Sphinx Journal of Mathematical Sciences
1(2), 94–106, 2026
DOI: https://doi.org/10.64389/sjms.2026.012127
https://sphinxsp.org/journal/index.php/sjms/

Research article

Comparative Analysis of Artificial Neural Network Statistical Parameters and Novel Techniques for Data Encryption and Decryption, Distribution, and Estimation

Samah Mohamed[1], **Ali Saad Gaballah**[2], **Eman Said Osman**[3,4], **Ayman Haggag**[4], **Khaled Elsharkawy**[3,*]

[1] Department of Computer Science, Nahda University (NUB) , Beni-Suef, Egypt.

[2] Department of Curricula and Teaching Methods, Faculty of Education, Banha University, Banha, Egypt.

[3] Basic Science Department, New Cairo University Technology, Egypt.

[4] Electronic Technology Department, Faculty of Technology and Education, Capital University, Egypt.

* **Correspondence:** Khaled.elsharkawy68@gmail.com

ARTICLE INFO

Keywords:
Artificial Neural Network;
Data Security;
Decryption;
Encryption;
Statistical analysis.

Mathematics Subject Classification:
62Exx, 68T07, 92B20

Important Dates:
Received: 25 May 2026
Revised: 20 June 2026
Accepted: 24 June 2026
Online: 25 June 2026

ABSTRACT

The use of statistical metrics to compare the application of several unique communication techniques that rely on a certain system is the foundation of this paper. By making it more difficult for attackers to anticipate patterns and the speed of the encryption and decryption processes, this technology protects communication routes. When compared to solutions using mathematical techniques, the employment of artificial neural networks (ANN) in data encryption and decryption, such as communications, demonstrates that the test patterns of cryptographic messages in our collection displayed a shorter decryption time. This study's methodology can be successfully applied to lengthy communications, where it is difficult and time-consuming to apply mathematical methods for encryption and decryption. We present various measures of fitting goodness to the ANN method in order to reach the most preferred distribution to analyse our results, along with the benefits and drawbacks of both approaches in terms of security level, encryption and decryption time, and speed of use.

1. Introduction

Encryption and decryption are fundamental methods in data security that serve as the cornerstone of defense against unwanted access to private data. The rise in digital communication and the quick expansion of data volume have increased the significance of robust and effective encryption techniques. The importance of encryption in today's digital environment is demonstrated by the need to safeguard data privacy, facilitate secure communications, and preserve the integrity of information across several platforms [10, 11, 2]. The structure and capabilities of the human brain serve as the model for ANN [5, 16]. have shown to be incredibly powerful tools in a variety of fields, including information security. They are particularly useful because of their ability to memorize information, identify intricate designs, and adapt to unused data. These characteristics make ANNs perfect for enhancing encryption and decryption procedures [20, 14], which essentially rely on the generation and identification of complex patterns in data.

This study's main goal is to compare two approaches: the first uses cutting-edge communication strategies that make use of intricate networks and technologies to improve security levels and encryption and decryption speeds [21]. ANN are used in the second method. We compare the length of encryption and decryption as well as the degree of data security. Our comparison is supported by a numerical example, which is followed by the findings, conclusions, and suggestions for further research.

1.1. Literature review

ANNs have been utilized recently as a safe way to encrypt and decrypt a variety of data, including messages. For instance, in order to make his network more secure and quick. Haggag et al. [6] combined the deep learning approach with the recognized pattern concept.

1.2. Related works

The approach used here in [14] needs to be straightforward, efficient, and precise. For cloud computing applications, a simple yet incredibly secure encryption-decryption (SHSED) technique has been introduced. Basic and effective logical operations including XOR, addition, subtraction, and byte shifting are used in this approach. Additionally, choosing the length of the secret key and the number of rounds to produce the ciphertext is made possible by the proposed method. In 2013, Mohammad [15] introduced A New Approach for Complex Encrypting and Decrypting Data," which improved security objectives and preserved communication channel security by making it more difficult for an adversary to predict a pattern and the pace of the encryption/decryption process.

1.3. Introduction to neural networks

Neural networks in computers simulate how organic neurons in the human brain work. Mathematical models of neural networks are used to simulate the behavior of actual genuine neurons in computational tasks.

