**ДИПЛОМНА РОБОТА**

**за першим (бакалаврським) рівнем вищої освіти**

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

|  |  |
| --- | --- |
| **Виконавець**: | студент ІV курсу, групи ПМ-13-1  спеціальності 6.040301 -  «прикладна математика»  Кривоносов Олександр Дмитрович  (прізвище та ініціали) |
| **Керівник**: | д.ф.-м.н., проф.,проф.  Кузьменко В.І.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) |
| **Рецензент:** | д.ф.-м.н., проф., зав. каф. ПКТ  Гук Н.А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) |

ДНІПРОПЕТРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ім. О. ГОНЧАРА

Факультет *прикладної математики* Кафедра\_\_\_\_\_\_\_\_\_*ОМ та МК\_\_\_\_\_\_\_\_*

Рівень (освітньо-кваліфікаційний рівень) *перший (бакалаврський)*

Напрям підготовки \_\_\_\_\_\_\_\_\_\_\_*6.040301 — Прикладна математика*\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Спеціальність\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЗАТВЕРДЖУЮ

Зав. кафедрою ОМ та МК

\_\_\_\_Турчина В.А.\_\_\_\_

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2017 р.

ЗАВДАННЯ

на дипломну роботу студентові

Кривоносову Олександру Дмитровичу

(прізвище, ім’я по батькові)

1. Тема роботи

*Прогнозування первинної інвалідності в Україні з використанням методів регресійного аналізу*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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 багатовимірними даними — часовими рядами*

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

*1.Огляд задач прогнозування часових рядів. 2.Розглянути методи, алгоритми прогнозування. 3.Розробити програмне забезпечення, у якому реалізувати ці методи, алгоритми; провести його тестування*. *4. Провести обчислювальні експерименти*.

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

*Графічне зображення початкового часового ряду, та 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 станiв дипломної роботи | Термiн виконання станiв роботи | Примітка |
| 1 | Огляд задачі прогнозування часових рядів та методами її розв’язання, опрацювання літературних джерел | 31.03.2017 |  |
| 2 | Розробка алгоритму прогнозування часових рядів | 15.04.2017 |  |
| 3 | Розробка програмного продукту, що реалізує алгоритм прогнозування | 20.04.2017 |  |
| 4 | Тестування програмного продукту | 11.05.2017 |  |
| 5 | Проведення обчислювальних експериментів | 25.05.2017 |  |
| 6 | Оформлення пояснювальної записки | 01.06.2017 |  |
| 7 | Представлення випускної роботи на кафедру | 12.06.2017 |  |
| 8 | Захист роботи в ДЕК | 16.06.2017 |  |

**Студент** Кривоносов О.Д.

(підпис) (прізвище та ініціали)

**Керівник роботи**  \_\_Кузьменко В.І.

(підпис) (прізвище та ініціали)

# РЕФЕРАТ

*Дипломна робота* *:\_\_ сторінок, \_\_ рисунків, \_\_ таблиць, \_\_ джерел.*

***Об’єктом дослідження*** є задача прогнозування часових рядів.

***Мета роботи:*** розробка алгоритму розв’язання задачі прогнозування часових рядів на основі штучної нейронної мережі для прогнозування первинної інвалідності.

***Методи дослідження***: методи апроксимації, методи оптимізації, методи регресійного аналізу.

***У процесі роботи*** реалізовані багатошаровий перцептрон для апроксимації функцій, гамма-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 силу вітру, даних щоденної кількості проданих товарів на торгових точках.

***В результаті роботи*** розроблено програмний продукт, призначений для розв’язання задач прогнозування часових рядів. Алгоритм створений на мові **С++**, 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.

# ANNOTATION

ЗМІС

[1. ВСТУП7](#_Toc484769981)

[2. ПОСТАНОВКА ЗАДАЧІ9](#_Toc484769982)

[3. ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ10](#_Toc484769983)

[3.1. Огляд методів прогнозування часових рядів10](#_Toc484769984)

[**3.1.1.** **Метод ковзного середнього ( moving average, MA).**10](#_Toc484769985)

[**3.1.2.** **Модель Бокса-Дженкінса (ARIMA)**11](#_Toc484769986)

[**3.1.3.** **Метод аналізу сингулярного спектру (SSA)**12](#_Toc484769987)

[**3.1.4.** **Локальна апроксимація (LA)**13](#_Toc484769988)

[3.1.5. Нейромережеве прогнозування часових рядів16](#_Toc484769989)

[4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ20](#_Toc484769990)

[4.1. Функціональні можливості та структура програми20](#_Toc484769991)

[4.2. Інструкція користувача24](#_Toc484769992)

[5. РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНИХ ЕКСПЕРИМЕНТІВ31](#_Toc484769993)

[6. ВИСНОВКИ48](#_Toc484769994)

[7. СПИСОК ЛІТЕРАТУРИ51](#_Toc484769995)

Список використаної літератури

# 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* - ряд значень будь-яких параметрів досліджуваного процесу за рівні проміжки часу.

*Скалярним часовим рядом* називається масив з чисел, що представляють собою значення деякої змінної , що спостерігається з деяким постійним кроком по часу, ; . 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 . 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 посилає 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 обсягу.

# ПОСТАНОВКА ЗАДАЧІ

Розробити та реалізувати метод, за допомогою якого розв’язується наступна задача:

***Дані на вхід:*** - часовий ряд (послідовність значень деякого показника впорядкована по даті фіксування; передбачається, що фіксування значень виконується з однаковим інтервалом), де - елементи ряду, - кількість елементів ряду;

***Вихід***: - функція прогнозування, що будується відштовхуючись від даних на вхід; за аргументи має номер елементу ряду (може бути більшим і меншим за ), а за значення має t-ий елемент цього ряду.

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

,

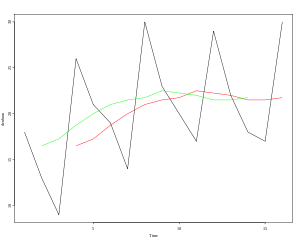
де - значення первинної інвалідності на 10000 населення внаслідок хвороби , зафіксоване у t-му році на певний адміністративній території для однієї з верств населення; - кількість років, упродовж яких проводиться моніторинг (у даному випадку ).

Ставиться задача прогнозування первинної інвалідності на наступний рік.

# ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

## 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 використовуються з даними часових рядів для згладжування короткострокових коливань і виділення основних тенденцій або циклів. Математично ковзне середнє є одним з видів згортки.

***Рис.1****. Просте ковзне середнє*

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

де - значення простого ковзного середнього в точці ,- кількість значень початкової функції для розрахунку ковзного середнього, чим ширше згладжує інтервал, тим більш плавним виходить графік функції, значення вихідної функції в точці .

На Рис.1 зображені вихідна функція і її 2 прості ковзаючі середні по чотирьом значенням (n = 4): зелена лінія - центрування по середині інтервалу (справжній стан), червона лінія - зсув графіка вправо до останнього значенням вікна.

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

де - значення експоненціального ковзного середнього в точці , - значення експоненціального ковзного середнього в точці , - значення вихідної функції в момент часу , - 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.

### **Модель Бокса-Дженкінса (ARIMA)**

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

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

де – стаціонарний часовий ряд,

– стаціонарний часовий ряд,

– оператор різниці часового ряду порядку d (послідовне взяття d раз різниць першого порядку - спочатку від тимчасового 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 багатомірного тимчасового ряду методу головних компонент.

Спосіб перетворення одновимірного ряду в багатовимірний є «згортку» часового ряду в матрицю, що містить фрагменти тимчасового 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, нестаціонарних, 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 x K траєкторна матриця ряду Х наступним чином:



де - вектори вкладення довжини L.  Матриця Х є ганкелевою, тобто має однакові елементи на антидіагоналях

*Крок 2.* Виконується сингулярне розкладання (SVD) траєкторної матриці Х. Нехай , позначимо власні числа S, взяті у незростаючому порядку і ортонормовану систему власних векторів матриці S. Нехай та  *.* У цих позначеннях сингулярне розкладання траєкторної матриці Х може бути записано як

*Крок 3.* Множина всіх індексів {1,…,d} розбивається на m непересічних підмножин Нехай . Тоді результуюча матриця Х, що відповідає групі I, записується як Результуючі матриці обчислюються за групами і згруповане SVD розкладання матриці X може бути записано як .

*Крок 4*. Кожна матриця згрупованого розкладання ганкелізуется (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 тимчасовими рядами. Діагональне усереднення, застосоване до кожної результуючої матриці , виробляє відновлені ряди . Таким чином, вихідний ряд розкладається в суму m відновлених рядів:

### **Локальна апроксимація (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 ≥ 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) тимчасовими 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 тимчасової ряд, що містить N значень, в матрицю затримок:



Розмірність матриці X визначається кількістю затримок p.

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

де a - вектор параметрів уявлення.

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

*Крок 2.* Вибір найближчих сусідів - виділення локальної підобласті фазового простору. Будемо вважати сусідніми до стартового вектору вектори, що задовольняють умові:



де ../Desktop/Снимок%20экрана%202017-06-08%20в%2017.35.10.png\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

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

Даний етап алгоритму - головна відмітна ознака методу і найважливіший його етап. Вибір "хороших" сусідів визначає якість прогнозу.

*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 ) оцінюються методом найменших квадратів (МНК). Це найпоширеніший і найбільш ефективний метод. Оцінка за МНК для вектора , позначимо її як знаходиться з умови:



Однак в більшості випадків застосування МНК в чистому вигляді неможливо через вирожденість матриці факторів, що є наслідком взаємної близькості сусідів. Тому зазвичай застосовується так званим моє 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):

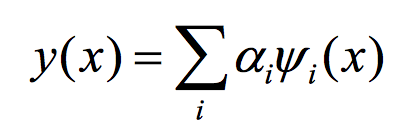
Таким чином, пройшовши три описаних кроку алгоритму LA, можна побудувати прогноз одного нового значення ряду.

