МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп’ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Антонюк Ігор Володимирович

Програмний засіб нейромережевого симетричного шифрування на базі базисної радіальної функції / The software for neural network symmetric encryption based on the basic radial function

напрям підготовки: 6.050102 - Комп’ютерна інженерія
фахове спрямування - Комп’ютерні системи та мережі

Бакалаврська робота

Виконав студент групи КСМ 41/1
Ігор Володимирович Антонюк

Науковий керівник:
к.т.н., доцент Якименко І.З.

2018

РЕЗЮМЕ

Дипломний проект містить 55 сторінок пояснюючої записки, 3 рисунки, 7 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою даного дипломного проекту є розроблення алгоритму нейромережевого симетричного шифрування на базі базисних радіальних мереж.

Проведено аналіз методів криптографічного захисту інформації починаючи з найдавніших часів і до сучасності. Розглянуто підходи до реалізації криптографічних методів, використовуваних на сьогоднішній день в практиці.

Вивчено можливості нейронних мереж та проведено аналіз їх структури і особливостей. Обґрунтовано вибір нейромережевої моделі для запропонованої системи шифрування, розглянуті сильні і слабкі сторони цього алгоритму.

Розроблено модель алгоритму на основі базової-радіальної мережі, оцінені можливості і властивості обраної моделі.

Розроблено методи попередньої обробки даних, складання навчальної множини і сам метод навчання радіально-базисної нейронної мережі. Здійснено програмну реалізацію алгоритму і його тестування.

Ключові слова: НЕЙРОННА МЕРЕЖА, КРИПТОГРАФІЧНИЙ ЗАХИСТ, БАЗОВО-РАДІАЛЬНА НЕЙРОННА МЕРЕЖА.

RESUME

The diploma project contains 55 pages of explanatory note, 3 figures, 7 tables, 2 appendices. Volume of graphic material 2 sheets of A3 format.

The aim of this diploma project is to develop a neural network symmetric encryption algorithm based on basic radial networks.

The analysis of methods of cryptographic protection of information from ancient times to the present is carried out. Approaches to the implementation of cryptographic methods used in practice today are considered.

The possibilities of neural networks have been studied and their structure and features have been analyzed. The choice of neural network model for the proposed encryption system is substantiated, the strengths and weaknesses of this algorithm are considered.

An algorithm model based on the base-radial network is developed, the capabilities and properties of the selected model are evaluated.

Methods of preliminary data processing, addition of a training set and the method of training of a radial-basis neural network are developed. The software implementation of the algorithm and its testing are carried out.

Keywords: NEURAL NETWORK, CRYPTOGRAPHIC PROTECTION, BASE-RADIAL NEURAL NETWORK.

ЗМІСТ

					ДП.КСМ.07096/14.00.00.000 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дат			
Розроб.		Антонюк			ПРОГРАМНИЙ ЗАСІБ НЕЙРОМЕРЕЖЕВОГО СИМЕТРИЧНОГО ШИФРУВАННЯ НА БАЗІ БАЗИСНОЇ РАДІАЛЬНОЇ ФУНКЦІЇ	Літ.	Арк. 8 / Акрушів 77
Перевір.		Якименко І.З					
Консульт		Пазлрій І.Р.				ТНЕУ. ФКІТ. КСМ – 41/1	
Н. Контр.		Гураль І.В.					
Затверд.		Березький					

ВСТУП

Криптографія в минулому використовувалася, перш за все, у військових цілях. Однак зараз, разом з формуванням інформаційного суспільства, криптографія стає одним з основних інструментів, що забезпечують конфіденційність, довіру, авторизацію, корпоративну безпеку і незліченна безліч інших важливих речей. Практичне застосування криптографії стало невід'ємною частиною життя сучасного суспільства - її використовують в таких галузях як електронна комерція, електронний документообіг (включаючи цифровий підпис), телекомунікації та інші. Дуже швидко після поширення комп'ютерів в діловій сфері практична криптографія зробила в своєму розвитку величезний стрибок, причому відразу по декількох напрямках:

– по-перше, були розроблені стійкі блокові шифри з секретним ключем, призначені для вирішення класичної завдання - забезпечення секретності і цілісності переданих або збережених даних, вони до цього часу залишаються "робочою конячкою" криптографії, найбільш часто використовуваними засобами криптографічного захисту;

– по-друге, були створені методи вирішення нових, нетрадиційних завдань сфери захисту інформації, найбільш відомими з яких є завдання підпису цифрового документа і відкритого розподілу ключів.

Історично першим завданням криптографії був захист переданих текстових повідомлень від несанкціонованого ознайомлення з їх змістом, що знайшло відображення в самій назві цієї дисципліни, цей захист базується на використанні "секретної мови", відомого тільки відправнику і одержувачу, всі методи шифрування є лише розвитком цієї філософської ідеї. З ускладненням інформаційних взаємодій в людському суспільстві виникли і продовжують виникати нові завдання по їх захисту, деякі з них були вирішені в рамках криптографії, що зажадало розвитку принципово нових

підходів і методів.

Серед усього спектру методів захисту даних від небажаного доступу особливе місце займають криптографічні методи. На відміну від інших методів, вони спираються лише на властивості самої інформації і не використовують властивості її матеріальних носіїв, особливості вузлів її обробки, передачі і зберігання.

Широке застосування комп'ютерних технологій і постійне збільшення обсягу інформаційних потоків викликає постійне зростання інтересу до криптографії. Останнім часом збільшується роль програмних засобів захисту інформації, просто модернізуються що не вимагають великих фінансових витрат в порівнянні з апаратними криптосистемами. Сучасні методи шифрування гарантують практично абсолютний захист даних, але завжди залишається проблема надійності їх реалізації.

У даній роботі розглядається алгоритм нейромережевого симетричного шифрування на базі базисних радіальних мереж (БРМ). Спочатку наводиться історія розвитку криптографії і нейронних мереж, а так само ключові етапи цього розвитку. Далі розглядається можливість злиття криптографічних та нейромережевих методів для створення нових алгоритмів шифрування. Більш докладно розглядається модель радіально базисної нейронної мережі, її сильні і слабкі сторони. І нарешті, описується підготовка і реалізація алгоритму симетричного шифрування, етапи передпідготовки тексту і навчання нейронної мережі, і аналізується стійкість алгоритму в цілому.

1 АНАЛІЗ АЛГОРИТМІВ СИМЕТРИЧНОГО ШИФРУВАННЯ НА БАЗІ БАЗИСНО-РАДІАЛЬНИХ ФУНКЦІЙ

1.1 Алгоритми симетричного шифрування і їхні особливості

Розглянемо формальне визначення криптосистеми. Криптографічна система повинна складатися з наступних компонентів [3]:

– простір вихідних повідомлень, тобто безліч рядків над деяким алфавітом;

– простір зашифрованих текстів, тобто безліч всіх можливих отриманих шифрів;

– простір ключів шифрування і простір ключів розшифрування, тобто безліч всіх можливих ключів шифрування і розшифрування;

– ефективний алгоритм генерації ключів;

– ефективний алгоритм шифрування;

– ефективний алгоритм розшифрування.

Симетричні криптосистеми відносяться до систем з секретним ключем, тобто зашифрування і розшифрування відбуваються за допомогою однакового ключа. А значить відправник повинен разом із зашифрованим повідомленням передати одержувачу і ключ.

Симетричні криптосистеми грунтуються на декількох базових класах:

– моноалфавітні і багатоалфавітні підстановки [4]. Моноалфавітні підстановки – це один з найбільш простих методів, що полягає в заміні символів даного алфавіту на інші символи того ж алфавіту по деякому певному правилу. У моноалфавітній підстановці правило залишається однаковим для кожного символу, а в багатоалфавітній при переході до нового символу правило змінюється;

– перестановки. Один з найпростіших методів шифрування, що полягає в перестановці символів алфавіту по деякому певному правилу. У

зв’язку з низькою стійкістю, цей метод використовують тільки разом з іншими, більш стійкими;

– блочні шифри. Цей метод полягає в оборотних перетвореннях блоків тексту фіксованої довжини, що по суті є системою підстановки на алфавіті блоків. Даний метод в даний час є одним з найбільш поширених;

– гамування. Даний метод є перетворенням тексту, що полягає в додаванні символів тексту по модулю, рівному потужності алфавіту, з символами деякої псевдовипадкової послідовності, отриманої за деяким законом. Через те, що псевдовипадкова послідовність може бути отримана тим же блоковим шифруванням, гамування в загальному не можна виділитися в окремий клас. Якщо ж послідовність є дійсно випадковою, отримана на основі даних, які не можна повторити, то тоді має місце випадок одноразового ключа, оскільки кожен фрагмент послідовності використовується лише один раз.

Симетричні алгоритми шифрування підстановки і перестановки вважаються класичними шифрами і лежать в основі більшості сучасних методів шифрування. При цьому без додаткового ускладнення вони є вельми нестійкими, оскільки легко піддаються криптоаналізу. Фундаментом криптоаналізу є знання розподілу частоти використання букв алфавіту, що дозволяє знайти закономірності і зламати шифр. Наприклад, відомо, що в англійських словах найчастіше зустрічається буква «е», що дозволяє, знайшовши часто вживану букву в шифрі, припустити, що це зашифрована «е».

Розглянемо найвідоміші алгоритми симетричного шифрування. Першим з них безумовно є алгоритм DES, що розшифровується Data Encryption Standard [3]. Цей алгоритм був першим схваленим міжнародною спільнотою і його модифікація були стандартами для шифрування відкритих даних (тобто не державної секретної інформації, що має відношення до національної безпеки) аж до 2002 року, коли на зміну йому прийшов AES [3].

Алгоритм DES є блоковим шифром. У цьому алгоритмі простір вихідних текстів збігається з простором зашифрованих текстів. Зокрема,

алгоритм шифрування отримує на вхід 64-бітовий текст і 56-бітовий ключ, а на виході виходить 64-бітовий зашифрований текст. Алгоритм дешифрування працює точно так само, тільки зашифрований і звичайний текст міняються місцями.

Розглянемо роботу цього алгоритму. Її можна розділити на три етапи.

1. До шифрованого блоку застосовується перестановка, яка змінює 32-бітну ліву частину блоку з 32-бітної правої. Ця перестановка є фіксованою і не несе криптографічного значення, вона потрібна для ускладнення алгоритму в цілому.

2. Потім виконується 16 раундів шифрування, кожен зі своїм ключем. У кожному раунді відбувається перестановка правої і лівої частин блоку (лівий блок, що надходить на вхід чергового раунду є правим блоком на виході попереднього раунду). При цьому лівий блок домножается на функцію підстановки, що використовує 48-бітову підстроку вихідного 56-бітового ключа. Загальний принцип можна побачити на рисунку 1.1:

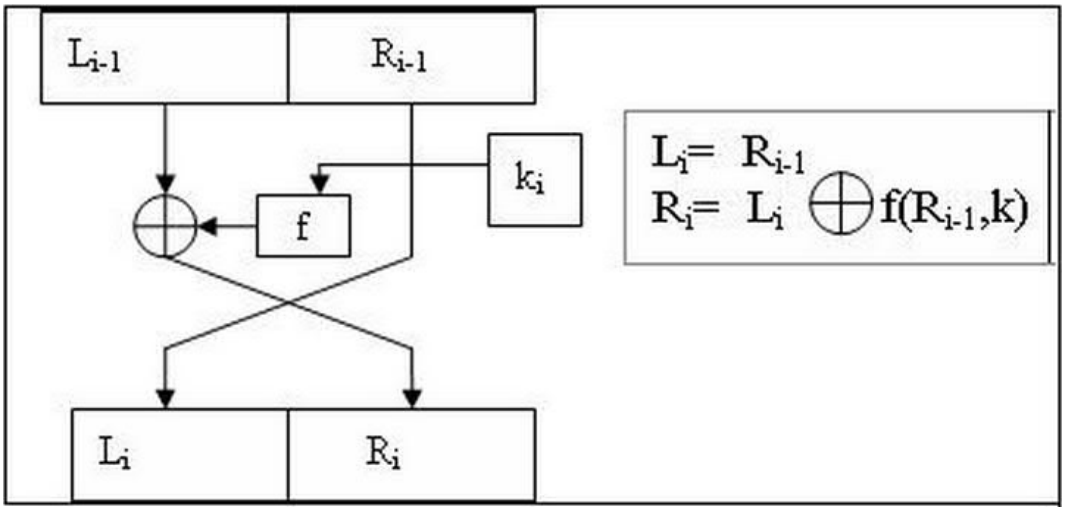


Рисунок 1.1– Загальна схема перестановок блоків при раундах шифрування (де *Li* і *Ri* – ліва і права частини блоку відповідно)

3. Далі, з результатом 16 раунду проводиться ще одна перестановка, що є зворотною до перестановки, що проводиться на першому кроці.

