

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет**  
**імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

**ДОПУЩЕНО ДО ЗАХИСТУ**

В.о. завідувача кафедри інтелектуальних  
інформаційних систем, канд. техн. наук, доц.

\_\_\_\_\_ Є. В. Сіденко

« \_\_\_\_ » \_\_\_\_\_ 2023 р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

**ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПРОГНОЗУВАННЯ**  
**ПРОДАЖУ МЕДИКАМЕНТІВ**

Спеціальність 122 «Комп'ютерні науки»

**122 – МКР – 601. 21710808**

*Виконав студент 6-го курсу, групи 601*

\_\_\_\_\_ *О. І. Єфімов*

*(підпис, ініціали та прізвище)*

«15» лютого 2023 р.

*Керівник: канд. фіз-мат. наук, доцент*

\_\_\_\_\_ *І. В. Кулаковська*

*(підпис, ініціали та прізвище)*

«15» лютого 2023 р.

**Чорноморський національний університет ім. Петра Могили**

**Факультет комп'ютерних наук  
Кафедра інтелектуальних інформаційних систем**

Освітньо-кваліфікаційний рівень **магістр**

Галузь знань **12 «Інформаційні технології»**

*(шифр і назва)*

Спеціальність **122 «Комп'ютерні науки»**

*(шифр і назва)*

**ЗАТВЕРДЖУЮ**

В.о. завідувача кафедри інтелектуальних  
інформаційних систем, канд. техн. наук, доцент

\_\_\_\_\_ Є. В. Сіденко

« » \_\_\_\_\_ **2022 р.**

**ЗАВДАННЯ**

**на магістерську кваліфікаційну роботу**

**Єфімову Олексію Ігоровичу**

*(прізвище, ім'я, по батькові)*

1. Тема магістерської кваліфікаційної роботи «Інтелектуальна система прогнозування продажу медикаментів».

Керівник роботи Кулаковська Інесса Василівна, канд. фіз-мат. наук, доц.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «03» листопада 2022 р. № 199

2. Строк подання студентом роботи 15 лютого 2023 р.

3. Вхідні (початкові) дані до роботи: набір продажів медикаментів, методи побудови нейронних мереж.

Очікуваний результат: інтелектуальна система прогнозування продажу медикаментів.

4. Зміст пояснювальної записки (перелік питань, які потрібно розглянути):

- аналіз актуальності теми для розробки системи прогнозування;
- огляд та аналіз існуючих систем-аналогів;
- огляд технологій нейронних мереж та програмних засобів для створення обраної системи;
- тестування та аналіз отриманої системи.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: опис основних питань охорони праці пов'язаних з професійною діяльністю та використання комп'ютерів для роботи в офісі

---

---

7. Консультанти:

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Григор'єва Л.І. докт. біол. наук, професор	
Методична частина	Кулаковська І. В. канд. фіз-мат. наук, доц.	

Керівник роботи канд. фіз-мат. наук, доц. Кулаковська І. В.  
(наук. ступінь, вчене звання, прізвище та ініціали)

---

(підпис)

Завдання прийнято до виконання Єфімов О. І.  
(прізвище та ініціали)

---

(підпис)

Дата видачі завдання «07» листопада 2022 р.

## КАЛЕНДАРНИЙ ПЛАН

### Виконання магістерської кваліфікаційної роботи

Тема: Інтелектуальна система прогнозування продажу медикаментів

№	Найменування роботи	Початок	Закінчення	Примітки
1	Визначення керівника і теми МКР. Подання заяви на затвердження теми МКР	17.10.2022	17.10.2022	Виконано
2	Отримання завдання на виконання МКР	07.11.2022	07.11.2022	Виконано
3	Складання календарного плану на період виконання МКР	12.11.2022	12.11.2022	Виконано
4	Огляд літератури за темою дослідження	16.11.2022	22.11.2022	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до МКР	28.11.2022	18.12.2022	Виконано
6	Аналіз предметної області та розробка технічного завдання. Моделювання результатів	19.12.2022	20.12.2022	Виконано
7	Опис фахової частини МКР, зокрема аналіз предметної сфери, дослідження аналогів та публікацій, дослідження моделей, методів, технологій для реалізації задачі, програмна реалізація з аналізом отриманих результатів	19.12.2022	25.01.2023	Виконано
8	Розробка спеціальної частини з охорони праці та методичної частини	26.01.2023	02.02.2023	Виконано
9	Попередній захист МКР на засіданні комісії кафедри	03.02.2023	03.02.2023	Виконано
10	Корегування роботи за результатами попереднього захисту	04.02.2023	06.02.2023	Виконано
11	Остаточне оформлення пояснювальної записки та слайдів доповіді для захисту	07.02.2023	09.02.2023	Виконано
12	Подання МКР рецензенту	09.02.2023	09.02.2023	Виконано
13	Рецензування МКР	11.02.2023	11.02.2023	Виконано
14	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	15.02.2023	15.02.2023	Виконано
15	Захист МКР перед екзаменаційною комісією (ЕК)	22.02.2023	22.02.2023	Виконано

Розробив студент Єфімов О. І. \_\_\_\_\_  
(прізвище та ініціали) (підпис)

Керівник роботи канд. фіз-мат. наук, доц. Кулаковська І. В. \_\_\_\_\_  
(наук. ступінь, вчене звання, прізвище та ініціали) (підпис)

«12» листопада 2022 р.

## АНОТАЦІЯ

до магістерської кваліфікаційної роботи  
студента групи 601 ЧНУ ім. Петра Могили

**Єфімова Олексія Ігоровича**

на тему: **“ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПРОГНОЗУВАННЯ ПРОДАЖУ  
МЕДИКАМЕНТІВ”**

**Актуальність** даного дослідження полягає у пришвидшенні процесу прогнозування та підвищенні якості прогнозу. Недостатня кількість виробленої кількості українських медикаментів через воєнні дії та великий імпорт зарубіжних медикаментів додали складнощів та зробили різкі зміни для прогнозування продажу.

**Об’єктом** дослідження є процеси прогнозування продажів медикаментів.

**Предметом** дослідження є методи та засоби прогнозування продажів медикаментів.

**Метою** дослідження є прогнозування продажу медикаментів за рахунок розробки відповідної інтелектуальної системи із застосуванням нейронних мереж.

В результаті виконання роботи було досліджено два методи, аналіз часових рядів та випадковий ліс, проаналізовано можливості робити прогнозування та особливості їх процесу роботи у прогнозуванні.

Дана робота складається з п’яти розділів. Кожен розділ відповідно присвячений: аналізу предметної сфери та постановці задачі, математичним моделям, методам та інформаційним технологіям використаним у магістерській роботі, програмна реалізація та тестування, аналізу отриманих результатів, охороні праці, методичній частині магістерської роботи. Загальний обсяг роботи – 120 сторінок. Магістерська кваліфікаційна робота містить два додатки, 90 рисунків, 2 таблиці і посилання на 49 літературних джерел.

**Ключові слова:** інтелектуальна система, машинне навчання, нейронна мережа, прогнозування, Python.

## **ABSTRACT**

to the master's qualification work by the student of the group 601 of Petro Mohyla  
Black Sea National University

**Oleksii Yefimov**

### **“AN INTELLIGENT SYSTEM FOR FORECASTING THE SALE OF MEDICINE”**

The relevance of this study lies in speeding up the process of forecasting and improving the quality of the forecast. Insufficient production of Ukrainian medicines due to hostilities and large imports of foreign medicines added difficulties and made drastic changes to sales forecasting.

An object of research is the processes of forecasting sales of medicines.

A subject of the research is methods and means of forecasting sales of medicines.

A purpose of the study is to predict the sale of medicines through the development of an appropriate intelligent system using neural networks.

As a result of the work, two methods, time series analysis and random forest, were studied, the possibilities of forecasting and the peculiarities of their work process in forecasting were analyzed.

This work consists of five sections. Each of them is devoted to: analysis of the subject area and problem formulation, mathematical models, methods and information technologies used in the master's work, software implementation and testing, analysis of the obtained results, labor protection and life safety, methodological part of the master's work.

The overall scope of the work is 120 pages. Thesis contains 2 application, 90 figures, 2 tables and 49 sources in it.

**Key words:** intelligent system, machine learning, neural network, prediction, Python.

## ЗМІСТ

ВСТУП.....	10
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ТА ПОСТАНОВКА ЗАДАЧІ .....	11
1.1 Опис предметної сфери .....	11
1.2 Огляд та аналіз наявних публікацій .....	17
1.3 Огляд та аналіз наявних аналогів .....	20
1.4 Постановка задачі.....	24
Висновки до розділу 1 .....	25
2 ТЕХНОЛОГІЇ, МЕТОДОЛОГІЇ ТА АРХІТЕКТУРА ГЛИБОКО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ПРОДАЖІВ МЕДИКАМЕНТІВ .....	27
2.1 Технології створення нейромережових систем.....	27
2.2 Інструменти, методології для вирішення поставленої задачі .....	33
2.3 Архітектура глибокого навчання LSTM.....	40
Висновки до розділу 2 .....	48
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ .....	49
3.1 Встановлення програмного забезпечення та їх підключення. ....	49
3.2 Підключення набору даних та перевірка його роботи.....	49
3.3 Поділ даних.....	52
3.4 Багатоступінчатий прогноз .....	53
3.5 Оцінка моделі .....	53
3.6 Модель сталості.....	54
3.7 Підготовка даних.....	54
3.8 Fit LSTM Network.....	55
3.9 Прогноз LSTM.....	57

3.10 Інвертування перетворення.....	57
3.11 Перевірка результатів .....	59
3.12 Тестування інтелектуальної системи .....	61
Висновки до розділу 3 .....	65
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
ДОДАТОК А Критерії бальної оцінки умов праці .....	72
ДОДАТОК Б Залежність категорії умов праці від величини інтегральної бальної оцінки.....	74





## ВСТУП

Прогнозування продажів – це поширене і важливе застосування машинного навчання. Прогнози продажів можна використовувати для визначення контрольних показників і визначення додаткових впливів нових ініціатив, планування ресурсів у відповідь на очікуваний попит і прогнозування майбутніх бюджетів.

Нейронні мережі – це розділ штучного інтелекту, в якому обробки сигналів використовуються явища, аналогічні тим, що відбувається в нейрони живих істот.

Найважливіша особливість мережі, що свідчить про її широкі можливостях та величезному потенціалі, полягає у паралельній обробці інформації всіма ланками, що дозволяє значно прискорити процес обробки інформації. Крім того, при великій кількості міжнейронних з'єднань мережа набуває стійкості до помилок, що виникають на деяких лініях.

Актуальність даного дослідження полягає у пришвидшенні процесу прогнозування та підвищенні якості прогнозу. Недостатня кількість виробленої кількості українських медикаментів через воєнні дії та великий імпорт зарубіжних медикаментів додали складнощів та зробили різкі зміни для прогнозування продажу.

Об'єктом дослідження є процеси прогнозування продажів медикаментів.

Предметом дослідження є методи та засоби прогнозування продажів медикаментів.

Метою дослідження є прогнозування продажу медикаментів за рахунок розробки відповідної інтелектуальної системи із застосуванням нейронних мереж.

В результаті виконання роботи було досліджено два методи, аналіз часових рядів та випадковий ліс, проаналізовано можливості робити прогнозування та особливості їх процесу роботи у прогнозуванні.

В даній роботі всі дослідження і експерименти проводилися з набором реальних даних для досягнення максимальної ефективності в задачі прогнозування продажів товарів.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Опис предметної сфери

Підсумки 2022 року [1] загального обсягу продажу медикаментів всіх категорій показали 61,7 млрд. грн., що демонструє зменшення на 4,6% у порівнянні з минулим роком. У еквіваленті долара відбулося зменшення на 7,7%. Якщо ж говорити про натуральне вираження, то обсяги продажу зменшились на 22,4% і становлять 666,1 млн. штук (рис. 1.1).

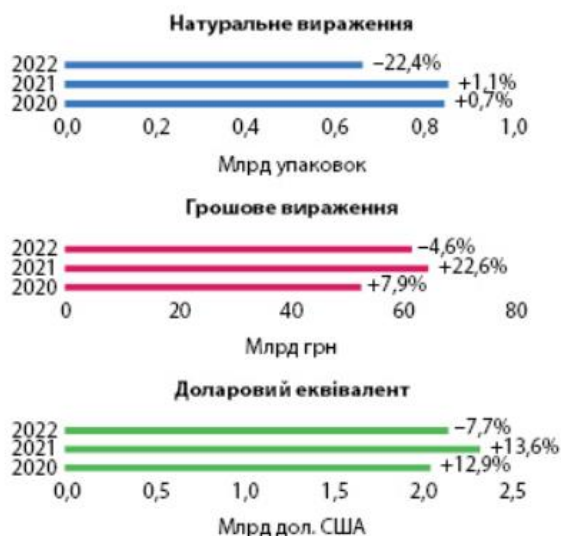


Рисунок 1.1 – Обсяги аптечних товарів в натуральному і грошовому (за курсом НБУ) вираженні

Серед аптечного продажу структура поділена на виробництво зарубіжного та українського видів. Серед натурального вираження частка українських препаратів переважає зарубіжного. Але при грошовому вираженні ситуація змінюється в сторону зарубіжних виробників (рис. 1.2).