1.3.1. The biological neuron

The terms commonly used to describe a biological neuron are explained in the glossary of terms that follows [16]. The neuron's nucleus is represented by the soma, as shown in Figure 1. Dendrites are the

long, asymmetrical fibers connected to the input channels of the soma. An axon is another kind of link connected to the soma's output channels [5].

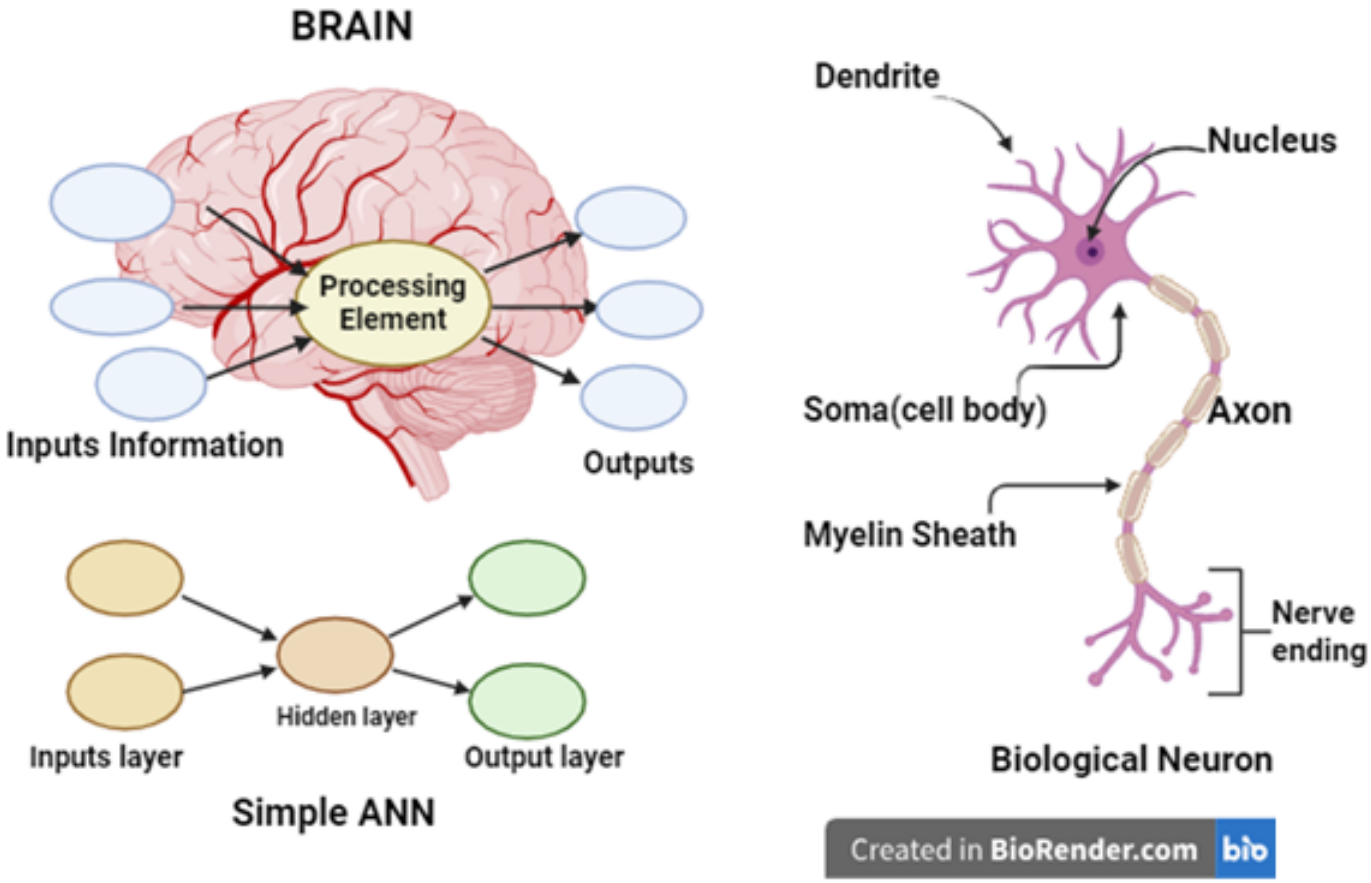


Figure 1. The biological neuron

An axon's output is a millisecond-long voltage spike, or pulse. Ions enter the cell when a neuron fires. Neurotransmitters produce this by forming ion channels on the cell membrane. Learning is linked to the size of synapses, which are the electrical locations where axons in neurons end. A "spine" is a tiny, membrane-covered projection that emerges from the dendritic spine, or spine, of a neuron. At the synapse, one axon usually provides the input to the spine.