## Нейромережеве прогнозування часових рядів

При вирішенні завдань прогнозування роль нейронної мережі полягає в передбаченні майбутньої реакції системи по її попередній поведінці. Володіючи інформацією про значеннях змінної x в моменти, які передують прогнозуванню:

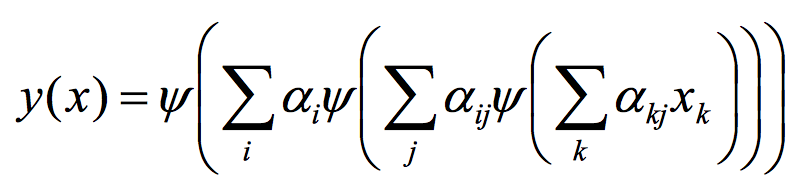
мережа виробляє рішення, яким буде найбільш ймовірне значення послідовності в поточний момент k. При цьому нейронна мережа грає роль універсального аппроксиматора функції від декількох змінних, реалізуючи нелінійну функцію: де x - це вхідний вектор, а y - реалізація векторної функції декількох змінних.

Відповідно до класичної теорії апроксимації, функції, що апроксимується представляється в наступному вигляді:



Тут набір функцій як правило, вибирається апріорі, виходячи з інтересів автора конкретного методу апроксимації і деяких властивостей даних функцій, доказуваних і використовуваних в процесі апроксимації.

Для нейронних мереж базове вираз для функції, що апроксимується виглядає дещо інакше, наприклад, для тришарової нейронної мережі з послідовними зв'язками:

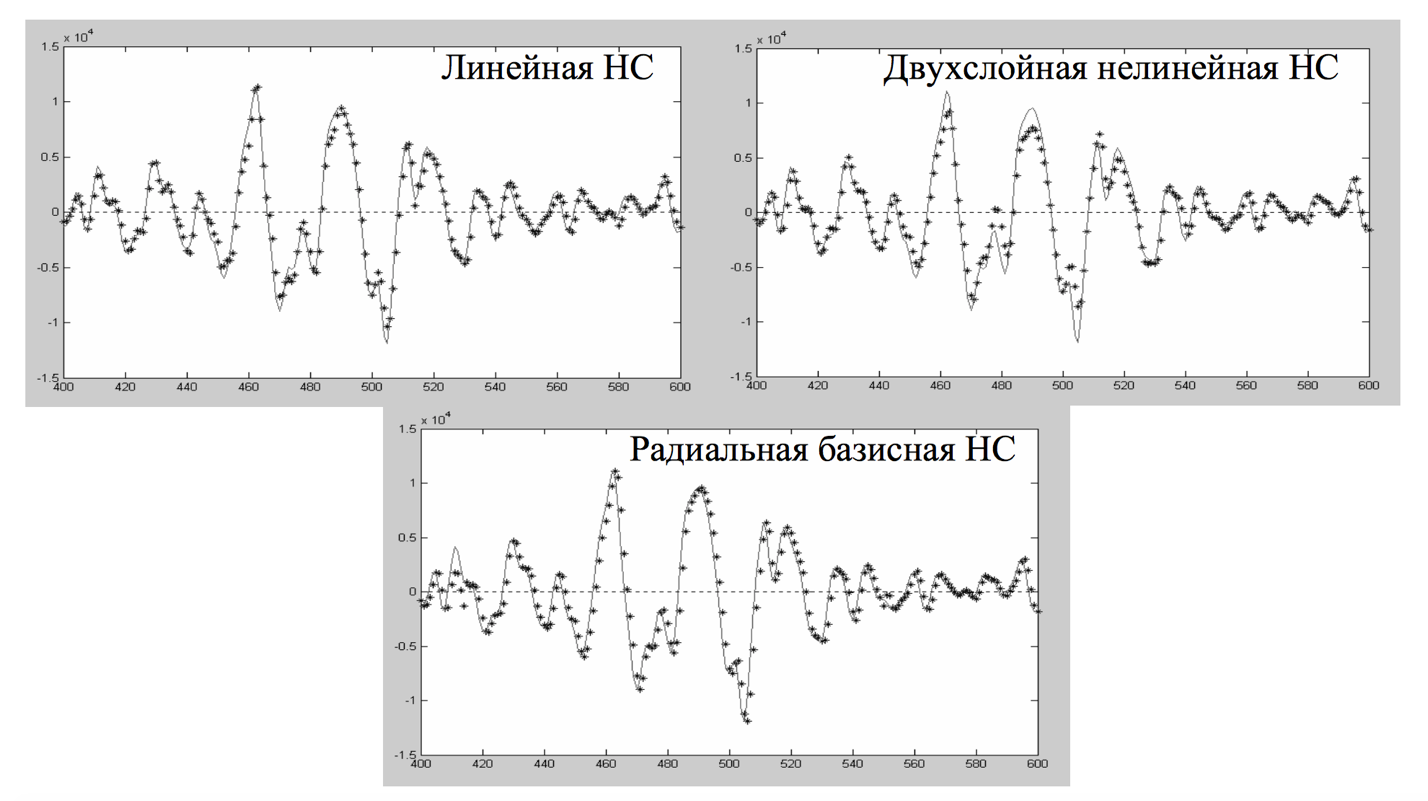


Зазвичай формування пари навчальних прикладів здійснюється за принципом «ковзного вікна»: тобто береться певний відрізок тимчасового ряду і з нього виділяється кілька спостережень, які і будуть представляти собою вхідний вектор. Значним бажаного виходу в навчальному прикладі буде наступне по порядку спостереження. Потім «ковзне вікно» зсувається на одну позицію в напрямку зростання часу, і процес формування наступної пари навчальної вибірки повторюється. Частинка тимчасового ряду для навчання нейронної мережі повинен бути дорівнює обсягу навчальної вибірки (кількості прикладів). При виборі числа входів нейронної мережі, що працює в режимі передбачення часового ряду, слід вибирати розумний компроміс між глибиною передбачення і якістю навчання, оскільки збільшення числа входів нейронної мережі відповідно зменшує кількість навчальних прикладів.

Згідно із загальноприйнятою принципом, якщо більш складна модель не дає результат краще, ніж більш проста, то з цих двох моделей слід віддати перевагу останню. Найпростішою архітектурою з нейронних мереж прямого поширення мають лінійні нейронні мережі. Будь-яка багатошарова нейронна мережа з лінійною функцією активації нейронних елементів еквівалентна одношарової лінійної нейронної мережі, тому лінійні нейронні мережі, як правило, не мають прихованих шарів. Під час своєї роботи лінійна нейронна мережа фактично примножує вектор входів на матрицю ваг, а потім до отриманого вектору додає вектор зміщення. Вихідне значення такої мережі визначається виразом:

де - вихід мережі; - матриця ваг; - вектор входів; - вектор зміщення.

Для організації «ковзного вікна» можна запропонувати використовувати на вході нейронної мережі лінію затримки. Послідовність значень вхідного сигналу надходить на лінію затримки, що складається з блоку запізнювання (по довжині «ковзного вікна»). Вихід лінії затримки - N-мірний вектор , складений з значень входу в моменти часу

Результат роботи лінійної нейронної мережі, що складається з одного лінійного нейрона і лінії затримки з десяти блоків запізнювання представлений на малюнку 1. Навчання даної нейронної мережі вироблялося за правилом Відроу-Хоффа, який передбачає мінімізацію середньоквадратичної помилки нейронної мережі.

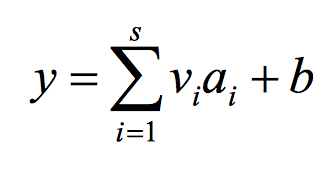
***Рис.2.*** *Результат передбачення нейронної мережі: - вхідні дані;*

*\* прогноз.*

Використання нелінійної функції активації дозволяє налаштувати нейронну мережу на реалізацію нелінійних зв'язків між входом і виходом. Однак якщо вихідний шар багатошарової мережі буде використовувати нелінійні сігмоідної функції активації, то виходи мережі будуть обмежені. Якщо ж у вихідному шарі використовуються лінійні нейрони, то виходи мережі можуть приймати довільні значення. Кількість нейронів у вихідному шарі визначається метою вироблених нейронною мережею обчислень. Таким чином, вихідний шар нейронної мережі для вирішення задачі передбачення динаміки економічних показників був прийнятий містить один лінійний нейрон. Оскільки функція активації нейронних елементів прихованого шару має область чутливості для входів в діапазоні трохи ширше інтервалу від -1 до +1, вхідні дані були піддані препроцессуванню приведенням до діапазону [-1;1] за формулою:

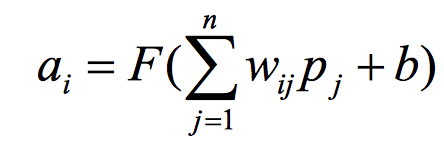
де - матриця нормованих векторів входу; - матриця векторів входу;

- мінімальний елемент входу; - максимальний елемент входу. Вихідна значення нейронної мережі:



де vi - i-й ваговий коефіцієнт вихідного нейрона; b - зсув вихідного нейрона; s - кількість нейронів в прихованому шарі; ai - вихід i-го нейрона прихованого шару.

Вихідні значення нейронних елементів прихованого шару:



де F - сігмоїдна функція активації прихованого шару ; wij - ваговий коефіцієнт від j-го входу до i-му нейрона; pj - j-й елемент входу; b - зсув нейрона прихованого шару.

# ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

# *Функціональні можливості та структура програми*

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 **С++** 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 інтерпритатор мови **R** і пакет **Shiny**. Але вже працюючий сервер надє змогу кліентам віконувати програму на інших машинах через сторінку браузера.

