

Machine learning of direct propagation neural networks for mobile smart systems

Ivan Tsmots[†], Yurii Opotyak[†], Yurii Lukashchuk^{*,†} and Sofiia Tesliuk[†]

Lviv Polytechnic National University, Stepan Bandera 12, 79013, Lviv, Ukraine

Abstract

A generalized analytical machine learning model for neuro-like data encryption and decryption has been developed. The main components are a neuro network architecture shaper, a weights matrix calculator, and a macropartial product table calculator. The implementation of which reduces setup time. An analysis of recent research and publications on the relevance of problems in the implementation of neuro-like cryptographic data encryption is carried out. The paper formulates rules for the formation of a neuro-network architecture. The structure of a neuro network for cryptographic data encryption is determined by the number of neuro elements. A weights matrix calculator has also been developed. For this purpose, the method of singular value decomposition and the Jacob rotation method were used to find eigenvectors and eigenvalues. A simulation model was developed to demonstrate the operation. A macropartial product table calculator based on the table-algorithmic method was developed. To implement these tasks, C# programming language and Visual Studio 2022 development environment were chosen. Windows Forms was chosen as the development technology. The Accord.Math library was added to the project for operations with matrixes. The practical value is that the developed tools provide fast calculation of coefficients for a given neural network architecture. As a result, the appliance of such generalized analytical model will ensure the speed of computational processes, including data encryption and decryption.

Keywords

neuron-like network; neuro element; macropartial products; tabular-algorithmic method; method of singular decomposition of the matrix; Jacobi rotation method; eigenvectors; eigenvalues

1. Introduction

The importance of secure data encryption is paramount in the modern digital landscape, where sensitive information is frequently transmitted and stored online. Cryptographic techniques are commonly employed to safeguard the confidentiality and integrity of data, but traditional methods can still be susceptible to attacks from hackers and other malicious entities.

Neural-based data encryption is a promising new approach that uses artificial neural networks to encrypt and decrypt data. This approach is based on the principle that neural networks can learn complex patterns and relationships in data, making them well-suited for encryption tasks.

However, one of the challenges in implementing neural-like data encryption is determining the optimal pre-settings for the neural network. This is where the generalized analytical model of pre-setting comes in, as it provides a systematic approach to determine the optimal settings for a given data set and encryption task.

Mobile smart systems refer to the combination of hardware and software solutions integrated into mobile devices, such as smartphones and tablets, which enable them to perform advanced computing, communication, and data processing tasks.

CIAW-2024: Computational Intelligence Application Workshop, October 10-12, 2024, Lviv, Ukraine

* Corresponding author.

[†] These authors contributed equally.

✉ ivan.h.tsmots@lpnu.ua (I. Tsmots); yurii.a.lukashchuk@lpnu.ua (Yu. Lukashchuk); yurii.v.opotyak@lpnu.ua (Yu. Opotyak); sofia.v.tesliuk@lpnu.ua (S. Tesliuk)

ORCID 0000-0002-4033-8618 (I. Tsmots); 0000-0002-8933-8635 (Yu. Lukashchuk); 0000-0001-9889-4177 (Yu. Opotyak); 0009-0005-6512-4447 (S. Tesliuk)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

These systems are highly intelligent, self-contained platforms equipped with sensors, processors, storage, and connectivity features. The primary goal of mobile smart systems is to provide users with seamless interaction with digital services while maintaining security, efficiency, and user experience.

One critical aspect of mobile smart systems is their ability to secure data through encryption. Encryption ensures that data transmitted over networks or stored on mobile devices is protected from unauthorized access. Mobile systems use various encryption protocols to safeguard user information, financial transactions, and communications.

Thus, the development of a generalized analytical model of pre-settings for neural-like cryptographic data encryption is a relevant research topic, as it has the potential to improve the security and efficiency of data encryption in various fields, including finance, healthcare, and public administration.

The object of research is the process of cryptographic encryption of data using neuron-like networks, as well as how weight coefficient matrices precomputation can be used to improve the efficiency and effectiveness of this process.

The subject of the research is a generalized analytical model of weight coefficient matrices precomputation for neuron-like cryptographic data encryption, which is aimed at improving the security and efficiency of cryptographic systems through the usage of neural networks.

The goal of this research is to develop a generalized analytical model for neuro-like cryptographic data encryption designed to enhance both the security and efficiency of encryption when compared to existing methods. Additionally, the study aims to assess the performance of the proposed model through a series of simulations and experiments.

