

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДНІПРОПЕТРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ОЛЕСЯ ГОНЧАРА

Факультет прикладної математики
Кафедра обчислювальної математики та математичної кібернетики

ДИПЛОМНА РОБОТА
за рівнем бакалавра

**на тему: «Прогнозування первинної інвалідності в Україні
з використанням методів регресійного аналізу»**

Виконав: студент 4 курсу, групи ПМ-13-1
напряму підготовки
6.040301 Прикладна математика
(шифр і назва напряму підготовки, спеціальності)
Кривоносов О.Д.
(прізвище та ініціали)

Керівник: д.ф.-м.н., проф., проф.
(наук. ступ., вчене звання, посада, прізвище та ініціали)
Кузьменко В.І.
(підпис)

Рецензент: д.ф.-м.н., проф., зав. каф. ПКТ
(наук. ступ., вчене звання, посада, прізвище та ініціали)
Гук Н.А.
(підпис)

ДНІПРОПЕТРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ОЛЕСЯ ГОНЧАРА

Факультет прикладної математики

Кафедра обчислювальної математики та математичної кібернетики

Рівень (освітньо-кваліфікаційний рівень) перший (бакалаврський)

Напрямок підготовки 6.040301 Прикладна математика

(шифр і назва)

Спеціальність _____

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної математики та математичної кібернетики
(повна назва)

Турчина В.А.

(підпис)

(П.І.Б.)

“ ” _____ 2017 року

З А В Д А Н Н Я **НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Кривоносову Олександр Дмитровичу

(прізвище, ім'я по батькові)

1. Тема роботи Прогнозування первинної інвалідності в Україні з використанням методів регресійного аналізу

керівник роботи Кузьменко Василь Іванович, д.ф.-м.н., проф.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від « 29 » березня 2017 р. № 464с

2. Термін здачі студентом закінченої роботи 05 червня 2017 р.

3. Вхідні дані до роботи Текстові файли з часовими рядами

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) 1. Ознайомитися з методом прогнозування на основі відновлення регресії. 2. Ознайомитися із застосуванням нейронних мереж для розв'язання задачі відновлення регресії. 3. Розробити програмне забезпечення, яке вирішує задачу прогнозування із використанням нейронних мереж. 4. За допомогою створеного програмного забезпечення провести прогнозування первинної інвалідності в Україні за різними нозологіями.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Графічне зображення початкового часового ряду та результатів його прогнозування. Скріншоти роботи програми, презентація у Microsoft PowerPoint.

6. Консультанти по роботі, Із зазначенням розділів проекту, що стосуються їх

| Розділ | Консультант | Підпис. дата | |
|----------|----------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| Розділ 1 | Кузьменко В.І. | | |
| Розділ 2 | Кузьменко В.І. | | |
| Розділ 3 | Кузьменко В.І. | | |

7. Дата видачі завдання 31.03.2017

КАЛЕНДАРНИЙ ПЛАН

| Пор. № | Назва станів дипломної роботи | Термін виконання станів роботи | Примітка |
|--------|---|--------------------------------|----------|
| 1 | Огляд задачі прогнозування часових рядів та методів її розв'язання, опрацювання літературних джерел | 31.03.2017 - 15.04.2017 | |
| 2 | Розробка програмного продукту, що реалізує алгоритм прогнозування | 14.04.2017 - 11.05.2017 | |
| 3 | Тестування програмного продукту | 11.05.2017 - 25.05.2017 | |
| 4 | Апробація програмного продукту на реальних часових рядах | 25.05.2017 - 01.06.2017 | |
| 5 | Оформлення пояснювальної записки | 01.06.2017 - 05.06.2017 | |

Студент _____
(підпис)

Кривоносов О.Д.
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Кузьменко В.І.
(прізвище та ініціали)

РЕФЕРАТ

Дипломна робота: «Прогнозування первинної інвалідності в Україні з використанням методів регресійного аналізу» 45 с., 24 Рисунок , 7 джерел.

Об'єктом дослідження є прогнозування часових рядів.

Мета роботи: реалізувати алгоритм розв'язання задачі прогнозування часових рядів на основі штучної нейронної мережі для прогнозування первинної інвалідності.

Методи дослідження: методи оптимізації, методи регресійного аналізу.

У процесі роботи реалізовані багатошаровий перцептрон для апроксимації функцій, гамма-юніти для згортки рядів; була створена програма для прогнозування і дослідження якості прогнозування часових рядів, вивчені моделі і методи розв'язування задач прогнозування часових рядів; програма випробувана на даних первинної інвалідності в Україні, даних щоденної кількості проданих товарів на торгових точках.

В результаті роботи розроблено програмний продукт, призначений для розв'язання задач прогнозування часових рядів. Алгоритм прогнозування створений на мові C++, інтерфейс користувача створений на мові R з використанням програмного пакету Shiny.

Ключові слова: часовий ряд, прогнозування, апроксимація функції, штучна нейронна мережа, перцептрон, гамма-пам'ять.

ABSTRACT

The graduation research of the fourth-year student Krivonosov Aleksandr deals with forecasting primary disability in Ukraine using regression analysis techniques.

Purpose of the work is to develop an algorithm for solving the problem of time series prediction based on artificial neural network for forecasting primary disability.

In the process of work there were implemented multilayer perceptron for approximation of functions, gamma-units for convolution of series; program was created for the prediction and research of time series prediction quality, studied models and methods for solving time-series forecasting; program tested on data from primary disability in Ukraine, data of daily sales.

Prediction algorithm is created in C++; user interface created in R- language using software package Shiny.

The results of the work may be used in the automation systems of scientific researches, forecasting medical and economic indices, application packages.

Bibliog. 7, ill. 24.

ЗМІСТ

| | |
|---|----|
| ВСТУП..... | 7 |
| ПОСТАНОВКА ЗАДАЧІ..... | 10 |
| 1 ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ | 11 |
| 1.1 Огляд методів прогнозування часових рядів | 11 |
| 1.2 Регресійні методи в прогнозуванні часових рядів..... | 16 |
| 1.3 Нейромережеве прогнозування часових рядів..... | 17 |
| 2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.... | 21 |
| 2.1 Функціональні можливості та структура програми | 21 |
| 2.2 Організація обчислювального процесу | 22 |
| 2.3 Інструкція користувача..... | 25 |
| 3 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНИХ ЕКСПЕРИМЕНТІВ | 33 |
| 3.1 Прогнозування первинної інвалідності за цереброваскулярними хворобами (на 10 тисяч населення). | 33 |
| 3.2 Прогнозування щоденної кількості проданого товару. | 38 |
| ВИСНОВКИ | 42 |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ | 44 |

ВСТУП

Для вивчення властивостей складних систем широко використовується підхід, заснований на аналізі сигналів, вироблених системою. Це дуже актуально в тих випадках, коли математично описати досліджуваний процес неможливо, але в нашому розпорядженні може бути деяка характерна спостережувана величина. Тому аналіз систем, особливо при експериментальних дослідженнях, часто реалізується за допомогою оброблення реєстрованих сигналів. Наприклад, в медицині — кардіограми, в сейсмології — коливання земної кори, в метеорології — дані метеоспостережень.

Часовий ряд — масив з $[x_i]_{i=1}^N$, що представляє собою впорядковану за часом послідовність значення деякої змінної $x(t)$, яка спостерігається з деяким постійним кроком τ по часу, $t_i = t_0 + (i - 1)\tau$; $x_i = x(t_i)$, $i = 1..N$. У аналізі часових рядів виділяються дві основні задачі: задача ідентифікації та задача прогнозу.

Задача ідентифікації при аналізі часового ряду передбачає відповідь на питання, які є параметри системи, що породила часовий ряд: розмірність вкладення, ентропія (перетворення) та інші. Розмірність вкладення — це мінімальне число динамічних змінних, що однозначно описують спостережуваний процес. Поняття ентропії пов'язане з передбачуваністю значень ряду і всієї системи.

Задача прогнозу має на меті за даними спостережень передбачити майбутні значення вимірюваних характеристик досліджуваного об'єкта, тобто побудувати прогноз на певний відрізок часу вперед. Є два основні класи методів прогнозу: локальні і глобальні. Такий поділ проводиться по області визначення параметрів апроксимуючої функції, що рекурентно встановлює наступне значення часового ряду за кількома попередніми.

Історично першими були розроблені глобальні методи, в яких на основі статистичного аналізу пропонувалося використовувати авторегресію, ковзне

середнє і інші. Пізніше в рамках нелінійної динаміки були розроблені нові практичні методики:

- сингулярний спектральний аналіз (SSA), який є глобальним методом;
- локальна апроксимація (LA);
- поєднання SSA-LA.

Дослідження часових рядів базується на ідеї, що прогнозувати ряд можна, якщо замість змінних, що входять у вихідну систему, використовувати так звані вектори затримок спостережень $z_i = [x_i, x_{i+1}, \dots, x_{i+m-1}]$. Є два варіанти того, як можна подати затримку спостережень на вхід до апроксиматора:

- Використовуючи неявне представлення.

Час представляється ефектом, який він справляє на обробку сигналу, тобто неявним чином. Можна застосувати згортку до вектора затримок спостережень і отримати одне число, яке і вважати параметром. Таким чином ефект, який справляє час на сигнал можна контролювати змінюючи функцію згортки вектора.

- Використовуючи явне представлення.

Час має власне конкретне представлення. Наприклад, система ехолокації кажана посиляє короткий частотно-модульований сигнал, при цьому встановлюючи єдиний рівень інтенсивності для кожного з частотних каналів на короткий період FM-розгортки. Для того щоб отримати точну інформацію про відстань до цілі, проводяться численні порівняння декількох різних частот, кодованих масивом слухових рецепторів. Коли відлуння отримується від об'єкта з невідомою затримкою, відповідає той нейрон (слухової системи), який має відповідну затримку в лінії. Таким чином оцінюється відстань до об'єкта.

Мета роботи: реалізувати алгоритм розв'язання задачі прогнозування часових рядів на основі штучної нейронної мережі для прогнозування первинної інвалідності.

Робота складається з трьох розділів.

- У першому розділі подано огляд методів прогнозування часових рядів.
- У другому розділі описується програмне забезпечення прогнозування часових рядів.
- У третьому розділі приводяться результати апробації програми.

ПОСТАНОВКА ЗАДАЧІ

Задано $[x_t; t = 1, n]$ – часовий ряд (послідовність значень деякого показника впорядкована по даті фіксування; передбачається, що фіксування значень виконується з однаковим інтервалом), де x_t – елементи ряду, n – кількість елементів ряду.

Ставиться задача прогнозування ряду на наступний момент часу.

Для її розв'язання у роботі необхідно:

1. Ознайомитися з методом прогнозування на основі відновлення регресії.
2. Ознайомитися із застосуванням нейронних мереж для розв'язання задачі відновлення регресії.
3. Розробити програмне забезпечення, яке вирішує задачу прогнозування із використанням нейронних мереж.

За допомогою створеного програмного забезпечення провести прогнозування первинної інвалідності в Україні за різними нозологіями.

1 ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

1.1 Огляд методів прогнозування часових рядів

Метод ковзного середнього (moving average, MA).

Ковзне середнє - загальна назва для сімейства функцій, значення яких в кожній точці визначення дорівнюють середньому значенню початкової функції за попередній період. Ковзне середнє зазвичай використовується з даними часових рядів для згладжування короткострокових коливань і виділення основних тенденцій або циклів. Математично ковзне середнє є одним з видів згортки.

Просте ковзне середнє, або арифметичне ковзне середнє **SMA** чисельно дорівнює середньому арифметичному значень вихідної функції за встановлений період і обчислюється за формулою:

$$SMA_t = \frac{1}{n} \sum_{i=0}^{n-1} x_{t-i} = \frac{x_t + x_{t-1} + \dots + x_{t-i} + \dots + x_{t-n+2} + x_{t-n+1}}{n},$$

де SMA_t – значення простого ковзного середнього в точці t , n – кількість елементів ряду для розрахунку ковзного середнього; чим більше число n , тим більш плавним виходить графік ковзного середнього, x_{t-i} – значення початкового ряду в точці $t - i$.