Ці три кроки виконуються як при зашифрованими, так і при

розшифровці. Єдиною відмінністю є порядок використання ключів, при дешифрування використовуються ключі шифрування в зворотному порядку.

Як ми бачимо, сутність алгоритму DES полягає в функції, яка застосовується до блоків в кожному раунді другого кроку.

Спори про стійкість даного алгоритму почалися майже відразу після прийняття його, як стандарту. Всі вони в загальному зводилися до того, що цей алгоритм використовує занадто короткий ключ, що є слабким місцем. Атаки на це слабке місце полягали в повному переборі ключів і називаються «атака в лоб». Однак така атака не рахується реальною і для обчислювальних потужностей 70-х років цей алгоритм є досить стійким. Але до 90-х років цей алгоритм перестав бути безпечним. У 1993 році було продемонстровано, що вартість машини, здатної зламати алгоритм DES, становить всього 1 мільйон доларів, і злом займе всього лише три з половиною години при відомій парі «вихідний текст – зашифрований текст». Далі, в 1998 році група дослідників з коаліції компаній Cryptography Research, Advanced Wireless Technology і Electronic Frontier Foundation створили машину DES-Cracker вартістю всього лише 250 тисяч доларів, якій вдалося зламати цей шифр за 56 годин [3]. Стало зрозуміло, що для обчислювальних потужностей цього часу 56-бітний ключ дійсно занадто короткий.

У 1997 році Національний інститут стандартів і технологій США почав розробку алгоритму, покликаного замінити стандарт DES. Цей алгоритм назвали AES, що розшифровується як Advanced Encryption Standard. Причому його розробка на відміну від DES була засекречена і широко висвітлювалася. У 1998 році був представлений набір з 15 алгоритмів. Відповідних на роль AES. Вони були зібрані з усього світу і вони були піддані глибокому дослідженню у всіх областях. Дослідження тривали 2 роки і в 2000 році було оголошено, що переміг алгоритм, який носив назву Rijndael, розроблений двома бельгійськими криптографами – Дааменом (Daemen) і Рійменом (Rijmen) [3].

Даний алгоритм вдає із себе так само блоковий шифр, але з однією відмінною особливістю – змінною довжиною ключа і блоків. Довжина може

приймати значення 128, 192 або 256 біт. Розглянемо цей алгоритм у дії на полегшеному прикладі, в якому використовуються розміри ключа і блоків тільки 128 біт.

Як і більшість симетричних блокових шифрів, AES складається з декількох раундів, для мінімального випадку 128 бітного числа раундів дорівнює 10, але воно може збільшуватися. Кожен раунд можна описати функцією Round (State, RoundKey), яка отримує на вхід матрицю State (в першому раунді це вихідний текст, в проміжних – його перетворення і в заключному раунді на виході State є зашифрованим текстом) і матрицю RoundKey, що є ключем раунду, за допомогою якого і відбувається перетворення змінної State. Сама функція Round складається з чотирьох підфункцій, які і виробляють перетворення. Ці функції: SubBytes (State), ShiftRows (State), MixColumns (State) і AddRoundKey (State, RoundKey). Останній раунд називається FinalRound (State, RoundKey) і збігається за своїм виконання з одним з проміжних раундів без функції MixColumns (), що робить його схожим на заключний раунд алгоритму DES. Для можливості розшифровки всі функції раундів і їх підфункції є оборотними. Розглянемо дію внутрішніх функцій кожного раунду. Всі функції визначені на кінцевому полі, що складається з поліномів по модулю даного незвідного полінома:

$$\int(x) = x^8 + x^4 + x^3 + x + 1. \qquad (2.1)$$

Блоки повідомлення і ключа в алгоритмі AES розбиті на байти. Ці байти, завдяки взаємно однозначного відображення, можна розглядати як елементи вищевказаного поля. Отже, розглянемо функції:

– SubBytes(State) – данная функция виконує підстановку каждого байта в змінній State. Відбувалося це за допомогою такого перетворення: y = $Ax^{-1} + b$ де x – це значення байта, A – деяка зворотня матриця 8x8, b – вертикальний вектор, y – отримана підстановка байта. Якщо число x є нульовим байтів, то в результаті отримаємо y = b. За рахунок оберненості

матриці A вся функція так само є оберненою;

– ShiftRows () – ця функція застосовується до матриці State через підрядник. Відбувається операція перестановки, яка для кожної і-го рядка зрушує її циклічно на n-і позицій вправо, де n – кількість рядків. Так як даний перестановочний шифр змінює лише елементи рядків, ця функція є оборотною;

– MixColumns() – дана функція застосовується до кожної колонки матриці State. Кожен стовпець у нашому випадку складається з 4 елементів (вся матриця 4x4). Стовпець перетворюється в поліном третього ступеня, де елементи шпальти – коефіцієнти полінома. Операцією з цим стовпцем є домноження отриманого полінома на деякий поліном третього ступеня по модулю $x^4 + 1$. Таке перетворення можна вважати поліалфавітною підстановкою з відомим ключем. Це перетворення так само є оборотним.

1.2 Нейроні мережі на основі радіально базисної функції

Багатошаровий персептрон, виконує апроксимацію стохастичну функцію декількома змінними шляхами перетворення множини вхідних змінних x $\in|R^N$ безліч вихідних змінних $x \in|R^N$. У такій мережі здійснюється апроксимація глобального типу, це означає, що при формуванні вихідного сигналу беруть участь вихідні сигнали багатьох або навіть усіх нейронів. Інший спосіб відображення вхідної безлічі в вихідних полягає в перетворенні заснованому на декількох одиночних аппроксимирующих функціях кожна з яких реалізує очікувані значення тільки обмеженій області багатовимірного простору. Найпростіша аналогія це апроксимація довільній одновимірної кривої сумою зважених гауссоїд (рисунок.

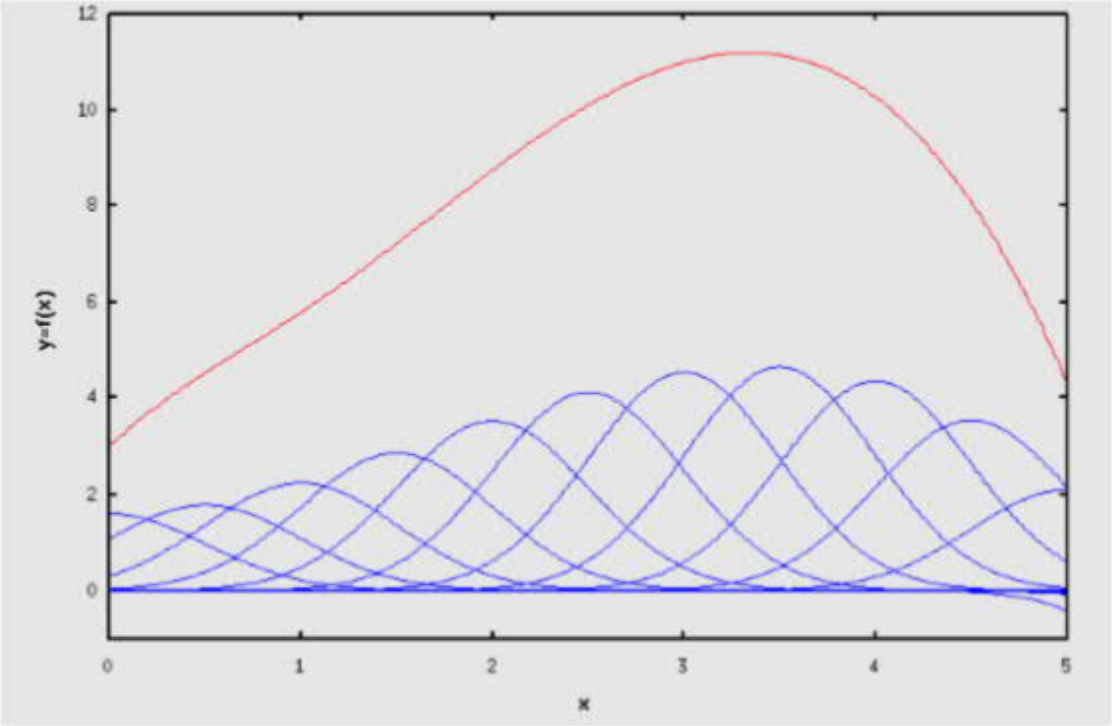


Рисунок 1.2 - Апроксимація довільній одновимірної кривої

На цьому малюнку червона крива представлена сумою синіх гауссоїд, ясно що для деякої точки x основний внесок дають лише кілька гауссоїди, центри яких близькі до цієї точки. Тому така апроксимація називається локальною. Можна сказати, що всі перетворення в цілому являє собою суму локальних перетворень на основі базисних функцій (гауссоїд в нашому прикладі).

У мережі з радіальними базисними функціями приховані нейрони реалізують функції радіально змінюючі навколо обраного центру і приймають ненульові значення тільки в околиці цього центру. Подібні функції, які визначаються у вигляді.

$$\phi_i(x) = \phi(\|x - c_i\|) \qquad (2.2)$$

називаються радіальними базисними функціями. У таких мереж роль прихованого нейрона полягає в відображенні радіального простору навколо одиночно заданої точки або навколо групи точок утворюють кластер. Суперпозиція сигналів від прихованих нейронів виконується вихідним

нейроном, дозволяє отримати відображення всієї заданої області багатовимірного простору.

Сигмоїдальний нейрон можна уявити в багатовимірному просторі гіперплоскость, який розділяє цей простір на два класи в яких виконується одна з двох умов:

$$\sum_{i=0}^{N} w_{ji} x_i > 0$$

або

$$\sum_{i=0}^{N} w_{ji} x_i < 0$$

Радіальний же нейрон розділяє простір гіперсферу навколо центральної точки і здійснює кульовий поділ простору. На відміну від багатошарового персептрона, мережа з радіальними базисними функціями має як правило тільки один прихований шар з числом нейронів істотно перевищує число входів мережі. Вихідний нейрон здійснює підсумовування сигналів, що генеруються прихованими нейронами.

Математичну основу функціонування мереж з радіальними базисними функціями (RBF-мережі) становить теорема Ковера про раздійності образів, яка стверджує що нелінійне перетворення складного завдання класифікації образів в простір більш високої розмірності підвищує ймовірність лінійної роздільності образів. Розглянемо для початка що таке роздільність образів.

Розглянемо сімейство поверхонь, кожна з яких ділить вхідний простір на 2 частини. Нехай X безліч, що складається з N образів (векторів) $x_1, x_2, \ldots x_N$, кожен з яких належить одному з 2-х класів. Така дихотомія або бінарне розподіл точок називається розділеними по відношенню до сімейства поверхонь, якщо в цьому сімействі існує поверхність, яка відокремлює точки класу x_1 від точок класу x_2. Для кожного способу $x \in X$ визначимо вектор,

що складається з безлічі дійсних значень функцій виду

$$\phi(x) = \left[\phi_1(x), \phi_2(x), \dots, \phi_{m_1}(x)\right]^T \quad (2.3)$$

Припустимо m_0 - це розмірність векторів *x*, тоді векторна функція $\phi(x)$ відображає точки m_0 мірного вхідного простору в простір розмірності m_1. Функції $\phi_i(x)$ називають прихованими, оскільки вони відіграють роль функцій активації прихованих нейронів в мережах прямого поширення. Простір утворене безліччю прихованих функцій називають прихованим простором або простором ознак.

Дихотомія $\{X_1, X_2\}$ безлічі X називається ϕ - роздільне (або ϕ - сепарабельне), якщо існує m_1 - мірний вектор *w*, для якого можна записати

$$w^T\phi(x) > 0,\ x \in X_1\ ,$$
$$w^T\phi(x) < 0,\ x \in X_2\ , \quad (2.4)$$

гіперплоскість:

$$w^T\phi(x) = 0\ , \quad (2.5)$$

описує поверхню в закритому просторі. Зворотний образ цієї поверхні:

$$\text{x: } w^T\phi(x) = 0\ , \quad (2.6)$$

розділяє поверхню у вхідному просторі. Розглянемо раціональні різноманіття r-го порядку у вхідному просторі розмірності m_0, тобто гіперповерхні описувані таким рівнянням порядку *r* в координатах вхідного вектора *x*:

$$\sum_{0 \le i_0 \le i_1 \le \cdots \le i_r \le m_0}^{K} a_{i i_{x \dots i_r}} x_{i_1} x_{i_2} \dots x_{i_r} = 0,$$

де вектор *x* доповнений як зазвичай компонентом $x_0 = 1$.