To achieve this goal, the following main tasks of the study are defined:

- analysis of the latest research and publications;
- choosing a Neural Network Architecture Shaper;
- development of a program for calculating the weights matrix based on an improved method of singular value decomposition;
- development of software for calculating the table of macropartial products.

The scientific novelty of the obtained results lies in the development of a generalized model for the precomputation of weights matrices, specifically for implementing neuro-like data encryption.

The core elements of the model include a neuro network architecture generator, a weights matrix calculator, and a macropartial product tables calculator. This proposed approach reduces the time required for the development of neuro-like networks

The practical significance of the results lies in the fact that the developed generalized model for weights matrix precomputation enhances both the security and efficiency of the data encryption process. By leveraging this model, encryption becomes more robust against potential vulnerabilities, while also improving operational efficiency.

2. Analysis of the latest research and publications

Analysis of the trends in the data processing systems development shows increased usage of neural and neuron-like methods [1-3]. Paper [1] elaborates on economic growth prediction based on the artificial neural network algorithm and RBF neural network algorithm by combining the theory for economic forecasting and the characteristics of the BP neural network algorithm with the RBF neural network.

In [2] considered the problem of handling sets of medical data and proposed an improved regression method based on artificial neural networks. Authors in [3] propose accurate performing network evaluation with the use of a graph convolutional network-based performance evaluation method for ultralarge-scale networks with significantly less time-consuming than traditional methods.

One of the trends in the development of embedded systems is the usage of hardware-based implementation of neuron-like networks that requires developing particular components for these cases. A forward propagation channel model was proposed in [4], where adjustment of model weights was achieved by developing supervised learning Tempotron algorithm and oriented on STM32 chips. In [5], field-programmable gate arrays (FPGAs) are used to create hardware neural networks, and a larger neuron-like network can be implemented in this case at a lower cost.

An image cryptosystem based on a non-linear component neuron-like scheme is proposed in [6]. Neuron-like learning algorithms realize a memorable diffusion algorithm and the inputs and weights of the neuron are regulated by the information of the image.

Analysis of publications [7-9] shows that neural network cryptographic data protection is mainly implemented by software. The main disadvantage of such an implementation is the difficulty of providing real-time. In [7] presented an approach to the neuron-like network implementation, oriented on embedded systems for real-time cryptographic data protection with symmetric keys for onboard communication systems in unmanned aerial vehicles because of its suitability for hardware implementation.

Neural cryptography is considered in [8], which is based on neural networks and Vector-Valued Tree Parity Machine (VVTPM), which is a generalized architecture of TPM models proposed by authors. In [9], a neural cryptography based on the complex-valued tree parity machine network (CVTPM) is proposed, where the input, output, and weights of CVTPM are complex values and can be considered as an extension of TPM.

In [10-13], the ways of adaptation of an auto-associative neural network with non-iterative learning to the tasks of cryptographic data protection are considered. Authors in [10] describe an auto-associative neural network concept of soft computing in combination with an encryption technique intended to send data securely on the communication network.

In [11], the AES algorithm was described to secure data bits and also designed as a two-stage encryption and decryption algorithm that can be applied to IoT networks. Study [12] proposes a new PQC neural network intended to map a code-based PQC method to a neural network structure. This approach aimed to enhance the security of ciphertexts based on non-linear activation functions, random perturbation of ciphertexts, and uniform distribution of ciphertexts.

The papers [13-15] analyze the main ways of development of on-board means of cryptographic protection of data transmission in real-time, which showed that a promising way of development is the use of neural network methods for data encryption and decryption. In [13] described the implementation of neuron-like data encryption using polynomials, where the structure of the device was developed with the usage of the base operation of neuron-like data encryption.

The hardware programming language VHDL is used to develop the base operation of data encryption for implementation on the FPGA. A security framework based on cellular automata is proposed in [14], which is composed of three parts: entity authentication, data encryption, and decryption, where authentication is based on a zero-knowledge protocol. System robustness is realized in cases when a shared secret is not only an NP-complete problem but is dynamically transformed by two-dimensional cellular automata into a more complex secret key.

The analysis of the papers shows [16-19] that for the preliminary calculation of the weights coefficients, the principal component method is used, which uses a system of eigenvectors that correspond to the eigenvalues of the covariance matrix of the input data.