Кафедра інтелектуальних інформаційних систем  
Інтелектуальна система прогнозування продажу медикаментів

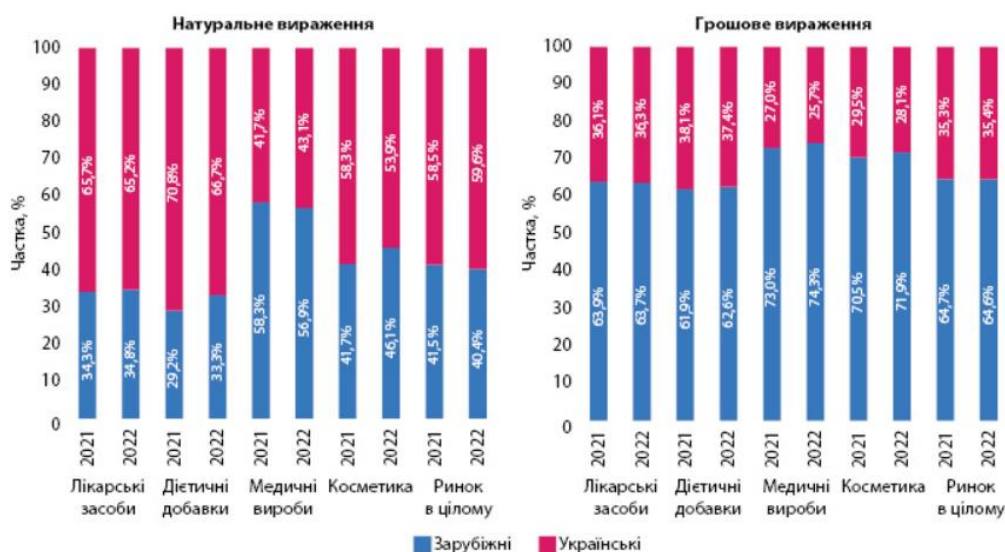


Рисунок 1.2 – Структура українського та зарубіжного продажу медикаментів у натуральному та грошовому еквівалентах

До початку війни продаж медикаментів [2] показував темпи зростання у грошовому вираженні (у січні – 31%, у лютому 45%). Але починаючи з березня продаж показав спадання у порівнянні з минулорічними показниками (рис. 1.3).

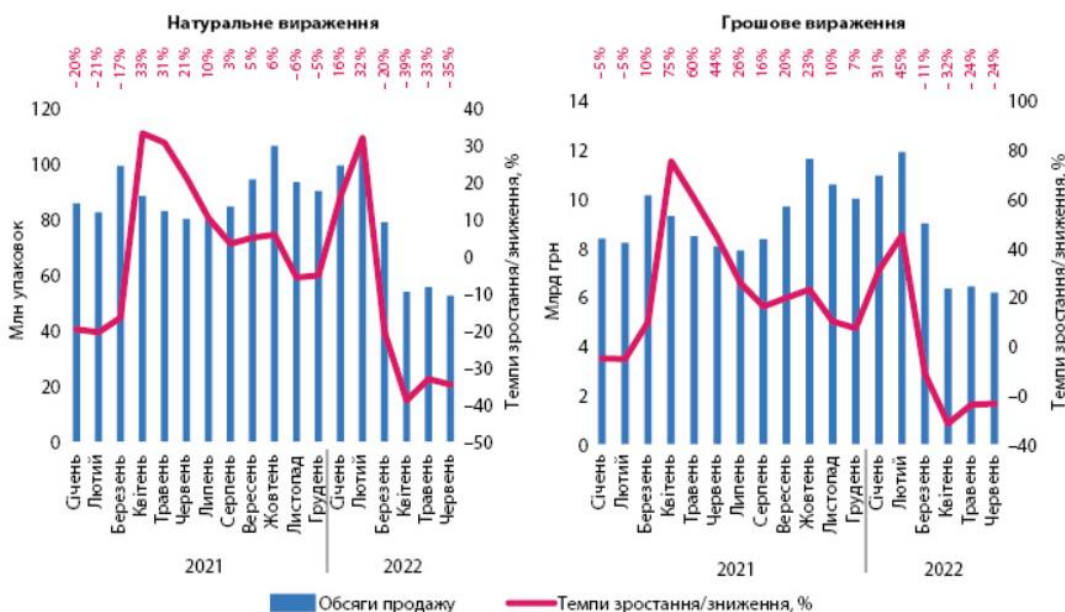
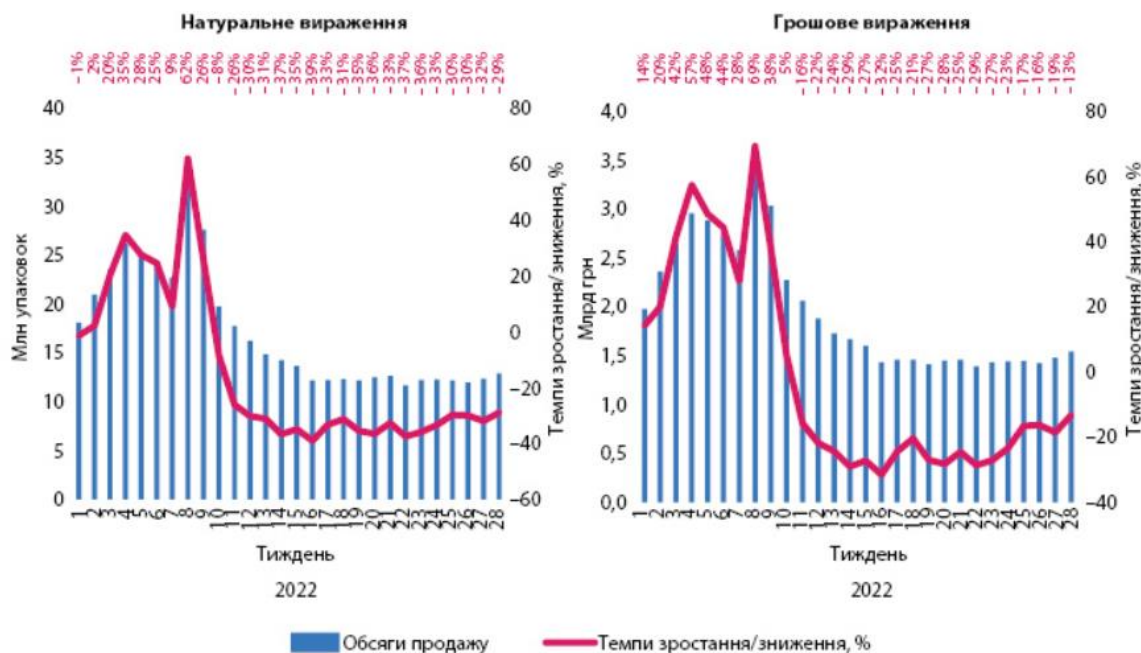


Рисунок 1.3 – Місячна динаміка продажу медикаментів у натуральному та грошовому вираженні

Якщо ж подивитись на тижневу динаміку, то можна побачити, що останні тижні обсяг продажів стабілізується, а темпи, в свою чергу, поступово сповільнюються (рис. 1.4).



Рисуюнок 1.4 – Тижнева динаміка продажу медикаментів у натуральному та грошовому вираженні

Основною проблемою, яка впливає на кількість продажу є війна, а саме:

- міграція населення та втрата доходів;
- руйнування, проблеми дистрибуції та нестача персоналу;
- бойові дії та тимчасова окупація.

З початку війни значних змін зазнала структура споживання медичних засобів у груп 2-го рівня АТС-класифікації (рис. 1.5). У порівнянні з довоєнним рівнем зросло споживання медичних засобів, що застосовуються при хронічних захворюваннях, а саме серцево-судинних та при цукровому діабеті. Це торкнулось також і споживання протизастудних препаратів, їх кількість також знизилась, це пов'язано з міграцією населення. Суттєво зменшилась кількість споживання антибіотиків.

Кафедра інтелектуальних інформаційних систем  
Інтелектуальна система прогнозування продажу медикаментів

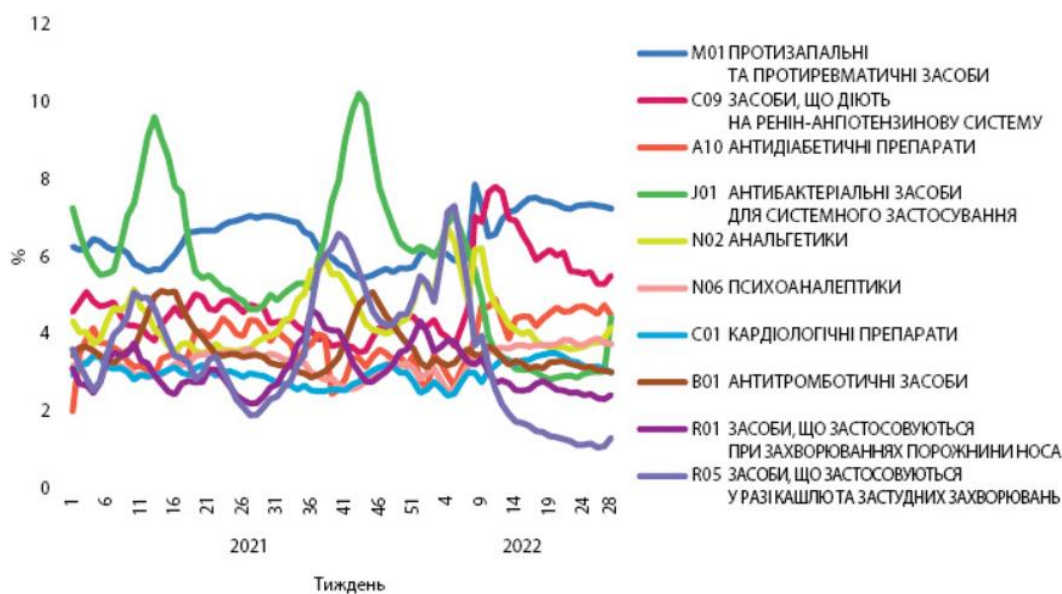


Рисунок 1.5 – Структура найпотрібніших 10 груп продажу медикаментів у грошовому вираженні

Споживання лікарських засобів за регіонами України показує найбільш проблемними є регіони, де проходять бойові дії та тимчасово окупована частина територій. У даних регіонах скоротилась частка використання медичних засобів. Через міграцію населення переважно в західні регіони, відбувся певне зміщення споживання в бік західних регіонів.

Сьогодні прогнозування продажів [3] дуже часто використовують компанії для вирішення задач, зокрема постановка цілей, отримана за річним прогнозом цифра – те, до чого компанія має прийти наступного року, той план, який потрібно виконати. Це частина бізнес-плану для підприємства і реальна, чітко прорахована мета відділу продажів, від якої можна відштовхуватися при нарахуванні премій та бонусів.

Дуже часто ціль ставиться з бажань, а не з реальних можливостей. Тому перед тим, як поставити ціль, необхідно спочатку зробити прогноз, а потім встановлювати ціль. Якщо мета вища за прогноз [4], то треба розуміти, за рахунок яких змін ціль буде виконана.

Формування необхідної бази трудових та виробничих ресурсів, виходячи з прогнозованої кількості клієнтів та обсягу продажів. Завдання: запланувати закупівлі та визначити майбутні потреби компанії в обладнанні та персоналі.

Управління складськими запасами. У кожний момент часу у розпорядженні виробництва буде складський залишок, достатній для виконання завдань на певному етапі. Жодного дефіциту чи надлишку матеріалів на складі – лише раціональне витрачання коштів!

Підвищення мобільності бізнесу. На прогнозному графіку (чи таблиці) можна заздалегідь побачити моменти можливого просідання обсягу продажу (наприклад, через сезонність продукту) і вжити заходів для коригування ситуації до закінчення періоду. Крім того, підвищуються шанси миттєво відстежити незапланований спад продажів, оперативно виявити причини зниження показників та вчасно виправити ситуацію.

Контроль та оптимізація витрат. Прогнозування [5] покаже, які витрати загалом понесе компанія на виробництво та реалізацію продукції. Отже, можна розробити бюджет і заздалегідь визначити, які витрати підлягають скороченню у разі невиконання прогнозу збільшення обсягу продажів.

Аналіз часових рядів [6] — це загальна сфера, яка має на меті робити прогнози на основі даних часових рядів, використовуючи серію часових індексованих точок. Традиційно в задачах прогнозування продажів використовуються прості моделі лінійної регресії [7] зі сфери статистики, а останнім часом – моделі випадкового лісу [8], розроблені в області машинного навчання. Методи штучного інтелекту є більш точними в певних ситуаціях, зокрема, коли функція не має лінійності.

Як правило, під час візуалізації часового ряду часто легко побачити закономірність, яка продовжує повторюватися з невеликими варіаціями. Часто набагато складніше знайти модель, яка точно відповідає такому шаблону; виклик ще більше ускладнюється, враховуючи постійно мінливі обставини, які лежать в

основі майбутніх даних. LTSM [9] довів свою ефективність у цій галузі, надаючи ближче до загального рішення проблеми прогнозування.

Імовірнісні та статистичні методи [10] намагаються використовувати розподіли ймовірностей, такі як нормальний розподіл, або математичні функції, такі як лінійні чи параболічні рівняння, щоб відповідати набору точок даних. Як правило, ці підходи можуть добре працювати, якщо дані тісно узгоджуються з змодельованою функцією. Однак ці методи вимагають експертного аналізу та є досить ручним процесом, і їх важко реалізувати на практиці в широкому масштабі, враховуючи необхідний ступінь спеціалізації. Також були запропоновані методи машинного навчання, такі як Випадковий ліс. Ряд дерев рішень будується з використанням псевдовипадкових послідовностей;

Доведено, що Random Forest [11] добре працює та уникає переобладнання, хоча цей підхід не забезпечує ефективного масштабування при прогнозуванні, оскільки набори даних стають великими та складними. Таким чином, важко реалізувати в практичних умовах для всіх, крім підмножини проблем, з дуже обмеженими наборами даних.

LTSM [12] долає обмеження попередніх підходів шляхом навчання варіанту нейронної мережі, призначеної для послідовного навчання для кожного тимчасового кроку та безпосереднього моделювання даних. Це досягається за допомогою серії воріт: входу, виходу та воріт забуття. Значення запам'ятовуються на кожному кроці часу, а шлюз регулює потік інформації між станами. По суті, мережа навчається функціонуванню даних, що дозволяє ШІ фіксувати складні взаємозв'язки. Розглянемо приклад нижче (рис. 1.6), зелена лінія представляє фактичні дані, а червона лінія представляє прогнозовані дані через LTSM, можна побачити, що прогноз дуже близький до передбачення фактичних значень.





Рисунок 1.6 – Приклад прогнозу за допомогою LSTM

Таким чином, організації можуть прогнозувати та прогнозувати набагато більше сценаріїв, ніж у них було б ресурсів у попередніх налаштуваннях.