Експоненціально зважене ковзне середнє **ЕМА** - різновид зваженої ковзної середньої, ваги якої зменшуються експоненціально і ніколи не дорівнюють нулю. Визначається наступною формулою:

$$EMA_t = \alpha * x_t + (1 - \alpha) * EMA_{t-1},$$

де EMA_t – значення експоненціального ковзного середнього в точці t , EMA_{t-1} – значення експоненціального ковзного середнього в точці $t - 1$, x_t – значення з ряду в момент часу t , α – коефіцієнт що характеризує швидкість зменшення вагів, приймає значення від 0 і до 1, чим менше його значення тим більше вплив попередніх значень на поточну величину середнього.

Для прогнозування метод ковзного середнього використовують аналогічно до того, як згладжують ряд — беруть n відомих значень ряду, обчислюють SMA_t і EMA_t і беруть одне з цих значень як оцінку наступного значення ряду.

Модель Бокса-Дженкінса (ARIMA)

ARIMA (autoregressive integrated moving average) - інтегрована модель авторегресії та ковзного середнього - модель і методологія аналізу часових рядів. Є розширенням моделей ARMA для нестационарних часових рядів, які можна зробити стаціонарними взяттям різниць деякого порядку від вихідного часового ряду.

Модель ARIMA(p,d,q) для нестационарного часового ряду X_t має вигляд:

$$\Delta^d X_t = c + \sum_{i=1}^p a_i \Delta^d X_{t-i} + \sum_{j=1}^q b_j \varepsilon_{t-j} + \varepsilon_t,$$

де ε_t – ряд похибок, c, a_i, b_j – параметри моделі, Δ^d – оператор різниці часового ряду порядку d (послідовне взяття d раз різниць першого порядку - спочатку від часового ряду, потім від отриманих різниць першого порядку, потім від другого порядку і т.д.).

Метод аналізу сингулярного спектру (SSA)

SSA (Singular spectrum analysis) - метод аналізу часових рядів, заснований на перетворенні одновимірного часового ряду в багатовимірний ряд з подальшим застосуванням до отриманого багатовимірного часового ряду методу головних компонент.

Спосіб перетворення одновимірного ряду в багатовимірний є «згортою» часового ряду в матрицю, що містить фрагменти часового ряду, отримані з деяким зрушенням. Загальний вигляд процедури нагадує «гусеницю», тому сам метод нерідко так і називають - «Гусениця»: довжина фрагмента називається довжиною «гусениці», а величина зсуву одного фрагмента щодо іншого кроком «гусениці».

SSA може бути використаний без попереднього завдання моделі ряду для аналізу довільних, в тому числі, нестационарних, рядів. Основна мета SSA - розкласти ряд в суму інтерпретованих компонент, таких як тренд, періодичні компоненти, шум. При цьому знання параметричної форми цих компонент не потрібно.

Базовий алгоритм SSA

Крок 1. Будується $L \times K$ траєкторна матриця ряду X наступним чином:

$$X = [X_1 : \dots : X_K] = (x_{ij})_{i,j=1}^{L,K} = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_K \\ x_2 & x_3 & x_4 & \dots & x_{K+1} \\ x_3 & x_4 & x_5 & \dots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \dots & x_N \end{bmatrix},$$

де $X_i = (x_i, \dots, x_{i+L-1})^T$ - вектори вкладення довжини L . Матриця X є ганкелевою, тобто має однакові елементи на антидіагоналях $i + j = \text{const}$.

Крок 2. Виконується сингулярне розкладання (SVD) траєкторної матриці X . Нехай $S = XX^T$, позначимо $\lambda_1, \dots, \lambda_L$ власні числа S , взяті у незростаючому порядку і ортонормовану систему власних векторів матриці S . Нехай $d = \text{rank}X = \max\{i: \lambda_i > 0\}$ та $V_i = \frac{X^T U_i}{\sqrt{\lambda_i}}$. У цих позначеннях сингулярне розкладання траєкторної матриці X може бути записано як $X = X_1 + \dots + X_d$.

Крок 3. Множина всіх індексів $\{1, \dots, d\}$ розбивається на m непересічних підмножин I_1, \dots, I_m . Нехай $I = \{i_1, \dots, i_p\}$. Тоді результуюча матриця X , що відповідає групі I , записується як $X_I = X_{i_1} + \dots + X_{i_p}$. Результуючі матриці обчислюються за групами і згруповане SVD розкладання матриці X може бути записано як $X = X_{I_1} + \dots + X_{I_m}$.

Крок 4. Кожна матриця X_{I_j} згрупованого розкладання ганкелізується (усереднюється по анти-діагоналям) і потім отримана ганкелева матриця трансформується в новий часовий ряд довжини N на основі взаємно-однозначної відповідності між ганкелевими матрицями і часовими рядами. Діагональне усереднення, застосоване до кожної результуючої матриці X_{I_k} ,

виробляє відновлені ряди $\tilde{X}^{(k)} = (\tilde{x}^{(k)}_1, \dots, \tilde{x}^{(k)}_N)$. Таким чином, вихідний ряд розкладається в суму m відновлених рядів:

$$x_n = \sum_{k=1}^m \tilde{x}^{(k)}_n \quad (n = 1, 2, \dots, N)$$

Прогнозування на основі локальної апроксимації (LA)

Відповідно до теорії Такенса-Мане прийнятний опис фазового простору динамічної системи можна отримати, якщо взяти замість реальних змінних системи p -мірні вектори затримок із значень ряду в послідовні моменти часу. При виконанні умови $p \geq 2d + 1$, де d - розмірність вкладення, можливо реконструювати фазовий простір (простір станів) системи. За умови стаціонарності часового ряду на базі цієї реконструкції будується прогноз його подальшої динаміки.

Способи визначення величини p :

- Алгоритм Грассбергера-Прокаччі. Недолік методу: неефективний при роботі з короткими (до 104 точок) часовими рядами.
- Інші методи теж мають свої недоліки: складність реалізації, велика тривалість розрахунків, неоднозначність або сумнівність результатів.

У зв'язку з цим величина p , за винятком модельних прикладів, в яких вона достовірно відома, як правило, визначається емпірично. Головний критерій у цьому випадку - вибір такого p , починаючи з якого припиняється якісна зміна прогнозу.

Побудова прогнозу на один крок вперед

Крок 1. Перетворимо скалярний часовий ряд, що містить N значень, в матрицю затримок:

$$\{x_1, x_2, \dots, x_N\} \rightarrow \mathbf{X}_{p \times (N-p+1)} = \begin{pmatrix} x_p & x_{p+1} & \cdots & x_N \\ \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & \cdots & x_{N-p+2} \\ x_1 & x_2 & \cdots & x_{N-p+1} \end{pmatrix}$$

Розмірність матриці \mathbf{X} визначається кількістю затримок p .

Після побудови матриці затримок обирається вид локального уявлення, тобто вид функції, що зв'язує таке значення ряду з попередніми:

$$x_{t+1} = f(x_t, a),$$

де a - вектор параметрів уявлення.

Найбільш поширений варіант - це лінійна апроксимація першого (LA1) порядку:

$$x_{t+1} = a_0 + x_t^T a$$

Крок 2. Вибір найближчих сусідів - виділення локальної підобласті фазового простору. Будемо вважати сусідніми до стартового вектору x_{N-p+1} вектори, що задовольняють умові:

$$\{\mathbf{x}_s\} : \sum_{s \in \omega_{\Xi}} \|\mathbf{x}_{N-p+1} - \mathbf{x}_s\| \rightarrow \min_{\omega_{N-p+1}}$$

де $s \in \omega_{\Xi} \subset \omega_{N-p+1}$, $\omega_{N-p+1} \equiv \{1, \dots, N - p + 1\}$.

Норма зазвичай береться евклідова, хоча можливе використання й інших норм. Вибір сусідів автоматично визначає локальну підобласть, в якій параметри уявлення покладаються незмінними.

Крок 3. Оцінка параметрів моделі і побудова прогнозу на один крок вперед.

Вибором типу лінійної апроксимації і розмірності реконструкції p встановлюється кількість невідомих параметрів, які потрібно оцінити. При цьому коло використовуваних даних обмежується набором сусідів стартового вектора.

Параметри моделі (вектор a) оцінюються методом найменших квадратів (МНК). Це найпоширеніший і найбільш ефективний метод. Оцінка за МНК для вектора a , позначимо її як \hat{a} знаходиться з умови:

$$\hat{a} : \sum_{\omega_s} \left(x_{s+1} - f(\mathbf{x}_s, \hat{a}) \right)^2 \rightarrow \min_{\hat{a}}.$$

Однак в більшості випадків застосування МНК в чистому вигляді неможливо через виродженість матриці факторів, що є наслідком взаємної

близькості сусідів. Тому зазвичай застосовується так зване сингулярне розкладання (SVD - Singular Value Decomposition). При цьому не завжди враховується, що оцінки, що даються цим методом, в загальному випадку зміщені, сильно залежать від машинної точності і вибору мінімального значущого сингулярного числа. Отже, при використанні SVD важливо завжди контролювати стійкість отриманих результатів.

Оцінивши значення параметрів апроксимації, неважко побудувати прогноз наступного значення ряду (стартовий вектор, позначимо індексом L):

$$\hat{x}_{L+1} = f(x_L, \hat{a})$$

Таким чином, пройшовши три описаних кроку алгоритму LA, можна побудувати прогноз одного нового значення ряду.

1.2 Регресійні методи в прогнозуванні часових рядів

Регресія – статистична залежність залежної змінної від незалежної. На відміну від чисто функціональної залежності, коли кожному значенню незалежної змінної x відповідає одне певне значення величини y , при регресійному зв'язку одному і тому ж значенню x можуть відповідати різні значення величини y .

Застосування регресійних методів під час прогнозування часових рядів базується на припущенні, що наступне значення часового ряду є функцією від попередніх значень та/або зовнішніх факторів. Задача полягає у відновленні такої регресійної залежності.

Авторегресія — частковий випадок регресійної залежності, коли в якості незалежних змінних функції регресії використовуються тільки попередні значення часового ряду.

Для вирішення задачі прогнозування часових рядів регресійним методом треба обрати апроксиматор, який буде відновлювати регресійну залежність наступного значення часового ряду від попередніх. За такий апроксиматор у роботі обрано штучну нейронну мережу.

1.3 Нейромережеве прогнозування часових рядів

Штучна нейронна мережа — математична модель, а також її програмне або апаратне втілення, побудована за принципом організації та функціонування біологічних нейронних мереж — мереж нервових клітин живого організму.

Перцептрон, або перцептрон — математична або комп'ютерна модель сприйняття інформації мозком, запропонована Френком Розенблатом в 1957 році і вперше реалізована у вигляді електронної машини «Марк-1» в 1960 році. Перцептрон складається з трьох типів елементів, а саме: сигнали, що надходять від датчиків передаються асоціативним елементів, а потім елементам, що реагують. Таким чином, перцептрон дозволяє створити набір «асоціацій» між входними стимулами і необхідною реакцією на виході. У біологічному плані це відповідає перетворенню, наприклад, зорової інформації в фізіологічну відповідь від рухових нейронів.