Для вхідного простору розмірності m_0 сума (4.6) містить $\frac{(m_0-r)!}{m_0!r!}$ доданків містять всілякі поєднання з r множників $x_{i_1}x_{i_2}\dots x_{i,}$ або одночленним.

У імовірнісному експерименті роздільність образів це випадкова подія залежить від обраної дихотомії і імовірнісного розподілу образів у вхідному просторі. Нехай образи x_1, x_2, ..., x_N вибираються незалежно, відповідно до імовірнісним розподілом, властивим вхідного простору, а всі можливі дихотомії $X = \{x_i\}_{i=1}^{N}$ рівноймовірно. Нехай *P(N, m_1)* ймовірність того, що випадково обрана дихотомія є ϕ - разделімого, якщо клас поділяють гіперповерхонь має m_1 ступенів свободи. Тоді

$$P(N, m_1) = \left(\frac{1}{2}\right)^{(N-1)} \sum_{m=0}^{m_1-1} \binom{N-1}{m}, \tag{2.8}$$

де $\binom{N-1}{m}$ це біноміальний коефіцієнт, який визначається за формулою

$$\binom{l}{m} = \frac{l\,(l-1)\dots(l-m+1)}{m!} \tag{2.9}$$

для всіх цілих *l* і *m*.

Рівняння (2.9) відображає сутність теореми Ковера про разделимости випадкових образів. Власне (2.9) це ймовірність того, що *(N-1)* підкидань монети призведе до випадання не більше $(m_1 - 1)$ решек. Ясно, що чим більше розмірність прихованого простору m_1, тим ближче ймовірність P (N, m_1), до одиниці. Зазначимо два основні моменти випливають з цієї теореми:

– необхідно визначити приховану функцію ϕ_i(x), i=1,2,…,m_1;

– більш висока розмірність прихованого простору в порівнянні з вхідним. Ця розмірність визначається числом прихованих нейронів m-1.

У деяких випадках однак, вдається домогтися лінійної роздільності досить нелінійного перетворення $\phi(x)$ без підвищення розмірності прихованого простору.

1.3 Постановка завдань на бакалаврську роботу

Ідея використання нейронних мереж для шифрування даних запозичена з методів інтелектуального аналізу даних. Особливість цих методів полягає в тому, що відсутні будь-які рамки і обмеження оброблюваної вибірки і розкиду значень аналізованих показників. Саме тому, за рахунок здатності до моделювання нелінійних процесів, адаптивності, уміння витягувати суттєві особливості з даних, що надходять, нейронні мережі найкращим чином відповідають для цього.

Як відомо, алгоритм шифрування складається з ключа шифрування і розшифровки, і алгоритмів шифрування і розшифровки. Розглянемо, як це співвідноситься з використанням нейронних мереж. Щоб використовувати мережу, треба задати її структурні характеристики - кількість шарів, кількість нейронів в шарі, функції активації, коефіцієнти. Значення цих характеристик стають ще одним ключем для шифрування і задають алгоритми шифрування і розшифровки. Ці значення залежать не тільки від специфіки обраної нейронної мережі, але і від особливостей коду: його довжини, кількості кодованих символів і т.д. Алгоритм розшифровки полягає в розпізнаванні інформації: на вхід нейронної мережі подається зашифрований текст, і на виході виходить розшифроване повідомлення. Алгоритм шифрування більш складний і заснований на пошуку спотвореного коду, який може бути відновлений нейронною мережею.

У даній роботі пропонується симетричний метод шифрування на базі

радіально-базисної нейронної мережі, що відноситься до методів шифрування на основі хаотичних полів. Математично ця задача зводиться до задачі навчання базисно-радіальної нейронної мережі для подальшої класифікації зашифрованих символів за відповідними їм кластерам.

Виходячи з вище сказаного метою дипломної робити є розробка методу шифрования великих масивів даних за допомогою нейромережевих парадигм і проаналізувати можливості, які відкриває використання нейронних мереж в криптографії, а також розробити, підготувати і навчити нейронну мережу для розшифрування даних.

Для цього необхідно вирішити низку взаємопов'язаних завдань:

– провести аналіз алгоритмів симетричного шиврування і вказати особливості їхнього застосування;

– проаналізувати можливості застосування радіальної базисної функції в задачах захисту інформації;

– побудувати структуру базисно радіальної нейроної мережі;

– розробити математично формалізацію нейроної мережі;

– розробити алгоритм шифрування даних на основі базисно радиальної функції;

– обгрунтувати вибір середовища програмування;

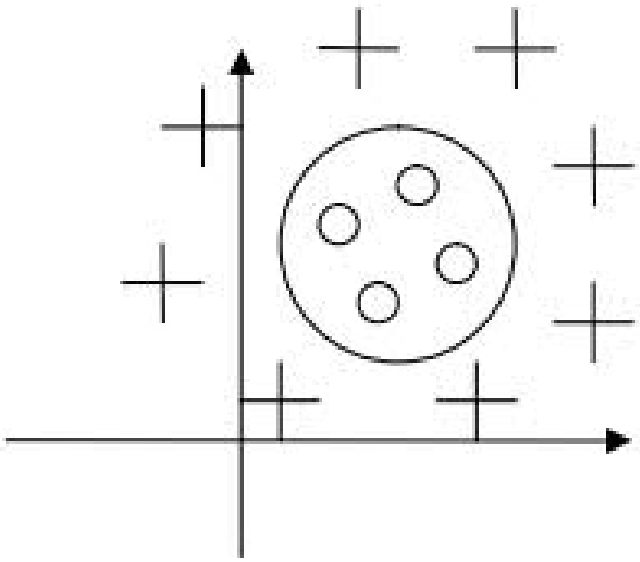
– провести навчання нейронної мережі для пошуку зашифрованого тексту та його дешифруння;

– провести програму реалізацію алгоритму шифрування даних на основі базисно радиальної функції.

Вирішення даного класу завдань дозволить досягнути мету дипломної роботи.

2 МАТЕМАТИЧНА МОДЕЛЬ НЕЙРОННОЇ МЕРЕЖІ НА БАЗІ БАЗИСНО-РАДІАЛЬНОЇ ФУНКЦІЇ

У даній роботі пропонується використовувати нейронну мережу на основі базисно-радіальних функцій [1,5]. У цьому типі нейронної мережі нейрони прихованого шару використовують функції, радіально змінюють свої значення в околицях деякого центру, і мають значення більше нуля тільки в деякій околиці цього центру. Нейрони радіально-базисної нейронної мережі подаються у вигляді гіперсфери в багатовимірному просторі (рисунок 2.1), що розділяє цей простір на два класи, для одного з яких виконується $\sum_j w_{ij} x_j > 0$, а для іншого $\sum_j w_{ij} x_j < 0$.



$$\sum_j w_{ij} x_j > 0, \sum_j w_{ij} x_j < 0$$

Рисунок 2.1 – Схема гіперсфери, і накладаються обмеження

Завдяки цьому можна істотно зменшити кількість нейронів прихованого шару, необхідних для поділу простору на класи. Відповідно ці нейрони можна розташувати у вигляді одного прихованого шару, чим дуже полегшується

завдання конструювання нейронної мережі завдяки тому, що розробнику не потрібно вирішувати питання про кількість шарів прихованого шару.

2.1 Структура базисної радіальної нейроної мережі

Розглянемо структуру нейронної мережі на основі радіально-базисних функцій. Радіально-базисна нейронна мережа містить три шари: вхідний шар, що виконує обробку вхідних даних; прихований шар, що складається з нейронів з радіально-базисної функцією, в яких відбувається обчислення значень на основі обробленого вхідного сигналу; вихідний шар, який здійснює підсумовування виходів нейронів прихованого шару. Структура радіально-базисної нейронної мережі приведена на (рисунку 2.2).

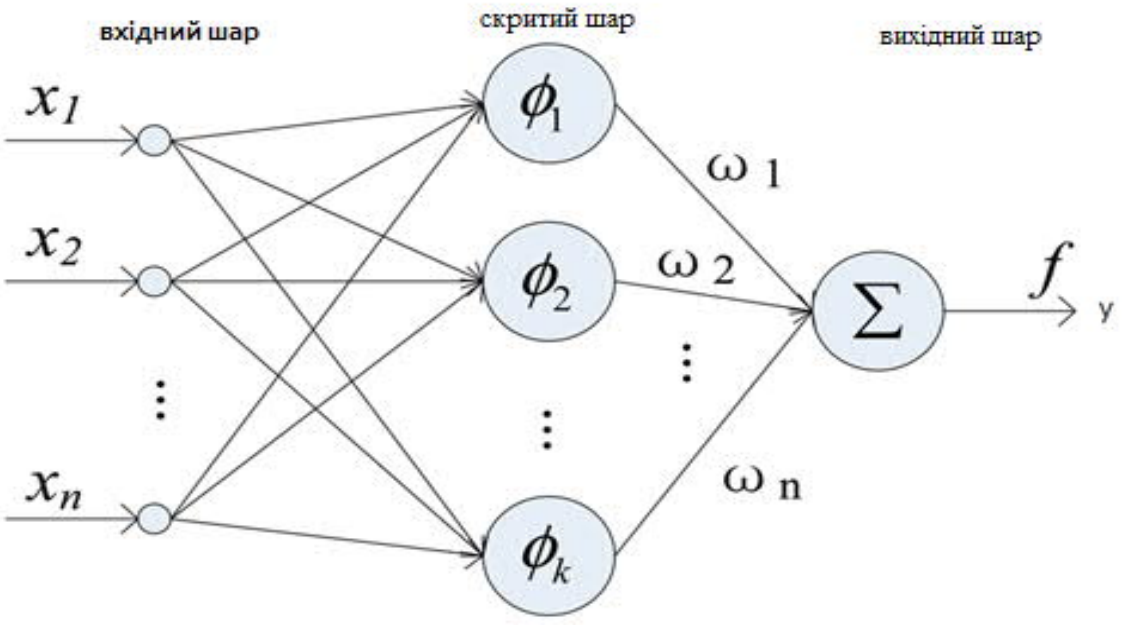


Рисунок 2.2 – Структура радіально-базисної нейронної мережі
(де x_i - входи мережі, y - вихід мережі, ϕ_i- функція активації, ω_i- ваги)

Функція перетворення нейронної мережі виглядає наступним чином:

$$F(x_p) = \sum_{i=1}^{K} \omega_i \phi(\|x_p - c_i\|), \quad (2.1)$$

де c_i - центри в функції Гаусса,

x_p - елементи навчальної вибірки від 1 до р.

Архітектура радіально-базисних нейронних мереж визначається тим, що вони складаються з трьох шарів, що виконують зовсім різні функції. Перший, вхідний, шар складається з вхідних елементів, що пов'язують нейронну мережу з зовнішнім середовищем. Наступний шар є єдиним «прихованим» і нелінійно перетворює вхідний простір в приховане, що має велику розмірність. Доцільність цього обумовлюється тим, що завдання класифікації в просторі вищої розмірності з більшою ймовірністю задовольняє потреби лінійної роздільності. І останній, вихідний шар, підсумовує обчислені нейронами приховані шари значення для подання їх на виходи нейронної мережі.

2.2 Побудова нейронної мережі

Побудова даного виду нейронної мережі розглядається як задача апроксимації кривої по точках в просторі високої розмірності. В основі роботи радіально-базисної нейронної мережі лежить традиційна інтерполяція в багатовимірному просторі. У термінології нейронних мереж, приховані нейрони містять набір функцій, які є «базисом» для розкладання вхідних значень «векторів».

На вхід нейронної мережі в даній роботі буде подаватися двійковий код символу при зашифровці, або вектор зашифрованого символу при розшифровці. Передобробка даних полягає в розподілі символів по

кластерам,

що залежать від частоти входження символу в шифрований текст або частоти використання символу в заданому алфавіті.

За передобробці даних і формуванням навчальної вибірки слід побудувати нейронну мережу. Формується її структура, шукаються значення вагових коефіцієнтів, проводиться навчання.

