НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Лукащук Юрій Андрійович

Кваліфікаційна наукова праця
на правах рукопису
УДК 004.422.81

Інформаційна технологія захисту даних у реальному часі для мобільних смарт-систем з використанням нейроподібних мереж

122 – комп'ютерні науки

Дисертація на здобуття наукового ступеня
доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело
_____/Ю. А. Лукащук/

Науковий керівник

Цмоць Іван Григорович,

доктор технічних наук, професор

Львів – 2023

АНОТАЦІЯ

Лукащук Ю.А. Інформаційна технологія захисту даних у реальному часі для мобільних смарт-систем з використанням нейроподібних мереж. Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» – Національний університет «Львівська політехніка», Львів, 2023.

Зміст анотації.

Дисертацію присвячено розробці нових і вдосконаленню існуючих методів, моделей та програмно-апаратних засобів інформаційної технології нейроподібного захисту даних у реальному часі з високими техніко-експлуатаційними характеристиками для мобільних смарт-систем.

У вступі обґрунтовано актуальність теми дисертаційної роботи, сформульовано мету та основні задачі досліджень, визначено наукову новизну роботи і практичне значення отриманих результатів, показано зв'язок роботи з науковими темами. Подано відомості про апробацію результатів роботи й особистий внесок автора та його публікації.

У вступі пояснено актуальність теми дисертаційної роботи, сформульовано основні задачі та мету дослідження, визначено практичне значення отриманих результатів й наукову новизну роботи, описано зв'язок роботи з науковими темами. Подано інформацію про апробацію результатів роботи а особистий внесок автора, а також його публікації.

У першому розділі дисертації проведено аналіз архітектур нейронних мереж, а також проаналізовано методи та алгоритми навчання у результаті чого орієнтовано задачі нейромережевого шифрування-дешифрування даних нейроподібної мережі прямого поширення автоасоціативного типу на основі парадигми «модель послідовних геометричних перетворень» шляхом неітеративного обчислення вагових коефіцієнтів, що забезпечило повторюваність результатів і орієнтацію на апаратну реалізацію.

Проведено аналіз елементної бази для реалізації нейронних мереж реального часу [53-57]. У результаті вибрано напрям вдосконалення методу вибору елементної бази для імітаційної моделі, який забезпечує автоматизацію

процесу вибору найефективнішої елементної бази для синтезу засобів криптографічного захисту даних у реальному часі.

Проаналізовано загальні вимоги та основні напрями вдосконалення засобів реалізації нейронних мереж реального часу для робототехнічних систем.

У другому розділі вдосконалено і орієнтовано на задачі нейромережевого шифрування-дешифрування даних нейроподібну мережу прямого поширення автоасоціативного типу на основі парадигми «модель послідовних геометричних перетворень».

Запропоновано синтез системи захисту та передавання даних (далі – СЗПД) у реальному часі з високими техніко-економічними характеристиками із використанням інтегрованого підходу.

Вибрано для синтезу СЗПД у реальному часі такі принципи: конвеєризації та просторового паралелізму; модульності; програмованості архітектури блоків кодування-декодування і шифрування-дешифрування даних за допомогою використання програмованих логічних інтегральних мікросхем; змінності складу обладнання; спеціалізації та адаптації апаратно-програмних засобів до структури алгоритмів нейроподібного шифрування та дешифрування даних; відкритості програмного забезпечення.

Розроблено структуру СЗПД з використанням шумоподібних кодів, яка, за рахунок можливості налаштування кодів маскування та вагових коефіцієнтів, програмованості архітектури нейроподібної мережі та генерації шумоподібних кодів різної розрядності, забезпечує високу завадостійкість, роботу у режимі реального часу та високі техніко-економічні характеристики.

Розроблено модель попередніх налаштувань для реалізації нейроподібного шифрування/дешифрування даних, основними компонентами якої є: блок обчислення таблиць макрочасткових добутків, блок формування архітектури нейроподіної мережі, а також блок обчислення матриць вагових коефіцієнтів. Разом реалізації цих блоків забезпечує зменшення часу для налаштувань.

Розроблено модель нейроподібного шифрування даних та команд управління за допомогою таблично-алгоритмічного методу, де основними

компонентами є: формувач адреси читання з таблиць, N суматорів і комутатор, перетворювач повідомлення, N таблиць макрочасткових добутків. Реалізація такої моделі забезпечує можливість тестування СЗПД у реальному часі.

У третьому розділі розроблено інформаційну технологію нейроподібного криптографічного захисту даних у реальному часі із симетричними ключами (матриці вагових коефіцієнтів, архітектура нейроподібної мережі та коди маскування), яка за рахунок динамічної зміни архітектури нейроподібної мережі та попереднього обчислення матриць вагових коефіцієнтів забезпечує апаратно-програмну реалізацію з високими техніко-економічними характеристиками та високу криптографічну стійкість.

Запропоновано розроблення інформаційної технології нейроподібного криптографічного захисту даних у реальному часі здійснювати на базі інтегрованого підходу, який охоплює: дослідження [52, 71, 83, 84] та розроблення теоретичних основ нейроподібного криптографічного захисту даних; дослідження та розроблення нових алгоритмів та структур нейроподібного шифрування та дешифрування даних, орієнтованих на сучасну елементну базу.

Визначено, що основними критеріями вибору елементної бази для синтезу засобів криптографічного захисту даних у реальному часі є: вартість, продуктивність, обсяг пам’яті, потужність енергоспоживання, коефіцієнт технологічності, температурний діапазон роботи, маса, габаритні показники, показниками надійності, коефіцієнт вібростійкості.

Вдосконалено метод вибору елементної бази для синтезу засобів криптографічного захисту даних у реальному часі, який за рахунок обчислення інтегрованої оцінки ефективності елементної бази та врахування вимог конкретного застосування забезпечує вибір найефективнішої елементної бази із множини елементних баз, які відповідають вимогам технічного завдання.

У четвертому розділі дисертаційної роботи на основі вдосконаленого методу вибору елементної бази розроблено імітаційну модель, яка за рахунок використання наповненої сучасною елементною базою бази даних і врахування вимог технічного завдання забезпечує автоматизацію процесу вибору

найефективнішої елементної бази для синтезу засобів криптографічного захисту даних у реальному часі.

Розроблено імітаційну модель знаходження вагових коефіцієнтів для заданої архітектури нейромережі. Роботоздатність такої моделі було показано на прикладі повідомлення 1100100000011110. Розрядність такого повідомлення – 16 (n=16), а розрядність входів – 2 (m=2). При цьому кількість нейроелементів є 8 (N=8) та кількість входів є 8 (k=8).

Удосконалено метод сингулярного розкладу матриці для знаходження матриці вагових коефіцієнтів.

Ключові слова: криптографічний захист даних, архітектура нейроподібної мережі, нейроелементи, матричні перетворення, метод обертання Якобі, шифрування даних, програмна реалізація.

ABSTRACT

Lukashchuk Yu.A. Information technology for real-time data protection for mobile smart systems using neural networks. Dissertation for obtaining the scientific degree of Doctor of Philosophy in specialty 122 «Computer Science» – Lviv Polytechnic National University, Lviv, 2023.

Abstract content. The dissertation is devoted to the development of new and improvement if existin methods, models, and software and hardware of information technology of neuro-like data protection in real time with high technical and operational characteristics for mobile smart systems.

The introduction explains the relevance of the topic of this dissertation, formulates the goal and main tasks of the research, defines the scientific novelty of this work and practical significance of the obtained results, show the connection of the work with scientific topics. Information about the approval of the results of the work and the personal contribution of the author and his publication is provided.

In the first chapter of the dissertation research, the analysis of the architectures of neural networks was carried out, as well as the methods and learning algorithms were analyzed, as a result of which the problems of neural network encryption-decryption of data of a neural network of direct propagation of the auto-associative type were oriented on the basis of the "model of successive geometric transformations" paradigm by means of non-iterative calculation of weighting coefficients, which ensured repeatability of results and orientation to hardware implementation.

The analysis of the element base for the implementation of real-time neural networks was carried out [53-57]. As a result, the direction of improvement of the method of selecting an elemental base is chosen, a simulation model that provides automation of the process of selecting the most effective elemental base for the synthesis of means of cryptographic protection of data in real time.

The general requirements and main directions of improvement of the means of implementing real-time neural networks for robotic systems are analyzed.

In the second chapter, a neuro-like network of direct propagation of the auto-associative type based on the paradigm «model of successive geometric

transformations» is improved and focused on the tasks of neural network encryption-decryption of data.

It is proposed to carry out the synthesis of data protection and transmission systems (DPTS) in real time with high technical and economic characteristics using an integrated approach

The following principles were chosen for the synthesis of DPTS in real time: variability of equipment composition; modularity; pipeline and spatial parallelism; software openness; specialization and adaptation of hardware and software tools to the structure of neuro-like data encryption and decryption algorithms; programmability of the architecture of data encryption-decryption and encoding-decoding blocks by using programmable logic integrated circuits.

The structure of DPTS using noise-like codes has been developed, which due to the possibility of setting masking codes and weighting coefficients, the programmability of the neural network architecture and the generation of noise-like codes of different bit rates provides high immunity, real-time operation and high technical and economic characteristics.

A model of preliminary settings for the implementation of neural-like encryption/decryption of data has been developed, the main components of which are the shaper of the neural-like network architecture, the calculator of weighting coefficients matrices, and the calculator of tables of macroparticle products, the implementation of which ensures a reduction in setup time.

A model of neural-like encryption of data and control commands has been developed using a tabular-algorithmic method, the main components of which are a message converter, a generator of reading addresses from tables, N tables of macroparticle products, N adders, and a switch, the implementation of which provides real-time testing of DPTS.

In the third section, the information technology of neural-like cryptographic protection of data in real time with symmetric keys (masking codes, neural-like network architecture and weighting coefficients matrices) is developed, which due to the pre-calculation of weighting coefficients matrices and dynamic changes of the neural-like

network architecture ensures high cryptographic stability and hardware- software implementation with high technical and economic characteristics.

It is proposed to develop the information technology of neuro-like cryptographic data protection in real time on the basis of an integrated approach, which includes: research and development of the theoretical foundations of neuro-like cryptographic data protection; research and development of new algorithms and structures of neuro-like encryption and decryption of data, oriented on a modern element base.

It was determined that the main criteria for choosing an element base for the synthesis of real-time cryptographic data protection tools are: cost, performance, memory capacity, power consumption, manufacturability factor, operating temperature range, weight, overall parameters, reliability indicators, vibration resistance factor.

The method of selecting an element base for the synthesis of cryptographic data protection in real time has been improved, which by calculating an integrated evaluation of the effectiveness of the element base and taking into account the requirements of a specific application ensures the selection of the most effective element base from a set of element bases that meet the requirements of the technical task.

In the fourth chapter of the dissertation, a simulation model was developed on the basis of the improved method of selecting an element base, which, due to the use of a database filled with a modern element base and taking into account the requirements of the technical task, provides automation of the process of selecting the most effective element base for the synthesis of cryptographic data protection tools in real time.

A simulation model for finding weighting coefficients for a given neural network architecture has been developed. The performance of this model was demonstrated on the example of the message 1100100000011110. The bit rate of the message in this case is 16 (n=16), the bit rate of inputs is 2 (m=2), the number of neuroelements is 8 (N=8), the number of inputs is 8 (k= 8).

The method of singular decomposition of the matrix for finding the matrix of weighting coefficients has been improved.

Keywords: cryptographic data protection, neural network architecture, neural elements, matrix transformations, Jacobi rotation method, data encryption, software implementation.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Стаття у науковому періодичному виданні іншої держави:

1. Tsmots I., Teslyuk V., Teslyuk T., Lukashchuk Y. The Method and Simulation Model of Element Base Selection For Protection System Synthesis and Data Transmission. *International Journal of Sensors, Wireless Communications and Control*. Vol. 11. Issue 5. 2021. DOI: 10.2174/2210327910999201022194630.
2. Tsmots I., Teslyuk V., Khavalko V., Lukashchuk Y. Use of Augmented Reality Technology to Develop an Application for Smart Factory Workers. *Proceedings of the 1st International Workshop on Digital Content & Smart Multimedia* (DCS Mart). 2019. Vol. 2533. P. 23–34.

Авторський твір:

3. Цмоць І. Г., Теслюк В. М., Опотяк Ю. В., Лукащук Ю. А. Програмний засіб розрахунку вагових коефіцієнтів для заданої архітектури нейроподібної мережі. Свідоцтво № 114363.

Статті у наукових фахових виданнях України:

4. Лукащук Ю. А. Імітаційна модель обчислення вагових коефіцієнтів для нейроподібного шифрування та дешифрування даних. *Науковий вісник НЛТУ України.* Львів, 2022. Випуск 6.
5. Цмоць І. Г., Лукащук Ю. А. Узагальнена аналітична модель попередніх налаштувань для нейроподібного криптографічного шифрування даних. *Науковий вісник НЛТУ України*. Львів, 2023. Том 33. № 2. URL: https://doi.org/10.36930/40330211
6. Цмоць І. Г., Лукащук Ю. А., Хавалко В. М., Рабик В. Г. Моделі нейроподібного елемента паралельно-паралельного типу. *Моделювання та інформаційні технології.* Київ, 2019. Вип. 86. С. 119–126.
7. Tsmots I., Rabyk V., Lukaschuk Y. Development of Mobile Facilities of Neuro-like Cryptographic Encryption and Decryption of Data in Real Time. *Series of Computer Sciences and Information Technologies*. Lviv, 2021. Vol. 9. P. 84–95.

8. Цмоць І. Г., Лукащук Ю. А., Ігнатєв І. В., Казимира І. Я. Компоненти апаратних нейронних мереж узгодженого паралельно-вертикального оброблення даних у реальному часі. *Український журнал інформаційних технологій.* Львів, 2021. Випуск 3. С. 63–72. URL: https://doi.org/10.23939/ujit2021.03.063.
9. Цмоць І. Г., Опотяк Ю. В., Різник О. Я., Березький О. М., Лукащук Ю. А. Архітектура та реалізація базових компонентів системи нейромережевого захисту і кодування передачі даних. *UJIT*. 2022. Випуск 4. № 1. С. 53–62. URL: https://doi.org/10.23939/ujit2022.01.053.
10. Tsmots I., Teslyuk V., Lukashchuk Y., Opotiak Y. Method of training and implementation on the basis of neural networks of cryptographic data protection. *COLINS*. 2022. Vol. 3171. P. 916–928.

Матеріали та тези конференцій:

11. Лукащук Ю. А. Використання технології доповненої реальності для оцінки економічних ризиків функціонування підприємства. *Інформаційні технології в освіті та практиці* : збірник наукових статей за матеріалами доповідей Всеукраїнської науково-практичної конференції 20 грудня 2019 року / упорядник Т. В. Магеровська. Львів : ЛьвДУВС, 2019. 246 с. С. 192–195.
12. Лукащук Ю. А., Лукащук Д. А. Хмарні сервіси для онлайн навчання. *Інформаційні технології в освіті та практиці* : збірник наукових статей за матеріалами доповідей Всеукраїнської науково-практичної конференції 18 грудня 2020 року ; упорядник Т. В. Магеровська. Львів : ЛьвДУВС, 2020. 140 с. С. 135–136.
13. Лукащук Ю. А., Лукащук Д. А. До питання цифровізації підприємств. *Менеджмент і безпека: теоретичні та прикладні аспекти* : матеріали науково-практичної інтернет-конференції (з міжнародною участю), приуроченої 15-річчю до дня створення кафедри менеджменту (м. Львів, 12 травня 2021 р.) ; упоряд. Г. З. Леськів. Львів : ЛьвДУВС, 2021. 304 с. С. 177–180.

14. Tsmots I., Rabyk V., Berezky O., Lukaschuk Y., Teslyuk V. Development of Modules of Neuro-Like Cryptographic Encryption and Decryption of Data and their Implementation on FPGA. *2021 IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*. DOI: 10.1109/CADSM52681.2021.9385228. Lviv, 2021. P. 53–57.
15. Лукащук Ю. А. Розрахунок вагових коефіцієнтів для криптографічного захисту під час передачі даних у реальному часу. *Інформаційні технології в освіті та практиці* : збірник наукових статей за матеріалами доповідей Всеукраїнської науково-практичної конференції 17 грудня 2021 року ; упорядник Т. В. Магеровська. Львів : ЛьвДУВС, 2021.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

СЗПД – Системи захисту та передавання даних

ПЛІС – Програмовані логічні інтегральні схеми

CPLD – Complex Programmable Logic Devices

SPLD – Simple Programmable Logic Devices

SoC – System-on-a-chip

CPU – Central processing unit

GPU – Graphics processing unit

FPGA – Field Programmable Gate Arrays

SVD – Singular Value Decomposition

ЦОС – Цифрова обробка сигналів

ВСТУП

Актуальність теми. Сучасний етап розвитку інформаційних технологій криптографічного захисту даних характеризується розширенням галузей застосування, значна кількість з яких орієнтована на використання у мобільних смарт-системах. У таких смарт-системах вимагається шифрування та дешифрування даних у реальному часі на апаратно-програмних засобах, які задовольняють обмеження щодо енергоспоживання та габаритів, а також часу й вартості необхідних для розробки. Для того щоб створити такі апаратно-програмні засоби інформаційних технологій криптографічного захисту необхідне широке використання сучасної елементної бази (програмованих логічних інтегральних схем (ПЛІС) типу FPGA, мікроконтролерів тощо) та розробки нових методів, алгоритмів і структур для реалізації алгоритмів криптографічного шифрування та дешифрування даних. Через це особливої актуальності набуває проблематика розробки нових і вдосконалення існуючих методів та апаратно-програмних засобів інформаційних технологій криптографічного захисту для мобільних смарт-систем, які забезпечать високі техніко-експлуатаційні показники.

Одним із шляхів розроблення таких апаратно-програмних засобів криптографічного захисту є використання автоасоціативної нейромережі прямого поширення, що навчається на основі методу головних компонентів. Як особливість таких нейромереж можна відзначити можливість використовувати таблиці макрочасткових добутків також наперед обчислювати вагові коефіцієнти та викорстовувати базис елементарних арифмеичних операцій для реалізації нейроподібних елементів. На основі зазначених нейроподібних елементів синтезується нейроподібна мережа, яка забезпечує шифрування та дешифрування даних. Реалізація нейроподібних засобів шифрування та дешифрування даних з високими техніко-експлуатаційними показниками досягається шляхом використання проблемно-орієнтованого підходу, який передбачає поєднання програмних і апаратних засобів. Процес взаємопроникнення програмного

(універсального) і апаратного (спеціалізованого) забезпечує високу ефективність використання обладнання та зменшує час їх розробки.

З наведеного випливає, що розроблення інформаційної технології нейроподібного лінійного захисту даних у реальному часу в мобільних смарт-системах найефективніше створювати саме за інтегрованим підходом, який охоплює засоби, методи та моделі шифрування та дешифрування даних, сучасну елементну базу, методи розпаралелювання, автоматизації процесу програмування й засоби автоматизованого проектування.

Розроблена у цій роботі інформаційна технологія нейроподібного лінійного захисту даних у реальному часі для мобільних смарт-систем ґрунтується на дослідженнях відомих зарубіжних та українських вчених, які напрацювали теоретичні та практичні засади її побудови, зокрема, К. Аудххаси (K. Audhkhasi), О. Особа (O. Osoba), Б. Кошко (B. Kosko) [32], В. (V. Shandor), С. Шевляков, Р. Ткаченко [27, 107, 137], О. Скубишевський та інші.

Проблематиці шифрування та дешифрування даних присвячені праці Е. Корона-Бермудес (Corona-Bermúdez E), Х. К. Чималь-Егіа (Chimal-Eguía J. C.), Х. Тельес-Кастильо (Téllez-Castillo G.) [30].

Крім того, теорією нейронних досліджень займалися В. Литвиненко, В. Войцік, А Фефелов, І. Лур`є [45], В. Коцовський, А. Батюк [106], С. Джонг (S. Jeong), Ч. Пак (Cheolhee Park) [25], Т. Донг (T. Dong), Т. Хуанг (T. Huang) [26], K. Hinkelmann [33], а також ще низка науковців [35, 37, 38, 46, 47, 48, 50, 56, 93].

Отже, актуальною науковою задачею є розроблення нових і вдосконалення існуючих методів, моделей та програмно-апаратних засобів інформаційної технології нейроподібного лінійного захисту даних у реальному часі з високими техніко-експлуатаційними характеристиками для мобільних смарт-систем.

Зв’язок роботи з науковими програмами, планами та темами. Дисертаційне дослідження проводилось згідно з планами науково-дослідних та навчальних робіт кафедри автоматизованих систем управління Національного університету «Львівська політехніка», у тому числі в межах держбюджетних науково-дослідних робіт: «Експериментальна система нейромережевого

криптографічного захисту та передачі даних у реальному часі з використанням баркероподібних кодів» (номер держ. реєстр. 0121U109503) і «Експериментальна мобільна робототехнічна платформа з інтелектуальною системою управління та захистом передачі даних» (номер держ. реєстр. 0122U000891).

Мета і завдання дослідження. Метою дисертаційного дослідження є розробка нових та вдосконалення існуючих методів, моделей та програмно-апаратних засобів інформаційної технології нейроподібного лінійного захисту даних у реальному часі з високими техніко-експлуатаційними характеристиками для мобільних смарт-систем.

Досягнення поставленої мети передбачало розв'язання таких завдань:

1. аналіз методів, алгоритмів і засобів нейромережевого захисту даних у мобільних смарт системах та визначення напрями їх розвитку;
2. вдосконалити нейроподібну мережу прямого поширення автоасоціативного типу на основі парадигми «модель послідовних геометричних перетворень» і адаптувати її до нейроподібного шифрування-дешифрування даних;
3. вдосконалити метод обчислення вагових коефіцієнтів для різних архітектур нейроподібних мереж;
4. розробити метод таблично-алгоритмічного обчислення скалярного добутку з плаваючою комою в нейроподібних елементах;
5. розробити модель попередніх налаштувань для реалізації нейроподібного шифрування/дешифрування даних;
6. розробити інформаційну технологію нейроподібного лінійного захисту даних у реальному часі з симетричними ключами для смарт-систем;
7. вдосконалити метод вибору елементної для синтезу засобів криптографічного захисту даних у реальному часі;
8. розробити апаратно-програмні засоби інформаційної технології нейроподібного лінійного захисту даних у реальному часі з симетричними ключами для смарт-систем.

Об’єктом дослідження є процеси обчислення матриць вагових коефіцієнтів та скалярного добутку з плаваючою комою, нейроподібного шифрування та дешифрування даних і вибору елементної бази.

Предметом дослідження є моделі, методи, алгоритми та програмно-апаратні засоби інформаційної технології нейроподібного лінійного захисту даних у реальному часі.

Методи дослідження. У дисертаційній роботі використано: парадигму «модель послідовних геометричних перетворень» – для розроблення нейроподібних мереж; метод головних компонент – для обчислення вагових коефіцієнтів; таблично-алгоритмічний метод – для обчислення скалярного добутку; теорію та методи розпаралелення алгоритмів, теорію автоматизованого проектування та автоматизації процесу програмування – для розроблення апаратно-програмних засобів інформаційної технології нейроподібного лінійного захисту даних у реальному часі.

Наукова новизна отриманих результатів. За результатами виконаних теоретичних та експериментальних досліджень розв’язано актуальну наукову задачу, а саме розроблення нових та вдосконалення існуючих методів, моделей та програмно-апаратних засобів інформаційної технології нейроподібного лінійного захисту даних у реальному часі з високими техніко-експлуатаційними характеристиками для мобільних смарт-систем. При цьому отримано такі нові результати:

вперше розроблено:

– інформаційну технологію нейроподібного захисту даних у реальному часі із симетричними ключами (архітектура нейромережі, матриці вагових коефіцієнтів та коди маскування) для смарт-систем, яка за рахунок використання моделі попередніх налаштувань, вдосконаленого методу обчислення та вагових коефіцієнтів, методу таблично-алгоритмічного обчислення скалярного добутку забезпечує високу криптографічну стійкість та апаратно-програмну реалізацію з високими техніко-експлуатаційними характеристиками;

– модель попередніх налаштувань для реалізації нейроподібного шифрування та дешифрування даних основними компонентами якої є блок обчислення матриць вагових коефіцієнтів, блок формування архітектури нейроподібної мереі та блок обчислення таблиць макрочасткових добутків;

– метод таблично-алгоритмічного обчислення скалярного добутку з плаваючою комою в нейроподібних елементах, який за рахунок приведення до найбільшого спільного порядку вхідних даних, вагових коефіцієнтів і використання таблиць макрочасткових добутків мантис забезпечує зменшення часу обчислення скалярного добутку та підвищення ефективності використання обладнання;

вдосконалено:

– нейроподібну мережу прямого поширення автоасоціативного типу на основі парадигми «модель послідовних геометричних перетворень» і адаптовано її до нейроподібного шифрування-дешифрування даних шляхом попереднього обчислення вагових коефіцієнтів, що забезпечило повторюваність результатів і орієнтацію на апаратну реалізацію;

– метод вибору елементної бази для синтезу засобів криптографічного захисту даних у реальному часі, який за рахунок використання наповненої сучасною елементною базою бази даних і врахування вимог технічного завдання забезпечує автоматизацію процесу вибору найефективнішої елементної бази;

– метод обчислення вагових коефіцієнтів, який за рахунок використання сингулярного розкладу матриць, з використанням алгоритму Якобі для знаходження власних значень та власних векторів, забезпечує швидке налаштування та обчислення матриць вагових коефіцієнтів для різних архітектур нейроподібних мереж.

Практичне значення отриманих результатів. Розроблена інформаційна технологію нейроподібного захисту даних у реальному часі з симетричними ключами (архітектура нейромережі, матриці вагових коефіцієнтів та коди маскування) для смарт-систем дає змогу:

– розробляти апаратно-програмні засоби нейроподібного шифрування/дешифрування даних у реальному часі із високими техніко-експлуатаційними характеристиками;

– зменшити час обчислення скалярного добутку з плаваючою комою в нейроподібних елементах;

– підвищити ефективність використання обладнання при реалізації нейроподібного елемента та нероподібної мережі;

– вибрати найефективнішу елементну базу для синтезу засобів криптографічного захисту даних у реальному часі;

– зменшити час налаштування нейроподібної мережі для реалізації нейроподібного шифрування та дешифрування даних.

Особистий внесок здобувача. Дисертація є самостійною науковою працею, в цій роботі автором особисто розроблено нові наукові ідеї та результати, що дозволили вирішити наукове завдання розроблення нових і вдосконалення існуючих методів, моделей та програмно-апаратних засобів інформаційної технології нейроподібного лінійного захисту даних у реальному часі з високими техніко-експлуатаційними характеристиками для мобільних смарт-систем.

Робота містить практичні і теоретичні положення, а також висновки, які були сформульовані дисертантом особисто. Гіпотези, положення чи ідеї інших авторів, що присутні в дисертації, мають відповідні посилання та використовуються лише для підкріплення результатів й ідей здобувача.

Усі наукові результати практичних і теоретичних досліджень, що представлені у дисертації, були одержані автором особисто. 1 стаття опублікована одноосібно. У працях, які здійснювались у співавторстві, здобувачеві належать: програмне забезпечення, результати та метод знаходження вагових коефіцієнтів.

Апробації результатів дисертації. Основні теоретичні положення і практичні результати дисертаційного дослідження обговорювались та доповідались на наступних конференціях: Міжнародній конференції з цифрового контенту та розумної мультимедії («1st International Workshop on Digital Content & Smart Multimedia»), DCSMart, (Львів, 2019); Міжнародній конференції «Досвід

проектування та застосування систем САПР» («International Conference on the Experience of Designing and Application of CAD Systems»), CADSM, (Львів, 2021); Міжнародній конференції з електроніки та інформаційних технологій («International Conference on Electronics and Information Technologies»), ELIT, (Львів, 2021); Міжнародній конференції з комп'ютерної лінгвістики та інтелектуальних систем («International Conference on Computational Linguistics and Intelligent System»), CoLIns, (Львів, 2022), а також на наукових семінарах кафедри автоматизованих систем управління Національного університету «Львівська політехніка» (2021-2023).

Публікації. У 14 наукових публікаціях повністю відображені основні результати дисертаційного дослідження, з цих робіт отримано вагомий науковий доробок аспіранта у вигляді опублікованих 6 статей у наукових фахових виданнях України; 2 статтей у наукових періодичних виданнях інших держав, що включені до наукометричних баз даних; 1 авторського твору та 5 тезах доповідей конференцій.

Структура й обсяг дисертації. Дисертаційна робота викладена на 152 сторінках та складається з анотації, змісту, переліку умовних скорочень, вступу, чотирьох основних розділів, в яких міститься 21 рисунок та 9 таблиць, висновку та списку використаних джерел з 152 найменувань. За структурою, стилем викладення та мовою дисертація відповідає вимогам МОН України. Робота написана грамотно українською мовою з використанням сучасної наукової термінології, а стиль викладення матеріалу є послідовним і логічним.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ, АЛГОРИТМІВ І ЗАСОБІВ НЕЙРОМЕРЕЖЕВОГО ЗАХИСТУ ДАНИХ У МОБІЛЬНИХ СМАРТ-СИСТЕМАХ

1.1. Архітектура нейронних мереж

Архітектура нейронних мереж – це структура або розташування взаємопов'язаних нейронів у нейронній мережі. Вона описує, як організовані нейрони і як вони взаємодіють один з одним, щоб виробляти вихідні дані на основі отриманих вхідних даних. Архітектура нейронної мережі визначає продуктивність мережі при вирішенні конкретних завдань, таких як обробка природної мови, розпізнавання мови, розпізнавання зображення та інших завдань машинного навчання [31, 145, 146].

Зокрема в статті [131] В. Литвин та інші науковці розглядають архітектуру нейронної мережі, яка може повністю відновити спотворене зображення. Для того, щоб зменшити час навчання, розмір нейронної мережі мінімізується шляхом спрощення її структури на основі одного з підходів: в основі першого лежить «регуляризація», в той час як другий базується на видаленні синаптичних зв'язків з нейронної мережі.

Архітектура нейронної мережі складається з шарів, кожен з яких виконує певний тип обчислень. Вхідний шар відповідає за отримання вхідних даних, а вихідний шар виробляє кінцевий результат роботи мережі. Шари між вхідним і вихідним шарами називаються прихованими. Кількість нейронів у кожному шарі та кількість прихованих шарів може змінюватися залежно від складності поставленої задачі [94, 108, 109].

Існують різні архітектури мереж, серед них рекурентні нейронні мережі [141], нейронні мережі прямого поширення, згорткові нейронні мережі та інші. Кожна з цих архітектур призначена для вирішення певних типів завдань, таких як послідовна обробка даних, розпізнавання зображень [113, 114], обробка природної мови.

Таким чином, архітектура нейронних мереж відіграє вирішальну роль у визначенні здатності мережі вирішувати конкретні завдання [116, 118]. Вибираючи відповідну архітектуру і налаштовуючи кількість шарів і нейронів, ми можемо досягти кращої продуктивності в різних завданнях машинного навчання.

1.2. Методи та алгоритми навчання нейронних мереж

Нейронні мережі навчаються за допомогою різних методів і алгоритмів, залежно від типу мережі та поставленого завдання. Загалом, процес навчання нейронної мережі передбачає подачу їй великого набору даних пар вхід-вихід, коригування ваг і зсувів мережі протягом декількох ітерацій (епох), щоб мінімізувати похибку між виходом мережі та бажаним виходом, а потім оцінку роботи навченої мережі на окремому валідаційному наборі [34, 36, 123].

Ось деякі з найпоширеніших методів і алгоритмів, що використовуються для навчання нейронних мереж:

1. Зворотне поширення (англ. Backpropagation). Це найпоширеніший алгоритм для навчання нейронних мереж прямого поширення. Розповсюдження працює шляхом обчислення похибки між виходом мережі та бажаним виходом, а потім розповсюдження цієї похибки назад через мережу для коригування ваг та зсувів [49].

2. Стохастичний градієнтний спуск (англ. Stochastic gradient descent, далі SGD). Це алгоритм оптимізації, який використовується для мінімізації похибки між виходом мережі та бажаним виходом. Він працює шляхом оновлення ваг і зсувів мережі в напрямку від'ємного градієнта функції помилки, використовуючи випадково вибрану підмножину навчальних даних (пакет) на кожній ітерації [93].

3. Міні-пакетний градієнтний спуск. Це різновид стохастичного градієнтного спуску, в якому навчальні дані розбиваються на невеликі партії, а ваги та зміщення мережі оновлюються на основі середнього градієнта, обчисленого для кожної партії. Міні-пакетний градієнтний спуск є більш обчислювально ефективним, ніж звичайний стохастичний градієнтний спуск, і часто використовується на практиці [96].

4. Адам. Це алгоритм оптимізації, який поєднує ідеї SGD та методів оптимізації на основі імпульсу. Він використовує адаптивну швидкість навчання для оновлення ваг та упереджень мережі, і було доведено, що він добре працює на різних завданнях.

5. Специфічні методи згорткових нейронних мереж (англ. Convolutional neural network, далі CNN). CNN – тип нейронних мереж, що використовуються для розпізнавання зображень і відео. Деякі специфічні методи, що використовуються для навчання штучних нейронних мереж, включають доповнення даних, регуляризацію відсіву та зменшення ваги.

6. Специфічні методи рекурентних нейронних мереж (англ. Recurrent neural network, далі RNN). RNN – це тип нейронних мереж, які зазвичай використовуються для обробки природної мови та прогнозування часових рядів. Деякі специфічні методи, що використовуються для навчання штучних нейронних мереж, включають зворотне поширення в часі, градієнтне відсікання та рекурентні одиниці із закритими воротами.

Існує багато різних методів і алгоритмів для навчання нейронних мереж і вибір методу залежить від конкретного завдання та архітектури мережі, що використовується. Важливо ретельно обирати метод навчання та гіперпараметри, для досягнення найкращих результатів у бажаному завданні.

У статтях [102, 103] автори пропонують новий алгоритм оптимізації під назвою «Адаптивні градієнтні методи з динамічним обмеженням швидкості навчання» (Adaptive Gradient Methods with Dynamic Bound of Learning Rate, AGD-DLR), який адаптивно налаштовує швидкість навчання під час навчання для досягнення швидшої збіжності та кращої точності, а також метод навчання нейронних мереж, які є агностичними до своїх ваг, що означає, що архітектура мережі є основним визначальним фактором її продуктивності, а не конкретні значення ваг.

У працях [104, 105] автори пропонують метод навчання глибоких нейронних мереж з використанням 8-бітних чисел з плаваючою комою, який зменшує вимоги до пам'яті та обчислювальних ресурсів для навчання без втрати

точності, а також метод для навчання великих нейронних мереж з розміром партії до 163 840 зразків, який значно скорочує час навчання без втрати точності.

1.3. Аналіз алгоритмів навчання та функціонування нейронних мереж та виділення базових компонентів

Аналіз галузей застосування нейротехнологій реального часу, архітектур, типових задач і нейромережевих алгоритмів показав, що вони мають такі особливості:

1. високу інтенсивність та постійність вхідних потоків даних;
2. підвищення вимог до точності результатів та постійне ускладнення алгоритмів;
3. можливість розпаралелення обробки у часі та у просторі;
4. здатність до абстрагування та узагальнення;
5. самонавчання, навчання та самоорганізації під впливом зовнішнього середовища.

Проведений аналіз показав, що забезпечити нейромережеву обробку інтенсивних потоків даних у реальному часі можна апаратними засобами [142-144]. Апаратні нейромережі реального часу ґрунтуються на операційному базисі, який наведений на рис. 1.1.

Нейромережний операційний базис складається із базових операцій попередньої обробки та процесорних операцій. При апаратній реалізації даних базових операцій використовуються елементарні арифметичні операції (рис. 1.1).

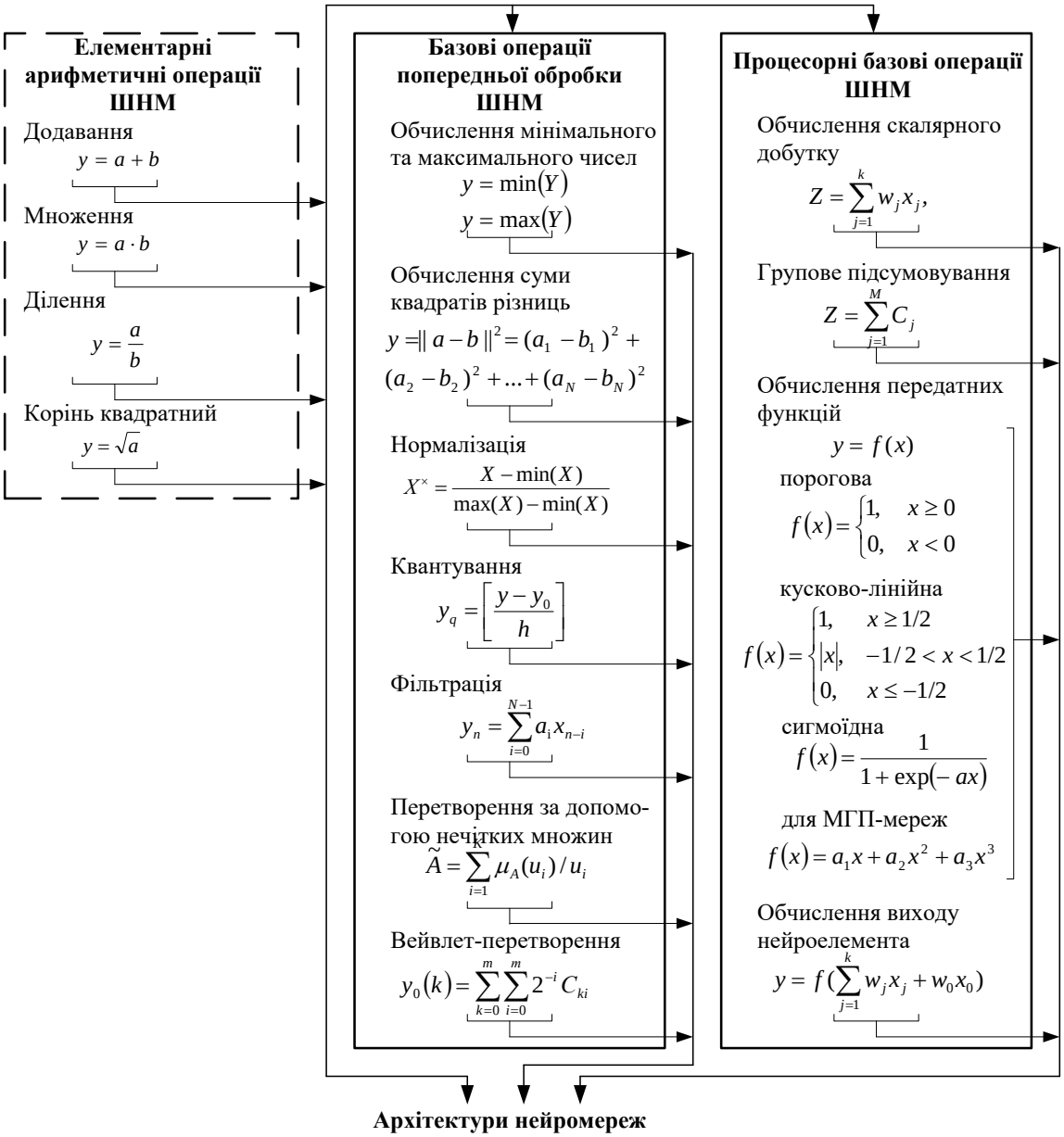


Рис. 1.1. Операційний базис апаратних нейромереж реального часу

На етапі попередньої обробки даних інформацію (початкові дані), що подаються на входи мережі, необхідно перетворити на таку, яка дасть найкращі результати. Навчальний вектор містить у собі по одному значенню для кожного входу мережі та, у залежності від типу навчання (із вчителем чи без), по одному значенню для виходу мережі. Зазвичай не вдається отримати якісні результи при використанні «сирого» набору для навчання мережі. Існує низка способів покращити «сприйняття» мережі:

1. Фільтрація робиться для «зашумлених» даних та полягає у відкиданні значеннь, які найімовірніше є некоректними.

2. Нормалізація застосовується тоді, коли подаються дані різної розмірності на різні входи мережі. До прикладу, на перший вхід нейромережі подаються значення з проміжку [0, 1], а вже на другий із [100, 1000]. Значно більший вплив на вихід нейромережі будуть мати значення з другого входу, за умови, якщо не було проведено нормування значень. Також, при нормалізації всі розмірності зводяться до одного діапазону, як для вхідних даних так і для вихідних.