Модель нейрону в штучній нейронній мережі має такий вигляд:

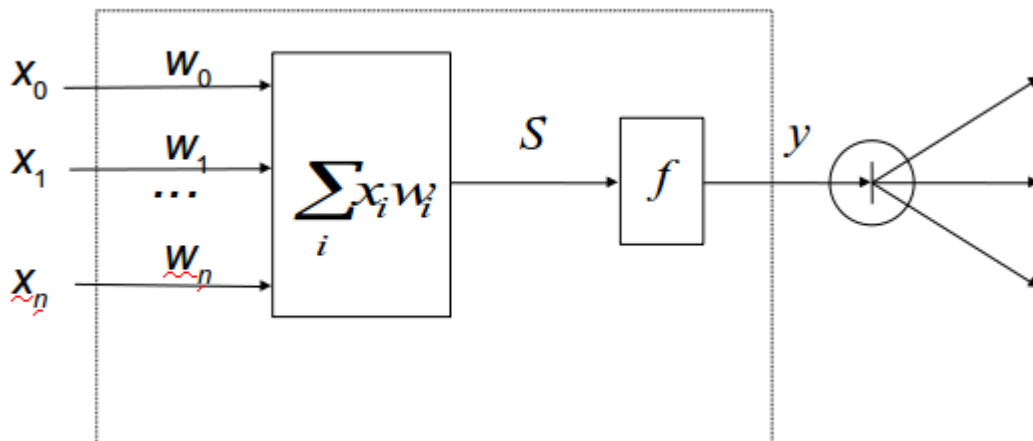


Рисунок 1.1 – Модель нейрону в штучній нейронній мережі

де x_i — значення передані нейрону на вхід, w_i — ваги для відповідних значень, $S = \sum_i^n x_i * w_i + b$, b (bias) — зсув, f — функція активації. За функцію активації найчастіше беруть сигмоїду:

$$f(s) = \frac{1}{1+e^{-2s}}.$$

Щоб перцептрон працював як апроксиматор, треба мінімізувати (оптимізувати) функцію помилок відносно вагів нейронів. Функція помилок визначається так:

$$E = \sum_{i \in \text{outputs}} (o_i - y_i)^2,$$

де *output*— множина номерів вихідних вузлів перцептрона, o_i — значення i -го виходу, y_i — значення, яке очікується отримати на i -му виході.

Для оптимізації функції помилок відносно вагів визначають градієнти вагів для функції помилок і використовують їх в одному з градієнтних методів, найчастіше використовують метод градієнтного спуску. Градієнт вага між i -ми та j -ми нейронами визначаються так:

$$\nabla w_{ij} = -\delta_j * o_i,$$

де (якщо функція активації є сигмоїдою) $\delta_j = i$, якщо j -й вузол є вихідним, і $\delta_j = o_j(1 - o_j) * (o_j - y_j)$ у інших випадках.

Щоб застосувати перцептрон для прогнозування часового ряду на вхід перцептрона подається N попередніх значень часового ряду $x(k-1), x(k-2), \dots, x(k-N)$, і, можливо, деякі допоміжні значення, а на виході отримується оцінка наступного значення часового ряду $x(k)$.

Вибрано використовувати для прогнозування нейронну мережу через те, що при використанні нейронної мережі не треба виділяти компоненти ряду (тренд, циклічність і т.д.), і також через те, що нейронна мережа є універсальним апроксиматором — у теорії, використовуючи нейронну мережу вдалої архітектури або мережу з дуже великою кількістю прихованих нейронів, можна відновити функцію будь-якої складності.

Для поліпшення моделі можуть використовуватися допоміжні показники. Це стосується не тільки нейромережових методів, а усіх методів, які здатні приймати на вхід змінну кількість параметрів і використовувати їх для покращення апроксимації. Допоміжними показниками можуть бути

перетворені початкові показники (квадрати, тригонометричні перетворення та інші).

Є особливий вид перетворення числової послідовності — згортка послідовності. Згортка одновимірної числової послідовності це таке її перетворення в результаті якого отримується скалярне число. Згортка багатовимірної числової послідовності це таке її перетворення в результаті якого отримується вектор тієї ж розмірності, що і послідовність.

Одне з таких перетворень є гамма-згортка, яке ще називається “Гамма-пам’ять”. В основі гамма-згортки лежить така функція відгуку (утворююче ядро):

$$g(n) = \mu * (1 - \mu)^{n-1},$$

де $n \geq 1$, а $\mu \in (0; 2)$. Обмеження гарантують, що $g(n)$ експоненційно прямує до нуля при зростанні n .

Значення згортки підраховується так:

$$y(n) = \sum_{k=1}^{n-1} g(k) * x(n - k)$$

Значення гамма-згортки буде використовуватися, як один з входів перцептрона, а параметр μ буде налаштовуватися у логічному блоці “гамма-юніт”.

Для налаштування параметра μ потрібно застосувати градієнтний метод, тобто треба аналітично вирахувати похідну функції помилок відносно параметра μ і застосувати в градієнтному методі оптимізації (навчання гамма-юніта). Градієнт параметра μ такий:

$$\nabla y = \left[\sum_{k=1}^{n-1} g'_{\mu}(k) * x(n - k) \right] * error,$$

де $error$ -значення помилки для вхідного значення перцептрона отримане на одній з навчальних ітерацій, а $g'_{\mu}(k)$ визначається так:

$$g'_{\mu}(k) = 1, \text{ коли } k = 1; g'_{\mu}(k) = (1 - k\mu) * (1 - \mu)^{k-2}, \text{ коли } k > 1.$$

Гамма-згортка має таку назву з того, що функція відгуку $g(n)$ є дискретним варіантом підінтегрального виразу гамма-функції.

Процес навчання моделі з гамма-юнітами здатними до навчання може бути дуже довгим, адже при зміні параметра μ , щоб отримати гамма-згортку, треба заново обчислювати всі доданки згортки, а їх кількість така ж як і кількість елементів початкового ряду, це означає що складність операції обчислення гамма-згортки $O(n * d)$, де n — кількість елементів навчального ряду, а d — розмірність ряду. Тому при навчанні моделі можна використовувати гамма-юніти двома способами:

- мати якомога більше гамма-юнітів з навчання вибраними параметрами μ і в процесі навчання моделі не змінювати μ ;
- вибрати оптимальну кількість гамма-юнітів здатних до навчання, перебираючи різні варіанти.

2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

2.1 Функціональні можливості та структура програми

Розроблено програмний продукт, призначений для розв'язання задач прогнозування часових рядів.

У програмі реалізовано такі функціональні можливості:

1. Завантажувати файл з даними часового ряду на сервер. Можна завантажувати файл будь-якого формату, але за замовченням в діалозі вибору файлу встановлений фільтр для *.csv і *.txt файлів. Для уточнення структури файлу користувач повинен налаштувати такі параметри:
 - «Заголовок» (Header), що вказує, чи має таблиця у файлі назви стовбців,
 - «Розділовий знак» (Separator), що вказує який розділовий знак використовується у файлі;
2. Задавати параметри навчання моделі такі як:
 - розмір навчальної вибірки;
 - кількість шарів прихованих вузлів;
 - кількість вузлів у кожному прихованому шарі окремо і разом (одним числом для всіх);
 - кількість гамма-юнітів;
 - кількість останніх елементів, що може бачити перцептрон;
 - параметри зупинки алгоритму навчання;
 - кількість елементів у навчальній ітерації.
3. Отримувати графіки прогнозів і початкових часових рядів та порівнювати їх;
4. Оцінювати якість прогнозу за графіками і за показниками якості RMSE і MRE.
5. Зберігати отриману модель у бінарний файл.

6. Завантажувати модель з файлу.

Програма складається з двох основних блоків: бібліотеки, що реалізує програмне забезпечення прогнозування часових рядів, користувацького інтерфейсу, що надає змогу скористатися можливостями бібліотеки без написання додаткових програм.

Бібліотека написана мовою **C++** і розділяється на дві логічні частини: незалежну від середовища, в якому буде виконуватися програма, і залежну. Такий спосіб розділення логіки надає змогу легко від'єднати незалежну частину бібліотеки і скористатися нею деінде.

Інтерфейс користувача створений на мові **R** з використанням програмного пакету **Shiny**. Таке поєднання визначає структуру частини програми відповідальної за інтерфейс. Є два логічних блоки, що відповідають за інтерфейс користувача: **UI** і **server**. Інтерфейс користувача в даному випадку — клієнт-серверна програма. **UI** відповідає за розміщення елементів інтерфейсу на екрані, а **server** за обробку даних.

Запустити сервер можливо лише на тій машині, на котрій встановлено інтерпретатор мови **R** і пакет **Shiny**. Але вже працюючий сервер надає змогу клієнтам виконувати програму на інших машинах через сторінку браузера.

2.2 Організація обчислювального процесу

Розроблені такі програмні модулі, які входять у бібліотеку (написану на **C++**) і не залежать від середовища виконання:

1. *Matrix* задає і реалізує інтерфейс і алгебраїчні операції для матриці дійсних чисел;
2. *Serialization* описує і визначає функції для серіалізації і десеріалізації (кодування і розкодування у буфер) важливих для бібліотеки структур даних;
3. *VectorAlgebra* описує і визначає функції алгебри векторів;
4. *Perceptron* відповідає за реалізацію багатошарового перцептрону. Надає можливість навчити його будь-якою кількістю паттернів завдяки функції `back_prop`. Також можна контролювати процес навчання завдяки

функціям `forward_prop`, `put_errors` і `flush`, перша з яких відповідає за `ForwardPropagation` і запам'ятовує у буфері виходи кожного зі слоїв, друга за `BackPropagation` і накопичує градієнт у буфері, а третя використовує цей градієнт для зміни вагів і очищує буфер. Використовувати ці функції при наявності масиву патернів треба так: (викликати `forward_prop`, передати помилку у `put_errors`) повторити для кожного патерну, викликати `flush`. Розділення способів навчання на більш простий і більш складний (1 і 3 функції) потрібне для того, щоб користувач об'єкта класу `Perceptron` мав змогу отримати помилки для першого слою перцептрону від функції `put_errors`, тобто, якщо дані для навчання перцептрону не є правильними (початковими, незмінними), користувач, базуючись на отриманих помилках, має змогу змінити дані для навчання перцептрону. Для цієї можливості і був створений клас `Perceptron`, а не використана вже готова програма чи бібліотека, бо такої можливості при навчанні нейронних мереж зазвичай не надають. При закінченні навчання перцептрон користувачу об'єкта класу `Perceptron` рекомендується викликати функцію `release_buffer`, що вивільняє усі ресурси, які були потрібні при навчанні перцептрону, але при використанні вже не є потрібними. Об'єкт класу `Perceptron` можна перевести у бінарний вигляд та зберігати у файлах чи бінарних змінних (сериалізувати); це надає змогу користувачеві відтворювати вже навчений перцептрон без витрачання часу на навчання. Сериалізацію реалізують функції `write_to_stream` і `from_stream`;

5. *GammaUnit* (гамма-юніт) прихована від користувача частина програми, але дуже важлива. Відповідає за згортку ряду, реалізує принцип гамма-пам'яті. Має змогу змінювати ваги гамма-пам'яті відповідно до величини помилки, що йому передалась. Цей логічний блок навчається у зв'язці з перцептором: перцептрону передають данні на вхід, серед яких є значення згортки ряду від конкретного гамма-юніту; перцептрон повертає значення помилок вхідних даних; гамма-юніту змінює

значення свого вагу або накопичує градієнт — залежить від кількості патернів;

6. *GammaNN* обгортає *Perceptron* для використання з часовими рядами, надає і реалізує інтерфейс для навчання перцептронів та гамма-юнітів, можливість контролювати кількість гамма-юнітів (*units*) і кількість вільних входів (*trace_size*), що бачать останні значення часового ряду. Має функцію *learn* для навчання, куди передаються номери елементів часового ряду, що є патернами для навчання. Має оператор *[]*, для отримання об'єкту часового ряду за його номером. Оптимізує використання *[]* таким чином, що запам'ятовує усі елементи ряду (елементи початкового ряду також), що були колись отримані. Ця оптимізація є вдалою, бо для визначення наступних значень часового ряду треба звертатися до попередніх, котрі треба було б теж обчислювати, якщо б їх не запам'ятали, і так доки не дійдемо до елементів початкового ряду. Як і *Perceptron*, *GammaNN* можна серіалізувати таким самим чином: функції *write_to_stream* і *from_stream*;
7. Тестування підпрограм (*UnitTests*). Для тестування *Perceptron* використовувалися навчальні дані задачі XOR та інші. Для тестування *GammaNN* використовувалися навчальні дані: послідовність числа 1, послідовність нулів та одиниць, двовірна послідовність нулів та одиниць, зростаюча послідовності чисел за кроком 1.

Залежність бібліотеки від середовища зумовлена тим, що було задумано створити бібліотеку для мови програмування **R** на якій був написаний інтерфейс.

Є такі функції, що експортуються у середовище мови **R**:

1. *learn* приймає на вхід початковий часовий ряд і параметри на для навчання *GammaNN*; віддає *NNptr* - вказівник на об'єкт класу *GammaNN*. З середовища мови **R** неможливо звертатися до членів об'єкту *GammaNN*, тому для нього цей вказівник виконує роль

дескриптора, який потрібно передавати в експортованій бібліотекою інтерфейс користування.