1.3.2. Back-Propagation Feed-Forward Networks

A neuron's output is either directly or indirectly returned to its input through connections with other related neurons in feed-forward back-propagation networks, such as the Elman Recurrent Network shown in Figure 2.

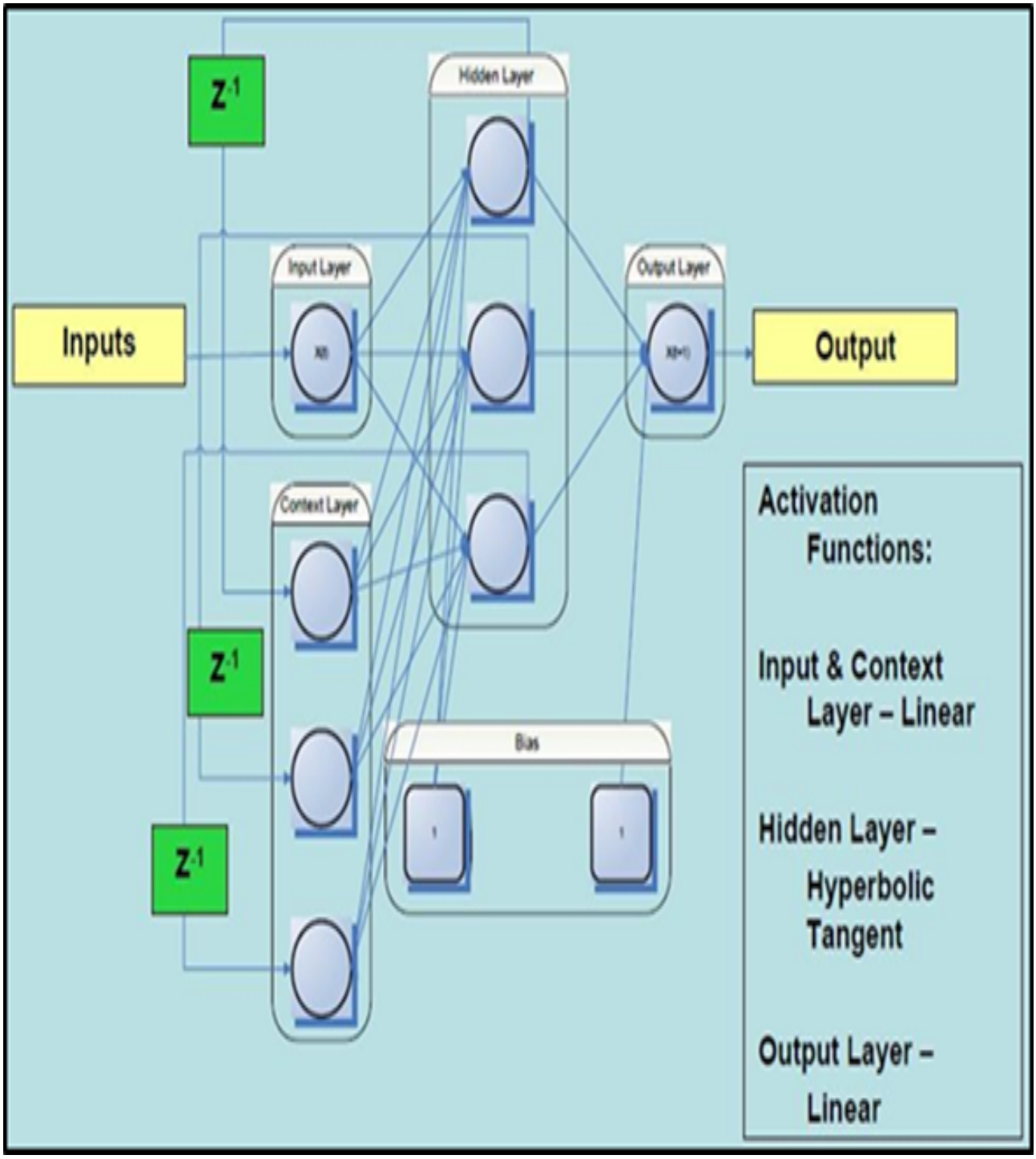


Figure 2. Feed-forward back-propagation networks

In our used ANN we implement the following features [6]:
Number of input layers: 2
Number of hidden layers: 2
Number of output layers: 1
Activation function: sigmoid function as shown in figures 3,4