Користувач програми має такі можливості:

1. Завантажувати файл з данними часового ряду на сервер. Можна завантажувати файл будь-якого формату, але за замовченняв в діалозі вибору файлу встановлений фільтр для \*.csv і \*.txt файлів. Для уточнення структури файлу користувач повинен налаштувати параметри «Заголовок» (Header), що вказує має таблиця у файлі назви стовбців чи ні, «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 візлів у кожному прихованому шарі окремо і разом (одним числом для всіх);
* кількість гамма-юнітів;
* кількість останніх елементів, що може бачити перцептрон;
* параметри зупинки алгоритиу навчання;
* кількість елементів у навчальній ітерації;
* яким чином (випадковим чи ні) обираються елементи для навчання.

# *Організація обчислювального процесу*

Є такі незалежні від середовища виконування програмні блоки:

1. *Matrix* задає і реалізує інтерфейс і алгебраїчні операції для матриці дійсних чисел;
2. *Serialization* описує і визначає функції для сериалізації і десериализації (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 контролювали процес навчання завдяки функціям forward\_prop, put\_errors і flush, перша з яких відповідає за ForwardPropagation і запамятовує у буфері виходи кожного зі слоїв, друга за BackPropagation і наколює 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 можливасті при навчанні нейронних мереж зазвичай не надають.  
   При закінченні навчання перцептрону 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. Cериализацию 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 навчаеться у зв’язці з перцептроном: перцептрону передають данні на вхід, серед яких є значення згортки ряду від конкретногогамма-юніту; перцептрон повертає значення помилок вхідних даних; гамма-юніту змінює значення свого вагу або накопичує гражіент — залежить від кітькості патернів;
6. *GammaNN* обгортає Perceptron для використання з часовими рядами, надає і реалізує інтерфейс для навчання перцептрону 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 жа 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 сериалізувати таким самим чином: функції write\_to\_stream і from\_stream;
7. Тестування підпрограм. Для тестування Percepton використовувалися навчальні дані задачі XOR та інші. Для тестування GammaNN використовувалися навчальні дані: послідовність числа 1, послідовність нулів та одиниць, двомірна 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 початковій 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 обєектів 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 путь до файлу, де сериалізований GammaNN об’єкт; віддає NNptr.
7. *write\_to\_file* приймає NNptr, путь до файлу, куди сериалізувати GammaNN об’єкт.

# *Інструкція користувача*

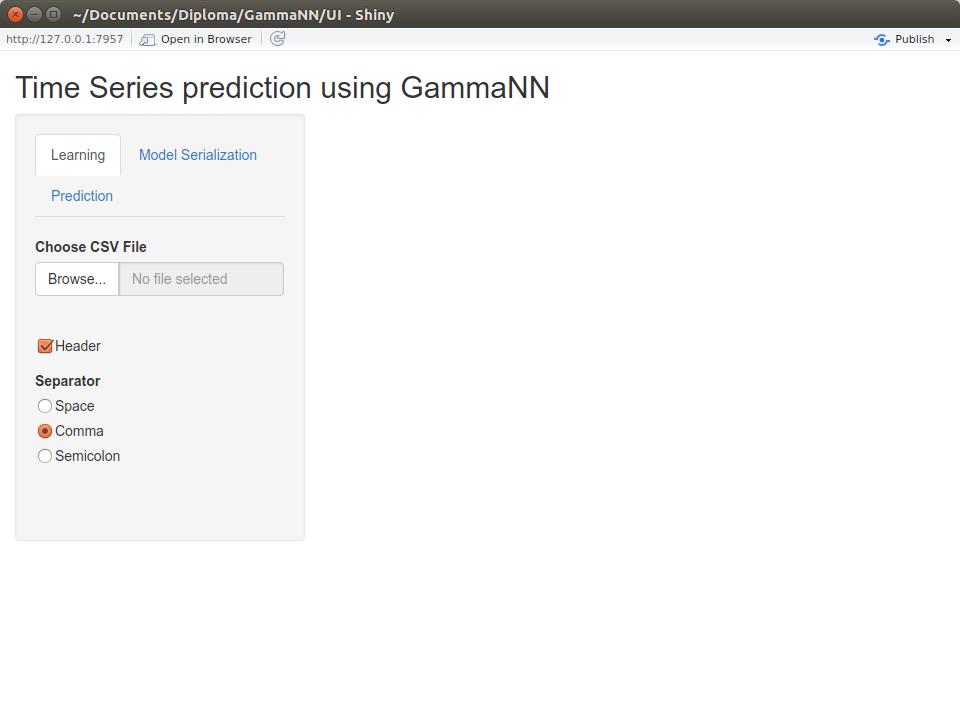
Розроблено програмний продукт, призначений для розв’язання задач прогнозування часових рядів. Алгоритм створений на мові С++, інтерфейс користувача створений на мові R з використанням програмного пакету Shiny.

*Опис інтерфейсу програми:*

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

Щоб почати роботу з програмою, необхідно запустити файл ***UI.Rproj*.**

Ви бачите інтерфейс:



***Рис.3.*** *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:



***Рис.4.*** *Обираємо файл з часовим рядом*

Після відкриття файлу стає доступним інтерфейс для навчання моделі.



***Рис.5.*** *Інтерфейс навчання моделі(1)*



***Рис.6.*** *Інтерфейс навчання моделі(2)*



***Рис.7.*** *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).



***Рис.8.*** *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 натиснути ***Learn***.



***Рис.9.*** *Приклад результат навчання моделі*

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



***Рис.10.*** *Вкладка Model Serialization*

У вкладці***Prediction*** з’явився результат роботи програми:



***Рис.11.*** *Вкладка Prediction*

***Range*** – проміжок на якому брати номери елементів з ряду;

***Mean relative error*** – середня відносна помилка;

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

Інтерфейс, що має підпис «Choose CSV File to set src data», для обрання файлу є аналогічним до того, що надає змогу виборати 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 фукціональна можливість вікористовується для моделей завантажених з файлу.

# РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНИХ ЕКСПЕРИМЕНТІВ

Були проведені обчислювальні експерименти з використанням таких даних:

Програма випробувана на даних первинної інвалідності в Україні; даних сили вітру; даних щоденної кількості проданого товару.

1. *Первинна інвалідність в Україні.* З них взято дані про цереброваскулярні хвороби Вінницької, Закарпатської, Рівненської та Тернопільської областей.
2. *DAYSALES* – масив даних щоденної кількості проданого товару.  
   Взято 196-ту та 285-ту торгову точку.
3. *Дані сили вітру* – масиви даних середнього значення сили вітру за день. Дані зібрані з декількох датчиків в одному районів. Взято дані окремо з датчику WP1; разом з датчиків WP1,WP4,WP6,WP7.

Будемо вважати, що прогноз на один крок вперед задовільний, якщо :

* RMSE ( *root mean squared error* – корінь середньоквадратичної помилки) ≤ 0,1

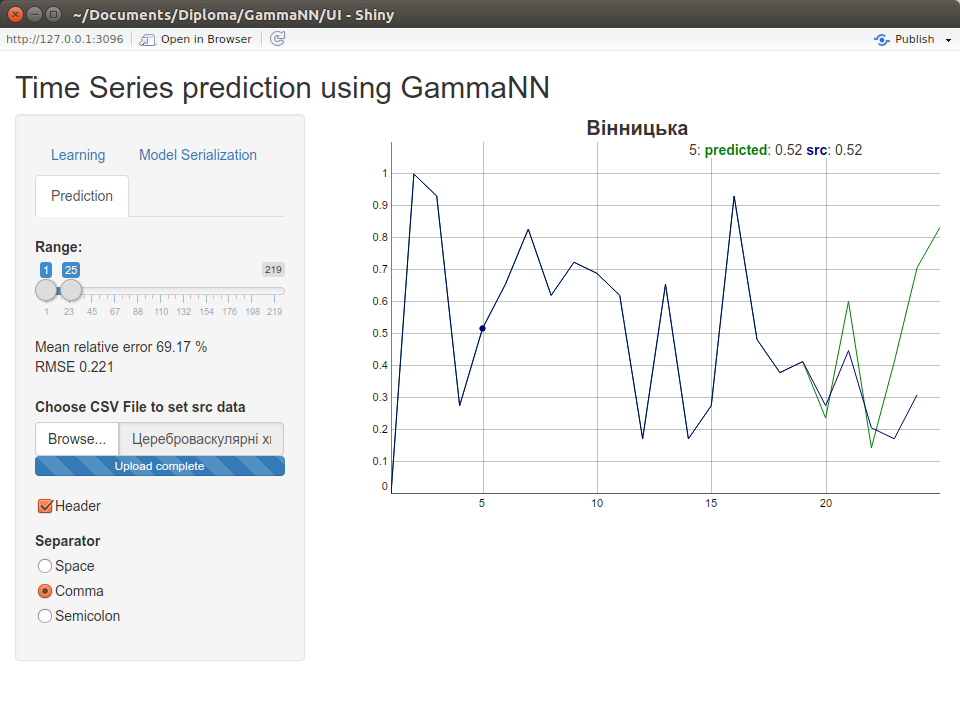
або

* MRE (*mean relative error* – середня відносна помилка) ≤ 10%.