In [20] proposed an algorithm for calculating the scalar product of operands in a floating-point format using a table-algorithmic method and an algorithm for forming tables of macropartial products that are used for developing neuron-like networks.

Analysis of the methods for calculating the dot product with weights coefficients, which are known in advance [21-24], showed that one of the most effective methods is the tabular-algorithmic one, which is reduced to the operations of reading macropartial products, addition, and shift.

3. Results of the study and their discussion

3.1. A generalized analytical model of machine learning for neuro-like encryption of data decryption

A neuro-like structure for data protection is proposed, which is focused on the use of the encryption method with symmetric keys. In such a structure, the encryption key and the decryption key are the same. The decryption key can be obtained by mathematical transformations from the encryption key.

Encryption is carried out over blocks of data using a key. The key in the case of using a neuron-like structure consists of a given number of neurons in a neural network N , a matrix of calculated weights W_{ji} .

Next, let's look at the main stages of the data encryption and decryption process.

The first step is to choose a neuron-like network architecture to encrypt and decrypt data. The architecture of a neuron-like network is determined by the number of neurons N , the number of inputs k , and the number of bits of inputs m . Incoming messages that are encrypted can have different bits, n , and be transmitted to a different number of inputs, k , which are equal to the number of neurons N . The bit size of the message n and the number of inputs k determine the architecture of the neural network.

Figure 1 shows the general structure of the neuro-like network used to encrypt data.

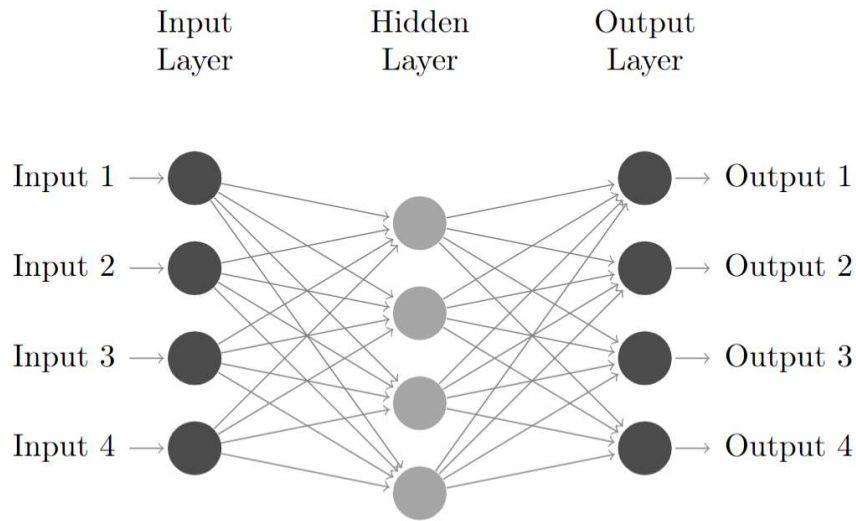


Figure 1: The general structure of the neuro-like network used to encrypt data.

However, the main disadvantage of classical neural networks, which significantly complicates their use in mobile smart systems, is the long learning process.

The proposed neuron-like architecture makes it possible to carry out the process of training a neural network by directly calculating the matrix of weighting coefficients W .

Once the values of the weighting matrix W are calculated, the basic operation of data encryption using such a neuron-like network is reduced to the procedure of multiplying the W weighting matrix by the input data vector \bar{x} according to the following formula:

$$y_j = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1k} \\ W_{21} & W_{22} & \cdots & W_{2k} \\ \vdots & \vdots & \cdots & \vdots \\ W_{N1} & W_{N2} & \cdots & W_{Nk} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} \quad (1)$$

As a result, multiplying the matrix of weighting coefficients W by the input vector is \bar{x} reduced to performing N operations on calculating the dot product.

Performing neuron-like cryptographic protection of data involves making pre-configurations. Such pre-configurations are reduced to the choice of the structure of the neuron-like network, the calculation of a matrix of weights coefficients, and a table of macropartial products. A generalized analytical model of presets is written as follows:

$$P_{Mi} = f_{P_{Mi}}(f_W(f_{(X \rightarrow N_m)})), \quad (2)$$

where P_{Mi} – macropartial product; $f_{P_{Mi}}$ – calculation of the macropartial products table; f_W – calculation of the matrix of weights; $f_{(X \rightarrow N_m)}$ – formation of the structure of a neuron-like network, the parameters of which are determined by the bit width m of the message X and the bit depth of the inputs of the neuroelement n .