2. *get_series* приймає NNptr і номери об'єктів часового ряду; віддає масив елементів часового ряді, номери яких прийшли на вхід.
3. *get_series_length* приймає NNptr; віддає кількість елементів ряду, що вже визначені в об'єкті GammaNN.
4. *to_GammaNN* приймає строку (сериалізований GammaNN об'єкт); віддає NNptr.
5. *to_str* обернено до *to_GammaNN*
6. *create_from_file* приймає шлях до файлу, де серіалізований GammaNN об'єкт; віддає NNptr.
7. *write_to_file* приймає NNptr, путь до файлу, куди серіалізувати GammaNN об'єкт.

2.3 Інструкція користувача

Розроблено програмний продукт, призначений для розв'язання задач прогнозування часових рядів. Алгоритм створений на мові C++, інтерфейс користувача створений на мові **R** з використанням програмного пакету **Shiny**.

Для запуску програми необхідно встановити на комп'ютер інтерпретатор мови **R** та середовище розробки **RStudio**.

Щоб почати роботу з програмою, необхідно запустити файл **UI.Rproj** і з'явиться головне вікно програми (Рисунок 2.1.)

Ви бачите інтерфейс:

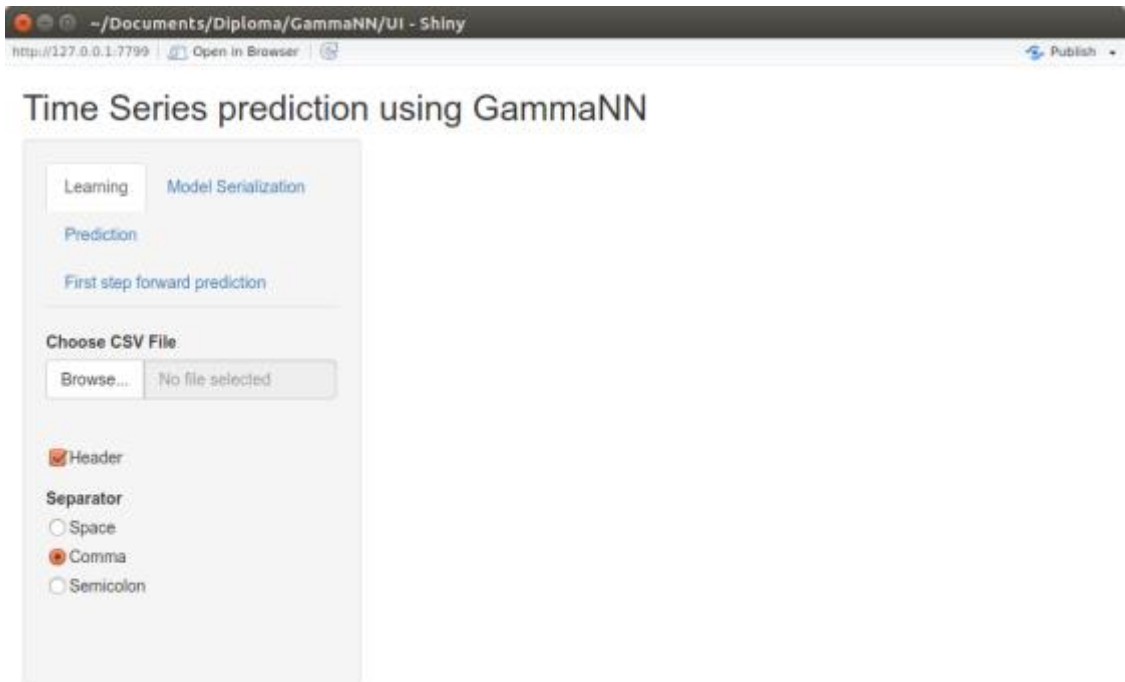


Рисунок 2.1 – Головна сторінка програми

Бачимо три вкладки: *Learning*, *Model Serialization* та *Prediction*.

У вкладці ***Learning*** необхідно вибрати текстовий або *.csv файл з даними часового ряду.

Галочка ***Header*** ставиться тоді, коли у файлі є назви стовпців.

Separator відповідає за те, який символ сприймається як розділовий знак:

- ***Space*** – пробіл,
- ***Comma*** – кома,
- ***Semicolon*** – крапка з комою.

Щоб обрати файл, треба натиснути кнопку ***Browse***. Відкриється діалогове вікно для вибору файлу.

Треба обрати файл:

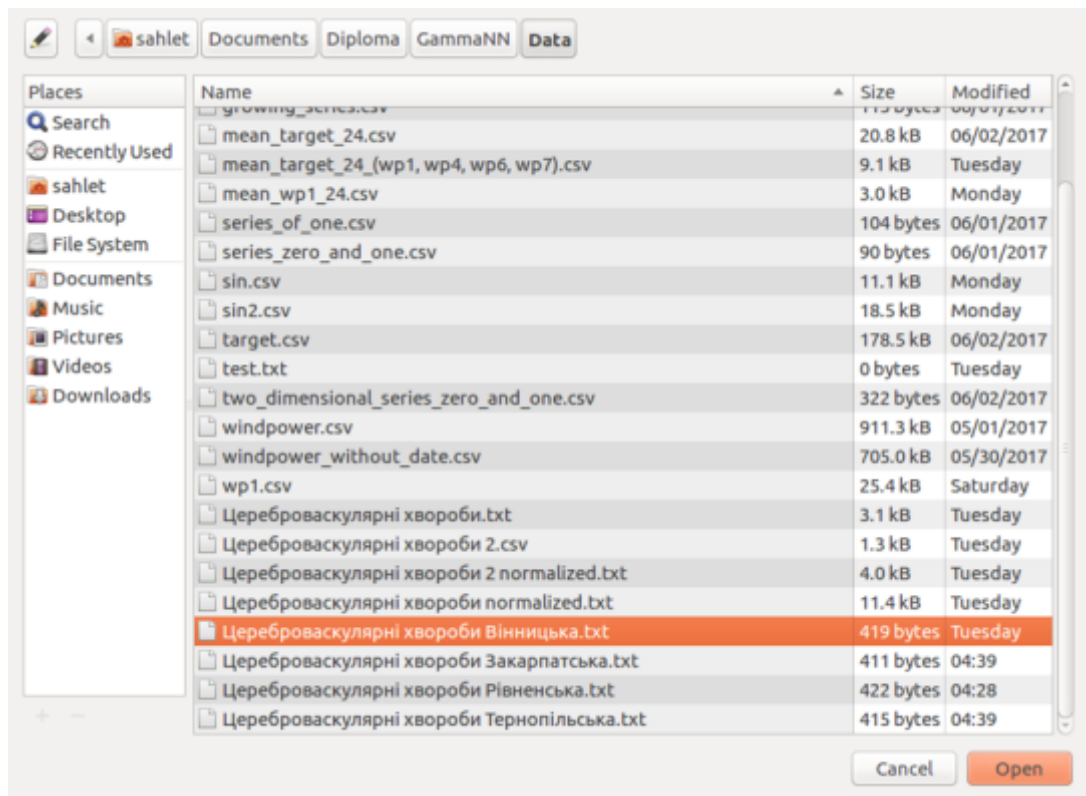


Рисунок 2.2 – Обираємо файл з часовим рядом

Після відкриття файлу стає доступним інтерфейс для навчання моделі.

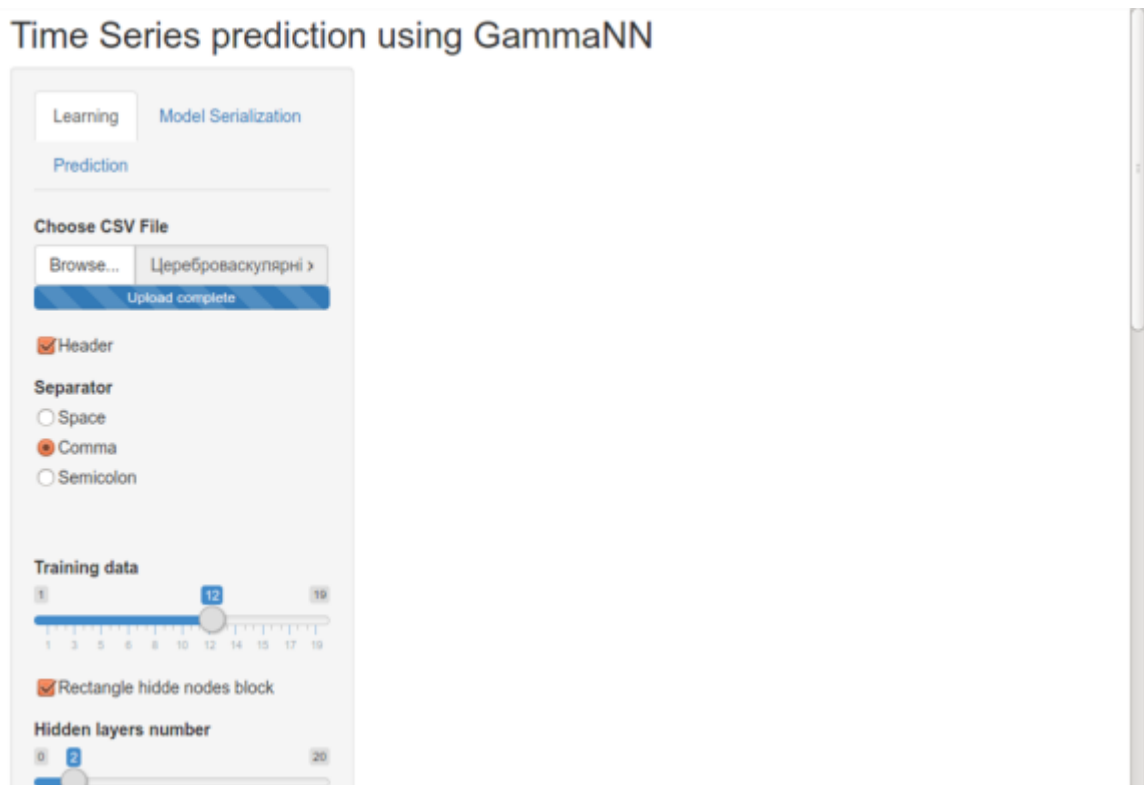


Рисунок 2.3 – Інтерфейс навчання моделі (1)

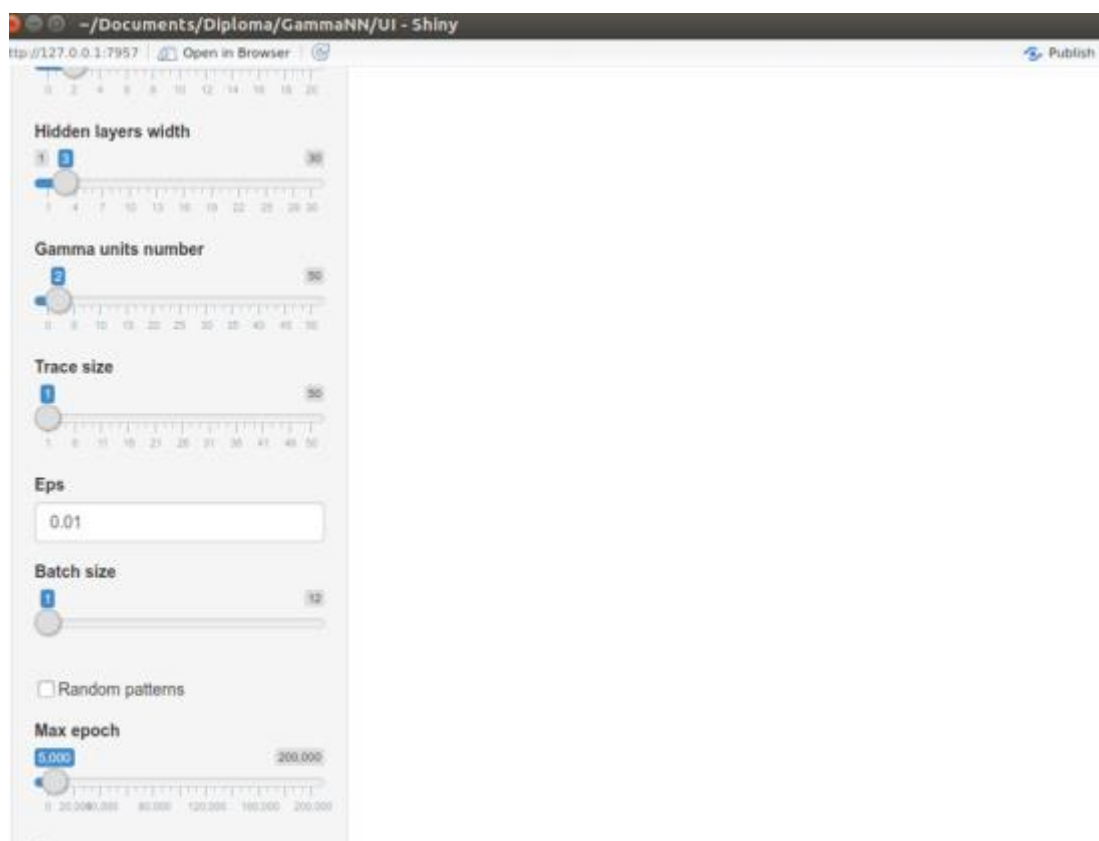


Рисунок 2.4 – Інтерфейс навчання моделі (2)