Використання різних типів нейронних мереж задає свої індивідуальні особливості, але головне - навчити нейронну мережу розпізнаванню спотвореного коду з областей, отриманих на попередньому етапі. Фактично, створюється ключ шифрування, який представляє з себе інформацію про тип нейромережі і значеннях її структурних характеристик, таких як кількість шарів і нейронів у них і вагові коефіцієнти.

Нейронні мережі на основі радіальних базисних функцій засновані на розбитті простору гіперсфери. Розглядаючи завдання апроксимації, приховані нейрони радіально-базисної мережі утворюють сукупність функцій, які становлять базис, який представляє вхідні дані в побудованому на ній просторі. Через це такі мережі не можуть виходити за рамки того, що відомо на основі навчальних даних, і при видаленні від них відгук досить швидко падає до нуля. Нейрони радіально-базисної мережі діють в малій області побудованого простору, і при великих відстанях між вхідними даними потрібна велика кількість нейронів [1].

При побудові радіально-базисної нейронної мережі як центри функцій прихованих нейронів зручно вибирати центри областей спотворення символів. У цьому випадку важливо вибрати ширину функції активації.

Так само, можна співвідносити деякі групи нейронів мережі з кожним символом, дозволяючи задавати вельми складні області спотворень цих символів. Однак потрібно прагнути до рівномірного розподілу символів в просторі шифрування, грунтуючись на частотному аналізі.

Значення вагових коефіцієнтів лінійного вихідного шару радіально-базисної нейронної мережі можна задавати як за допомогою навчання, так і

вручну. Зокрема, можна співвіднести кожен нейрон з конкретним символом, задавши значення вагового коефіцієнта, який зв'язує прихований нейрон з нейроном вихідного шару, що дорівнює одиниці, а всі інші нулю.

2.2.1 Математична формалізація нейроної мережі

Теоретичну основу підходу побудови радіально-базисних нейронних мереж являє теорема Ковера про роздільні множини. Вона стверджує наступне: «Нелінійне перетворення складного завдання класифікації образів в простір більш високої розмірності підвищує ймовірність лінійної роздільності образів.»

Дана теорема базується на двох моментах:

1. Задання нелінійної прихованої функції $\phi(x)$:

де x - вхідний вектор, $i = 1 \ldots m_1$,m_1 - розмірність прихованого простору.

2. Велика розмірність прихованого простору, в порівнянні з вхідним. Розмірність задається значенням m_1, що дорівнює кількості прихованих нейронів.

Ковер довів, що будь-яка множина образів, розміщених на деякому просторі, тим більше роздільна, чим більше це простір в порівнянні з початковим простором цих образів. Якщо задати, як ймовірність роздільності, то:

$$P(N, m_1) = \left(\frac{1}{2}\right)^{N-1} \sum_{m=0}^{m_1-1} \binom{N-1}{m}, \quad (2.2)$$

де $\binom{N-1}{m}$ визначається формулою:

$$\binom{l}{m} = \frac{l(l-1) \ldots (l-m+1)}{m!}. \quad (2.3)$$

Так само з теореми Ковера можна сформулювати, що якщо вектор

радіальних функцій в N-вимірному вхідному просторі позначити $\phi(x)$, то це простір є нелінійним ϕ-разделений на два просторових класи $X + i\ X -$ тоді, коли існує такий вектор ваг ω, такий що:

$$\omega^T \phi(x) < 0, x \epsilon X -$$
$$\omega^T \phi(x) > 0, x \epsilon X +, \quad (2.4)$$

причому межа між цими класами визначається рівнянням:

$$\omega^T \phi(x) = 0. \quad (2.5)$$

Збільшення розмірності прихованого шару дозволяє збільшити ймовірність лінійної роздільності, однак для деяких випадків досить простого нелінійного перетворення без збільшення розмірності. Найчастіше при перетворенні Т різних векторів необхідно використовувати Т нейронів прихованого шару. За рахунок цього в контексті завдання розробки алгоритму симетричного шифрування, слід використовувати лише один нейрон прихованого прошарку, як є кластерів для шифрованих символів.

Як функції активації в радіально-базисної нейронної мережі найчастіше використовують функцію Гаусса:

$$\phi = \exp\left(-\frac{r^2}{2\upsilon^2}\right), \upsilon > 0, r\epsilon|R, \quad (2.6)$$

де $r = x\text{-}c_i$ - відстань від даної точки до центру c_i гіпершару;

σ - ширина функції.

Центр гіпершару і значення σ визначаються в процесі навчання мережі. Радіальна мережа з функцією Гаусса виглядає так (рисунок 2.3):

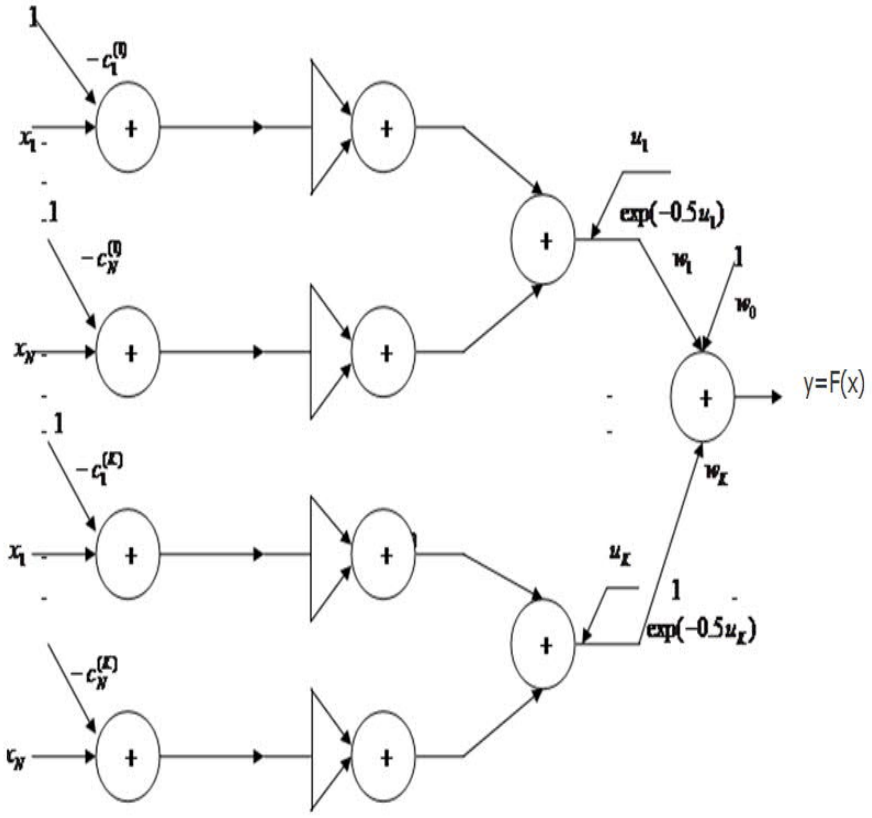


Рисунок 2.3 – Радіально-базисна нейронна мережа на основі функцій Гаусса

Кожен нейрон має свою власну функцію активації, із заданим центром і шириною. При надходженні вхідного вектора на нейрони вхідного шару, кожен нейрон прихованого шару обчислює значення функції активації, отримуючи таким чином відстань від даного вектора до центра кластера, заданого в цьому нейроні. Таким чином можна визначати приналежність вектора до кластера, або ж вибирати випадковий вектор з потрібного кластеру.

2.3 Алгоритм шифруваня RSA на основі радиальних базисних нейромережевих систем

В світі комунікацій, запевняють, що:

1. Повідомлення захищено від неавторизованих осіб, читаючи інформацію, яка, як вважається, повинна зберігатись у приватному порядку.

2. Повідомлення не випадково або навмисно не змінюється під час транзиту шляхом заміни, вставки або видалення.

3. Повідомлення надходить від джерела, з якого він стверджує, що воно прийде. Наша модель заснована на кількох механізмах, такі як, штучні нейронні мережі (ANN), коди автентифікації повідомлень (MAC), алгоритм шифрування RSA.

Для побудови алгоритму шифрування в нашій схемі ми використовували радіальні базисні функції нейронних мереж.

2.3.1 Коди автентифікації повідомлень (MAC)

MAC - це метод, який передбачає використання секретного ключа для створення невеликого блоку фіксованих розмірів, відомого як криптографічна контрольна сума або MAC, який додається до повідомлення. Це забезпечує спосіб перевірки цілісності переданої інформації серед сторін зв'язку.

Для нашої демонстрації ми використовували механізм CBC-MAC [16], як показано на (рисунку 2.4).

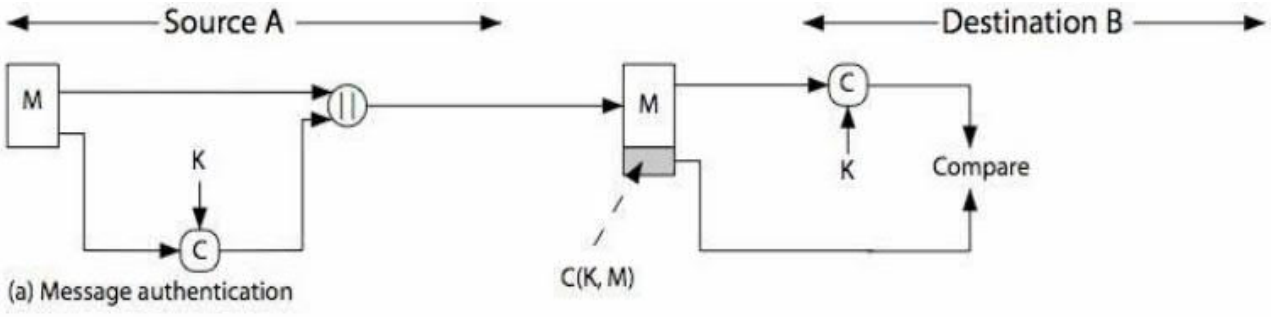


Рисунок 2.4 – Використання MAC

RSA - це алгоритм шифрування з відкритим ключем [14], який використовується тут для підписання величини MAC шляхом шифрування

його закритим ключем відправника, комбінацією відкритого ключа методу шифрування і MAC сформований механізм для забезпечення цифрового підпису пропонованої схеми.

Цифровий підпис визначається як зашифрований дайджест повідомлень, закритий ключ відправника, доданий до повідомлення, яке потім надсилається на приймач. Цифровий підпис дозволяє одержувачу інформації перевіряти справжність джерела інформації, а також перевіряти, чи ця інформація є недоторканою. Таким чином, він забезпечує автентифікацію та цілісність даних.

2.3.2 Пропонована схема

Пропоновану схему можна підсумувати на наступних етапах. На стороні відправника необхідно виконати кілька етапів:

– стадія генерації ключів:

на цьому етапі створюються і видаються чотири види секретних ключів між сторонами спілкування, як показано на рисунку нижче:

1) використовуючи алгоритм RSA, для кожного користувача комунікації, приватного ключа (d, n_1), який виключно обмежується користувачем, і відкритого ключа (e, n_1), який публікується серед інших членів спілкування;

2) інший ключ k_{mac}, генерується для використання в обчисленні MAC значення вихідного повідомлення;

3) нарешті, інший ключ k_{mac} , призначений для використання в алгоритмі шифрування (моделі нейронної мережі);

– вироблення схеми цифрового підпису:

1) цей етап складається з двох фаз, перша з яких виконується шляхом обчислення значення CBC-MAC повідомлення M після поділу його на набір блоків з фіксованим розміром (m):

$$M = \{ b_1, b_2, .., b_n \}, \quad (2.7)$$

де n - кількість блоків.

Кожен блок складається з *m* елементів:

$$b_i = \{ x_1, x_2, .., x_m \}. \quad (2.8)$$

2) по-друге, підписується значення MAC, шифруючи його за допомогою приватного ключа відправника (d, n_1) для отримання підписаного MAC(MAC'):

$$MAC' = MAC^{\,d} \bmod n_1. \quad (2.9)$$

– етап шифрування:

1) блоки зашифруються таким чином:

для кожного блоку b_i в повідомленні тренується нейронна мережева модель на основі використання РБФ, поставивши b_i як ціль та ключовий k_{mac} як вхід до цієї моделі. Після завершення тренінгу згенерований набір величин ваг і відхилень від моделі віртуальної мережі зберігаються щоразу.

$$W = \{w_1, w_2 ..., w_n\}. \quad (2.10)$$

За поколінням значення (W, MAC′) цього етапу йдуть до кінця, і ці значення надсилаються у вигляді групи для приймача. На ДП.КСМ. 07096/14.00.00.000.С0 показані кроки на стороні відправника.