## 1.2 Огляд та аналіз наявних публікацій

Більшість фармацевтичних компаній використовували методи прогнозування на основі машинного навчання, щоб заздалегідь визначити майбутні продажі [13]. Головною перевагою використання штучних нейронних мереж або методів інтелектуального аналізу даних є їх здатність давати високоякісні прогнози, споживаючи менше часу. Практично неможливо отримати таку вищу точність за допомогою існуючих стандартних статистичних методів прогнозування [14]. У наступній публікації [15], автори зазначили, що компанії не змогли вчасно доставити постачальникам необхідну кількість продукції. Головна причина полягає в тому, що ці компанії не змогли визначити швидкі зміни в кількості попиту та не знали про попит інших конкурентів в управлінні ланцюгом поставок. Таким чином, автори запропонували цим організаціям використовувати інструмент прогнозування для управління прогнозуванням продажів та управління відповідними функціями.

Дуже популярними та успішними лінійними моделями часових рядів для прогнозування є авторегресійне інтегроване ковзне середнє (ARIMA) [16], сезонне

авторегресійне інтегроване ковзне середнє (SARIMA) [17] та авторегресійне ковзне середнє (ARIMA), але якість прогнозу постраждала через припущення прямого дії [18]. Це призводить до швидкого розвитку методів машинного навчання, таких як методи інтелектуального аналізу даних і штучні нейронні мережі (ШНМ) для прогнозування даних часових рядів, що забезпечує кращу точність прогнозування [19]. Останнім часом зростання моделі прогнозування часових рядів на основі штучних нейронних мереж було величезним. Крім того, він прокладає шлях для створення точних прогнозів шляхом обробки нелінійних вхідних і вихідних даних. Методи лінійного прогнозування, згадані вище, не підходять для моделювання даних, що містять нелінійну поведінку [20]. Останнім часом ШНМ широко використовується в прогнозуванні часових рядів для кращої якості прогнозу [21]. Тому в численних дослідженнях у літературі пропонується вивчити ефективність прогнозування часових рядів на основі ШНМ у порівнянні з використанням традиційних методів прогнозування. З іншого боку, ці дослідження показали, що для домену часових рядів не було контрольного налаштування параметрів нейронної мережі. Крім того, це повністю залежить від предметної області.

Загалом, прогнозування попиту є найбільшою проблемою для програми великих даних, яка може використовувати додаткову інформацію про дані, таку як цінність і достовірність, для кращого прогнозування. Це дослідження в основному було зосереджено на прогнозуванні продажів на аналітику великих даних [22]. Це пояснює включення різних джерел даних у можливий процес прогнозування, але для виконання завдання потрібен аналітик даних, відповідна технічна база та кошти в технології [23].

Попередня література підкреслює недостатність статистичних методів для оптимізації фармацевтичної діяльності. Зважаючи на це обмеження, деякі дослідники [24] прийняли імпровізований підхід, який дозволив поєднати методи моделювання та аналіз даних. Він рекомендував оптимізувати фармацевтичний підрозділ для оптимізації продуктивності [25].

В іншому дослідженні [26], метод згорткової нейронної мережі (CNN), генетичний алгоритм на основі NN і методи оптимізації рою частинок (PSO) на основі NN виконуються для прогнозування попиту на споживання енергії. Згідно з результатами моделювання, методи прогнозування енергії на основі нейронної мережі передбачили продажі з більшою точністю, ніж підхід CNN.

Автори [27] розробили генетичні алгоритми (GA), байєсівську структуру та структуру посилення штучних нейронних мереж (RANN) для прогнозування короткострокових продажів. Експериментальне дослідження цих моделей підкреслило перевагу моделі ШНМ, оскільки результати, отримані нею, показали вищу якість прогнозування порівняно з традиційними алгоритмами.

Багато дослідників намагалися створити майже ідеальні моделі прогнозування з нульовою помилкою. Цікаво, що в одному з таких досліджень було використано нейронечіткий підхід [28] для прогнозування майбутніх продажів на основі минулих даних про продажі. ANFIS був реалізований разом із ШНМ як «нейро-нечіткий підхід», незважаючи на специфікацію, він показує ефективність підходу. Є примітне дослідження [29], де розглядалися сучасні підходи та важливі питання прогнозування продажів ліків фармацевтичних компаній. Вони використали три сценарії прогнозування попиту. Було зроблено висновок, що «модель прогнозування на основі символічної регресії» має кращу відповідність моделі та низький рівень помилок для запиту даних спостережень, ніж інші моделі. Деякі автори намагалися спрогнозувати потік доходів фармацевтичних компаній. Одне з таких досліджень [30] мало на меті запровадити новий підхід до прогнозування доходів фармацевтичних компаній. Цей підхід реалізував комбінацію методів прогнозування на основі NN для набору даних часових рядів.

У дослідженні [31] автори запровадили підхід прогнозування для аналізу наявних даних про продажі та порівняли його з різними моделями машинного навчання для прогнозування майбутнього попиту. Метод прогнозування продажів на основі LSTM був розроблений для виявлення нелінійності, наявної в бізнесі

електронної комерції. Результати цього дослідження показали, що LSTM перевершує прогностичні однофакторні методи.

Дослідження [32] запропонувало гібридний підхід, який досліджував довгі часові ряди пам'яті для інтеграції авторегресійної дробово-інтегрованої ковзної середньої (ARFIMA) і нейронних мереж прямого зв'язку (FFNN). Автори сказали, що гібридна модель забезпечила кращу точність прогнозування, ніж інші стандартні підходи.

### **1.3 Огляд та аналіз наявних аналогів**

Прогнозування продажу товарів це дуже важкий процес для людини. Для полегшення даної задачі розробники винайшли інструменти для прогнозування, серед яких багато використовують нейромережеву технологію.

#### **MATLAB.**

Одним із цікавих програмних засобів прогнозування, в яких реалізовано технологію нечіткого моделювання, зокрема прогнозування, є система MATLAB [33]. ANFIS-адаптивна система нейронечіткого виведення є основою для побудови штучної нейронної мережі, яка дозволяє здійснити прогноз стану економіки на майбутній період.

У MATLAB для прогнозування потрібно виконати наступні кроки:

- вибір даних для подальшого аналізу;
- вибір архітектури;
- визначення структури нейронної мережі;
- навчання мережі.
- тестування мережі.

Існує кілька алгоритмів навчання:

- Levenberg-Marquardt (алгоритм Левенберга-Марквардта) призначений для оптимізації параметрів нелінійних регресійних моделей. Передбачається, що критерієм оптимізації використовується середньоквадратична помилка моделі на

навчальній вибірці. Алгоритм полягає в послідовному наближенні заданих початкових значень параметрів до локального оптимуму;

– Bayesian Regularization (алгоритм Байєсівської Регуляризації) – алгоритм, який мінімізує лінійну комбінацію квадратів помилок та ваг. Він також змінює лінійну комбінацію так, щоб наприкінці навчання отримана мережа мала гарні якості узагальнення;

– Scaled Conjugate Gradient (алгоритм сполучених градієнтів) – це різновид методу сполучених градієнтів, що дозволяє уникнути пошуку рядків на одну ітерацію навчання, використовуючи підхід Левенберга-Марквардта для масштабування розміру кроку. Використовуючи механізм масштабування розміру кроку, цей метод дозволяє уникнути трудомісткого пошуку рядків на кожну ітерацію навчання.

### **Deductor.**

Deductor [34] містить усі необхідні інструменти для вирішення складного завдання прогнозування продажів. Використання даної платформи забезпечує підтримку всього циклу робіт: від збору даних до побудови моделей та передачі результатів у сторонні системи. Deductor включає практично всі сучасні алгоритми, що застосовуються для побудови прогностичних моделей від найпростіших до найскладніших.

Крім вирішення окремої задачі прогнозування продажів, існують комплексні рішення автоматизації бізнес-процесів. Один з таких продуктів – Deductor Demand Planning [35], вирішує завдання планування закупівлі товарів з метою мінімізації дефіциту та підвищення оборотності.

### **GMDH Shell.**

GMDH Shell [36], професійне нейромережеве програмне забезпечення, вирішує завдання прогнозування часових рядів та інтелектуального аналізу даних шляхом побудови штучних нейронних мереж та застосування їх до вхідних даних. Вона дозволяє впоратися з будь-яким завданням прогнозування або аналізу даних

з мінімальними зусиллями. Завдяки методу прогнозування GMDH і найсучаснішій технології паралельної обробки, що лежить в основі програми, вона здатна надавати найточніші прогнози часових рядів, і робить це набагато швидше, ніж звичайні штучні нейронні мережі (рис. 1.7).

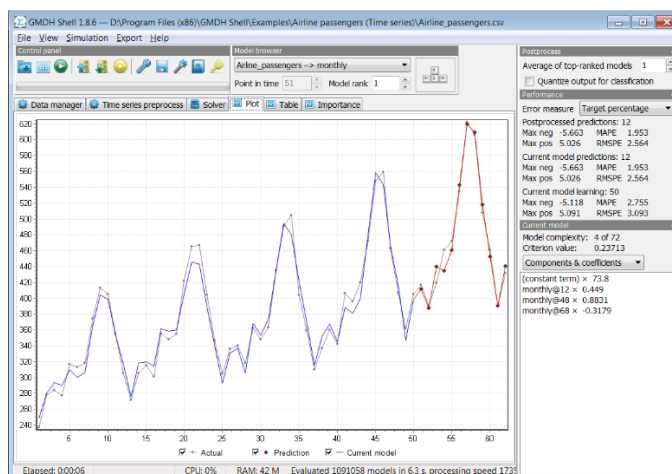


Рисунок 1.7 – Робоча область GMDH Shell

GMDH Shell базується на методі прогнозування GMDH, який є розширеною версією класичного методу регресійного аналізу, розробленого в 60-х роках.

Оболонка GMDH, яка працює на передовій математиці та є універсальним рішенням, ідеальним для прогнозування «під ключ» і легкого аналізу часових рядів. У той же час, повний спектр параметричних налаштувань і вільний вибір методів прогнозування, а також безліч унікальних для галузі опцій дозволяють налаштувати програму під будь-яке конкретне завдання від аналізу біології та хімії до прогнозування продажів, прогнозування фондового ринку, прогнозування запасів, прогнозування електричного навантаження, прогнозування попиту та аналіз часових рядів.

Нейромережеве прогнозування гнучкіше, ніж типові лінійні або поліноміальні апроксимації, і тому більш точне. За допомогою нейронних мереж експерт може виявити та взяти до уваги нелінійні зв'язки та відносини між даними та побудувати модель-кандидат з високою міцністю прогнозування. Крім того, 2023 р.

GMDH Shell не вимагає попередньої нормалізації даних і не дотримується абсолютної точності припасування, що значно скорочує час обчислень. GMDH Shell автоматично тренує нейронні мережі та застосовує їх для аналізу, таким чином, отримання точних спортивних, ділових чи біржових прогнозів не вимагає від вас великих зусиль чи часу. Завдяки унікальній системі балансування навантаження на центральний процесор, GMDH Shell отримує вигоду з усіх вільних ресурсів, які має ваш ПК, спрямовуючи їх застосування на нейромеревевий аналіз. Це означає більш швидкі та точні результати, ніж будь-коли раніше.

### **STATISTICA Automated Neural Networks.**

STATISTICA Automated Neural Networks (SANN) [37] є одним з найпередовіших та найефективніших нейромеревевих продуктів на ринку. Він пропонує безліч унікальних переваг та багатих можливостей (рис. 1.8).

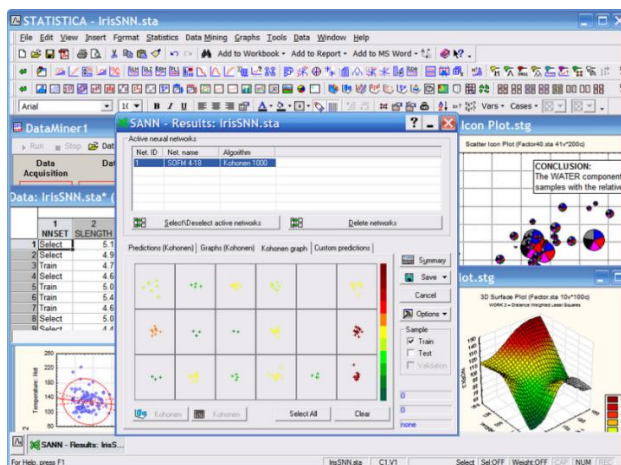


Рисунок 1.8 – Робоча область STATISTICA

### Переваги використання STATISTICA Automated Neural Networks:

- пре- та пост-процесування, включаючи вибір даних, кодування номінальних значень, шкалювання, нормалізацію, видалення пропущених даних з інтерпретацією для класифікації, регресію та завдання часових рядів;

- повний контроль над усіма параметрами, що впливають на якість мережі, такими як функції активації та помилок, складність мережі;
- повна інтеграція із потужними автоматичними інструментами STATISTICA; запис повноцінних макросів для будь-яких аналізів; створення власних нейромережевих аналізів і додатків за допомогою STATISTICA Visual Basic, виклик STATISTICA Automated Neural Networks з будь-якої програми, що підтримує технологію CO;
- вибір найбільш популярних мережевих архітектур, включаючи багатосарові перцептрони, радіальні базисні функції і карти ознак, що самоорганізуються;
- підтримка різного роду статистичного аналізу та побудова прогнозуючих моделей, включаючи регресію, класифікацію, тимчасові ряди з безперервною та категоріальною залежною змінною, кластерний аналіз для зниження розмірності та візуалізації.

Генератор коду STATISTICA Automated Neural Networks може згенерувати вихідний системний програмний код нейромережевих моделей мовами C, Java та PMML (Predictive Model Markup Language). Генератор коду є додатком до системи STATISTICA, який дозволяє користувачам на основі проведеного нейромережевого аналізу генерувати C або Java-файл з вихідним кодом моделей та інтегрувати його до незалежних зовнішніх додатків.

#### **1.4 Постановка задачі**

Дослідивши можливості знайдених платформ з аналітичними інструментами для прогнозування продажів товарів, було виявлено недолік.

Головним недоліком платформ є їх закритість програмного коду, що обмежує змогу використовувати та змінювати внутрішні аналітичні інструменти даних платформ для налаштування потрібних параметрів під свою компанію.



Враховуючи знайдений недолік платформ, створена система матиме відкрите програмне забезпечення, в якому буде можливість використовувати та корегувати потрібні параметри під кожну задачу.

Процес прогнозування виконується за допомогою набору даних, який накопичував дані з продажів минулих років. Тому для отримання правильного та точного результату прогнозування, мінімізації помилки навчання та тестування для даної системи потрібні дані мінімум за три роки.