3. Квантування застосовується для неперервних величин. Для них виділяється скінченний набір дискретних значень. Цей спосіб використовується при розпізнаванні мови, а саме для задання частот звукових сигналів.

Нормалізацію вхідних даних рекомендовано робити завжди. Нормалізацією називають процедуру попередньої обробки вхідних даних (робочих, навчальних і тестових вибірок), при якій значення ознак, що формують вхідний вектор, приводиться до певного заданого діапазону. Як результат, всі значення вхідних даних (ознак) будуть приведені до вузького діапазону (зазвичай [-1, 1] чи [0, 1]).

Існує багато способів нормалізації вхідних значень. Найпростішою та ефективною є лінійна нормалізація. Якщо початкові дані необхідно привести до діапазону [0, 1], то:

$$X^{\times} = \frac{X - \min(X)}{\max(X) - \min(X)}. \qquad (1.1)$$

У випадку приведення початкових даних до діапазону [-1, 1] лінійна нормалізація виглядає так:

$$X^{\times} = \frac{X}{\max(|X|)} \qquad (1.2)$$

Використання лінійної нормалізації можна вважати оптимальним, якщо вхідні дані X щільно заповнюють певний інтервал, оскільки при цьому не треба здійснювати складні обчислення.

У випадках, коли існують рідкісні відхилення, які значно більші за типові значення, лінійна нормалізація не є ефективною. Тут нормалізація призведе до того, що велика кількість значень початкових даних будуть близькими до 0.

Цього недоліку не має нормалізація за допомогою стандартного відхилення:

$$X^{\times} = \frac{X - \overline{X}}{\sigma},\ \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i,\ \sigma = \frac{1}{n\text{-}1}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2. \qquad (1.3)$$

При такій нормалізації досягається більш рівномірний розподіл вхідних даних. Але нормалізація за допомогою стандартного відхилення має суттєвий недолік – отримані нормалізовані значення не обов'язково будуть належати до одиничного інтервалу. Для вхідних даних це не важливо, але вихідні дані можуть використовуватися як еталонні значення для виходів нейронів. А це недопустимо у випадку, коли функція активації нейрона є сигмоїдною, оскільки вона приймає значення тільки в одиничному діапазоні.

Цього недоліку позбавлена нелінійна нормалізація, наприклад:

$$X^{\times} = f\left(\frac{X - \overline{X}}{\sigma}\right),\ f(a) = \frac{1}{1 + e^{-a}} \qquad (1.4)$$

Функцією (1.4) можна рівномірно нормалізувати більшість значень. Також, можна гарантувати, що отримане, під час нормалізації, значення буде в межах [0, 1]. Однак, варто зазначити, що недоліком є складність апаратної реалізації.

При апаратній реалізації найкращим рішенням буде використовувати саме лінійну нормалізацію показану в формулі (1.2). Але тільки у тих випадках, коли вхідні дані є рівномірними, або ж близькими до таких. При цьому, для такої апаратної реалізації треба розробити структури і методи для обчислення базових операцій ділення та знаходження максимального числа.

Однак, залежно від типу нейромережі, після проведення нормалізації, можуть застосовуватися й інші процедури для попередньої обробки даних. Наприклад, для мереж типу RBF або ж GRNN [133] необхідно здійснити розрахунок евклідової відстані від кожного вхідного вектора даних до усіх інших. Щоб знайти цю відстань використовують таку формулу:

$$y = \| x_i^e - x_i^b \|^2 = (x_1^e - x_1^b)^2 + (x_2^e - x_2^b)^2 + ... + (x_N^e - x_N^b)^2 . \quad (1.5)$$

Для інших типів нейронних мереж можуть використовуватися інші види попередньої обробки даних. Наприклад, для нейро-нечітких мереж над вхідними даними здійснюються перетворення за допомогою нечітких множин та правил нечіткої логіки, які мають добрі апроксимуючі здатності. У вейвлет нейронних мережах для аналізу різних частотних компонентів вхідних даних використовуються вейвлет-перетворення, наприклад, вейвлет Хаара, або вейвлети Добеші.

На наступному етапі операції над вхідними даними проводяться безпосередньо у самій мережі у процесі навчання та функціонування. Опираючись на результат аналізу існуючих алгоритмів [20, 21, 26] можна сказати, що основні операції обчислення в нейромережах зводяться до базових операцій обчислення скалярного добутку, операцій групового підсумовування та операцій обчислення передатних функцій.

Варта уваги операція обчислення сум парних добутків [70], оскільки ця операція застосовується найчастіше у нейроалгоритмах:

$$Z = \sum_{j=1}^{k} W_j X_j, \quad (1.6)$$

де X_j – j-тий вхід, W_j – j-й ваговий коефіцієнт, а k – кількість входів.

Якщо говорити про апаратну реалізацію, то є два підходи для реалізації цієї операції. Перший ґрунтується на базових операціях додавання і множення та переважно використовується при синтезі пристроїв на основі окремих мікросхем для обчислення суми спарених добутків. Другий, в свою чергу, заснований на операціях інверсії, додавання та зсуву. Цей підхід використовується при НВІС-реалізаціях. Що цікаво, використання другого підходу дозволяє оптимізувати пристрій, а саме за швидкодією та апаратними витратами [54].

Основним елементом пристрою обчислення скалярного добутку [20-24] є багатовходовий суматор. Тобто, обчислення скалярного добутку в базисі зводиться до макрооперації групування макрочасткових добутків:

$$Z = \sum_{j=1}^{M} C_j \,, \tag{1.7}$$

де M – кількість доданків; C_j – j-й доданок.

Передатна функція – алгоритмічний процес під час якого сигнал, одержаний після обчислення скалярного добутку, трансформується у вихідний сигнал. Для того щоб визначити значення виходу нейрона треба загальну суму порівняти із певним порогом. Залежно від результату порівняння генерується сигнал елементом обробки, якщо значення порогу є меншим за суму. В іншому випадку генерується гальмуючий сигнал або ж сигнал взагалі не генерується.

Нелінійна передатна функція застосовується частіше. Причина цьому це те, що лінійні або ж прямолінійні функції є обмеженими та вихід є прямо пропорційним до входу. На рис. 1.2 зображені типові передатні функції.

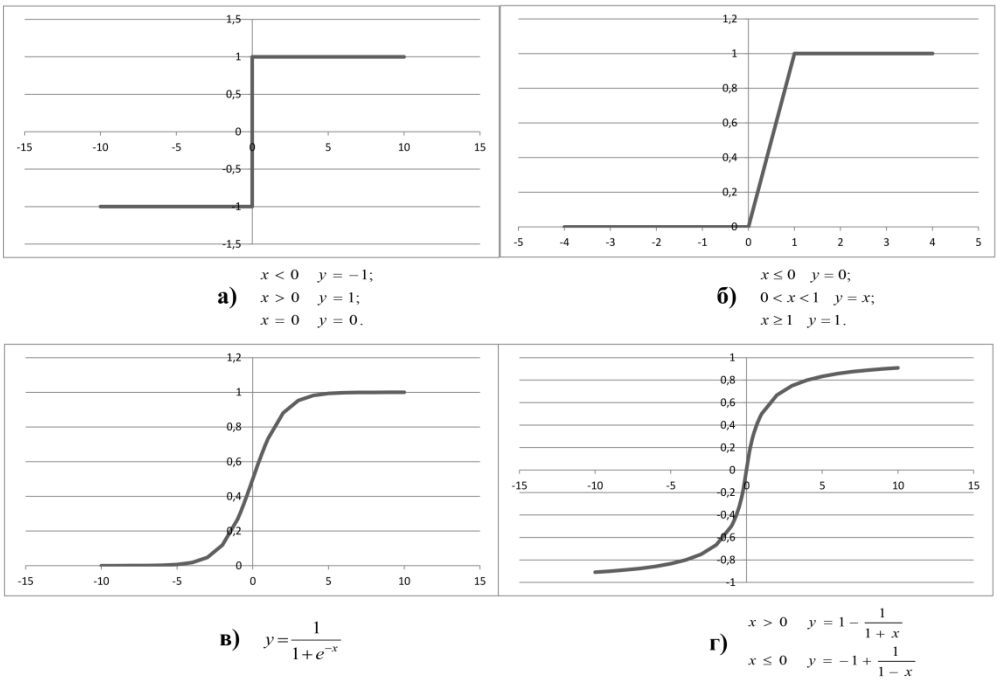


Рис. 1.2. Вигляд типових передатних функцій: а) жорстка порогова функція; б) лінійна з насиченням; в) сигмоїда; г) гіперболічний тангенс.

Передатна функція називається «жостким обмежувачем» (рис. 1.2а), якщо нейромережа видаватиме числові комбінації вигляду 0 та 1, 1 та -1 або ж інші. Також таку функцію ще називають – порогова функція.

Інший тип передатної функції – лінійна з насиченням (рис. 1.2б). Для заданого діапазону така функція віддзеркалює вхід. За межами даного діапазону вона є жорстким обмежувачем. Хоч ця функція і є лінійною, але вона обмежується до мінімальних і максимальних значень, і таким чином вона стає нелінійною.

Далі розглянемо S-подібні криві. Цей тип передатних функцій наближує мінімальне та максимальне значення у асимптотах. Коли діапазон такої функції є [0, 1], то тоді вона називається сигмоїдою (рис. 1.2в), якщо ж діапазон визначається [-1, 1] – гіперболічним тангенсом (рис. 1.2г). Особливістю таких кривих можна назвати неперервність функцій та відповідних похідних. Такі функції мають широке застосування та дають хороші результати.

До того як обчислити передатну функцію до вхідного сигналу можуть передавати також випадковий однорідно розподілений шум. Такий шум ще називають «температурою» штучних нейронів, для якого кількість та джерело визначає тип навчання.

1.4. Нейромережеві задачі та їхні особливості у робототехнічних системах

Область, що охоплює широкий клас об’єктів, покликаних замінити людину під час виконання небезпечних, важких або ж неприємних робіт, називається *робототехнічною системою*. До цього класу можна віднести як найпростіші іграшки, так і повністю автоматизовані механізми. Ядром такої системи є сучасні програмно-апаратні методи та засоби для обчислення.

Одне з найпоширеніших завдань нейронних мереж у робототехнічних системах – розпізнавання об'єктів. Це передбачає навчання нейронної мережі розпізнавати об'єкти на сцені, таких як люди, автомобілі або меблі. Цю задачу можна вирішити за допомогою нейронних мереж прямого поширення або згорткових нейронних мереж.

Іншим завданням нейронних мереж у робототехнічних системах є виявлення об'єктів. Це передбачає розпізнавання об'єктів на сцені та визначення

їхнього розташування і меж. Це завдання можна вирішити за допомогою алгоритмів виявлення об'єктів на основі згорткових нейронних мереж.

Одне з важливих завдань нейронних мереж у робототехнічних системах – планування руху. Це передбачає визначення оптимального шляху, яким повинен рухатися робот, щоб досягти заданого пункту призначення, уникаючи перешкод. Це завдання можна вирішити за допомогою нейронних мереж, таких як рекурентні нейронні мережі або алгоритми навчання з підкріпленням.

Ще одним завданням нейронних мереж у робототехнічних системах є управління. Це передбачає використання нейронної мережі для управління рухом і поведінкою робота, наприклад, регулювання його швидкості, положення або орієнтації. Таке завдання можна виконати за допомогою нейромережевих контролерів або алгоритмів навчання з підкріпленням.

Нейронні мережі також можна використовувати для злиття сенсорів, об'єднуючи інформацію з декількох датчиків (таких як камер або радарів) для створення більш точного і повного уявлення про навколишнє середовище. Це завдання можна виконати за допомогою нейромережевих архітектур, таких як рекурентні нейронні мережі або мультимодальні мережі злиття.

Однією з ключових особливостей нейронних мереж у робототехнічних системах є їхня здатність навчатися на власному досвіді. Це означає, що в міру того, як робот взаємодіє з навколишнім середовищем і отримує зворотний зв'язок про свої дії, нейронна мережа може коригувати свої параметри і покращувати свою продуктивність [6, 128, 129].

Ще однією особливістю нейронних мереж у робототехнічних системах є їхня здатність до узагальнення в нових ситуаціях. Це означає, що навіть якщо робот стикається з новим середовищем або об'єктом, нейромережа може використовувати свій попередній досвід для прогнозування та прийняття рішень.

У статті [79] представлено метод навігації роботів з використанням комбінації методів глибокого підкріплення та SLAM (одночасної локалізації та мапування). Автори використовують глибоку Q-мережу (DQN) для вивчення політики навігації робота і поєднують її з алгоритмом SLAM для оцінки

положення та орієнтації робота. Результати дослідження показують, що запропонований метод є ефективним для забезпечення навігації робота в складних умовах.

1.5. Аналіз елементної бази для реалізації нейронних мереж реального часу

Нейроподібні засоби криптографічного шифрування та дешифрування даних переважно реалізувалися програмно-апаратним шляхом з використанням нейрочіпів, процесорів цифрової обробки сигналів, систем на кристалі, мікроконтролерів і програмованих логічних інтегральних схем. Розглянемо детальніше перераховану елементу базу.

Нейрочіпи. Останні дослідження показують, що на даний час доцільно реалізовувати СЗПД у вигляді нейрочіпів, оскільки вони орієнтовані на виконання нейромережевих алгоритмів та забезпечують велику швидкодію при виконанні нейромережевих операцій. Перевагою вибору нейрочіпів при реалізації СЗПД є те, що на їх основі можна реалізувати високопаралельні системи. Нейрочіпи бувають загального призначення і спеціальні. Спеціальні нейрочіпи здатні реалізувати один нейромережевий алгоритм для певного застосування, в той час як нейрочіпи загального призначення можуть реалізувати більш ніж один нейромережевий алгоритм. Також вони можуть містити схеми налаштування ваг при навчанні або передбачати зовнішнє завантаження ваг. За типом інформаційного носія вони поділяються на цифрові, аналогові і гібридні.

Однією з переваг цифрових нейрочіпів є використання простих методів виготовлення, зберігання ваг в пам’яті та гнучкість розробки. Створення пристрою обчислення скалярного добутку входів нейрону на ваги можна назвати найбільш проблематичним для розробників, оскільки таке обчислення зазвичай є найповільнішим елементом обробки в мережі. Виділяють такі категорії цифрових нейрочіпів: розрядно-модульні, з одним потоком команд та багатьма потоками даних (SIMD) та систолічні матриці.

При використанні розрядно-модульної архітектури кожен модуль призначений для обробки декількох розрядів машинного слова, а слово в цілому

обробляться групою модулів або секцій, з’єднаних між собою. Прикладами таких нейрочіпів є Neuralogix NLX-420 Neural Processor, Micro Devices MD1220 Neural Bit Slice, а також Philips Lneuro chip.

SIMD системи орієнтовані на паралелізм даних, що дає підвищення продуктивності і підходять для реалізації СЗПД. У такій системі обробляється декілька потоків даних окремими елементами, але під загальним керуванням. Прикладом таких нейрочіпів є Adaptive Solutions N64000 з 64 процесорними елементами.

У випадку нейрочіпів на основі систолічної матриці кожен процесорний елемент робить рівно один крок обчислення синхронно з іншими процесорними елементами, а потім передає результат обчислення на наступний процесор в конвеєрі, в результаті чого дана архітектура підходить для реалізації основної операції нейрообчислень – множення з накопиченням. Зазвичай систолічні матриці – це спеціалізовані масиви, для яких характерним є дрібнозернистий паралелізм. Тому вони добре підходять для реалізації СЗПД з низькою пропускною здатністю. Як недолік треба зазначити велику складність системи взаємодії та контролю. Прикладом таких нейрочіпів є Siemens MA-16.

Процесори цифрової обробки сигналів (ЦОС). Процесори ЦОС є перспективною елементною базою для реалізації СЗПД. Перевагами процесорів ЦОС є висока продуктивність, апаратна реалізація операції множення з накопиченням, велика кількість технологічних засобів розробки та відлагодження програмного забезпечення. Існує велика кількість і різноманітність процесорів ЦОС, тому вибір того чи іншого процесора є актуальною і багатокритеріальною задачею, що напряму залежить від кінцевого застосування. Різні типи процесорів використовуються для різних галузей застосувань, тому при виборі слід враховувати вимоги, що ставляться до всієї системи загалом. Порівняно з іншою елементною базою реалізації ШНМ, основні переваги процесорів ЦОС полягають у використанні базових операцій нейрообчислень, а саме множення з накопиченнями (з англ. multiply-accumulate, MAC) і присутність апаратного

перемножувача, який може виконувати множення щонайменше двох чисел за 1 такт.

Процесори ЦОС можуть ефективно використовуватись для реалізації нескладної архітектури СЗПД, до яких не висуваються складні апаратні вимоги. Основною причиною вибору реалізації даної СЗПД на процесорі ЦОС стала його невисока вартість і задовільна швидкодія. Було показано, що реалізація на процесорах ЦОС є в 2-4 рази швидшою порівняно з програмною реалізацією, і для отримання такого результату не потрібно було використовувати складну елементну базу.

Також процесори ЦОС можуть використовуватись для розв'язання складних задач у реальному часі. Перші місця на ринку процесорів ЦОС займають такі фірми, як Texas Instruments і Analog Devices. Найкращим варіантом за критерієм якість/вартість є процесори фірми Analog Devices (аналогічні процесори Texas Instruments є дорожчими).

Процесори сімейства Blackfin (Analog Device) можуть ефективно використовуватись для нейромережевого криптографічного захисту. Крім того, деякі процесори ЦОС містять ефективні засоби узгодження роботи процесорів, які дозволяють реалізовувати структури, що можуть працювати паралельно. Серед таких можна відзначити процесори сімейства SHARC і TigetSHARC. Вони характеризуються високошвидкісними засобами обміну даних та мають високу продуктивність. Такі процесори використовуються для створення ефективних багатопроцесорних систем.

Процесори ЦОС, в залежності від копанії-виробника, можна поділити на два класи. Перший це дешеві мікропроцесори. Зазвичай використовуються при обробці даних з фіксованою комою. Другий – дорогі мікропроцесори, які можуть опрацьовувати дані у форматі з плаваючою комою. Також слід зазначити, що різні типи процесорів використовуються для різних галузей застосувань. Для портативних пристроїв таких як мобільні телефони чи портативні плеєри, вартість, ступінь інтеграції і споживана потужність є найважливішими, а висока продуктивність найчастіше не потрібна (оскільки вона вимагає значного

підвищення споживаної потужності, не даючи переваг при обробці щодо низькошвидкісних аудіоданих). Для ж радіолокаційних систем вартість неважливий критерій. Для таких систем основними є швидкість роботи, проста система розроблення програмного забезпечення та присутність швидкісних інтерфейсів. Більшість процесорів ЦОС з фіксованою комою мають малий об'єм внутрішньої пам’яті і невисоку розрядність зовнішніх шин даних. У той же час, процесори ЦОС з плаваючою комою створення щоб працювати як з великими масивами даних так і з складними алгоритмами. Вони мають великий об’єм вбудованої пам’яті. Також можуть мати велику розрядність адресних шин, щоб можна було підключити зовнішню пам’ять. Отже, вибір типу і об’єму пам’яті має бути результатом ретельного аналізу області застосування, в якій використовуються процесори ЦОС.

Програмовані логічні інтегральні схеми (ПЛІС). ПЛІС є ефективним ресурсом для апаратної реалізації СЗПД. Основними перевагами використання ПЛІС є: невисока вартість, що зумовлена масовим виробництвом, доступність, висока швидкодія, продуктивність і надійність, універсальність, різноманітність у виборі напруги живлення і параметрів сигналів вводу/виводу [115]. Ще до переваг варто віднести низьке енергоспоживання. Це є дуже важливий критерій, коли йдеться про створення портативної апаратури. Також різноманітність програмних засобів для автоматизованого проектування. І як фінальний критерій можна зазначити низьку затрату часу при проектування й відлагодженні проектів. Однак, попри всі переваги є проблема із розведенням необхідних з’єднань, яка виникає, коли треба реалізувати велику СЗПД із значною кількістю нейронів на ПЛІС.

Забезпечення часткової реконфігурації в процесі роботі є особливістю ПЛІС нового покоління. Системи на такій базі адаптуються для СЗПД з конкретною конфігурацією. Якщо порівнювати покоління ПЛІС, то останнє явно перемагає у рівні продуктивності, при збільшені якого можна побачити зниження рівня споживаної потужності. Як приклад порівняємо ПЛІС сімейства Artix-7 та Ізфкефт-6. Використання першого дозволяє в два рази зменшити

енергоспоживання, а продуктивність підвищити на 30%. Схожа ситуація спостерігається і для Kintex-7, який успішно може замінити Virtex-6.

ПЛІС поділяються на 4 класи, а саме: прості та складні ПЛІС (SPLD ат CPLD), ПЛІС із комбінованою архітектурою та програмовані користувачем вентильні матриці (FPGA). Останні використовуються найчастіше при реалізації СЗПД. Що цікаво ПЛІС із комбінованою архітектурою часто відносять до FPGA через те що в них чітко видно характеристики цього класу. Також програмовані користувачем вентильні матриці позволяють реалізувати складні СЗПД, бо мають більшу кількість логічних блоків відносно інших класів. Це чітко видно, якщо порівняти SPLD, які містять сотні логіних вентилів та FPGA, що мають кілька мільйонів [28, 29, 73].

Система на кристалі (з англ. system on a chip, SoC). Система SoC це сукупність необхідних електронних компонент та систем розміщених на невеликому інтегрованому чіпі. Також, ця система може містити цифрові, аналогові або радіочастотні функції. До компонент відносять центральний процесор (може бути багатоядерним), графічний процесор та системна пам’ять [90].

SoC на противагу багаточипових систем має багато переваг. Серед них варто згадати менше споживання енергії, також кращу продуктивність, і ,як плюс, ця система займає мало місця і є надійнішою. Такі чіпи зараз використовуються майже у всіх мобільних пристроях. Система на кристалі може створити всі потрібні схеми на одному пристрої. Також, варто уваги те, що SoC може включити електронні схеми великої кількості комп’ютерних коопонентів на одному інегрованому чіпі.

Аналіз ринку елементної бази показує, що для забезпечення конкурентоспроможними на ринку, сучасні SoC повинні володіти низкою критеріїв, серед яких хотілося б виділити декілька, а саме: необхідність у високій продуктивності базової платформи, щоб забезпечити можливість модернізації системи, необхідність зберігання великих обсягів даних, у тому числі в енергонезалежній пам’яті та на зовнішніх носіях тощо.

Сьогодні на ринку представлена велика різноманітність готових SoC, а також рішень, інтегрованих на ПЛІС, від провідних світових виробників: Intel, Cypress, Sharp, NetSilicon, Texas Instruments тощо. Проектування пристроїв на базі таких кристалів відкриває нові можливості.

Система на кристалі SoC, як правило, містить високопродуктивне процесорний ядро і великий набір периферії, яку ми звикли бачити в персональних комп'ютерах і сучасних мобільних пристроях. Велике число виробників і «необмежений» вибір SoC вносять деяку плутанину при виборі елементної бази для реалізації того чи іншого пристрою. Існує ряд ознак, за якими можна класифікувати системи SoC, а саме:

– процесорне ядро: ARM, MIPS, PowerPC, x86 і ін.;
– продуктивність ядра та частота системної шини;
– набір інтерфейсів;
– вартістю кристала і його мінімальної функції;
– виробник системи SoC.
– Системи SoC за використанням можна розділити на такі групи:
– бюджетних застосувань;
– пристроїв віддаленого управління;
– термінальних пристроїв;
– двоядерні (dual core), призначені для обробки даних;
– для спеціалізованих обчислювачів на основі ПЛІС.

Для синтезу СЗПД [124, 125] доцільно використовувати спеціалізовані обчислювачів на основі ПЛІС.

Мікроконтроллери. Мікроконтролер представляє собою невеликий та недорогий комп'ютер, який орієнтований розв’язання конкретних завдань, таких як кодування-декодування даних, нейроподібне шифрування-дешифрування даних, управління обчислювальним процесом у бортових і вбудованих системах. Мікроконтролери характеризуються розрядністю шин даних і адреси, наборами команд і структурами пам’яті. Класифікувати мікроконтролери можна здійснити за їх розрядністю:

1. 8-розрядний мікроконтролер: означає, що процесор або операційний пристрій можуть опрацьовувати 8-бітні дані в одному такті. Прикладами 8-розрядних мікроконтролерів є Intel 8031/8051.

2. 16-розрядний мікроконтролер: він забезпечує більшу точність і продуктивніть у порівнянні з 8-розрядним мікроконтролером. Прикладами 16-розрядних мікроконтролерів є Intel 8096 и Motorola MC68HC12.

3. 32-розрядний мікроконтролер: він використовує 32-бітні інструкції для виконання арифметичних та логічних операцій Прикладами 32-розрядних мікроконтролерів є - сімейство Intel / Atmel 251, PIC3x, ARM, STM32G4.

1.6. Загальні вимоги та основні напрями вдосконалення засобів реалізації нейронних мереж реального часу для робото технічних систем

Аналіз використання нейронних мереж у мобільних робототехнічних системах для реалізації криптографічного захисту даних [1, 2, 59, 60] показує, що вони мають такі суттєві недоліки:

1. не враховують вимоги конкретних застосувань щодо вартості, габаритів і споживаної потужності;

2. не орієнтовані на використання сучасних обчислювальних засобів і систем на кристалі;

3. їх реалізація часто не забезпечує режим реального часу.

З аналізу [3, 4] видно, що для забезпечення режиму реального часу та високих техніко-економічних характеристик необхідно використовувати нейроподібні мережі з попередньо обчисленими ваговими коефіцієнтами. У роботах [5, 6] авторами було розглянуто систему криптографічного захисту прямого поширення. Для навчання було використано алгоритм зворотнього поширення похибки. У [7, 8] продемонстровано застосування мереді Гопфілда, як приклад використання рекурентних нейромереж.

Також, у роботах [9, 10] розглянуто використання нейронних мереж зустрічного поширення (counter propagation) та радіальних базисних функцій для реалізації криптографічного захисту даних. Спільним недоліком розглянутих

нейронних мереж є використання алгоритмів ітераційного навчання, які не орієнтовані на реалізацію криптографічного захисту у системах реального часу.

У роботах [11, 12] здійснено адаптацію автоасоціативної нейронної мережі з неітераційним навчанням для задачі криптографічного шифрування та дешифрування даних. Вагові коефіцієнти розраховуються в результаті навчання на основі методу головних компонентів. Як особливість цього алгоритму можна зазначити використання системи власних векторів, що рівні власним значенням розрахованої коваріаційної матриці [13, 14]. Автоасоціативна нейромережа з обчисленими ваговими коефіцієнтами є нейроподібною мережею, орієнтованою на криптографічний захист. Аналіз нейроподібних засобів [15, 16], які використовуються для криптографічного шифрування (дешифрування), показав, що основою таких засобів є нейроподібні елементи. Особливістю таких нейроелементів [55] є те що, для них треба обчислити скалярний добуток з використанням попередньо обчислених вагових коефіцієнтів. У роботі [17] проаналізовано елементну базу та основні шляхи реалізації мобільних засобів криптографічного захисту передачі даних у реальному часі. Аналіз показує, що перспективним шляхом створення таких засобів є використання проблемно-орієнтованого підходу та сучасних мікрокомп’ютерих засобів [18, 19].

У роботі [107] автори дослідили нейромережеві методи доповнення даних. А саме дослідили варіаційний автокодер та підхід на основі GAN для генерації штучних числових даних і подальшого їх використання класифікаторами на основі машинного навчання.

Отже, розроблення на основі сучасної елементної бази з використанням нейроподібних мереж засобів криптографічного захисту у реальному часі з високими техніко-економічними характеристиками слід вважати актуальним завданням.

1.7. Висновки до розділу 1

1. Проведено аналіз архітектур нейронних мереж, а також проаналізовано методи та алгоритми навчання у результаті чого орієнтовано

задачі нейромережевого шифрування-дешифрування даних нейроподібної мережі прямого поширення автоасоціативного типу на основі парадигми «модель послідовних геометричних перетворень» шляхом неітеративного обчислення вагових коефіцієнтів, що забезпечило повторюваність результатів і орієнтацію на апаратну реалізацію.

2. Проаналізовано статті про алгоритми навчання нейронних мереж. Загалом, ці останні статті демонструють, що дослідження методів та алгоритмів навчання нейронних мереж є активним напрямком розвитку, в якому регулярно пропонуються нові методи та підходи.

3. Проаналізувавши статті, стає зрозуміло, що завдання нейронних мереж та їхні особливості є важливими напрямками досліджень у робототехніці. Методи глибокого навчання, включаючи ШНМ і навчання з підкріпленням, використовуються для того, щоб роботи могли виконувати широкий спектр завдань, включаючи розпізнавання об'єктів, виявлення об'єктів, планування руху, управління і злиття сенсорів. Однією з ключових проблем у цій галузі є розробка алгоритмів, здатних навчатися на основі обмежених даних, а також розробка методів інтеграції декількох сенсорних модальностей для створення більш повного уявлення про навколишнє середовище. Загалом, використання нейронних мереж у робототехніці має великі перспективи, оскільки дозволяє роботам виконувати складні завдання в реальних умовах.

4. Проведено аналіз елементної бази для реалізації нейронних мереж реального часу. У результаті вибрано напрям вдосконалення методу вибору елементної бази імітаційну модель, яка забезпечує автоматизацію процесу вибору найефективнішої елементної бази для синтезу засобів криптографічного захисту даних у реальному часі.

РОЗДІЛ 2
АДАПТАЦІЯ АВТОАСОЦІАТИВНОЇ НЕЙРОННОЇ МЕРЕЖІ ДО ЗАДАЧ КРИПТОГРАФІЧНОГО ЗАХИСТУ ДАНИХ І РОЗРОБЛЕННЯ ІМІТАЦІЙНОЇ МОДЕЛІ ОБЧИСЛЕННЯ ВАГОВИХ КОЕФІЦІЄНТІВ

2.1. Формування вимог і вибір принципів побудови засобів захисту даних для мобільних смарт-систем

Задача синтезу та створення системи захисту та передачі даних у реальному часі з високими техніко-економічними показниками потребує низки критеріїв для вирішення. Серед них можна виділити: розроблення нових структур, алгоритмів і методів для шифрування-дешифрування та кодування-декодування [132], та широке застосування сучасної елементної бази. При синтезі блоків СЗПД виникає проблема забезпечення режиму реального часу, підвищення криптостійкості, завадостійкості у той самий час, коли має бути зменшена вага, вартість, споживання енергії та зменшення габаритів. Реальний час обумовлений відсутністю накопичень затримок при шифруванні (дешифруванні) даних, а час шифрування $t_ш$ та час дешифрування $t_д$ повинні бути рівними. Шифрування у реальному часі обмежує $t_ш$, оскільки цей час не має бути більшим за час надходження даних $t_{нд}$, тобто:

$$t_ш \le t_{нд} \tag{2.1}$$

Час надходження даних $t_{нд}$ опирається декілька критеріїв. Серед них: обсяг (N), розрядність (n), частоти (F_d) надходження вхідних даних (X_{jj}), кількість каналів (k) та їх розрядності (n_k). Визначається так:

$$t_{нд} = \frac{Nn}{F_d k n_k} \tag{2.2}$$

Для шифрування (дешифрування) потоків даних в реальному часі за допомогою апаратно-програмних засобів їх продуктивність повинна бути:

$$П \ge \frac{\beta R F_d k n_k}{Nn}, \tag{2.3}$$

де R – складність; β – коефіцієнт особливостей засобів реалізації.

Застосування блоків шифрування (дешифрування) та кодування (декодування) даних у галузях, де апаратура є бортовою, мається на увазі такою, що возиться, літає і плаває, тобто є мобільною, накладає обмеження на масогабаритність. Одночасно до блоків шифрування (дешифрування) та кодування (декодування) висуваються жорсткі вимоги щодо споживаної потужності, яка впливає на розміри джерел живлення та засобів для відведення тепла. Потреба у задоволенні вимог забезпечення масогабаритності, споживання енергії та вартості змушують при синтезі описаних блоків дуже строго підходити до вибору елементної бази. Це проявляється у бажанні зменшити довжину розрядності сітки, де операнди предсталені з фіксованою комою, зменшити список використаних команд і число ліній адресної шини, що визначають ємність пам’яті, доступну для користувача.

Крім того, до блоків шифрування (дешифрування) та кодування (декодування) даних ставляться високі вимоги щодо живучості, надійності, а також забезпечення перевірки працездатності, швидкої локалізації і знешкодження несправностей. Проблема високої живучості блоків шифрування (дешифрування) даних виникає при використанні їх в системах управління особливо відповідальними об’єктами, розміщеними на великій відстані від людини. Щоб забезпечити достатньо високу живучіть блоків шифрування (дешифрування) та кодування (декодування) даних, необхідна взаємозаміна його структурних частин. Вирішити дану задачу можна тільки при однотипності складових частин блоків шифрування (дешифрування) та кодування (декодування) і однорідності їхніх архітектури.

Зменшення енергоспоживання та масогабаритних характеристик, підвищення надійності блоків СЗПД та забезпечення режиму реального часу можу бути досягнуто шляхом використання сучасних надвеликих інтегральних схем (НВІС). При НВІС-реалізації блоки шифрування (дешифрування) та кодування (декодування) даних повинні забезпечити високу ефективність використання обладнання [120, 121, 122], яка враховує однорідність структури,

зв’язує продуктивність з витратами на обладнання, а також дає оцінку елементам за продуктивністю.

Щоб вирішити задачу синтезу блоків шифрування (дешифрування) та кодування (декодування) даних у реальному часі із високою ефективністю при використанні обладнання необхідно звести до мінімуму апаратні затрати. Також треба мінімізувати кількість виводів на інтерфейсах та збільшити однорідність структури.

Висока ефективність використання обладнання при апаратній реалізації блоків шифрування (дешифрування) даних досягається узгодженням інтенсивності P_d надходження вхідних даних X_j із інтенсивністю D_o обчислень даних блоків. Інтенсивність P_d надходження вхідних даних X_j визначається так:

$$P_d = k n_k F_d, \tag{2.4}$$

де k – кількість каналів надходження вхідних даних X_j; n_k – розрядність каналів надходження даних; F_d – частота надходження даних, а інтенсивністю D_o обчислень блоків СЗПД даних визначається так:

$$D_o = N_{HE} m F_к, \tag{2.5}$$

де N_{HE} – кількість нейроподібних елементів; m – кількість входів нейроподібних елеменів; F_K – тактова частота роботи блоків шифрування (дешифрування) даних.

Незважаючи на досягнуті успіхи в області автоматизації проектування апаратно-програмних засобів, одним з найтрудоємніших етапів у створенні блоків шифрування (дешифрування) та кодування (декодування) даних є моделювання та відлагодження. Для проведення відлагодження синтезовані блоки даних повинні забезпечувати керованість, спостережуваність і передбачуваність. Керованість – властивість блоків, при якій його поведінка піддається керуванню, тобто є можливість запустити, зупинити, продовжити роботу з будь-якої адреси. Спострежуваність – властивість блоків, що дозволяє слідкувати за її поведінкою і зміною внутрішніх станів. Передбачуваність – властивість блоків, що дозволяє встановити їх у стан, з якого всі наступні стани можуть бути передбачені [127].

Синтез СЗПД у реальному часі з високими техніко-економічними характеристиками пропонується здійснювати з використанням інтегрованого підходу, до якого входять:

– дослідження та розроблення теоретичних основ нейроподібного шифрування-дешифрування даних і синтезу шумоподібних кодів;

– розроблення нових алгоритмів та структур нейроподібного шифрування-дешифрування даних, орієнтованих на сучасну елементну базу;

– розроблення нових алгоритмів та адаптивних структур кодування-декодування даних з використанням шумоподібних кодів;

– вибір сучасної елементної бази з можливістю програмування структури;

– використання засобів автоматизованого проектування програмно-апаратних засобів.

При синтезі СЗПД необхідно забезпечити вимоги технічного завдання та високу ефективіність використання обладнання, що зв’язує продуктивність з витратами обладнання та дає оцінку елементам системи за продуктивністю:

$$E_{СЗПД} = \frac{\beta_з R_{ш-к/д-д}}{W_{СЗПД} t_{ш-к/д-д}} \tag{2.6}$$

де $\beta_З$ – коефіцієнт врахування засобів реалізації алгоритмів, $W_{СЗПД}$ – апаратні затрати на реалізацію СЗПД, $R_{ш\text{-}к/д\text{-}д}$ – складність алгоритмів шифрування (дешифрування)-кодування (декодування), $t_{ш\text{-}к/д\text{-}д}$ – час шифрування (дешифрування)-кодування (декодування).

Задача синтезу СЗПД з високою ефективністю використання обладнання зводиться до забезпечення реального часу при мінімальних апаратних затратах:

$$\begin{aligned} W_{СЗПЛ} &= W_{ЗШКП} + W_{ЗДДП} = \\ &= 2W_{ПЯ} + 4W_{ВМ} + W_{БШ} + W_{БК} + 2W_{БПК} + 2W_{БПКод} + W_{БДШ} + W_{БДК} + W_{Пер} + W_{Пр} \Rightarrow min \end{aligned} \tag{2.7}$$

де $W_{ЗШКП}$ і $W_{ЗДДП}$ – апаратні затрати на реалізацію відповідно засобів шифрування, кодування та передавання даних та засобів дешифрування, декодування та приймання даних, $W_{ПЯ}$, $W_{ВМ}$, $W_{БШ}$, $W_{БК}$, $W_{БПК}$, $W_{БПКод}$, $W_{БДШ}$, $W_{БДК}$, $W_{БКод}$, $W_{Пер}$, $W_{Пр}$ – апаратні затрати на реалізацію відповідно процесорного ядра,

вузла маскування, блоків шифрування, блоків кодування, блока пам’яті ключів, блока пам’яті кодів, блока дешифрування, блока декодування, передавача та приймача.

Для синтезу ефективних СЗПД у реальному часі з використанням шумоподібних кодів вибрано такі принципи:

– змінності складу обладнання, що передбачає наявність процесорного ядра та змінних модулів (нейроподібних елементів), це дає змогу ядру адаптуватися;

– модульності, для розробки компонентів СЗПД у вигляді функціонально завершених пристроїв;

– конвеєризації та просторового паралелізму при нейроподібному шифруванні-дешифруванні даних;

– відкритості програмного забезпечення, що дає можливість збільшувати та вдосконалювати використання драйверів та програмних засобів;

– адаптації та спеціалізації апаратно-програмних засобів до структури алгоритмів нейроподібного шифрування (дешифрування);

– програмованості архітектури блоків шифрування-дешифрування та кодування-декодування шляхом використання програмованих логічних інтегральних мікросхем.

2.2. Адаптація автоасоціативної нейронної мережі прямого поширення до задач криптографічного захисту даних

В основі парадигми МПГП лежить неінтераційний підхід до навчання нейроподібної мережі, який передбачає пряме обчислення вагових коефіцієнтів під час плавного зменшення розмірності простору вхідних багатовимірних даних на нейронах прихованого шару [137, 138, 139]. При цьому виконується представлення вхідних багатовимірних даних у новому ортогональному базисі, що робиться алгоритмом найвіддаленішої точки.

Щоб спростити апаратну реалізацію нейроподібних мереж на основі НВІС-структур треба подати вхідні, вихідні дані, а також вагові коефіцієнти МПГП

мережі у форматі з фіксованою комою. Для цього передбачається попереднє масштабування вхідних даних [135].

При виборі структури нейроподібної мережі для шифрування-дешифрування потоків даних у реальному часі запропоновано до використання автоасоціатину мережу з одним прихованим шаром [1, 2] (рис. 1.2).

Така структура мережі є досить універсальною і може бути використана при розв'язанні різних задач, які передбачають перетворення вхідних даних з подальшим їх відновленням: кодування вхідних даних з метою їх стиснення, блочне симетричне шифрування, накладання цифрових водяних знаків та стеганографії [3]. Модулі шифрування та дешифрування даних, відповідно, формують та використовують ключ, який утворюють параметри навченої нейромережі. Для підвищення криптостійкості даних на виході прихованого шару структуру мережі можна доповнити блоком шифрування за методом одноразового блокноту (накладанням маски операцією XOR).