На стороні приймача робляться інші кроки для розшифровки зашифрованого тексту та отримання оригінального повідомлення, крім забезпечення автентичності та цілісності повідомлень за наступними етапами:

– етап розшифрування:

1) використовуючи отримані ваги W як маси з'єднання на моделі нейронної мережі РБФ, а також ключ k_{mac} як вхід до цієї моделі. Після тренування моделі блоки повідомлень одержуть одне за одним як виходи цієї моделі. Будується повідомлення M з отриманих блоків, як у рівнянні (2.9);

– етап перевірки:

1) обчислюється значення MAC отриманого повідомлення (Obt-MAC);

2) розшифровується отримане підписане MAC-значення MAC' за відкритим ключем відправника для перевірки ідентичності:

$$\text{MAC} = (\text{MAC}')^{e} \bmod n_1 . \qquad (2.11)$$

3) порівнюється отримане значення MAC з рівняння (2.11) з значенням MAC побудованого повідомлення Obt-MAC. Якщо отримане узгодження (MAC = Obt-MAC), то це вказує на те, що повідомлення не змінилося в процесі передачі між сторонами зв'язку. На ДП.КСМ. 07096/14.00.00.000.С1 показані кроки на стороні приймача.

3 РЕАЛІЗАЦІЯ АЛГОРИТМІВ ШИФРУВАННЯ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ РАДИАЛЬНИХ БАЗИСНИХ ФУНКЦІЙ

3.1 Обгрунтування середовища програмування

Microsoft Visual Studio - це програмна средапоразработке додатків для ОС Windows, як консольних, так і з графічним інтерфейсом.

У комплект входять наступні основні компоненти:

1. Visual Basic.NET - для розробки додатків на VisualBasic;

2. Visual C ++ - на традиційній мові C ++;

3. Visual C # - наязике C # (Microsoft);

4. Visual F # - на F # (Microsoft Developer Division).

Функціональна структура середовища включає в себе:

– редактор вихідного коду, який включає безліч додаткових функцій, як автодоповнення IntelliSense, рефракторінг коду і т. д .;

– відладчик коду;

– редактор форм, призначений для спрощеного конструювання графічних інтерфейсів;

– веб-редактор;

– дизайнер класів;

– дизайнер схем баз даних.

Visual Studio також дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (Subversion і VisualSourceSafe), додавання нових наборів інструментів (для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування або інструментів для інших аспектів процесу розробки програмного забезпечення).

Комерційні версії в порядку зростання ціни: Visual Studio Professional, Visual Studio Premium і Visual Studio Ultimate.

Інтегроване середовище розробки (IntegratedDevelopmentEnvironment - IDE) Visual Studio пропонує ряд високорівневих функціональних можливостей, які виходять за рамки базового управління кодом.

Нижче перераховані основні переваги IDE-середовища Visual Studio.

Вбудований Web-сервер. Для обслуговування Web-додатки ASP.NET необхідний Web-сервер, який буде очікувати Web-запити і обробляти відповідні сторінки. Наявність в Visual Studio інтегрованого Web-сервера дозволяє запускати Web-сайт прямо з середовища проектування, а також підвищує безпеку, виключаючи ймовірність отримання доступу до тестового Web-сайту з якого-небудь зовнішнього комп'ютера, оскільки тестовий сервер може приймати з'єднання лише з локального комп'ютера.

Підтримка безлічі мов при розробці. Visual Studio дозволяє писати код своєю рідною мовою чи будь-яких інших бажаних мовами, використовуючи весь час один і той же інтерфейс (IDE). Більш того, Visual Studio також ще дозволяє створювати Web-сторінки на різних мовах, але поміщати їх все в один і той же Web-додаток. Єдиним обмеженням є те, що в кожній Web-сторінці можна використовувати тільки якусь один мову (очевидно, що в іншому випадку проблем при компіляції було б просто не уникнути).

Менше коду для написання. Для створення більшості додатків потрібно пристойну кількість стандартного стереотипного коду, і Web-сторінки ASP. NET тому не виключення. Наприклад, додавання Web-елемента управління, приєднання обробників подій і коригування форматування вимагає установки в розмітці сторінки ряду деталей. У Visual Studio такі деталі встановлюються автоматично.

Інтуїтивний стиль кодування. За замовчуванням Visual Studio форматує код у міру його введення, автоматично вставляючи необхідні відступи і застосовуючи колірне кодування для виділення елементів типу коментарів. Такі незначні відмінності роблять код більш зручним для читання і менш схильним до помилок. Застосовувані Visual Studio автоматично параметри форматування можна навіть налаштовувати, що дуже зручно у випадках, коли розробник вважає за краще інший стиль розміщення дужок (наприклад,

стиль K & R, при якому відкриває дужка розміщується на тому самому рядку, що і оголошення, якому вона передує).

Більш висока швидкість розробки. Багато з функціональних можливостей Visual Studio спрямовані на те, щоб допомагати розробнику робити свою роботу якомога швидше. Зручні функції, на зразок функції IntelliSense (яка вміє перехоплювати помилки і пропонувати правильні варіанти), функції пошуку і заміни (яка дозволяє відшукувати ключові слова як в одному файлі, так і в усьому проекті) і функції автоматичного додавання і видалення коментарів (яка може тимчасово приховувати блоки коду), дозволяють розробнику працювати швидко і ефективно.

Можливості налагодження. Пропоновані в Visual Studio інструменти налагодження є найкращим засобом для відстеження загадкових помилок і діагностування дивної поведінки. Розробник може виконувати свій код по рядку за раз, встановлювати інтелектуальні точки переривання, при бажанні зберігаючи їх для використання в майбутньому, і в будь-який час переглядати поточну інформацію з пам'яті.

Visual Studio також має і безліч інших функцій: можливість управління проектом; вбудована функція управління вихідним кодом; можливість рефакторізації коду; потужна модель розширюваності. Більш того, в разі використання Visual Studio 2008 Team System розробник отримує розширені можливості для модульного тестування, спільної роботи і управління версіями коду (що значно більше того, що пропонується в більш простих інструментах на кшталт Visual SourceSafe).

Як недолік можна відзначити неможливість відладчика (Microsoft Visual Studio Debugger) відстежувати в коді режиму ядра. Налагодження в Windows в режимі ядра в загальному випадку виконується при використанні WinDbg, KD або SoftICE.

3.2 Алгоритм симетричного шифрування даних

Реалізація алгоритму шифрування складається з декількох основних етапів:

– на попередньому етапі, під час якого здійснюється попередня обробка даних і потім формується навчальна вибірка з урахуванням частоти;

– появи символів вихідного алфавіту;

– далі йде побудова самої нейронної мережі - Виберіть тип використовуваної нейронної мережі, визначення її структури та навчання (одержання значень вагових коефіцієнтів);

– і нарешті, основний етап, на якому відбувається сам процес шифрування.

Перші два етапи обов'язково повинні передувати етапу шифрування в перший раз створення і навчання нейронної мережі, але після отримання навченої мережі вони не потрібні, і нейронна мережа може виробляти шифрування багаторазово.

На (рисунку 3.1) показана основна схема роботи алгоритму:

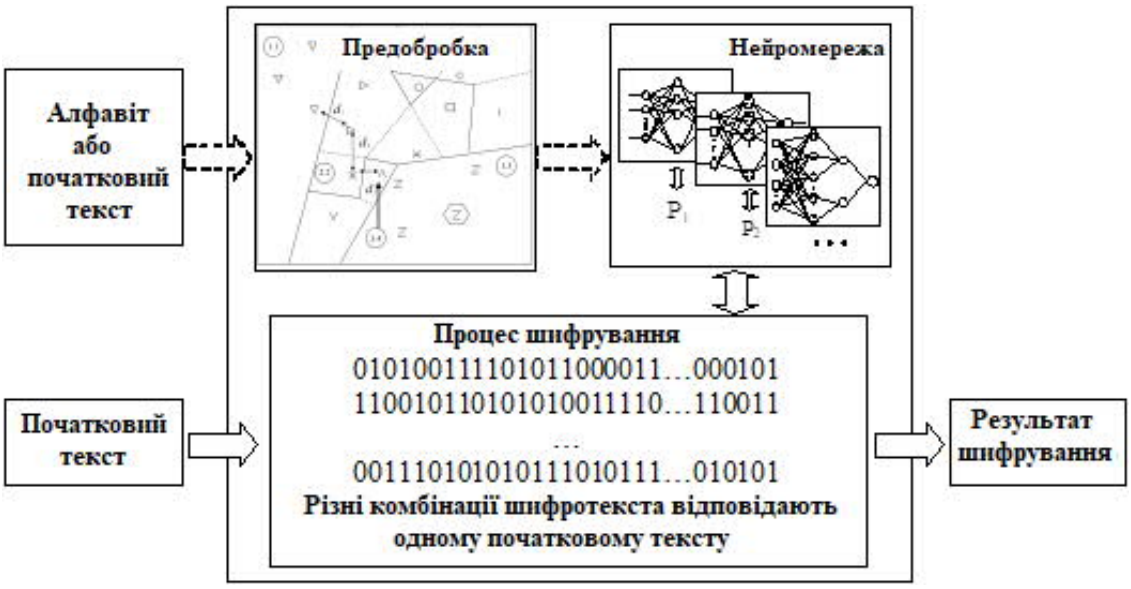


Рисунок 3.1 – Загальна схема роботи алгоритму шифрування

Розглянемо основні етапи при створенні навчальної вибірки використовується інформація про частоту появи символів з вихідного

алфавіту або вихідного тексту $\alpha(n_1 \dots n_t)$, і розраховується частота використання кожного символу в шифрованому тексті, де - кількість різних символів, n_1 - число використання 1-го символу в тексті.

Далі відбувається формування областей простору шифрування, в яких буде відбуватися спотворення символу. Це включає в себе кілька кроків:

1. Для кожного символу генерується кількість областей з урахуванням частоти входження символів, і виходить вектор кількості областей $\gamma(m_1 \dots m_1)$, де t - кількість різних символів, m_1 - число областей для кожного класу.

2. Далі випадково розподіляються області для кожного символа, в залежності від частоти повторюваності символу кількість областей буде різна. Області задаються у вигляді навчальної вибірки - набору близько розташованих векторів із зазначенням їх класу. Важливо, що нові згенеровані вектора порівнюються з вже існуючими. Це гарантує, що кожен вектор буде належати єдиному класу.

Як приклад розглянемо таке шифроване повідомлення: «jerome k. jerome. three men in a boat (to say nothing of the dog). there were four of us - george, and william samuel harris, and myself, and montmorency». Очевидно, що частота використання символів не рівномірна. Наприклад пробіл зустрічається 28 разів, «e» - 17 разів, символи «k, b, c, (,)» - по одному разу. Розглянемо основні статистичні показники тексту:

– загальна кількість символів в повідомленні: 154;

– розмір алфавіту: 27;

– середнє значення частоти появи символу: 0,037;

– мінімальне значення частоти появи символу: 0,0065;

– максимальне значення частоти появи символу: 0,1818.

На (рисунку 3.2) показано порівняння частоти використання символів у вхідній послідовності і розподілу точок по простору шифрування з відповідними класами вхідної послідовності.