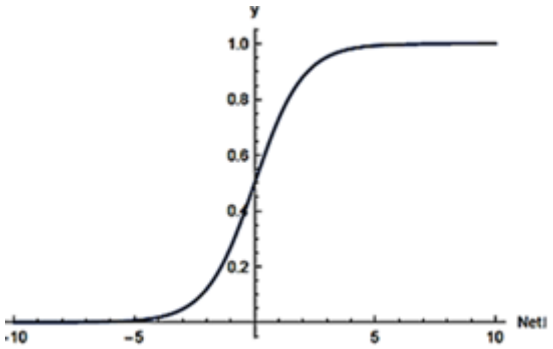


Figure 3. Graph of the function $[y = \frac{1}{1 + \exp\{-Net\}}]$

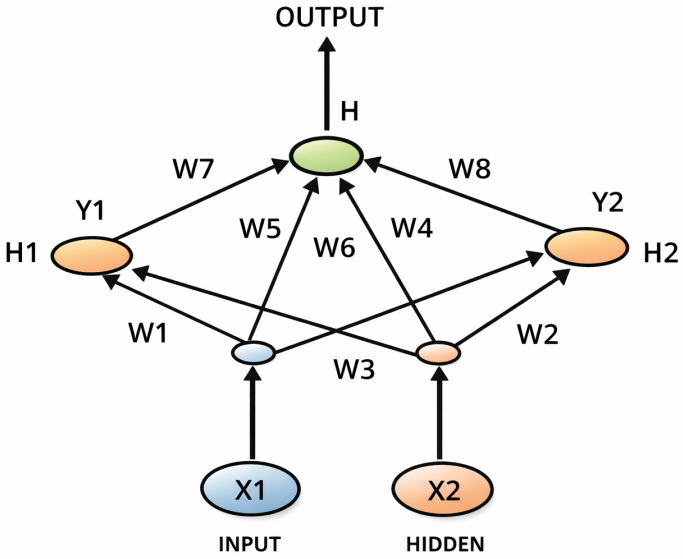


Figure 4. Sigmoid function of neural network training

Learning Techniques

Artificial neural networks function by learning, which includes reaching ideal weight values. This learning phase's objective is to instruct the network to produce a particular output in response to a given input [16]. Thanks to the fine-tuned optimal weights, the trained neural network should be able to produce the correct output for the associated input pattern within an acceptable accuracy range once the learning process is over.

Weight Modifications

Two distinct kinds of supervised learning algorithms are distinguished by the timing and method of weight updates. In real-time, stochastic, or delta learning, neural network weights are modified with each distinct input pattern. This technique lessens the potential for bias arising from the input pattern sequence of the training set by selecting a random pattern from the training set for the subsequent input. Conversely, batch learning, sometimes referred to as offline learning, entails accumulating neural network modifications and applying them to the weights only after each training sequence has been processed.

1.4. Methodology

To compare the novel strategy by Obadiah Mohammad with the ANN method, we first select relevant statistical characteristics such encryption time, decryption time, and security level. A set of data is encrypted and decrypted using both methods in the experimental setting, and the results are carefully examined. Data collection involves generating random data sets of different sizes in order to assess each approach's scalability and efficacy. The statistical analysis included a number of techniques and tools, including probability of a security breach, mean time to encrypt/decrypt, and standard deviation.

2. Selection of Statistical Parameters

The selected parameters for comparison consist of:

- **Encryption Time (ET)**: The mean amount of time required for data encryption.
- **Decryption Time (DT)**: The mean amount of time needed to decode information.
- **Security Level (SL)**: Defined as the likelihood of improper decryption efforts succeeding and standard deviation.

2.1. Experimental setup

The datasets used range in size from small text documents (1 KB) to larger files (10 MB). To ensure statistical significance, the encryption and decryption processes are performed 100 times using both ANN and a novel method [7, 8, 18].

2.2. Data collection

A Python program was used to construct unstructured data collections, ensuring that they contained a variety of data forms (text, binary, etc.). While the novel approach was applied directly to the data, these collections were divided into training and evaluation sets using the ANN technology [19, 17, 9].

