system that accepts input (plaintext or ciphertext) and outputting the equivalent text led to speedy implementation.

Reducing the nodes in each hidden layer, provided that it does not exceed two times the input nodes, and use L2 regularization repository to eliminate the problem of overfitting, and this additional term penalizes the weight values.

The most efficient multilayer feedforward artificial neural network in encryption has been dealt with, which is the Backpropagation network, which depends on propagating the error between the calculated output and the target (plaintext or ciphertext) through the hidden layers up to the input layer, and then adjusting the weights according to that error. Furthermore, because it uses the "Leaky ReLU" function, it is excellent and fast to calculate.

Training the network on a smaller number of outputs facilitates the network's training in parallel from (1 - 32)bits and(33 -64) bits of the so-called series-parallel system to give better results more speed than large networks.

Results show the closeness of the results achieved by the proposed NN-based optimized Blowfish cryptosystem with that of the optimized Blowfish.

The proposed system is quick, easy to implement, and accurate in performance

. **References:**

1. Maitri P V, Verma A. Secure file storage in cloud computing using hybrid cryptography algorithm. In: Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016.; 2016:1635-1638. doi:10.1109/WiSPNET.2016.7566416.
2. Bhuvaneshwari M, Tenmozhi S. 2016, A VLSI architecture for security based stenographic processor with AES algorithm. International Journal of Electrical and Computer Engineering; pp 1–6.
3. Stallings W 2002 Data and Computer Communications.
4. Tanenbaum A 1996 Computer Networks,
5. K. Acharya, M. Sajwan, and S. Bhargaya, “Analysis of Cryptographic Algorithms for Network Security" International Journal of Computer Applications Technology and Research., vol. 3, issue no. 2, pp.130- 135–8887, 2013.
6. Manikandan G, Rajendran P, Chakarapani K, Krishnan G and Sundarganesh G, "A Modified Crypto Scheme for Enhancing Data Security", Journal of Theoretical and Applied Information Technology, Vol. 35, No.2, pp.149-154, 2012.
7. "Type-3 Feistel Network of The 128-bits Block Size Improved Blowfish Cryptographic Encryption", Ashwaq T. Hashim Received on: 28/5/2008 Accepted on: 6/11/2008
8. Diaa Salama, Abdul. Elminaam, Hatem Mohamed, Abdul Kader and Mohie Mohamed Hadhoud, “Performance Evaluation of Symmetric Encryption Algorithms”, International Journal of Computer Science and Network Security, vol.8 No.12, December 2008.
9. BRUCE SCHNEIER, “Description of new variablelength key, 64-bit Block Cipher (Blowfish).” Workshop on Fast Software Encryption, December 1993; published by Springler-Verlag.
10. Zaid Hamid Mahmoud, Hanif Barazandeh, Seyed Mojtaba Mostafavi, Kirill Ershov, Andrey Goncharov, Alexey S Kuznetsov, Olga D Kravchenko, Yu Zhu. identification of rejuvenation and relaxation regions in a Zr-based metallic glass induced by laser shock peening. Journal of Materials Research and Technology. 2021; 11, 2015-2020.
11. Khaled Alallayah, Mohamed Amin, Waiel Abd El-Wahed, and Alaa Alhamami, “Attack and Construction of Simulator for Some of Cipher Systems Using Neuro-Identifier”, The International Arab Journal of Information Technology, October 2010.
12. Dan W. Patterson, “Artificial Neural Networks, Theory and Application”, prentice hall, 1996.
13. Bart Kosko, “ Neural Networks and Fuzzy Systems”, university of southern California, 1992.
14. Romariz A., “Neural Network Applied to Nonlinear Modeling,” www.ene.unb.br/romariz/, Last Visited 1996
15. J. Principe, N. Euliano, and W. Lefebvre, “Neural and Adaptive Systems: Fundamentals through Simulations”, John Wiley& Sons, Inc., 2000.
16. Khaled Alallayah, Mohamed Amin, Waiel Abd El-Wahed, and Alaa Alhamami, “Attack and Construction of Simulator for Some of Cipher Systems Using Neuro-Identifier”, The International Arab Journal of Information Technology, October 2010..
17. S. Vaudenay, "On the Weak Keys in Blowsh," Fast Software Encryption, Third International Workshop Proceedings, SpringerVerlag, 1996, pp. 27-32.
18. M Kavitha, Zaid Hamid Mahmoud, Kakarla Hari Kishore, AM Petrov, Aleksandr Lekomtsev, Pavel Iliushin, Angelina Olegovna Zekiy, Mohammad Salmani. Application of Steinberg Model for Vibration Lifetime Evaluation of Sn-Ag-Cu-Based Solder Joints in Power Semiconductors. IEEE Transactions on Components, Packaging and Manufacturing Technology. 2021; 11(3);444-450
19. Christina L, Joe Irudayaraj V S, “Optimized Blowfish Encryption Technique”,International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization)Vol. 2, Issue 7, July 2014.
20. Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers; Gaurang Panchal1 , Amit Ganatra2 , Y P Kosta3 and Devyani Panchal; International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011 ISSN: 1793-8201.
21. Deep learing (Adaptive computation and machain learing series)illustrated edition by lan goodfellow ;yoshuabengio