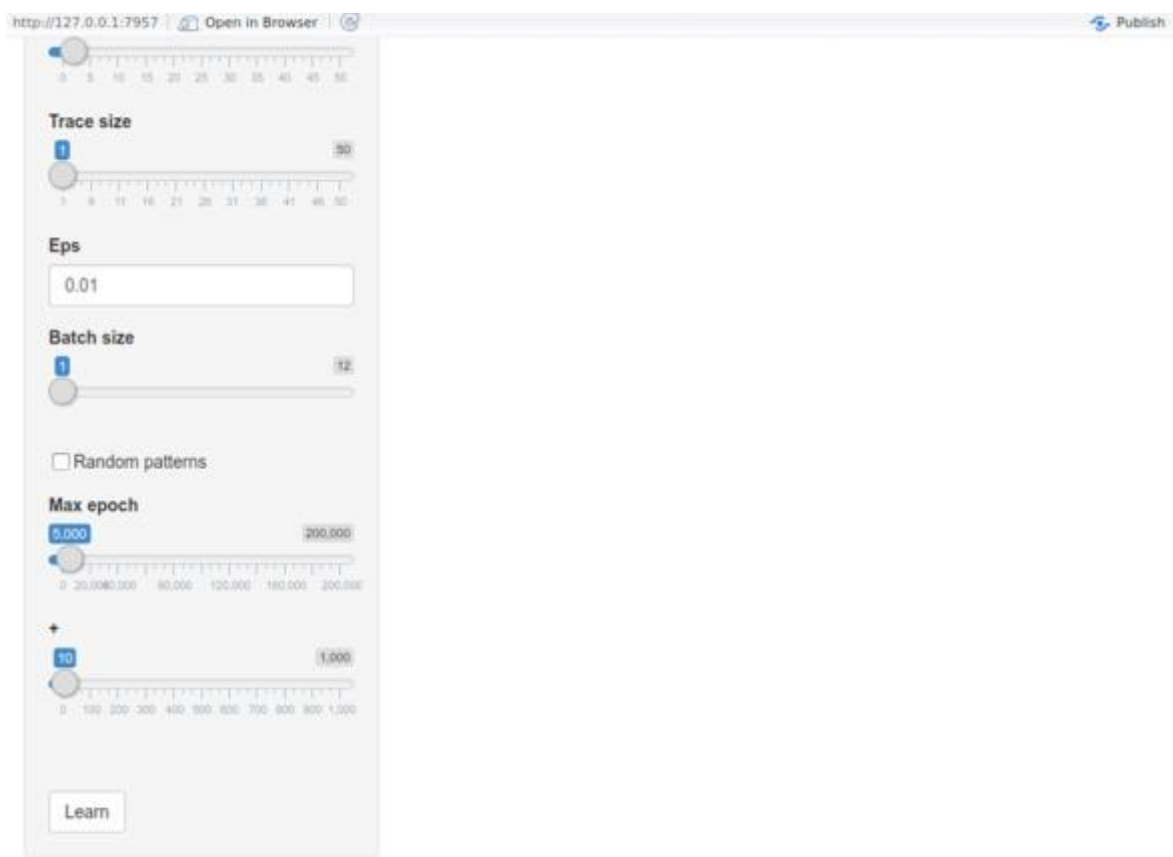


Рисунок 2.5 – Інтерфейс навчання моделі (3)

Training data – встановлює кількість даних, які будуть використовуватись у моделі навчання (від 50% до 80% усіх даних);

Rectangle hide nodes block – задання прямокутного масиву прихованих вузлів. Якщо прибрати галочку, то у **Hidden layers** записується через кому кількість прихованих вузлів у відповідному шарі (кількість чисел = кількість прихованих вузлів).

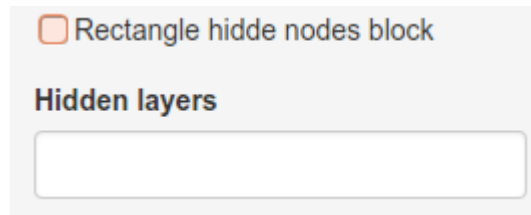


Рисунок 2.6 – Hidden layers

Hidden layers number – кількість прихованих шарів;

Hidden layers width – ширина прихованого шару;

Gamma units number – кількість гамма-блоків;

Trace size – кількість останніх видимих елементів ряду;

Eps – параметр зупинки алгоритму навчання;

Batch size – кількість елементів у навчальній ітерації;

Random patterns – випадковим чином обираються елементи для навчання;

Max epoch – максимальна кількість епох;

+ – додаткова кількість епох.

Для запуску навчання моделі потрібно натиснути **Learn**.

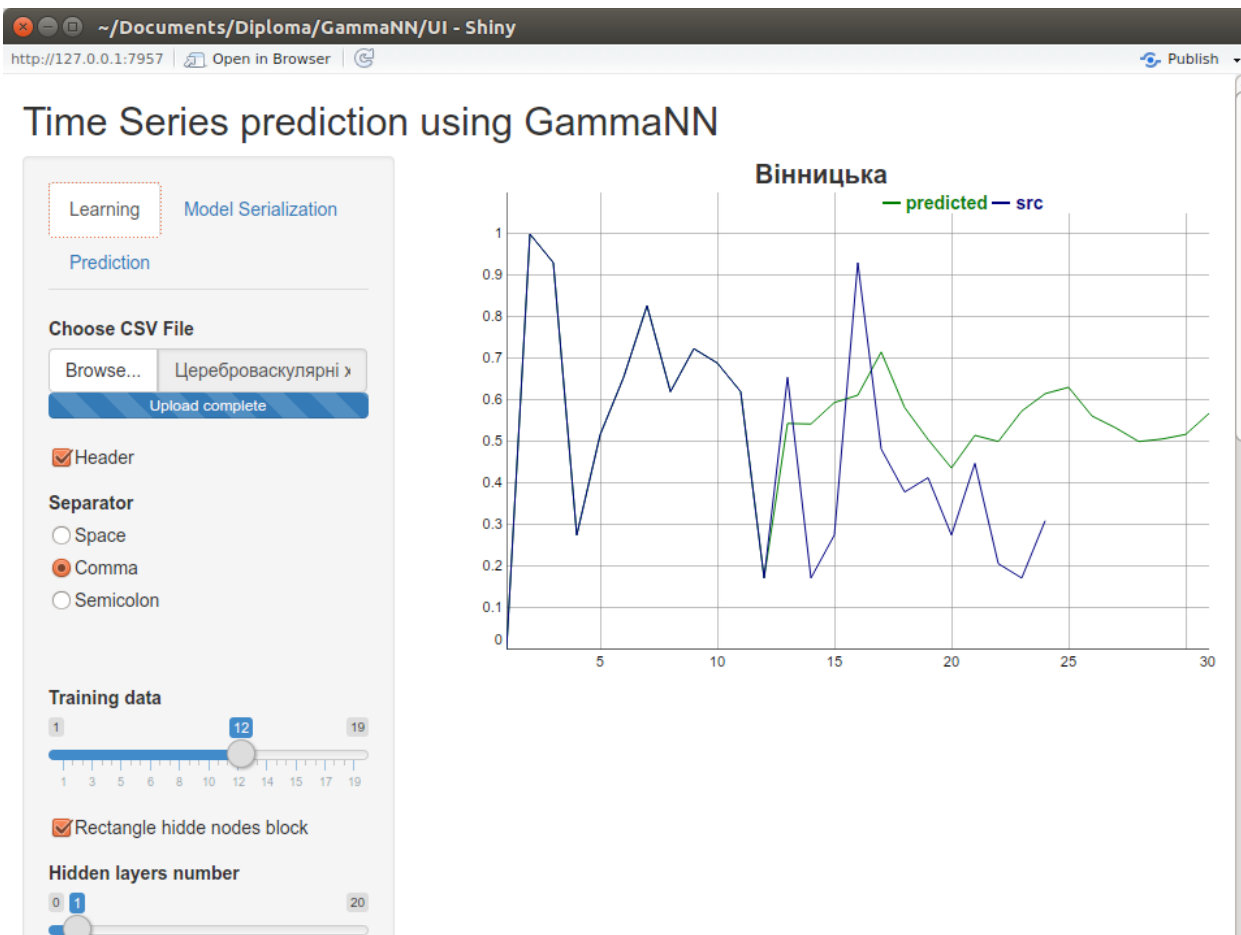


Рисунок 2.7 – Результат навчання моделі

У вкладці **Model Serialization** можна обрати або зберегти файл з моделлю.

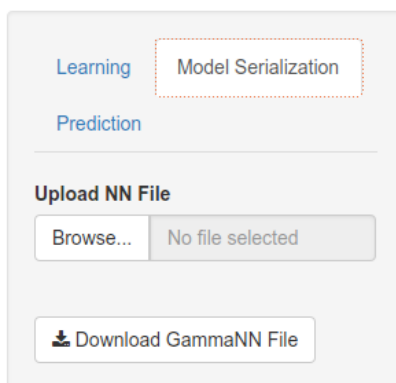


Рисунок 2.8 – Вкладка Model Serialization

У вкладці **Prediction** з'являться показники якості прогнозування:

Рисунок 2.9 – Вкладка Prediction

Range – проміжок на якому брати номери елементів з ряду;

Mean relative error (MRE) – середня відносна помилка;

RMSE – *root mean squared error* – корінь середньоквадратичної помилки.

MRE і RMSE обчислюються на всьому проміжку Range, тому, якщо треба дізнатися значення цих показників для прогнозування на 1 крок вперед, необхідно перейти у вкладку ***First step forward prediction***.

Інтерфейс, що має підпис «Choose CSV File to set src data», для обрання файлу є аналогічним до того, що надає змогу обрати файл для навчання. Файл вибраний за допомогою цього інтерфейсу буде використовуватися для порівняння прогнозу із рядом що знаходиться у файлі (буде сприйматися як початковий). Розмірність ряду має бути така сама, що і в моделі, яка завантажена. Ця функціональна можливість використовується для моделей завантажених з файлу.