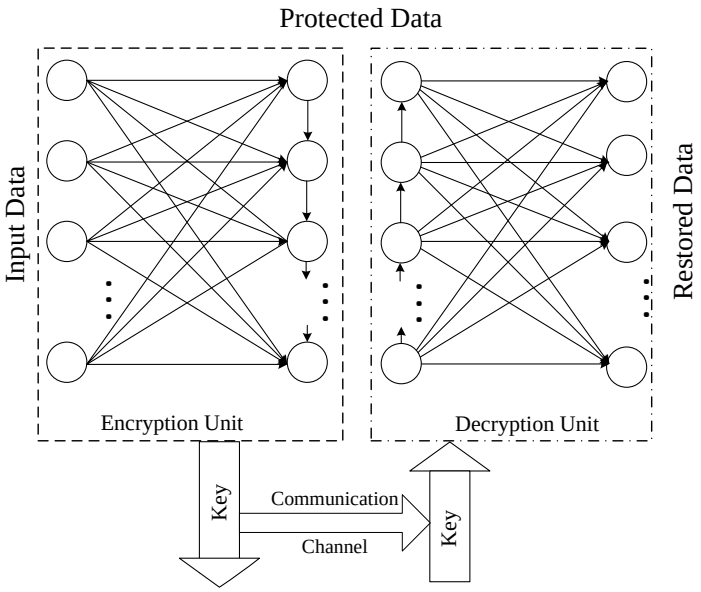


Рис. 2.1. Граф нейроподібної мережі для шифрування-дешифрування даних на основі моделі послідовних геометричних перетворень

При кількості нейронів у прихованому шарі рівній кількості вхідних і вихідних нейронів забезпечується можливість відновлення без втрат на виході

мережі захищених даних, які будуть отримані на виходах нейронів прихованого шару. Зворотні послідовні геометричні перетворення між нейронами прихованого та вихідного шару використовуються для відновлення даних на виходах мережі.

Передбачена можливість послідовного з'єднання декількох блоків шифрування та наступних відповідних блоків дешифрування з утворенням каскадної багатошарової мережі.

Функціонування модуля шифрування даних можна описати наступним чином:

$$h = F(X, key), \tag{2.8}$$

де X, h – сигнали на вході та на виході модуля, відповідно; $key = \{N, W, mask\}$ - ключ, яких складається із заданою кількістю нейронів у вхідному (вихідному, прихованому) шарі мережі N, матриці вагових коефіцієнтів W , та маски для одноразового блокноту $mask$, F - функція, яка задає прямі послідовні геометричні перетворення.

Тоді функціонування модуля дешифрування описується як:

$$\tilde{X} = \bar{F}(h, key), \tag{2.9}$$

де h - сигнал на вході модуля (захищені дані), $\tilde{X}$ – сигнал на виході модуля (відновлені дані), $\bar{F}$ - функція, яка описує зворотні послідовні геометричні перетворення.

Час життя ключа *key* тоді залежить від:

– кількості нейронів N ;
– розрядності маски *n*;
– кількості послідовно з'єднаних блоків шифрування та дешифрування у каскадній мережі *k*.

Із розвитком технологій зберігання і передачі даних дедалі більшою проблемою стає захист інформації, у тому числі, за допомогою методів шифрування. Сучасні технології криптоаналізу, які використовують доступні ресурси для високопродуктивних обчислень, змусили засумніватися у стійкості

багатьох класичних алгоритмів шифрування даних і навіть сприяли розвитку нових алгоритмів.

Технології штучних нейронних мереж використовують для розробки ефективних, а головне стійких методів для криптографічного захисту даних. Такі мережі мають різноманітну архітектуру, а також алгоритми навчання. Штучні нейронні мережі можна гнучко налаштовувати, у тому числі для задач симетричного шифрування даних. Також архітектури та методи навчання нейронних мереж характеризуються здатністю до розпаралелювання обчислень, що відкриває можливості ефективної апаратної реалізації.

Відомі підходи передбачають використання для побудови систем криптографічного захисту прямого поширення, які навчаються за алгоритмом зворотного поширення похибки. Також розглядаються можливості застосування рекурентних нейронних мереж, зокрема, мережі Гопфілда, нейронних мереж синхронізації і хаотичних нейронних мереж. Використання алгоритмів ітераційного навчання є спільною особливістю даних підходів, і це можливості їх використання у системах захисту даних в реальному часі. Крім того, типовою практикою є ініціалізація мережі випадковими значеннями параметрів, зокрема, вагових коефіцієнтів, що веде до неповторюваності результатів навчання таких мереж.

Метою досліджень, які описані у цьому розділі, є розроблення методу симетричного шифрування даних на основі нейронних мереж з неітераційним навчанням.

Метод нейромережевого криптографічного захисту даних

Запропонований метод симетричного шифрування опирається на використанні архітектури автоасоціативної нейронної мережі прямого поширення (АНМ) із латеральними зв’язками між нейронами прихованого шару, що навчається на основі парадигми «модель послідовних геометричних перетворень» (МПГП) (рис. 2.1). Нейроподібні структури МПГП [134] забезпечують швидкість, повторюваність, можливість оперування з даними великих обсягів.

У якості базової структури розглянемо автоасоціативну мережу лінійного типу, яка повинна забезпечити необхідну для нашого завдання симетрію процедури шифрування-дешифрування без втрат інформації і в реальному часі.

Визначальною особливістю АНМ на основі багатошарових перцептронів є те, що вхідні вектори з навчальної множини є тотожними, але приховані шари містять менше нейронів ніж вхідні та вихідні. Тоді виходи прихованого шару відображають представлення даних у просторі меншої розмірності, а відтворення вихідних компонентів відбувається з певними похибками методу, що неприпустимо з огляду симетрії операцій шифрування-дешифрування.

Розглянемо наш варіант АНМ на основі моделі послідовних геометричних перетворень.

Базова концепція моделі в автоасоціативному режимі передбачає пряме і зворотне перетворення даних без втрат інформації. Дані, що використовуються для навчання АНМ утворюють матрицю, кожен $i-$ й рядок (для $i=\overline{1,N}$) якої відповідає вектору перетворень j-ту компоненту вектора-рядка позначимо $x_{i,j}$. У даній постановці задачі значення усіх компонент векторів є відомі. Подамо матрицю вхідних даних як таблицю результатів обчислень значень деякої функції двох змінних $F(i,j)$. Базова ідея моделі МПГП – представлення функції двох змінних скінченою сумою добутків функцій однієї змінної, яке здійснюється в такій послідовності:

1. Задаємо перший крок перетворень (*m=1)*.
2. Обираємо базовий рядок матриці, що складається з елементів $x_{b,j}^m$ для $j=\overline{1,n}$, відносно якого дисперсія векторів початкового набору є максимальною.
3. Від кожного вектора-рядка матриці віднімається добуток базового рядка на коефіцієнт K_i^m, що забезпечує мінімум різниці в сенсі критерію найменших квадратів

   а. $x_{i,j}^{m+1}=x_{i,j}^m-K_i^m x_{b,j}^m$, (2.10)

b. $$K_i^m = \frac{\sum_{j=1}^{n} x_{i,j}^m x_{b,j}^m}{\sqrt{\sum_{j=1}^{n} (x_{b,j}^m)^2}}, \quad (2.11)$$

для $i = \overline{1, N}$.

4. Інкрементуємо m=m+1 і переходимо на К. 2.

Послідовні перетворення (2.10, 2.11) над матрицею даних забезпечують її обнуління, а отримані вектори K_i^m відповідають значенням сигналів головних компонент. У режимі зворотного перетворення довільний елемент матриці визначається через її головні компоненти:

$$x_{i,j}^1 = \sum_{m=1}^{n} K_i^m x_{b,j}^m . \quad (2.12)$$

Таким чином у прихованому шарі мережі виконується побудова гіперплощини заданої розмірності у просторі вхідних даних. Ця гіперплощина наближає сукупність вхідних даних з мінімальною залишковою дисперсією (за аналогією із статистичним методом головних компонент), використовуючи ортогоналізацію Грамма-Шмідта при переході до нового базису.

Модель послідовних геометричних перетворень лежить в основі неітеративного методу навчання автоасоціативної нейронної мережі для операцій шифрування (дешифрування), для якої величини змінних, отримані в результаті навчання мережі, перераховуються у відповідні вагові коефіцієнти міжнейронних зв’язків (рис. 2.2).

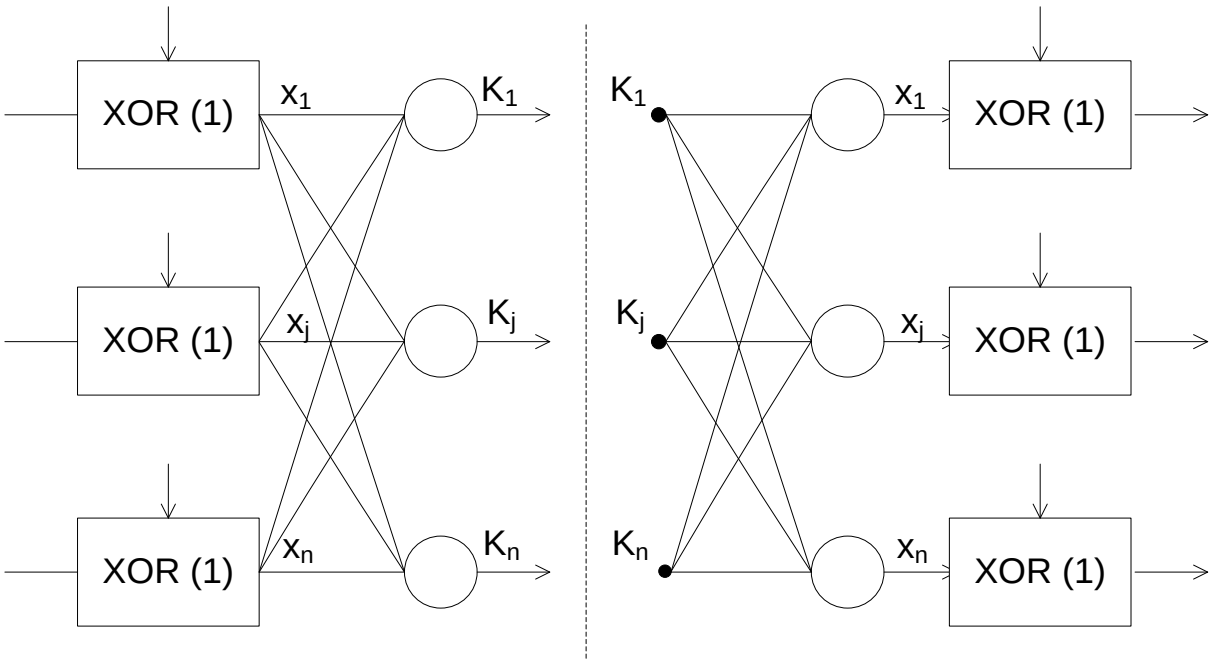


Рис. 2.2. Автоасоціативна нейронна мережа, доповнена маскуванням вхідних даних

Для даної структури блок цілих чисел вхідної інформації $X_{i,1},...,X_{i,j},...,X_{i,n}$, після маскування операцією *XOR1,* подається на вхід попередньо навченої АНМ МПГП, де на виходах нейронних елементів прихованого шару отримуємо сигнали головних компонентів K_i^m у форматі дійсних чисел з пливучою комою. Отримані сигнали є шифром, що передається в канал зв'язку. Дані сигнали на приймальній стороні системи проходять через вихідний каскад мережі, перетворюються у формат цілих чисел з фіксованою комою і через операцію *XOR1* перетворюються в початково заданий код. Елементами таємного ключа в даній системі є код маски *XOR1,* а також матриця даних, на яких навчалася нейромережа. Слабкою стороною подібної системи шифрування є можливість зламу шифру крім методу брутальної атаки на складову *XOR,* застосуванням методу розв'язку систем лінійних алгебраїчних рівнянь в цілих числах.

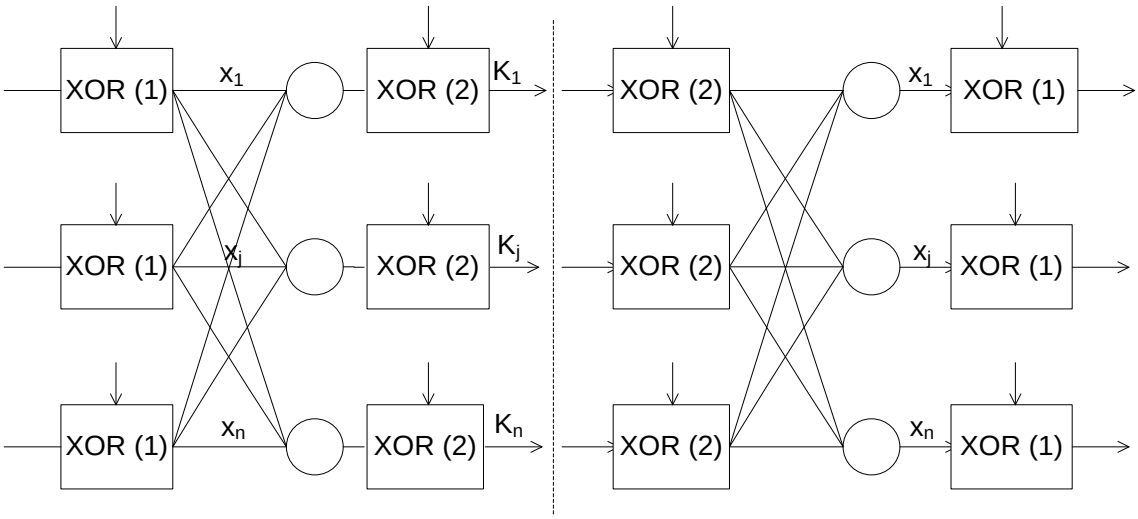


Рис. 2.3. Автоасоціативна нейронна мережа, доповнена маскуванням вхідних даних та даних прихованого шару

Для уникнення останньої криптоатаки пропонується ввести додаткове маскування сигналів K_i^m за допомогою, коду *XOR2*, що унеможливлює розв'язування методом цілих чисел (рис. 2.3). Зокрема, можна маскувати лише порядки дійсних чисел, або також і відповідні мантиси. В результаті отримуємо ключі достатньо великої довжини для забезпечення високої криптостійкості [51].

Схожий геометричний підхід до розробки ефективних методів та засобів шифрування і дешифрування зображень розглядається в [10-13].

2.3. Розроблення структури засобів захисту та передачі даних у реальному часі

Структура СЗПД у реальному часі з використанням шумоподібних кодів повинна бути орієнтована на ефективну реалізацію алгоритмів шифрування (дешифрування) та кодування (декодування) на сучасній елементній базі. Одним із шляхів досягнення високих техніко-економічних характеристик СЗПД є використання для криптографічного захисту нейроподібної мережі прямого поширення автоасоціативного типу, яка навчається неітеративним методом послідовних геометричних перетворень. Особливістю таких нейроподібних мереж є принципова можливість неітеративного обчислення вагових коефіцієнтів

синаптичних зв’язків між нейронними елементами. Використання попередньо обчислених вагових коефіцієнтів і вдосконаленого таблично-алгоритмічного методу забезпечить ефективну реалізації алгоритмів нейромережевого шифрування-дешифрування даних на базі універсального процесорного ядра доповненого спеціалізованими апаратно-програмними засобами. Підвищення завадостійкості передачі даних досягається за рахунок використання баркероподібних кодів і засобів адаптації їх розрядності до величини завад.

З використанням інтегрованого підходу та вибраних принципів розроблена структура СЗПД, яка наведена на рис. 2.4, де ССШ – співвідношення сигнал/шум, А – повідомлення, В – шифротекст, К – ключ шифрування (дешифрування), R – закодований шифротекст.

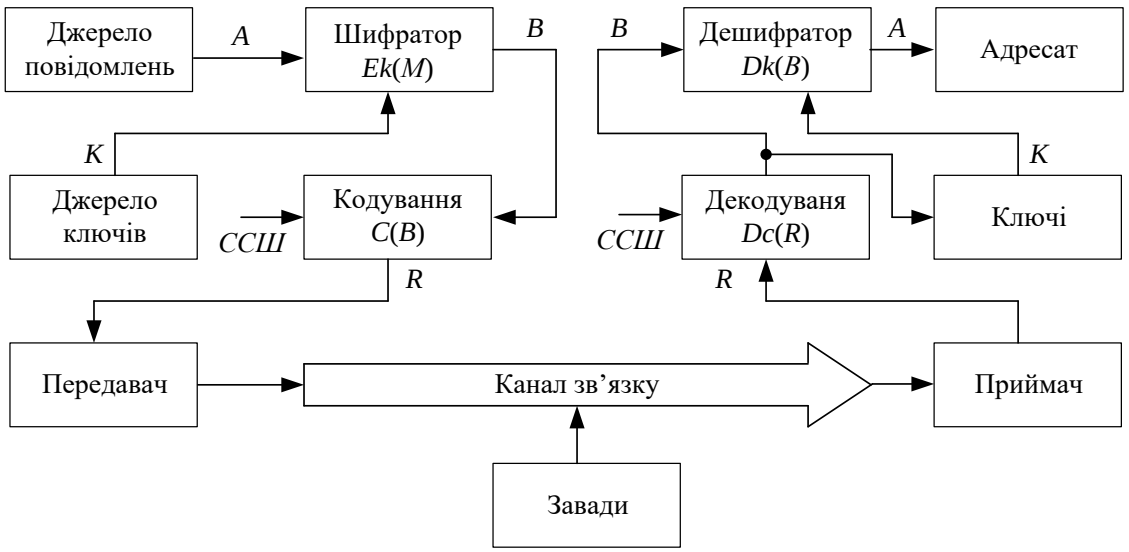


Рис. 2.4. Структура СЗПД з використанням шумоподібних кодів

У СЗПД з використанням шумоподібних кодів є такі об’єкти:

1. Алфавіт M, в якому записуються повідомлення (відкриті тексти). Повідомлення A є словом в цьому алфавіті і може складатися з багатьох слів у звичайному лінгвістичному розумінні, тобто $A \epsilon M^*$, де M^* – простір повідомленнь.

2. Алфавіт S, в якому записуються шифротексти. Відповідно $B \epsilon S^*$, де B – шифротекст, S^* – просторів шифротекстів.

3. Прострів ключів K^*, що складається з архітектури нейроподібних мереж матриць вагових коефіцієнтів W_{ji} (j=1, ..., N, де N – кількість нейроподібних елементів у нейроподібних мережах, i=1, ..., m, де m – кількість вагових коефіцієнтів для j-го нейроподібного елемента) та розрядності масок n.

4. Шумоподібні коди $\{a_i\}_{i=1}^{Q}$ на основі квазібаркерних послідовностей для кодування та декодування даних [76].

СЗПД використовує шифрування з симетричними ключами, де ключі шифрування та дешифруванні рівні, або ж їх легко обчислити маючи один з них.. При передачі зашифрованих даних здійснюється їх кодування з використанням шумоподіних кодів. У системі захисту та передачі даних виконуються такі операції:

1. Шифрування $E_k : M^* \to S^*, K \in K^*$ відбуваються над відкритим текстом з використанням ключа, який складається із заданої кількості нейронів у вхідному шарі мережі N, матриці вагових коефіцієнтів Wji та базових операцій: обчислення скалярного добуту та додавання.

2. Кодування шифротексту B з використанням шумоподібних кодів $C_{ССШ} : B \to R,$ довжина яких визначається співвідношенням сигнал/шум (ССШ):

$$ССШ = 20 \log_{10} \frac{A_{сигнал}}{A_{шум}}$$

де $A_{сигнал}$ – амплітуда сигналу, $A_{шум}$ – амплітуда шуму.

3. Декодування шифротексту R з використанням шумоподібних кодів:

$$\mathrm{Dc}_{\mathrm{ССШ}} : R \to B \ .$$

4. Дешифрування $D\mathrm{k}_K : B^* \to A^*, K \in K^*$ перетворює шифротекст у вихідний відкритий текст.

СЗПД з використанням шумоподібних кодів працює так. Повідомлення A поступає на вхід блоку шифрування *Ek(M)*, де воно шифрується з використанням ключа K, який визначається архітектурою нейроподібної мережі (кількість нейроподібних елементів N), матрицею вагових коефіцієнтів W_{ji} та розрядністю маски n. На виході блоку шифрування *Ek(M)* отримаємо шифротекст B, який

поступає на вхід блоку кодування *C(B)*. На виході блоку кодування *C(B)* отримуємо закодований шифротекст *R*, де кодний двійковий розряд кодується шумоподібним кодом. Довжина такого коду визначається ССШ. Чим більше значення ССШ, тим менша розрядність шумоподібного коду, яка використовується для кодування шифротексту B, а при зменшенні значення ССШ збільшується розрядність шумоподібного коду. Закодований шифротекст R надходить на передавач і каналом зв'язку передається на приймач, з виходу якого надходить на блок декодування *Dc(R)*. У блоці декодування *Dc(R)* закодований шифротекст декодується у шифротекст B. При декодуванні шифротексту R використовується шумоподібний код, розрядність якого відповідає розрядності шумоподіного коду, що використовувався при кодуванні шифротексту *B*. Шифротекст *B* з виходу блоку декодування *Dc(R)* надходить на блок дешифрування *Dk(B)*, де він дешифрується з використанням операцій *XOR*, ключа *K* (архітектура нейроподібної мережі, матриці вагових коефіцієнтів). На виході блоку дешифрування *Dk(B)* отримуємо вихідне відкрите повідомлення *A*, яке передається адресату.

Одним із шляхів підвищення криптостійкості СЗПД є використання каскадної нейроподібної мережі для шифрування (дешифрування) даних, яка утворюється послідовним з'єднанням декількох блоків шифрування *Ek(M)* (дешифрування *Dk(B)*). Час життя ключа *K* у системі захисту та передачі даних залежить від:

– кількості *P* послідовно з'єднаних блоків шифрування чи дешифрування;
– кількості нейронів N_g (g=1, ..., P) у кожному блоці шифрування чи дешифрування даних;
– кількості матриць вагових коефіцієнтів *W*;
– розрядності масок *n* для операцій *XOR*.

СЗПД з використанням шумоподібних кодів характеризується простотою налаштування кодів маскування, нейроподібних мереж, матриць вагових коефіцієнтів і генерації шумоподібних кодів різної розрядності. Висока швидкодія

для забезпечення режиму реального часу досягається шляхом використання таблично-алгоритмічного методу реалізації нейроподібних мереж шифрування-дешифрування даних.

2.4. Вдосконалення методу сингулярного розкладу матриці та орієнтація його на обчислення вагових коефіцієнтів нейроподібних мереж

Класичний процес навчання нейромережі зводиться до визначення матриці вагових коефіцієнтів [40, 41]. Для реалізації цього завдання було обрано метод сингулярного розкладу матриці (англ. Singular Value Decomposition, далі SVD [39, 42]). Сингулярний розклад матриці – певного типу розкладання прямокутної матриці і має застосування в силу своєї наочної геометричної інтерпретації при вирішені багатьох завдань. Переформулювання сингулярного розкладання – так зване розкладання Шмідта має застосування у квантовій теорії інформації. SVD схожий на метод головних компонентів (МГК), але є більш загальним. МГК передбачає, що вхідна матриця має бути квадратною, коли ж SVD не має такого обмеження.

Загальна формула SVD така:

$$A = UDV^T, \tag{2.13}$$

де A – матриця вхідних даних $N\ x\ n$; U – ліва сингулярна матриця $N\ x\ N$, стовпці містять власні вектори матриці AA^T; D – діагональна матриця $N\ x\ n$, що містить сингулярні (власні) значення; V – права сингулярна матриця $n\ x\ n$, стовпці містять власні вектори матриці A^TA.

Для розрахунку власних значень та власних векторів використовувався метод обертання Якобі, при якому обчислення власних значень та власних векторів симетричної матриці здійснюється ітераційним шляхом. Такий процес обчислення власних векторів відомий як діагоналізація. Суть методу Якобі [42, 43] зводиться до того щоб, для заданої матриці $S = S^{(0)}$ побудувати послідовність ортогональних подібних матриць $S^{(1)}$, ..., $S^{(m)}$, які сходяться до діагональної матриці, на діагоналях якої стоять власні значення матриці S. Для цього

використовується спеціальна матриця обертання J_i, така що норма наддіагональної частини:

$$||A^{(i)}||_{off} = \sqrt{\sum_{1 \le j < k \le n} (a_{jk}^{(i)})^2}, \quad (2.14)$$

зменшується при кожному двосторонньому обертанні матриці

$$A^{(i+1)} = J_i^T A^{(i)} J_i. \quad (2.15)$$

Для розрахунку матриці U на вхід методу обертання Якобі передається результат добутку AA^T. Для знаходження матриці V передається результат A^TA. Для того щоб знайти матрицю D достатньо просто взяти власні значення, які були знайдені при розрахунку матриці U і розмістити їх на головній діагоналі.

Після того як були знайдені матриці U, V, D відбувається розрахунок вагових коефіцієнтів. За основу взята формула із джерела [4]:

$$Aw = UD, \quad (2.16)$$

де A – вхідна матриця розмірністю N x n, w – матриця вагових коефіцієнтів розмірністю n x n, матриці U та D беруться з SVD.

Щоб знайти матрицю з ваговими коефіцієнтами треба:

$$w = A^{-1}UD, \quad (2.17)$$

Матрицю A^{-1} можна розписати за формулою запропонованою у джерелі [4]:

$$A^{-1} = VD^{-1}U^T, \quad (2.18)$$

Підставивши (2.18) у (2.17) отримаємо формулу розрахунку вагових коефіцієнтів:

$$w = VD^{-1}U^TUD, \quad (2.19)$$

Варто зазначити, що у випадку, коли матриця D буде прямокутною, буде розраховуватись псевдо-інверсна матриця.

2.4.1. Метод головних компонент

Метод головних компонент – спосіб зменшення розмірності даних, втративши при цьому найменшу кількість інформації. Метод запропонований К. Пірсоном в 1901 р. Використовується в багатьох областях: економетриці, біоінформатиці, обробці зображень, для стиснення даних, в суспільних науках [89, 105].

Обчислення головних компонент зводиться до обчислення власних значень та власних векторів коваріаційної матриці розрахованої з матриці вихідних даних. Або ж обчислення сингулярного розкладу цієї ж матриці.

Теоретичні основи методу головних компонент

Нехай задана матриця розмірності (*p*, *n*) спостережень випадкової векторної змінної $\mathbf{X} = [X_1 \dots X_p]^t$ з вектором середніх $\boldsymbol{\mu}_x = [\mu_1, \dots, \mu_p]^t$ і коваріаційною матрицею K_X, яка визначає структуру залежності між змінними X_j, $j = 1, \dots, p$. Потрібно знайти лінійне перетворення, яке дозволило б отримати стиснене представлення вихідних даних меншим числом змінних без істотної втрати інформації, що міститься у вихідній матриці. Перетворимо ці спостереження (*p*, *p*) - ортогональною матрицею виду:

$$\hat{\mathbf{O}} = [\varphi_1 \dots \varphi_p]^t, \tag{2.20}$$

де $\varphi_j = [\varphi_{1j} \dots \varphi_{pj}]', (j = 1, \dots, p)$ - система *p*- мірних ортонормованих векторів. Це означає, що для скалярного добутку (*,*) є справедливим:

$$(\varphi_i, \varphi_j) = \begin{cases} 1, & для \quad i = j, \\ 0, & для \quad i \neq j. \end{cases} \tag{2.21}$$

Тоді отримуємо випадкову векторну змінну ***Y*** з некорельованими компонентами:

$$\mathbf{Y} = [Y_1 \dots Y_p]' = \hat{\mathbf{O}}\mathbf{X}, \tag{2.22}$$

де Y_j – лінійна комбінація координат X_j, $j = 1, \dots, p$:

$$Y_j = \varphi_{1j} x_{j1} + \dots + \varphi_{pj} x_{jp}, \; j = 1, \dots, p. \tag{2.23}$$

З виразу (2.21) випливає, що $\hat{\mathbf{O}}\hat{\mathbf{O}}^t = \hat{\mathbf{O}}^t\hat{\mathbf{O}} = \mathbf{E}$ та $\hat{\mathbf{O}}^t = \hat{\mathbf{O}}^{-1}$, тому $\mathbf{X} = \hat{\mathbf{O}}^t\mathbf{Y}$ або

$$\mathbf{X} = \varphi_1 Y_1 + \dots + \varphi_p Y_p. \tag{2.24}$$

Коваріаційна матриця даних *X* рівна:

$$\mathbf{K_x} = M\{(\mathbf{X} - \boldsymbol{\mu}_x)(\mathbf{X} - \boldsymbol{\mu}_x)^t\} \tag{2.25}$$

Визначник $|\mathbf{K_x}|$ коваріаційної матриці $\mathbf{K_x}$ називають узагальненою дисперсією матриці даних ***X***.

Коваріаційна матриця $\mathbf{K_Y}$ випадкової векторної величини $\boldsymbol{Y}$ визначається виразом:

$$\begin{aligned}\mathbf{K_y} &= M\{(\mathbf{Y}-\boldsymbol{\mu}_y)(\mathbf{Y}-\boldsymbol{\mu}_y)^t\} = M\{\hat{\mathbf{O}}(\mathbf{X}-\boldsymbol{\mu}_x)(\mathbf{X}-\boldsymbol{\mu}_x)^t\hat{\mathbf{O}}^t\} = \\ &= \hat{\mathbf{O}}M\{(\mathbf{X}-\boldsymbol{\mu}_x)(\mathbf{X}-\boldsymbol{\mu}_x)^t\}\hat{\mathbf{O}}^t = \hat{\mathbf{O}}\mathbf{K_x}\hat{\mathbf{O}}^t.\end{aligned} \quad (2.26)$$

Так як $\mathbf{K}_x$ і $\hat{\mathbf{O}}$ є квадратними матрицями, то визначник коваріаційної матриці $\mathbf{K_y}$ рівний:

$$|\mathbf{K}_y| = |\hat{\mathbf{O}}\mathbf{K_x}\hat{\mathbf{O}}'| = |\hat{\mathbf{O}}\hat{\mathbf{O}}'||\mathbf{K}_x| = |\mathbf{K}_x|. \quad (2.27)$$

Це означає, що узагальнені дисперсії матриць $\boldsymbol{X}$ і $\boldsymbol{Y}$ є рівні.

Найкраще ортогональне перетворення повинно забезпечити найменшу надлишковість. Це означає, що матриця $\boldsymbol{Y}$ повинна мати некорельовані компоненти Y_j, $j=1,\ldots,p$. Іншими словами матриця $\mathbf{K}_y$ повинна бути діагональною:

$$\mathbf{K}_y = diag[\sigma^2_{y_1},\ldots,\sigma^2_{y_p}], \quad (2.28)$$

де $\sigma^2_{y_j}$ - дисперсія j- ї компоненти випадкової векторної величини $\boldsymbol{Y}$.

Позначимо $\lambda_j = \sigma^2_{y_j}$, $j=1,\ldots,p$. Тоді

$$|\mathbf{K}_y| = \prod_{j=1}^{p}\lambda_j. \quad (2.29)$$

Вважатимемо, що дисперсії впорядковані – $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_p \geq 0$. Якщо не всі λ_j рівні між собою, то матрицю $\boldsymbol{Y}$ можна стиснути відкиданням компонент з малими дисперсіями. Нехай $Y_1 - (n\times 1)$ - вектор є першою головною компонентою матриці $\boldsymbol{X}$ ($Y_1 = \sum_{i=1}^{p}\varphi_{i1}x_{1i}$).

Знайдемо дисперсію цієї головної компоненти:

$$\sigma^2_{y_1} = \varphi_1'\mathbf{K}_x\varphi_1 = \sum_{r=1}^{p}\sum_{i=1}^{p}\varphi_{1i}\varphi_{r1}M[(X_1-\mu_1)(X_1-\mu_1). \quad (2.30)$$

Вимагатимемо, щоб перша компонента Y_1 мала найбільшу дисперсію за умови збереження ортогональності векторів φ_i матриці $\boldsymbol{\Phi}$.

Тоді задача знаходження найкращого перетворення φ_1 зводиться до знаходження максимуму функції $\varphi_1' \mathbf{K}_x \varphi_1$ при умові:

$$(\varphi_1^t, \varphi_1) = \sum_{j=1}^{p} \varphi_{1j}^2 = 1. \qquad (2.31)$$

Для розв'язання цієї задачі оптимізації введемо функцію Лагранжа:

$$L(\varphi) = \varphi_1^t \mathbf{K}_x \varphi_1 - \lambda_1(\varphi_1^t \varphi_1 - 1), \qquad (2.32)$$

де λ_1 - множник Лагранжа. Необхідну умову екстремуму отримаємо, прирівнявши до нуля частинні похідні $\partial L / \partial \varphi_1$:

$$\partial L / \partial \varphi_1 = 2\left(\mathbf{K}_x \varphi_1 - \lambda_1 \varphi_1\right) = 2(\mathbf{K}_x - \lambda_1 \mathbf{E})\varphi_1 = 0, \qquad (2.33)$$

де ***E*** – одинична матриця. Так як нас цікавлять розв'язки, при яких $\varphi_1 \neq 0$, то повинна задовольнятися умова на визначник:

$$\left|\mathbf{K}_x - \lambda_1 \mathbf{I}\right| = 0. \qquad (2.34)$$

Звідси випливає, що λ_1 є власне число матриці $\mathbf{K}_x$, а φ_1 - власний вектор, який відповідає цьому власному числу. Вираз (2.33) може бути переписаний у вигляді: $\mathbf{K}_x \varphi_1 = \lambda_1 \varphi_1$.

Домножуючи останній вираз зліва на φ_1' і враховуючи співвідношення (2.21), отримаємо:

$$\varphi'_1 \mathbf{K}_x \varphi_1 = \lambda_1 \varphi_1' \varphi_1 = \lambda_1. \qquad (2.35)$$

Ліва частина виразу (2.35) є $\sigma_{y_1}^2$. Так як знаходився розв'язок задачі максимізації $\sigma_{y_1}^2$, то λ_1 є максимальним власним числом матриці $\mathbf{K}_x$. Для знаходження другої головної компоненти $\mathbf{Y}_2 = \varphi_2^t \mathbf{X}$ необхідно виконання двох умов – умови нормування $(\varphi_2^t, \varphi_2) = \sum_{i=1}^{p} \varphi_{2i}^2 = 1$ і умови ортогональності (φ_2^t, φ_2) =0. Вектор φ_2 тепер визначається так, щоб дисперсія $\sigma_{y_2}^2$ була максимальною при виконанні двох приведених умов. В цьому випадку потрібно використовувати два множники Лагранжа λ_2 і β. Потрібно максимізувати вираз:

$$\varphi'_2 \mathbf{K}_x \varphi_2 - \lambda_2(\varphi_2' \varphi_2 - 1) - \beta(\varphi_1' \varphi_2 - 1). \qquad (2.36)$$

Візьмемо похідну від виразу (2.18) і прирівняємо її до нуля. У відповідності з виразом (2.21) знаходимо, що β=0. Враховуючи умови нормування, отримаємо, що λ_2 є друге по величині власне значення матриці $\mathbf{K}_x$, рівне дисперсії другої головної компоненти $\lambda_2 = \sigma^2_{y_2}$, а φ_2 - відповідний власний вектор. Процес повторюється доти, поки не будуть знайдені всі власні числа і власні відповідні їм вектори, які є дисперсіями і коефіцієнтами лінійних комбінацій головних компонент.

Використання головних компонент для стиснення стаціонарного випадкового сигналу. Розглянемо задачу стиснення стаціонарного випадкового сигналу з дискретним часом і з корельованими елементами. Кореляція елементів сигналу, з одного боку обумовлює надлишковість, а з другого боку, саме кореляція несе інформацію про динамічні властивості джерела, генеруючого цей сигнал.

Тому ми вважаємо, що вихідні дані надані з оптимальним кроком дискретності в часі, що дозволяє із заданою точністю відновити кореляційні зв'язки. Це означає, що пряме стиснення сигналу шляхом відкидання частини елементів вектора сигналу є неможливим. При стисненні такого сигналу виникають серйозні труднощі як теоретичного, так і технічного плану. Багато проблем можна подолати за допомогою ортогонального перетворення вихідного вектора в вектор з некорельованими елементами, який можна стиснути методами квантування і кодування за умови нерівномірності розподілу дисперсії елементів вектора.

Нехай заданий сигнал у вигляді неперервної послідовності з дискретним часом $x(t_i)$, i = 1, ..., N. Цю послідовність можна розглядати як точку N-вимірного векторного простору E_N. Тоді кожний i- й відлік представляє собою i- ту координату N- мірного вектора X.

$$\mathbf{x} = [x(t_1)\ldots x(t_N)]^t = [x_1\ldots x_n]^t. \qquad (2.37)$$

Вважатимемо, що вектор (2.19) є реалізацією *N*- мірного стаціонарного випадкового вектора $\mathbf{X} = [\mathbf{X}_1 \ldots \mathbf{X}_N]^t$ з сумісною функцією густини ймовірностей $f(\mathbf{x}) = F(x_1, \ldots, x_N)$, з вектором середніх $\boldsymbol{\mu} = [\mu_1 \ldots \mu_N]'$ і коваріаційною матрицею

$$\mathbf{K}_x = M[(\mathbf{X} - \boldsymbol{\mu}_x)(\mathbf{X} - \boldsymbol{\mu}_x)^t] = [K_{ij}] = \sigma^2 \mathbf{R}_x, \; i, \; j = 1, \ldots, N . \qquad (2.38)$$

де $\mathbf{R}_x$ – кореляційна матриця, σ^2 – дисперсія. Кореляція елементів x_i òà x_j, $i, \; j = 1, \ldots, N$ обумовлює надлишковість вектора ***X***, яку вимірюють, використовуючи диференційну ентропію H_0(***X***) сигналу ***X***:

$$H_0(x) = \frac{N}{2} \log 2\pi\sigma^2 - \frac{1}{2} \log |\mathbf{R}|^{-1} .$$

(2.39)

Якщо елементи вектора ***X*** некорельовані, то $\log |\mathbf{R}|^{-1} = 0$ і диференційна ентропія є максимальною. Звідси випливає, що при заданій коваріаційній матриці $\mathbf{K}_x$ вектор ***X*** має надлишковість $0.5 \cdot \log |\mathbf{R}|^{-1}$ біт на елемент вектора. Відомо, що таку ж надлишковість має цифровий сигнал (дискретизований в часі та квантова ний по рівню).

Розглянемо стиснення випадкового сигналу, представленого у вигляді вектора-стовпця. Вирази, приведені вище для випадкової векторної змінної, справедливі і у випадку $\mathbf{Y} = [Y_1 \ldots Y_N]' = \hat{\mathbf{O}}\mathbf{X}$, де ***Φ*** – (N, N) – матриця ортогонального проектування.

Елементи некорельованого перетворення сигналу Y є головними компонентами і мають різні середньоквадратичні відхилення $\sigma^2_{y_1} \geq \ldots \geq \sigma^2_{y_N}$. Перетворений сигнал можна стиснути з допомогою квантування. Наприклад, значення сигналу, менші дисперсії квантування, можна не передавати і не зберігати. При цьому можливі різні стратегії квантування і кодування перетвореного сигналу. Найчастіше використовують дві стратегії стиснення – зональну і порогову.

При зональній стратегії компоненти ***Y*** (трансформанти) розбивають на ряд областей (зон) за величиною дисперсії $\sigma^2_{y_i}$, $i = 1, \ldots, N$. Трансформанти кожної

зони кантуються і кодуються кодовою комбінацією з числом розрядів, пропорційних середній дисперсії трансформант зони. Якщо середня дисперсія квантова них трансформант зони є меншою похибки квантування, то ці трансформанти прирівнюються до нуля.

При пороговій стратегії кодування для трансформант, які перевищили деякий заданий поріг, встановлюється єдиний рівень квантування і, отже, постійна довжина кодової комбінації. Значення, які опинилися нижче порога, прирівнюються до нуля.

Використання головних компонент в задачі класифікації. Розглянемо задачу виділення характерних ознак в образах з метою зниження розмірності. Вибір ознак можна виконувати не залежно від методів класифікації. Класифікація зводиться до зменшення віддалі між об'єктами в середині класу і максимізації віддалі між класами. Одними з найчастіше використовуваних методів зменшення розмірності простору ознак є методи, що базуються на ортогональних перетвореннях.

Переваги цих методів полягають в тому, що для їх використання не потрібно знати функцію розподілу, і вони мають чисельний дисперсійний критерій.

Серед всіх ортогональних методів оптимальним є метод головних компонент [61, 62]. Використаємо перетворення методу головних компонент для вибору розмірності простору ознак. Розглянемо $\boldsymbol{M}$ класів $\omega_1,\ldots,\omega_M$, образи яких представлені векторами $\mathbf{X}_i, i=1,\ldots,M$. Кожний образ описується p ознаками. Тоді спостереження можна представити матрицею:

$$\mathbf{X}=\{x_{ij}\}, \quad i=1,\ldots,M, \;\; j=1,\ldots,p. \tag{2.40}$$

Вважатимемо, що відома апріорна ймовірність $p(\omega_i)$ появи i- го класу. Тоді алгоритм виділення ознак є наступним:

1. Використовуючи образи, які входять в навчаючу вибірку, знаходимо коваріаційну матрицю $\mathbf{K}_x$:

$$\mathbf{K}_x=\sum_{i=1}^{M} p(\omega_i)M[\mathbf{X}_i\mathbf{X}_i^t] \tag{2.41}$$

або обчислюємо її оцінку.