2.3. Statistical analysis

The chance of a security breach, average encryption duration, and average decryption length are among the numerical features computed for contrast. We investigate these features for both the ANN method and the novel approach using Python and statistical tools.

2.4. Encryption and decryption time

For example, if the average encryption time (ET) for a 512-bit key using ANN is 0.05 seconds with a standard deviation of 0.1 and the novel approach achieves the same with an average time of 0.3 seconds and 0.1 seconds SHSED with a standard deviation of 0.05, we can assess the efficacy and consistency of each method. The same holds true for the decryption time (DT), and we calculate each method's average decryption time [3, 4, 13].

2.5. *Security level*

The percentage of successful decryption attempts without the key determines the likelihood of a security breach SL. For example, a breach possibility of 0.01% might be indicated by an ANN, whereas a greater probability of 0.05% might be suggested by the innovative techniques [9, 12].

3. Numerical Example

Our message is the message (Evaluation), which consists of 10 letters. By using the algebraic hill cipher method to encrypt and decrypt the following simple message algebraically [17, 9, 6].

Step 1: Create the table of the message utilizing the Hill cipher with a mode 26 system.

E	V	A	L	U	A	T	I	O	N
4	21	0	11	20	0	19	8	14	13

Step 2: split the initial letter into segments, where each segment consists of two letters. (2×2) and put these blocks in the matrix $P \longrightarrow$

$$P = \begin{pmatrix} 4 & 0 & 20 & 19 & 14 \\ 21 & 11 & 0 & 8 & 13 \end{pmatrix}$$

Step 3: Choose the encryption matrix E (public key) [12] to have an inverse and can be multiplied by P

Let

$$E = \begin{pmatrix} 3 & 6 \\ 1 & 5 \end{pmatrix}$$

Step 4: Obtain the matrix Q as the outcome of the multiplication of $(E \times P)$ and then transfer it to mode 26.

Step 5: split the matrix Q into blocks to obtain the encrypted message.

8	5	14	3	8	20	1	7	16	1
I	F	O	D	I	U	B	H	Q	B

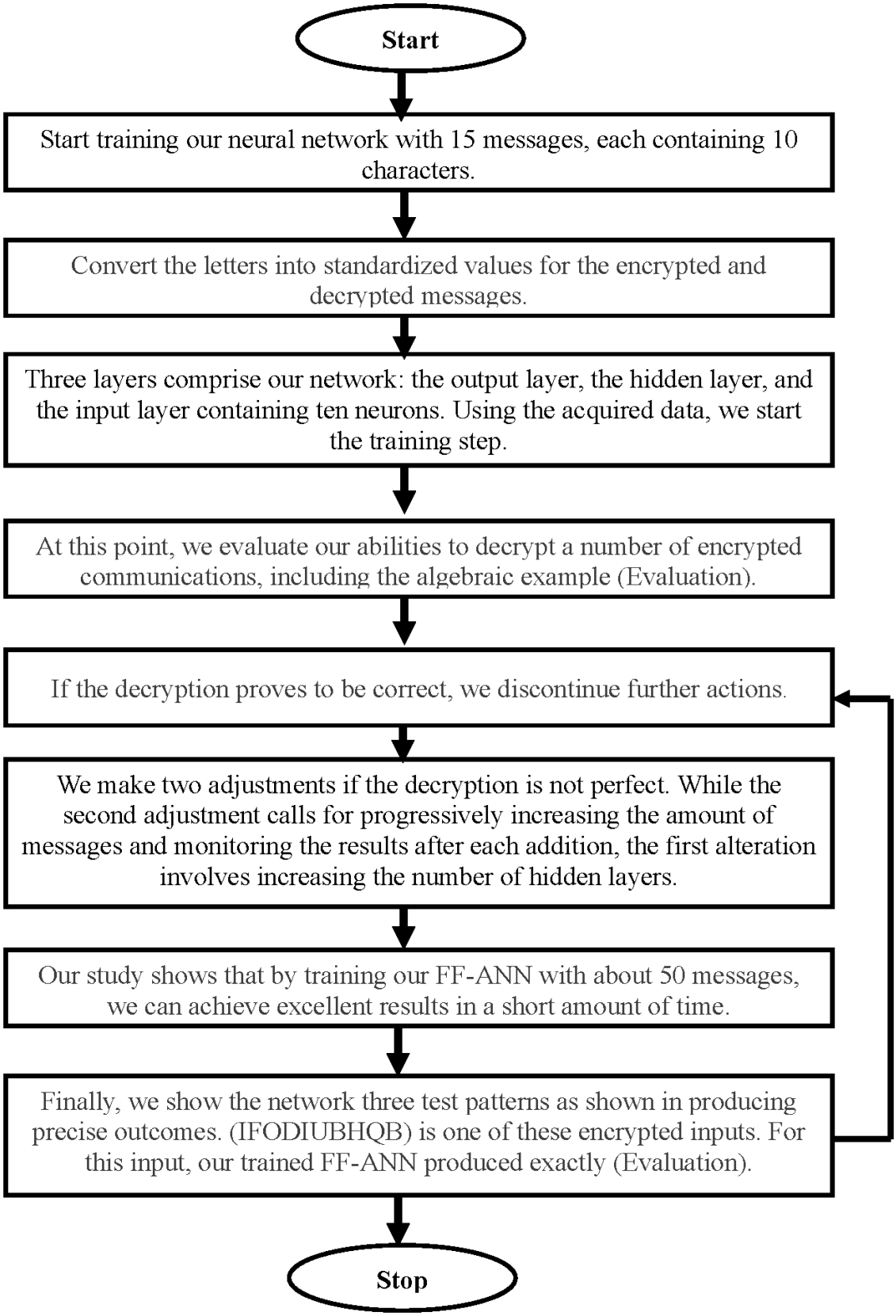
Thus, the encoded message Q is $(IFODIUBHQB)$.