Основною метою даної роботи є розробка інтелектуальної системи для прогнозування продажу товарів на основі зібраних даних за минулі роки шляхом використання нейронної мережі.

Для реалізації системи потрібно реалізувати наступні задачі:

- проаналізувати системи-аналоги;
- вибрати та використати тип нейронної мережі для прогнозування продажів медикаментів;
- спроектувати інтелектуальну систему для прогнозування продажів медикаментів;
- провести тестування на різних наборах даних для визначення якості роботи системи, точності її прогнозування та швидкості її роботи.

## **Висновки до розділу 1**

Сьогодні прогнозування продажів дуже часто використовують компанії для вирішення задач, зокрема постановка цілей, отримана за річним прогнозом. Це частина бізнес-плану для підприємства і реальна, чітко прорахована мета відділу продажів, від якої можна відштовхуватися компанії.

Інтелектуальні системи набагато прискорюють процес прогнозування та роблять прогнози набагато точніші.

Такий тип прогнозування використовується у наступних процесах:

- формування необхідної бази трудових та виробничих ресурсів, виходячи з прогнозованої кількості клієнтів та обсягу продажів;
- управління складськими запасами;
- підвищення мобільності бізнесу. На прогнозному графіку (чи таблиці) можна заздалегідь побачити моменти можливого просідання обсягу продажу і вжити заходів для коригування ситуації до закінчення періоду;
- контроль та оптимізація витрат. Прогнозування покаже, які витрати загалом понесе компанія на виробництво та реалізацію продукції.

Серед систем-аналогів було розглянуто наступні системи:

- MATLAB;
- Deductor;
- GMDH Shell;
- STATISTICA Automated Neural Networks.

Розглянувши проблеми систем-аналогів, знайдено недолік у вигляді закритості програмного коду, що обмежує змогу використовувати та змінювати внутрішні аналітичні інструменти даних платформ для налаштування потрібних параметрів під свою компанію.

## **2 ТЕХНОЛОГІЇ, МЕТОДОЛОГІЇ ТА АРХІТЕКТУРА ГЛИБОКО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ПРОДАЖІВ МЕДИКАМЕНТІВ**

### **2.1 Технології створення нейромережових систем**

Технологія нейронної мережі [38], іноді відома як штучна нейронна мережа (або ШНМ), – це метод спроби відтворити роботу людського мозку в комп'ютері. Він був розроблений з кінцевою метою дозволити комп'ютеру навчатися на досвіді майже так само, як це роблять люди.

Штучні нейронні мережі (ШНМ) складаються з вузлових шарів, що містять вхідний рівень, один або більше прихованих шарів і вихідний рівень. Кожен вузол, або штучний нейрон, з'єднується з іншим і має відповідну вагу та поріг. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі.

Нейронні мережі поширені у багатьох галузях. В тому числі:

- діагностика за допомогою класифікації медичних зображень;
- цільовий маркетинг за допомогою фільтрації соціальних мереж та аналізу поведінкових даних;
- фінансові прогнози за допомогою обробки історичних даних фінансових інструментів;
- прогнозування електричного навантаження та потреби в енергії;
- контроль відповідності вимогам та якості;
- визначення хімічних сполук.

Нижче представлені чотири найважливіші завдання, які допомагають вирішити нейронні мережі.

#### **Машинний зір.**

Машинний зір – це здатність комп'ютерів отримувати інформацію та сенс із зображень та відео. За допомогою нейронних мереж комп'ютери можуть розрізняти

та розпізнавати зображення так, як це роблять люди. Машинний зір застосовується в кількох областях, наприклад:

- візуальне розпізнавання у безпілотних автомобілях, щоб вони могли реагувати на дорожні знаки та інших учасників руху;
- модерація контенту для автоматичного видалення небезпечного або неприйняттого контенту з архівів зображень та відео;
- розпізнавання осіб для ідентифікації людей та розпізнавання таких атрибутів, як відкриті очі, окуляри та рослинність на обличчі;
- маркування зображення для ідентифікації логотипів бренду, одягу, захисного спорядження та інших деталей зображення.

### **Розпізнавання мови.**

Нейронні мережі можуть аналізувати людську мову незалежно від її мовних моделей, висоти, тону, мови та акценту. Віртуальні помічники, такі як Amazon Alexa та програмне забезпечення для автоматичної транскрипції, використовують розпізнавання мови для виконання наступних завдань:

- допомога операторам кол-центру та автоматична класифікація дзвінків;
- перетворення клінічних рекомендацій на документацію в режимі реального часу;
- точні субтитри до відео та записів нарад для ширшого охоплення контенту.

### **Обробка природної мови.**

Обробка природної мови (NLP) – це здатність обробляти природний, створений людиною текст. Нейронні мережі допомагають комп'ютерам отримувати інформацію та зміст із текстових даних та документів. NLP має кілька сфер застосування, у тому числі:

- автоматизовані віртуальні агенти та чат-боти;
- автоматична організація та класифікація записаних даних;

- бізнес-аналітика довгих документів: наприклад, електронних листів та форм;
- індексація ключових фраз, що вказують на настрої: наприклад, позитивних та негативних коментарів у соціальних мережах;
- узагальнення документів та генерація статей з заданої теми.

### **Сервіси рекомендацій.**

Нейронні мережі можуть відстежувати дії користувачів для розробки персональних рекомендацій. Вони також можуть аналізувати всі дії користувачів та виявляти нові продукти чи послуги, які цікавлять конкретного споживача. Наприклад, стартап із Філадельфії Curalate допомагає брендам конвертувати повідомлення у соціальних мережах у продажі. Бренди використовують службу інтелектуального маркування продуктів (IPT) Curalate для автоматизації збору та обробки контенту користувачів соціальних мереж. IPT використовує нейронні мережі для автоматичного пошуку та рекомендації продуктів, що відповідають активності користувача у соціальних мережах. Споживачам не потрібно копатися в онлайн-каталогах, щоб знайти конкретний продукт із зображення в соціальних мережах. Натомість вони можуть використовувати автоматичне маркування Curalate.

### **Прогнозування.**

Однією з основних сфер застосування ШНМ є прогнозування. ШНМ є привабливим альтернативним інструментом як для дослідників прогнозів, так і для практиків. Деякі відмінні риси ШНМ роблять їх цінними та привабливими для завдання прогнозування. По-перше, на відміну від традиційних методів, заснованих на моделях, ШНМ є самоадаптивними методами, керованими даними, оскільки існує кілька апріорних припущень щодо моделей для досліджуваних проблем. Вони навчаються на прикладах і вловлюють тонкі функціональні зв'язки між даними, навіть якщо базові зв'язки невідомі або важко описати. Таким чином, ШНМ добре підходять для проблем, вирішення яких вимагає знань, які важко

визначити, але для яких є достатньо даних або спостережень. У цьому сенсі їх можна розглядати як один із багатовимірних нелінійних непараметричних статистичних методів. Цей підхід до моделювання з можливістю вивчення досвіду є дуже корисним для вирішення багатьох практичних проблем, оскільки часто легше мати дані, ніж мати хороші теоретичні припущення про базові закони, що керують системами, з яких генеруються дані.

Навчання нейронної мережі це досить трудомісткий процес, який має свої закономірності та інструменти. В наш час нейромережі можна зустріти у безлічі сфер, серед яких фінансове прогнозування найбільше зустрічається. Саме нейронні мережі ефективно та швидше працюють, що дає змогу компанії отримати більший прибуток.

Перед самим процесом прогнозування, необхідно розглянути набір функціональних можливостей, які треба проаналізувати та обрати для зручності та якості створення системи.

Функціональні характеристики нейромережевих технологій.

#### **Збереження даних:**

- необхідність зображення великого обсягу даних. Можливість зберігати великий обсяг даних, щоб покривати постійно зростаючі потреби у пам'яті;
- доступність. Доступність даних при помилках у роботі системи або часткових відмовах.

#### **Дослідження даних:**

- візуалізація підсумкових таблиць і шаблонів у вхідних даних. Уміння знаходити закономірності у вхідних даних.

#### **Підготовка даних:**

- вилучення функцій. Маніпулювання вихідними даними для вилучення функцій, необхідних для виконання алгоритму;
- розподілене виконання. Можливість розподіленого маніпуляції з даними при великому обсязі даних для скорочення часу виконання операцій.

**Розробка:**

- підтримувані мови. Підтримка мов програмування для розробки;
- легкість розробки. Наскільки легко розробляти і виконувати скрипти на платформі.

**Модель:**

- підтримувані алгоритми. Наявність різноманітних реалізацій алгоритмів;
- розподілене виконання при створенні моделі. Можливість розподіленого створення моделей для економії часу;
- підтримка глибокого навчання. Підтримка алгоритмів глибокого навчання;
- підтримка графічного процесора. Підтримка виконання на GPU;
- гнучкість навчання моделі. Рівень, на якому можна налаштовувати параметри навчання моделі;
- гнучкість перевірки моделі. Рівень, на якому можна налаштовувати перевірку моделі в результаті навчання;
- легкість переключення між моделями. Переключення між різними моделями під час роботи.

**Візуалізація даних:**

- візуалізація та створення графіків результатів. Наявність різноманітних діаграм для візуалізації результатів.

**Виробництво:**

- простота розгортання моделі у виробничому середовищі на вебсервері. Можливість розгортання моделі в Інтернеті.

**Підтримка:**

- офіційна підтримка з активним розвитком. Наявність комерційної підтримки для платформи або рішення.

Наступною задачею є визначити та обрати метод прогнозування. Їх поділяють на дві групи [7]:

– **інтуїтивні** – мають справу з судженнями та оцінками експертів. На сьогоднішній день вони часто застосовуються в маркетингу, економіці, політиці, оскільки система, поведінку якої необхідно спрогнозувати, або дуже складна і не піддається математичному опису, або дуже проста і такого опису не потребує;

– **формалізовані** – описані у літературі методи прогнозування, у яких будують моделі прогнозування, тобто визначають таку математичну залежність, що дозволяє обчислити майбутнє значення процесу, тобто зробити прогноз.

Серед методів, для даної роботи обрано метод – формалізований.

Далі потрібно обрати модель прогнозування у формалізованому методі. Моделі поділяють на дві групи:

– **моделі предметної області** – такі математичні моделі прогнозування, для побудови яких застосовують закони предметної галузі. Наприклад, модель, на якій роблять прогноз погоди, містить рівняння динаміки рідин та термодинаміки. Прогноз розвитку популяції виробляється моделі, побудованої на диференціальному рівнянні;

– **моделі часових рядів** – математичні моделі прогнозування, які прагнуть знайти залежність майбутнього значення від минулого всередині самого процесу і на цій залежності обчислити прогноз. Ці моделі універсальні для різних предметних областей, тобто їхній загальний вигляд не змінюється в залежності від природи тимчасового ряду. Їх можна використовувати у нейронних мережах для прогнозування температури повітря, а потім аналогічну модель на нейронних мережах застосувати для прогнозу біржових індексів.

Для даної роботи обрано модель часових рядів, так як вона більш краща для прогнозування з використанням нейронної мережі.

Наступним кроком є вибір моделі часових рядів. Їх можна поділити на дві групи:

1) **статичні моделі** [10]. У статистичних моделях залежність майбутнього значення від минулого задається як деякого рівняння. До них відносяться:



- регресійні моделі (лінійна регресія, нелінійна регресія);
- авторегресійні моделі (ARIMAX, GARCH, ARDLM);
- модель експонентного згладжування;
- модель з вибірки максимальної подоби та інші.

2) **структурні моделі** [10]. У структурних моделях залежність майбутнього значення від минулого задається у вигляді деякої структури та правил переходу по ній. До них відносяться:

- нейромережеві моделі;
- моделі з урахуванням ланцюгів Маркова;
- моделі на основі класифікаційно-регресійних дерев та інші.

Для прогнозування з використанням нейромережі краще підходять структурні моделі, тому обрані саме вони.

## 2.2 Інструменти, методології для вирішення поставленої задачі

Дуже важливим кроком є вибір мови програмування. Для вирішення даної задачі підходять наступні мови:

- R;
- Python;
- Java;
- C/C++;
- JavaScript.

Якщо подивитись на загальну популярність мов машинного навчання. Python [39] лідирує в списку: 57% дослідників даних і розробників машинного навчання використовують його, а 33% віддають йому пріоритет у розробці. Не дивно, враховуючи всю еволюцію фреймворків глибокого навчання Python за останні 2 роки, включаючи випуск TensorFlow і широкий вибір інших бібліотек. Python часто порівнюють із R, але їх далеко не порівняти за популярністю: R посідає четверте місце за загальним використанням (31%) і п'яте за пріоритетністю (5%). Фактично

R є мовою з найнижчим співвідношенням пріоритетності до використання серед п'яти, лише 17% розробників, які її використовують, надають їй пріоритет. Це означає, що в більшості випадків R є додатковою мовою, а не першим вибором. Таке ж співвідношення для Python становить 58%, що є найвищим серед п'яти мов, це чітка ознака того, що тенденції використання Python прямо протилежні тенденціям використання R. Python є не лише найпоширенішою мовою, але й основним вибором для більшості її користувачів.

C/C++ з великим відривом поступається Python як за використанням (44%), так і за пріоритетністю (19%). Java дуже сильно слідує за C/C++, тоді як JavaScript займає п'яте місце за використанням, хоча з дещо кращою продуктивністю пріоритетності, ніж R (7%).

Інші мови, які використовуються в машинному навчанні, включаючи звичайні підозрювані Julia, Scala, Ruby, Octave, MATLAB і SAS, але всі вони опускаються нижче 5% пріоритетності та нижче 26% використання.

Науковці з машинного навчання, які працюють над аналізом настроїв, більше віддають перевагу Python (44%) і R (11%), а JavaScript (2%) і Java (15%) менше, ніж розробники, що працюють в інших областях. Навпаки, ті, хто працює над мережевою безпекою/кібератаками та виявленням шахрайства, надають більше пріоритету Java – двом сферам, де Python має найменший пріоритет. Алгоритми мережевої безпеки та виявлення шахрайства створюються або використовуються переважно у великих організаціях, особливо у фінансових установах, де Java є фаворитом більшості внутрішніх команд розробників. У сферах, які менш орієнтовані на підприємства, як-от обробка природної мови (NLP) і аналіз настроїв, розробники обирають Python, який пропонує простіший і швидший спосіб створення високопродуктивних алгоритмів завдяки великій колекції спеціалізованих бібліотек, які постачаються разом із ним.