1. Для $\mathbf{K}_x$ матриці знаходимо власні числа і ортонормовані власні вектори.

2. З r власних векторів формуємо $(r \times p)$ матрицю ортогонального перетворення

$$\hat{\mathbf{O}} = \left[\varphi_1 \quad \ldots \quad \varphi_r\right]^t. \tag{2.42}$$

4. Знаходимо головні компоненти $Y_i = \hat{\mathbf{O}} X_i, \quad i = 1,\ldots,r$. Так як $\mathbf{\Phi}$ - матриця розміру (r,p) і X_i - p- мірний вектор, то компоненти Y_i при $r < p$ представляють собою образи, які мають розмірність меншу p.

5. Виконуємо класифікацію одним з відомих методів кластерного або дискримінантного аналізів.

Для того, щоб використання методу головних компонент призводило до отримання оптимальних результатів, необхідно виконання умови $MY_i = 0$, що рівносильно умові $MX_i = 0$. Остання виконується автоматично, якщо окремі класи характеризуються нульовими математичними сподіваннями.

2.4.2. Метод Якобі для знаходження власних значень та власних векторів

Метод обертання Якобі – це чисельний метод розв'язання задачі власних значень та власних векторів для симетричної матриці з дійсних чисел.

Метод застосовується до симетричної матриці і полягає у виконанні ітераційних перетворень, які зводять її до діагонального вигляду:

$$\Lambda = UAU^T = diag(\lambda_i) \tag{2.43}$$

Припустимо, що A це симетрична матриця і G=G(i,j,ɵ) – матриця повороту Ґівенса. Тоді матриця:

$$A' = G^T AG \tag{2.44}$$

є симетрична та подібна до A.

Елементи матриці A' знаходяться так:

$$A'_{ii} = c^2 A_{ii} - 2scA_{ij} + s^2 A_{jj} \tag{2.45}$$

$$A'_{jj} = s^2 A_{ii} + 2scA_{ij} + c^2 A_{jj} \tag{2.46}$$

$$A'_{ij} = A'_{ji} = (c^2 - s^2)A_{ij} + sc(A_{ii} - A_{jj}) \tag{2.47}$$

$$A'_{ik} = A'_{ki} = cA_{ik} - sA_{jk} \quad 2.48$$

$$A'_{jk} = A'_{kj} = sA_{ik} + cA_{jk} \quad 2.49$$

$$A'_{kl} = A_{kl} \quad 2.50$$

де s = sin(Θ) та c = cos(Θ).

Оскільки вони подібні, то A та A’ мають однакову норму Фробеніуса, тобто сума квадратів всіх компонент, однак ми можемо обрати Θ таке, що:

$$A'_{ij} = 0$$

а A’ має більшу суму квадратів на діагоналі:

$$A'_{ij} = \cos(2\Theta)\, A_{ij} + \frac{1}{2}\sin(2\Theta)(A_{ii} - A_{jj}) \quad 2.51$$

Прирівнявши до 0 отримаємо:

$$\tan(2\Theta) = \frac{2A_{ij}}{A_{jj} - A_{ii}} \quad 2.52$$

Для оптимізації, A_{ij} обирають найбільшим за модулем недіагональним елементом, що називають опорним. Метод обертання Якобі постійно повторює обертання доки, поки матриця не стане майже діагональною. Тоді елементи на діагоналі стають наближеними до власних значень A. Ці значення є стовпцями матриці і рахуються так:

$$U = \prod_{k=1}^{n} G_k \quad 2.53$$

2.5. Розроблення імітаційної моделі обчислення вагових коефіцієнтів нейроподібних мереж

Результатом роботи є розроблене програмне забезпечення [140] для розрахунку вагових коефіцієнтів, які використовуються для шифрування вхідного повідомлення. Для реалізації поставленої задачі було обрано мову програмування C# [147, 148, 150] і середовище розробки Visual Studio 2022 [149]. Розроблені програмні засоби використовують вдосконалений метод сингулярного розкладу матриці, а для знаходження власних значень та власних векторів використовується метод обертання Якобі. Розроблені засоби дають змогу швидко обчислити коефіцієнти для заданої архітектури нейромережі. Також розроблено

зручний інтерфейс, який надає змогу зрозуміло та детально ознайомитись із роботою алгоритму.

Для криптографічного шифрування та дешифрування даних використано нейроподібні мережі, архітектура яких визначається кількістю нейроелементів *N,* кількістю входів *к* та їх розрядністю *m*. Кількість нейронних елементів у нейроподібній мережі визначається за такою формулою:

$$N = \frac{n}{m} \tag{2.54}$$

де n – розрядність вхідного повідомлення, m – розрядність входів.

Вхідні повідомлення, які шифруються, можуть мати різну розрядність *n,* а для їх шифрування використовуються нейроподібної мережі з різною архітектурою. Від значення розрядності повідомлення *n* та кількості входів *к* залежить архітектура нейроподібної мережі. Для повідомлення розрядністю *n*=16 можливі такі варіанти архітектури нейроподібної мережі:

m	k	N
2	8	8
4	4	4
8	2	2

А коли *n*=24 такі:

m	k	N
2	12	12
3	8	8
4	6	6
6	4	4
8	3	3

Роботу розробленої імітаційної моделі продемонстровано далі. Для прикладу взято вхідне повідомлення із розрядністю – 16 та розрядністю входів – 2. Тоді вхідна матриця буде мати розмірність 8x2. Вхідне повідомлення задається

користувачем. У результаті розраховується матриця вагових коефіцієнтів розмірністю 2x2. У подальшому ця матриця буде використовуватись при шифруванні та дешифруванні вхідного повідомлення. Однак для повідомлення є вимоги, а саме: розрядність повідомлення та розрядність входів має бути такими, як і при знаходженні матриці вагових коефіцієнтів.

Для тестового прикладу використовувалось шістнадцяти розрядне повідомлення з розрядністю входів – 2. Опираючись на ці дані розраховувались кількість нейроелементів та кількість входів – 8.

Вхідними даними (рис. 2.5) є навчальна матриця, розрядність повідомлення (n) та розрядність входів (m).

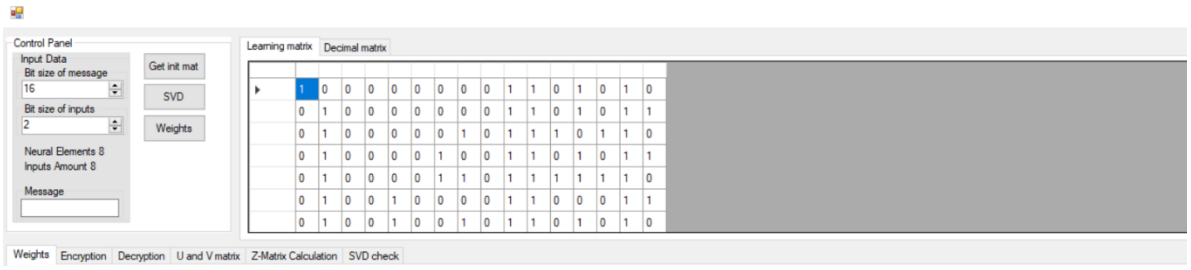


Рис. 2.5. Вхідні дані програми.

Розмірність навчальної матриці – 14x16, де кожен рядок – шістнадцяти розрядна команда (рис. 2.6).

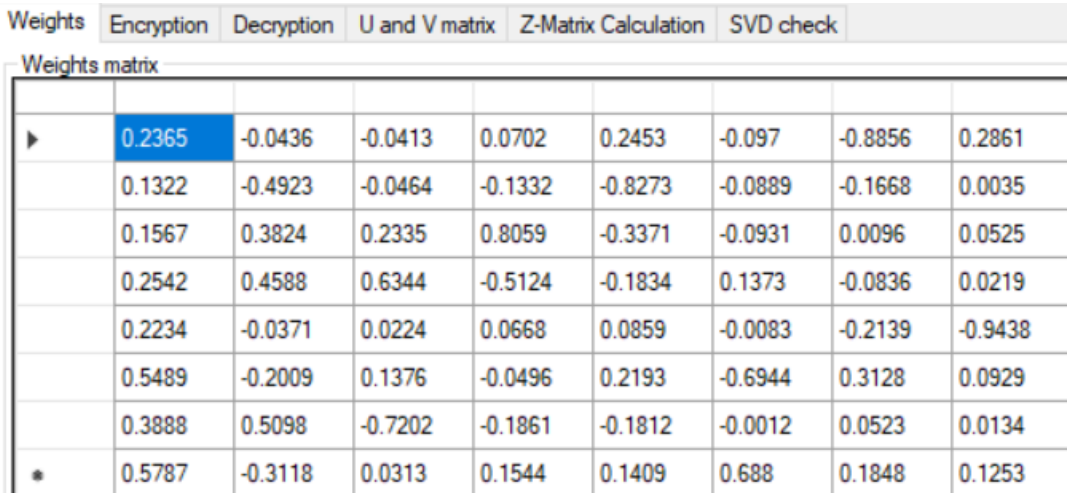
1	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0
0	1	0	0	0	0	0	0	0	1	1	0	1	0	1	1
0	1	0	0	0	0	0	1	0	1	1	1	0	1	1	0
0	1	0	0	0	0	1	0	0	1	1	0	1	0	1	1
0	1	0	0	0	0	1	1	0	1	1	1	1	1	1	0
0	1	0	0	1	0	0	0	0	1	1	0	0	0	1	1
0	1	0	0	1	0	0	1	0	1	1	0	1	0	1	0
0	1	0	0	1	0	1	0	0	1	1	1	1	1	1	1
0	1	0	0	1	0	1	1	0	1	1	0	0	0	1	0
0	1	0	1	0	0	0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

Рис. 2.6. Навчальна матриця.

Після введення даних програма розраховує кількість нейроелементів та входів. За цю операцію відповідає кнопка *Get init mat*.

Далі відбувається знаходження матриць U, V та D за допомогою алгоритму SVD, який спрацьовує при натисканні відповідної кнопки. Для знаходження U на вхід SVD подається результат добутку AA^T, де A – навчальна матриця, а для знаходження V – результат добутку A^TA. Матриця D складається із власних значень, розміщених на головній діагоналі.

Для розрахунку вагових коефіцієнтів користувачу необхідно натиснути кнопку *Weights*. Коефіцієнти рахуються за формулою (2.19). Для перевірки правильності пророблених розрахунків використовується формула (2.16). Результат продемонстровано на рис. 2.7.



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

Рис. 2.7. Матриця вагових коефіцієнтів.

Тепер цю матрицю можна застосувати для шифрування вхідного повідомлення, яке користувач може ввести у спеціальне текстове поле – *Message*.

Алгоритмічна реалізація методу головних компонент. Нехай $X_1, X_2, \dots, X_p$ вхідні дані (ознаки) і кожний з векторів $\mathbf{X}_i$ має розмірність n. Розглянемо покроковий алгоритм знаходження головних компонент з ілюстрацією на

конкретному прикладі. Об'єднаємо ці p векторів ознак у матрицю D розмірністю (n, p):

$$\mathbf{D} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{21} & \dots & x_{2p} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}. \qquad (2.55)$$

Крок 1. Знаходження для кожного з векторів ознак $\mathbf{X}_i$ середнього значення (математичного сподівання):

$$\mathbf{X}_j^{ave} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}, \quad j = 1, \dots, p. \qquad (2.56)$$

Крок 2. Центрування вихідних даних (елементів матриці ***D***) так, щоб середні значення ознак стали рівними нулю. Для цього віднімемо від кожного з елементів вектора $\mathbf{X}_i$ середнє значення його елементів:

$$\bar{x}_{ij} = x_{ij} - \mathbf{X}_j^{ave}, i = 1, \quad , n; \quad j = 1, \dots, p. \qquad (2.57)$$

Крок 3. Нормування відцентрованих вихідних даних. Виконується у випадку, коли величини ознак сильно відрізняються своїми порядками:

$$\bar{x}_{ij}^{n} = \frac{\bar{x}_{ij}^{n}}{\sqrt{\left(x_{ij} - \mathbf{X}_j^{ave}\right)^2}}, i = 1, \quad , n; \quad j = 1, \dots, p. \qquad (2.58)$$

Операція нормування переводить випадкову величину x_{ij} в безрозмірну випадкову величину $\bar{x}_{ij}^{n}$.

Крок 4. Величиною лінійного зв'язку між ознаками є коефіцієнт кореляції або в більш загальному випадку коефіцієнт коваріації. Кореляційна матриця є симетричною з одиничною головною діагоналлю. Обчислюємо кореляційну матрицю для відцентрованих та нормованих вхідних даних $\bar{x}_{ij}^{n}$. Вона буде мати розмірність (p, p) і обчислюється згідно наступного виразу:

$$CorrD = \left(\bar{D}_n\right)^t \cdot \bar{D}_n = \sum_{k=1}^{n} \bar{x}_{ki}^{n} \cdot \bar{x}_{kj}^{n}, i = 1, \dots, p; \; j = 1, \dots, p. \qquad (2.59)$$

Крок 5. Обчислюємо власні вектори, що відповідають цим власним значенням за методом Якобі [42]:

$$\mathrm{C}_1 = \begin{pmatrix} c_{11} \\ c_{12} \\ \dots \\ c_{1p} \end{pmatrix}; \quad \mathrm{C}_2 = \begin{pmatrix} c_{21} \\ c_{22} \\ \dots \\ c_{2p} \end{pmatrix}; \ \dots, \qquad \mathrm{C}_p = \begin{pmatrix} c_{p1} \\ c_{p2} \\ \dots \\ c_{pp} \end{pmatrix}. \tag{2.60}$$

Крок 6. Визначаємо головні компоненти $Z_1, Z_2, ..., Z_p$ - це нові ознаки, які є лінійними комбінаціями вихідних факторів. Вони теж будуть включати по n змінних і обчислюються згідно виразів:

$$\begin{aligned} & z_{1i} = c_{11} \cdot x_{1i} + c_{12} \cdot x_{2i} + \dots c_{1p} \cdot x_{pi} \\ & z_{2i} = c_{21} \cdot x_{1i} + c_{22} \cdot x_{2i} + \dots c_{2p} \cdot x_{pi} \quad \forall i = 1, ..., n \\ & \dots \quad \dots \quad \dots \quad \dots \\ & z_{pi} = c_{p1} \cdot x_{1i} + c_{p2} \cdot x_{2i} + \dots c_{pp} \cdot x_{pi} \end{aligned}. \tag{2.61}$$

В основі імітаційної моделі є програмний додаток PCA, призначений для знаходження власних значень та власних векторів кореляційної матриці масиву вхідних даних та головних компонент цього масиву. Алгоритм, реалізований в додатку, детально описаний вище. Додаток PCA реалізований мовою C# [112] в середовищі Visual Studio 2019 [149].

При запуску цього додатку відкриється головне вікно інтерфейсу, яке зображене на рис. 2.8.

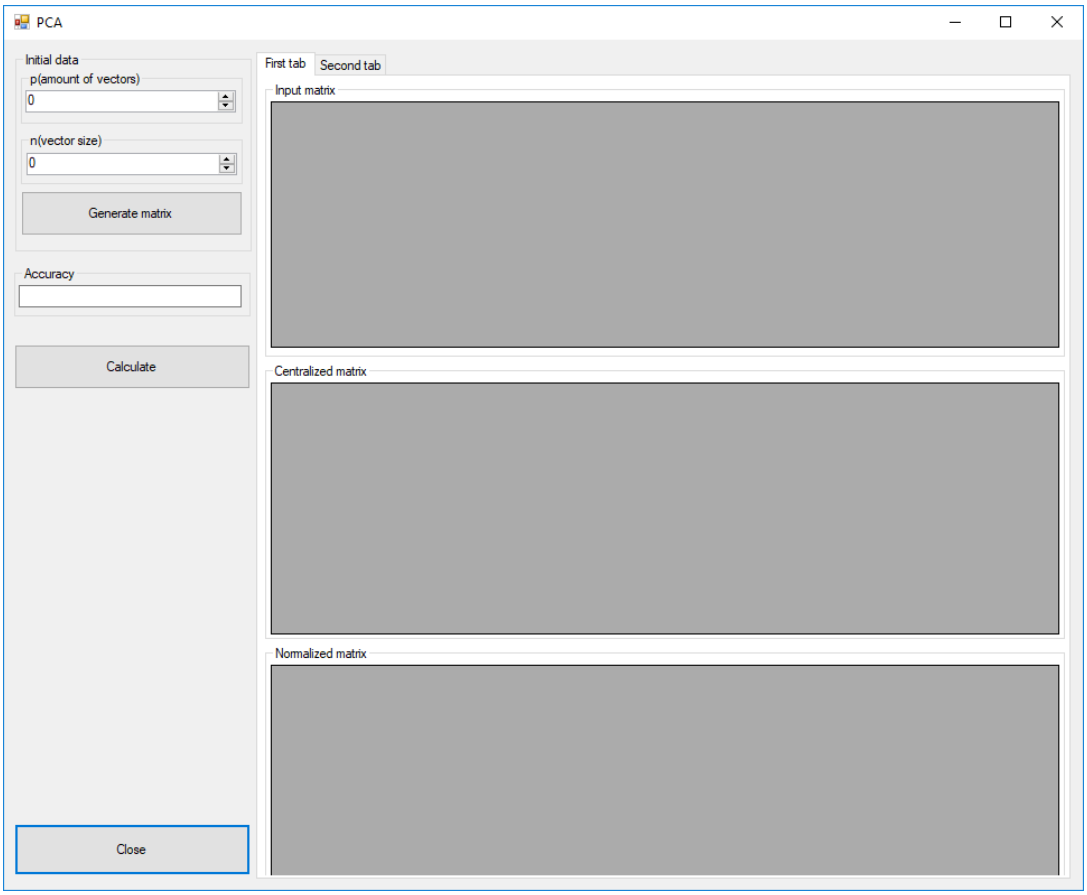


Рис. 2.8. Вигляд інтерфейсу програмного додатку PCA

Вхідні дані генеруються випадковим чином з 0 і 1.

Для роботи цього додатку необхідно ввести в головному вікні інтерфейсу наступні дані:

p – кількість векторів $\{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n\}$ вхідних даних;

n – розмірність кожного з цих векторів.

Результат формування вхідних даних зображено на рис. 2.9. У цьому випадку генерується матриця 6x8.

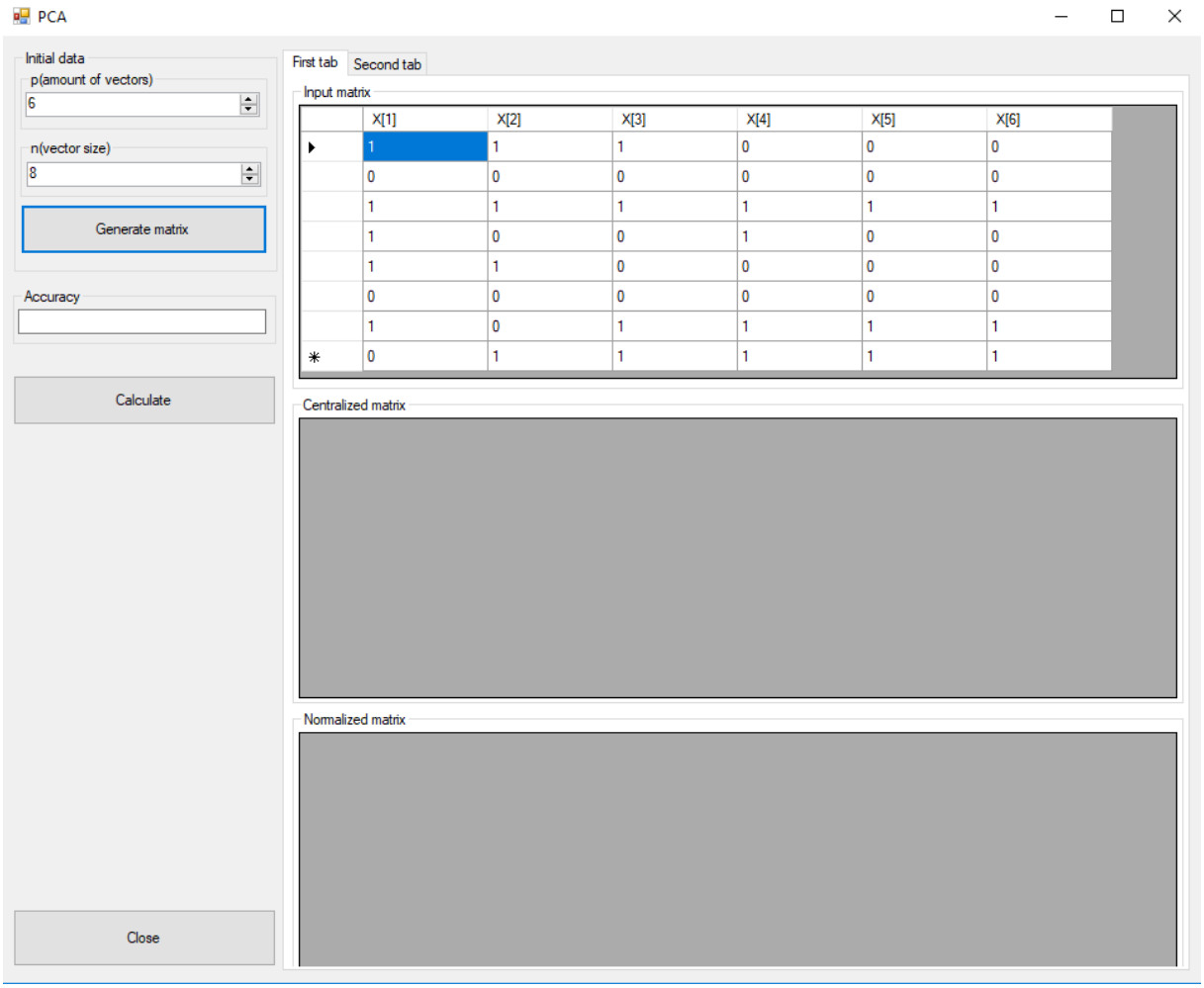


Рис. 2.9. Згенерована вхідна матриця

Кожен стовпець цієї матриці відповідає за команду. А кількість рядків відповідають за розрядність команди.

Наступним кроком ця матриця буде відцентрована і нормована. Результат зображено на рис. 2.10.

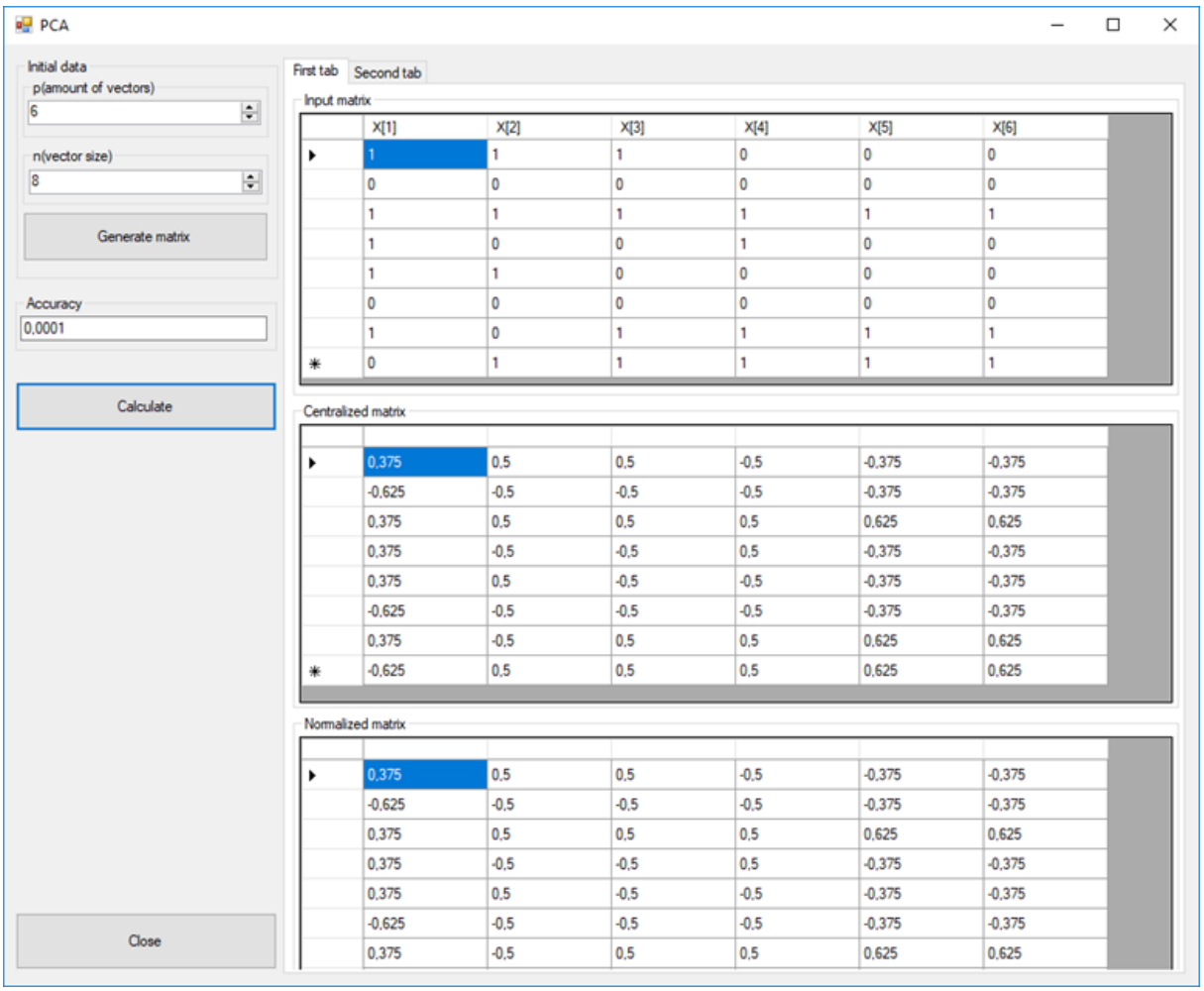


Рис. 2.10. Центрована і нормована вхідна матриця

Далі обраховується кореляційна матриця. Оскільки вона симетрична, то можна застосувати метод обертання Якобі, щоб знайти власні вектори вхідної матриці. Далі для знаходження головних компонентів, перемножуються між собою вхідна матриця(нормована) і матриця власних векторів. В результаті ми отримуємо матрицю головних компонентів, де кожен стовпчик відповідає за відповідну головну компоненту (рис. 2.11.)

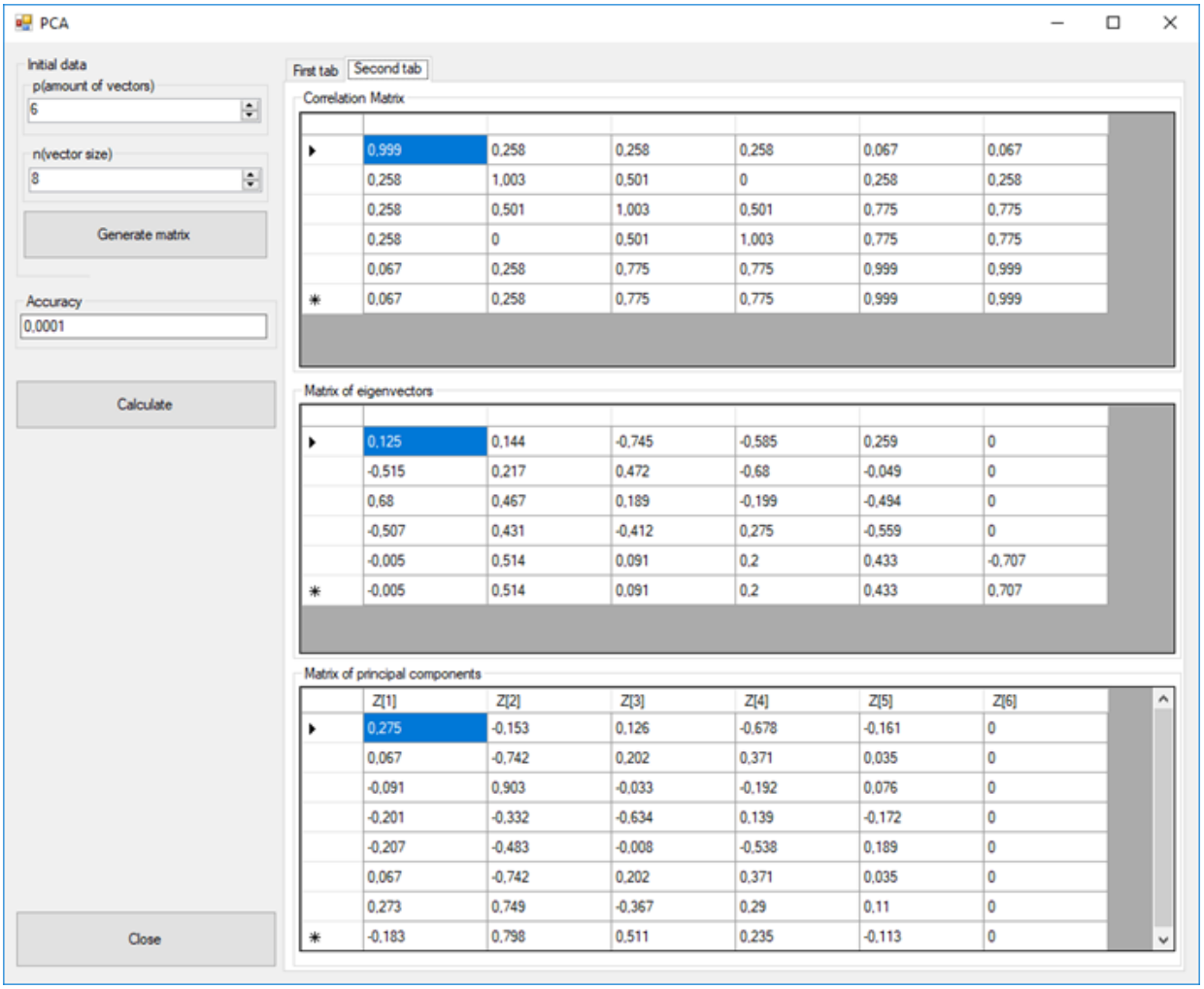


Рис. 2.11. Результати обчислень: кореляційна матриця, матриця власних векторів та матриця головних компонентів.

Таким чином, результати експериментальних досліджень розробленої імітаційної моделі шифрування-дешифрування даних на основі методу головних компонент підтверджують коректність обраного підходу до криптографічного захисту даних і доцільність проведення подальших досліджень його ефективності [66-69]. Також важливим завданням є дослідження нейромережевої реалізації методу головних компонент, зокрема, на основі моделі послідовних геометричних перетворень МПГП.

2.6. Модель попередніх налаштувань для нейроподібного шифрування даних

Мобільна система криптографічного захисту [87] та передачі даних (СКЗПД) використовує нейроподібне шифрування з симетричними ключами, у якій ключі шифрування та дешифрування є ідентичними або ж ключ дешифрування просто обчислити із ключа шифрування. Шифрування відбувається над відкритим текстом з використанням ключа, яких визначається заданою кількістю нейроподібних елементів N, матрицею вагових коефіцієнтів W_{ji} і операціями маскування.

Виконання нейроподібного криптографічного шифрування передбачає здійснення попередніх налаштувань. Такі налаштування зводяться до вибору структури нейроподібної мережі, обчислення матриці вагових коефіцієнтів і таблиці макрочасткових добутків. Узагальнена аналітична модель попередніх налаштувань запишеться так:

$$P_{Mi} = f_{P_{Mi}}(f_W(f_{(X \to Nm)})), \tag{2.62}$$

де P_{Mi} – макрочастковий добуток; $f_{P_{Mi}}$ – обчислення таблиці макрочасткових добутків P_{Mi}; f_W – обчислення матриці вагових коефіцієнтів; $f_{(X \to Nn)}$ – формування структури нейроподібної мережі, параметри якої визначаються розрядністю m повідомлення X та розрядністю входів нейроелемента n.

Із формули (2.62) видно, що модель попередніх налаштувань реалізується на базі трьох компонентів: перший – формувача структури нейроподібної мережі; другий – обчислення матриці вагових коефіцієнтів; третій – обчислення таблиці макрочасткових добутків.

Перша компонента забезпечує формування структури нейроподібної мережі для шифрування/дешифрування даних. Структура нейроподібної мережі визначається кількістю нейроподібних елементів, які обчислюються за формулою:

$$N = \frac{m}{n}, \tag{2.63}$$

де N – кількість нейроелементів. Для системи команд управління, розрядність якої m=16, кількість нейроподібних елементів N може бути 16, 8, 4 і 2, а розрядність входів n відповідно 1, 2, 4, 6 і 8.

Друга компонента забезпечує обчислення матриці вагових коефіцієнтів для нейроподібної мережі. Для її обчислення використаємо метод сингулярного розкладу матриці:

$$A = UDV^T, \tag{2.64}$$

де A – матриця вхідних даних N *x* n; U – ліва сингулярна матриця N *x* N; D – діагональна матриця N *x* n; V – права сингулярна матриця n *x* n.

Далі матрицю вагових коефіцієнтів вираховуємо з:

$$AW = UD, \tag{2.65}$$

де A – вхідна матриця розмірністю $N \times n$, W – матриця вагових коефіцієнтів розмірністю $n \times n$. Обчислення матриці вагових коефіцієнтів W виконується за такою формулою:

$$W = A^{-1}UD, \tag{2.66}$$

де матриця A^{-1} рівна:

$$A^{-1} = VD^{-1}U^T. \tag{2.67}$$

Підставивши (2.67) у (2.66) отримаємо формулу для обчислення вагових коефіцієнтів, яка запишеться так:

$$W = VD^{-1}U^TUD. \tag{2.68}$$

Розмірність таблиць вагових коефіцієнтів визначається кількістю нейроподібних елементів, на основі яких синтезована нейроподібна мережа. Так для шифрування/дешифрування команд управління використовуються нейроподібні мережі з кількість нейроподібних елементів 16, 8, 4, і 2. Розмірність матриць вагових коефіцієнтів для таких нейроподібних мереж відповідно буде 16×16, 8×8, 4×4, 2×2.

Третя компонента забезпечує обчислення таблиць макрочасткових добутків. Обчислення таблиць макрочасткових добутків для вагових коефіцієнтів

з плаваючою комою $W_j = w_j 2^{E_{Wj}}$ (де w_j – мантиса W_j вагового коефіцієнта, E_{W_j} – порядок W_j вагового коефіцієнта) передбачає виконання таких операцій:

1 визначення найбільшого спільного порядку вагових коефіцієнтів E_{Wmax};

2 обчислення різниці порядків для кожного W_j вагового коефіцієнта $\Delta E_{W_j} = E_{Wmax} - E_{W_j}$;

3 зсунення вправо мантиси w_j на різницю порядків ΔE_{W_j};

4 визначення максимальної кількості розрядів переповнення q для макрочасткових добутків P_{Mi};

5 отримання масштабованих мантис w_j^h шляхом їх зсуву вправо на q розрядів переповнення обчислених макрочасткових добутків P_{Mi};

6 додавання до найбільшого спільного порядку E_{Wmaxc} кількості розрядів переповнення $E_{Wmax}^h = E_{Wmax} + q$.

Таблиця макрочасткових добутків обчислюється за наступною формулою:

$$P_{Mi} = \begin{cases} 0, \quad якщо \quad x_{1i} = x_{2i} = x_{3i} = ... = x_{Ni} = 0 \\ w_1^h, \quad якщо \quad x_{1i} = 1, \quad x_{2i} = x_{3i} = ... = x_{Ni} = 0 \\ w_2^h, \quad якщо \quad x_{1i} = 0, x_{2i} = 1, x_{3i} = ... = x_{Ni} = 0 \\ w_1^h + w_2^h, \quad якщо \quad x_{1i} = 1, x_{2i} = 1, x_{3i} = ... = x_{Ni} = 0 \\ \vdots \\ w_2^h + ... + w_N^h, \cdots якщо \quad x_{1i} = 0, x_{2i} = x_{3i} = ... = x_{Ni} = 1 \\ w_1^h + w_2^h + ... + w_N^h, \quad якщо \quad x_{1i} = x_{2i} = x_{3i} = ... = x_{Ni} = 1 \end{cases}, \tag{2.69}$$

де x_{1i}, x_{2i}, ..., x_{Ni} – адресні входи таблиці, w_j^h – мантиса W_j вагового коефіцієнта приведена до найбільшого спільного порядку.

Кількість таблиць макрочасткових добутків для шифрування/дешифрування команд дорівнює кількості нейроподібних елементів у мережі. Обсяг пам’яті необхідної для зберігання таблиці макрочасткових добутків дорівнює:

$$Q=2^k, \tag{2.70}$$

де k – кількість входів нейроподібного елемента.

Для нейроподібних мереж з 16, 8, 4 і 2 нейроподібними елементами кількість таблиць та їх обсяги відповідно дорівнюють: 16 табл. обсягом $Q=2^{16}$; 8 табл. обсягом $Q=2^{8}$; 4 табл. обсягом $Q=2^{4}$; 2 табл. обсягом $Q=2^{2}$.

2.7. Узагальнена модель нейроподібного шифрування команд управління з використанням таблично-алгоритмічного методу.

Для нейроподібного шифрування команд управління з використанням таблично-алгоритмічного методу розроблена узагальнена аналітична модель, яка записується так:

$$y_j = f_{(N\to 1)}(f_{(y_{1i},\ldots,y_{Ni})}(f_{(P_{M1i},\ldots,P_{MNi})}(f_{(x_{1i}\ldots x_{Ni})}(f_{(P\to S)})))), \qquad (2.71)$$

де y_j – зашифрована j-а частина команди управління; $f_{(P\to S)} = R^m \to NR^n$ перетворення m розрядного повідомлення на N частин розрядністю n; $f_{(y_{1i},\ldots,y_{Ni})}$ – виконання N підсумовувань макрочасткових добутків P_{Mji} у відповідності до формули $y_{ji} = 2^{-1} y_{j(i-1)} + P_{Mji}$, де y_{j0}=0; $f_{(P_{M1i},\ldots,P_{MNi})}$ – паралельне читання з таблиць, обчислених за формулою (2.69), N макрочасткових добутків $P_{M1i},\ldots,P_{MNi}$; $f_{(x_{1i}\ldots x_{Ni})}$ – формування розрядного зрізу для N частин повідомлення, який є адресом для читання макрочасткових добутків $P_{M1i},\ldots,P_{MNi}$ з таблиць; $f_{(N\to 1)}$ – послідовна передача частинами зашифрованого повідомлення.

Структуру моделі нейроподібного шифрування команд управління з використанням таблично-алгоритмічного методу, яка реалізує вираз (2.71), подано на рис. 2.12.

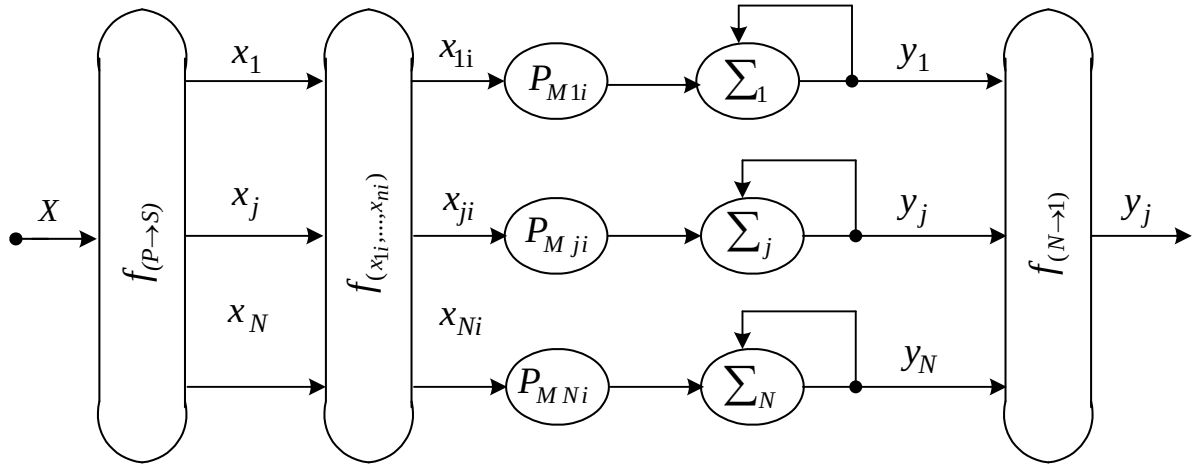


Рис. 2.12. Модель нейроподібного шифрування команд управління з використанням таблично-алгоритмічного методу

Основними компонентами даної моделі є: перетворювач *m* розрядного повідомлення на *N* частин розрядністю *n*, формувач адреси читання з таблиць, N таблиць макрочасткових добутків, N комутаторів та суматорів для послідовної передачі частинами зашифрованого повідомлення. З аналізу моделі нейроподібного шифрування команд управління видно, що збільшення кількості нейроподібних елементів веде до збільшення кількості частин зашифрованого повідомлення.