Step 6: Initiate the decryption process, obtain the inverse of matrix, call it E^{-1}, and multiply it by the matrix Q so the result is a matrix called D where

$$D = (E^{-1} \times Q)_{\text{mode } 26} \longrightarrow D = \begin{pmatrix} 4 & 0 & 20 & 19 & 14 \\ 21 & 11 & 0 & 8 & 13 \end{pmatrix}$$

This is the same message (EVALUATION) with the block chain. We shall use the training of FF-ANN to decrypt the same Messages.

3.1. *Using FF-ANN algorithm to decrypt messages*


Start
Start training our neural network with 15 messages, each containing 10 characters.
Convert the letters into standardized values for the encrypted and decrypted messages.
Three layers comprise our network: the output layer, the hidden layer, and the input layer containing ten neurons. Using the acquired data, we start the training step.
At this point, we evaluate our abilities to decrypt a number of encrypted communications, including the algebraic example (Evaluation).
If the decryption proves to be correct, we discontinue further actions.
We make two adjustments if the decryption is not perfect. While the second adjustment calls for progressively increasing the amount of messages and monitoring the results after each addition, the first alteration involves increasing the number of hidden layers.
Our study shows that by training our FF-ANN with about 50 messages, we can achieve excellent results in a short amount of time.
Finally, we show the network three test patterns as shown in producing precise outcomes. (IFODIUBHQB) is one of these encrypted inputs. For this input, our trained FF-ANN produced exactly (Evaluation).
Stop

3.2. Security analysis

The algorithm is put through a wide range of tests and analysis to determine how secure it is. Numerous cryptanalysis studies, the NIST statistics package, and a variety of other statistical analyses are used in the evaluation. The algorithm is subjected to a number of analytical methods, such as an evaluation of information entropy [17] and correlation analysis between public and private locations [9]. Brute-force is a broad method that attempts all possible key [21]. Its chances of success are determined by the vital space size. Cryptanalytic attacks, such as chosen-plaintext attacks (CPA), chosen-cipher text attacks (CCA), or linear/differential cryptanalysis, take advantage of mathematical faults or vulnerabilities in the algorithm's architecture that go beyond the key length [2, 3]. The encryption is considered "broken" if an attacker can obtain the key or plaintext in significantly fewer steps than a brute-force attack [2, 3]. This most potent attack's complexity is then effectively reduced to the security level of the algorithm. For example, 3DES's security level is only 112 bits despite having a 168-bit key because of effective cryptanalytic assaults [4]. In order to assess the security level of the proposed method, we updated the manuscript with a number of tests and studies. The NIST statistical suite, other cryptanalysis publications, and a variety of other statistical analyses are the sources of some of these tests. The following analytical methods are applied to the algorithm: information entropy [17] and correlation analysis between the public and private positions [9].