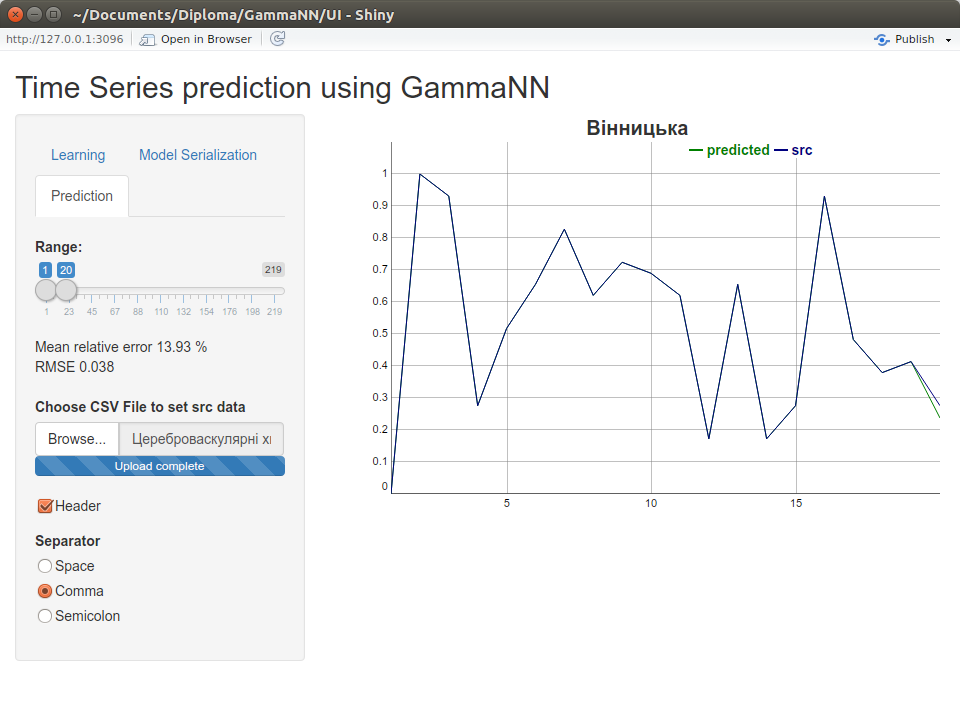
Будемо вважати, що загальний прогноз характеру ряду задовільний, якщо динаміка прогнозованого ряду схожа з динамікою початкового ряду.

*Прогнозування цереброваскулярних хвороб:*

Мають місце дані моніторингу первинної інвалідності в Україні за 24 роки (1992-2015).



***Рис.12.*** *Прогнозування “Цереброваскулярні хвороби Вінницької області”*

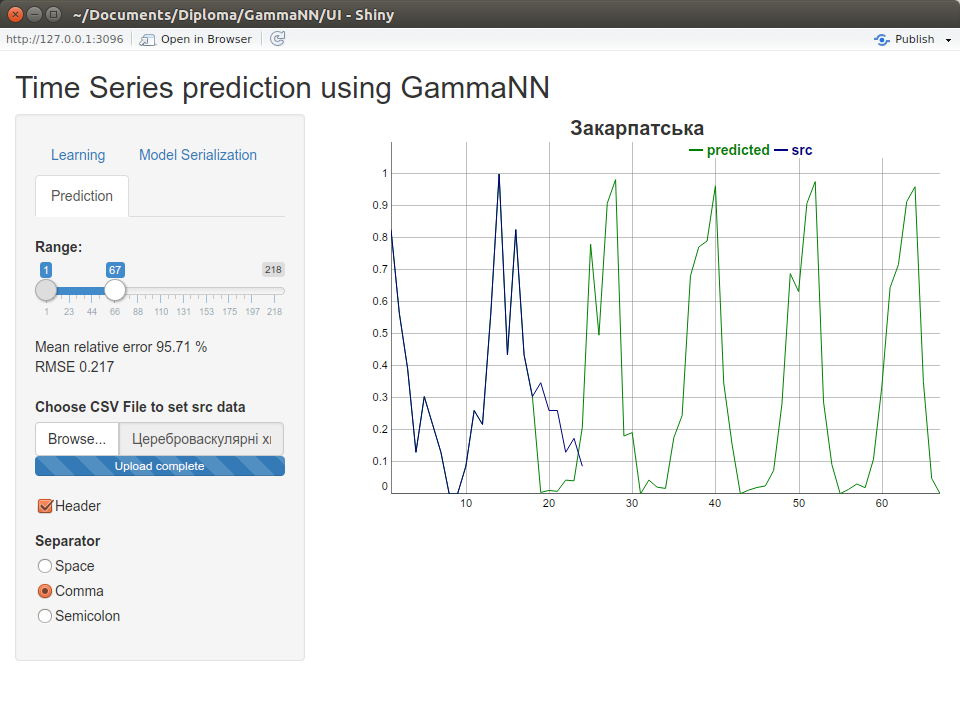


***Рис.13.*** *Прогнозування для першого елементу “Цереброваскулярні хвороби Вінницької області”*

Ця модель має 4 гамма-юніти.

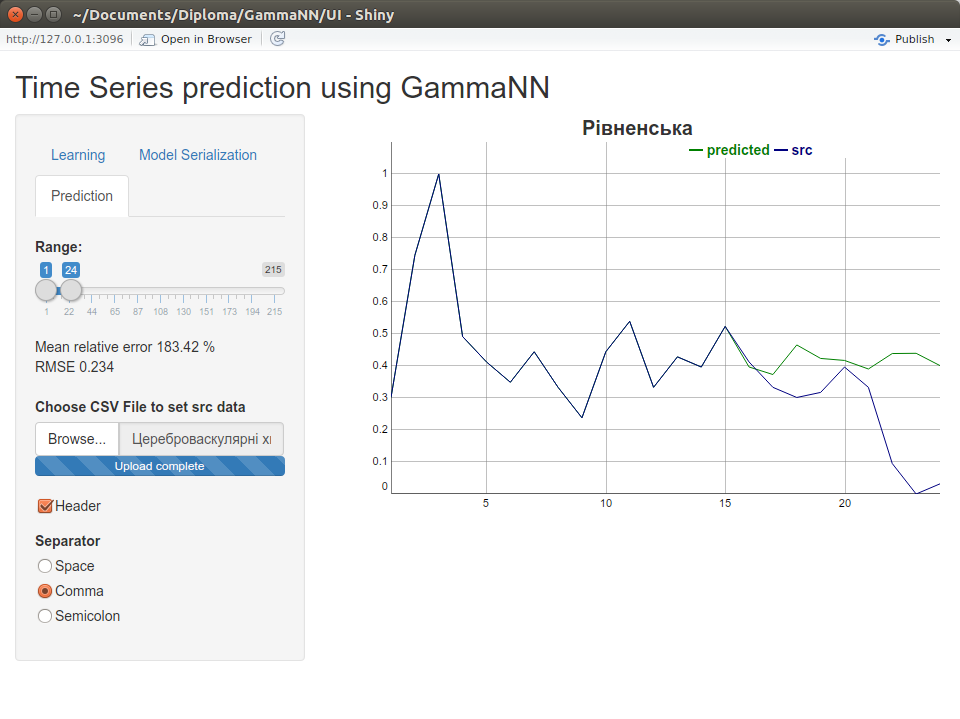
Як бачимо, для першого прогнозування RMSE = 0,038; RME=13,93% і динаміка прогнозованого ряду схожа з динамікою початкового ряду, отже загальний прогноз характеру ряду і прогноз на один крок вперед задовільні.

Надалі ми не будемо розглядати графік для прогнозування на один крок вперед, а просто будемо вказувати RMSE та RME для нього.



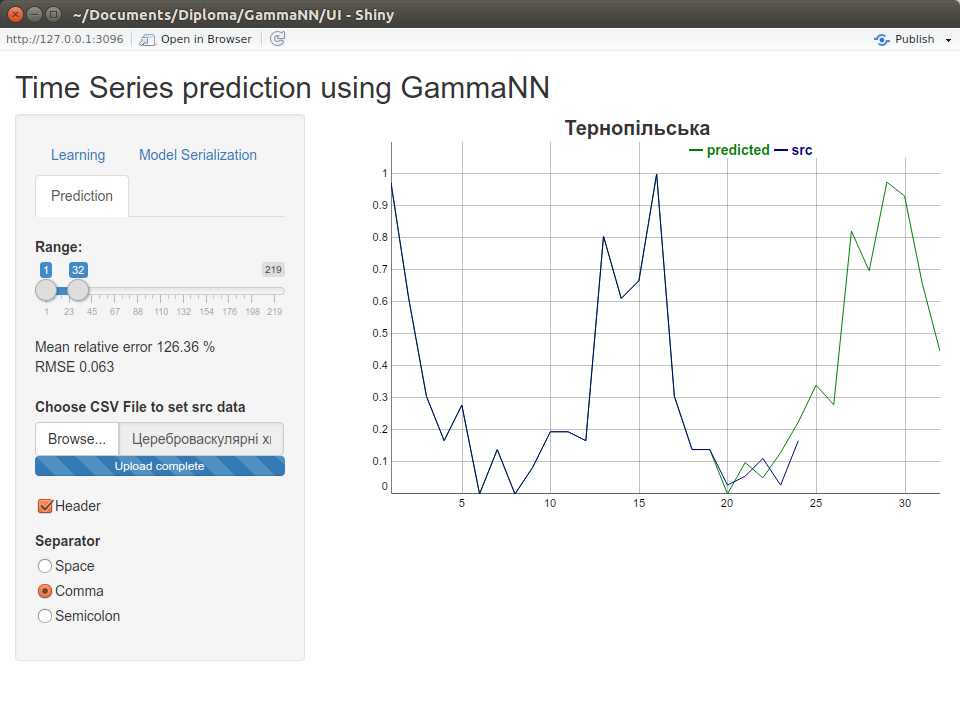
***Рис.14.*** *Прогнозування “Цереброваскулярні хвороби Закарпатської області”*

Для першого прогнозування: RMSE = 0,343; RME=98,56% - прогноз не є задовільним, загальний прогноз характеру ряду є задовільний. Тут можна побачити явний приклад перенавчання нейронної мережі. Це є наслідком того, що даних для навчання дуже мало, параметри виходу з циклу навчання є великими.