2.8. Висновки до розділу 2.

1. Вдосконалено і орієнтовано на задачі нейромережевого шифрування-дешифрування даних нейроподібну мережу прямого поширення автоасоціативного типу на основі парадигми «модель послідовних геометричних перетворень» шляхом неітеративного обчислення вагових коефіцієнтів, що забезпечило повторюваність результатів і орієнтацію на апаратну реалізацію.

2. Запропоновано синтез СЗПД у реальному часі з високими техніко-економічними характеристиками здійснювати з використанням інтегрованого підходу, який охоплює: 1) розроблення і дослідження теоретичних основ нейроподібного шифрування (дешифрування) даних та генерацію шумоподібних кодів; 2) розроблення нових алгоритмів та структур нейроподібного шифрування-

дешифрування даних, орієнтованих на сучасну елементну базу; 3) сучасну елементну базу з можливістю програмування структури; 4) засоби автоматизованого проектування програмно-апаратних компонентів.

3. Вибрано для синтезу СЗПД у реальному часі такі принципи: модульності; змінності складу обладнання; конвеєризації та просторого паралелізму; відкритості програмного забезпечення; спеціалізації та адаптації апаратно-програмних засобів до структури алгоритмів нейроподібного шифрування та дешифрування даних; програмованості архітектури блоків шифрування (дешифрування) та кодування (декодування) шляхом використання програмованих логічних інтегральних мікросхем.

4. Показано, що задача синтезу засобів нейроподібного шифрування (дешифрування) у реальному часі із високою ефективністю використання обладнання можна звести до мінімізації апаратних затрат при забезпеченні множини вимог, характеристик і обмежень.

5. Розроблена структуру СЗПД з використанням шумоподібних кодів, яка за рахунок можливості налаштування кодів маскування та вагових коефіцієнтів, програмованості архітектури нейроподібної мережі та генерації шумоподібних кодів різної розрядності забезпечує високу завадостійкість, роботу у режимі реального часу та високі техніко-економічні характеристики.

6. Розроблено модель попередніх налаштувань для реалізації нейроподібного шифрування/дешифрування основними компонентами є: блок формування архітектури нейроподібнох мережі; блок обчислення таблиць макрочасткових добутків; блок обчислення матриць вагових коефіцієнтів.

7. Розроблено модель нейроподібного шифрування даних та команд управління із використанням таблично-алгоритмічного методу. Основними компонентами є перетворювач повідомлення, формувач адреси читання з таблиць, N таблиць макрочасткових добутків, N комітаторів і суматорі, реалізація забезпечує тестування СКЗПД у реальному часі.

РОЗДІЛ 3
РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НЕЙРОПОДІБНОГО КРИПТОГРАФІЧНОГО ЗАХИСТУ ДАНИХ

3.1. Розроблення структури інформаційної технології нейроподібного криптографічного захисту даних

При розробленні інформаційної технології нейромережевого криптографічного захисту даних [95-99] з високими техніко-економічними характеристиками пропонується використовувати інтегрований підхід, який охоплює:

– дослідження теоретичних основ нейроподібного криптографічного захисту даних та їхнє розроблення;

– дослідження та розроблення нових алгоритмів та структур нейроподібного шифрування та дешифрування даних, орієнтованих на сучасну елементну базу;

– сучасну елементну базу з можливістю програмування структури;

– засоби автоматизованого проектування програмно-апаратних засобів.

Розробляється макет експериментальної системи криптографічного нейромережевого захисту та передачі даних (СКНЗПД) у реальному часі із високими техніко-економічними показниками. Розроблення макету буде виконуватися з використанням інтегрованого підходу, який включає в себе нейроподібні методи шифрування (дешифрування) даних, метод синтезу шумоподібних кодів, алгоритми та структури для реалізації нейроподібних елементів, сучасну елементну базу з можливістю програмування архітектури та засоби проєктування програмно-апаратних засобів.

Структура інформаційної технології нейромережевого криптографічного захисту даних наведена на рис. 3.1.

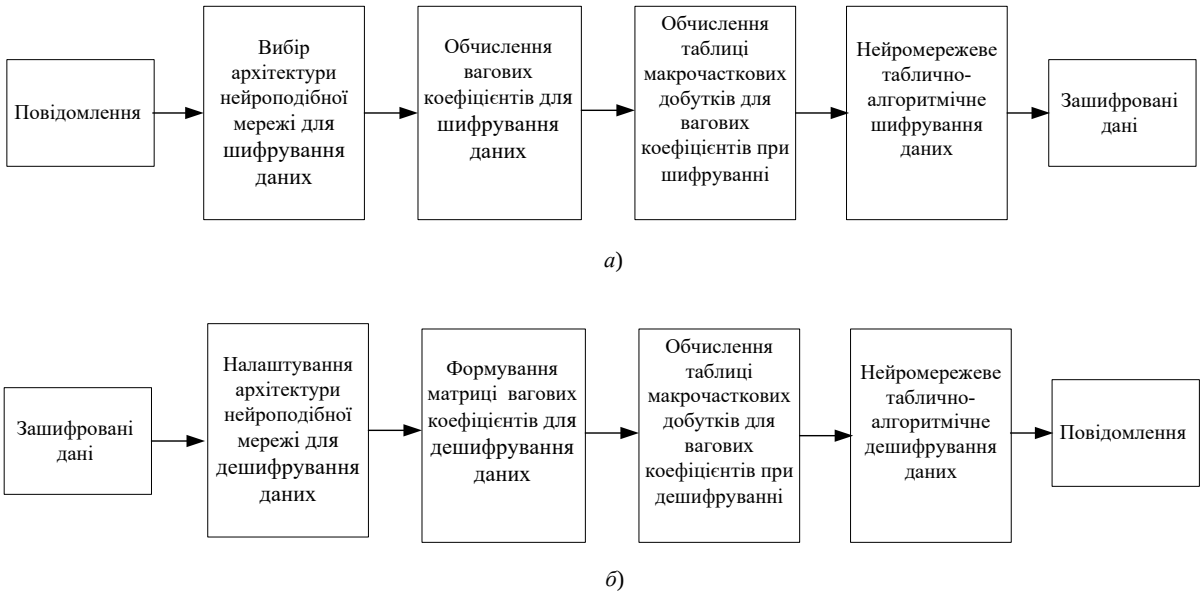


Рис. 3.1. Структура інформаційної технології нейромережевого криптографічного захисту даних: а) процес шифрування; б) процес дешифрування.

Показана інформаційна технологія нейромережевого криптографічного захисту даних орієнтована на шифрування із симетричними ключами. Ключі шифрування та дешифрування – однакові, або ж ключ дешифрування легко обчислити з ключа шифрування.

3.2. Основні етапи нейроподібного криптографічного шифрування даних

Шифрування відбувається над відкритим текстом з використанням ключа. Цей ключ складається із заданої кількості нейронів у нейромережі N, матриці вагових коефіцієнтів W_{ji} і операцій маскування. Розглянемо основні етапи шифрування повідомлення.

3.2.1. Вибір архітектури нейроподібної мережі

Архітектура нейроподібної мережі визначається кількістю нейроелементів N, кількістю входів k та розрядність входів m. Кількість нейронних елементів визначаються за такою формулою:

$$N = \frac{n}{m}, \tag{3.1}$$

де n – розрядність вхідного повідомлення, m – розрядність входів. Вхідні повідомлення, що шифруються, можуть мати різну розрядність та ділитися на різну кількість входів, які дорівнюють кількості нейроелементів [44].

3.2.2 Обчислення матриці вагових коефіцієнтів

Для шифрування (дешифрування) даних будемо використовувати автоасоціативну нейронну мережу. Ця мережа навчається з використанням МГК, який виконує лінійне перетворення за формулою:

$$\bar{y} = W \cdot \bar{x}. \tag{3.2}$$

У формули (3.2) з допомогою матриці $W \in R^{n*n}$ виконується перетворення вхідного вектору $\bar{x} \in R^n$ в вихідний вектор $\bar{y} \in R^n$. Перетворення виконується за системою лінійно незалежних векторів. Цією системою треба вибрати ортонормовану систему власних векторів, що відповідають власним значенням коваріаційної матриці вхідних даних.

Нехай вхідні дані задано у вигляді сукупності N векторів $\bar{x}_j$, де j=1, ..., N. Кожен вектор має розмірність n, $\bar{x}_j = (x_{j1}, x_{j2}, ..., x_{jn})$:

$$X = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_N)^t. \tag{3.3}$$

Автоковаріаційну матрицю для *N* векторів $\bar{x}_j$ можна записати у вигляді:

$$R = X^t \cdot X, \tag{3.4}$$

де кожен її елемент визначається так:

$$r_{jl} = \sum_{i=1}^{N} \bar{x}_{jl} \bar{x}_{il} = \sum_{i=1}^{N} (\bar{x}_{ji} - \mu_j)(\bar{x}_{il} - \mu_l), \tag{3.5}$$

де *j, l*=1, …, *n*, μ_i – математичні сподівання векторів $\bar{x}_j$, $\bar{x}_l$.

Власні значення невід'ємної симетричної матриці *R* є дійсними додатніми числами. Тепер відсортуємо їх за спаданням $\lambda_1 > \lambda_2 > ... > \lambda_n$. Аналогічно розмістимо і власні вектори, що відповідають λ_i. Тоді матриця *W* визначає лінійне перетворення (3.2), де $\bar{y} = (y_1, y_2, ..., y_n)$ вектор головних компонент МГК і він відповідає вхідному вектору даних $\bar{x}$. Кількість векторів головних компонент та

кількість векторів вхідних даних – рівні. Матриця вагових коефіцієнтів, яка використовується для шифрування даних буде мати такий вигляд:

$$\begin{vmatrix} W_{11} & W_{12} & \cdots & W_{1k} \\ W_{21} & W_{22} & \cdots & W_{2k} \\ \vdots & \vdots & \cdots & \vdots \\ W_{N1} & W_{N2} & \cdots & W_{Nk} \end{vmatrix} \quad (3.6)$$

Обчислення скалярного добутку є базовою операцією нейроподібної мережі, яка використовується для шифрування даних. Цю операцію треба реалізовувати з використанням таблично-алгоритмічного методу, оскільки матриця вагових коефіцієнтів є наперед обчисленою (W_{js}, де j=1, …, N; s=1, …, k).

Для навчання моделі обчислення вагових коефіцієнтів нейроподібних мереж використовувалась матриця з 16-ти розрядних повідомлень:

З використанням моделі обчислення вагових коефіцієнтів обчислюємо матрицю вагових коефіцієнтів для нейроподібної мережі з 8-а нейроелементами

$$\begin{vmatrix} W_{11} & W_{12} & \cdots & W_{18} \\ W_{21} & W_{22} & \cdots & W_{28} \\ \vdots & \vdots & \cdots & \vdots \\ W_{81} & W_{82} & \cdots & W_{88} \end{vmatrix}. \quad (3.7)$$

Обчислені вагові коефіцієнти для нейроподібної мережі з 8-а нейроелементами показано в табл. 3.1.

Таблиця 3.1.

Вагові коефіцієнти для нейроподібної мережі з 8-а нейроелементами

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

Для нейроподібної мережі з 4-а нейроелементами матриця вагових коефіцієнтів має вид:

$$\begin{vmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \\ W_{41} & W_{42} & W_{43} \cdots & W_{44} \end{vmatrix} \quad (3.8)$$

Обчислені вагові коефіцієнти для нейроподібної мережі з 4-а нейроелементами (табл.3.2).

Таблиця 3.2.

Вагові коефіцієнти для нейроподібної мережі з 4-а нейроелементами

0,3508	-0,0559	0,5704	0,7406
0,3036	0,9197	-0,2277	0,1008
0,4657	0,063	0,5815	-0,6641
0,7537	-0,3836	-0,5331	0,0253

Для нейроподібної мережі з 2-а нейроелементами матриця вагових коефіцієнтів має вид:

$$\begin{vmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{vmatrix} \quad (3.9)$$

Обчислені вагові коефіцієнти для нейроподібної мережі з 2-а нейроелементами наведені в табл.3.3.

Таблиця 3.3.

Вагові коефіцієнти для нейроподібної мережі з 2-а нейроелементами

0,5934	0,8048
0,8049	-0,5934

3.2.3. Обчислення таблиці макрочасткових добутків для шифрування даних

Для шифрування даних вагові коефіцієнти є попередньо обчисленими (константами) і є у форматі з плаваючою комою. Вхідні дані Xj поступають у форматі з фіксованою комою (фіксацією її перед старшим розрядом числа).

Обчислення скалярного добутку з використанням таблично-алгоритмічного методу робиться так:

$$Z = \sum_{j=1}^{N} W_j X_j = \sum_{i=1}^{n} 2^{-i} \sum_{j=1}^{N} W_j X_{ji} = \sum_{i=1}^{n} 2^{-i} \sum_{j=1}^{N} P_{ji} = \sum_{i=1}^{n} 2^{-i} P_{Mi},, \quad (3.10)$$

де N – кількість добутків, X_j – j-і вхідні дані, W_j – j-й ваговий коефіцієнт, n – розрядність вхідних даних, P_{ji} – ji-й частковий добуток, P_{Mi} – і-й макрочастковий добуток, який формується додаванням N добутків P_{ji}.

Формування таблиць макрочасткових добутків для вагових коефіцієнтів з плаваючою комою $W_j = w_j 2^{m_{Wj}}$, де w_j – мантиса W_j вагового коефіцієнта, m_{W_j} - порядок W_j вагового коефіцієнта. Для цього треба зробити такі операції:

- визначення найбільшого спільного порядку вагових коефіцієнтів m_{Wmaxc};
- обчислення різниці порядків для кожного W_j вагового коефіцієнта $\Delta m_{W_j} = m_{Wmaxc} - m_{W_j}$;
- зсунення вправо мантиси w_j на різницю порядків Δm_{W_j};
- обчислення макрочасткового добутку P_{Mi} для випадку, коли $x_{1i}=x_{2i}=\ldots=x_{Ni}=1$;
- визначення кількості розрядів переповнення q в P_{Mi}, коли $x_{1i}=x_{2i}=\ldots=x_{Ni}=1$;
- отримання масштабованих мантис w_j^h зсунувши їх вправо на кількість розрядів переповнення;
- додавання до найбільшого спільного порядку m_{Wmaxc} кількості розрядів переповнення $m_j = m_{Wmaxc} + q$.

Таблиця макрочасткових добутків обчислюється за такою формулою (2.69).

Кількість можливих варіантів макрочасткових добутків і відповідно обсяг таблиці визначаються так:

$$Q = 2^N. \quad (3.11)$$

Обсяг пам’яті можна зменшити за рахунок поділу всіх добутків N на частини N_1 та N_2. Для кожної із цих частин формуються окремі таблиці макрочасткових добутків P_{N1Mi} та P_{N2Mi}. Ці таблиці можуть зберігатися як в

окремих блоках пам’яті так і в одному. При використанні двох блоків пам’яті частини макрочасткових добутків зчитуються за один такт, а при використанні одного блоку – за два такти. Макрочастковий добуток P_{Mi} є сумою двох частин макрочасткових добутків P_{N1Mi} та P_{N2Mi}.

3.2.4 Нейроподібне шифрування даних

Навчання нейронної мережі є у визначені матриці вагових коефіцієнтів W. Ця матриця утворюється з власних векторів автоковаріаційної матриці вхідних даних R. Нейроподібна мережа, яка використовується для шифрування, продемонстрована на рис. 3.2, де XOR – операція маскування з використанням елементів Виключне АБО.

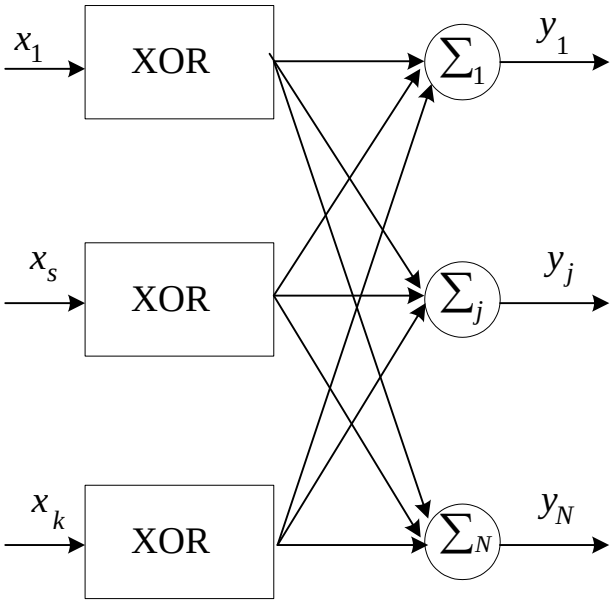


Рис. 3.2. Структура нейроподібної мережі для шифрування

Основна операція нейромережевого шифрування даних зводиться до множення матриці вагових коефіцієнтів на вектор вхідних даних:

$$y_j = \begin{vmatrix} W_{11} & W_{12} & \cdots & W_{1k} \\ W_{21} & W_{22} & \cdots & W_{2k} \\ \vdots & \vdots & \cdots & \vdots \\ W_{N1} & W_{N2} & \cdots & W_{Nk} \end{vmatrix} \times \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{matrix} \qquad (3.12)$$

Це множення зводиться до виконання N операцій обчислення скалярного добутку:

$$y_j = \sum_{s=1}^{k} W_{js} x_s \,, \tag{3.13}$$

де k – кількість добутків, s=1, ..., k, j =1, …, N.

Обчислення скалярних добутків будемо виконувати з використанням таблично-алгоритмічного методу, де вагові коефіцієнти W_{js} задаються у форматі з плаваючою комою, а вхідні дані x_s у форматі з фіксованою комою (фіксацією коми перед старшим розрядом). Таблично-алгоритмічне обчислення мантиси скалярного добутку зводиться до зчитування макрочасткового добутку P_{Mi} з j-ї таблиці за адресом, який відповадає і-му розрядному зрізу N вхідних даних та його додавання до раніше накопичених сум:

$$y_{мji} = 2^{-1} y_{мj(i-1)} + P_{Mji} \,, \tag{3.14}$$

де y_{j0}=0, i=1, ..., m, m – розрядність вхідних даних.

Кількість таблиць макрочасткових добутків відповідає кількості стрічок матриці, тобто N. Результат обчислення y_j скалярного добутку складається з мантиси y_{Mj} та порядку mj.

Час обчислення мантиси скалярного добутку розраховується так:

$$t_{СД} = m(t_{табл} + t_{Рг} + t_{См}) \tag{3.15}$$

де $t_{СД}$ – час обчислення скалярного добутку, $t_{табл}$ – час читання з таблиці (пам’яті), $t_{Рг}$ – час читання з регістра, $t_{См}$ – час додавання.

У залежності від необхідної швидкодії, шифрування може виконуватися як послідовно так і паралельно. У випадку послідовного шифрування час необхідний на шифрування визначається за формулою:

$$t_{Шифр} = Nm(t_{табл} + t_{Рг} + t_{См}) \,, \tag{3.16}$$

де $t_{Шифр}$ – час шифрування. Зменшити час шифрування можна за рахунок паралельного виконання N операцій з обчислення скалярного добутку.

На виході нейронної мережі отримуємо N зашифрованих даних у такому вигляді $y_j=y_{Mj}2^{mj}$. Для передачі зашифрованих даних на дешифрування всі зашифровані дані доцільно звести до найбільшого спільного порядку. Зведення до найбільшого спільного порядку робиться в три етапи:

– визначаємо найбільший порядок $m_ш$

– обчислюємо для кожного зашифрованого даного y_j різниці порядків $\Delta m_j = m_u - m_j$;

– виконуємо зсунення вправо мантиси y_{Mj} на різницю порядків Δm_j та отримуємо мантису зашифрованого даного y^h_{Mj} приведену до найбільшого спільного порядку.

На дешифрування передається мантиси зашифрованих даних y^h_{Mj}, які приведені до найбільшого спільного порядку та найбільший спільний порядок m_u.

Для нейроподібної мережі з 8-а нейроелементами шифрування команд управління рухом МРС передбачає множення матриці розміром 8×8 порахованих вагових коефіцієнтів W на вектор з 8-и вхідних даних x_k розрядністю 2-а у відповідності з наступною формулою

$$y_j = \begin{vmatrix} W_{11} & W_{12} & \cdots & W_{18} \\ W_{21} & W_{22} & \cdots & W_{28} \\ \vdots & \vdots & \cdots & \vdots \\ W_{81} & W_{82} & \cdots & W_{88} \end{vmatrix} \times \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{matrix} \qquad (3.17)$$

Приклад шифрування команди управління «Віддаль, *km*» з використанням мережі з 8-и нейроподібних елементів наведений на рис.3.3.

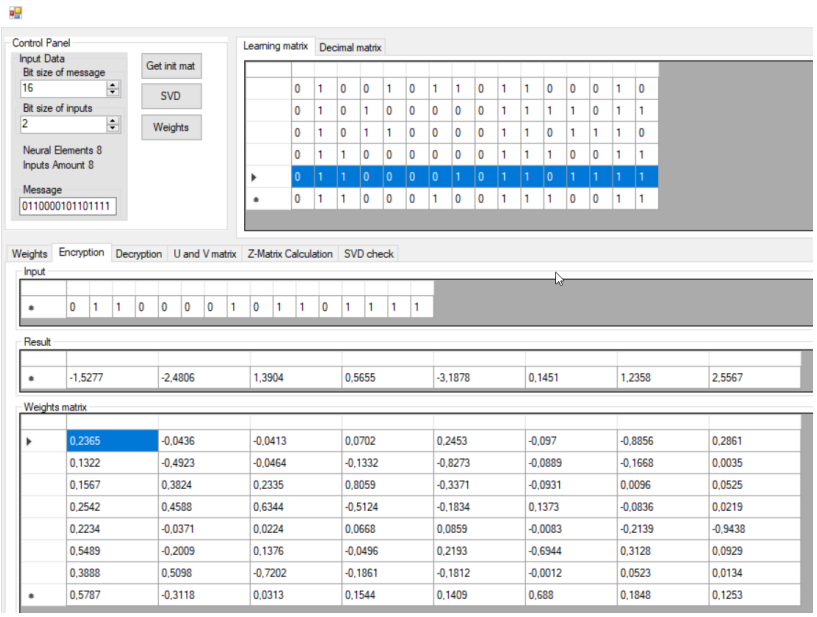


Рис. 3.3. Шифрування команди управління «Віддаль, *km*» з використанням мережі з 8-и нейроподібних елементів

Для нейроподібної мережі з 4-а нейроелементами шифрування даних передбачає множення матриці розміром 4×4 порахованих вагових коефіцієнтів W на вектор з 4-и вхідних даних x_k розрядністю 4-и:

$$y_j = \begin{vmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \\ W_{41} & W_{42} & W_{43} & W_{44} \end{vmatrix} \times \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} \tag{3.18}$$

Приклад шифрування команди управління «Віддаль, *km*» з використанням мережі з 4-х нейроподібних елементів наведений на рис.3.4.

Для нейроподібної мережі з 2-а нейроелементами шифрування даних передбачає множення матриці розміром 2×2 порахованих вагових коефіцієнтів W на вектор з 2-и вхідних даних x_k розрядністю 8-м у відповідності з наступною формулою:

$$y_j = \begin{vmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{vmatrix} \times \begin{matrix} x_1 \\ x_2 \end{matrix} \tag{3.19}$$

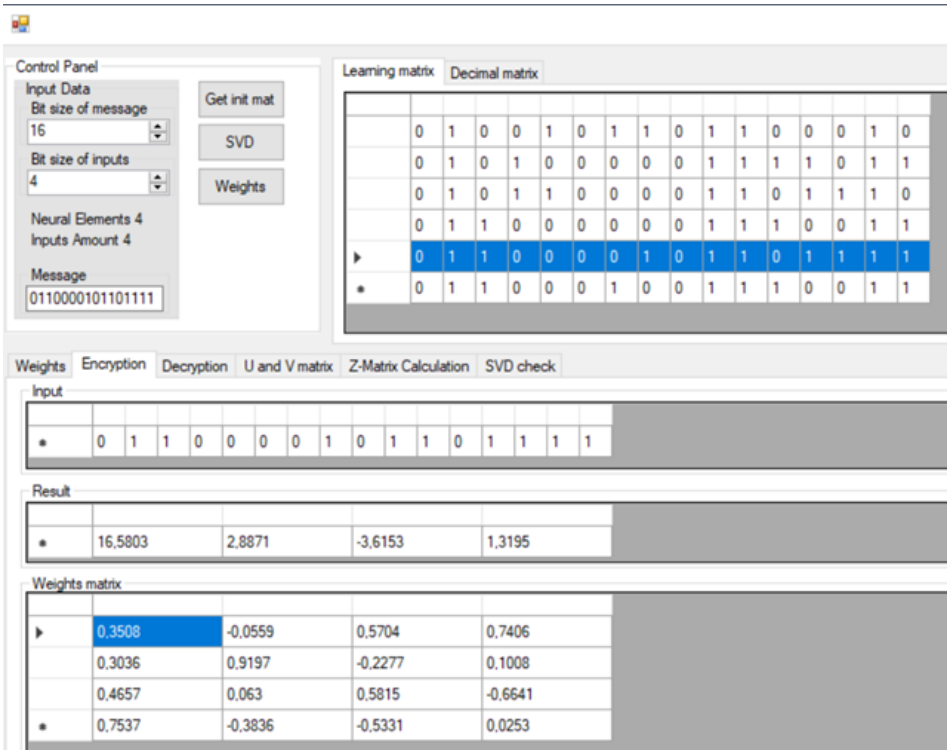


Рис. 3.4. Шифрування команди управління «Віддаль, *km*» з використанням мережі 4 нейроподібних елементів

Приклад шифрування команди управління «Віддаль, *km*» з використанням мережі з 2-х нейроподібних елементів наведений на рис.3.5.

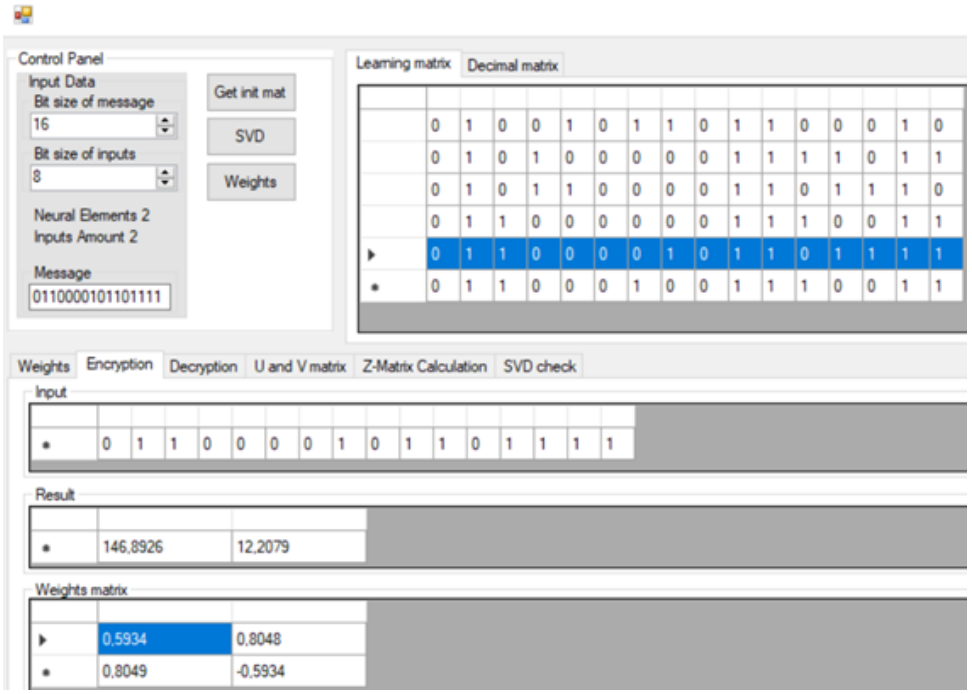


Рис.3.5. Шифрування команди управління «Віддаль, *km*» з використанням мережі з 2-х нейроподібних елементів

3.3. Основні етапи нейроподібного криптографічного дешифрування даних

Зашифровані дані у вигляді мантис $y_{мj}^{h}$, які приведені до найбільшого спільного порядку і найбільшого спільного порядку $m_{ш}$ надходять на дешифрування. Розглянемо основні етапи дешифрування даних.

3.3.1. Вибір архітектури нейроподібної мережі для дешифрування зашифрованих даних

Архітектура нейромережі для дешифрування зашифрованих даних за кількістю нейроелементів відповідає нейромережі, яка використовується для шифрування даних. У даній нейромережі кількість входів і кількість нейронів відповідають кількості зашифрованих мантис $y_{мj}^{h}$. Архітектура нейронної мережі, що використовуєть для дешифрування даних показана на рис. 3.6.

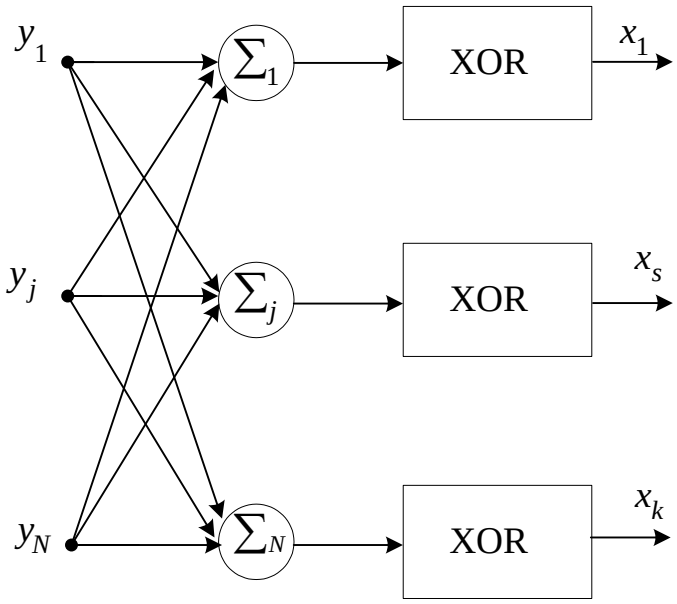


Рис.3.6. Архітектура нейроподібної мережі для дешифрування зашифрованих даних

У нейронній мережі для дешифрування даних розрядність входів відповідає розрядності зашифрованих мантис $y^h_{мj}$, яка визначає час дешифрування. Для зменшення даного часу можна зменшувати кількість розрядів, які не впливають на відновлення початкового повідомлення.

3.3.2 Формування матриці вагових коефіцієнтів

Матриця вагових коефіцієнтів для дешифрування даних обчислюється на основі матриці вагових коефіцієнтів для шифрування шляхом транспонування:

$$\begin{vmatrix} W_{11} & W_{12} & \cdots & W_{1k} \\ W_{21} & W_{22} & \cdots & W_{2k} \\ \vdots & \vdots & \cdots & \vdots \\ W_{N1} & W_{N2} & \cdots & W_{Nk} \end{vmatrix}^T = \begin{vmatrix} W_{11} & W_{21} & \cdots & W_{N1} \\ W_{12} & W_{22} & \cdots & W_{N2} \\ \vdots & \vdots & \cdots & \vdots \\ W_{1k} & W_{2k} & \cdots & W_{Nk} \end{vmatrix} \quad (3.20)$$

Як для шифрування вхідних даних, так для дешифрування зашифрованих даних базовою операцією є обчислення скалярного добутку, яка реалізовується з використанням таблично-алгоритмічного методу.

3.3.3. Обчислення таблиці макрочасткових добутків для дешифрування зашифрованих даних

Особливістю операції обчислення скалярного добутку при дешифруванні даних є те, що вагові коефіцієнти є попередньо обчислені і задаються у форматі із плаваючою комою. Зашифровані дані y_i поступають у форматі з блочно-плаваючою комою. Для обчислення використовується формула (3.10). Підготовка та обчислення можливих варіантів макрочасткових добутків виконується за формулою (3.11). Кількість можливих варіантів макрочасткових добутків P_{Mi} і відповідно обсяг таблиці залежить від кількості зашифрованих даних. Для кожної таблиці макрочасткових добутків обчислюється свій найбільший спільний порядок m_{Pms}.

3.3.4. Нейроподібне дешифрування зашифрованих даних

Основна операція нейромережевого дешифрування зашифрованих даних зводиться до множення матриці вагових коефіцієнтів, де кожен елемент множиться на вектор зашифрованих даних за формулою:

$$x_s = \begin{vmatrix} W_{11} & W_{21} & \cdots & W_{N1} \\ W_{12} & W_{22} & \cdots & W_{N2} \\ \vdots & \vdots & \cdots & \vdots \\ W_{1k} & W_{2k} & \cdots & W_{Nk} \end{vmatrix} \times \begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{matrix} \qquad (3.21)$$

Множення транспонованої матриці вагових коефіцієнтів на вектор вхідних даних зводиться до обчислення скалярного добутку N разів:

$$x_s = \sum_{j=1}^{N} W_{sj} y_j \,, \qquad (3.22)$$

де N – кількість добутків, s=1, …, k, j=1, …, N.

Таблично-алгоритмічне обчислення мантиси скалярного добутку зводиться до зчитування макрочасткового добутку P_{Mi} з таблиці за адресом, що відповідає і-му розрядному зрізу k вхідних даних та його додавання до раніше накопичених сум за формулою:

$$x_{msi} = 2^{-1} y_{ms(i-1)} + P_{Msi} \,, \qquad (3.23)$$

де xs0=0, i=1, …, g, g – розрядність мантиси зашифрованих даних.

Час для обчислення мантиси скалярного добутку розраховується так:

$$t_{СД} = g(t_{табл} + t_{Рг} + t_{См}) \quad (3.24)$$

де $t_{СД}$ – час обчислення скалярного добутку, $t_{табл}$ – час читання з таблиці, $t_{Рг}$ – час читання з регістра, $t_{См}$ – час додавання. Результат обчислення x_s скалярного добутку складається з мантиси x_{MS} та порядку $m_{Дш s}=m_{PMs}+m_{ш}$.

На виході нейронної мережі отримуємо k дешифрованих даних у наступному вигляді $x_s=x_{Ms}2^{mДш s}$. Для отримання вхідних даних необхідно s-у мантису x_{Ms} зсунути на значення $m_{Дш s}$.

Архітектура нейроподібної мережі для дешифрування зашифрованих даних за кількістю нейроелементів рівна нейроподібній мережі для шифрування даних. У цій нейроподібній мережі кількість входів та кількість нейроподібних елементів рівна кількості зашифрованих даних y_j. У нейроподібній мережі для дешифрування зашифрованих даних розрядність входів y_j відповідає розрядності зашифрованих мантис, що визначає час на дешифрування. Для зменшення часу дешифрування даних можна відкинути молодші розряди мантиси. Але тільки ті, які не впливають на відновлення початкового вхідного повідомлення.

Для нейроподібної мережі з 8 нейроелементами матриця вагових коефіцієнтів формується так:

$$\begin{vmatrix} W_{11} & W_{12} & \cdots & W_{18} \\ W_{21} & W_{22} & \cdots & W_{28} \\ \vdots & \vdots & \cdots & \vdots \\ W_{81} & W_{82} & \cdots & W_{88} \end{vmatrix}^{T} = \begin{vmatrix} W_{11} & W_{12} & \cdots & W_{81} \\ W_{12} & W_{22} & \cdots & W_{82} \\ \vdots & \vdots & \cdots & \vdots \\ W_{18} & W_{28} & \cdots & W_{88} \end{vmatrix} \quad (3.25)$$

Обчислені вагові коефіцієнти для дешифрування команд управління рухом МРС нейроподібною мережею з 8-а нейроелементами наведені в табл.3.4.

Таблиця 3.4.

Вагові коефіцієнти дешифрування команд управління рухом МРС нейроподібною мережею з 8-а нейроелементами

0,2365	0,1322	0,1567	0,2542	0,2234	0,5489	0,3888	0,5787
-0,0436	-0,4923	0,3824	0,4588	-0,0371	-0,2009	0,5098	-0,3118
-0,0413	-0,0464	0,2335	0,6344	0,0224	0,1376	-0,7202	0,0313

0,0702	-0,1332	0,8059	-0,5124	0,0668	-0,0496	-0,1861	0,1544
0,2453	-0,8273	-0,3371	-0,1834	0,0859	0,2193	-0,1812	0,1409
-0,097	-0,0889	-0,0931	0,1373	-0,0083	-0,6944	-0,0012	0,688
-0,8856	-0,1668	0,0096	-0,0836	-0,2139	0,3128	0,0523	0,1848
0,2861	0,0035	0,0525	0,0219	-0,9438	0,0929	0,0134	0,1253

Матриця вагових коефіцієнтів для дешифрування даних з використанням нейроподібної мережі з 4-а нейроелементами формується так:

$$\begin{vmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & \vdots w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43}\cdots & w_{44} \end{vmatrix}^{T} = \begin{vmatrix} w_{11} & w_{21} & w_{31} & w_{41} \\ w_{12} & w_{22} & w_{32} & w_{42} \\ w_{13} & \vdots w_{23} & w_{33} & w_{43} \\ w_{14} & w_{24} & w_{34}\cdots & w_{44} \end{vmatrix} \quad (3.26)$$

Обчислені вагові коефіцієнти для дешифрування команд управління рухом МРС нейроподібною мережею з 4-а нейроелементами наведені в табл.3.5.

Таблиця 3.5.

Вагові коефіцієнти для дешифрування команд управління рухом МРС нейроподібною мережею з 4-а нейроелементами

0,3508	0,3036	0,4657	0,7537
-0,0559	0,9197	0,063	-0,3836
0,5704	-0,2277	0,5815	-0,5331
0,7406	0,1008	-0,6641	0,0253

Матриця вагових коефіцієнтів для дешифрування даних з використанням нейроподібної мережі з 2-а нейроелементами формується так:

$$\begin{vmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{vmatrix}^{T} = \begin{vmatrix} W_{11} & W_{21} \\ W_{12} & W_{22} \end{vmatrix} \quad (3.27)$$

Обчислені вагові коефіцієнти для дешифрування команд управління рухом МРС нейроподібною мережею з 2-а нейроелементами наведені в табл.3.6.

Таблиця 3.6.

Вагові коефіцієнти для дешифрування команд управління рухом МРС нейроподібною мережею з 2-а нейроелементами

0,5934	0,8049
0,8048	-0,5934

Основна операція нейроподібного дешифрування зашифрованих даних зводиться до множення обчислених матриць вагових коефіцієнтів W^T на вектор зашифрованих даних $\bar{y}$ за наступною формулою:

$$x_s = \begin{vmatrix} W_{11} & W_{21} & \cdots & W_{N1} \\ W_{12} & W_{22} & \cdots & W_{N2} \\ \vdots & \vdots & \cdots & \vdots \\ W_{1k} & W_{2k} & \cdots & W_{Nk} \end{vmatrix} \times \begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{matrix} \tag{3.28}$$

Для нейроподібної мережі з 8-а нейроелементами дешифрування даних передбачає множення транспонованої матриці розміром 8×8 порахованих вагових коефіцієнтів на вектор $\bar{y}$ з 8-и зашифрованих даних за наступною формулою:

$$x_s = \begin{vmatrix} W_{11} & W_{12} & \cdots & W_{18} \\ W_{21} & W_{22} & \cdots & W_{28} \\ \vdots & \vdots & \cdots & \vdots \\ W_{81} & W_{82} & \cdots & W_{88} \end{vmatrix} \times \begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_8 \end{matrix} = \tag{3.29}$$

Приклад дешифрування команди управління «Віддаль, *km*» з використанням мережі з 8-х нейроподібних елементів наведений на рис.3.7.