3.2. Neural Network Architecture Shaper

The structure of a neuron-like network for cryptographic data encryption is determined by the number of neuro-like elements, which are calculated using the formula:

$$N = \frac{n}{m}, \quad (3)$$

where N – is the number of neuro elements. For data with a bit width of $m = 16$, the number of neuro-like elements N can be 16, 8, 4, and 2 with the number of inputs n 1, 2, 4, and 8, respectively.

3.3. Simulation model for calculating matrices of weights coefficients

The general formula for the Singular Value Decomposition (SVD) method is the next:

$$A = UDV^T, \quad (4)$$

where A is an $N \times n$ input data matrix; U is a left singular $N \times N$ matrix, columns contain eigenvectors of the AA^T matrix; D is a diagonal $N \times n$ matrix containing singular (eigen) values; V is a right singular $N \times n$ matrix, columns contain eigenvectors of the $A^T A$ matrix.

The calculation of eigenvalues and eigenvectors is performed using the Jacobi rotation method, in which the calculation of eigenvalues and eigenvectors of a symmetric matrix is performed iteratively. This process of calculating eigenvectors is known as diagonalization. To construct this sequence, a specially selected rotation matrix J_i is used, the norm of the supra-diagonal part of which is calculated as follows:

$$\|A^{(i)}\|_{off} = \sqrt{\sum_{1 \leq j < k \leq v} (a_{jk}^{(i)})^2}, \quad (5)$$

and decreases with each two-way rotation of the matrix:

$$A^{(i+1)} = J_i^T \cdot A^{(i)} \cdot J_i \quad (6)$$

To calculate the U matrix using the Jacobi method, the result of the AA^T product is passed, and to find the V matrix, the result of the $A^T A$ product is passed. When finding the D matrix, it is enough to take the eigenvalues that were found when calculating the U matrix or the V matrix and place them on the main diagonal. After finding the matrices U , V , and D , the weight coefficients are calculated using the following formula:

$$Aw = UD, \quad (7)$$

where A is an input matrix of dimension $N \times n$, W is a weighting matrix of dimension $n \times n$. The weights matrix W is calculated by the following formula:

$$w = A^{-1} \cdot UD, \quad (8)$$

where the matrix A^{-1} is equal to:

$$A^{-1} = VD^{-1} \cdot U^T \quad (9)$$

To calculate the weights coefficients matrix, the singular value decomposition method was used:

$$W = VD^{-1}U^TUD, \quad (10)$$

where U – a left singular matrix $N \times N$, the columns contain the eigenvectors of the matrix AA^T ; D – a diagonal matrix $N \times n$ containing singular (eigen) values; V – a right singular matrix $n \times n$, the columns contain the eigenvectors of the ATA matrix. A is the input matrix $N \times n$.

Figure 2 presents the table that was used for the simulation. The parameters in this case will be the next: $N=8$, $n = 8$, $m = 2$.

1	1	0	1	1	0	0	1	0	1	1	0	1	0	1	1
1	0	0	0	0	1	1	0	0	1	0	1	1	0	1	1
1	0	1	0	1	0	0	0	1	1	1	0	1	0	1	0
1	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1
0	1	0	0	1	1	0	1	1	1	1	0	0	0	0	1
1	0	0	1	0	0	1	1	0	1	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0
0	0	1	0	0	1	0	1	0	1	0	0	0	1	1	1
1	0	0	1	1	1	0	0	0	1	0	1	0	0	1	0
0	1	1	0	1	1	1	1	1	1	1	1	0	1	0	0
1	0	1	0	0	0	1	1	1	1	0	0	0	0	0	1
0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	1
0	1	1	0	0	0	1	1	0	1	0	0	0	1	0	1

Figure 2: Learning matrix for weights calculation.