У вкладці **First step forward prediction** з'явилася оцінка прогнозування на один крок вперед і елемент управління кількістю прогнозів:

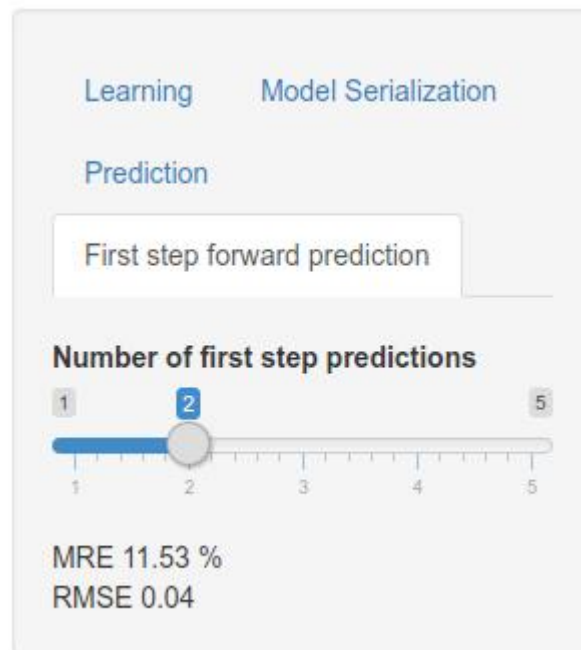


Рисунок 2.10. – Вкладка First step forward prediction

При зміні параметра **Number of first step prediction** малюється графік, що показує результати прогнозування на один крок вперед. Береться стільки точок, скільки вказано у параметрі:

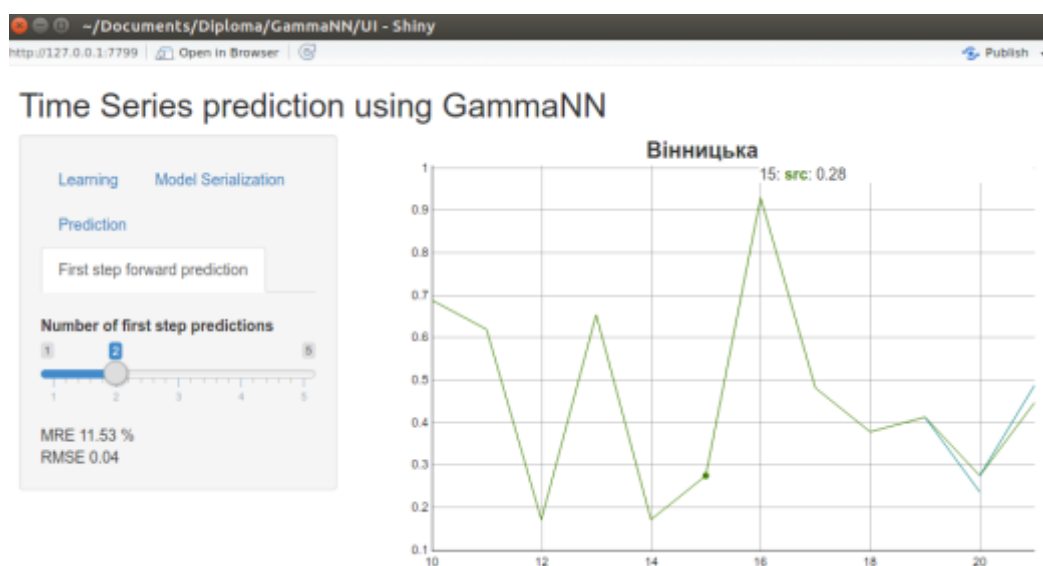


Рисунок 2.11 – Прогнозування на один крок вперед

3 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНИХ ЕКСПЕРИМЕНТІВ

3.1 Прогнозування первинної інвалідності за цереброваскулярними хворобами (на 10 тисяч населення).

Мають місце дані моніторингу первинної інвалідності в Україні за 24 роки (1992-2015). Для кожної адміністративної території, хвороби та типу населення дані представляють собою часовий ряд вигляду $[x_t; t = 1, n]$, де x_t - значення первинної інвалідності на 10 тисяч населення внаслідок хвороби x , зафіксоване у t -му році на певній адміністративній території для однієї з верств населення; n - кількість років, упродовж яких проводиться моніторинг (у даному випадку $n = 24$).

Ставиться задача прогнозування первинної інвалідності на наступний рік.

Прогнозування первинної інвалідності за цереброваскулярними хворобами в Вінницькій області.

Для навчання цієї моделі були задані такі параметри (Рисунок 3.1): розмір навчального ряду 19, приховані вузли - 1 шар 4 вузли, 4 гамма-юніти, модель бачить 5 останніх значень ряду; параметри зупинки навчання: $\epsilon = 0.01$, максимальна кількість епох навчання 20000.

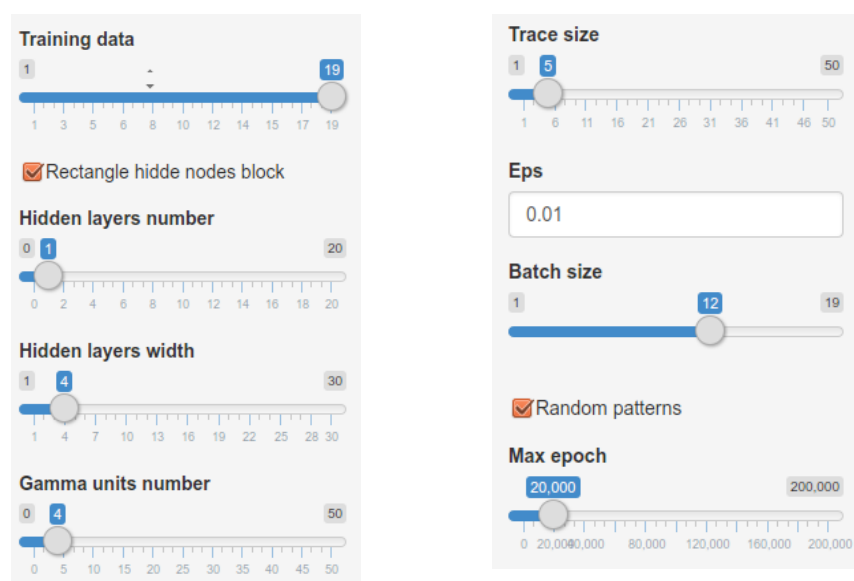


Рисунок 3.1 – Параметри моделі для прогнозування первинної інвалідності за цереброваскулярними хворобами в Вінницькій області

Параметри були підбрані таким чином, щоб модель прогнозувала найкраще. Аналогічна модель, але без гамма-юнітів виявилася гіршою.

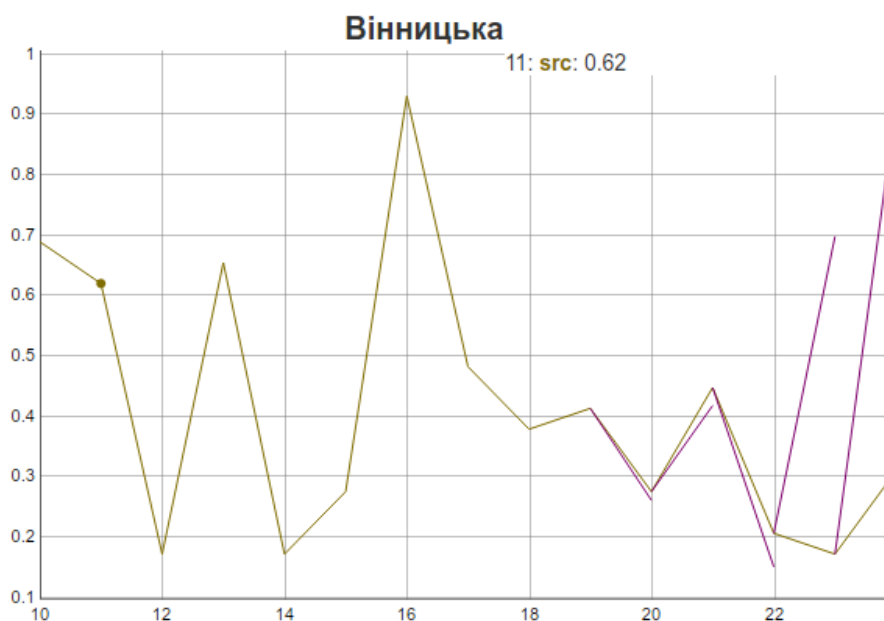


Рисунок 3.2 – Прогнозування первинної інвалідності за cerebroваскулярними хворобами в Вінницькій області

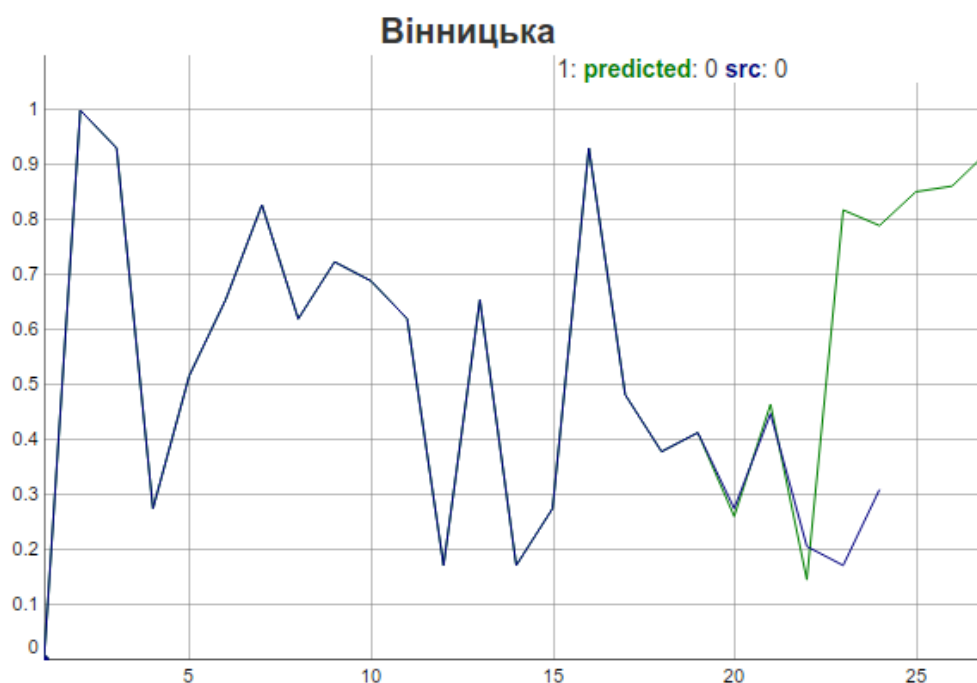


Рисунок 3.3 – Прогнозування первинної інвалідності за cerebroваскулярними хворобами в Вінницькій області на один крок вперед (5 точок)

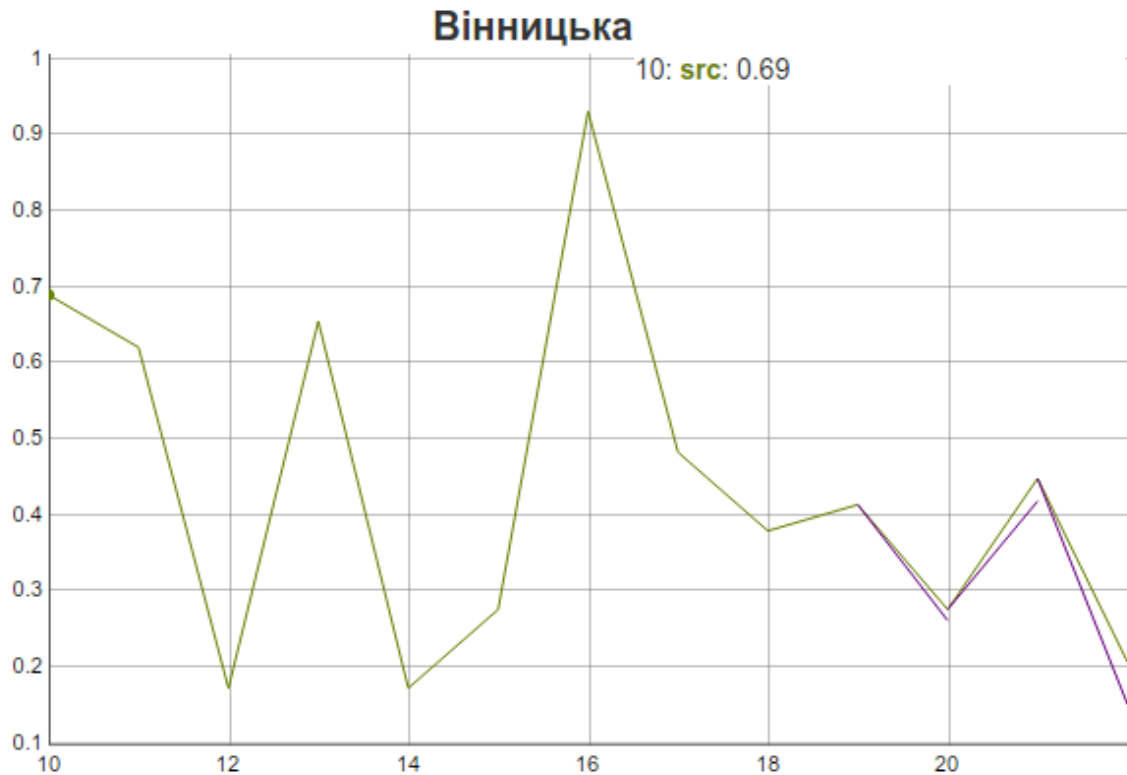


Рисунок 3.4 – Прогнозування первинної інвалідності за цереброваскулярними хворобами в Вінницькій області на один крок вперед (3 точок)

Прогнозування на один крок вперед для 5 точок має показники якості $RMSE = 0,36$ і $MRE=107,88\%$, а для 3 точок $RMSE = 0,037$ і $MRE=12,93\%$. Така різниця між показниками $RMSE$ і MRE для 5 і 3 точок зумовлена тим, що ряд для навчання моделі має 19 елементів, тобто мало.