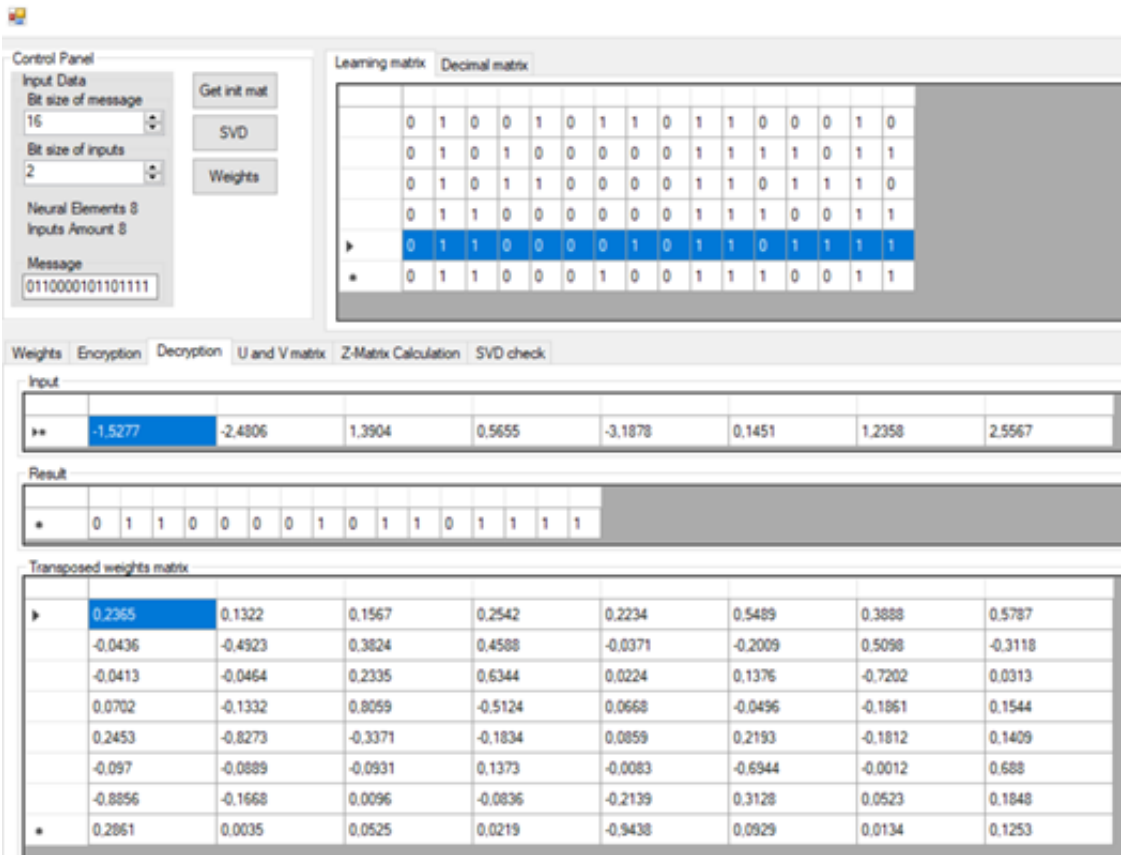


Рис.3.7. Дешифрування команди управління «Віддаль, *km*» з використанням мережі з 8-и нейроподібних елементів

Для нейроподібної мережі з 4-а нейроелементами дешифрування даних передбачає множення транспонованої матриці розміром 4×4 вагових коефіцієнтів на вектор $\bar{y}$ з 4-и зашифрованих даних у відповідності з наступною формулою

$$x_s = \begin{vmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \\ W_{41} & W_{42} & W_{43} & W_{44} \end{vmatrix} \times \begin{matrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{matrix} \qquad (3.30)$$

Приклад дешифрування команди управління «Віддаль, *km*» з використанням мережі з 4-х нейроподібних елементів наведений на рис.3.8.

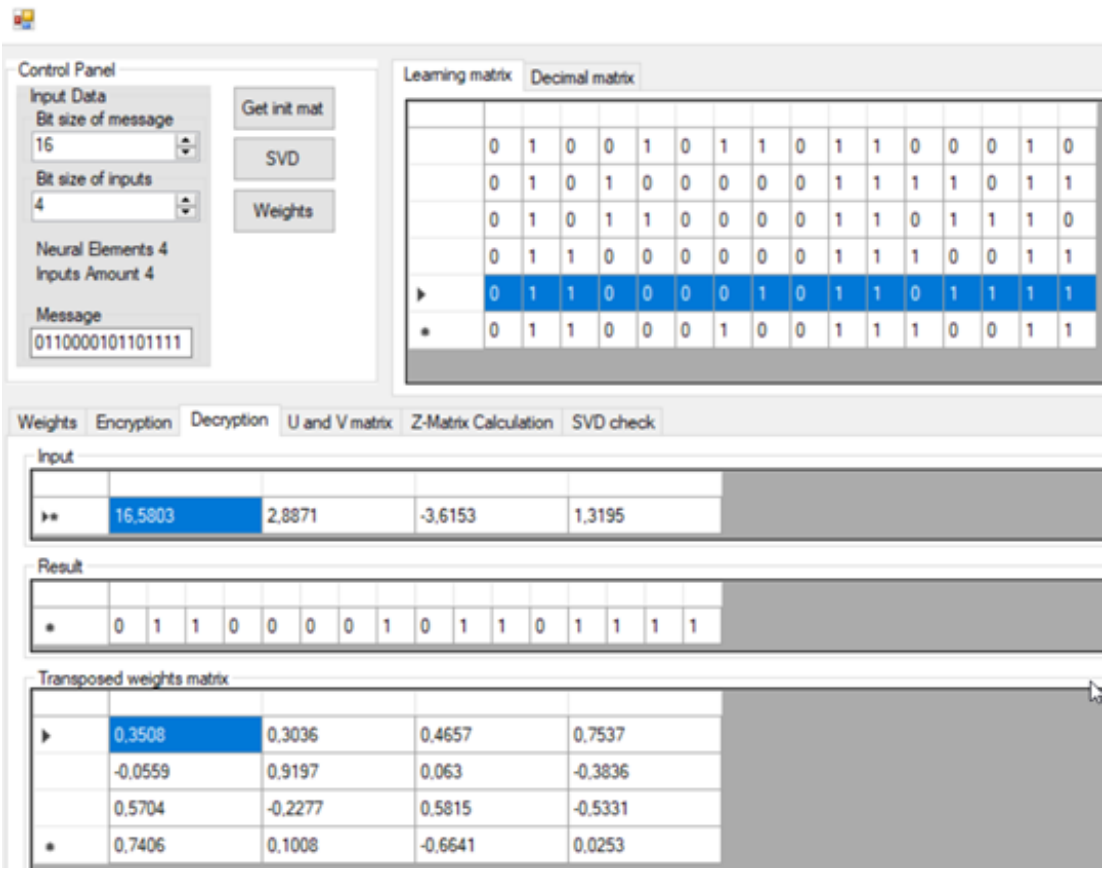


Рис.3.8. Дешифрування команди управління «Віддаль, *km*» з використанням мережі з 4-х нейроподібних елементів

Для нейроподібної мережі з 2-а нейроелементами дешифрування даних передбачає множення транспонованої матриці розміром 2×2 вагових коефіцієнтів на вектор $\bar{y}$ з 2-и зашифрованих даних у відповідності з наступною формулою

$$y_j = \begin{vmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{vmatrix} \times \begin{matrix} x_1 \\ x_2 \end{matrix} \qquad (3.31)$$

Приклад дешифрування команди управління «Віддаль, *km*» з використанням мережі з 4-х нейроподібних елементів наведений на рис.3.9.

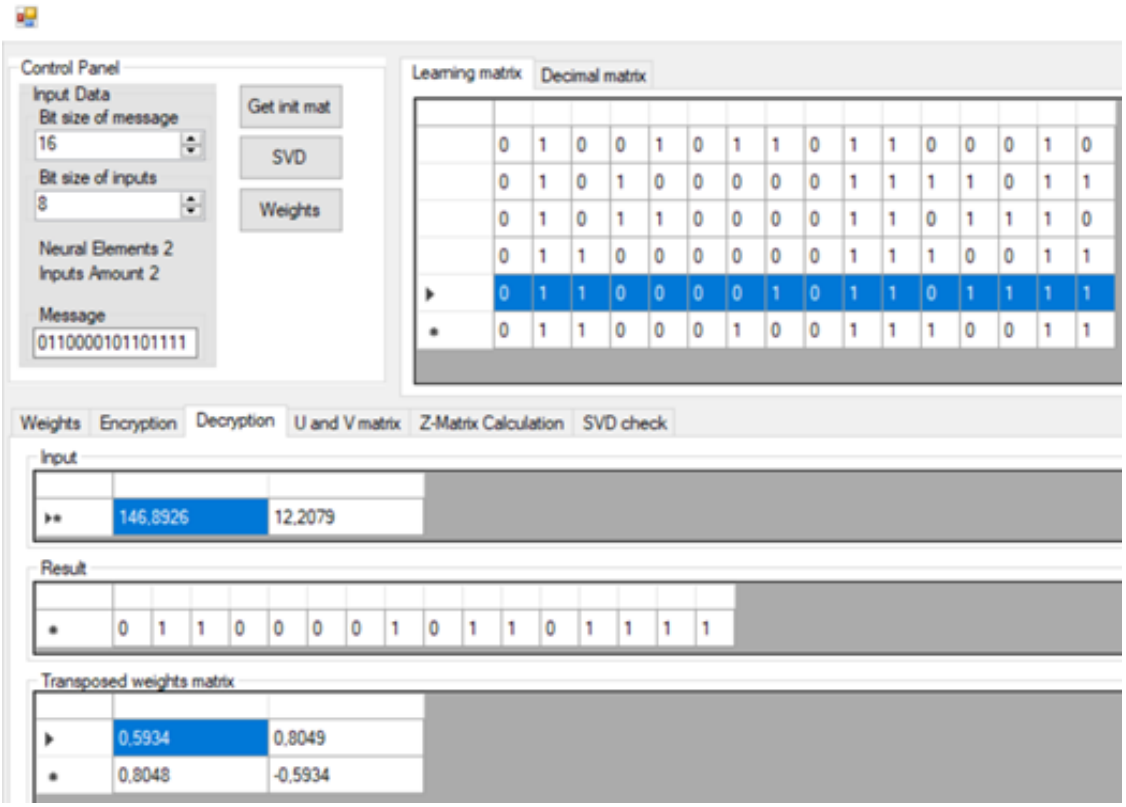


Рис.3.9. Дешифрування команди управління «Віддаль, *km*» з використанням мережі з 4-х нейроподібних елементів

3.4. Синтез мобільних засобів нейроподібного криптографічного шифрування та дешифрування даних

При розробці бортових засобів нейромережевого криптографічного захисту даних виникає проблема забезпечення режиму реального часу, підвищення криптостійкості, завадостійкості та зменшення габаритів, маси, енергоспоживання та вартості. Одним із шляхів забезпечення високих техніко-економічних характеристик таких засобів це використання для криптографічного захисту саме автоасоціативної нейроподібної мережі прямого поширення, що навчається на основі методу моделі послідовних геометричних перетворень. Базовою операцією нейроподібної мережі є обчислення скалярного добутку

$$Y_j = X_1 W_{1,} + ... + X_i W_j + ... + X_N W_N = \sum_{j=1}^{N} X_j W_j \,. \qquad (3.32)$$

де X_j – вхідні дані, W_j – вагові коефіцієнти.

Структура автоасоціативної нейронної мережі, яка використовується для нейроподібного шифрування даних, наведено на рис.3.10., де X_j – j-і вхідні дані, Y_j – j-і зашифровані дані.

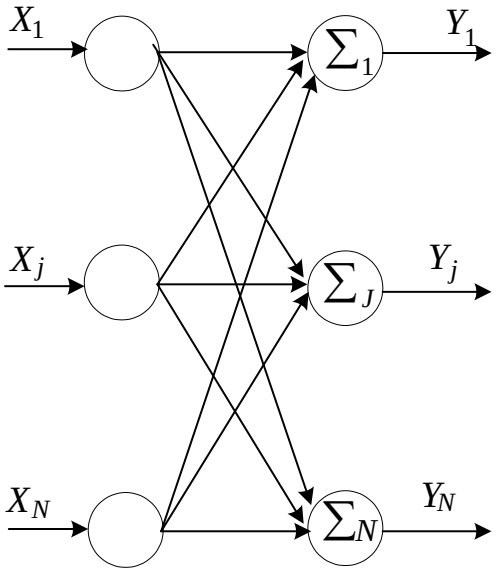


Рис. 3.10. Структура автоасоціативної нейронної мережі для нейроподібного шифрування даних

Особливістю таких нейроподібних структур є можливість наперед обчислити вагові коефіцієнти, що дає змогу використовувати табличко-алгоритмічний алгоритм для реалізації нейроподібних елементів. Розробка засобів нейроподібного криптографічного шифрування та дешифрування даних з високими техніко-економічними показниками вимагає широкого використання сучасної елементної бази (мікрокомп’ютерів, мікроконтролерів, програмованих логічних інтегральних схем (ПЛІС) типу FPGA [58, 75]), розроблення нових методів, алгоритмів і структур, орієнтованих на апаратно-програмну реалізацію [74, 78]. Перспективним шляхом реалізації засобів нейроподібного криптографічного шифрування (дешифрування) є використання універсального процесорного ядра доповненого спеціалізованими апаратно-програними засобами. Процес взаємопроникнення універсального та спеціалізованого, програмного і апаратного забезпечує ефективну реалізацію алгоритмів нейромережевого криптографічного шифрування та дешифрування даних. Для реалізації спеціалізованих апаратних засобів криптографічного шифрування та

дешифрування даних пропонується використовувати ПЛІС типу FPGA [77, 101], забезпечує можливість зміни структури та нарощування функцій.

Вихідною інформацією для розроблення бортових засобів нейроподібного криптографічного шифрування/дешифрування [117]:

– апаратно-орієнтовані методи реалізації базових операцій лінійного та нелінійного шифрування/дешифрування даних;

– графове відображення алгоритмів реалізації базових операцій лінійного та нелінійного шифрування/дешифрування даних;

– структури (рекурсивна, нерекурсивна) для реалізації базових операцій лінійного та нелінійного шифрування/дешифрування даних;

– кількість вхідних даних N та формат їх подання – фіксована або плаваюча кома;

– використання вагових коефіцієнтів у форматі із плаваючою комою;

– точність обчислення і кількість нейронів;

– інтенсивність з якою надходять вхідні дані;

– вимоги до інтерфейсу;

– техніко-економічні вимоги і обмеження.

Загалом, задачі синтезу мобільних засобів нейроподібного криптографічного шифрування (дешифрування) можна сформулювати наступним чином:

– вибрати алгоритм нейроподібного шифрування та дешифрування даних і продемонструвати його у вигляді потокового конкретизованого графу;

– розробити структуру блоків нейроподібного шифрування (дешифрування) з максимальною ефективністю використання обладнання, що враховує усі обмеження та забезпечує обробку даних у реальному часі;

– визначити основні характеристики нейроподібних елементів та синтезувати їх;

– обрати способи обміну, визначити потрібні зв’язки та розробити систему для обміну між компонентами блоків нейроподібного шифрування (дешифрування);

– установити порядок реалізації у часі для алгоритму нейроподібного шифрування/дешифрування даних, а також розробити алгоритм управління цим процесом.

3.5. Розроблення методу вибору елементної бази з врахуванням вимог конкретних застосувань для синтезу засобів захисту даних у реальному часі

3.5.1. Формування критеріїв вибору елементної бази

Сучасний етап розвитку систем захисту та передачі даних у реальному часі із допомогою шумоподібних кодів описується розширенням галузей застосування, у більшості з яких вимагає криптографічного нейромережевого шифрування (дешифрування) та кодування (декодування) даних у реальному часі з використанням баркероподібних кодів на апаратних засобах, які задовольняють обмеження щодо вартості та часу розробки, габаритів і енергоспоживання. Створення програмно-апаратних засобів вимагає широкого використання сучасної елементної бази, а саме: мікроконтролерів, мікропроцесорів загального призначення, процесорів цифрової обробки сигналів, нейропроцесорів, систем на кристалі, програмованих логічних інтегральних схеми (ПЛІС), замовних надвеликих інтегральних схем (НВІС). Ефективне використання сучасної елементної вимагає розроблення нових методів, алгоритмів криптографічного нейромережевого шифрування (дешифрування) та кодування (декодування) даних з використанням баркероподібних кодів і НВІС-структур для їх реалізації. При розробці блоків криптографічного нейромережевого шифрування та дешифрування даних, кодування та декодування зашифрованих даних необхідно забезпечити режим реального часу [130] та враховувати масштабованість, вартість розробки, сумісність з існуючими і майбутніми версіями.

Вибір елементної бази для синтезу СЗПД у реальному часі з використанням шумоподібних кодів здійснюється з врахуванням наступного:

– сукупність критеріїв повинна забезпечувати вибір елементної бази для синтезу СЗПД з високими техніко-економічними показниками;

– критерії вибору повинні мати однозначне поняття для елементної бази;

– кількість критеріїв вибору елементної бази повинна бути обмежена і орієнтована на елементної бази.

Сформуємо групу критеріїв вибору елементної бази, які достатньо повно її характеризують і орієнтовані на реалізацію СЗПД з високими техніко-економічними показниками. У дану групу доцільно включити такі критерії:

– вартість елементної бази, яка складає приблизно одну третю вартості СЗПД і суттєво впливає на її конкурентноздатність;

– продуктивність, яка визначає здатність реалізації нейромережевих криптографічних алгоритмів шифрування та дешифрування даних у реальному часі;

– потужність енергоспоживання, яка визначає вимоги до блоків живлення і характеризує температурний режим;

– коефіцієнт технологічності, який визначає вартість технологічного обладнання для розроблення і відлагодження апаратно-програм засобів;

– температурний діапазон роботи елементної бази, який впливає на умови експлуатації СЗПД;

– кількість мікросхем, що необхідні для синтезу СЗПД, які визначають надійність системи та вартість її виготовлення;

– габаритні показники елементної бази, які визначають габарити СЗПД, що є актуальним для бортових застосувань;

– критерії маси елементної бази, які визначають масу СЗПД і мають велике значення для бортової апаратури;

– критерій завадостійкості елементної бази, який визначається перепадом напруги між лог.0 і лог.1;

– показниками надійності елементної бази, а саме інтенсивністю відмов і ймовірністю безвідмовної роботи;

– коефіцієнт вібростійкості елементної бази, який характеризує стійкість до механічних дій.

3.5.2. Метод вибору елементної бази

Перспективною елементною базою для синтезу високоефективних СЗПД з використанням шумоподібних кодів у реальному часі є системи на кристалі SoC на основі вентильних матриць FPGA (Field Programmable Gate Arays) [85, 86] з високою інтеграцією вентилів на кристалі (більше 10 млн. вентилів), які включають складні функціональні блоки (СФ-блоки) та вбудовані процесорні ядра. В СЗПД СФ-блоки використовуються для реалізації нейроподібних елементів, які налаштовуються під кількість входів і їх розрядність. Використання SoC при проектуванні СЗПД забезпечить:

1. високу системну інтеграцію (процесорне ядро, СФ-блоки, засоби введення-виведення);
2. збільшену системну продуктивність (високошвидкісне процесорне ядро, нейроподібні елементи);
3. зменшення витрат на комплектуючі;
4. зменшення енергоспоживання (процесорне яро і програмована логіка мають незалежні ланки живлення, що забезпечує можливість відключення програмованої логіки);
5. зменшення часу розробки (великий вибір засобів розробки та відлагодження).

В процесі проектування СЗПД на SoC виконуються такі роботи:

– вибір або самостійна розробка СФ-блоків і процесорного ядра;

– розбиття СЗПД на апаратне (СФ-блоки, процесорне ядро) та програмне забезпечення;

– визначення алгоритмів роботи СЗПД, які реалізуються програмним шляхом і які реалізуються апаратно;

– розробити програмно-апаратні засоби верифікації, тестування та комплексного відлагодження програмного і апаратного забезпечення як на етапі шифрування та кодування даних, так і на етапі дешифрування та декодування даних;

– розробити інтерфейси взаємодії з зовнішнім середовищем, апаратними та програмними компонентами.

Для вибору елементної бази для синтезу СЗПД у реальному часі з використанням шумоподібних кодів розробимо метод. Цей метод ґрунтується на застосуванні теорії багатокритеіального аналізу та бере до уваги вимоги конкретного застосування (обсяг пам’яті, енергоспоживання, продуктивність, потужність, швидкість передачі даних, вартість, масу, габарити, температурний діапазон, надійність, стійкість до спеціальних факторів тощо). Основою цього методу є обчислення інтегрованої оцінки ефективності для СЗПД на основі часткових критеріїв ефективності. Ці критерії формуються для кожного конкретного застосування. Схема компромісів застосовується для обчислення інтегрованої оцінки ефективності. А саме, інтегрована оцінка ефективності j-ї елементної бази $E_{ІЕБj}$ рахується так:

$$E_{ІЕБj} = \sum_{i=1}^{n} \lambda_i E_{нКpi} \Rightarrow \max, \qquad (3.33)$$

де i=1,…,n – кількість включених в згортку часткових критеріїв ефективності апаратно-програмної компоненти; λ_i – i-й ваговий коефіцієнт; $E_{нКpi}$ – нормована оцінка ефективності i-о часткового критерію.

Метод вибору елементної бази для синтезу СЗПД вимагає виконання наступних етапів:

– сформувати перелік часткових критеріїв, які впливають на ефективність елементної бази;

– визначити шкалу зміни значень часткових критеріїв ефективності;

– визначити множину елементної бази, яка відповідає вимогам технічного завдання для компонентів СЗПД;

– розрахувати значення вагових коефіцієнтів, для визначення відносної важливості i-uj часткового критерію;

– розрахунок значень і-х часткових нормованих критеріїв ефективності;

– обчислити інтегровану оцінку ефективності для кожної елементної бази;

– порівняти та вибрати базу для синтезу СЗПД.

На першому етапі вибору елементної бази для синтезу СЗПД формується перелік часткових критеріїв ефективності. Перелік часткових критеріїв ефективності показано у табл. 3.7.

Таблиця 3.7

Перелік часткових критеріїв ефективності елементної бази для синтезу СЗПД

Назва критерію	**Позначення**
Продуктивність *j*-о процесорного ядра	$П_{ПЯj}$
Обсяг пам’яті *j*-ї SoC	Q_j
Розрядність *j*-о процесорного ядра	$D_{ПЯj}$
Тактова частота роботи *j*-о процесорного ядра	$F_{ПЯj}$
Потужність енергоспоживання *j*-о процесорного ядра	$P_{ПЯj}$
Тактова частота роботи *j*-о СФ-блока	$F_{СФj}$
Надійність *j*-х SoC	H_{SoCj}
Потужність енергоспоживання *j*-о СФ-блока	$P_{СФj}$
Зовнішній інтерфейс *j*-о SoC	R_{SoCj}
Площа кристала *j*-о SoC	Y_{SoCj}
Наявність інструментальних засобів розробки *j*-о SoC	B_{SoC}
Максимальна температура роботи *j*-о SoC	t_{max}
Мінімальна температура роботи *j*-о SoC	t_{min}
Маса СЗПД на основі *j*-о SoC	$M_{СЗПДj}$
Габарити СЗПД на основі *j*-о SoC	$S_{СЗПДj}$
Стійкість *j*-о SoC до спецфакторів (радіація)	γ_{SoCj}
Вартість *j*-о SoC	C_{Aj}
Вартість інструментальних засобів розробки та відлагодження *j*-о SoC	C_{SoCj}
Вартість *j*-о процесорного ядра	$C_{ПЯj}$
Вартість *j*-о СФ-блока	$C_{СФj}$

Витрати на експлуатацію СЗПД на основі *j*-о SoC	$C_{ЕСЗПДj}$
Маса *j*-о приймально-передавального пристрою	$M_{ППj}$
Габарити *j*-о приймально-передавального пристрою	$S_{ППj}$
Потужність енергоспоживання *j*-о приймально-передавального пристрою	$P_{ППj}$
Швидкість передачі даних *j*-о приймально-передавальним пристроєм	$V_{ППj}$
Відстань передачі даних *j*-о приймально-передавальним пристроєм	$L_{ППj}$
Вартість *j*-о приймально-передавального пристрою	$C_{ППj}$
Максимальна температура роботи *j*-о приймально-передавального пристрою	$t_{maxПП}$
Мінімальна температура роботи *j*-о приймально-передавального пристрою	$t_{minПП}$
Стійкість *j*-о приймально-передавального пристрою до спецфакторів (радіація)	$\gamma_{ППj}$

На другому етапі вибору елементної бази для синтезу СЗПД обирається шкала зміни числових значень часткових критеріїв ефективності для елементів апаратно-програмних модулів. Формування такої шкали здійснюється на основі завдання для розробки СЗПД.

На третьому етапі вибору елементної бази для синтезу СЗПД вибирається множина елементів, що відповідають вимогам технічного завдання синтезу СЗПД. Для вибору такої множини використовуються порогові коефіцієнти. Вибір можливої бази для синтезу СЗПД рахується таке:

$$W_{E_К} = \sum_{j=1}^{N} E_{Кj} n_j q_j p_j c_j m_j s_j f_j d_j r_j y_j b_j t_j h_j \gamma_j, \tag{3.34}$$

$$W_{E_{ПП}} = \sum_{j=1}^{N} E_{ППj} v_j l_j p_j c_j m_j s_j t_j h_j \gamma_j, \tag{3.35}$$

де W_{Ek}, W_{Enn} – множина відповідно комп'ютерних елементів та приймально-передавальних пристроїв; E_{kj} – j-й комп'ютерний елемент, $E_{ППj}$ - j-й приймально-передавальний пристрій; N – множина елементів; n_j, q_j, p_j, v_j,d_j, f_j,c_j, m_j, s_j, r_j, y_j, l_j, b_j, t_j, h_j, τ_j – це порогові коефіцієнти j-ї елементної бази за продуктивністю, обсягом пам'яті, потужністю енергоспоживання, швидкістю передачі даних, розрядністю процесорного ядра, тактовою частотою роботи, вартістю, масою, габаритами, інтерфейсом, площею кристала, віддаллю передачі, інструментальними засобами розробки, температурним діапазоном, надійністю, стійкістю до спецфакторів. Такі порогові коефіцієнти для вибору множини комп'ютерних елементів і множини приймально-передавальних пристроїв визначаються так:

$$n_i = \begin{cases} 0, & коли \quad П_i < П_з \\ 1, & коли \quad П_i \geq П_з \end{cases}, \tag{3.36}$$

$$q_i = \begin{cases} 0, & коли \quad Q_i < Q_з \\ 1, & коли \quad Q_i \geq Q_з \end{cases}, \tag{3.37}$$

$$p_i = \begin{cases} 0, & коли \quad P_i > P_з \\ 1, & коли \quad P_i \leq P_з \end{cases}, \tag{3.38}$$

$$v_i = \begin{cases} 0, & коли \quad V_i < V_з \\ 1, & коли \quad V_i \geq V_з \end{cases}, \tag{3.39}$$

$$d_i = \begin{cases} 0, & коли \quad D_i < D_з \\ 1, & коли \quad D_i \geq D_з \end{cases}, \tag{3.40}$$

$$f_i = \begin{cases} 0, & коли \quad F_i > F_з \\ 1, & коли \quad F_i \leq F_з \end{cases}, \tag{3.41}$$

$$c_i = \begin{cases} 0, & коли \quad C_i > C_з \\ 1, & коли \quad C_i \leq C_з \end{cases}, \tag{3.42}$$

$$m_i = \begin{cases} 0, & коли \quad M_i > M_з \\ 1, & коли \quad M_i \leq M_з \end{cases}, \tag{3.43}$$

$$s_i = \begin{cases} 0, & коли \quad S_i > S_з \\ 1, & коли \quad S_i \leq S_з \end{cases}, \tag{3.44}$$

$$r_i = \begin{cases} 0, & коли \quad R_i < R_з \\ 1, & коли \quad R_i \geq R_з \end{cases}, \tag{3.45}$$

$$y_i = \begin{cases} 0, & коли \quad Y_i > Y_з \\ 1, & коли \quad Y_i \le Y_з \end{cases}, \tag{3.46}$$

$$l_i = \begin{cases} 0, & коли \quad L_i < L_з \\ 1, & коли \quad L_i \ge L_з \end{cases}, \tag{3.47}$$

$$b_i = \begin{cases} 0, & коли \quad B_i < B_з \\ 1, & коли \quad B_i \ge B_з \end{cases}, \tag{3.48}$$

$$t_i = \begin{cases} 0, & коли \quad t_{\min i} < t_{\min з} \quad або \quad t_{\max i} < t_{\max з} \\ 1, & коли \quad t_{\min i} > t_{\min з} \quad і \quad t_{\max i} > t_{\max з} \end{cases}, \tag{3.49}$$

$$h_i = \begin{cases} 0, & коли \quad H_i < H_з \\ 1, & коли \quad H_i \ge H_з \end{cases}, \tag{3.50}$$

$$\gamma_i = \begin{cases} 0, & коли \quad \gamma_i < \gamma_з \\ 1, & коли \quad \gamma_i > \gamma \end{cases}, \tag{3.51}$$

де $П_з$, $Q_з$, $P_з$, $V_з$, $D_з$, $F_з$, $C_з$, $M_з$, $S_з$, $R_з$, $Y_з$, $L_з$, $B_з$, $t_з$, $H_з$, $\gamma_з$ – задані в технічному завдання параметри: продуктивності, обсягу пам'яті, потужності енергоспоживання, швидкості передачі даних, розрядность *j*-о процесорного ядра, тактова частота роботи, вартості, маси, габаритів, вимоги до інтерфейсу, площа кристала, віддаль передачі даних, повнота та наявність інструментальних засобів розробки, температурний діапазон роботи, надійність елементної бази, стійкість елементної бази до спецфакторів.

На четвертому етапі визначаємо значення вагових коефіцієнтів λ_i для часткових критеріїв ефективності елементів. Значення вагових коефіцієнтів визначається важливістю критерія для функціонування СЗПД. Під час визначення вагових коефіцієнтів необхідно врахувати таке – сума усіх коефіцієнтів повинна бути рівна $\sum_{i=1}^{n} \lambda_i = 1$. Визначення вагових коефіцієнтів здійснюємо шляхом експертного опитування. В процесі розроблення СЗПД досить часто використовують метод приписування балів або метод ранжування.

На п'ятому етапі вибору елементної бази для синтезу СЗПД робиться нормування часткових критеріїв ефективності. Нормування критеріїв: продуктивності ($E_{нПj}$), обсягу пам'яті ($E_{нQj}$), потужності енергоспоживання ($E_{нPj}$),

швидкості передачі даних ($E_{нVj}$), вартості ($E_{нCj}$), маси ($E_{нMj}$), габаритів ($E_{нSj}$) та надійсноті ($E_{нHj}$) рахуються так:

$$E_{нП_j} = \frac{П_j}{П_з}, \tag{3.52}$$

$$E_{нQ_j} = \frac{Q_j}{Q_з}, \tag{3.53}$$

$$E_{нP_j} = \frac{P_j}{P_з}, \tag{3.54}$$

$$E_{нV_j} = \frac{V_j}{V_з}, \tag{3.55}$$

$$E_{нC_j} = \frac{C_j}{C_з}, \tag{3.56}$$

$$E_{нM_j} = \frac{M_j}{M_з}, \tag{3.57}$$

$$E_{нS_j} = \frac{S_j}{S_з}, \tag{3.58}$$

$$E_{нH_j} = \frac{H_j}{H_з}, \tag{3.59}$$

Також можна використати й інші методи нормування часткових коефіцієнтів оптимальності.

Якщо відомо діапазони від f^{min} до f^{max}, то наступні вирази для нормування можна використати:

$$f_{H,i} = \frac{f_i^0}{f_i^{\max} - f_i^{\min}}, \text{ де } i = 1,2,...,m, \tag{3.60}$$

$$\text{чи } f_{H,i} = \frac{f_i^0 - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \qquad \text{де } i = 1,2,...,m. \tag{3.61}$$

На шостому етапі вибору елементної бази для синтезу СЗПД виконується обчислення інтегрованої оцінки ефективності j-ї бази, що використовується для синтезу комп'ютерних засобів [80-82], засобів зв'язку, а також давачів. Обчислення такої оцінки E_{IEk} та приймально-передавальних пристроїв $E_{IE_{ПП}}$ виконується за наступними формулами:

$$E_{IE_{Kj}} = \lambda_{П} E_{нПj} + \lambda_{Q} E_{нQj} + \lambda_{P} E_{нPj} + \lambda_{C} E_{нCj} + \lambda_{M} E_{нMj} + \lambda_{S} E_{нSj} + \lambda_{H} E_{нHj}, \quad (3.63)$$

$$E_{IE_{ППj}} = \lambda_{П} E_{нПj} + \lambda_{P} E_{нPj} + \lambda_{C} E_{нCj} + \lambda_{M} E_{нMj} + \lambda_{S} E_{нSj} + \lambda_{H} E_{нHj} + \lambda_{L} E_{нLj}. \quad (3.64)$$

На сьомому етапі вибору елементної бази визначається елементна база [92], яка буде використовуватися для синтезу СЗПД. Із множини бази, яка відповідає вимогам технічного завдання, треба обрати ту елементну базу, де інтегрована оцінка ефективності є найбільшою: E_{IEk} *max* та E_{IEnn} *max*

Запропонований метод дає змогу автоматизувати вибір оптимальної елементної бази для синтезу СЗПД опираючись на вимоги технічного завдання. Особливістю розробленого методу є врахування вимог для конкретного застосування. Воно має забезпечувати вибір найефективнішої елементної бази для синтезу СЗПД.

3.6. Висновки до розділу 3

1. Запропоновано розроблення інформаційної технології нейроподібного криптографічного захисту даних у реальному часі здійснювати на базі інтегрованого підходу, що включає: розроблення та дослідження теоретичних основ нейроподібного криптографічного захисту даних; дослідження та розроблення нових алгоритмів та структур нейроподібного шифрування та дешифрування даних, орієнтованих на сучасну елементну базу.

2. Розроблено інформаційну технологію нейроподібного криптографічного захисту даних у реальному часі із симетричними ключами (до який належать: коди маскування, архітектура нейроподібної мережі та матриці вагових коефіцієнтів), яка за рахунок попереднього обчислення матриць вагових коефіцієнтів і динамічної зміни архітектури нейроподібної мережі забезпечує високу криптографічну стійкість і апаратно-програмну реалізацію із високими техніко-економічними характеристиками.

3. Визначено, що архітектура нейроподібної мережі для криптографічного захисту даних у реальному часі визначається кількість нейроподібних елементів, входів і їх розрядністю.

4. Визначено, що основними критеріями вибору елементної бази для синтезу засобів криптографічного захисту даних у реальному часі є: вартість, продуктивність, обсяг пам’яті, потужність енергоспоживання, коефіцієнт технологічності, температурний діапазон роботи, маса, габаритні показники, показниками надійності, коефіцієнт вібростійкості.

5. Вдосконалено метод вибору елементної бази для синтезу засобів криптографічного захисту даних у реальному часі, який за рахунок обчислення інтегрованої оцінки ефективності елементної бази та врахування вимог конкретного застосування забезпечує вибір найефективнішої елементної бази із множини елементних баз, які відповідають вимогам технічного завдання.

РОЗДІЛ 4
РОЗРОБЛЕННЯ ЗАСОБІВ НЕЙРОПОДІБНОГО КРИПТОГРАФІЧНОГО ШИФРУВАННЯ ТА ДЕШИФРУВАННЯ ДАНИХ У РЕАЛЬНОМУ ЧАСІ

4.1. Розроблення імітаційної моделі вибору елементної бази для синтезу засобів криптографічного захисту даних у реальному часі

На основі вдосконаленого методу вибору елементної бази та синтезу СЗПД у реальному часі із використанням шумоподібних кодів розробляється імітаційна модель. Алгоритм роботи імітаційної моделі вибору елементної бази складається з таких кроків:

1. Запуск та ініціалізація імітаційної моделі.
2. Ініціалізація зв’язку із базою даних про структуру СЗПД, елементу базу, та вимоги технічного завдання.
3. Зчитування технічних характеристик елементної бази.
4. Зчитування даних про критерії, а також обмеження пошуку.
5. Зчитування інформації про структуру СЗПД.
6. Перебір та фільтрація елементної бази відповідно до мінімально та максимального значень критеріїв.
7. Нормалізація вагових коефіцієнтів для критеріїв.
8. Нормування часткових критеріїв для відфільтрованих елементів.
9. Обчислення інтегрованої оцінки ефективності для елементів.
10. Впорядкування елементів за спаданням значення інтегрованої ефективності.
11. Синтез альтернативи з’єднання підмножини елементів у модуль. Перевірка інтерфейсів компоненти та відбір альтернатив, що задовольняють сумісності інтерфейсів.
12. Розрахунок інтегрованої оцінки ефективності для кожного із синтезованих модулів.
13. Впорядкування елементів за спаданням значень інтегрованої ефективності.

14. Вивід результатів через інтерфейс для користувача.

Алгоритм роботи імітаційної моделі у формі блок-схеми наведено на рис. 4.1.

Рис. 4.1. Алгоритм синтезу компонент СЗПД

Цей алгоритм має лінійну структуру та складається з двох послідовних блоків. У процесі практичної реалізації блоки оформлені у вигляді підпрограм.

Перша підпрограма відповідає за завантаження варіантів елементної бази із бази даних, їхню попередню фільтрацію згідно із вимогами технічного завдання, обчислення інтегрованої оцінки ефективності для кожного типу елементної бази та сортування результатів по спаданню оцінки ефективності.

Друга підпрограма синтезу структури компонент СЗПД, що приймає на вхід варіанти компонент із першої підпрограми. Далі відбувається генерація варіантів об’єднання елементів у єдину компоненту. Ці варіанти фільтруються на основі сумісності за інтерфейсами. Ті варіанти, які пройшли перевірку – зберігаються та для них обчислюється загальна оцінка ефективності. Варіанти впорядковуються за спаданням загальної оцінки ефективності.

Розроблений алгоритм автоматизує процес синтезу СЗПД з врахуванням інтерфейсів компонент.

4.2. Засоби автоматизації вибору елементної бази

В загальному випадку, задача синтезу вимагає вибору базових елементів і аналізу великих обсягів даних [72]. Для зменшення часу, який необхідний для пошуку та автоматизації проектних рішень доцільно використовувати програмні засоби, які здатні пришвидшити вибір елементної бази та синтез нових компонентів і систем. Під час розроблення програми синтезу систем захисту та передачі даних з використанням шумоподібних кодів необхідно врахувати специфіку завдання та існуючі інструменти для дизайну та розробки.

Зокрема, розроблена програма синтезу складається з декількох модулів:

– модуль роботи з базою даних;
– модуль генерації бази даних;
– модуль роботи із елементами інтерфейсу користувача;
– модуль вибору елементної бази;
– модуль синтезу системи захисту та передачі даних;
– модуль відображення результатів.

У процесі розроблення системи синтезу проаналізовано існуючі інструменти для розробки програмних компонент. Однією із важливих вимог є можливість запускати програму на різних операційних системах без встановлення додаткових компонентів. Необхідно використовувати об’єктно орієнтований підхід при реалізації системи задля полегшення роботи та підтримки проекту. Також необхідно використовувати поширені бібліотеки для роботи з модулями системи. На рис. 4.2 наведений користувацький інтерфейс для засобів автоматизації синтезу СЗПД у реальному часі з використанням шумоподібних кодів [110, 111].