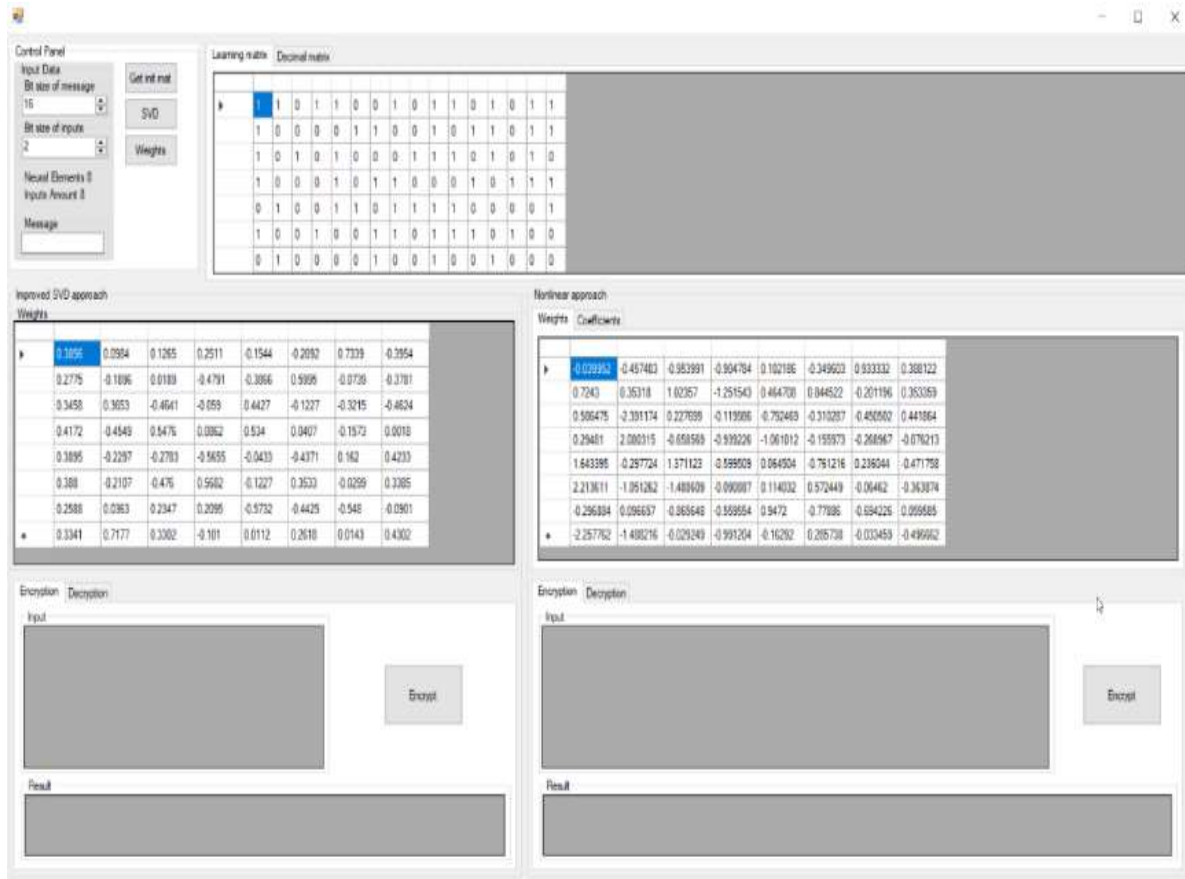


Figure 3: User interface of the simulation model software for calculating weight.

The dimension of the weights tables is determined by the number of neuro-like elements on the basis of which the neuro-like network is synthesized. For example, to encrypt data, neuron-like networks with the number of neuro-like elements of 16, 8, 4, and 2 may be used. The dimensions of the matrices of weights coefficients for such neuron-like networks will be 16×16 , 8×8 , 4×4 , and 2×2 , respectively.

A simulation model software was developed to calculate the neuron weight in the case of cryptographic encryption of incoming data. The software implementation was carried out by C# programming language in the Visual Studio 2022 development environment using the Windows Forms development technology (Fig.3).

The developed simulation model software provides a calculation of a matrix of weights for a given structure of a neuro-like network. The user interface of the simulation model for calculating weighting coefficients is shown.

3.4. Simulation model for calculating macropartial product tables

Calculating macro partial product tables for floating-point weights (where w_j is the mantissa W_j of the weights, and W_j is the order of the weights) involves the following operations:

- determining the largest common order of the weight's coefficients ;
- calculating the order difference for each weighting coefficient W_j ;
- shift to the right of the mantissa w_j by the order difference ;
- determining the maximum number of overflow bits q for macro-partial products PM_i ;
- obtaining scaled mantises by shifting them to the right by q overflow bits of the calculated macro-partial products PM_i ;
- adding the number of overflow bits to the highest common order.