***Рис.15.*** *Прогнозування “Цереброваскулярні хвороби Рівненської області”*

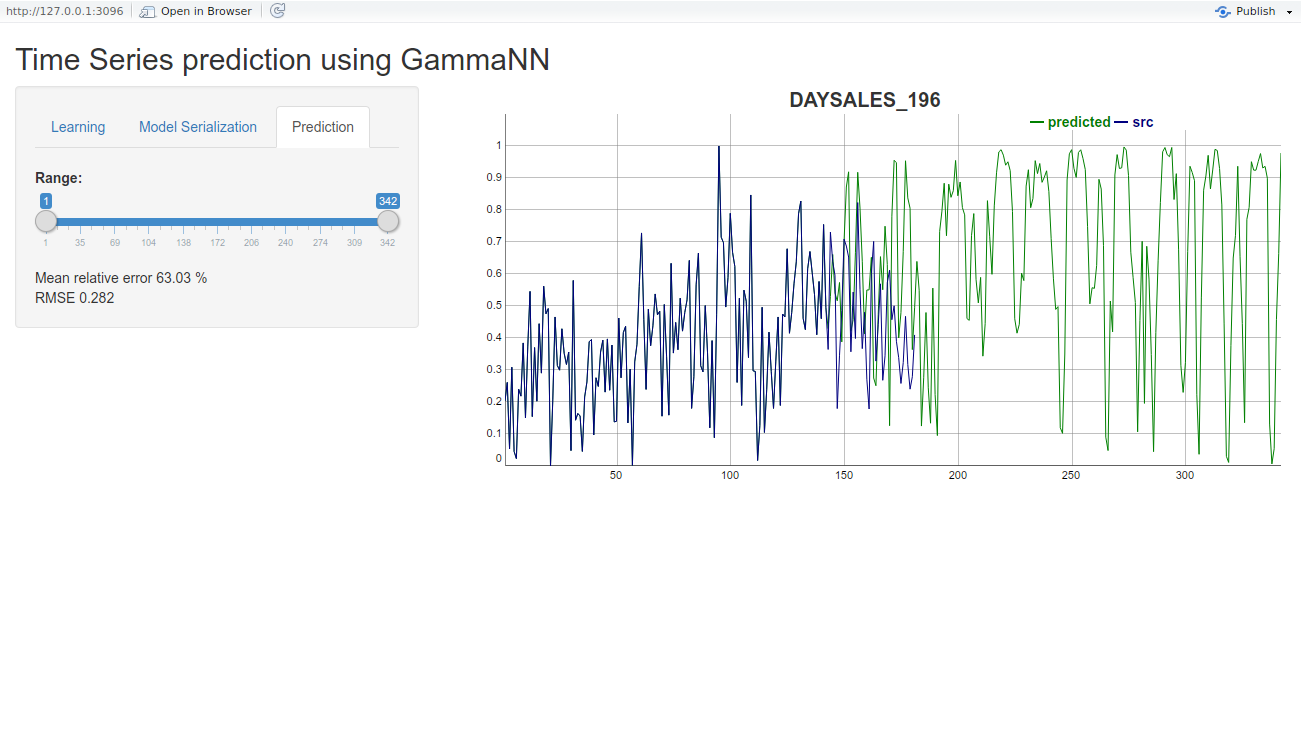
Для першого прогнозування: RMSE = 0,016; RME=3,81% - прогноз є задовільним, але загальний прогноз характеру ряду не є задовільний, бо модель швидко збігається до конкретного значення.



***Рис.16.*** *Прогнозування “Цереброваскулярні хвороби Тернопільської області”*

Для першого прогнозування: RMSE = 0,028; RME=99,34% - прогноз є задовільним, загальний прогноз характеру ряду є задовільний.

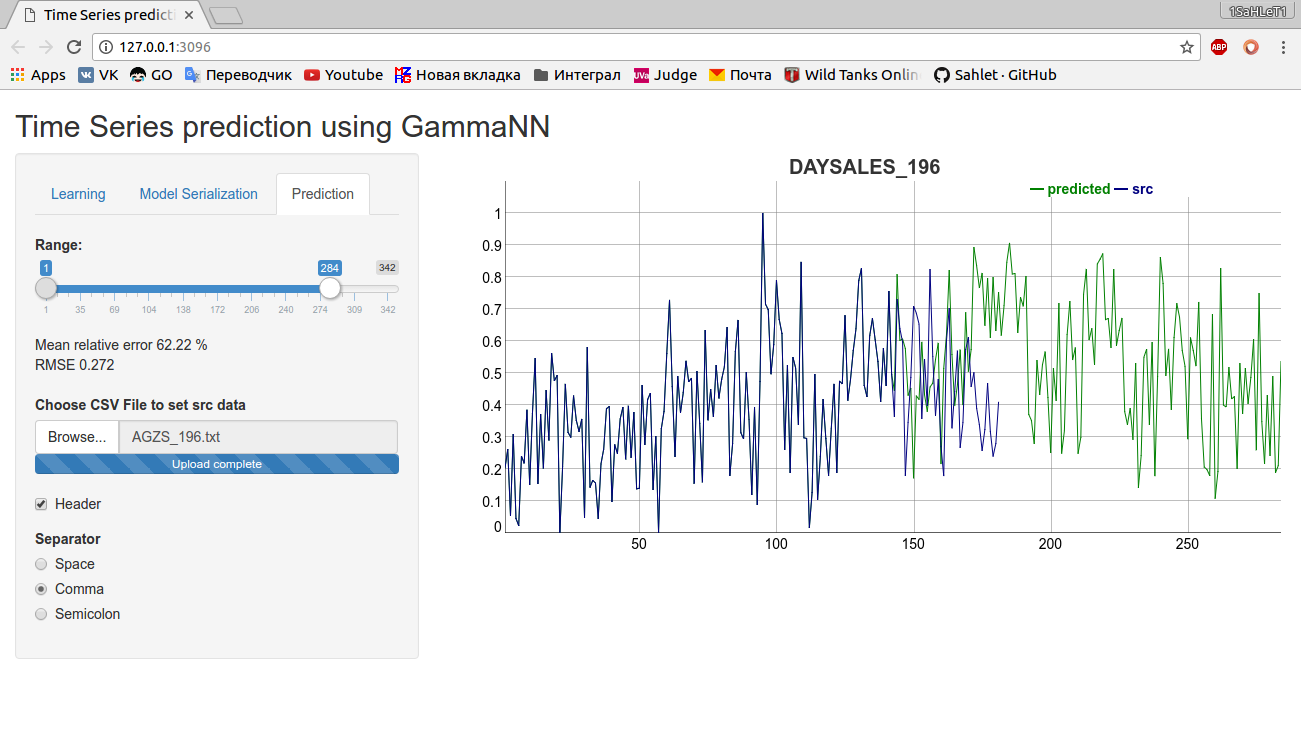
*Прогнозування щоденної кількості проданого товару.*



***Рис.17.*** *Прогнозування “DAYSALES\_196” без гамма-юнітів*

Ця модель не має гамма-юнітів, trace\_size = 22, hidden\_layer = {12}.

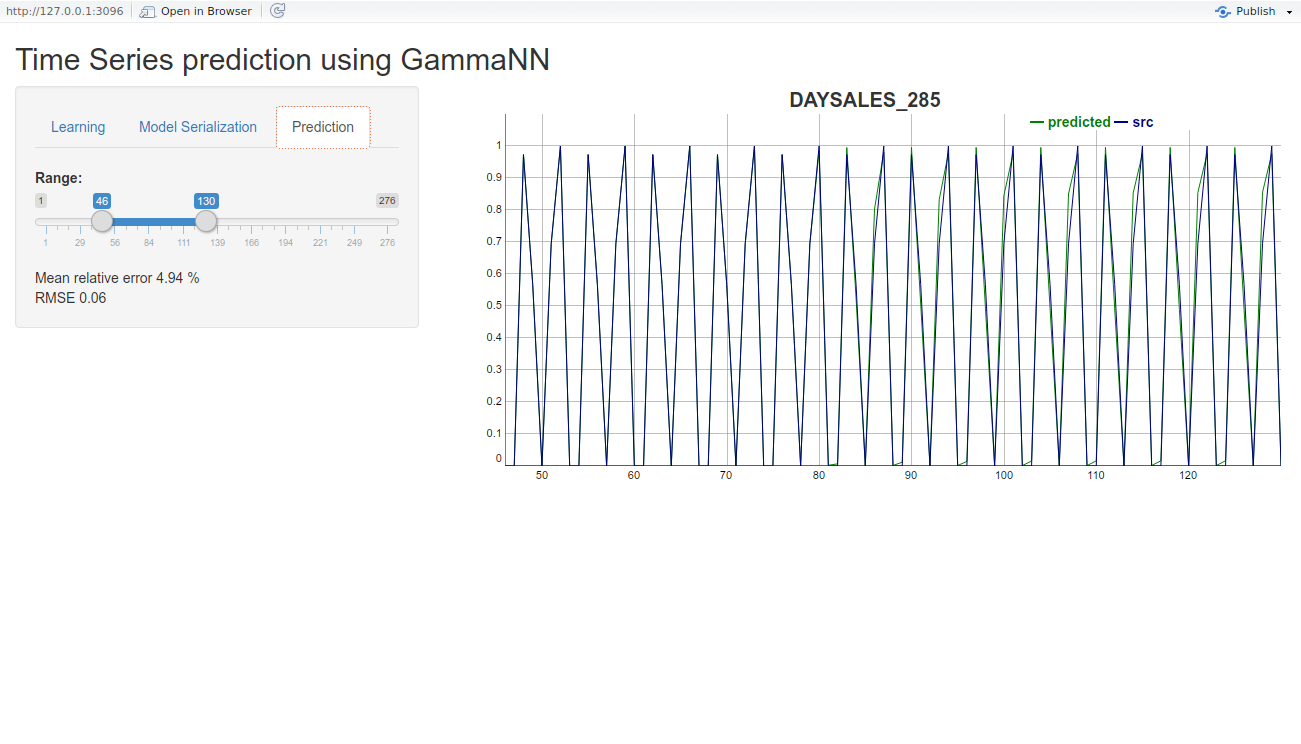
Для першого прогнозування: RMSE = 0,065; RME=17,93% - прогноз є задовільним, але загальний прогноз характеру ряду не є задовільний.