3.3. Encryption and Decryption Times

We use both ANN and novel techniques to encrypt and decrypt 10 MB of data. We carry out this procedure one hundred times to guarantee statistical significance as shown in table 1 and table 2.

Table 1. Encryption timetable comparison in 512 –bits key length

Experiment	ANN	Obadiah	SHSED	AES	DES
1	0.051	0.295	0.090	0.890	0.070
2	0.054	0.321	0.100	0.899	0.073
3	0.059	0.330	0.080	0.902	0.071
4	0.047	0.294	0.087	0.908	0.069
5	0.050	0.304	0.098	0.910	0.068
24	0.050	0.298	0.083	0.893	0.072
56	0.056	0.311	0.095	0.894	0.070
74	0.052	0.301	0.100	0.896	0.066
94	0.0555	0.312	0.104	0.910	0.071
100	0.050	0.289	0.100	0.890	0.070
Mean	0.0554	0.3114	0.9776	0.8342	0.072
Standard deviation	0.133	0.155	0.165	0.178	0.178

Table 2. Decryption timetable comparison in 512 –bits key length

Experiment	ANN	Obadiah	SHSED	AES	DES
1	0.059	0.330	0.102	0.640	0.077
2	0.060	0.321	0.110	0.636	0.073
3	0.054	0.319	0.109	0.643	0.078
4	0.058	0.320	0.086	0.628	0.081
5	0.060	0.322	0.120	0.642	0.085
24	0.055	0.309	0.980	0.635	0.086
56	0.050	0.330	0.100	0.631	0.077
74	0.056	0.3211	0.9787	0.6342	0.080
94	0.060	0.3198	0.981	0.630	0.088
100	0.050	0.312	0.100	0.639	0.080
Mean	0.0564	0.3214	0.9876	0.6342	0.082
Standard deviation	0.136	0.165	0.169	0.176	0.178

3.4. Security Level

We use 10,000 brute-force attack attempts at unauthorized decryption to determine the security level, and we log the success rate [17, 9, 14].

3.4.1. ANN Method:

- Attempts: 10,000
- Successful Breaches: 1
- Security Level (SL): 0.01% breach probability

3.4.2. Novel Approaches

- Attempts: 10,000
- Successful Breaches: 5
- Security Level (SL): 0.05% breach probability

3.5. Speed time comparison

The time needed for encryption and decryption can be used to gauge the algorithm's speed. For all algorithms, this parameter is measured: Tables 3 and 4 display the ANN, Obaidah and AES.

Table 3. Speed for Encryption in different key lengths [2, 3, 1]

	Key Length			
Algorithm	128-Bits	192-Bits	256-Bits	512-Bits
ANN Technique	0.1713525	0.1813330	0.2066433	0.2577629
Obaidah Technique	0.1814575	0.2014440	0.2377533	0.2972729
AES	0.5500000	0.6043321	0.7132599	0.8918188

Table 4. Speed for Decryption in different key lengths

	Key Length			
Algorithm	128-Bits	192-Bits	256-Bits	512-Bits
ANN Technique	0.2401222	0.2546122	0.2663842	0.2858633
Obaidah Technique	0.2701433	0.2968133	0.2978936	0.3187594
AES	0.5402866	0.5936262	0.5957872	0.6375188

The aforementioned definition of ”Security Level” (fraction of successful brute-force decryption attempts out of 10,000) is illogical in a practical security context because it ignores several important elements that genuinely affect an encryption algorithm's strength: The most important component is key length. An algorithm with a short key (like 40 bits) may have a measurable success rate in 10,000 attempts with modern computers, while a typical modern key (like 256-bit AES) makes success almost impossible within that range. **Attack Model**: The attacker's capabilities, such as the quantity of ciphertext available to them or their processing power (e.g., normal PC vs. supercomputer vs. quantum computer), are not described in the description [10]. Brute Force Method: It assumes impractically that there is a standard method for carrying out the attack. Brute force simply means trying every possible key; the length of this process depends not on a fixed number of attempts, like 10,000, but on the size of the key space and computing power. In real-world cryptography, ”security level” is typically measured in bits of security, which quantifies the number of operations required by the most successful known attack to break the cipher (usually 2^N) operations, where N is the security level in bits, rather than an arbitrary percentage of successful attempts.