The table of macropartial products is calculated using the following formula:

$$P_{Mi} = \begin{cases} 0, \text{ if } x_{1i} = x_{2i} = x_{3i} = \dots = x_{Ni} = 0 \\ w_1^h, \text{ if } x_{1i} = 1, x_{2i} = x_{3i} = \dots = x_{Ni} = 0 \\ w_2^h, \text{ if } x_{1i} = 0, x_{2i} = 1, x_{3i} = \dots = x_{Ni} = 0 \\ w_1^h + w_2^h, \text{ if } x_{1i} = 1, x_{2i} = 1, x_{3i} = \dots = x_{Ni} = 0 \\ \vdots \\ w_2^h + \dots + w_N^h, \text{ if } x_{1i} = 0, x_{2i} = x_{3i} = \dots = x_{Ni} = 1 \\ w_1^h + w_2^h + \dots + w_N^h, \text{ if } x_{1i} = x_{2i} = x_{3i} = \dots = x_{Ni} = 1 \end{cases}, \quad (11)$$

where $x_{1i}, x_{2i}, x_{3i}, \dots, x_{Ni}$ – table address inputs, w_j^h – the mantissa W_j of the weights is reduced to the largest common order.

The number of macro-partial product tables for encrypting/decrypting commands is equal to the number of neuronal elements in the network. The amount of memory required to store the macro-partial product table is equal to:

$$Q = 2^k, \quad (12)$$

where k is the number of inputs of the neural element.

For neural networks with 16, 8, 4, and 2 neural elements, the number of tables and their sizes are respectively equal: 16 tables with a volume of $Q=2^{16}$.

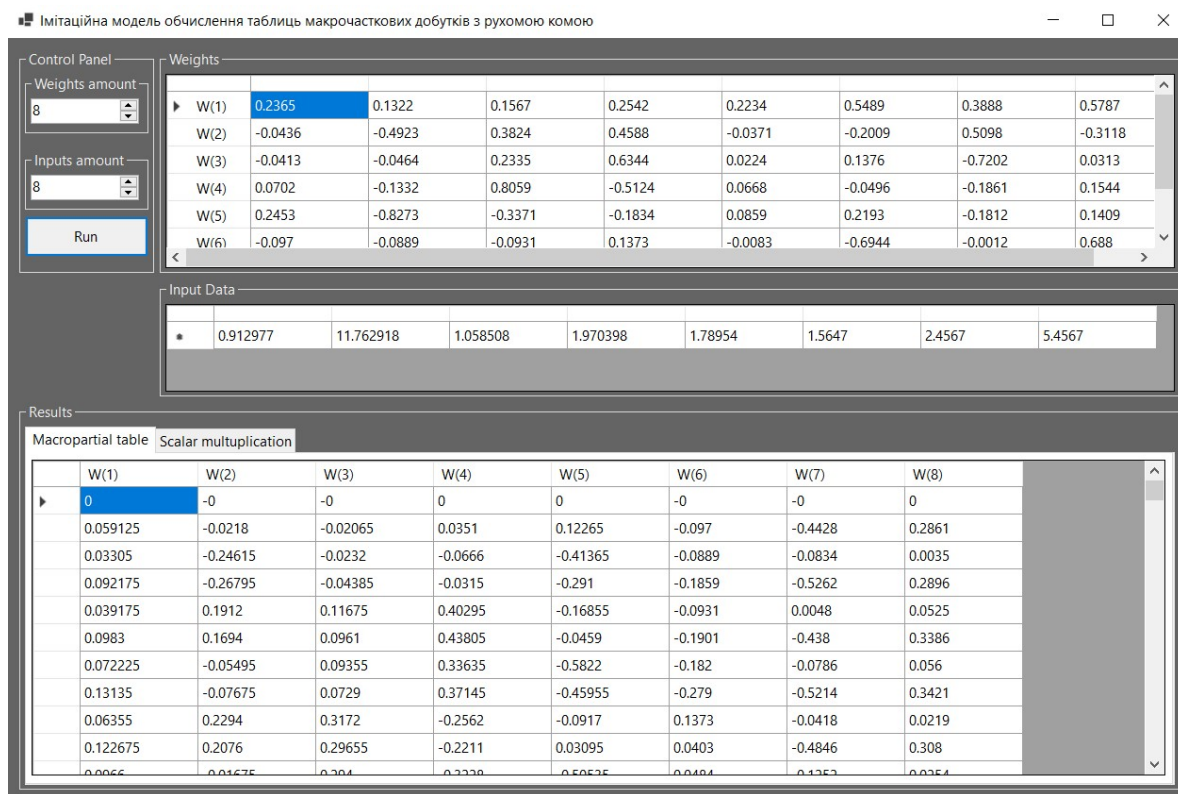


Figure 4: User interface of the simulation model software for calculating the macropartial products table.