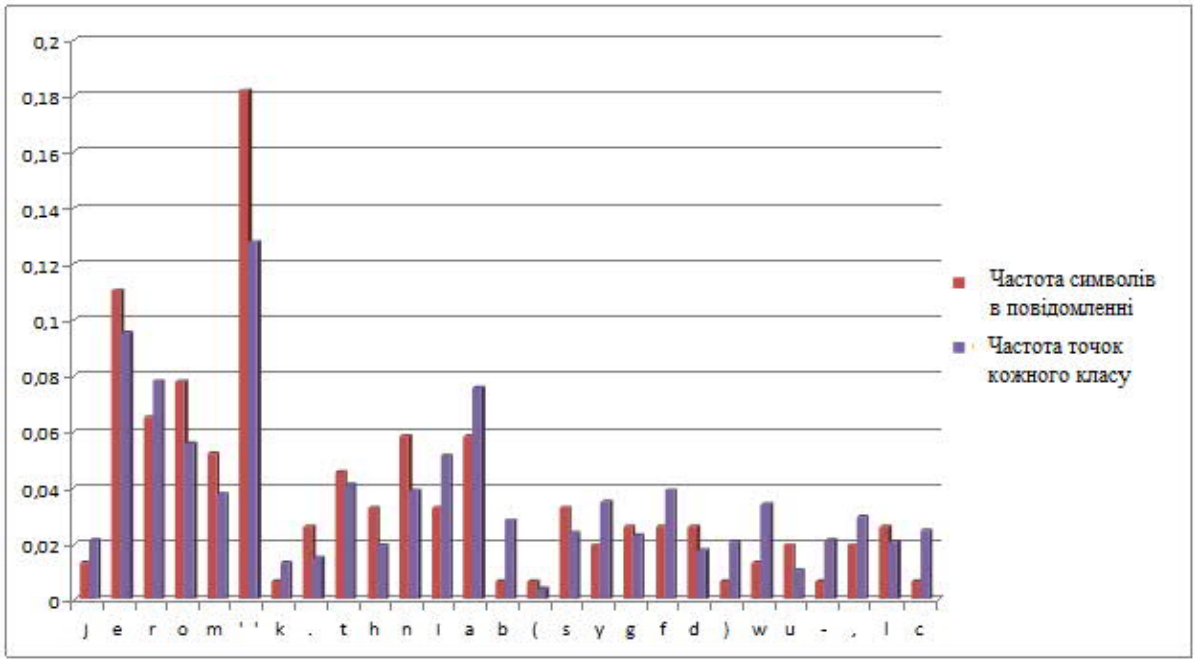


Рисунок 3.2 – Порівняння частотності символів в повідомленні і точок в кластерах

Видно, що вони дуже близькі. Це гарантує рівномірний розподіл символів по області шифрування і робить скрутним застосування до даного алгоритму методів частотного криптоаналізу.

3.3 Експериментальні дослідження реалізації алгоритму шифруваня RSA на основі радиальних базисних нейромережевих систем

Для оцінки ефективності запропонованої схеми проведено наступний експеримент для вимірювання рівня плутанини та дифузії, шляхом порівняння відкритого тексту та шифру як метричної моделі для забезпечення безпеки. Ці експерименти з моделювання були виконані на реченні M, що представляє оригінальне повідомлення:

M = "Криптографія - це наука про відкрите таємне писання", щоб зашифрувати це повідомлення за запропонованою моделлю, ми використовували його як ціль для моделі нейронної мережі, а ключ *k* - як

вхідний. Припустимо, що k_{mac} є "abcde". На (рисунку 3.3) представлений інтерфейс програмної реалізації шифрування, а на (рисунку 3.4) дешифрування:

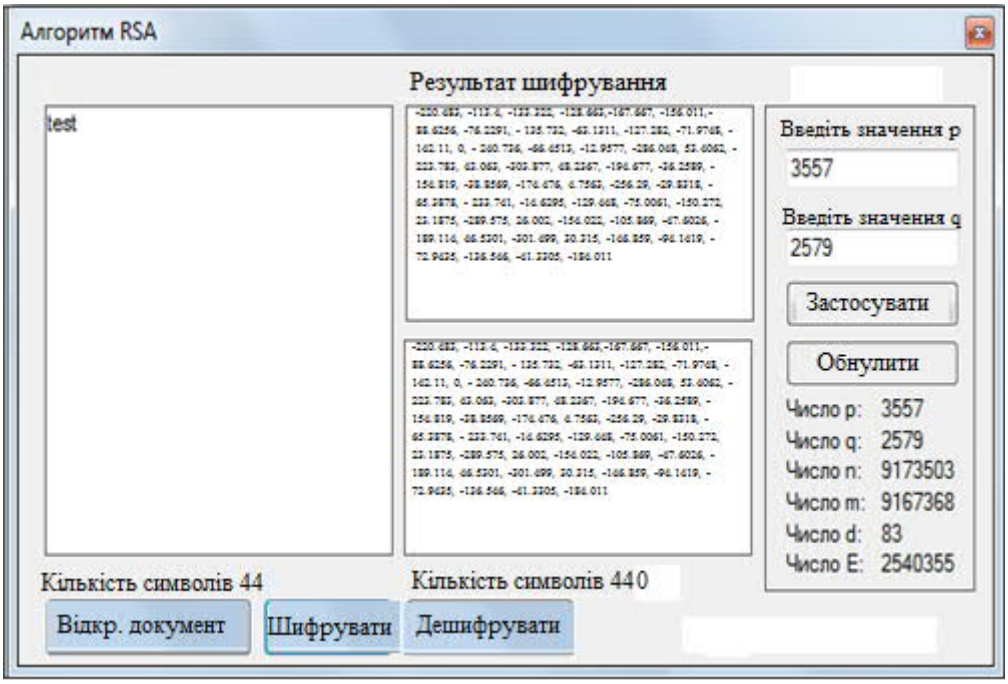


Рисунок 3.3 – Отримання кодованого тексту

Розшифрування кодів проходить після натискання кнопки «Дешифрувати».

Наведений зашифрований текст був таким: {-220.483, -113.4, -133.322, -128.663, -167.667, -156.011, -88.6256, -76.2291, - 135.732, -63.1311, -127.282, -71.9748, -142.11, 0, - 240.736, -66.4513, -12.9577, -286.048, 53.4062, -223.783, 43.063, -303.877, 48.2367, -194.677, -36.2589, -154.819, -38.8569, -174.476, 4.7563, -256.29, -29.8318, -65.3878, - 233.741, -14.6295, -129.448, -75.0061, -150.272, 23.1875, -289.575, 26.002, -154.022, -105.869, -47.6026, -189.114, 46.5301, -301.499, 30.315, -146.859, -94.1419, -72.9435, -136.546, -41.3305, -184.011}.

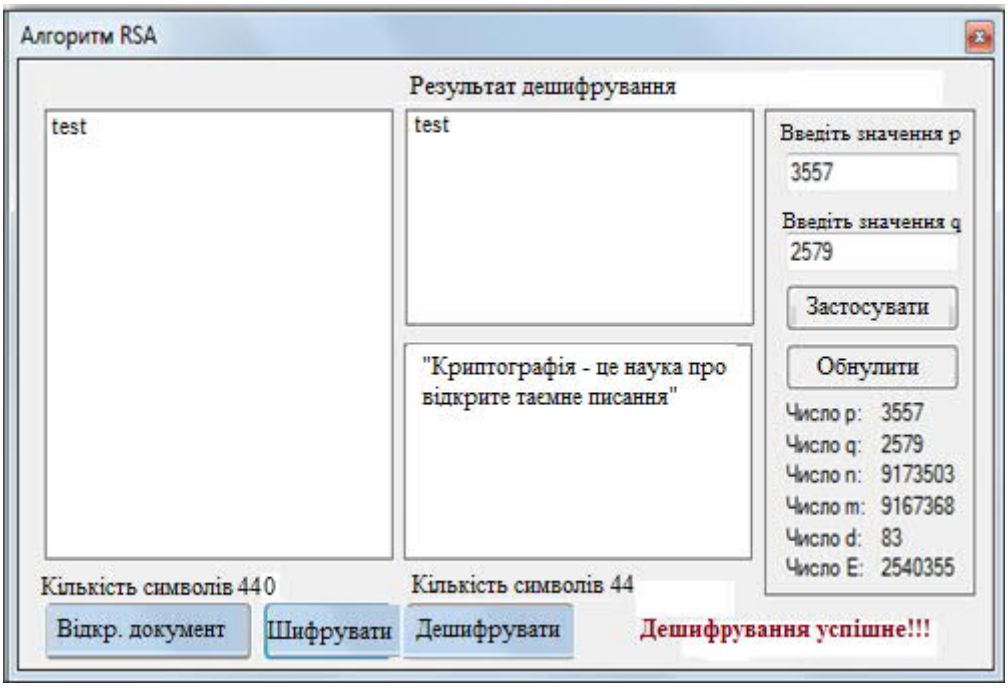


Рисунок 3.4 – Розшифрування повідомлення

Контраст між відкритим текстом та шифрованим текстом показаний на (рисунку 3.5).

З приведеного шифрувального тексту ми можемо чітко помітити, що для кожного символа в оригінальному повідомленні в шифрованому тексті з'являється інше значення, і немає прямого зв'язку між відкритим текстом (виходом моделі) та текстом шифру (вагами). Це вказує на те, що пропонована модель має велику плутанину, оскільки взаємозв'язок між входом (ключем) та виходом (повідомленням) є нелінійним, оскільки він залежить від природи нейронної мережі.

Також ми помітили, що повідомлення містить кілька повторюваних символів, таких як символ "e", наприклад (повторюється 3 рази), і кожен раз, коли приведений шифр відрізняється від іншого, повторювані значення зникають у наведеному шифрованому тексті. На (рисунку 3.6) показано розподіл "e" символу в шифрованому тексті. Це вказує на те, що запропонована модель забезпечує високий рівень дифузії.