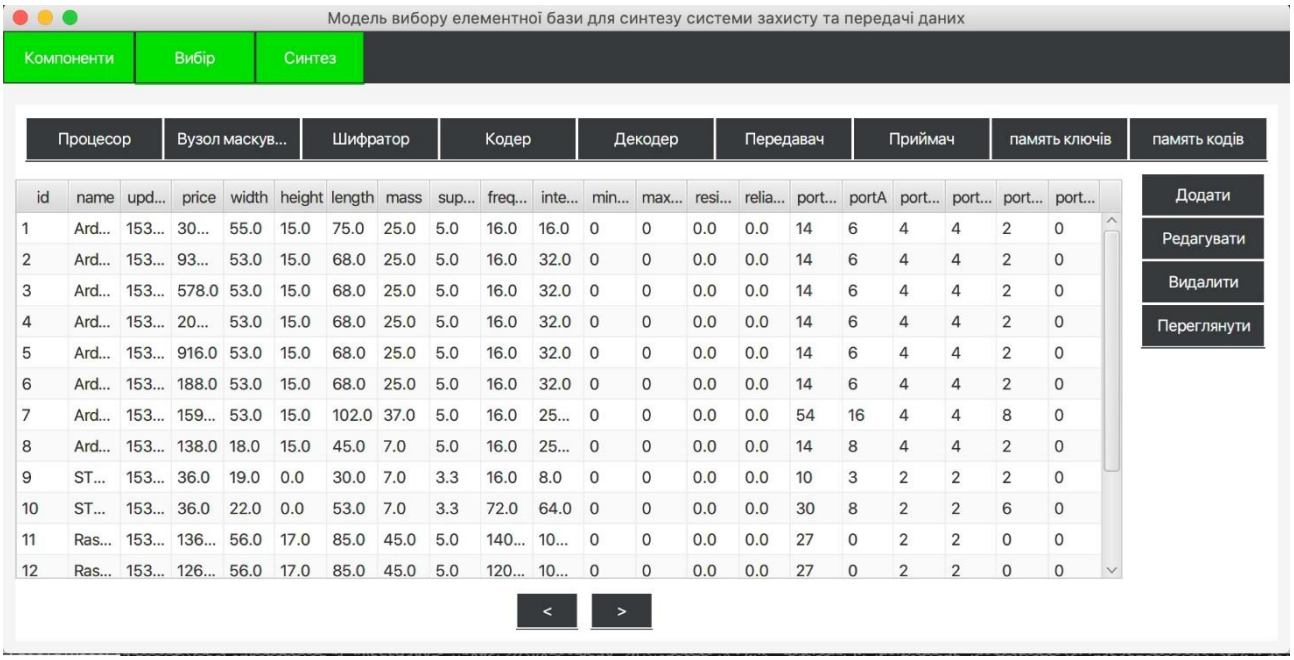


Рис. 4.2. Користувацький інтерфейс для засобів автоматизації синтезу СЗПД у реальному часі з використанням шумоподібних кодів

Розроблені засоби написані з використанням мови Java [119, 151], яка є кросплатформенною та дозволяє запускати програму без надлишкових налаштувань завдяки компіляції виконуючого коду у байткод, який однаково виконується на різних операційних системах. Java є об’єктно-орієнтованою мовою програмування. Мова була заснована у 1995 році та є однією з найпопулярніших мов програмування. У зв’язку із популярністю існує велика кількість сторонніх бібліотек для роботи із елементами інтерфейсів користувача, базами даних, тощо.

В процесі реалізації системи використано бібліотеки: JavaFX [152] – платформа та набір інструментів для генерації прогресивних додатків із різноманітними варіантами інтерфейсів користувача. В даному випадку JavaFX використана для генерації інтерфейсів користувача та опрацювання команд від адміністратора.

Загалом система включає меню для відображення та роботи із даними про базові компоненти (додавання, видалення, редагування та перегляд), для роботи із системою вибору (завантаження умов вибору компонент та модифікація умов вибору елементної бази), меню для системи синтезу, де можна вказати бажану конфігурацію системи та екрани для збереження та відображення результатів роботи системи вибору та синтезу.

Для реалізації елементів бази даних (БД) [126] було використано SQLite [91] – це є полегшена реляційна база даних, яка містити зменшений набір типів даних та зберігає БД як єдиний файл. Запропонована БД підходить для невеликих проектів і є простою у використанні. Реляційна база даних містить інформацію про існуючі елементи для системи синтезу та включає наступні таблиці (рис. 4.3):

– базова таблиця, яка містить усі спільні параметри елементів із елементної бази, вона описує загальні властивості компонентів, такі як назва, робочі температури, розміри, вага, необхідна напруга, тощо;

– основні компоненти, до яких належать процесори, вузли маскування, шифратори, дешифратори, кодери, декодери, передавачі, приймачі, елементи пам’яті для ключів та кодів.

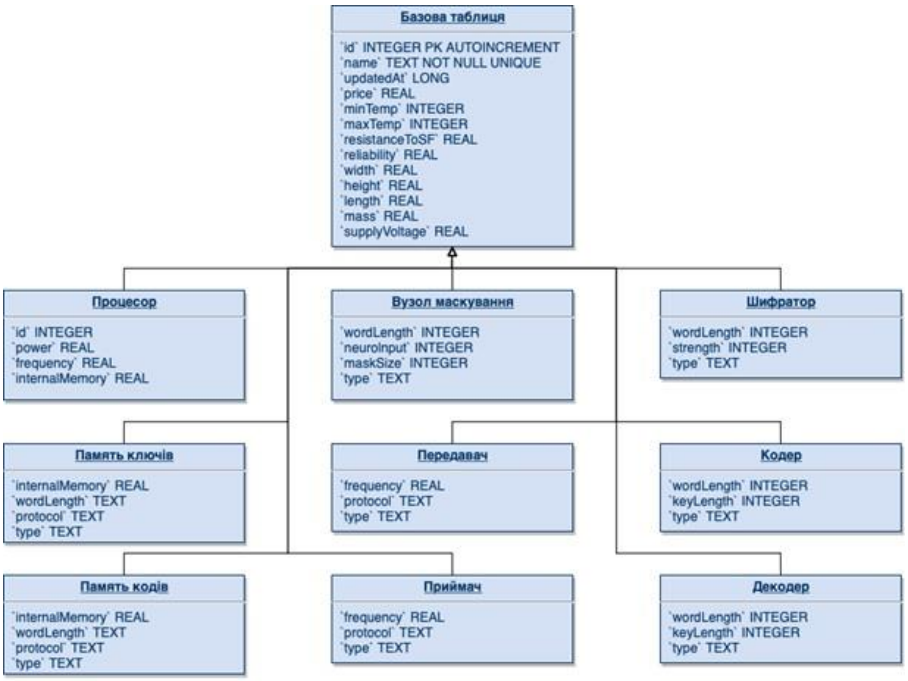


Рис. 4.3. Структура бази даних засобів автоматизованого синтезу СЗПД у реальному часі з використанням шумоподібних кодів

Для роботи із програмою оператору необхідно виконати декілька основних кроків, таких як:

Крок 1. Необхідно зайти у систему та перевірити актуальність даних про елементну базу.

Крок 2. Необхідно сформулювати умови для задачі вибору елементної бази, та зберегти налаштування пошуку елементної бази кожного вказаного типу.

Крок 3. Необхідно вказати умови задачі синтезу, який тип компонентів необхідно для реалізації системи та інші допоміжні параметри. На цьому етапі обчислювальне ядро системи запускає модулі пошуку елементної бази та отримані множини компонентів використовуються як вхідні дані для модуля синтезу. По завершенню обчислень формуються результати, які сортуються по спаданню інтегрованої оцінки ефективності компонент.

Приклад залежності кількості згенерованих альтернатив проектованого засобу захисту та передачі даних з використанням шумоподібних кодів з врахування параметру сумісності по портах та без його врахування наведений на рис. 4.4.

Рис. 4.4. Графік залежності варіантів поєднання елементів: синій – загальна кількість варіантів; червоний – кількість варіантів відсіяних фільтром суміності за портами

З отриманих результатів слідує, що врахування параметра сумістності базових елементів по портах дає змогу зменшити кількість альтернатив на від 30 до 40 %.

У роботі проведено серію симуляцій вибору елементної бази та синтезу СЗПД у реальному часі, різними технічними параметрами. Під час кожної із симуляцій здійснюється вибір мікроконтролера, блоків пам’яті, операційних вузлів і інтерфейсів зв’язку. Випробування проводилось на Macbook Pro 2015. Процесор i7 та 16 Гб оперативної пам’яті. Результат отриманих даних наведено у табл. 4.1.

Таблиця 4.1.

Залежність часу симуляції від кількість параметрів

Час (мс)	Кількість параметрів	Економія обчислення	Повний перебір	Відсіяні варіанти	Різниця
1	2	0	49	49	0
2	3	0	196	196	0
3	4	0	588	588	0
7	5	0	1764	1764	0
28	6	0	12348	12348	0
42	7	0	24696	24696	0
63	8	0	49393	49392	0
217	9	0	148176	148176	0
454	10	12,2448	296352	260063	36287
6546	11	36,7346	2074464	1312416	762048
20797	12	40,52478	6223392	3701376	2522016

Одним із важливих параметрів роботи засобів вибору елементної бази та синтезу СЗПД у реальному часі є залежність кількості варіантів поєднання компонентів від кількості параметрів пошуку. Засоби синтезу містять додатковий фільтр, що дозволяє відкидати ті варіанти, які не є сумісними за інтерфейсами зв’язку з оточуючим середовищем.

4.3. Розроблення програмних засобів нейроподібного криптографічного шифрування та дешифрування даних

Нейроподібне криптографічне шифрування та дешифрування даних реалізується за допомогою програмних модулів шифрування та дешифрування [71]. Вказані програмні модулі реалізуються за принципом відкритості програмного забезпечення, що передбачає можливості нарощування та автономного відлагодження.

Результатом роботи є розроблене програмне забезпечення для розрахунку вагових коефіцієнтів, які використовуються для шифрування вхідного повідомлення. Для реалізації поставленої задачі було обрано мову програмування C# та технологію Windows Forms.

Вхідними даними є розрядність повідомлення (n), розрядність входів (m) та вхідне повідомлення (рис. 4.5).

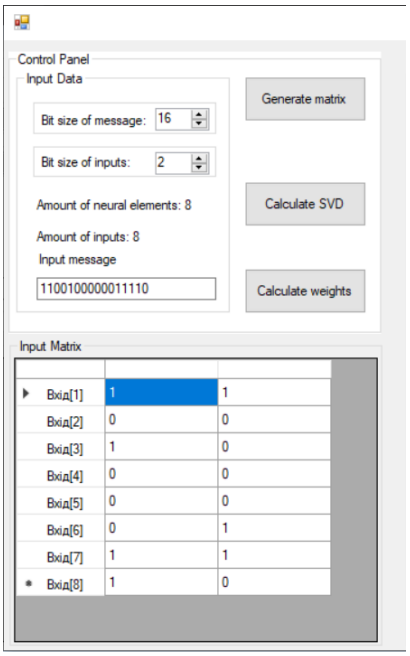


Рис. 4.5. Вхідні дані програми.

Після введення даних програма розраховує кількість нейроелементів та входів. За цей розрахунок відповідальна кнопка – *Generate matrix*.

Далі знаходяться матриці U, V та D за допомогою алгоритму SVD. Для запуску алгоритму необхідно натиснути відповідну кнопку *SVD*.

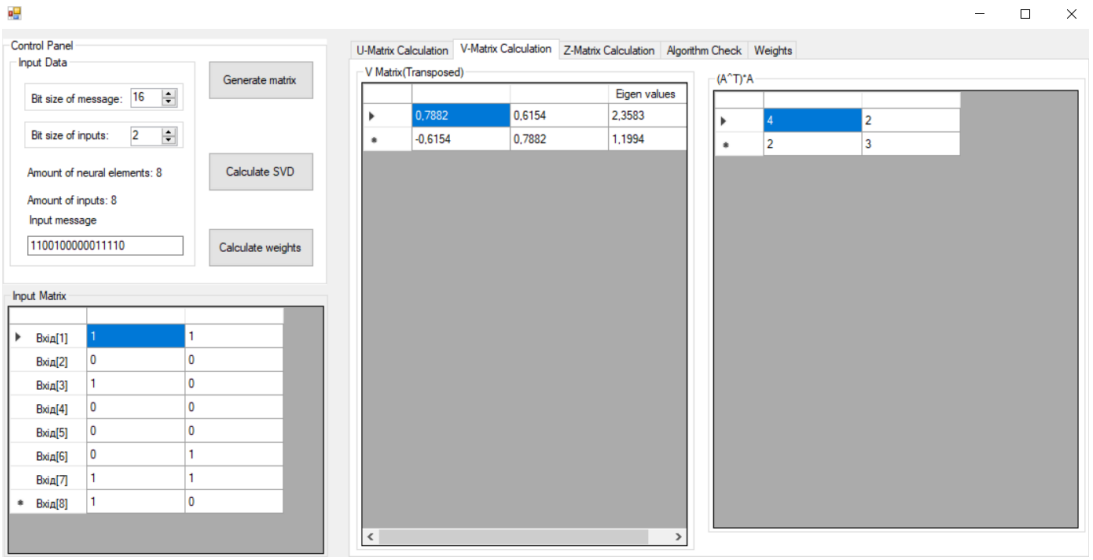


Рис. 4.6. Розрахунок матриці U.

Справа на рис.4.6 показано результат методу Якобі над результатом добутку матриць AA^T.

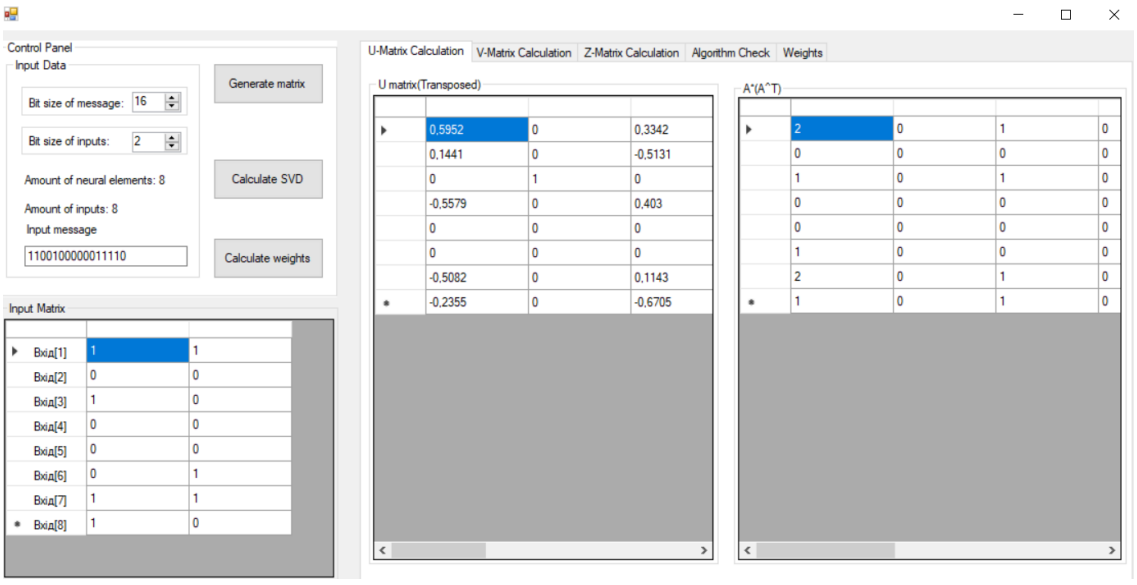


Рис. 4.7. Розрахунок матриці V.

На рис.4.7 показано результат добутку A^TA за методом обертання Якобі.

Матриця D складається з власних значень, розміщених на головній діагоналі (див. Рис.4.8). Далі відбувається знаходження матриці Z, яка буде використовуватись для перевірки роботи SVD. Ця матриця є результатом множення матриці U на матрицю D.

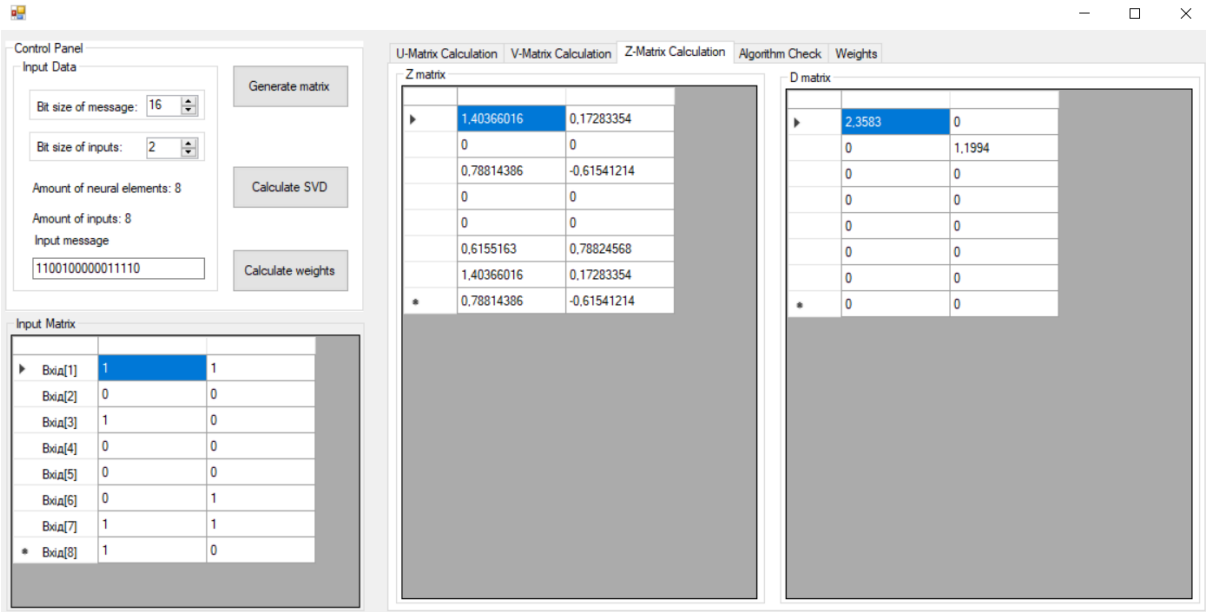


Рис. 4.8. Знаходження матриці D та розрахунок матриці Z.

На наступному рисунку зображено матрицю, яка є результатом множення матриці Z на матрицю V. За рахунок неї можна переконатись у правильності роботи алгоритму. Оскільки ця матриця має бути рівною вхідній матриці, хоча і з певною похибкою.

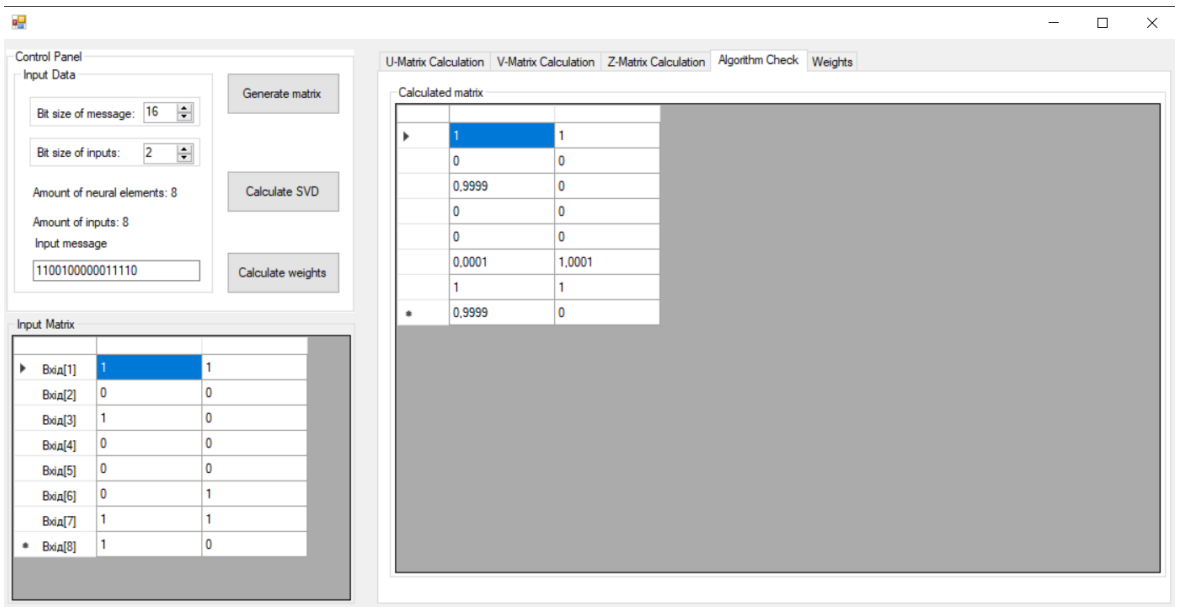


Рис. 4.9. Матриця перевірки.

Для знаходження вагових коефіцієнтів, які розраховуються за формулою (2.19) треба натиснути на *Calculate weights*. Для перевірки правильності проробленних розрахунків використовується формула (2.16).

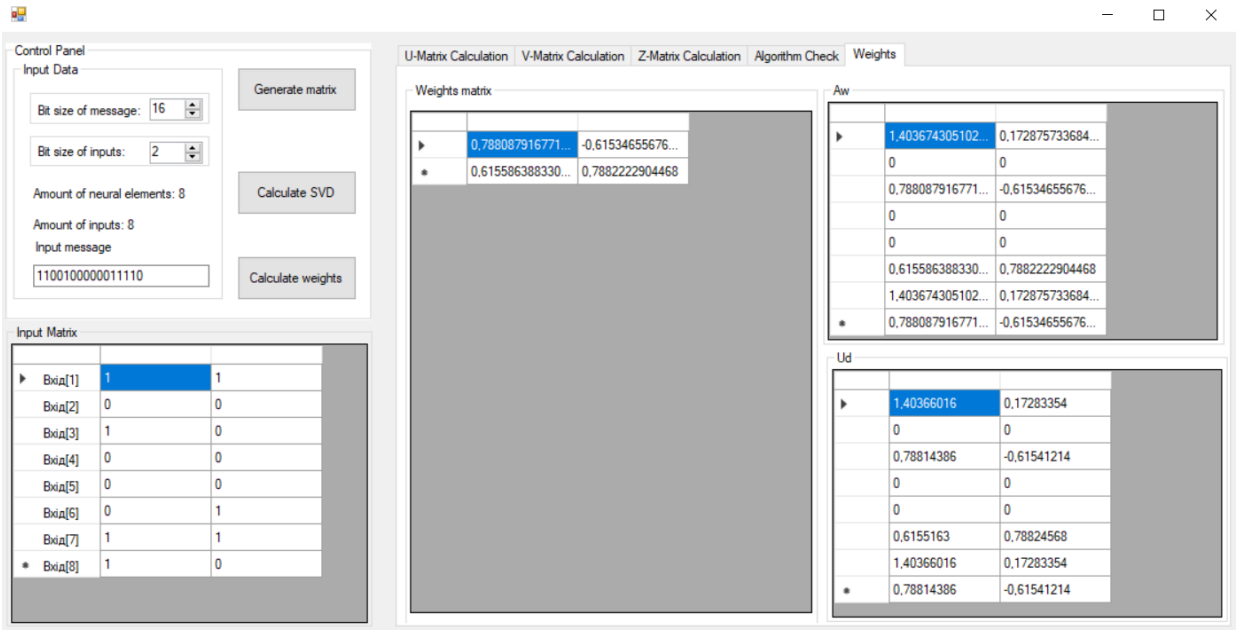


Рисунок 4.10. Матриця вагових коефіцієнтів (Matrix of weights)

4.4. Оцінювання часу шифрування та дешифрування команд управління рухом МРС на базі мікроконтролера

Далі було виконане відпрацювання вказаних засобів нейромережевого криптографічного шифрування та дешифрування для мобільної платформи [63, 64, 65], для чого у якості процесорного ядра використовувався мікрокомп’ютер NanoPi Duo фірми FriendlyElec. Він використовує SoC Cortex-A7 H2+ фірми Allwinner та містить 512 Мб DDR3 оперативної пам'яті та вбудований модуль WiFi. У якості ОС застосовується UbuntuCore [100].

Для тестування програмних модулів мобільної частини експериментальної системи нейромережевого криптографічного шифрування/дешифрування, кодування/декодування даних було виконано такі кроки. Для контролю процесів відлагодження за допомогою ПК з ОС Windows було використано програмний додаток PuTTY [136], для взаємодії з файловою системою мікрокомп’ютера було використано програмні засоби WinSCP. Така комбінація програмних засобів дозволила на ПК виконувати редагування, для маніпуляції з файлами

використовувати WinSCP, а компіляцію та тестове виконання розроблених програмних засобів нейромережевого криптографічного шифрування та дешифрування даних виконувати на мікрокомп’ютері через консоль за допомогою PuTTY.

Для цього використовувався макету мобільної частини експериментальної системи для відпрацювання засобів нейромережевого криптографічного шифрування та дешифрування детально описаний у попередньому звіті.

Слід зауважити, що через стандартний послідовний інтерфейс UART 0 реалізується консоль керування мікрокомп’ютером, а передача команд через радіоканал на блок прийомо-передавача для мобільної частини виконується через послідовний порт UART 1.

Як зазначалося вище, для шифрування 16 розрядної команди управління на мікрокомп’ютері було відлагоджено та протестовано три можливі варіанти архітектури нейроподібної мережі у такій конфігурації: кількість входів (k) – 8, 4, 2, розрядність входів (m) – 2, 4, 8 та кількість нейроелементів (N) – 8, 4, 2 відповідно.

З цією метою було створено три окремі каталоги для тестування раніше розроблених і відлагоджених файлів програм Training_ANN, EnCrypt_ANN, DeCrypt_ANN (рис. 4.11) засобами WinSCP. Вказані програми було скориговано для реалізації вказаних трьох конфігурацій нейромереж.

Для їх компіляції було застосовано штатний компілятор GCC з слід використанням команди на компіляцію з підключенням відповідної бібліотеки *math.*

gcc Training_ANN.c -o Training_ANN -lm,

де ключ –lm ініціалізує підключення бібліотеки math у процесі компіляції.

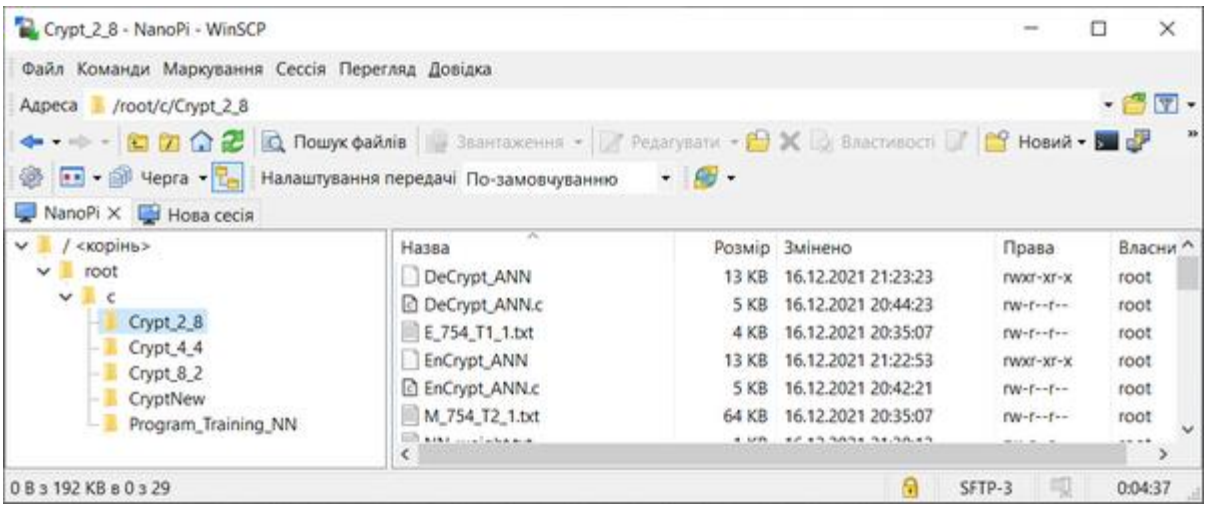


Рис. 4.11. Вміст папки мікрокомп’ютера з трьома каталогами варіантів архітектури нейроподібної мережі

Аналогічно здійснювалася компіляція модулів EnCrypt_ANN, DeCrypt_ANN. Спочатку за допомогою програмного модуля Training_ANN для заданих розрядності навчаючих векторів, розрядності вхідних нейронів нейроподібної мережі, кількості нейронів вхідного шару нейроподібної мережі відповідно до першого варіанту конфігурації – *m=2, k=8, N=8* було отримано вагові коефіцієнти для цієї архітектури та сформовано файли налаштування конфігурації. Робоче вікно наведено на рис. 4.12.

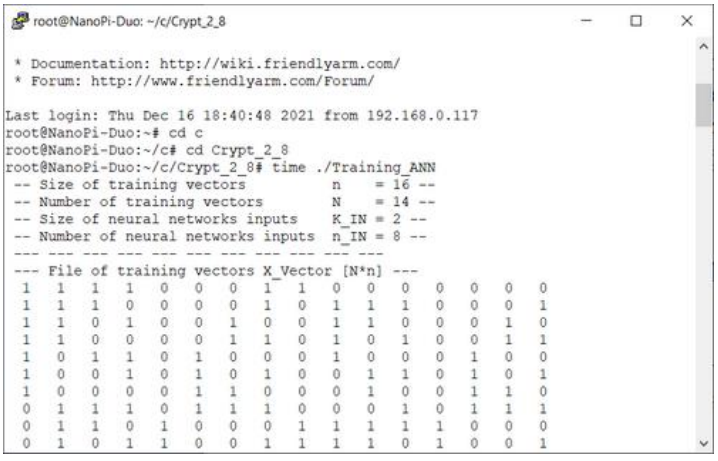


Рис. 4.12. Робоче вікно PuTTY при запуску *Training_ANN* на мікрокомп’ютері

При цьому, отримані у процесі обробки на мікрокомп’ютері дані повністю збігаються з наведеними у прикладі вище.

Далі виконувалося послідовно за допомогою нейромережі з визначеною архітектурою, яка використовує попередньо створені файли налаштування, шифрування даних на основі нейроподібної мережі за допомогою програмного модуля EnCrypt_ANN та їх дешифрування за допомогою програмного модуля DeCrypt_ANN,. Для тестування процесу шифрування використовувалися різні вхідні дані у вигляді вектора розрядністю n=16 з файлу X_Encrypt_16.txt, а результат дешифрування за допомогою вказаної архітектури нейромережі отримувався у файлі X_DE_OUT.txt. Вміст вхідного файлу на шифрування і вихідного дешифрованого порівнювався штатною командою

diff X_Encrypt_16.txt X_DE_OUT.txt.

При цьому, після виконання команд ./EnCrypt_ANN та ./DeCrypt_ANN вміст файлів був ідентичним, однак, при внесенні змін у файл з закодованими даними X_EN_OUT.txt, що імітувало збійні дані, вміст декодованого файлу X_DE_OUT.txt та оригінального відрізнявся, чого і слід було очікувати (рис. 4.13).

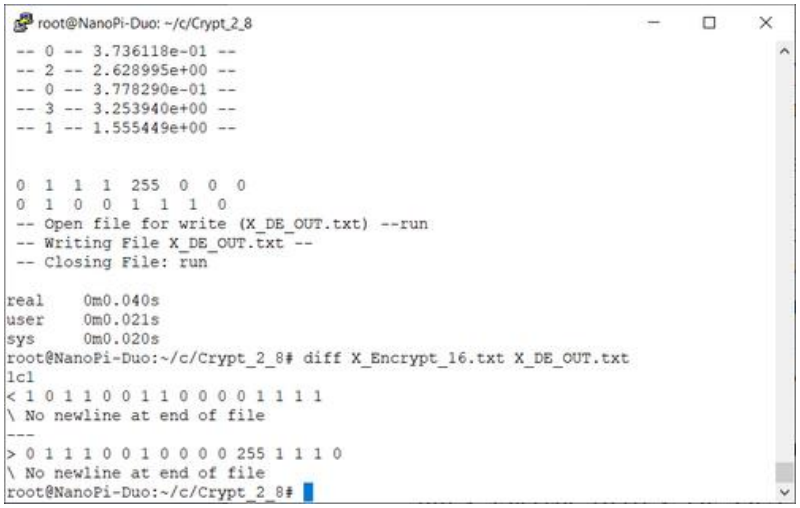


Рис. 4.13. Робоче вікно PuTTY при некоректному декодуванні

Аналогічні тестування було виконано для двох інших конфігурацій архітектури нейромереж – m=4, k=4, N=4 та m=8, k=2, N=2. При цьому, отримано аналогічні результати, тобто нейромережі з вказаними архітектурами успішно

забезпечували шифрування та дешифрування аналогічних файлів з вхідними векторами та реагували на пошкоджені вхідні дані.

Було отримано наступні результати шифрування окремих команд управління мобільною робототехнічною системою при 16-и розрядній системі команд.

Для команди Стоп, яка виглядає: 1000 0000 0000 0000, отримуємо зашифроване значення (файл X_EN_OUT.txt в форматі дійсних чисел):

-8.891016e-01 1.009038e+00 5.051861e-03 -3.808250e-01 -1.236387e+00 -1.327926e-01 7.070168e-01 -1.127011e-02,

Відповідні значення у форматі чисел IEEE 754 із плаваючою комою, що використовуються для програмних реалізацій арифметичних дій та для багатьох апаратних реалізацій (файл X_EN_OUT_754.txt):

BF639C29 3F812829 3BA58A14 BEC2FB7F

BF9E41F0 BE07FACB 3F34FF0E BC38A644.

Для порівняння, у випадку команди Вліво з кутом повороту 165 градусів, виглядає: 0101000010100101, отримуємо зашифроване значення у форматі дійсних чисел:

1.667356e-01 7.958972e-01 1.061270e+00 -7.180629e-01 8.315110e-01 -7.109410e-01 2.839336e+00 6.619364e-01,

Відповідні значення у форматі чисел IEEE 754:

3E2ABCBC 3F4BBFEB 3F87D7B3 BF37D2F8

3F54DDE7 BF36003B 4035B7B0 3F2974AA.

Для команди *Вперед* зі швидкість руху 7, яка виглядає: 0100001100000111, отримуємо зашифроване значення у форматі дійсних чисел:

-2.282535e+00 -2.001943e+00 -1.507132e+00 -9.636008e-02 2.819344e-02 2.705832e-02 2.839338e+00 6.619372e-01.

Відповідні значення у форматі чисел IEEE 754:

C012150E C0001FD4 BFC0E9B6 BDC5586F

3CE6F5F0 3CDDA967 4035B7B6 3F2974B7.

Отримувані у результаті шифрування дані не зберігають заданої структури команд і мають випадковий характер.

Далі було виконано оцінку часових затрат на забезпечення нейромережевого криптографічного шифрування та дешифрування даних за допомогою процесорного ядра на базі мікрокомп'ютера для вказаних трьох архітектур. Для оцінки часу застосовано штатну команду ОС *time* у форматі

time ./DeCrypt_ANN.

Результати її виконання наведено на рис. 4.14.

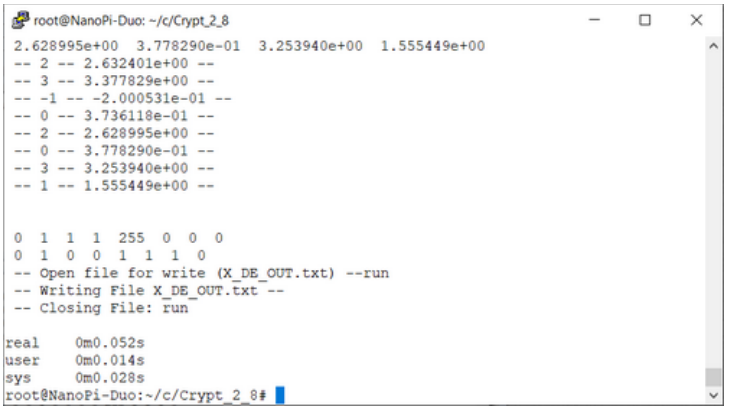


Рис. 4.14. Вікно PuTTY з часом виконання дешифрування *DeCrypt_ANN*

За результатами тестування для трьох архітектур нейромережі було отримано наступні дані (табл. 4.2). Для отримання більш точних результатів запуск програмних модулів виконувався декілька разів, а результати усереднювалися. Отримані результати візуалізовано на рис. 4.15.

Як показують результати тестування роботи програмних реалізацій нейромережевого шифрування/дешифрування даних найбільш тривала операція – формування і навчання нейромережі, а час її виконання на мікрокомп'ютері становить біля 200 мс і не сильно залежить від архітектури обраної нейромережі. З іншої сторони, час виконання процедур нейромережевого криптографічного шифрування та дешифрування блоків даних при реалізації на мікрокомп'ютері становить відповідно 30-38 мс та 23-35 мс.

Таблиця 4.2.

Час виконання операцій навчання, шифрування та дешифрування для трьох архітектур нейромережі

Модуль \ Архітектура (m; k; N;)	2; 8; 2;	4; 4; 4	8; 2; 2;
Training_ANN, мс	189,7	197,4	203,2
EnCrypt_ANN, мс	37,2	29,3	37,8
DeCrypt_ANN, мс	38,4	22,9	24,9

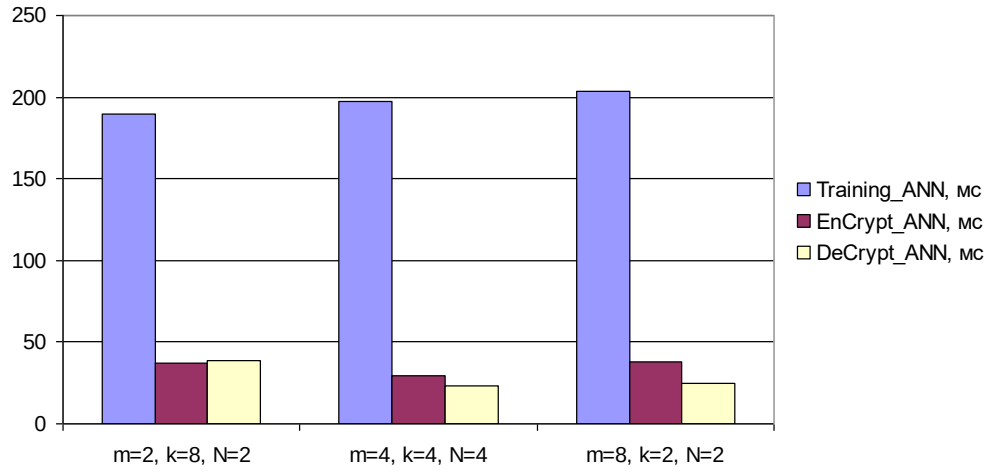


Рис. 4.15. Гістограма часу виконання операцій навчання, шифрування та дешифрування для трьох архітектур нейромережі

Отже, можна зробити висновок, що отриманий час шифрування/дешифрування становить біля 30 мс не сильно залежить від архітектурної конфігурації нейромережі і є прийнятним для реалізації вказаних задач.

При цьому, процедура формування і навчання нейромережі хоч і є на порядок тривалішою, однак виконується одноразово при зміні конфігурації нейромережі, а отже, не впливає на тривалість затримок у процесі шифрування/дешифрування.

Отримані значення часу виконання операцій можна зменшити шляхом подальшої оптимізації відповідних програмних модулів.

4.5. Висновки до розділу 4

1. Розроблено на основі вдосконаленого методу вибору елементної бази імітаційну модель, яка за рахунок використання наповненої сучасною елементною базою бази даних і врахування вимог технічного завдання забезпечує автоматизацію процесу вибору найефективнішої елементної бази для синтезу засобів криптографічного захисту даних у реальному часі.

2. Розроблено імітаційну модель знаходження вагових коефіцієнтів для попередньо заданої архітектури нейромережі. Приклад моделі продемонстровано на основі вхідного повідомлення – 110010000011110. Розрядність вхідних даних 16, розрядність входів 2, кількість нейронів 8 та кількість входів також 8.

3. Удосконалено метод сингулярного розкладу матриці для знаходження матриці вагових коефіцієнтів.

4. Практична цінність полягає у тому, що використання розробленої імітаційної моделі забезпечує швидке обчислення вагових коефіцієнтів

ВИСНОВКИ

У дисертаційному дослідженні, на основі виконаних теоретичних, а також експериментальних досліджень, розв’язно актуальне наукове завдання, а саме розроблення нових і вдосконалення існуючих методів, моделей та програмно-апаратних засобів інформаційної технології нейроподібного лінійного захисту даних у реальному часі з високими техніко-експлуатаційними характеристиками для мобільних смарт-систем.

Результи проведеного дослідження:

1. Вдосконалено нейроподібну мережу прямого поширення автоасоціативного типу на основі парадигми «модель послідовних геометричних перетворень» і адаптувано її до нейроподібного шифрування-дешифрування даних.

2. Вдосконалено метод обчислення вагових коефіцієнтів для різних архітектур нейроподібних мереж за рахунок покращення методу сингулярного розкладу матриць.

3. Розроблено метод таблично-алгоритмічного обчислення скалярного добутку з плаваючою комою в нейроподібних елементах.

4. Розроблено модель попередніх налаштувань для реалізації нейроподібного шифрування/дешифрування даних.

5. Вдосконалено метод вибору елементної бази для синтезу засобів криптографічного захисту даних у реальному часі.

6. Розроблено апаратно-програмні засоби інформаційної технології нейроподібного захисту даних у реальному часі із симетричними ключами для смарт-систем.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

[1] Цимбал Ю. В. Нейромережевий метод симетричного шифрування даних. *Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі.* 2018. № 901. С. 118–122.