4. Results and Discussion

It can be concluded that using the ANN is faster in encryption and decryption than using the novel methods because, according to the paper's numerical example and the results presented numerically through statistical measures, the mean average speed encryption time is 0.05 in the ANN, 0.3 in the Obaidah method, and 0.9 in the SHSED method, while the mean average speed decryption time is 0.06 in the ANN, 0.3 in the Obaidah method, and 0.9 in the SHSED method. Additionally, it may be said that in terms of security level, A breach possibility of 0.01% might be indicated by an ANN, however a higher probability of 0.05% could be suggested by the innovative techniques.

5. Conclusion and Future Work

The quality of the novel approaches in terms of encryption and decryption security was shown by comparing their use with the artificial neural network method. On the other hand, the ANN shown its speed and efficiency in applications as well as being more effective in terms of speed protocol. Undoubtedly, we anticipate research that aims to integrate the benefits of both approaches in the near future. Additionally, we anticipate that the ANN approach will be successful in the future for data greater than 10 MB with excellent security and good encryption and decryption speeds.

Conflict of Interest

The authors declare there is no existing conflict of interest.

References

1.Aliwarge, H. K., Gozali, L., and Nasution, S. R. (2020). Development of artificial intelligence (ai) to improve agriculture, business and education in indonesia by umg idealab. In *Proceedings of the International Conference on Industrial Engineering and Operations Management*, Dubai, UAE.

2.Anderson, C. and Davis, B. (2020). Evaluating security in modern encryption methods. *Journal of Cryptographic Research*, 35(2):77–93.

3.Bai, K. Z. and Fossaceca, J. M. (2025). Em-auc: A novel algorithm for evaluating anomaly based network intrusion detection systems. *Sensors*, 25(1):78.

4.Castro, F., Impedovo, D., Pirlo, G., and Clark, S. (2023). A medical image encryption scheme for secure fingerprint-based authenticated transmission. *Applied Sciences*, 13(10):6099.

5.Gershenson, C. (2003). Artificial neural networks for beginners. arXiv.

6.Haggag, A., Elsharkawy, K., and Said, E. (2020). Utilization of artificial neural networks for fast encryption and decryption of messages. *International Journal of Artificial Neural Networks*, 9(4).

7.Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

8.James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.

9.Jirwan, N., Singh, A., and Vijay, S. (2013). Review and analysis of cryptography techniques. *International Journal of Scientific and Engineering Research*, 3(4):1–6.

10.Jones, L. and Brown, M. (2019). Novel methods in data encryption. *Data Security Journal*, 22(4):456–478.

11.Kumar, R. and Gupta, S. (2020). Comparative analysis of encryption algorithms. *Journal of Information Security*, 33(1):111–125.

12.Lyubashevsky, V. and Seiler, G. (2019). Nttru: Truly fast ntru using ntt. In *IACR Transactions on Cryptographic Hardware and Embedded Systems*, volume 2019, pages 180–201.

13.Martinez, J. and Lopez, G. (2019). Integrating encryption techniques into computer science curriculum. *Educational Computing Research Journal*, 27(2):134–148.

14.Mirzajani, S., Moafimadani, S. S., and Roohi, M. (2024). A new encryption algorithm utilizing dna subsequence operations for color images. *Journal of Applied Mathematics*, 4:1382–1402.

15.Mohammad, O. (2013). A new approach for complex encrypting and decrypting data. *International Journal of Computer Networks & Communications (IJCNC)*, 5(2).

16.Petriu, E. M. (2017). Neural networks basics.

17.Sharma, N., Prabhjot, and Kaur, H. (2017). A review of information security using cryptography technique. *International Journal of Advanced Research in Computer Science*, 8(Special Issue):323–326.

18.Singh, V. (1998). Uniform distribution. In *Entropy-Based Parameter Estimation in Hydrology*, volume 30 of *Water Science and Technology Library*. Springer, Dordrecht.

19.Vidhyayana (2023). A comparative analysis of modern encryption algorithms in cyber defense.

20.Wadho, S. A., Meghji, A. F., Yichiet, A., Kumar, R., and Shaikh, F. B. (2023). Encryption techniques and algorithms to combat cybersecurity attacks: a review. *VAWKUM Transactions on Computer Sciences*, 11(1):295–305.

21.White, P. and Black, T. (2019). Encryption and decryption using neural networks. *IEEE Transactions on Cybernetics*, 47(6):225–233.