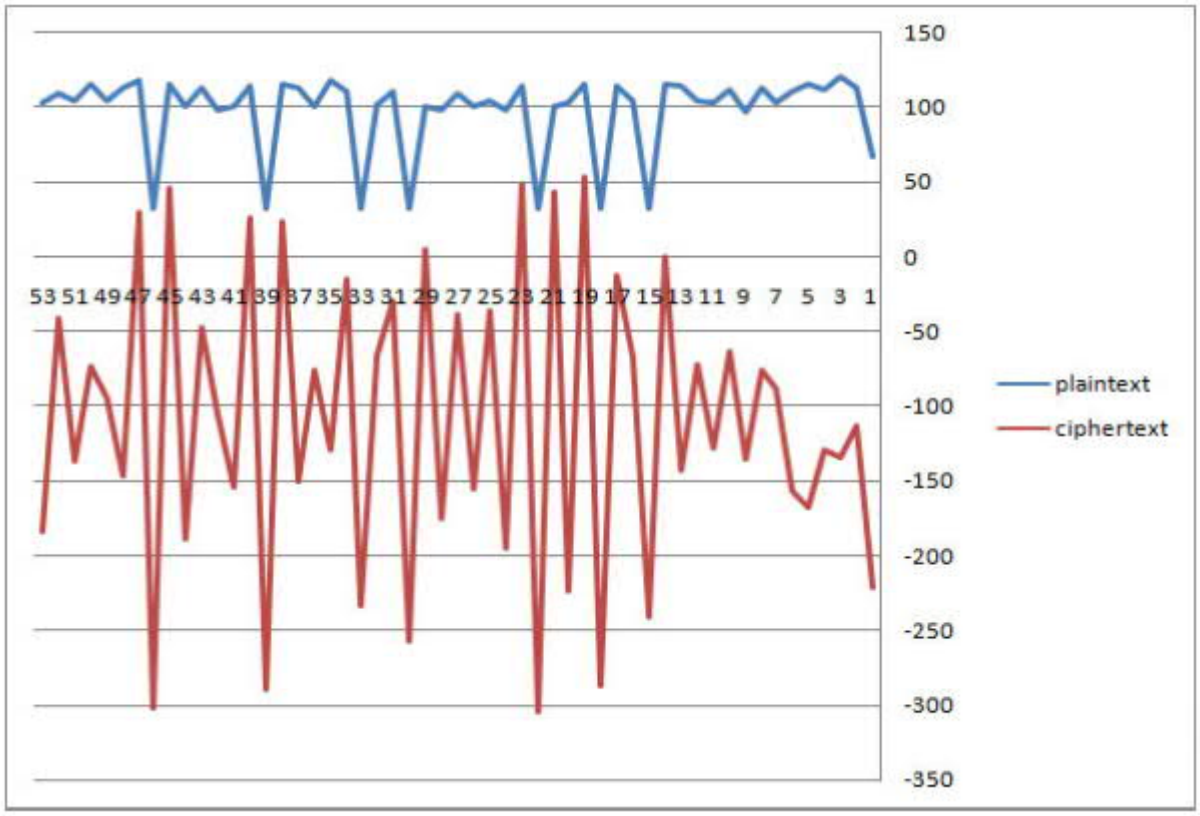


Рисунок 3.5 – Контраст між звичайним текстом та шифрованим текстом

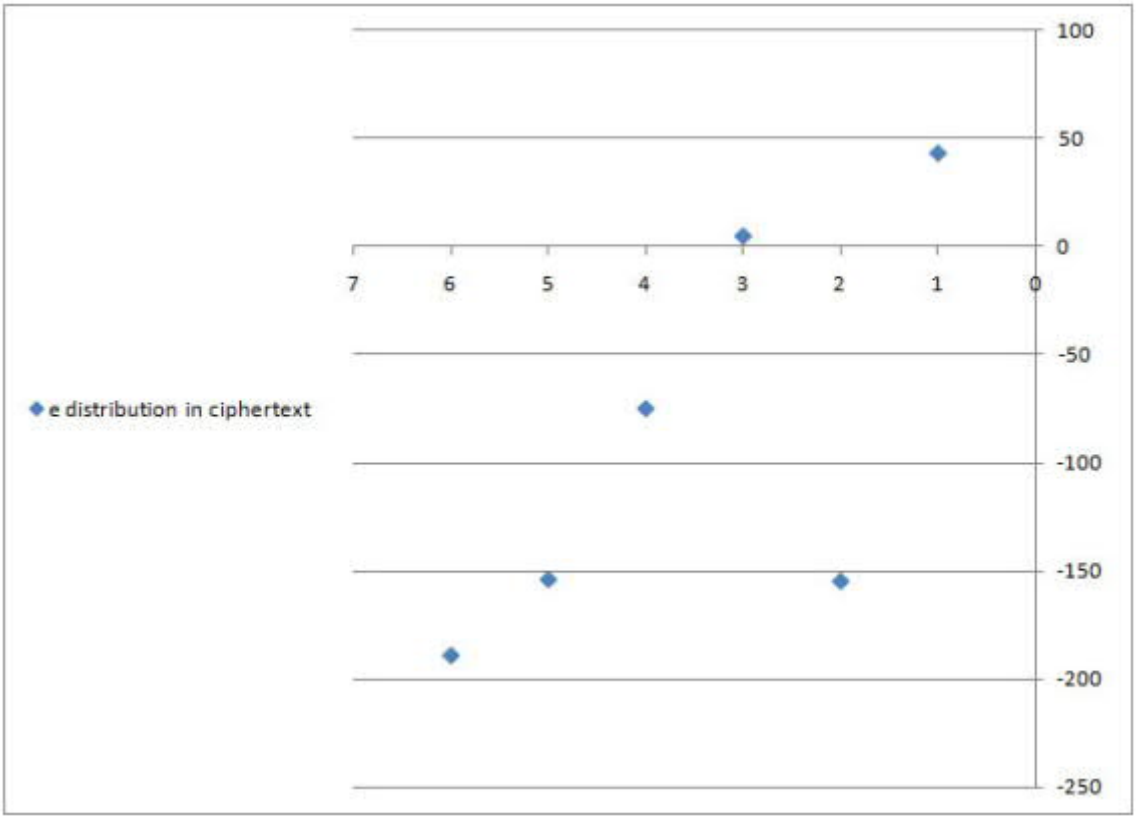


Рисунок 3.6 – Розподіл символу 'е' на шифрований текст

Аналогічний експеримент також проводився до речення, що складається з послідовних *m* як звичайного тексту з довжиною = 10, та тією ж літерою, що й попередній експеримент, приведений шифрований текст: {0, 0.9589, 0.7183, 0.7183, 0.9589, 0.9619, 0.263, 0.7581, 0.263, 0.9619}.

Навіть при цьому дуже екстремальних умовах не можна помітити жодного відношення між відкритим текстом і шифрованим текстом, а поширення шифрувального тексту є випадковим. Це підтверджує те, що ми згадали про плутанину і дифузійні властивості, яка забезпечується запропонованою схемою. Для підтвердження наших результатів проводиться ще один експеримент. Ми зашифрували інше повідомлення, подібне до попереднього, використовуючи той самий ключ, який раніше використовувався, щоб побачити, що трапляється, коли два дуже схожі тексти зашифровані під одним ключем. Припустимо M = "mmmnmmmomm". Приведений зашифрований текст: {0, 0.0892, -0.2902, 1.1603, - 0.2678, 1.0292, -2.5630, 4.4555, -2.5630, 1.0292}.

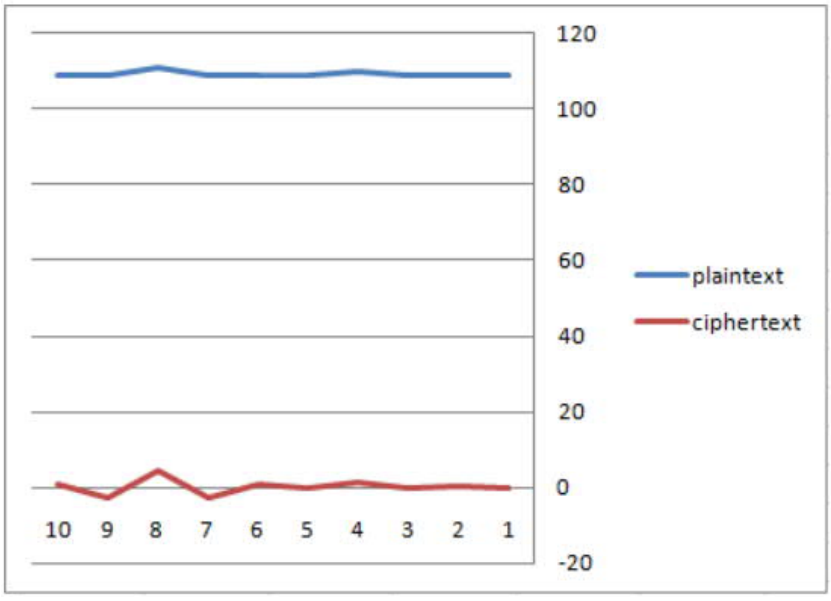


Рисунок 3.7 – Контраст між звичайним текстом та шифрованим текстом

Як ми бачимо чітко, приведений зашифрований текст цілком відрізняється від попереднього шифрувального тексту, хоча ці два повідомлення однакові. Контраст між відкритим текстом і шифрованим

текстом показаний на (рисунку 3.7). Ще один експеримент виконується на попередньому повідомленні за допомогою аналогічного ключа. M = "mmmnmmmomm", krbf = "aacde", приведений зашифрований текст: {-0.8200, -0.7501, -1.2799, 0.4468, -1.3067, 0.0000, - 2.8444, 3.6444, -2.8444, 0}.

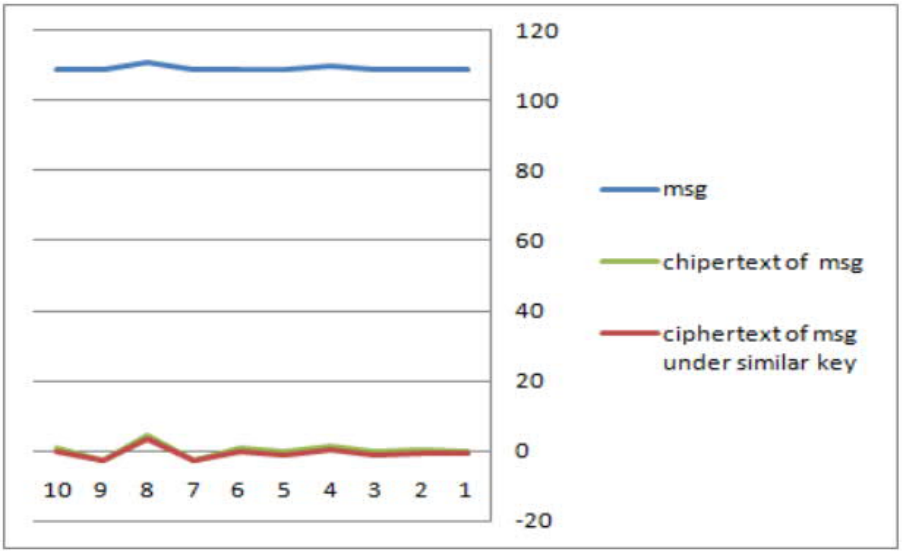


Рисунок 3.8 – Контраст між тим самим повідомленням та його шифрованим текстом за подібними ключами

Знову бачимо, що приведений шифр абсолютно відрізняється від попереднього експерименту, як показано на рисунку 3.8 хоча це одне і те ж повідомлення і зашифровано за подібним ключем.

У даній роботі пропонується нова криптосистема на основі запропонованих гібридних підходів. Запропонована криптосистема надавала послуги з багажним захистом, таких як конфіденційність, автентифікація та цілісність, які в більшості додатків розглядаються як одна з важливих служб безпеки. Для забезпечення конфіденційності даних запропоновано алгоритм шифрування на основі нейронної мережі, тоді як комбінація між технікою MAC та алгоритмом шифрування RSA виробляла цифровий підпис схеми для надання інших згаданих служб безпеки. У цій схемі цифровий підпис, по-перше, здійснюється шляхом обчислення MAC повідомлення, а потім підписується приватним ключем відправника, створеного алгоритмом RSA. Кодування повідомлення здійснюється за допомогою використання

нейронної мережі, а ключ зі складанням як вхід до моделі, і дані повинні бути захищені, буде вихід з нього, а сформований набір ваг буде діяти як шифр. Нарешті, підписані MAC і шифр будуть відправляти в кібер-простір. На кінцевому приймачі, після дешифрування, цифровий підпис може бути використаний для перевірки цілісності повідомлення та аутентифікації відправника. Результати експериментів вказують на те, що запропонована криптосистема має високі плутанини та дифузійні властивості, тому вона має високу безпеку і підходить для безпечної комунікації.

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

Метою техніко – економічного розділу дипломної роботи є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності програмного засобу нейромережевого симетричного шифрування на базі базисної радіальної функції та прийняття рішення про його подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки. Для проведення даного дослідження необхідно провести ряд розрахунків.

4.1 Розрахунок витрат на розробку програмного модуля

Витрати на розробку і впровадження програмного модуля для маршрутизації запитів у комп'ютерній мережі ($К$) включають:

$$К = К_1 + К_2,$$

де $К_1$ - витрати на розробку апаратного та програмного забезпечення грн.;

$К_2$ - витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку апаратних та програмних засобів включають:

- витрати на оплату праці розробників ($В_{ОП}$);
- витрати на відрахування у спеціальні державні фонди ($В_ф$);
- витрати на матеріали та комплектуючі ($П_в$);
- накладні витрати ($Н$);
- інші витрати ($І_в$)
- витрати на використання комп'ютерної техніки($В_{КТ}$).

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудоємності відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломант; консультант техніко-економічного розділу (таблиця 4.1).

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

№п/п	Посада виконавців	Місячний оклад, грн.
1	Керівник ДП, викладач	6026
2	Консультант техніко-економічного розділу, Доцент	6026
3	Студент	1100

Витрати на оплату праці розробників проекту визначаються за наступною формулою:

$$B_{ОП} = \sum_{i=1}^{N} \sum_{j=1}^{M} n_{ij} \cdot t_{ij} \cdot C_{ij} \ , \qquad (4.1)$$

де n_{ij} – чисельність розробників i-ої спеціальності j-го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником i-ої спеціальності j-го тарифного розряду, год;

C_{ij} – годинна ставка працівника i-ої спеціальності j-го тарифного розряду, грн.,

Середньогодинна ставка працівника може бути розрахована за такою формулою:

$$C_{ij} = \frac{C_{ij}^{0}(1+h)}{РЧ_i}, \qquad (4.2)$$

де C_{ij} – основна місячна заробітна плата розробника *i*-ої спеціальності *j*-го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$РЧ_i$ - місячний фонд робочого часу працівника *i*-ої спеціальності *j*-го тарифного розряду, год. (приймаємо 168 год.).

Коефіцієнт *h*, який визначає розмір додаткової заробітної плати, для керівника та консультанта техніко-економічного розділу дорівнює 0,47.

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 - Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
1	Керівник ДП, доцент	16	88,6	1417,6
2	Консультант техніко-економічного розділу, доцент	2	88,6	177,2
3	Студент	144	6,55	943,2
Разом				2538

Відрахування на соціальні заходи. Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5% від суми заробітної плати: $B_Ф = \frac{20,5}{100} \cdot 2538 = 520,29$ грн.

Загальна сума витрат на матеріальні ресурси (B_M) визначається за формулою:

$$B_M = \sum_{i=1}^{n} K_i \cdot Ц_i, \qquad (4.3)$$

де K_i - витрата i-го типу матеріалу, натуральні одиниці вимірювання;

$Ц_i$ - ціна за одиницю i-го типу матеріалу, грн.;

i - тип матеріального ресурсу;

n - кількість типів матеріальних ресурсів.