Для розробки системи використано мову Python. Саме ця мова найчастіше використовується для вирішення подібних задач з використанням нейронних

мереж. Така популярність обумовлена її простотою та зручністю, тому що існує безліч бібліотек для полегшення вирішення різноманітних складних задач.

Python – це активно розвиваюча скриптова мова, яку використовують для вирішення великого обсягу найрізноманітніших проблем та завдань. Python стане в нагоді у створенні комп'ютерних та мобільних додатків, його застосовують у роботі з великим обсягом інформації, при розробці web-сайтів та інших різноманітних проектів, використовують у машинному навчанні.

Він має величезну кількість бібліотек і фреймворків: мова Python постачається з багатьма бібліотеками та фреймворками, які полегшують кодування. Це також значно економить час.

Найпопулярнішими бібліотеками є NumPy, яка використовується для наукових розрахунків; SciPy для складніших обчислень; і scikit для вивчення інтелектуального аналізу даних.

Ці бібліотеки працюють разом із такими потужними фреймворками, як TensorFlow, CNTK і Apache Spark. Ці бібліотеки та фреймворки необхідні, коли мова йде про проекти машинного та глибокого навчання.

Простота: код Python є лаконічним і читабельним навіть для нових розробників, що корисно для проектів машинного та глибокого навчання. Завдяки простому синтаксису розробка додатків на Python є швидкою порівняно з багатьма мовами програмування. Крім того, це дозволяє розробнику тестувати алгоритми без їх впровадження.

Читальний код також життєво важливий для спільного кодування. Багато людей можуть працювати разом над складним проектом.

Можна легко знайти розробника Python для команди, оскільки Python є знайомою платформою. Таким чином, новий розробник може швидко ознайомитися з концепціями Python і миттєво працювати над проектом.

Масштабна онлайн-підтримка: Python є мовою програмування з відкритим кодом і користується відмінною підтримкою багатьох ресурсів і якісної

документації по всьому світу. Він також має велику та активну спільноту розробників, які надають допомогу на будь-якому етапі розробки.

Більшість науковців прийняли Python для проектів машинного та глибокого навчання, а це означає, що більшість найяскравіших умів у всьому світі можна знайти в спільнотах Python.

Швидка розробка: Python має легкий для розуміння та зрозумілий синтаксис. Крім того, численні фреймворки та бібліотеки прискорюють розробку програмного забезпечення. Використовуючи готові рішення, багато чого можна зробити за допомогою декількох рядків коду. Python хороший для розробки прототипів, що підвищує продуктивність.

Гнучка інтеграція: проекти Python можна інтегрувати з іншими системами, закодованими різними мовами програмування. Це означає, що його набагато легше поєднати з іншими проектами AI, написаними іншими мовами.

Крім того, оскільки Python є розширюваним і портативним, його можна використовувати для виконання міжмовних завдань. Адаптивність Python полегшує науковцям і розробникам даних навчати моделі машинного навчання.

Швидкі тести коду: Python надає багато інструментів для перевірки коду та тестування. Розробники можуть швидко перевірити правильність і якість коду.

Проекти штучного інтелекту, як правило, займають багато часу, тому потрібне добре структуроване середовище для тестування та перевірки на помилки. Python є ідеальною мовою, оскільки він підтримує ці функції.

Продуктивність. Деякі розробники стверджують, що Python відносно повільний порівняно з іншими мовами програмування. Незважаючи на те, що швидкість не є однією з сильних сторін Python, вона забезпечує рішення, відоме як Cython. Це надмножина мови Python, призначена для досягнення такої ж продуктивності коду, як і мова C.

Розробники можуть використовувати Cython для кодування розширень C так само, як вони кодують у Python, оскільки його синтаксис майже однаковий. Cython значно підвищує продуктивність мови.

Інструменти візуалізації: Python постачається з великою різноманітністю бібліотек. Деякі з цих фреймворків пропонують хороші інструменти візуалізації. У штучному інтелекті, машинному та глибокому навчанні важливо подавати дані в зручному для читання форматі. Тому Python є ідеальним вибором для реалізації цієї функції.

Деякі бібліотеки, такі як Matplotlib, дозволяють дослідникам даних створювати діаграми, гістограми та графіки для кращого представлення даних і візуалізації. Крім того, різні API, які підтримує Python, покращують процес візуалізації.

Для розробки використано наступні бібліотеки:

- Scikit-Learn;
- Pandas;
- NumPy;
- Matplotlib.

### **Scikit-Learn.**

Scikit-Learn [40] – один із найбільш широко використовуваних пакетів Python для Data Science та Machine Learning. Він дозволяє виконувати безліч операцій та надає безліч алгоритмів. Scikit-Learn також пропонує чудову документацію про свої класи, методи та функції, а також опис алгоритмів, що використовуються;

Scikit-Learn підтримує:

- попередню обробку даних;
- зменшення розмірності;
- вибір моделі;
- регресії;
- класифікації;

- кластерний аналіз.

Компоненти scikit-learn:

- алгоритми керованого навчання: будь-який алгоритм керованого машинного навчання з дуже високою ймовірністю є частиною scikit-learn. Починаючи від узагальнених лінійних моделей (наприклад, лінійної регресії), опорних векторних машин (SVM), дерев прийняття рішень до байєсівських методів – усі вони є частиною інструментарію scikit-learn. Поширення алгоритмів машинного навчання є однією з головних причин широкого використання scikit-learn;
- перехресна перевірка: існують різні методи перевірки точності контрольованих моделей на невидимих даних за допомогою sklearn;
- алгоритми неконтрольованого навчання. Є велика кількість алгоритмів машинного навчання – починаючи від кластеризації, факторного аналізу, аналізу головних компонентів до неконтрольованих нейронних мереж;
- різні набори даних: це стало в нагоді під час навчання scikit-learn;
- вилучення функцій: Scikit-learn для вилучення функцій із зображень і тексту (наприклад, Bag of words).

### **Pandas.**

Pandas [41] – це пакет Python з відкритим вихідним кодом, який найбільш широко використовується для завдань науки про дані/аналізу даних і машинного навчання. Він побудований на основі іншого пакета під назвою NumPy, який забезпечує підтримку багатовимірних масивів. Будучи одним із найпопулярніших пакетів обробки даних, Pandas добре працює з багатьма іншими модулями науки про дані в екосистемі Python і зазвичай включається в кожен дистрибутив Python, від тих, що постачаються з вашою операційною системою, до комерційних дистрибутивів постачальників, таких як ActivePython від ActiveState.

Pandas спрощує виконання багатьох трудомістких і повторюваних завдань, пов'язаних із роботою з даними, зокрема:

- очищення даних;
- заповнення даних;
- нормалізація даних;
- зливається і приєднується;
- візуалізація даних;
- статистичний аналіз;
- перевірка даних;
- завантаження та збереження даних та інше.

### **NumPy.**

NumPy [42] – один із найбільш часто використовуваних пакетів для наукових обчислень на Python. Він надає об'єкт багатовимірного масиву, а також варіанти, такі як маски та матриці, які можна використовувати для різних математичних операцій. NumPy сумісний і використовується багатьма іншими популярними пакетами Python, включаючи pandas і matplotlib.

NumPy робить багато математичних операцій, які широко використовуються в наукових обчисленнях, швидкими та простими у використанні, наприклад:

- множення вектора на вектор;
- множення матриця-матриця та матриця-вектор;
- елементні операції над векторами та матрицями (тобто додавання, віднімання, множення та ділення на число);
- елементне або масивне порівняння;
- застосування функцій для кожного елемента до вектора/матриці (наприклад, pow, log і exp);
- багато операцій лінійної алгебри можна знайти в NumPy.linalg;
- скорочення, статистика та багато іншого.

### **Matplotlib.**

Matplotlib [43] – це кросплатформна бібліотека візуалізації даних і графічного побудови для Python і його числового розширення NumPy. Таким

чином, він пропонує життєздатну альтернативу MATLAB з відкритим кодом. Розробники також можуть використовувати API matplotlib (інтерфейси прикладного програмування) для вбудовування графіків у програми GUI.

Сценарій Matplotlib Python структурований так, що в більшості випадків для створення візуальної діаграми даних потрібно лише кілька рядків коду. Рівень сценаріїв matplotlib накладає два API:

- API pyplot — це ієрархія об'єктів коду Python, очолювана matplotlib.pyplot;
- набір об'єктів OO (об'єктно-орієнтованого) API, який можна зібрати з більшою гнучкістю, ніж pyplot. Цей API забезпечує прямий доступ до базових рівнів Matplotlib.

API pyplot має зручний інтерфейс зі збереженням стану в стилі MATLAB. Насправді matplotlib спочатку був написаний як альтернатива MATLAB з відкритим кодом. API та його інтерфейс є більш настроюваними та потужними, ніж pyplot, але вважаються більш складними у використанні. Як наслідок, інтерфейс pyplot використовується частіше.

### **2.3 Архітектура глибокого навчання LSTM**

Глибоке навчання швидко змінило область обробки природної мови з її нейронними мережами, що повторюються. LSTM – одна з таких популярних осередків, що використовуються для побудови цих повторюваних архітектур глибокого навчання на основі нейронних мереж.

LSTM – це особливий тип блоку, який вимагає стану осередку і прихованого стану разом з вхідними даними на кожній тимчасовій мітці для виконання своїх операцій. По суті, LSTM складається з трьох типів вентилів, а саме: вентиль забуття, вхідний вентиль і вихідний вентиль, який визначає релевантну і нерелевантну інформацію з вхідних даних.



Це різновид рекурентних нейронних мереж (RNN [44]), які здатні вивчати довгострокові залежності, особливо в задачах прогнозування послідовності. LSTM має зв'язки зі зворотним зв'язком, тобто він здатний обробляти всю послідовність даних, за винятком окремих точок даних, таких як зображення. Це знаходить застосування в розпізнаванні мовлення, машинному перекладі тощо. LSTM – це особливий вид RNN, який демонструє видатну продуктивність у багатьох проблемах.

### Логіка LSTM

Центральну роль у моделі LSTM відіграє комірка пам'яті, відома як «стан комірки», яка зберігає свій стан з часом. Стан комірки – це горизонтальна лінія, яка проходить через верхню частину діаграми нижче. Це можна візуалізувати як конвеєр, через який інформація просто тече, незмінна (рис. 2.1).

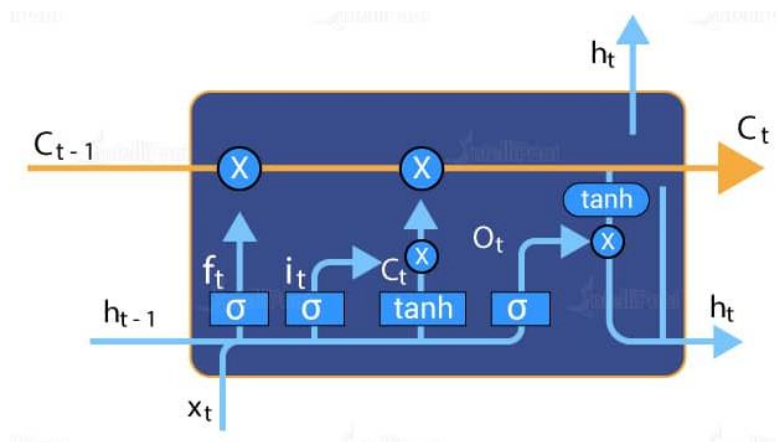


Рисунок 2.1 – Комірка пам'яті LSTM

Інформація може бути додана або видалена зі стану комірки в LSTM і регулюється вентилями. Ці ворота додатково пропускають інформацію в клітину та з неї. Він містить операцію поточкового множення та шар сигмоподібної нейронної мережі, які допомагають механізму (рис. 2.2).

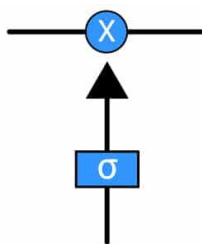


Рисунок 2.2 – Операція поточкового множення та шар сигмоподібної нейронної мережі

Сигмоїдний шар видає числа від нуля до одиниці, де нуль означає «нічого не повинно пропускатися», а одиниця означає «все має пропускатися».

### LSTM проти RNN

Для прикладу взято задачу зміни певної інформації в календарі. Для цього RNN повністю змінює існуючі дані, застосовуючи функцію. У той час як LSTM вносить невеликі зміни в дані шляхом простого додавання або множення, які проходять через стани комірки. Таким чином LSTM забуває та запам'ятовує речі вибірково, що робить його кращим у порівнянні з RNN.

Отже, потрібно знати, які дані потрібно забути, а які потрібно зберегти для подальшого використання. В іншому випадку потрібно мати дуже хорошу пам'ять. Теоретично рекурентні нейронні мережі, добре справляються з цим. Однак у них є два недоліки: вибуховий градієнт і зникаючий градієнт, які роблять їх зайвими.

Тут LSTM вводить одиниці пам'яті, які називаються станами комірки, щоб вирішити цю проблему. Спроектвані клітини можна розглядати як диференційовану пам'ять.

### Програми LSTM

Мережі LSTM знаходять корисні застосування в таких областях:

- мовне моделювання;
- машинний переклад;
- розпізнавання рукописного тексту;
- підписи до зображень;

- генерація образів за допомогою моделей уваги;
- відповідь на запитання;
- перетворення відео в текст;
- поліморфне музичне моделювання;
- синтез мовлення;
- прогнозування вторинної структури білка.

Цей список дає уявлення про сфери, в яких використовується LSTM, але не про те, як саме він використовується.

Нейронні мережі LSTM [45] здатні вирішувати численні завдання, які не вирішуються попередніми алгоритмами навчання, такими як RNN. Довгострокові тимчасові залежності можуть бути ефективно охоплені LSTM без особливих перешкод для оптимізації. Це використовується для вирішення проблем високого рівня.

### **Двонаправлені LSTM**

У двонаправлених LSTM кожна навчальна послідовність представлена вперед і назад, щоб розділити повторювані мережі. Обидві послідовності підключені до одного вихідного рівня. Двонаправлені LSTM мають повну інформацію про кожну точку в даній послідовності, все до і після неї.