Прогнозування первинної інвалідності за цереброваскулярними хворобами в Тернопільській області.

Для навчання цієї моделі були задані такі параметри (Рисунок 3.7): розмір навчального ряду 19, приховані вузли - 1 шар 4 вузли, 0 гамма-юнітів, модель бачить 5 останніх значень ряду; параметри зупинки навчання: $eps = 0.01$, максимальна кількість епох навчання 25000.

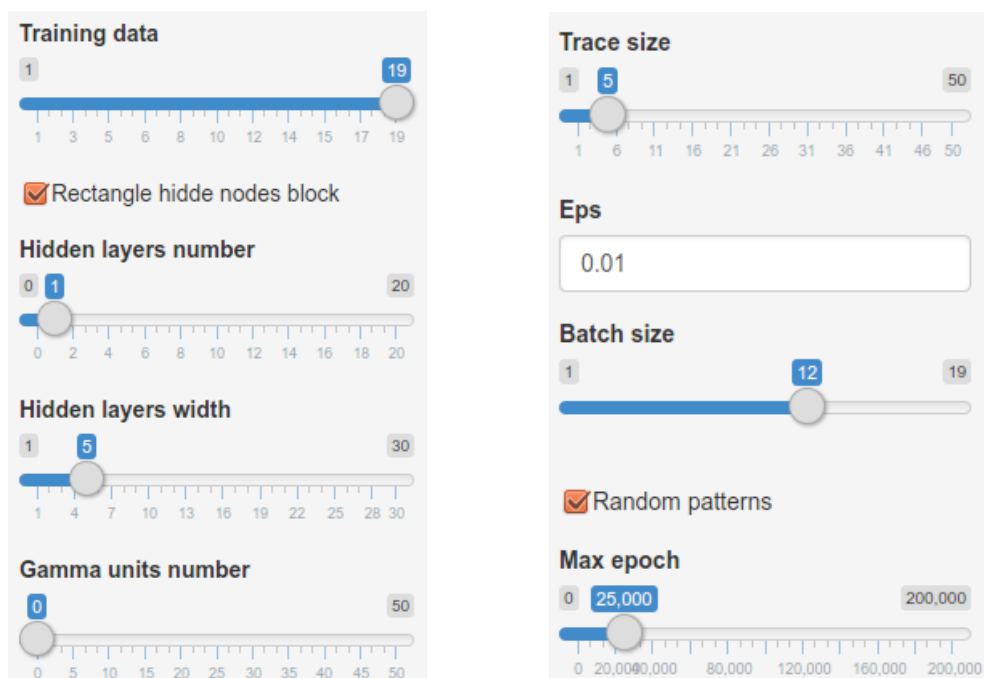


Рисунок 3.5 – Параметри моделі для прогнозування первинної інвалідності за цереброваскулярними хворобами в Тернопільській області

Параметри були підібрані таким чином, щоб модель прогнозувала найкраще.

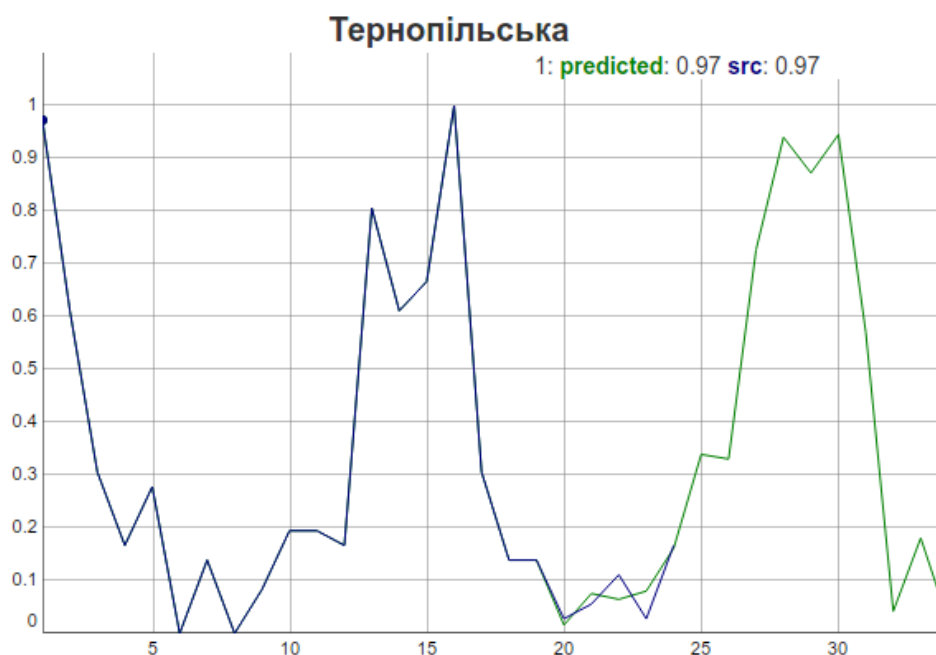


Рисунок 3.6 – Прогнозування первинної інвалідності за цереброваскулярними хворобами в Тернопільській області



Рисунок 3.7 – Прогнозування первинної інвалідності за цереброваскулярними хворобами в Тернопільській області на один крок вперед (5 точок)

Прогнозування на один крок вперед для 5 точок має показники якості $RMSE = 0,029$ і $MRE=54,14\%$.

3.2 Прогнозування щоденної кількості проданого товару.

Прогнозування DAYSALLES_858 з гамма-юнітами, не здатними до навчання.

Для навчання цієї моделі були задані такі параметри (Рисунок 3.10): розмір навчального ряду 59, приховані вузли — 1-й шар 30 вузли, 2-й шар 3 вузли, 2 гамма-юніти, модель бачить 14 останніх значень ряду; параметри зупинки навчання: $\text{eps} = 0.05$, максимальна кількість епох навчання 30000.

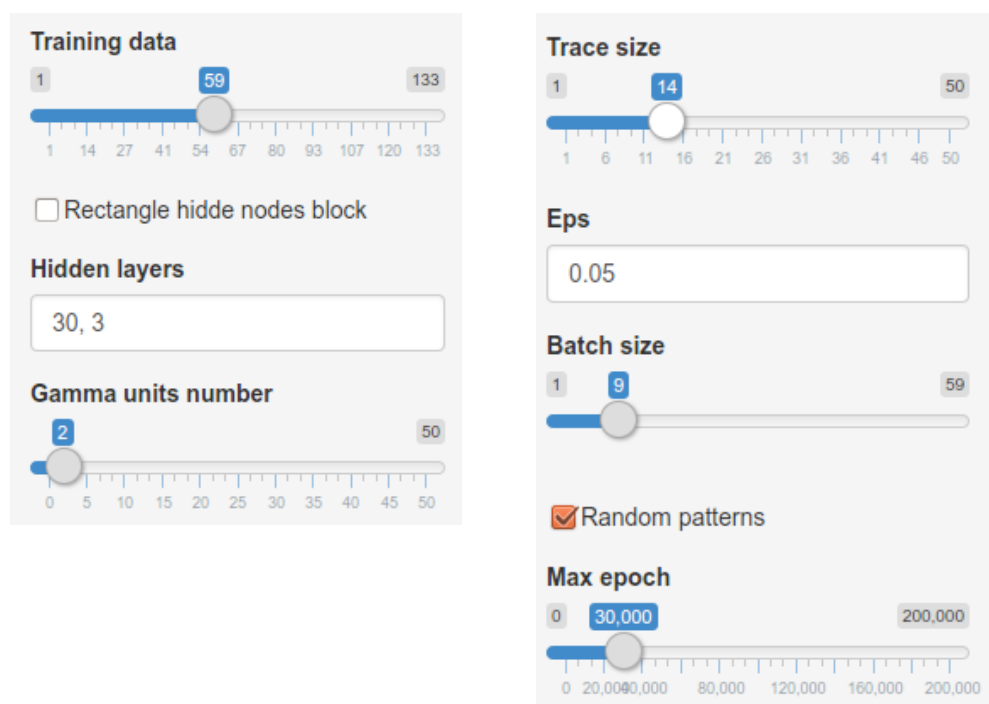


Рисунок 3.8 – Параметри моделі для прогнозування DAYSALLES_858

Параметри були підібрані, тобто з цими параметрами модель прогнозує найкраще.