***Рис.18.*** *Прогнозування “DAYSALES\_196” з 4 гамма-юнітами*

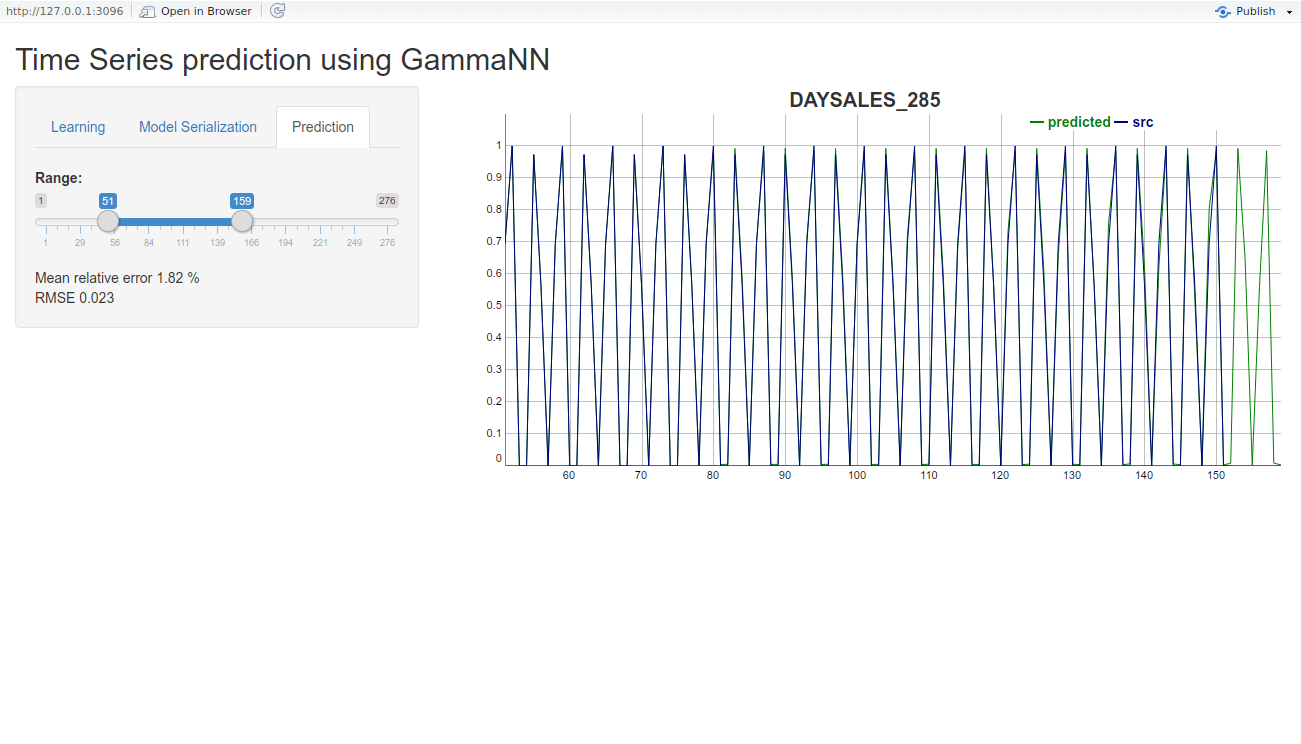
Ця модель має 4 гамма-юніти, trace\_size = 22, hidden\_layer = {12}.

Для першого прогнозування: RMSE = 0,081; RME=22,16% - прогноз є задовільним, загальний прогноз характеру ряду є задовільний.



***Рис.19.*** *Прогнозування “DAYSALES\_285” без гамма-юнітів*

Для першого прогнозування: RMSE = 0; RME = 0,01% - прогноз є задовільним, загальний прогноз характеру ряду є задовільний.



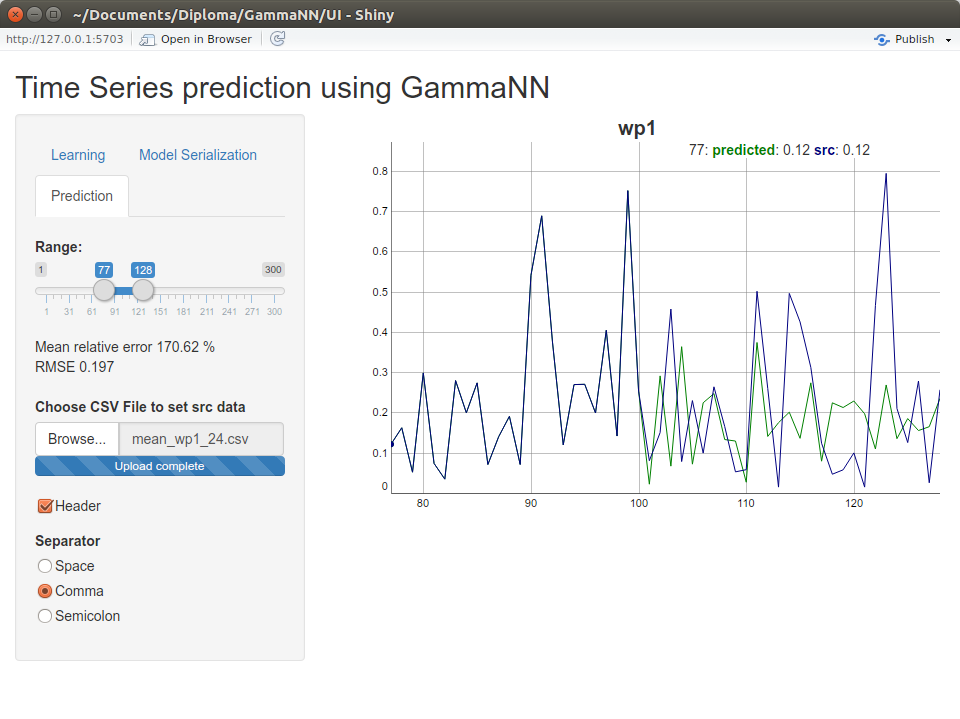
***Рис.20.*** *Прогнозування “DAYSALES\_196” з 1 гамма-юнітом*

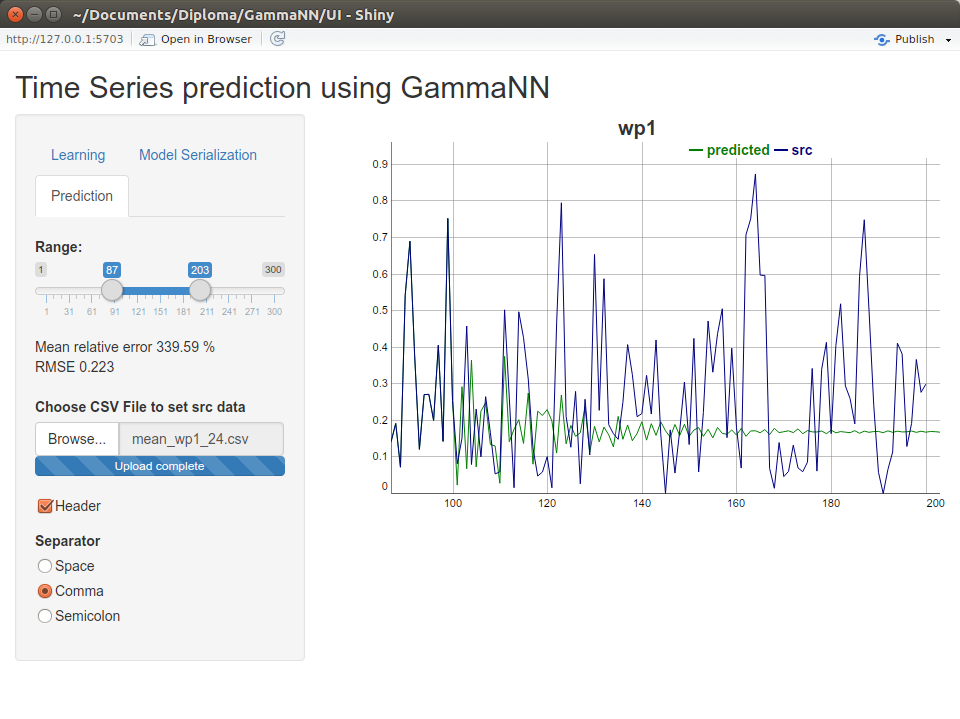
Для першого прогнозування: RMSE = 0,003; RME=0,49% - прогноз є задовільним, загальний прогноз характеру ряду є задовільний.

*Прогнозування сили вітру wp1.*

Всі експерименти проведені на даних силу вітру мали такі спільні параметри:

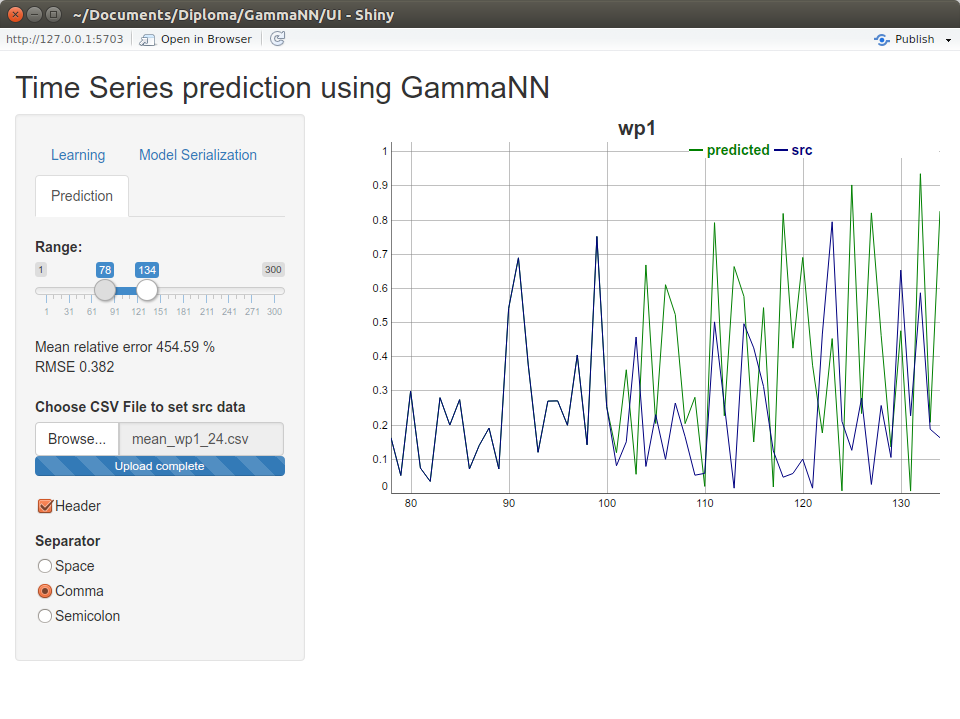
* розмір ряду для навчання 100;
* кількість елементів, що бачить перцептрон 10;
* кількість прихованих шарів перцептрона 1 і його розмір 60.