Людський мозок використовує органи чуття, щоб отримати інформацію зі слів, звуків або цілих речень, які спочатку можуть не мати сенсу, але щось означатимуть у майбутньому контексті. Звичайні рекурентні нейронні мережі здатні використовувати лише попередній контекст для отримання інформації. У той час як у двонаправлених LSTM інформація отримується шляхом обробки даних в обох напрямках у межах двох прихованих шарів, що надходять до одного вихідного рівня. Це допомагає двонаправленим LSTM отримувати доступ до далекого контексту в обох напрямках.

Для тимчасових рядів, таких як текст, сигнали, ціни на акції і т. д., архітектура LSTM є найбільш підходящою для моделювання тимчасових зв'язків у глибоких

нейронних мережах. Вона долає проблему зникаючого градієнта в рекурентній нейронній мережі для довгострокового навчання залежності даних з використанням осередків пам'яті та вентилів.

Під час процесу навчання мережі головною метою є мінімізація втрат (з точки зору помилок або вартості), які спостерігаються на виході, коли навчальні дані надсилаються через нього. Ми обчислюємо градієнт, тобто втрату відносно певного набору ваг, відповідно коригуємо вагові коефіцієнти та повторюємо цей процес, доки не отримаємо оптимальний набір ваг, для якого втрата є мінімальною. Це концепція бектрекінгу.

Іноді буває так, що градієнт практично незначний. Слід зазначити, що градієнт шару залежить від певних компонентів у наступних шарах. Якщо деякі з цих компонентів малі (менше 1), отриманий результат, яким є градієнт, буде ще меншим. Це відоме як ефект масштабування. Коли цей градієнт помножити на швидкість навчання, яка сама по собі є невеликим значенням у межах 0,1-0,001, це призводить до меншого значення. Як наслідок, зміна ваги є досить незначною, що дає майже такий самий результат, як і раніше.

Подібним чином, якщо значення градієнтів є досить великими через великі значення компонентів, ваги оновлюються до значення, що перевищує оптимальне значення. Це відоме як проблема вибухових градієнтів. Щоб уникнути цього ефекту масштабування, блок нейронної мережі було перебудовано таким чином, що коефіцієнт масштабування був зафіксований на одиниці. Потім комірку було збагачено декількома блоками затворів і названо LSTM. Це відоме як проблема вибухових градієнтів. Щоб уникнути цього ефекту масштабування, блок нейронної мережі було перебудовано таким чином, що коефіцієнт масштабування був зафіксований на одиниці. Потім комірку було збагачено декількома блоками затворів і названо LSTM. Це відоме як проблема вибухових градієнтів. Щоб уникнути цього ефекту масштабування, блок нейронної мережі було перебудовано

таким чином, що коефіцієнт масштабування був зафіксований на одиниці. Потім комірку було збагачено декількома блоками затворів і названо LSTM.

Згорткові нейронні мережі є групою алгоритмів глибокого навчання, які спочатку були розроблені для класифікації зображень. Мережа бере зображення, пропускає його через набір фільтрів, застосовуючи вагові коефіцієнти до різних аспектів зображення, і зрештою будує прогноз. Це працює як система відбору ознак, завдяки якій з часом мережа "дізнається", які фільтри аспектів найбільше важливі класифікації зображення. Аналогічний метод можна застосувати та до тимчасових рядів. Хоча тимчасовий ряд не має "фізичних" ознак, як зображення, дані часового ряду містять просторові ознаки часу. Якщо тимчасовий ряд можна подати як зображення, то згортку мережу можна представити як ковзне вікно, яке сканує через даний тимчасовий період, відображаючи форму нашого ряду в цей період, а потім фільтруючи його, так щоб знайти ознаку, яка найбільше її характеризує. Проста ілюстрація цього підходу наведено на рис. 2.3.

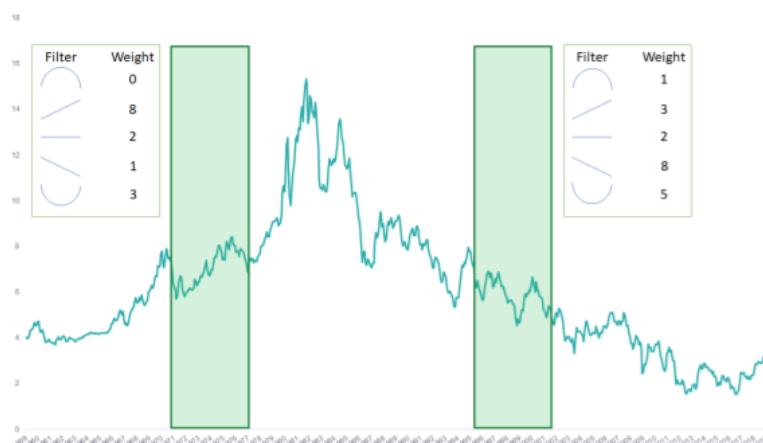


Рисунок 2.3 – Подання ковзного вікна згорткової нейронної мережі для відбору ознак

Ця згорткова модель може бути розширена за допомогою включення до її архітектури прихованих шарів long short-term memory (LSTM), щоб з'ясувати, які з цих просторових показників надають максимальний вплив на швидкість навчання

мережі. Осередок LSTM складається з кількох функцій вентилів, які визначають, чи нова інформація є важливою для завдання прогнозування і чи залишається стара інформація актуальною. Ця пам'ять називається станом осередку і може зберігати всю раніше вивчену релевантну інформацію для повної обробки послідовності тимчасових рядів. Це дозволяє зберігати інформацію, отриману набагато раніше у послідовності, під час повної обробки. У цьому полягає основна перевага архітектури LSTM перед іншими структурами RNN. LSTM здатний переносити інформацію від початку обробки до кінця, тоді як інші мережі RNN просто оновлюють розуміння з кожним додатковим входом у послідовності. Ці особливості роблять мережу LSTM дуже потужною для аналізу послідовностей, таких як часові ряди.

Зі зростанням популярності LSTM в звичайну архітектуру LSTM випробували різні зміни, щоб спростити внутрішню конструкцію комірок, щоб вони працювали ефективніше та зменшили обчислювальну складність. Герс і Шмідхубер запровадили з'єднання «peerhole», які дозволили рівням воріт мати інформацію про стан клітини в кожному мить. Деякі LSTM також використовували зв'язаний вхід і забутий шлюз замість двох окремих шлюзів, що допомагало приймати обидва рішення одночасно. Іншим варіантом було використання Gated Recurrent Unit (GRU), який покращив складність конструкції за рахунок зменшення кількості воріт. Він використовує комбінацію стану комірки та прихованого стану, а також шлюз оновлення, який забув, і вхідні шлюзи об'єднані в нього.

На рис. 2.4 показано LSTM (A), DLSTM (B), LSTM (C) і DLSTM (D)

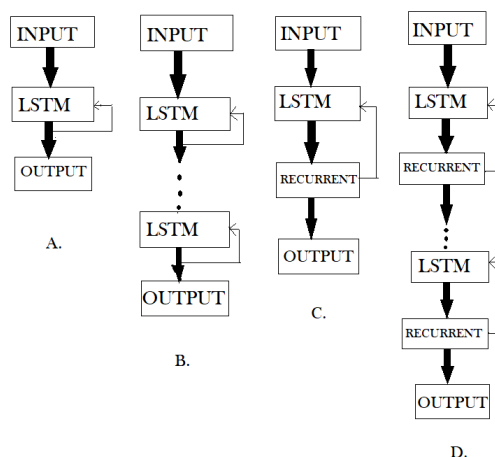


Рисунок 2.4 – LSTM, DLSTM, LSTMP і DLSTMP

На фрагменті А показано, як виглядає базова мережа LSTM. Тут показано лише один шар LSTM між вхідним і вихідним шарами.

На фрагменті В зображено Deep LSTM, який включає кілька рівнів LSTM між входом і виходом. Перевага полягає в тому, що вхідні значення, що подаються в мережу, не тільки проходять через кілька рівнів LSTM, але й поширюються в часі в одній комірці LSTM. Таким чином, параметри добре розподіляються між кількома шарами. Це призводить до ретельного процесу введення на кожному кроці часу.

Фрагмент С представляє LSTM із шаром повторюваної проекції, де повторювані з'єднання беруться з рівня проекції на вхід рівня LSTM. Ця архітектура була розроблена, щоб зменшити високу обчислювальну складність навчання ( $O(N)$ ) для кожного тимчасового кроку) стандартного LSTM RNN.

Фрагмент D зображено Deep LSTM із повторюваним проекційним шаром, що складається з кількох шарів LSTM, де кожен шар має власний проекційний шар. Збільшена глибина дуже корисна у випадку, якщо розмір пам'яті занадто великий. Збільшення глибини запобігає переобладнанню моделей, оскільки входи в мережу повинні проходити через багато нелінійних функцій.

Крім нейронних мереж для прогнозування використовувався метод градієнтного бустингу XGBoost. XGBoost-алгоритм машинного навчання, 2023 р. Єфімов О. І. 122 – МКР – 601. 21710808

спрямований на прискорення процесу побудови дерев рішень. Технічні індикатори, такі як MACD, Bollinger, MA, EWM та інші можуть бути використані як ознаки для навчання моделі XGBoost.

## Висновки до розділу 2

Проаналізувавши технології створення нейронних мереж, було обрано наступні технології:

- серед методів, для даної роботи обрано метод – формалізований;
- для даної роботи обрано модель часових рядів, так як вона більш краща для прогнозування з використанням нейронної мережі;
- для прогнозування з використанням нейромережі краще підходять структурні моделі, тому обрані саме вони.

Для розробки системи використано мову Python. Саме ця мова найчастіше використовується для вирішення подібних задач з використанням нейронних мереж. Така популярність обумовлена її простотою та зручністю, тому що існує безліч бібліотек для полегшення вирішення різноманітних складних задач.

Для розробки використано наступні бібліотеки:

- Scikit-Learn;
- Pandas;
- NumPy;
- Matplotlib.

Для тимчасових рядів, таких як текст, сигнали, ціни на акції і т. д., архітектура LSTM є найбільш підходящою для моделювання тимчасових зв'язків у глибоких нейронних мережах. Вона долає проблему зникаючого градієнта в рекурентній нейронній мережі для довгострокового навчання залежності даних з використанням осередків пам'яті та вентилів.



### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

В даному розділі показано програмну реалізацію, проведено тестування та сформовано висновки отриманої інтелектуальної системи.

Програмна реалізація складається з 12 підрозділів.

#### 3.1 Встановлення програмного забезпечення та їх підключення.

Для розробки системи встановлено програмне забезпечення:

- PIP;
- Python 3.6;
- програмне забезпечення CUDA від NVIDIA. Вона потрібна для встановлення бібліотеки TensorFlow;
- бібліотеки TensorFlow (Keras), Scikit-Learn, Pandas, NumPy и Matplotlib.

#### 3.2 Підключення набору даних та перевірка його роботи.

Дані набору використані у форматі «csv». Вони представлені на рис. 3.1.

```
"Month", "Sales"  
2020-01, 3448.12  
2020-02, 5576.23  
2020-03, 2964  
2020-04, 4814.6  
2020-05, 2798  
2020-06, 3674.25  
2020-07, 2558.3  
2020-08, 4165.9  
2020-09, 5938.67  
2020-10, 5650.45  
2020-11, 5847.27  
2020-12, 5360.31  
2021-01, 6595.16  
2021-02, 5543.29  
2021-03, 4103.58  
2021-04, 5935.37  
2021-05, 3749.28  
2021-06, 4780.02
```

```

2021-07,4974.34
2021-08,6011
2021-09,6673.5
2021-10,5314.23
2021-11,4294.14
2021-12,6745.09
2022-01,4011.29
2022-02,4132.3
2022-03,5245
2022-04,6853.45
2022-05,6442.4
2022-06,5272
2022-07,4073.5
2022-08,5291.58
2022-09,7956
2022-10,4846.55
2022-11,3027.22
2022-12,4571.11

```

Рисунок 3.1 – Набір даних

Структура набору даних представлена у табл. 3.1.

Таблиця 3.1 – Структура набору даних

Ім'я стовпця	Тип даних	Дозволити нульові значення
Month	text	NOT NULL
Sales	float	NULL

Стовпець «Month» має текстовий тип даних для обрізання дати до потрібного формату, коректного відображення на графіку та не дозволено мати нульові значення, тому що такі значення призведуть до некоректного місячного розрахунку та відображення підписів осі x.

Стовпець «Sales» має нецілочисловий тип даних, тому що дані набору представлені у мільярдах гривень. Також йому дозволено мати нульові значення через те, що купівля медикаментів може не відбутись і значення продажів буде рівна 0.

Наступним кроком є підключення набору даних. Воно відбувається у змінній «series», продемонстрованій на рис 3.2.

```
# load dataset  
series = read_csv('med_sales.csv', header=0, parse_dates=[0], index_col=0, squeeze=True, date_parser=parser)
```

Рисунок 3.2 – Підключення набору даних

Для візуального перегляду даних про продажі, використана бібліотека Matplotlib, яка демонструє набір даних у вигляді графіка (рис. 3.3).

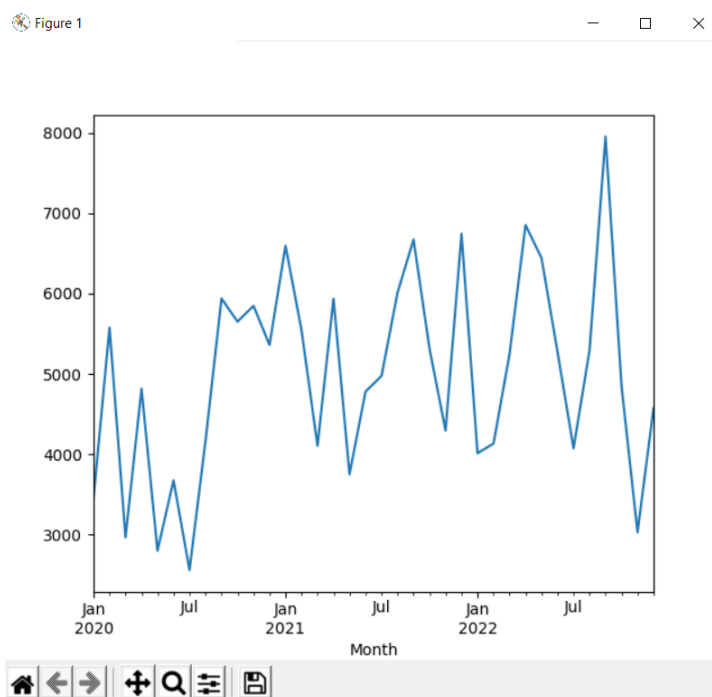


Рисунок 3.3 – Графічне представлення набору даних

Для перевірки правильної роботи системи, виконано зміни в наборі та перевірено результат. Результат зображений на рис. 3.4.