[2] Tsmots I., Tsymbal Y., Skorokhoda O., Tkachenko R. Neural-like methods and hardware structures for real-time data encryption and decryption. *Комп'ютерні науки та інформаційні технології, CSIT-2019 :* матеріали XIV Міжнародної науково-технічної конференції (17–20 вересня 2019). Львів, 2019. С. 248–253.

[3] Різник О. Я., Ткаченко Р. О., Кинаш Ю. Є. Нейромережева технологія захисту та передачі даних у реальному часі з використанням шумоподібних кодів. Інноваційні технології у розвитку сучасного суспільства : збірник тез доповідей міжнародної науково-практичної конференції (18–19 квітня 2019 р.). Львів, 2019. С. 19–23.

[4] Diamantaras K. I., Kung S. Y. Principal Component Neural Networks. *Theory and Applications.* Wiley, 1996. 270 s.

[5] Цмоць І. Г., Рабик В. Г., Лукащук Ю. А. Розроблення мобільних засобів нейроподібного криптографічного шифрування та дешифрування даних у реальному часі. *Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі.* 2021. № 9. С. 84–95.

[6] Chi Zhang, Wei Zou, Liping Ma, Zhiqing Wang. Biologically inspired jumping robots. *A comprehensive review, Robotics and Autonomous Systems.* Volume 124. 2020.

[7] Śledź S., Ewertowski M. W., Piekarczyk J. Applications of unmanned aerial vehicle (UAV) surveys and Structure from Motion photogrammetry in glacial and periglacial geomorphology. *Geomorphology.* 2021. P. 378.

[8] Verma A., Ranga V. Security of RPL based 6LoWPAN Networks in the Internet of Things: A Review. *IEEE Sens.* J. 2020. Vol. 20. P. 5666–5690.

[9] Volna E., Kotyrba M., Kocian V., Janosek M. Cryptography Based on Neural Network. *Proceedings of the 26th European Conference on Modeling and Simulation.* 2012. P. 386–391.

[10] Shihab K. A backpropagation neural network for computer network security. *Journal of Computer Science.* Vol. 2. № 9. 2006. P. 710–715.

[11] Sagar V., Kumar K. A Symmetric Key Cryptographic Algorithm Using Counter Propagation Network (CPN). *Proceedings of the 2014 ACM International Conference on Information and Communication Technology for Competitive Strategies.* 2014. No 51. P. 1–5.

[12] Arvandi M., Wu S., Sadeghian A., Melek W. W., Woungang I. Symmetric cipher design using recurrent neural networks. *Proceedings of the IEEE International Joint Conference on Neural Networks.* 2006. P. 2039–2046.

[13] Tsmots I., Tsymbal Y., Khavalko V., Skorokhoda O., Tesluyk T. Neural-Like Means for Data Streams Encryption and Decryption in Real Time. *Processing of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing,* DSMP. 2018. P. 438–443.

[14] Rabyk V., Tsmots I., Lyubun Z., Skorokhoda O. Method and Means of Symmetric Real-time Neural Network Data Encryption. *2020 IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies,* CSIT 2020 – Proceedings, 2020. № 1. P. 47–50.

[15] Khavalko Viktor, Tsmots Ivan. (2019). Image classification and recognition on the base of autoassociative neural network usage. *2019 IEEE 2nd Ukraine conference on electrical and computer engineering, UKRCON-2019 : conference proceedings* (Lviv, July 2–6, 2019). S. 1118–1121.

[16] Tsmots Ivan, Rabyk Vasyl, Skorokhoda Oleksa, Teslyuk Taras. (2019). Neural element of parallel-stream type with preliminary formation of group partial products. *Electronics and information technologies (ELIT-2019) :* proceedings of the XIth International scientific and practical conference (16 –18 September, 2019). Lviv, 2019. C. 154–158.

[17] Tsmots I., Rabyk V., Skorokhoda O., Tsymbal Y. (2021). Neural-like real-time data protection and transmission system. *Advances in Intelligent Systems and Computing (AISC).* Vol. 1293 : Advances in Intelligent Systems and Computing V.

Selected papers from the International conference on computer science and information technologies.

[18] Цмоць І. Г., Лукащук Ю. А., Хавалко В. М., Рабик В. Г. (2019). Моделі нейроподібного елемента паралельно-паралельного типу. *Моделювання та інформаційні технології.* Вип. 86. С. 119–126.

[19] Tsmots Ivan, Skorokhoda Oleksa, Ignatyev Ihor, Rabyk Vasyl. (2017). Basic Vertical-Parallel Real Time Neural Network Components. Proceedings of XIIth International Scientific and Technical Conference CSIT 2017 (5–8 September). Lviv, 2017. P. 344–347.

[20] Цмоць І. Г., Скорохода О. В. (2011). Пристрій для обчислення скалярного добутку. Патент України на корисну модель № 66138, бюл. № 24.

[21] Цмоць І. Г., Скорохода О. В., Теслюк В. М. (2013). Пристрій для обчислення скалярного добутку. Патент України на винахід № 101922, 13.05.2013, бюл. № 9.

[22] Цмоць І. Г., Скорохода О. В., Медиковський М. О. Пристрій для обчислення скалярного добутку. Патент України на винахід № 118596, 11.02.2019, бюл. № 3.

[23] Цмоць І. Г., Теслюк В. М., Теслюк Т. В., Медиковський М. О., Цимбал Ю. В. (2019). Пристрій для обчислення сум парних добутків. Патент України № 120210, 25.10.2019, бюл. № 20/2019.

[24] Kinzel Wolfgang, Kanter Ido. Neural cryptography. Proceedings of the 9th International Conference on Neural Information Processing. *ICONIP'02*. URL: http://dx.doi.org/10.1109/ICONIP.2002.1202841

[25] Sooyong Jeong, Cheolhee Park, Dowon Hong, Changho Seo, Namsu Jho. Neural Cryptography Based on Generalized Tree Parity Machine for Real-Life Systems. *Security and Communication Networks.* 2021. URL: https://doi.org/10.1155/2021/6680782

[26] Dong T., Huang T. Neural cryptography based on complex-valued neural network. *IEEE Transactions on Neural Networks and Learning Systems.* Vol. 31. № 11. 2019.

[27] Tkachenko R., Izonin I. 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. V*ol. 754. 2019. P. 578–587.

[28] Chang A. X. M., Martini B., Culurciello E. Recurrent neural networks hardware implementation on FPGA: arXiv preprint arXiv:1511.05552. 2015.

[29] Amos R. O., Jagath C. R. FPGA Implementations of Neural Networks. Springer, 2006. 363 p.

[30] Corona-Bermúdez E., Chimal-Eguía J. C., Téllez-Castillo G. Cryptographic Services Based on Elementary and Chaotic Cellular Automata. *Electronics.* 2022. Vol. 11(4). P. 613. URL: https://doi.org/10.3390/electronics11040613.

[31] Субботін С. О. Нейронні мережі: теорія та практика : навч. посіб. Житомир : Вид. О. О. Євенок, 2020. 184 с.

[32] Audhkhasi K., Osoba O., Kosko B. Noise-boosted back propagation and deep learning neural networks. 2016.

[33] Hinkelmann K. Neural Networks. University of Applied Sciences, Northwestern Switzerland, 2018. P. 7. URL: https://web.archive.org/web/20181006235506/http://didattica.cs.unicam.it/lib/exe/fetch.php?media=didattica:magistrale:kebi:ay_1718:ke-11_neural_ networks.pdf. (дата звернення: 10. 01.2023).

[34] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks.* 2015. Vol. 61. P. 85–117. URL: https://doi.org/10.1016/j.neunet.2014.09.003

[35] Valueva M. V., Nagornov N. N., Lyakhov P. A., Valuev G. V., Chervyakov N. I. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation.* 2020. 177. P. 232–243. URL: https://doi.org/10.1016/j.matcom.2020.04.031

[36] Breiman L. Random forests. *Machine Learning.* 2001. 45. P. 5–32. URL: https://doi.org/10.1023/A:1010933404324

[37] Witten I., Frank E., Hall M. Data Mining: Practical Machine Learning. Morgan Kaufmann, 2011. 629 c.

[38] Bishop C. Pattern recognition and machine learning. Berlin : Springer, 2006. 738 c.

[39] Björk A. L. Numerical methods in matrix computations. Texts in applied mathematics. Vol. 59. Springe, 2015.

[40] Brandts J. The riccati algorithm for eigenvalues and invariant subspaces of matrices with inexpensive action. *Linear Algebra.* Appl. 358. 2003. P. 335–365.

[41] Hari V. On the global convergence of the block Jacobi method for the positive definite generalized eigenvalue problem. Calcolo 58. 2021. P. 24.

[42] Loan C. V. The block Jacobi method for computing the singular value decomposition. In: Byrnes C., Lindquist A. (eds.). *Computational and Combinatorial Methods in Systems Theory.* P. 245–255. Amsterdam, Elsevier Science Publishers B. V. North-Holland, 1986.

[43] Mahoney M. W. et al.: Randomized algorithms for matrices and data, Foundations and Trends. Mach. Learn. 3, 2011. P. 123–224.

[44] MathWorks Release 14. MATLAB 7. URL: http: www.mathworks. com/products/ new_ products/ R14_transition. html.

[45] Lytvynenko V., Wojcik W., Fefelov A., Lurie I., Savina N., Voronenko M., Boskin O. and Smailova S. Hybrid Methods of GMDH-Neural Networks Synthesis and Training for Solving Problems of Time Series Forecasting. *Advances in Intelligent Systems and Computing.* Vol. 1020. P. 513–531, 2020. DOI: 10.1007/978-3-030-26474-1_36.

[46] Руденко О. В., Бодянський Є. В. Штучні нейронні мережі : навчальний посібник. Харків : ТОВ «Компанія СМІТ», 2006. 404 с.

[47] Maass W., Natschläger T. and Markram H. Computer models and analysis tools for neural microcircuits, in edit. R. Kötter : Neuroscience Databases. A Practical Guide. Chapter 9, Boston : Kluwer Academic Publishers, 2003. P. 123–138.

[48] Bohte S., Kok J., Poutré H. La. Errorbackpropagation in temporally encoded networks of spiking neurons. *Neurocomputing.* 2002. Vol. 48. № 1– 4. P. 17–37.

[49] Moore S. Back-Propagation in Spiking Neural Networks, M. Sc. thesis, University of Bath. URL: http://www.simonchristianmoore.co.uk.

[50] Prots'ko I. Synthesis of Efficient Algorithms of DST for Types I, IV via Cyclic Convolutions. *International Journal of Electronic Engineering and Computer Science.* 2016. Vol.1. № 1. P. 6–13.

[51] Elmaghraby A. S., Losavio M. M. Cyber security challenges in Smart Cities: safety, security and privacy. *Journal of Advanced Research.* 2014. Vol. 5. P. 491–497.

[52] Yakymenko I. Z., Kasianchuk M. M., Ivasiev S. V., Melnyk A. M., Nykolaichuk Y. M. Realization of RSA cryptographic algorithm based on vector-module method of modular exponention. *Proceedings of 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET–2018).* 2018. P. 550–554.

[53] Babash A., Baranova E. Cryptographic methods of information protection. Publishing House : Knorus, 2016. P. 190.

[54] Marler R., Arora J. Survey of multi-objective optimization methods for engineering. *Struct Multidisc Optim.* № 26. 2004. P. 369–395.

[55] Tsmots I., Skorokhoda O. Methods and VLSI-structures for Neural Element Implementation. Perspective Technologies and Methods in MEMS Design, MEMSTECH'2010. *Processing of the 6th International Conference.* Polyana. P. 135.

[56] Kotsovsky V., Geche F., Batyuk A. Artificial complex neurons with half-plane-like and angle-like activation function. *Proceedings of the International Conference on Computer Sciences and Information Technologies.* CSIT, 14–17 September, 2015, Lviv. P. 57–59.

[57] Shagurin I., Shaltyrev V. Creation of systems on a chip based on FPGAs using synthesized processor cores. *Problems of the development of promising microelectronic systems : Sat. scientific tr.* M. : IPPM RAS, 2006. P. 382–385.

[58] Palagin A., Yakovlev Yu. Features of designing computer systems based on FPGAs. *Mathematical Machines and Systems.* 2017. № 2. P. 3–14.

[59] Marković M. Data Protection Techniques and Cryptographic Protocols in Modern Computer Networks. *13-th International Conference on Telecommunications ICT* (May 9–12). 2006.

[60] Schneier B. "Applied Cryptography" in Protocols Algorithms and Source Code in C. New York, Chichester, Brisbane, Toronto, Singapore : John Wiley & Sons, Inc., 1996.

[61] Biazar J., Ghanbary B. A new approach for solving systems of non-linear equations. *International Mathematical Forum.* 2008. № 3(38). P. 1885–1889.

[62] Crina G., Ajith A. A new approach for solving non-linear equations system. *IEEE Transaction on Systems, Man and Cybernetic.* 2008. № 38(3). P. 698–714.

[63] Diaa S. A. M., Hatem M. A. K., Mohiy M. H. (2010). Evaluating the performance of symmetric encryption algorithms. *International Journal of Network Security.* Vol. 10(3). P. 213–219.

[64] Kessler G. C. Handbook on local area networks: An overview of cryptography. United Kingdom : Auerbach. 2010. URL: http://www.garykessler.net/library

[65] Su N., Zobel R. N., Iwu F. O. Simulation in cryptographic protocol design and analysis. *Proceedings of 15th European Simulation Symposium.* University of Manchester. 2003, UK.

[66] Talbot J., Welsh D. Complexity and cryptography : An introduction. New York : Cambridge University Press, 2006.

[67] Maniyath S. R., Thanikaiselvan V. An Efficient Image encryption using Deep Neural Network and Chaotic Map. Microprocess. Microsyst. P. 103–134, 2020. DOI: 10.1016/j.micpro.2020.103134

[68] Sharma A. Comparative Study of Symmetric Cryptography Algorithm. *Pacific Univ.* December. 2014. P. 73. DOI: 10.13140/RG.2.1.1031.5601.

[69] Ahmad J. I., Din R., Ahmad M. Analysis review on public key cryptography algorithms. *Indones. J. Electr. Eng. Comput.* Sci. 12 (2). 2018. P. 447–454. URL: https://doi.org/10.11591/ijeecs.v12.i2.

[70] Bjorn Solvang, Gabor Sziebig, and Peter Korondi. Multilevel Control of Flexible Manufacturing Systems / Int. *Conf. on Flexible Manufacturing Systems*. Krakow, Poland, May 25–27, 2008, pp. 785–790.

[71] Bhardwaj A., Som S. Study of different cryptographic technique and challenges in future. *1st Int. Conf. Innov. Challenges Cyber Secur. ICICCS.* 2016. № Iciccs. P. 208–212. DOI: 10.1109/ICICCS.2016.7542353.

[72] Frolov V., Pidorich A. Optimization of the elemental base of electronic equipment. *Radioelectronics and informatics: scientific-technical journal.* Kh. : KhTURE, 2000. Volume 1. P. 26–27.

[73] Kravets P., Shymkovych V. Hardware Implementation Neural Network Controller on FPGA for Stability Ball on the Platform. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. *Advances in Intelligent Systems and Computing.* Volume 938. Springer, Cham. P. 247–256.

[74] Teslyuk T., Denysyuk P., Tsmots I., Kernytskyy A., Teslyuk V. and Berezsky O. Interface-Sensitive Method of Synthesis of Microcontroller-Based System Structures. *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM).* Polyana, Ukraine, 2019. P. 1–4. DOI: 10.1109/CADSM.2019.8779304.

[75] Tsmots I., Skorokhoda O., Rabyk V. Structure Software Model of a Parallel-Vertical Multi-input Adder for FPGA Implementation. Computer Sciences and Information Technologies. P*roceedings of 11th International Scientific and Technical Conference CSIT,* 2016. Lviv. P. 158–160.

[76] Kellman M., Rivest F., Pechacek A. Sohn L. and Lustig M. Barker-Coded node-pore resistive pulse sensing with built-in coincidence correction. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* New Orleans, LA, 2017. P. 1053–1057.

[77] Tsmots I., Skorokhoda O., Tesliuk T., Rabyk V. Designing Features of Hardware and Software Tools for Intelligent Processing of Intensive Data Streams.

Processing of the 2016 IEEE First International Conference on Data Streams and Processing. DSMP, 2016. Lviv. P. 332–335.

[78] Denysyuk P., Teslyuk V., Tariq Ali AlOmari and Teslyuk T. Development and study of subsystem for solution of tasks of multicriterial optimization. *5th International Conference on Perspective Technologies and Methods in MEMS Design.* Zakarpattya, 2009. P. 166–167.

[79] Naveed K., Latif Anjum M., Hussain W., Lee D. Deep introspective SLAM: deep reinforcement learning based approach to avoid tracking failure in visual SLAM. *Auton Robot.* Vol. 46. 2022. P. 705–724. URL: https://doi.org/10.1007/s10514-022-10046-9

[80] Li D., Ma G., He W., Ge S. S., Lee T. H. Cooperative Circumnavigation Control of Networked Microsatellites. *IEEE Trans.* Cybern. 2020. Vol. 50. P. 4550–4555.

[81] Shafique A., Mehmood A., Elhadef M., Khan K. H. A lightweight noise-tolerant encryption scheme for secure communication : An unmanned aerial vehicle application. *PLoS ONE.* 2022. P. 17.

[82] Verma A., Ranga V. Security of RPL based 6LoWPAN Networks in the Internet of Things: A Review. *IEEE Sens.* J. 2020. Vol. 20. P. 5666–5690.

[83] Srivastava S., Bhatia A. On the Learning Capabilities of Recurrent Neural Networks : A Cryptographic Perspective. *In Proceedings of the 2018 IEEE International Conference on Big Knowledge (ICBK)* (Singapore, 17–18 November). 2018. P. 162–167.

[84] Zhu Y., Vargas D. V., Sakurai K. Neural Cryptography Based on the Topology Evolving Neural Networks. I*n Proceedings of the 2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW)* (27–30 November, 2018). Takayama, Japan, 2018. P. 472–478.

[85] Duan X., Han Y., Wang C., Ni H. Optimization of Encrypted Communication Model Based on Generative Adversarial Network. *In Proceedings of the 2022 International Conference on Blockchain Technology and Information Security (ICBCTIS)* (Huaihua City, China, 15–17 July). 2022. P. 20–24.

[86] Karakaya B., Celik V., Gulten A. Realization of Delayed Cellular Neural Network model ON FPGA. *In Proceedings of the 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT).* Istanbul, Turkey, 18–19 April 2018. P. 1–4.

[87] Volna E., Kotyrba M., Kocian V., Janosek M. Cryptography Based on Neural Network. *In Proceedings of the 26th European Conference on Modeling and Simulation (ECMS 2012)* (Koblenz, Germany, 29 May–1 June). 2012; Troitzsch K. G., Moehring M., Lotzmann U., Eds. European Council for Modeling and Simulation : Caserta, Italy, 2012. P. 386–391.

[88] Shihab K. A backpropagation neural network for computer network security. *J. Comput. Sci.* 2006. Vol. 2. P. 710–715.

[89] Maćkiewicz A., Ratajczak W. Principal components analysis (PCA). 1993. Volume 19. Issue 3. P. 303–342. URL: https://doi.org/10.1016/0098-3004(93)90090-R

[90] The Fast Code. URL: https://www.thefastcode.com/uk-uah/article/what-is-a-system-on-a-chip-soc-. (дата звернення 23.04.2021).

[91] SQLite. URL: https://www.sqlite.org/index.html (дата звернення: 15.06.2019).

[92] Gribachev V. P. Element base of hardware implementations of neural networks. *Components and technologies.* № 8. 2006.

[93] Haykin S. Neural networks and learning machines 3rd ed. New York : Prentice Hall, 2009.

[94] Bodyanskiy Ye. V. and Rudenko O. G. Artificial neural networks : architectures learning applications. Kharkiv : TELETEH, 2004.

[95] Zolfaghari B., Koshiba T. The Dichotomy of Neural Networks and Cryptography: War and Peace. *Appl. Syst. Innov.* Vol. 5. № 61. 2022. URL: https://doi.org/10.3390/asi5040061.

[96] Dong T. and Huang T. Neural Cryptography Based on Complex-Valued Neural Network. *IEEE Transactions on Neural Networks and Learning Systems.* Vol. 31. № 11. P. 4999–5004. Nov. 2020.

[97] Meraouche I., Dutta S., Tan H. and Sakurai K. Neural Networks-Based Cryptography: A Survey. *IEEE Access.* 2021. Vol. 9. P. 124727–124740.

[98] Saraswat P., Garg K., Tripathi R., Agarwal A. Encryption Algorithm Based on Neural Network. *4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU).* 2019. P. 1–5.

[99] Forgáč R., Očkay M. Contribution to Symmetric Cryptography by Convolutional Neural Networks. *Communication and Information Technologies (KIT).* 2019. P. 1–6.

[100] Brassai S. T., Hammas A., Bustya B. Framework for neural network hardware implementation. *23rd International Carpathian Control Conference (ICCC).* 2022. P. 387–391.

[101] Vinaykumar S. and T. R. FPGA Implementation of Artificial Neural Network (ANN) for ECG Signal Classification. *IEEE International IOT Electronics and Mechatronics Conference (IEMTRONICS).* 2022. P. 1–6.

[102] Gaier A., Ha D. Weight Agnostic Neural Networks. arXiv:1906.04358. 2019.

[103] Luo L., Xiong Y., Liu Y., Sun X. Adaptive Gradient Methods with Dynamic Bound of Learning Rate. arXiv:1902.09843. 2019.

[104] You Y., Li J., Reddi S., Hseu J., Kumar S., Bhojanapalli S., Song X., Demmel J., Keutzer K., Hsieh C.-J. Large Batch Optimization for Deep Learning : Training BERT in 76 minutes. arXiv:1904.00962. 2019.

[105] Wang N., Choi J., Brand D., Chen C.-Y., Gopalakrishnan K. Training Deep Neural Networks with 8-bit Floating Point Numbers. arXiv:1812.08011. 2018.

[106] Kotsovsky V., Batyuk A. Representational Capabilities and Learning of Bithreshold Neural Networks. In: Babichev S., Lytvynenko V., Wójcik W., Vyshemyrskaya S. (eds) Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2020. *Advances in Intelligent Systems and Computing.* 2021. Vol. 1246. Springer, Cham. P. 499–514.

[107] Izonin I., Tkachenko R., Pidkostelnyi R., Pavliuk O., Khavalko V., Batyuk A. Experimental evaluation of the effectiveness of ANN-based numerical data augmentation methods for diagnostics tasks. *CEUR Workshop Proceedings.* Vol. 3038: Proceedings of the 4th International conference on informatics & data-driven medicine IDDM 2021, Valencia, Spain (November 19–21). 2021. P. 223–232.

[108] Geche F., Mulesa O., Batyuk A., Voloshchuk V. Properties of Logical Functions Implemented by One Generalized Neural Element over the Galois Field. In: Shakhovska N., Medykovskyy M. O. (eds). Advances in Intelligent Systems and Computing IV. CSIT 2019. *Advances in Intelligent Systems and Computing.* 2020. Vol. 1080. Springer, Cham. P. 202–213.

[109] Kotsovsky V., Geche F., Batyuk A. On the Computational Complexity of Learning Bithreshold Neural Units and Networks. In: Lytvynenko V., Babichev S., Wójcik W., Vynokurova O., Vyshemyrskaya S., Radetskaya S. (eds) Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2019. *Advances in Intelligent Systems and Computing*. 2019. Vol. 1020. Springer, Cham. P. 189–202.

[110] Batyuk A., Voityshyn V. Distributed software system with web interface for automated business process discovery. *Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі*. 2019. Вип. 5. С. 70–77.

[111] Мулеса О. Ю., Гече Ф. Е., Батюк А. Є., Мельник О. О. Інформаційна технологія прогнозування часових рядів методом синтезу прогнозної схеми. *Український журнал інформаційних технологій.* 2021. Вип. 3. № 2. С. 81–86.

[112] C# Advanced Topics. URL: https://net-informations.com/csharp/adv/default.htm. (дата звернення: 12.10.2020).

[113] Lytvyn V., Peleshchak R., Peleshchak I., Cherniak O., & Demkiv L. Building a mathematical model and an algorithm for training a neural network with sparse dipole synaptic connections for image recognition. *Eastern-European Journal of Enterprise Technologies*. 2021. № 6 (4 (114). P. 21–27.

[114] Peleshchak Roman, Lytvyn Vasyl, Doroshenko Mykola, Peleshchak Ivan, Sidletskyi Sviatoslav. Distorted character recognition by an incompatible single-layer dipole neural network. *Vìsnik Nacìonal'nogo unìversitetu «L'vìvs'ka polìtehnìka». Serìâ Ìnformacìjnì sistemi ta merežì.* 2022. P. 199–207.

[115] Peleshchak Roman, Lytvyn Vasyl, Bihun Oksana, Peleshchak Ivan. Structural Transformations of Incoming Signal by a Single Nonlinear Oscillatory Neuron or

by an Artificial Nonlinear Neural Network. *International Journal of Intelligent Systems and Applications (IJISA).* 2019. Vol. 11. № 8. P. 1–10.

[116] Ivantyshyn Danylo, Burov Yevhen, Lytvyn Vasyl. Architecture of intellectual system for research of space weather parameters. *Vìsnik Nacìonal'nogo unìversitetu «L'vìvs'ka polìtehnìka».* Serìâ Ìnformacìjnì sistemi ta merežì. 2021.

[117] Peleshchak Roman, Lytvyn Vasyl, Peleshchak Ivan, Vysotska Viktoriia, Cherniak Oksana. Construction of an Optimized Multilayer Neural Network Within a Nonlinear Model of Generalized Error. *Vìsnik Nacìonal'nogo unìversitetu «L'vìvs'ka polìtehnìka». Serìâ Ìnformacìjnì sistemi ta merežì.* 2021.

[118] Lytvyn Vasyl, Vysotska Victoria, Demchuk Andrii, Demkiv Ihor, Ukhanska Oksana, Hladun Volodymyr, Kovalchuk Roman, Petruchenko Oksana, Dzyubyk Lyudmyla, Sokulska Nataliia. Design of the architecture of an intelligent system for distributing commercial content in the internet space based on SEO-technologies, neural networks and Machine Learning. *Eastern-European Journal of Enterprise Technologies.* 2019. Vol. 2(2 (98). P. 15–34.

[119] GUI Programming – Java Programming Tutorial. URL: https://www3.ntu.edu.sg/home/ehchua/programming/java/j4a_gui.html. (дата звернення: 16.09.2020).

[120] Lytvynenko V., Naumov O., Voronenko M., Krejci J., Naumova L., Nikytenko D., Savina N. Dynamic bayesian networks application for evaluating the investment projects effectiveness. *Advances in Intelligent Systems and Computing*, 1246 AISC. 2021. P. 315–330.

[121] Lurie I., Lytvynenko V., Olszewski S., Voronenko M., Woicik W., Boskin O., Zhunissova U., Sherstiuk M. Application of inductive bayesian hierarchical clustering algorithm to identify brain tumors. *Advances in Intelligent Systems and Computing*, 1246 AISC. 2021. P. 567–584.

[122] Voronenko M., Naumov O., Naumova L., Topalova E., Filippova V., Lytvynenko V. Analysis of the Effectiveness of an Investment Project Using Statistical Bayesian Networks. *10th International Conference on Advanced Computer Information Technologies, ACIT*. 2020. Proceedings. Art. № 9208982. P. 408–411.

[123] Fefelova I., Fefelov A., Voronenko M., Kornelyuk A., Sachenko A., Ryzhkov E., Lytvynenko V. Predicting the Protein Tertiary Structure by Hybrid Clonal Selection Algorithms on 3D Square Lattice (2020) Proceedings. *15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET*. 2020. Art. № 9088634. P. 965–968.

[124] Lytvynenko V., Wojcik W., Fefelov A., Lurie I., Savina N., Voronenko M., Boskin O., Smailova S. Hybrid Methods of GMDH-Neural Networks Synthesis and Training for Solving Problems of Time Series Forecasting. *Advances in Intelligent Systems and Computing*. 2020. Vol. 1020. P. 513–531. Cited 23 times.

[125] Fefelova I., Fefelov A., Lytvynenko V., Dzierżak R., Lurie I., Savina N., Voronenko M., Vyshemyrska S. Protein Tertiary Structure Prediction with Hybrid Clonal Selection and Differential Evolution Algorithms. *Advances in Intelligent Systems and Computing*. 2020. Vol. 1020. P. 673–688.

[126] Database Tutorial. URL: https://www.quackit.com/database/tutorial/ (дата звернення: 23.09.2020).

[127] Murzenko O., Olszewski S., Boskin O., Lurie I., Savina N., Voronenko M., Lytvynenko V. Application of a combined approach for predicting a peptide-protein binding affinity using regulatory regression methods with advance reduction of features (2019) *Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. IDAACS. 2019, 1, art. № 8924244. P. 431–435.

[128] Vynokurova O., Peleshko D., Peleshko M. Hybrid Deep Convolutional Neural Network with Multimodal Fusion. In: Babichev S., Peleshko D., Vynokurova O. (eds) Data Stream Mining & Processing. DSMP 2020. *Communications in Computer and Information Science*. Vol. 1158. Springer, Cham.

[129] Vlasenko A., Rashkevych Y., Vlasenko N., Peleshko D., Vynokurova O. A hybrid EMD – Neuro-fuzzy model for financial time series analysis. *Proceedings of the 2020 IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP*. 2020. P. 112–115.

[130] Peleshko D., Vynokurova O., Oskerko S., Maksymiv O., Voloshyn O. (2020). Real-Time Flame Detection Using Hypotheses Generating Techniques. In: Hu Z., Petoukho S., Dychka I., He M. (eds). Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. *Advances in Intelligent Systems and Computing.* Vol. 938. Springer, Cham.

[131] Vlasenko A., Vlasenko N., Vynokurova O., Peleshko D. An Empirical Mode Decomposition Based Method to Synthesize Ensemble Multidimensional Gaussian Neuro-Fuzzy Models in Financial Forecasting. In: Babichev S., Peleshko D., Vynokurova O. (eds). Data Stream Mining & Processing. DSMP 2020. *Communications in Computer and Information Science.* 2020. Vol. 1158. Springer, Cham.

[132] Vynokurova O., Peleshko D., Zhernova P., Perova I., Kovalenko A. Solving Fraud Detection Tasks Based on Wavelet-Neuro Autoencoder. In: Babichev S., Lytvynenko V., Wójcik W., Vyshemyrskaya S. (eds) Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2020. *Advances in Intelligent Systems and Computing.* 2021. Vol. 1246. Springer, Cham. P. 535–546.

[133] Tkachenko R., Izonin I., Kryvinska N., Dronyuk I., Zub K. An approach towards increasing prediction accuracy for the recovery of missing IoT data based on the GRNN-SGTM ensemble. *Sensors (Switzerland)*. 2020. Vol. 20. Iss. 9. P. 26–25.

[134] Izonin I., Tkachenko R., Verhun V., Zub K. An approach towards missing data management using improved GRNN-SGTM ensemble method. *Engineering Science and Technology, an international journal.* 2021. Vol. 24, iss. 3. P. 749–759.

[135] Tkachenko R., Izonin I., Dronyuk I., Logoyda M., Tkachenko P. Recovery of missing sensor data with GRNN-based cascade scheme. *International Journal of Sensors, Wireless Communications and Control*. 2021. Vol. 11. № 5. P. 531–541.

[136] PuTTY. URL: https://www.putty.org/ (дата звернення: 21.06.2021).

[137] Izonin I., Tkachenko R., Shakhovska N., Lotoshynska N. The additive input-doubling method based on the SVR with nonlinear kernels: small data approach. *Symmetry*. 2021. Vol. 13. Iss. 4. 612.

[138] Вітинський П. Б., Ткаченко Р. О., Ізонін І. В. Ансамбль мереж GRNN для розв'язання задач регресії з підвищеною точністю. *Науковий вісник НЛТУ України.* 2019. Т. 29. № 8. С. 120–124.

[139] Вітинський П. Б., Ткаченко Р. О., Ізонін І. В., Кустра Н. О. Ансамблі нейроподібних структур МПГП з RBF розширенням входів для задач регресії та класифікації. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2019. № 4 (275). С. 72–79.

[140] Batyuk Anatoliy, Voityshyn Volodymyr. Process mining: applied discipline and software implementations. *Наукові вісті Національного технічного університету України «Київський політехнічний інститут»*. 2018. № 5. С. 22–36.

[141] Іванишин О. В., Батюк А. Є. Метод автоматичного екстрактивного узагальнення тексту на основі рекурентних нейронних мереж. *Наукові вісті Національного технічного університету України «Київський політехнічний інститут»*. 2018. № 4. С. 25–29.

[142] Batyuk Anatoliy, Voityshyn Volodymyr. Software architecture design of the information technology for real-time business process monitoring. *EconTechMod.* 2018. Vol. 7. № 3. P. 13–22.

[143] Batyuk A., Voityshyn V. Process mining-based information technology for operational support of software projects estimation. *Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту : матеріали міжнародної наукової конференції,* 25–29 травня 2020 р. Залізний Порт, 2020. С. 9–11.

[144] Batyuk Anatoliy, Voityshyn Volodymyr. Real-time process monitoring platform: technical implementation. *Інформація, комунікація, суспільство 2018 : матеріали 7-ої Міжнародної наукової конференції ІКС-2018*, 17–19 травня 2018 р. Чинадієво, 2018. С. 275–276.

[145] Kotsovsky V., Batyuk A., Yurchenko M. V., Mykoryak I. I. Representational capabilities of bithreshold neurons. *Інтелектуальні системи прийняття рішень*

і проблеми обчислювального інтелекту : матеріали міжнародної наукової конференції (25–29 травня 2020 р.). Залізний Порт, 2020. С. 19–20.

[146] Kotsovsky V., Batyuk A., Yurchenko M., Geche F., Mykoryak I. I. Complexity of learning bithreshold neural units. *Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту : матеріали Міжнародної наукової конференції*, 21–25 травня 2019 р. Залізний Порт, 2019. С. 92–93.

[147] C# Documentation. URL: https://learn.microsoft.com/en-us/dotnet/csharp/ (дата звернення: 13.10.2020).

[148] Introduction to C# Windows Forms Applications. URL: https://www.geeksforgeeks.org/introduction-to-c-sharp-windows-forms-applications/ (дата звернення: 25.11.2020).

[149] Visual Studio. URL: https://visualstudio.microsoft.com/ (дата звернення: 13.10.2020).

[150] C# Tutorial. URL: https://www.csharptutorial.net/ (дата звернення: 05.01.2021).

[151] Oracle Java documentation. URL: https://docs.oracle.com/javase/tutorial/ (дата звернення: 15.06.2019).

[152] Java point. URL: https://www.javatpoint.com/javafx-tutorial (дата звернення: 20.07.2019).

ЗАТВЕРДЖУЮ
Проректор з науково-педагогічної роботи
Національного університету
«Львівська політехніка»
Олег ДАВИДЧАК
«07» 09 2023 р.

АКТ

про впровадження в навчальний процес результатів дисертаційної роботи Лукащука Юрія Андрійовича

Цей акт складено про те, що результати дисертаційно роботи Лукащука Юрія Андрійовича впроваджено у навчальний процес кафедри «Автоматизованих систем управління» Національного університету «Львівська політехніка».

Впровадження результатів дисертаційної роботи полягає в їхньому використанні при викладанні навчальних дисциплін як окремих розділів лекційних курсів, так і в циклах лабораторних робіт.

Зокрема, для викладання дисципліни 6.122.00.О.112 «Технологія захисту інформації» для студентів освітньо-кваліфікаційного рівня «бакалавр», що навчаються за напрямком 122 «Комп'ютерні науки», використано так результати:

- інформаційна технологія нейроподібного захисту даних у реальному часі з симетричними ключами (коди маскування, архітектура нейромережі та матриці вагових коефіцієнтів) для смарт-систем, яка за рахунок використання моделі попередніх налаштувань, вдосконаленого методу обчислення та вагових коефіцієнтів, методу таблично-алгоритмічного обчислення скалярного добутку забезпечує високу криптографічну стійкість і апаратно-програмну реалізацію з високими техніко-експлуатаційними характеристиками.

Директор ІКНІ,
д.т.н., професор Микола МЕДИКОВСЬКИЙ

Завідувач кафедри АСУ,
д.т.н., професор Василь ТЕСЛЮК

Доцент кафедри АСУ,
к.т.н., доцент Роман МАРЦИШИН

ЗАТВЕРДЖУЮ
Проректор з наукової роботи
Національного університету
«Львівська політехніка»
д.т.н., проф. Іван ДЕМИДОВ
«__» _____ 2023 р.

АКТ
про використання результатів дисертаційної роботи
Лукащука Юрія Андрійовича
«Інформаційна технологія захисту даних у реальному часі для мобільних смарт-систем з використанням нейроподібних мереж»,
представленої на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки»,
при виконанні науково-дослідної роботи за темою:
«Експериментальна система нейромережевого криптографічного захисту та передачі даних у реальному часі з використанням баркероподібних кодів»

Комісія у складі – начальника НДЧ д.т.н., ст. досл. Романа НЕБЕСНОГО та членів: зав. відділу науково-організаційного супроводу наукових досліджень к.т.н. Галини ЛАЗЬКО, завідувача кафедри автоматизованих систем управління д.т.н., проф. Василя ТЕСЛЮКА та заст. начальника планово-фінансового відділу Ірини ФАСТ цим актом підтверджують, що результати дисертаційної роботи здобувача наукового ступеня доктора філософії Лукащука Юрія Андрійовича на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» використані при виконанні науково-дослідної роботи, яка виконувалась за рахунок коштів загального фонду державного бюджету за темою: «Експериментальна система нейромережевого криптографічного захисту та передачі даних у реальному часі з використанням баркероподібних кодів» (номер державної реєстрації 0121U109503).

Зокрема Юрієм ЛУКАЩУКОМ удосконалено метод обчислення вагових коефіцієнтів (Розділ 3. Розроблення інформаційної технології нейроподібного криптографічного захисту даних), що забезпечує швидке налаштування та обчислення матриць вагових коефіцієнтів.

Голова комісії:
Начальник науково-дослідної частини, д.т.н, ст. досл. — Роман НЕБЕСНИЙ

Члени комісії:

Зав.відділу науково-організаційного супроводу наукових досліджень — Галина ЛАЗЬКО

В.о. заступника начальника планово-фінансового відділу — Ірина ФАСТ

Завідувач кафедри автоматизованих систем управління, д.т.н, проф. — Василь ТЕСЛЮК

ЗАТВЕРДЖУЮ
Проректор з наукової роботи
Національного університету
«Львівська політехніка»
д.т.н., проф. Іван ДЕМИДОВ
«04» 09 2023 р.

АКТ
про використання результатів дисертаційної роботи
Лукащука Юрія Андрійовича
«Інформаційна технологія захисту даних у реальному часі для мобільних смарт-систем з використанням нейроподібних мереж»,
представленої на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки»,
при виконанні науково-дослідної роботи за темою:
«Експериментальна мобільна робототехнічна платформа з інтелектуальною системою управління та захистом передачі даних»

Комісія у складі – начальника НДЧ д.т.н., ст. досл. Романа НЕБЕСНОГО та членів: зав. відділу науково-організаційного супроводу наукових досліджень к.т.н. Галини ЛАЗЬКО, завідувача кафедри автоматизованих систем управління д.т.н., проф. Василя ТЕСЛЮКА та заст. начальника планово-фінансового відділу Ірини ФАСТ цим актом підтверджують, що результати дисертаційної роботи здобувача наукового ступеня доктора філософії Лукащука Юрія Андрійовича на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» використані при виконанні науково-дослідної роботи, яка виконувалась за рахунок коштів загального фонду державного бюджету за темою: «Експериментальна мобільна робототехнічна платформа з інтелектуальною системою управління та захистом передачі даних» (номер державної реєстрації 0122U000891).

Зокрема Юрієм ЛУКАЩУКОМ розроблено модель попередніх налаштувань для реалізації нейроподібного шифрування/дешифрування даних (Розділ 3. Розроблення інформаційної технології нейроподібного криптографічного захисту даних), реалізація якої забезпечує зменшення часу налаштування нейронної мережі.

Голова комісії:
Начальник науково-дослідної частини, д.т.н, ст. досл. — Роман НЕБЕСНИЙ

Члени комісії:

Зав.відділу науково-організаційного супроводу наукових досліджень — Галина ЛАЗЬКО

В.о. заступника начальника планово-фінансового відділу — Ірина ФАСТ

Завідувач кафедри автоматизованих систем управління, д.т.н, проф. — Василь ТЕСЛЮК