Таблиця 4.3 - Зведені розрахунки матеріальних витрат

Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна за одиницю, грн.	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
Arduino UNO	шт	1	412	412	41,2	452.2
Папір (формат А4)	уп	2	80	160	16	176
Ручка кулькова	шт	2	10	20	2	22
Датчик температ. та вологості	шт	1	100	100	10	110
Герконовий датчик	шт	2	35.1	70.2	7	77.2
Датчик руху	шт	1	127	127	12.7	139.7
Датчик відстані	шт	1	88.5	88.5	8,85	96,85
Р а з о м						1073,95

Витрати на використання комп'ютерної техніки ($В_{КТ}$) включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 6 грн. Середній щоденний час роботи на комп'ютері – 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4- Розрахунок витрат на використання комп'ютерної техніки

№ п/п	Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера грн.
1	Проведення досліджень та оформлення їх результатів	60	360
2	Оформлення техніко-економічного розділу	8	48
3	Оформлення ДП	12	72
Разом		80	480

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати: $H = 1{,}5 \cdot 1860{,}4 = 2790{,}6$ (грн).

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати: $I_B = 1860{,}4 \cdot 0{,}1 = 186{,}04$ (грн).

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{ОП} + B_Ф + B_M + H + I_B + B_{КТ},$$

$$K_1 = 1860{,}4+381{,}38+1111{,}00+2790{,}6+186{,}04+480{,}00=6809{,}42 \text{ (грн)}.$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від} \qquad (4.4)$$

де $S_{м.г.}$ - вартість однієї машино-години роботи ПК, грн./год;

$t_{від}$ - комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 6 грн., тому $К_2$= 6·60 = 360 грн.

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблювальної системи моніторингу слід порівняти її з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_П = E_{1П} + E_{2П},$$

де $E_П$ - одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1П}$ - вартість підготовки даних для експлуатації ПЗ (аналогу), грн.;

$E_{2П}$ - вартість роботи комп'ютера для виконання проектного рішення (аналогу), грн.

Річні експлуатаційні витрати $B_{ЕП}$ визначаються за формулою:

$$B_{ЕП} = E_П * N_П,$$

де $N_П$ - періодичність експлуатації ПЗ (аналогу), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{1П} = \sum_{l=1}^{n} n_i t_i c_i ,$$

де i - категорії працівників, які приймають участь у підготовці даних (i=1,2,...n);

n_i - кількість працівників i-ої категорії, осіб.;

t_i - трудомісткість роботи співробітників i-ої категорії по підготовці даних, год.;

c_i - середнього годинна ставка працівника i-ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0 (1+b)}{m},$$

де c_i^0 - основна місячна заробітна плата працівника i-ої категорії, грн.;

b - коефіцієнт, який враховує додаткову заробітну плату (приймемо 0,57);

m - кількість робочих годин у місяці, год.

Для роботи з даними як для проектного рішення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає: c = 3723 грн. Тоді:

$$c_1 = \frac{3723(1+0,57)}{22*8} = 33,21 \text{грн/год}$$

Трудомісткість підготовки даних для проектного рішення складає 1 год., для аналога 1,5 год.

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2П} = t * S_{МГ},$$

де t - витрати машинного часу для реалізації рішення (аналогу), год.;

$S_{МГ}$ - вартість однієї години роботи комп'ютера, грн./год.

Таблиця 4.5 - Розрахунок витрат на підготовку даних та реалізацію проектного рішення на комп'ютері

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати , грн.
Проектне рішення			
1	1	33,21	33,21
Аналог			
2	1,5	33,21	66,42

Далі:

$$E_{2П} = 1* 6 = 6 \text{ грн.}; \quad E_{2А}=1,5*6 = 9 \text{ грн.}$$

$$E_{П}=33,21+6=39,21 \text{ грн.} ; \quad E_{А}=66,42+9 = 75,42 \text{ грн.}$$

$$B_{ЕП}= 39,21*252=9880,92 \text{ грн.} ; \quad B_{ЕА}=75,42 *252=19005,84 \text{ грн.}$$

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 60–100 % від суми основної та додаткової заробітної плати працівників.

$$H_B = 0,7 * B_{ОП}, \qquad (4.5)$$

де H_B – накладні витрати.

$$H_B=0,7*5845,11=4091,58 \text{ грн.}$$

Результати проведених розрахунків зведемо у таблицю 4.6.

Таблиця 4.6 - Кошторис витрат

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	1860,4
2	Відрахування у спеціальні державні фонди	381,38
3	Витрати на матеріали та комплектуючі	1074,00
4	Накладні витрати на розробку	2790,6
5	Інші витрати	186,04
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту	360
7	Накладні витрати експлуатацію	4091,58
8	Річні експлуатаційні витрати	19005,84
Разом		29770,09

Договірна ціна ($Ц_Д$) для проектних рішень розраховується за формулою:

$$Ц_Д = B_{КС} \cdot \left(1 + \frac{p}{100}\right), \quad (4.6)$$

де $B_{КС}$ – кошторисна вартість, грн.;

p - середній рівень рентабельності, % (приймаємо 26% за погодженням з керівником): $Ц_Д$ =29770,09·(1+0,26)=37524,47 грн.

4.3 Визначення економічної ефективності і терміну окупності капітальних вкладень

Економічна ефективність ($E_ф$) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_{ф} = \frac{П}{B_{КС}} \ , \qquad (4.7)$$

де $П$ – прибуток, грн.;

$B_{КС}$ – кошторисна вартість, грн..

$E_{ф}$=7812,27 грн. / 29786,09 грн. = 0,25.

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень ($Тр$):

$$T_{Р} = \frac{1}{E_{Р}} \ , \qquad (4.8)$$

Тобто: $T_{Р}$ =1/0,25 = 4р.

Прийнятним вважається термін окупності, близький до 7 років.

Розраховані економічні показники проекту занесемо до (таблиці 4.7).

Таблиця 4.7 - Економічні показники розробки

№ п/п	Показник	Значення
1.	Собівартість, грн.	29786,09
2.	Плановий прибуток, грн.	7744,38в
3.	Ціна, грн.	37524,47
4.	Економічна ефективність	0,25
5.	Термін окупності, рік	4

Враховуючи основі економічні показники з таблиці 4.7, можна зробити висновок, що при економічній ефективності 0,25 та терміні окупності 4 роки проводити роботи по впровадженню даного програмного модуля є доцільним та економічно вигідним.

ВИСНОВКИ

У даній роботі розглянуто спосіб шифрування тексту за допомогою нейронних мереж, зокрема нейронних мереж на основі радіально-базисних функцій.

1. Проведено аналіз методів криптографічного захисту інформації починаючи з найдавніших часів і до сучасності. Розглянуто підходи до реалізації криптографічних методів, використовуваних на сьогоднішній день в практиці.

2. Вивчено можливості нейронних мереж та проведено аналіз їх структури і особливостей. Обґрунтовано вибір нейромережевої моделі для запропонованої системи шифрування, розглянуті сильні і слабкі сторони цього алгоритму.

3. Розроблено модель алгоритму на основі базової-радіальної мережі, оцінені можливості і властивості обраної моделі.

4. Розроблено методи попередньої обробки даних, складання навчальної множини і сам метод навчання радіально-базисної нейронної мережі. Здійснено програмну реалізацію алгоритму і його тестування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Осовский С. Нейронні мережі для обробки інформації : підручник / С. Осовский. – Вид. 9-те, доповн. та переробл. – Х. : Співак Т. К., 2011.– 4 с.

2. Жельников В. Криптографія від папірусу до комп'ютера. — М.: ABF, 1996. - 335 с. — ISBN 5-87484-054-0.

3. Мао Венбо. Сучасна криптографія: теорія і практика. Пер. с англ. — М. : Видавничий будинок «Вільямс», 2005. — 768 с.: ил. — Парал. тит. англ.

4. Баричев С.Г. Основи сучасної криптографії./ С.Г. Баричев, В.В. Гончаров, Р.Е. Серов— М.: Горячая линия — Телеком, 2002. — 175 с.

5. Дэвид Кан. Зломщики кодів. — Центрполіграф, 2000 год. — 480 с. — (Секретна папка). — 10 000 экз. — ISBN 5-227-00678-4.

6. Jesse Russell Искусственная нейронная сеть / Jesse Russell. - М.: VSD, 2012. - 532 с.

7. Барский А. Б. Логические нейронные сети / А.Б. Барский. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2007. - 352 с.

8. Барский, А.Б. Логические нейронные сети / А.Б. Барский. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2013. - 279 с.

9. Бунаков В. Е. Нейронная физика. Учебное пособие: моногр. / В.Е. Бунаков, Л.В. Краснов. - М.: Издательство Санкт-Петербургского университета, 2015. - 200 с.

10. Головинский, П. А. Математические модели. Теоретическая физика и анализ сложных систем. Книга 2. От нелинейных колебаний до искусственных нейронов и сложных систем / П.А. Головинский. - М.: Либроком, 2012. - 234 с.

11. Денис, Хусаинов Механизмы ритмической активности нейронов виноградной улитки / Хусаинов Денис , Иван Коренюк und Татьяна Гамма. - М.: LAP Lambert Academic Publishing, 2012. - 108 с.

12. Круглов В.В. Искусственные нейронные сети. Теория и практика: моногр. / В.В. Круглов, В.В. Борисов. - М.: Горячая линия - Телеком; Издание 2-е, стер., 2002. - 382 с.

13. Мандельштам Ю. Е. Нейрон и мышца насекомого: моногр. / Ю.Е. Мандельштам. - М.: Наука, 1983. - 168 с.

14. Парвин Манучер Из серого. Концерт для нейронов и синапсов / Манучер Парвин. - М.: Страта, 2015. - 408 с.

15. Позин Н. В. Моделирование нейронных структур / Н.В. Позин. - М.: Наука, 1970. - 264 с.

16. Рассел Джесси Вербализация нейронных сетей / Джесси Рассел. - М.: VSD, 2013. - 127 с.

17. Рассел Джесси Искусственный нейрон / Джесси Рассел. - М.: VSD, 2013. - 167 с.

18. Татузов А. Л. Нейронные сети в задачах радиолокации / А.Л. Татузов. - М.: Радиотехника, 2009. - 432 с.

19. Толкачев С. Нейронное программирование диалоговых систем / С. Толкачев. - Москва: РГГУ, 2016. - 192 с.

20. Шибзухов З. М. Конструктивные методы обучения сигма-пи нейронных сетей / З.М. Шибзухов. - М.: Наука, 2006. - 160 с.

21. Юревич Артур Нейронные сети в экономике / Артур Юревич. - М.: LAP Lambert Academic Publishing, 2014. - 80 с.

22. В.А.Головко. Нейронные сети: обучение, организация и применение. М., ИПРЖР, 2001. - 148 с.

23. Тархов Д.А. Нейронные сети. Модели и алгоритмы. (Справочник.)/ Д.А. Тархов– М., Радиотехника, 2005. - 108 с.

24. Круглов В.В. Нечёткая логика и искусственные нейронные сети. // В.В.Круглов, М.И.Дли, Р.Ю.Голунов./ Физматлит, 2001. - 96 с.

25. Галушкин А.И. Нейронные сети. Основы теории. М., Горячая линия - Телеком, 2010. - 110 с.

26. Комарцова Л.Г. Нейрокомпьютеры.// Л.Г.Комарцова, А.В.Максимов./ М., Изд-во МГТУ им.Баумана, 2004. - 203 с.

27. Хайкин С. Нейронные сети: полный курс.// С.Хайкин. / 2-е изд. М., "Вильямс", 2006. - 155 с.

28. Яхъяева Г.Э. Нечеткие множества и нейронные сети: Учебное пособие / Г.Э. Яхъяева. - М.: БИНОМ. ЛЗ, ИНТУИТ.РУ, 2012. - 316 с.

29. Ширяев В.И. Финансовые рынки: Нейронные сети, хаос и нелинейная динамика / В.И. Ширяев. - М.: КД Либроком, 2016. - 232 с.

30. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп’ютерна інженерія» фахового спрямування «Комп’ютерні системи та мережі» / О.М. Березький, Л.О.Дубчак, Р.Б. Трембач, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2016. – 65 с.

31. Методичні вказівки до написання техніко-економічного розділу для дипломних проектів на здобуття освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп’ютерна інженерія» / І.Р.Паздрій. Тернопіль: ТНЕУ, 2015.– 36 с.