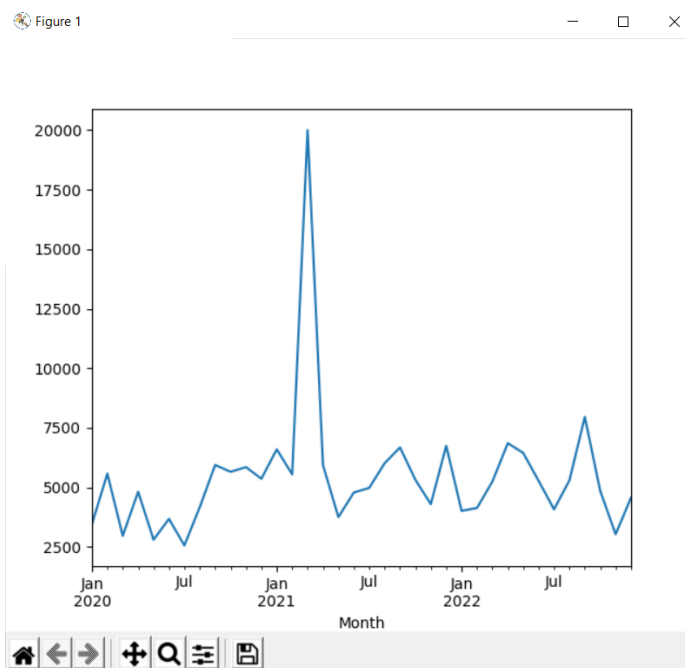


Рисунок 3.4 – Тестування роботи системи

З рис. 3.4. видно, що дані з набору відображаються коректно.

### 3.3 Поділ даних

На даному етапі розділено набір даних на дві частини: навчальний та тестовий набір.

Перші два роки даних взяті для набору навчальних даних, а дані, що залишилися, за один рік використані для тестового набору.

Моделі розроблялись з використанням навчального набору даних і виконують прогнози на тестовому наборі даних.

Останні 12 місяців спостережень такі (рис. 3.5).

```

2022-01, 4011.29
2022-02, 4132.3
2022-03, 5245
2022-04, 6853.45
2022-05, 6442.4
2022-06, 5272
2022-07, 4073.5
2022-08, 5291.58
2022-09, 7956
2022-10, 4846.55
2022-11, 3027.22
2022-12, 4571.11

```

Рисунок 3.5 – Спостереження за останні 12 місяців

### 3.4 Багатоступінчатий прогноз

Останніх 12 місяців набору даних перетворено у тримісячний прогноз.

Це з урахуванням історичних спостережень ( $t-1$ ,  $t-2$ ,  $t-n$ ) прогнозу  $t$ ,  $t+1$  і  $t+2$ .

Зокрема, з грудня у другому році прогнозовано січень, лютий та березень. З січня прогнозовано лютий, березень та квітень. Таким чином до жовтня, листопада, грудня прогноз із вересня на третій рік.

Усього потрібно 10 тримісячних прогнозів, а саме (рис. 3.6).

```

Dec,   Jan, Feb, Mar
Jan,   Feb, Mar, Apr
Feb,   Mar, Apr, May
Mar,   Apr, May, Jun
Apr,   May, Jun, Jul
May,   Jun, Jul, Aug
Jun,   Jul, Aug, Sep
Jul,   Aug, Sep, Oct
Aug,   Sep, Oct, Nov
Sep,   Oct, Nov, Dec

```

Рисунок 3.6 – Тримісячні прогнози

### 3.5 Оцінка моделі

Для оцінки моделі використаний сценарій ковзного прогнозу, також званий валідацією моделі.

Кожен процес кроку тестового набору даних проходить по одному. Модель використовувалась для складання прогнозу для тимчасового кроку, потім взято фактичне очікуване значення для наступного місяця з набору тестів та надано моделі для прогнозу на наступному тимчасовому кроці.

Це імітує реальний сценарій, в якому нові спостереження за продажами будуть доступні щомісяця та використовуватимуться у прогнозуванні наступного місяця.

Це змодельовано структурою поїзних та тестових наборів даних.

Зібрано всі прогнози в наборі тестових даних і розраховано оцінку помилки, щоб узагальнити навички моделі для кожного з тимчасових кроків прогнозу. Використовувалась середньоквадратична помилка (RMSE), оскільки вона слідує за великими помилками і призводить до отримання оцінки в тих самих одиницях, що й дані прогнозу, а саме щомісячні продажі.

### 3.6 Модель сталості

Хорошою базою для прогнозування часових рядів є модель сталості.

Це модель прогнозування, у якій останнє спостереження зберігається. Через свою простоту його часто називають наївним прогнозом.

### 3.7 Підготовка даних

Дані повинні бути підготовлені, перш ніж є змога їх використовувати для навчання LSTM.

Зокрема, необхідні дві додаткові зміни:

– **стаціонарна**, дані показують тенденцію до збільшення, яка має бути усунена шляхом відмінностей.

– **масштаб**, масштаб даних повинен бути зменшений до значень між -1 та 1, функцією активації блоків LSTM.

Запроваджено функцію, яка робить дані стаціонарними `difference()`, це перетворює ряд значень у ряд відмінностей, що спрощує уявлення (рис. 3.7).

```
# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - interval]
        diff.append(value)
    return Series(diff)
```

Рисунок 3.7 – Стаціонарна функція

Для масштабування даних використано `MinMaxScaler` з бібліотеки `Sklearn`.

З'єднавши це, було оновлено `prepare_data()` функцію, щоб спочатку відрізати дані та масштабувати їх, потім виконати перетворення на контрольоване завдання навчання та навчити тестові набори, як це було зроблено раніше зі сталістю.

Функція (рис. 3.8) повертає масштабуючий пристрій крім наборів даних `train` і `test`.

```
# transform series into train and test sets for supervised learning
def prepare_data(series, n_test, n_lag, n_seq):
    # extract raw values
    raw_values = series.values
    # transform data to be stationary
    diff_series = difference(raw_values, 1)
    diff_values = diff_series.values
    diff_values = diff_values.reshape(len(diff_values), 1)
    # rescale values to -1, 1
    scaler = MinMaxScaler(feature_range=(-1, 1))
    scaled_values = scaler.fit_transform(diff_values)
    scaled_values = scaled_values.reshape(len(scaled_values), 1)

    # transform into supervised learning problem X, y
    supervised = series_to_supervised(scaled_values, n_lag, n_seq)
    supervised_values = supervised.values
    # split into train and test sets
    train, test = supervised_values[0:-n_test], supervised_values[-n_test:]
    return scaler, train, test
```

Рисунок 3.8 – Функція для оновлення даних

### 3.8 Fit LSTM Network

Далі пристосовано модель LSTM до даних навчання.

Спочатку необхідно було, щоб навчальний набір даних був перетворений з двовимірного масиву [зразки, особливості] на 3D-масив [зразки, тимчасові кроки, особливості]. Встановлено тимчасові кроки в двовимірний масив так, щоб ця змінна була простою.

Наступним кроком було необхідно спроектувати мережу LSTM. Для цього використано просту структуру з одним прихованим шаром з одною одиницею LSTM, потім вихідний шар з лінійною активацією та трьома вихідними значеннями. Мережа використовує середньоквадратичну функцію втрати помилок та ефективний алгоритм оптимізації ADAM.

LSTM є станом; це означає, що стан мережі потрібно було скинути наприкінці кожної епохи навчання. Мережа вміщає 1500 епох.

Один і той самий розмір партії використовується для навчання та прогнозування, також зроблено налаштування, щоб прогнози робилися на кожному тимчасовому етапі набору тестових даних. Це означає, що розмір партії повинен дорівнювати одиниці. Розмір першого пакета також називається онлайн-навчанням, оскільки ваги мережі оновлюються під час навчання після кожного шаблону навчання (на відміну від міні-пакета або пакетного оновлення).

Зібрано все це разом у функцію під назвою `fit_lstm()`, Функція (рис. 3.9) приймає ряд ключових параметрів, які можна використовувати для наступного налаштування мережі, і функція повертає відповідну модель LSTM, готову для прогнозування.

```
# fit an LSTM network to training data
def fit_lstm(train, n_lag, n_seg, n_batch, nb_epoch, n_neurons):
    # reshape training into [samples, timesteps, features]
    X, y = train[:, 0:n_lag], train[:, n_lag:]
    X = X.reshape(X.shape[0], 1, X.shape[1])
    # design network
    model = Sequential()
    model.add(LSTM(n_neurons, batch_input_shape=(n_batch, X.shape[1], X.shape[2]), stateful=True))
    model.add(Dense(y.shape[1]))
    model.compile(loss='mean_squared_error', optimizer='adam')
    # fit network
    for i in range(nb_epoch):
        model.fit(X, y, epochs=1, batch_size=n_batch, verbose=0, shuffle=False)
        model.reset_states()
    return model
```

Рисунок 3.9 – Функція для налаштування мережі



### 3.9 Прогноз LSTM

Наступним кроком було використання відповідної мережі LSTM для складання прогнозів.

Єдиний прогноз можна зробити за допомогою відповідної мережі LSTM, викликавши `model.predict()`, Знову ж таки, дані повинні бути відформатовані в 3D-масив з форматом [зразки, тимчасові кроки, особливості].

Тому укладено це у функцію під назвою `forecast_lstm()` (рис. 3.10).

```
# make one forecast with an LSTM,
def forecast_lstm(model, X, n_batch):
    # reshape input pattern to [samples, timesteps, features]
    X = X.reshape(1, 1, len(X))
    # make forecast
    forecast = model.predict(X, batch_size=n_batch)
    # convert to array
    return [x for x in forecast[0, :]]
```

Рисунок 3.10 – Функція для прогнозування

Викликано цю функцію з `make_forecasts()` і оновлено її, щоб прийняти модель як аргумент. Оновлена версія наведена нижче на рис. 3.11.

```
# evaluate the persistence model
def make_forecasts(model, n_batch, train, test, n_lag, n_seq):
    forecasts = list()
    for i in range(len(test)):
        X, y = test[i, 0:n_lag], test[i, n_lag:]
        # make forecast
        forecast = forecast_lstm(model, X, n_batch)
        # store the forecast
        forecasts.append(forecast)
    return forecasts
```

Рисунок 3.11 – Оновлена версія функції

### 3.10 Інвертування перетворення

Після того, як були зроблені прогнози, потрібно інвертувати перетворення, щоб повернути значення у вихідну шкалу.

Це необхідно для того, щоб була можливість розрахувати оцінки помилок та графіки, які можна порівняти з іншими моделями, як прогноз стійкості, наведений вище.

Інвертовано шкалу прогнозів безпосередньо, використовуючи `MinMaxScaler` об'єкт, який пропонує функція `inverse_transform()` (рис. 3.13).

Далі інвертовано різницю, додавши значення останнього спостереження (продажу за попередні місяці) до першого прогнозованого значення, а потім розповсюджуючи значення за прогнозом.

Це трохи незручно; тому оберано поведінку в ім'я функції `inverse_difference()` вона приймає останнє значення перед прогнозом і прогноз як аргументи і повертає інвертований прогноз (рис. 3.12).

```
# invert differenced forecast
def inverse_difference(last_ob, forecast):
    # invert first forecast
    inverted = list()
    inverted.append(forecast[0] + last_ob)
    # propagate difference forecast using inverted first value
    for i in range(1, len(forecast)):
        inverted.append(forecast[i] + inverted[i-1])
    return inverted
```

Рисунок 3.12 – Функція інвертованого прогнозу

З'єднуючи це разом, створено функцію `inverse_transform()` (рис. 3.13), яка працює через кожен прогноз, спочатку інвертуючи шкалу, а потім інвертуючи відмінності, повертаючи прогнози в їхню початкову шкалу.

```
# inverse data transform on forecasts
def inverse_transform(series, forecasts, scaler, n_test):
    inverted = list()
    for i in range(len(forecasts)):
        # create array from forecast
        forecast = array(forecasts[i])
        forecast = forecast.reshape(1, len(forecast))
        # invert scaling
        inv_scale = scaler.inverse_transform(forecast)
        inv_scale = inv_scale[0, :]
        # invert differencing
        index = len(series) - n_test + i - 1
        last_ob = series.values[index]
        inv_diff = inverse_difference(last_ob, inv_scale)
        # store
        inverted.append(inv_diff)
    return inverted
```

Рисунок 3.13 – Функція, яка повертає прогнози в їх початкову шкалу

Наступним кроком спрощено розрахунок балів RMSE, очікуючи, отримано тестові дані, які містять лише вихідні значення, як показано нижче на рис. 3.14.

```
# invert differenced forecast
def inverse_difference(last_ob, forecast):
    # invert first forecast
    inverted = list()
    inverted.append(forecast[0] + last_ob)
    # propagate difference forecast using inverted first value
    for i in range(1, len(forecast)):
        inverted.append(forecast[i] + inverted[i-1])
    return inverted
```

Рисунок 3.14 – Функція спрощення розрахунку балів RMSE

### 3.11 Перевірка результатів

При виконанні програми спочатку RMSE друкується для кожного з прогнозованих тимчасових кроків (рис. 3.15).

```
t+1 RMSE: 1234.994887
t+2 RMSE: 1311.230417
t+3 RMSE: 1260.955787
```

Рисунок 3.15 – RMSE кожного з прогнозованих тимчасових кроків

Як видно з рисунку, оцінки на кожному прогнозованому тимчасовому кроці кращі, а в деяких випадках набагато кращі, ніж прогноз стійкості.

Це показує, що налаштований LSTM має навичку вирішення проблеми.