***Рис.21.*** *Прогнозування “wp1” без гамма-юнітів*

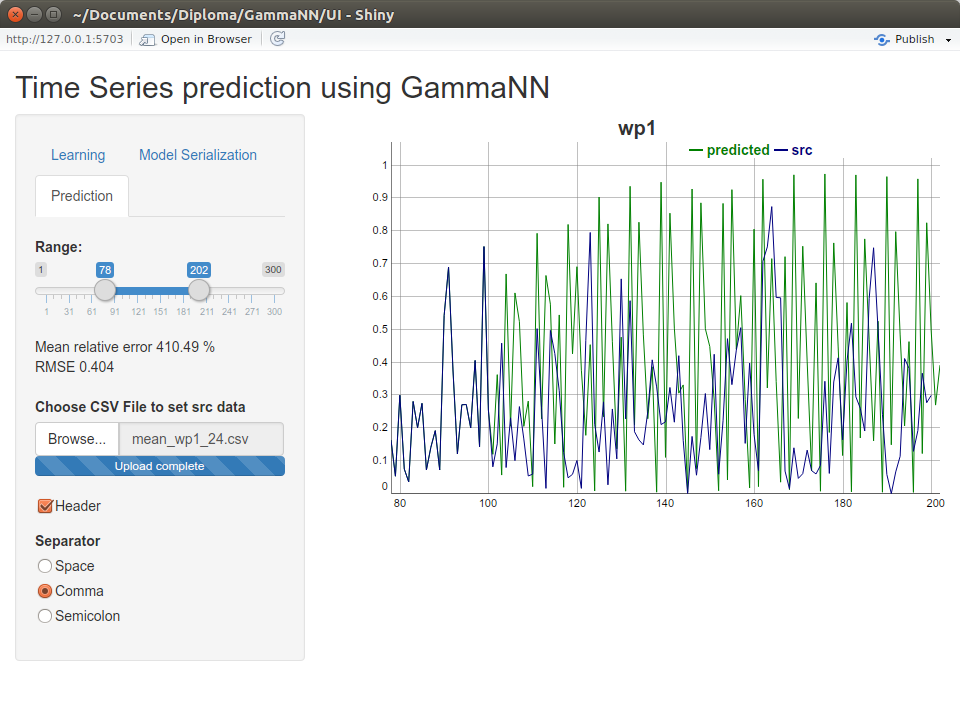
**

***Рис.22.*** *Прогнозування “wp1” без гамма-юнітів (глобальний прогноз)*

Ця модель не має гамма-юнітів. Для першого прогнозування: RMSE = 0,058; RME=70,52% - прогноз є задовільним, але загальний прогноз характеру ряду не є задовільний, бо модель швидко збігається до конкретного значення, це є наслідком перенавчання моделі.



***Рис.23.*** *Прогнозування “wp1” з 4 гамма-юнітами*



***Рис.24.*** *Прогнозування “wp1” з 4 гамма-юнітами (глобальний прогноз)*

Для першого прогнозування: RMSE = 0,038; RME=46,19% - прогноз є задовільним (***Рис.23***), загальний прогноз характеру ряду не є задовільний (***Рис.24***). Модель з гамма-юнітами має кращій вигляд характеру ряду, але він близький до циклічного, це також є наслідком перенавчання моделі.

Перенавчання моделей зумовлюється тим, що кількість прихованих вузлів завелика для такої кількості входів. Наслідком цього є велика кількість схожих значень на вході вихідного шару перцептрона, а після складання цих значень взагалі виходить число, яке майже не змінюється при зміні вхідних аргументів перцептрона.

*Прогнозування сили вітру wp1, wp4, wp6, wp7.*

Ціллю цього експерименту є прогнозування одного показника wp1, а (wp4, wp6, wp7) є допоміжними.

Причиною того, навіщо було взято з 7 можливих показників сили вітру (wp1, wp2, wp3, wp4, wp5, wp6, wp7) взято саме ці (wp1, wp4, wp6, wp7), є думка про те, що показники, які корелюють між собою, прогнозувати разом можна ефективніше ніж окремо кожен із них.

Було написано тестову функцію, яка виводить на екран коефіцієнти кореляції для кожної пари елементів з кортежу (wp1, wp2, wp3, wp4, wp5, wp6, wp7). Ось результат її роботи:

corrs

wp1 and wp2 0.5233627

wp1 and wp3 0.7325224

wp1 and wp4 0.8684162

wp1 and wp5 0.7119886

wp1 and wp6 0.8113615

wp1 and wp7 0.8216116

wp2 and wp3 0.5323716

wp2 and wp4 0.6730180

wp2 and wp5 0.3392181

wp2 and wp6 0.6508898

wp2 and wp7 0.6843198

wp3 and wp4 0.7811174

wp3 and wp5 0.5576312

wp3 and wp6 0.7968291

wp3 and wp7 0.8298956

wp4 and wp5 0.6536351

wp4 and wp6 0.9476037

wp4 and wp7 0.9484640

wp5 and wp6 0.6158766

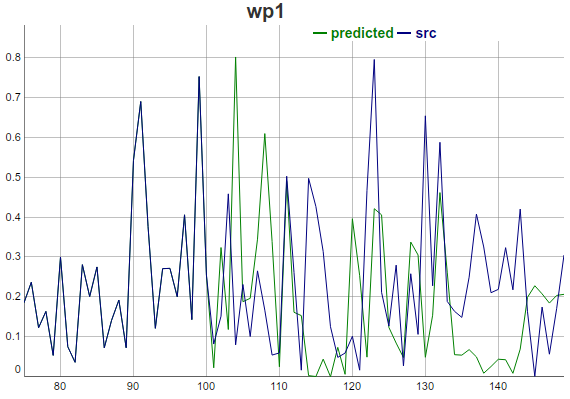
wp5 and wp7 0.6316573

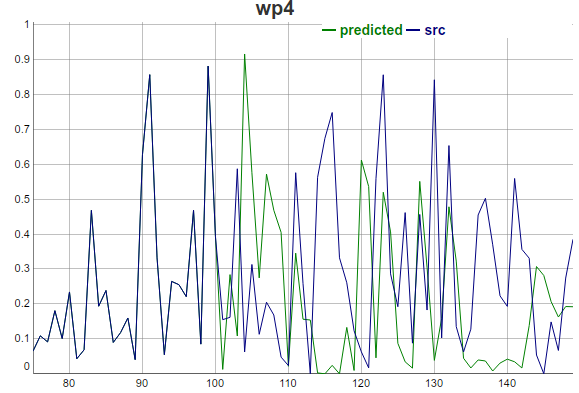
wp6 and wp7 0.9459604

***Таб.1.*** *Коефіцієнти кореляції між показниками силу вітру.*

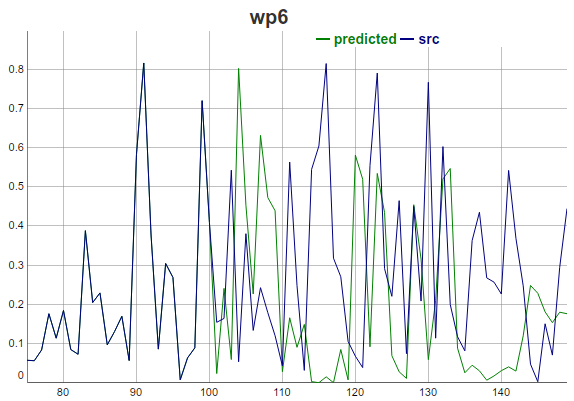
Для wp1 найвищі коефіцієнти кореляції з wp4, wp6, wp7, тому їх було вибрано як допоміжні показники.

Для першого обчислювального експерименту будуть приведені графіки всих прогнозованих часових рядів (wp1, wp4, wp6, wp7), а далі будуть приводитися графіки лише для wp1.