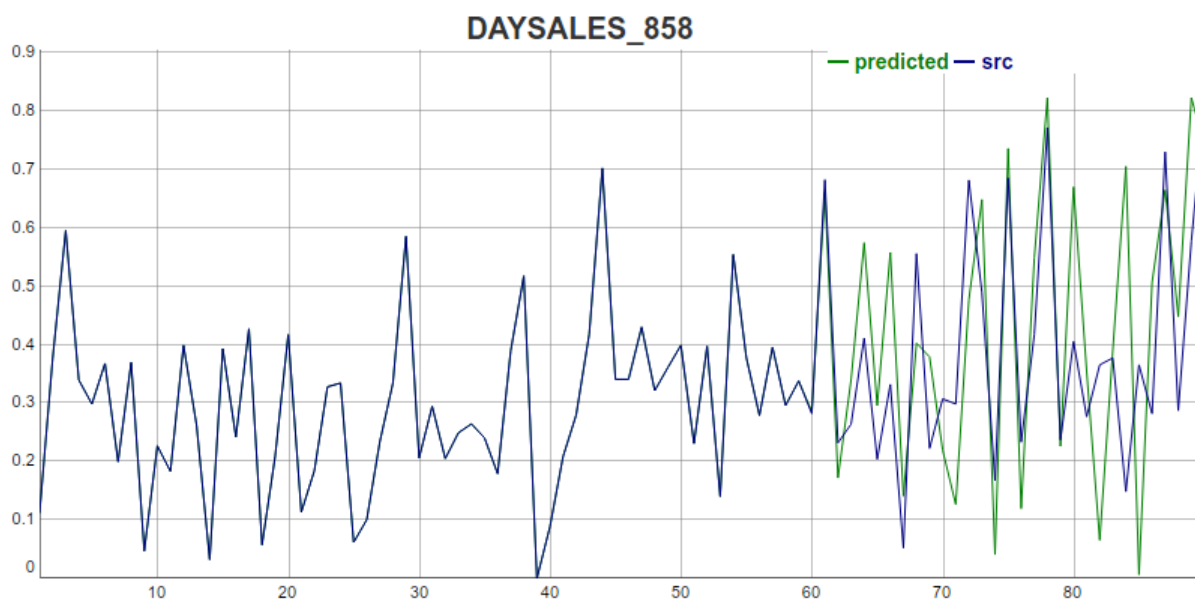


Рисунок 3.9 – Прогнозування DAYSALES_858

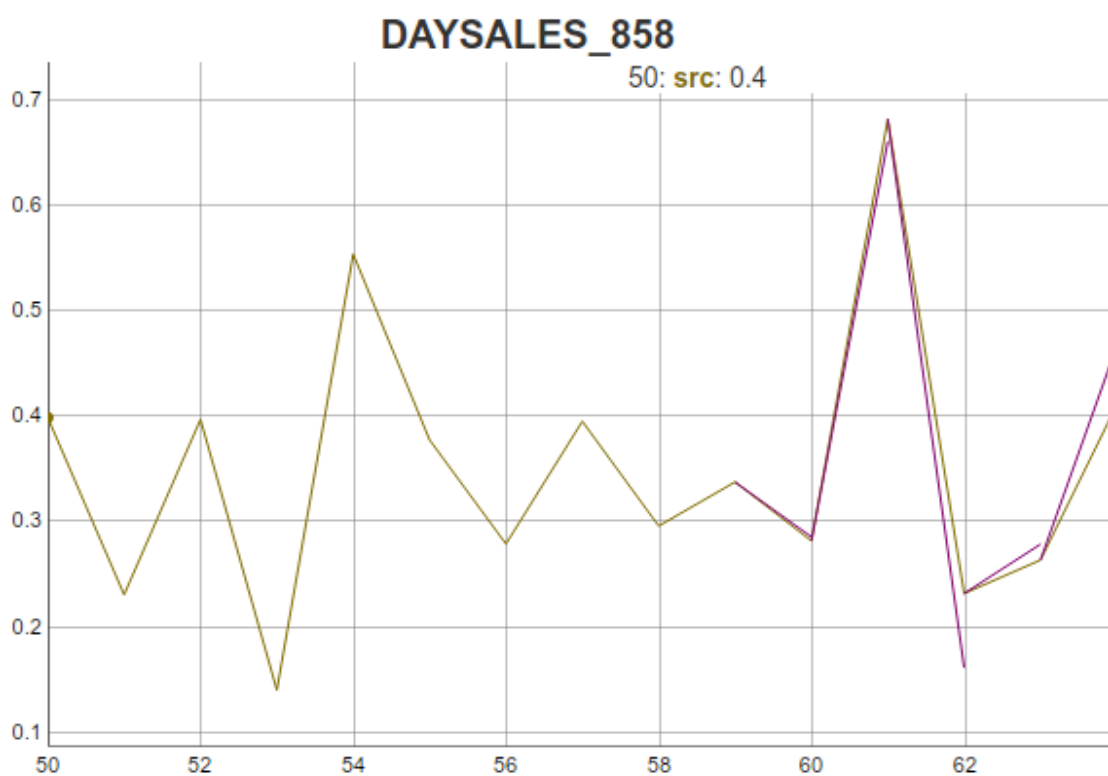


Рисунок 3.10. – Прогнозування DAYSALES_858 на один крок вперед (5 точок)

Прогнозування на один крок вперед для 5 точок має показники якості $RMSE = 0,042$ і $MRE=10,81\%$.

Прогнозування DAYSALES_858 з гамма-юнітами, здатними до навчання.

Для навчання цієї моделі були задані такі самі параметри, як і для моделі прогнозування DAYSALES_858 з гамма-юнітами не здатними до навчання (Рисунок 3.10). Відрізняється ця модель тим, що при її навчанні використовувалися гамма-юніти здатні до навчання.

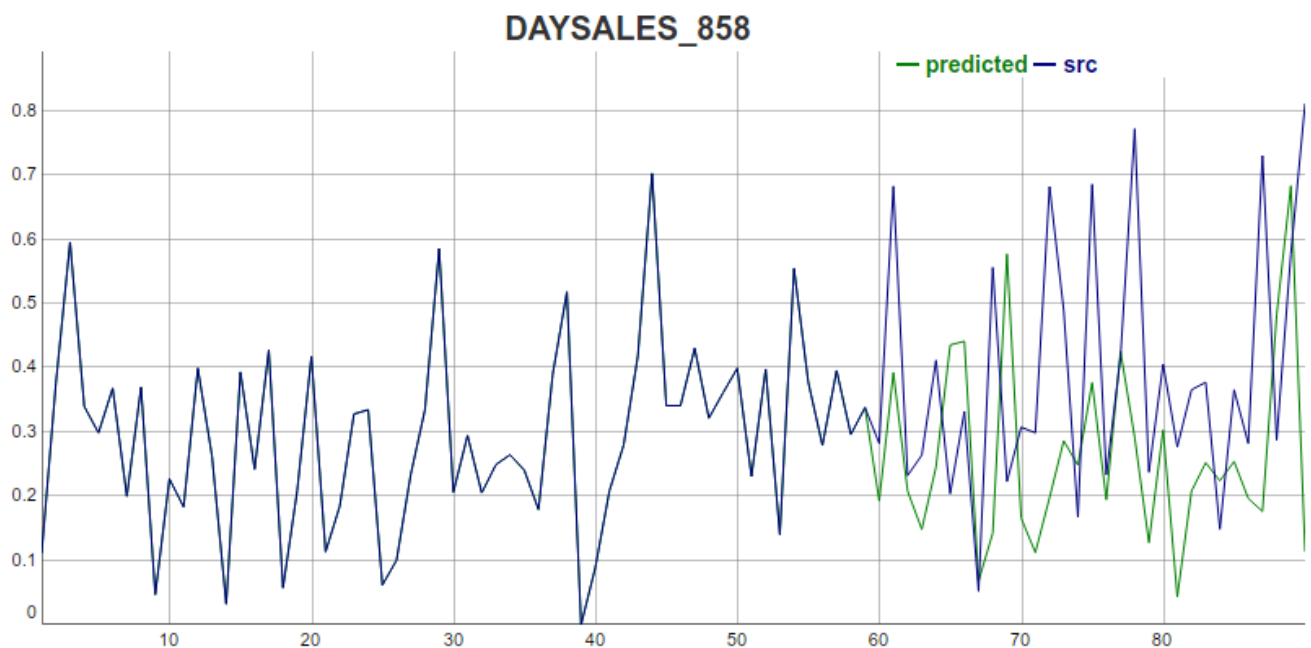


Рисунок 3.11 – Прогнозування DAYSALES_858 з гамма-юнітами здатними до навчання

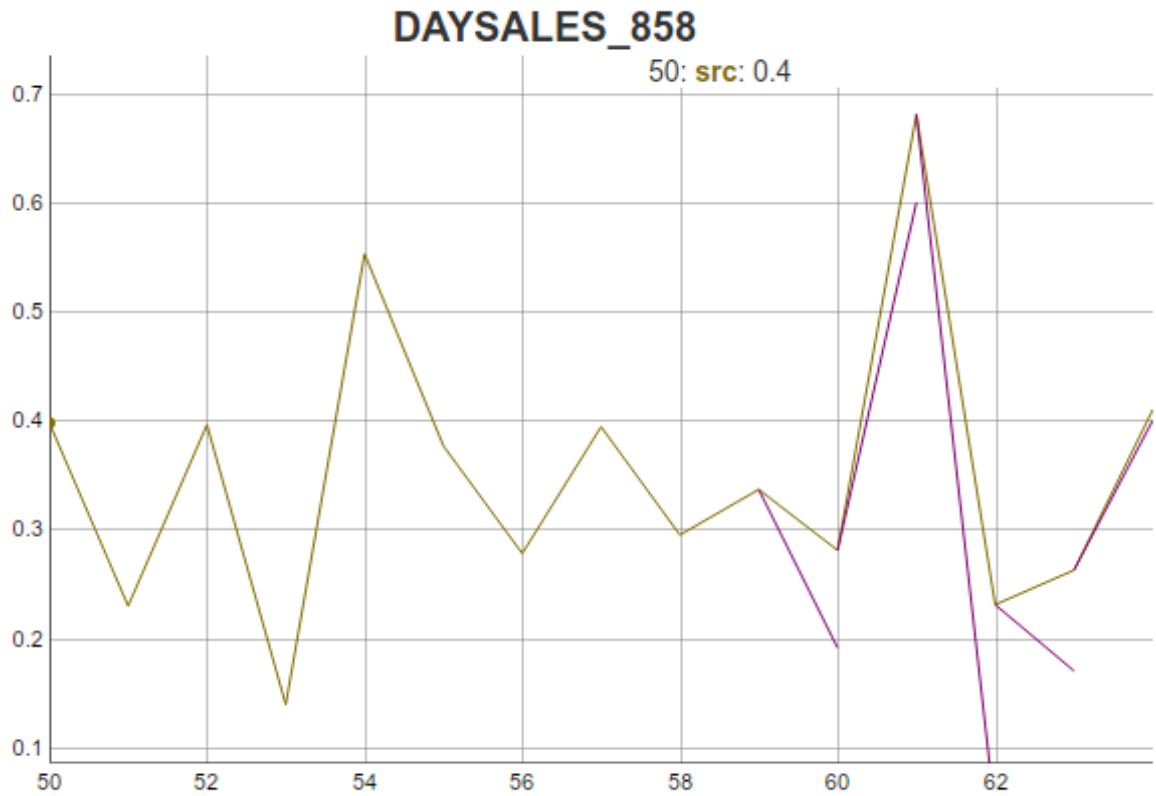


Рисунок 3.12 – Прогнозування DAYSALES_858 на один крок вперед (5 точок) з гамма-юнітами здатними до навчання

Прогнозування на один крок вперед для 5 точок має показники якості $RMSE = 0,11$ і $MRE=32,74\%$.

ВИСНОВКИ

Під час виконання дипломної роботи було проведено огляд існуючих методів прогнозування часових рядів, у тому числі розглянуто визначений постановкою задачі метод на основі відновлення регресії. Даний метод базується на відновленні регресійної залежності значення ряду в наступний момент часу від минулих значень. Для відновлення регресії обрано нейронні мережі.

Було розроблено програмне забезпечення, яке дозволяє проводити прогнозування часових рядів з використанням нейронних мереж. Програмне забезпечення надає користувачу можливості:

1. Завантажувати файл з початковим часовим рядом.
2. Задавати параметри навчання моделі.
3. Навчати модель.
4. Отримувати графіки прогнозів на один і на декілька кроків вперед.
5. Отримувати оцінки якості прогнозування: RMSE і MRE.
6. Зберігати отриману модель у бінарний файл.
7. Завантажувати модель з файлу.

За допомогою створеного програмного забезпечення проведено прогнозування первинної інвалідності в Україні за 24 роки (1992-2015) (на 10 тисяч населення). Одержані результати свідчать:

- Якщо початковий ряд не є складним для прогнозування, нейромережева модель здатна прогнозувати ряд навіть якщо навчальний ряд має невелику кількість елементів.
- Гамма-юніти можуть поліпшити модель.

Програмне забезпечення також було застосовано для прогнозування щоденної кількості проданого товару. Одержані результати свідчать:

- Показники якості прогнозування моделі з гамма-юнітами не здатними до навчання виявилися кращими ніж показники якості прогнозування моделі з гамма-юнітами здатними до навчання. Використання гамма-

юнітів здатних до навчання не покращує якість прогнозування, а навпаки може її погіршити.

Загальні висновки щодо обчислювальних експериментів:

- Гамма-юніти треба застосовували для відновлення моделей з великою кількістю початкових даних, бо для невеликих масивів даних гамма-юніти можуть ускладнити модель і призвести до перенавчання.
- Нейронні мережі в поєднанні з гамма-юнітами і без них є хорошим апаратом для відтворення і прогнозування часових рядів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Лоскутов А.Ю. Основи теорії складних систем./А.Ю. Лоскутов, А.С. Міхайлов.//Регулярна і хаотична динаміка. – М.,2007. – 620 с.
2. Хайкін С. Нейронні мережі повний курс./С. Хайкін – М., 2006. – 1104 с.
3. Безручко Б.П. Математичне моделювання та хаотичні часові ряди./ Б.П. Безручко, Д.О. Смирнов – Саратов: «Коледж», 2005 – 320 с.
4. Малінецький Г.Г. Сучасні проблеми нелінійної динаміки./ Г.Г. Малінецький, А.Б. Потапов. – М., 2000. – 360 с.
5. Лоскутов А.Ю. Проблеми нелінійної динаміки. Локальні методи прогнозування часових рядів./А.Ю. Лоскутов, О.Л. Котляров, І.А. Істомін, Д.І. Журавлев. – М., 2002. – 36 с.
6. Лоскутов А.Ю. Аналіз часових рядів. Курс лекцій./А.Ю. Лоскутов – Фізичний факультет МГУ – 113 с.