On Figure 5 presented the weights matrix used for experiment

0,2365	-0,0436	-0,0413	0,0702	0,2453	-0,097	-0,8856	0,2861
0,1322	-0,4923	-0,0464	-0,1332	-0,8273	-0,0889	-0,1668	0,0035
0,1567	0,3824	0,2335	0,8059	-0,3371	-0,0931	0,0096	0,0525
0,2542	0,4588	0,6344	-0,5124	-0,1834	0,1373	-0,0836	0,0219
0,2234	-0,0371	0,0224	0,0668	0,0859	-0,0083	-0,2139	-0,9438
0,5489	-0,2009	0,1376	-0,0496	0,2193	-0,6944	0,3128	0,0929
0,3888	0,5098	-0,7202	-0,1861	-0,1812	-0,0012	0,0523	0,0134
0,5787	-0,3118	0,0313	0,1544	0,1409	0,688	0,1848	0,1253

Figure 5: Weights matrix used for calculations.

To demonstrate the operation of the program, an example was prepared for working with the following architecture: $m=2$, $k=8$, $N=8$, as well as a matrix of weights coefficients for a neuro-like network with 8 neuro elements.

4. Conclusions

A software model for the implementation of neuro-like data encryption and decryption has been developed. The main components: neural network architecture shaper, the weights matrix calculator, and the macropartial product table calculator.

An imitation model was developed and a user interface for calculating weights matrices for a given neural architecture was presented.

An imitation model and a user interface for calculating macropartial product tables for the table-algorithmic implementation of a scalar product are developed.

The use of the developed models reduces the time of setting up neural networks for computational processes, including data encryption and decryption.

References

- [1] M. Du, "Economic Forecast Model and Development Path Analysis Based on BP and RBF Neural Network," 2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2023, pp. 619-624, doi: 10.1109/CSNT57126.2023.10134678.
- [2] Izonin, I., Tkachenko, R., Dronyuk, I., Tkachenko P., Gregus, M., Rashkevych, M. Predictive modeling based on small data in clinical medicine: RBF-based additive input-doubling method [J]. Mathematical Biosciences and Engineering, 2021, 18(3): 2599-2613. doi: 10.3934/mbe.2021132
- [3] C. Wang, N. Yoshikane and T. Tsuritani, "Usage of a Graph Neural Network for Large-Scale Network Performance Evaluation," 2021 International Conference on Optical Network Design and Modeling (ONDM), Gothenburg, Sweden, 2021, pp. 1-5, doi: 10.23919/ONDM51796.2021.9492331.
- [4] J. Dai and W. Zhang, "Embedded System Implementation of Spiking Neural Network Propagation Model," 2023 5th International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 2023, pp. 187-193, doi: 10.1109/IAECST60924.2023.10503419.
- [5] M. T. Ali and B. H. Abd, "An Efficient area Neural Network Implementation using tan-sigmoid Look up Table Method Based on FPGA", in: 3rd International Conference for Emerging Technology (INCET), Belgaum, India, 2022, pp. 1-7, doi: 10.1109/INCET54531.2022.9825348.
- [6] Y. Zhang and B. Li, "The Memorable Image Encryption Algorithm Based on Neuron-Like Scheme," in IEEE Access, vol. 8, pp. 114807-114821, 2020, doi: 10.1109/ACCESS.2020.3004379
- [7] Tsmots I, Teslyuk V, Łukaszewicz A, Lukashchuk Y, Kazymyra I, Holovatyy A, Opotyak Y. An Approach to the Implementation of a Neural Network for Cryptographic Protection of Data Transmission at UAV. Drones. 2023; 7(8):507. <https://doi.org/10.3390/drones7080507>
- [8] Sooyong Jeong, Cheolhee Park, Dowon Hong, Changho Seo, and Namsu Jho. Neural Cryptography Based on Generalized Tree Parity Machine for Real-Life Systems. Security and Communication Networks. Volume 2021. Article ID 6680782 | <https://doi.org/10.1155/2021/6680782>
- [9] T. Dong and T. Huang, "Neural cryptography based on complex-valued neural network," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 11, 2019.
- [10] P. Saraswat, K. Garg, R. Tripathi and A. Agarwal, "Encryption Algorithm Based on Neural Network," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-5, doi: 10.1109/IoT-SIU.2019.8777637.
- [11] A. Sarin, D. Thanawala, S. Verma and C. Prakash, "Implementation of New Approach to Secure IoT Networks with Encryption and Decryption Techniques," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225279.
- [12] A. C. H. Chen, "Post-Quantum Cryptography Neural Network," 2023 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES), Tumakuru, India, 2023, pp. 1-6, doi: 10.1109/ICSSES58299.2023.10201083.
- [13] I. Tsmots, R. Tkachenko, V. Teslyuk, V. Rabyk and Y. Opotyak, "Development of a Device on FPGA to Implement the Base Operation of Neural-like Data Encryption Using Polynomials," 2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT), Lviv, Ukraine, 2023, pp. 1-5, doi: 10.1109/CSIT61576.2023.10324267.
- [14] Corona-Bermúdez, E., Chimal-Eguía, J. C., & Téllez-Castillo, G. (2022). Cryptographic Services Based on Elementary and Chaotic Cellular Automata. Electronics, 11(4), 613. <https://doi.org/10.3390/electronics11040613>
- [15] I. Tsmots, V. Rabyk, R. Tkachenko, Y. Opotyak and V. Teslyuk, "Implementation of Base Components of Neuro-like Cryptographic Data Protection Systems on FPGA," 2023 IEEE 13th International Conference on Electronics and Information Technologies (ELIT), Lviv, Ukraine, 2023, pp. 1-6, doi: 10.1109/ELIT61488.2023.10310958.