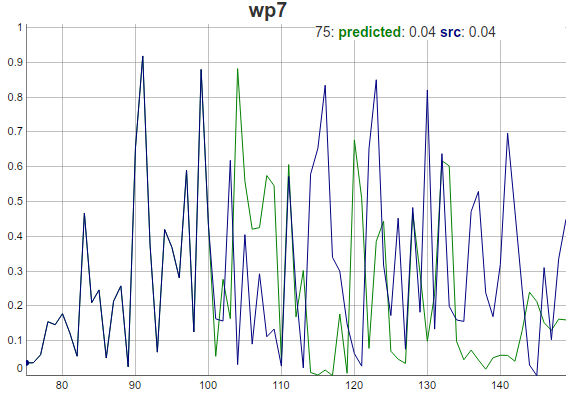
***Рис.******25.*** *Прогнозування “wp1, wp4, wp6, wp7” без гамма-юнітів, графік для wp1*



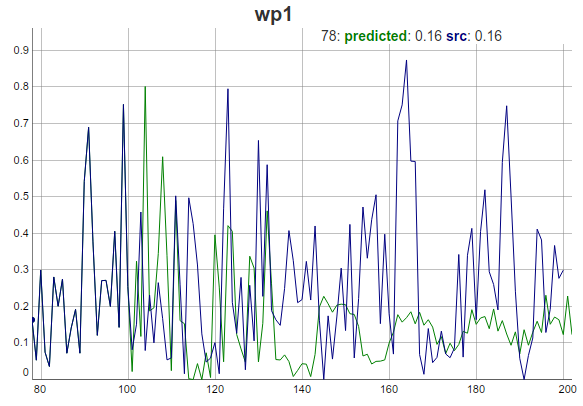
***Рис.26.*** *Прогнозування “wp1, wp4, wp6, wp7” без гамма-юнітів, графік для wp4*

******

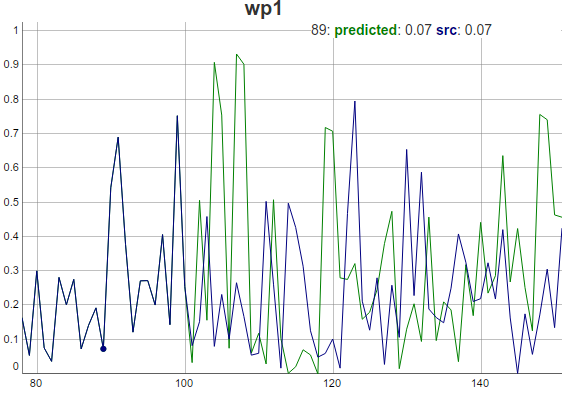
***Рис.27.*** *Прогнозування “wp1, wp4, wp6, wp7” без гамма-юнітів, графік для wp6*

******

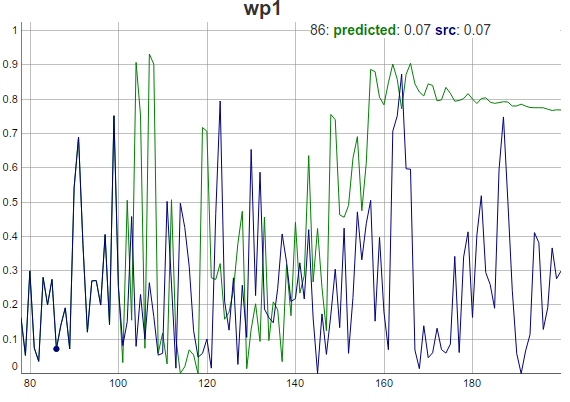
***Рис.28.*** *Прогнозування “wp1, wp4, wp6, wp7” без гамма-юнітів, графік для wp7*

***Рис.29.*** *Прогнозування “wp1, wp4, wp6, wp7” без гамма-юнітів, графік для wp1 (глобальний прогноз)*

Для першого прогнозування показника wp1: RMSE = 0.06; RME = 72,12% - прогноз є задовільними, але загальний прогноз характеру ряду не є задовільний.

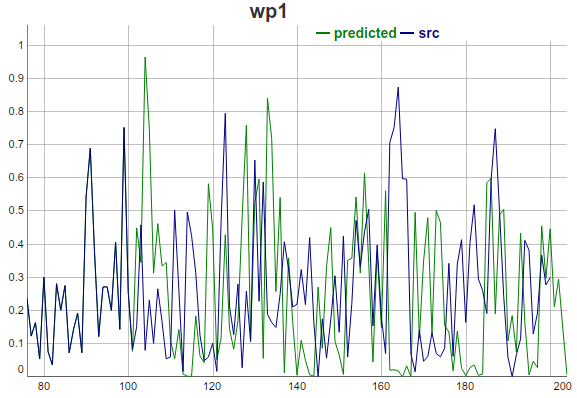


***Рис.30.*** *Прогнозування “wp1, wp4, wp6, wp7” з 4 гамма-юнітами, графік для wp1*

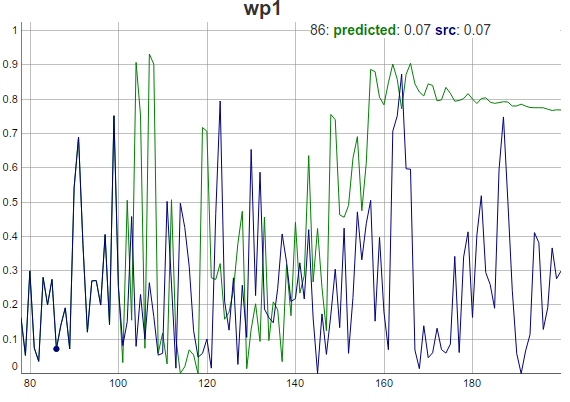
***Рис.31.*** *Прогнозування “wp1, wp4, wp6, wp7” з 4 гамма-юнітами, графік для wp1 (глобальний прогноз)*

Для першого прогнозування: RMSE = 0,05; RME=59,91% - прогноз є задовільними, загальний прогноз характеру ряду на перших 60 кроках є кращим ніж у випадку без гамма-юнітів, але потім модель вироджується — характер ряду не зберігається. Неспроможність загального прогнозу зумовлюється тим, що коефіцієнтами гамма-юнітів створювалися навмання і не змінювалися протягом навчання моделі. Це призвело до накопичення великого значення на одному з входів перцептрона, відповідного до виходу гамма-юніта, а значення на інших входах стали незначними.

Такого можна уникнути при навчанні моделі за гамма-юнітами, здатними до зміни параметрів для оптимізацію функції помилок (здатними до навчання).



***Рис.32.*** *Прогнозування “wp1, wp4, wp6, wp7” з 4 гамма-юнітами здатними до навчання, графік для wp1*

**

***Рис.33.*** *Прогнозування “wp1, wp4, wp6, wp7” з 4 гамма-юнітами здатними до навчання, графік для wp1 (глобальний прогноз)*

Для першого прогнозування: RMSE = 0,005; RME=6,4% - прогноз є задовільними, загальний прогноз характеру ряду є задовільним.

# ВИСНОВКИ

З результатів прогнозування рядів “Цереброваскулярні хвороби” можна зробити такі висновки:

* Через невелику кількість даних отримані прогнозі неможна розцінювати як вагоме ствердження;
* Здебільшого загальній характер ряду можна відновити і на невеликій кількості даних;

З результатів прогнозування рядів DAYSALES можна зробити такі висновки:

* Просту періодичну модель як “DAYSALES\_285” можна відтворити незалежно від того, чи є гамма-юніти чи немає;
* Різницю між помилками першого прогнозування “DAYSALES\_196” можна вважати вагомою – помилка моделі без гамма-юнітів менша ніж помилка моделі з гамма-юнітами, але загальний прогноз характеру ряду моделі з гамма-юнітами є кращий;

З результатів прогнозування ряду wp1 можна зробити такі висновки:

* У разі прогнозування складного ряду треба намагатися не ускладнити модель, у випадку з перцептроном — не обмежуватися малою кількістю вхідних параметрів і великою кількістю елементів на єдиному прихованому шарі; треба знайти оптимальну (близьку до оптимальної) архітектуру мережу — експериментувати з різною кількістю шарів перцептрона;
* Ускладнена модель перенавчається швидко. У випадку з гамма-юнітами і без гамма-юнітів вироджується у циклічно повторювані значення або число (число — окремий випадок циклу);

З результатів прогнозування рядів (wp1, wp2, wp3, wp4) можна зробити такі висновки:

* Показники, які корелюють між собою, прогнозувати разом можна ефективніше ніж окремо кожен із них;
* Чим більше показників для прогнозування — тим делатьніше треба налаштовувати модель;
* Використання гамма-юнітів здатних до навчання приводить до кращих результатів без зайвого ускладнення моделі (більше гамма-юнітів — складніша модель). Такі гамма-юніти треба застосовувати при остаточному навчанні моделі, тобто коли вибрана архітектура перцептрона і покращити модель вже нічим неможна. Це зумовлене тим, що процес навчання моделі з гамма-юнітами здатними до навчання повільніший від навчання звичайної моделі в декілька разів, бо зміна параметру одного гамма-юніта призводить до виконання операції порядку O(n\*d) де n — кількість елементів навчального ряду, а d — розмырнысть ряду.

Загальні висновки можна зробити такі:

* Гамма-юніти треба застосовували для відновлення моделей з великою кількістю початкових даних, бо для невеликих масивів даних гамма-юніти ускладнюють модель і призводять до перенавчання моделі;
* Нейронні мережі в поєднанні з гамма-юнитами і без них є хорошим апаратом для відтворення і прогнозування часових рядів;
* Допоміжні показники призводять до покращення моделі при правильному налаштуванні параметрів моделі. Допоміжними показниками можуть бути корелюючі вектори, перетворений початковій вектор;
* Треба переробити програму так, щоб вхідні параметри перцептрону перетворювалися сигмоїдною функції і мати змогу уника цього;
* Треба доповнити функціональність програми таким чином, щоб лінійність чи нелінійність вихідних вузлів перцептрону задавалсь динамічно;

# СПИСОК ЛІТЕРАТУРИ

1. А.Ю.Лоскутов, А.С.Міхайлов. Основи теорії складних систем. – М. Іжевськ: НДЦ – Регулярна і хаотична динаміка, 2007 – 620 c.
2. С.Хайкін. "Нейронні мережі повний курс", 2-е видання. Видавничий дім "Вільямс", 2006. – 1104 с.
3. Безручко Б.П., Смирнов Д.О. Математичне моделювання та хаотичні тимчасові ряди, Саратов: ГосУНЦ «Коледж», 2005 – 320 c.
4. Г.Г.Малінецький, А.Б.Потапов. Сучасні проблеми нелінійної дінаміки. – М. УРСС, 2000 – 360c.
5. А.Ю.Лоскутов, О.Л.Котляров, І.А.Істомін, Д.І.Журавлев. Проблеми нелінійної динаміки. Локальні методи прогнозування тимчасових рядів. – Москва, 2002– 36 с.
6. А.Ю.Лоскутов. Аналіз часових рядів. Курс лекцій., Фізичний факультет МГУ, 113 с.
7. Ніколаєва І.В. Застосування штучних нейронних мереж для прогнозування динаміки економічних показників. Стаття. – Краснодарський філіал РГТЕУ, 2012 – 32 с.