22. Saravana Kumar, A.Shanmugam, “Modified F – Function for Feistel Network in Blowfish Algorithm”, International Journal of Engineering and Innovative Technology (IJEIT) Volume 4, Issue 4, October 2014.
23. Artificial Neural Networks for Cryptanalysis of DES; Bahubali Akiwate and Veena Desai; Vol. 2 Issue 4 August 2013.
24. Volna, Eva, et al. Cryptography Based On Neural Network. ECMS. 2012.
25. Siddeeq. Y. Ameen and Ali H. Mahdi; " AES Cryptosystem Development Using Neural Networks" International Journal of Computer and Electrical Engineering, Vol. 3, No. 2, April, 2011 1793-8163.
26. Tsaregorodtsev, Victor G., Parallel implementation of back-propagation neural network software on smp computers, LNCS, (2005):P 186-192
27. 19. Khan Asif Ullah T. K. Bandopadhyaya, Genetic algorithm based back propagation neural network performs better than back propagation neural network in stock rates prediction, IJCSNS, International Journal of computer science and network security, Vol. 8, No. 7, (2008): P 1-5.
28. Henning, Kai Thorsten dc, Multi sensor system for fast analysis in environmental monitoring with an application in wastetreatment, EARsel- SIG, No. 1, (2000):P 61- 67.
29. Valdes, Julia J., Behavior of similarity- based neuro- fuzzy networks and evolutionary algorithms in time series model mining, NRCCNRC, (2002): P 1-6.
30. Nikravesh, Masoud F. Aminzadeh, Past, present and future intelligent reservoir characterization trends, Journal of petroleum science and engineering 31. , (2001): P 67-79. [31]. Kasabov, Nikola Senior Member, Evolving fuzzy neural networks for supervised/ unsupervised, IEEE, Vol. 3, No. 6, (2001):P 1-67.
31. Sajda, Paul, Learning contextual relationships in mammograms using a hierarchical pyramid neural network, IEEE, Vol. 21, No. 3. (2002).
32. Chang, Ray-I, Liang-Bia, Intrusion detection by back propagation neural networks with sample-query and attribute-query, International Journal of computational Intelligence Research ISSN., Vol. 3, No. 1, (2007):P 6-10.
33. Mitrassushmita,Senior Member, Data mining in soft computing framework a survey, IEEE, Vol. 13, No. 1. (2002).
34. Sharhui, Liu Yao Hongeun, Neural network based steganalysis in still image, IEEE, (2003): P 1-4.
35. Lakshmi, Seetha Shooyu Zhou, Selectivity estimation in extensible data bases - a neural network approach, 24th ULDB conference new york, (1998): P 1-5.