- [16] Q. Xie et al., "Research on Inversion Algorithm of Aerosol Extinction Coefficient Based on Elman Neural Network," 2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA), Chengdu, China, 2021, pp. 62-66, doi: 10.1109/ICIEA51954.2021.9516085.
- [17] Y. Kuroe, H. Iima and Y. Maeda, "Four Models of Hopfield-Type Octonion Neural Networks and Their Existing Conditions of Energy Functions," 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020, pp. 1-7, doi: 10.1109/IJCNN48605.2020.9206838.
- [18] G. Zhang, K. Niwa and W. B. Kleijn, "Projected Weight Regularization to Improve Neural Network Generalization," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 4242-4246, doi: 10.1109/ICASSP40776.2020.9054133.
- [19] B. Tao, M. Xiao, W. X. Zheng, J. Cao and J. Tang, "Dynamics Analysis and Design for a Bidirectional Super-Ring-Shaped Neural Network With n Neurons and Multiple Delays," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 7, pp. 2978-2992, July 2021, doi: 10.1109/TNNLS.2020.3009166.
- [20] I. Tsmots, V. Rabyk, V. Teslyuk and Y. Opotyak, "Floating-Point Number Scalar Product Hardware Implementation for Embedded Systems," 2023 17th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Jaroslaw, Poland, 2023, pp. 6-10, doi: 10.1109/CADSM58174.2023.10076502.
- [21] R. Tkachenko and I. Izonin, "Model and Principles for the Implementation of Neural-Like Structures based on Geometric Data Transformations", Advances in Computer Science for Engineering and Education. ICCSEEA2018. Advances in Intelligent Systems and Computing, vol. 754, pp. 578-587, 2019.
- [22] M. Bharathi and Y. J. M. Shirur, "Efficiency Evaluation of Scalable Multiply and Accumulate Architectures in DSP: A Comparative Study of LUT Based and LUT-Less Based Approaches," 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bangalore, India, 2024, pp. 1-5, doi: 10.1109/IITCEE59897.2024.10467265.
- [23] R. Rohit, S. Dudeja and M. Rao, "VLUT: Design and Evaluation of Variable band LUT to realize Activation Functions," 2023 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Istanbul, Turkiye, 2023, pp. 1-4, doi: 10.1109/ICECS58634.2023.10382912.
- [24] M. Ebrahimi, R. Sadeghi and Z. Navabi, "LUT Input Reordering to Reduce Aging Impact on FPGA LUTs," in IEEE Transactions on Computers, vol. 69, no. 10, pp. 1500-1506, 1 Oct. 2020, doi: 10.1109/TC.2020.2974955.