Відзначено, що RMSE не став прогресивно гіршим з довжиною горизонту прогнозу, як і було очікувано. Це відзначається тим фактом, що прогноз  $t + 2$  легше прогнозувати, ніж  $t + 1$ . Це може бути пов'язано з тим, що низхідний тик легше прогнозувати, ніж висхідний тик, зазначений у серії (це можна підтвердити за допомогою більш глибокого аналізу результатів).

Лінійний сюжет серії (синій) із прогнозами (червоний) також створюється. Він показаний на рис. 3.16.

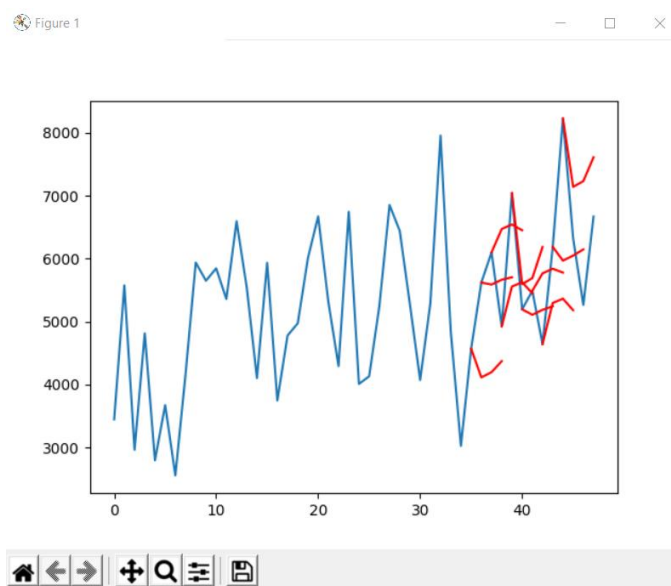


Рисунок 3.16 – Результат прогнозування

Для перевірки роботи системи змінено дані набору та перевірено результат. Він продемонстрований на рис. 3.17 та рис. 3.18.

```
t+1 RMSE: 1653.357529
t+2 RMSE: 1736.269610
t+3 RMSE: 1494.563511
```

Рисунок 3.17 – RMSE кожного з прогнозованих тимчасових кроків

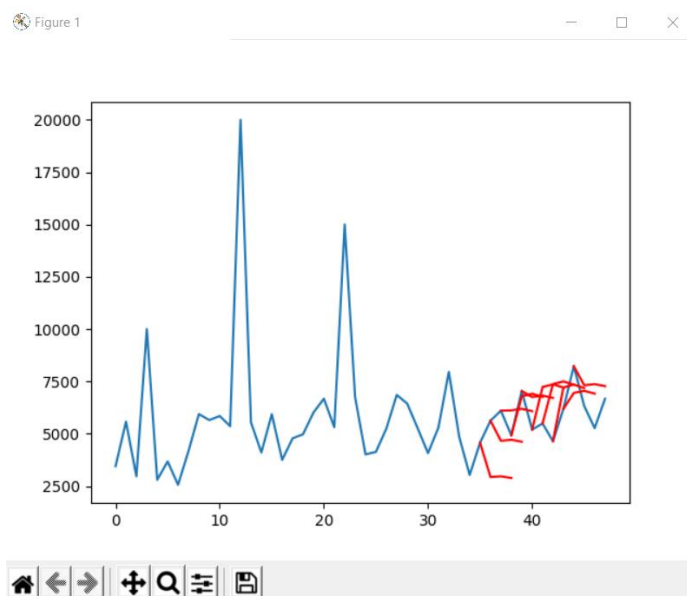


Рисунок 3.18 – Перевірка роботи системи

### 3.12 Тестування інтелектуальної системи

Для першого тестування системи взято набір медикаментів, але їх кількісну величину у одиницях. Для другого тесту взято у грошовому еквіваленті, а саме у доларі США (рис. 3.19).

"Month", "Sales"	"Month", "Sales"
2020-01, 55.46	2020-01, 86.20
2020-02, 89.69	2020-02, 139.41
2020-03, 47.68	2020-03, 74.10
2020-04, 77.44	2020-04, 120.37
2020-05, 45.01	2020-05, 69.95
2020-06, 59.10	2020-06, 91.86
2020-07, 41.15	2020-07, 63.96
2020-08, 67.01	2020-08, 104.15
2020-09, 95.52	2020-09, 148.47
2020-10, 90.89	2020-10, 141.26
2020-11, 94.04	2020-11, 146.16
2020-12, 86.22	2020-12, 134.01
2021-01, 87.45	2021-01, 164.88
2021-02, 73.51	2021-02, 138.58
2021-03, 54.42	2021-03, 102.59
2021-04, 78.71	2021-04, 148.38
2021-05, 49.72	2021-05, 93.73
2021-06, 63.38	2021-06, 119.50
2021-07, 65.96	2021-07, 124.36
2021-08, 79.71	2021-08, 150.28
2021-09, 88.49	2021-09, 166.84
2021-10, 70.47	2021-10, 132.86
2021-11, 56.94	2021-11, 107.35
2021-12, 89.44	2021-12, 168.63
2022-01, 43.29	2022-01, 100.28
2022-02, 44.60	2022-02, 103.31
2022-03, 56.60	2022-03, 131.13
2022-04, 73.96	2022-04, 171.34
2022-05, 69.53	2022-05, 161.06
2022-06, 56.89	2022-06, 131.80
2022-07, 43.96	2022-07, 101.84
2022-08, 57.11	2022-08, 132.29
2022-09, 85.86	2022-09, 198.90
2022-10, 52.30	2022-10, 121.16
2022-11, 32.67	2022-11, 75.68
2022-12, 49.33	2022-12, 114.28

Рисунок 3.19 – Набори даних

Візуальний перегляд даних про продажі представлено на рис. 3.20 та 3.21.

Кафедра інтелектуальних інформаційних систем  
Інтелектуальна система прогнозування продажу медикаментів

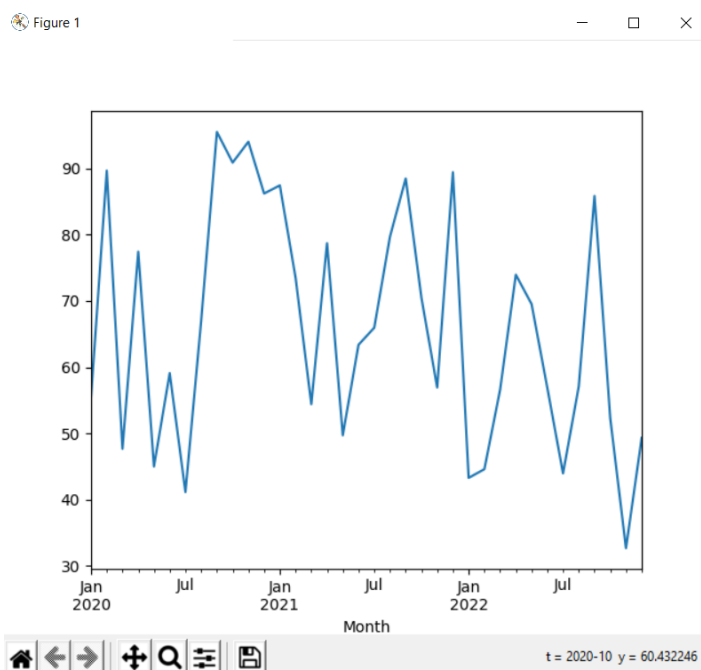


Рисунок 3.20 – Графічне представлення набору даних у кількісному еквіваленті

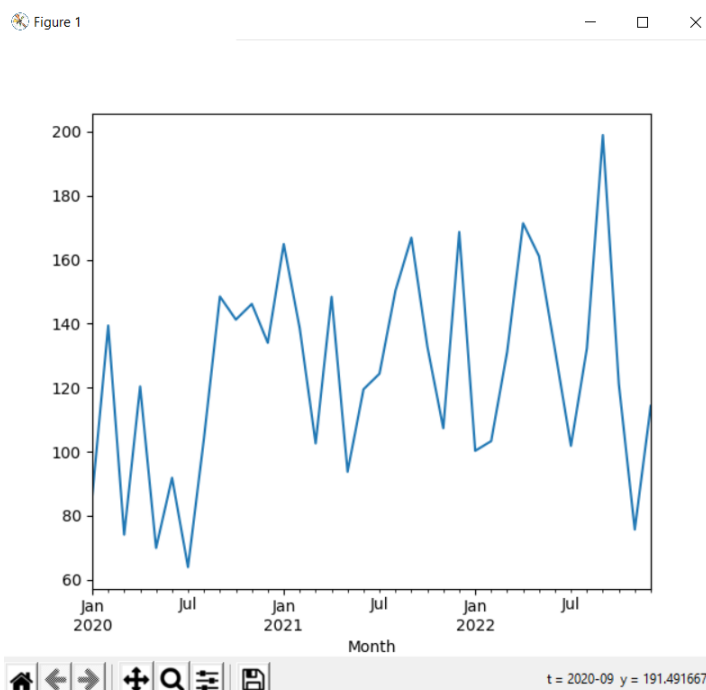


Рисунок 3.21 – Графічне представлення набору даних у грошовому еквіваленті  
(долар США)

На початковій стадії системи отримано RMSE для кожного з прогнозованих тимчасових кроків (рис. 3.22 та рис. 3.23).

```
t+1 RMSE: 21.488911
t+2 RMSE: 20.957899
t+3 RMSE: 19.222076
```

Рисунок 3.22 – RMSE кожного з прогнозованих тимчасових кроків у кількісному еквіваленті

```
t+1 RMSE: 31.820303
t+2 RMSE: 30.650592
t+3 RMSE: 31.708650
```

Рисунок 3.23 – RMSE кожного з прогнозованих тимчасових кроків у грошовому еквіваленті (долар США)

Як видно з рисунків, оцінки на кожному прогнозованому тимчасовому кроці кращі, а в деяких випадках набагато кращі, ніж прогноз стійкості.

Це показує, що налаштований LSTM працює коректно.

Створені лінійні сюжети серій (синій) із прогнозами (червоний). Вони показані на рис. 3.24 та рис. 3.25.

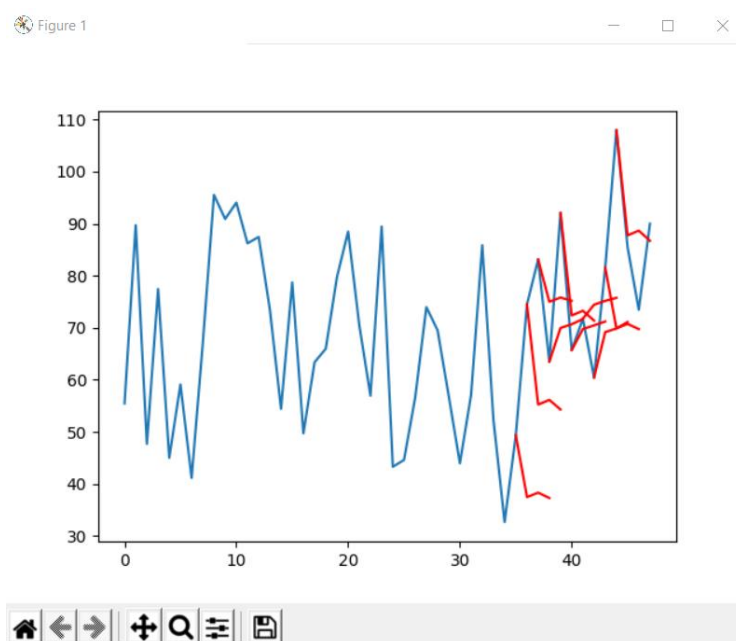


Рисунок 3.24 – Результат прогнозування у кількісному еквіваленті



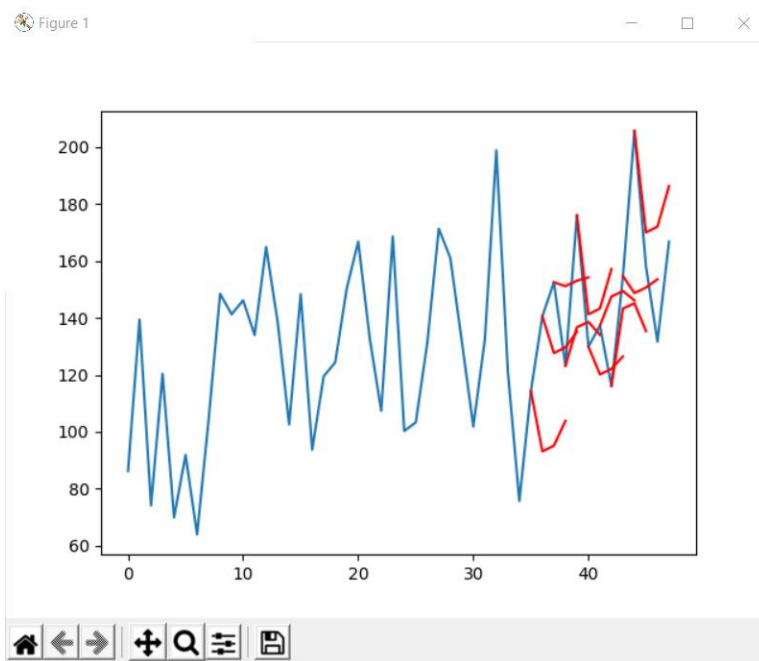


Рисунок 3.25 – Результат прогнозування у грошовому еквіваленті (долар США)

### Висновки до розділу 3

Під час виконання переддипломної практики було створено інтелектуальну систему, котра складається з 12 підрозділів.

У першому підрозділі було встановлено програмне забезпечення, бібліотеки, а також їх підключено.

Другий підрозділ передбачав собою опис набору, його стовпців, далі його підключення та перевірка роботи за допомогою змін у наборі даних.

У третьому підрозділі було виконано поділ даних на піднабори, а саме на навчальний та тестовий.

У четвертому підрозділі було складено багатоступіньчатий прогноз на основі тримісячних прогнозів серед останніх 12 місяців набору.

У п'ятому підрозділі проводилась оцінка моделі на основі сценарію ковзного прогнозу (валідації моделі).

У шостому підрозділі виконувалась підготовка даних, створено 2 нові змінні, стаціонарна та масштабна.

Сьомий підрозділ представляв собою пристосування моделі LSTM до даних навчання «Fit LSTM Network».

У восьмому підрозділі складались прогнози в створеній нейронній мережі LSTM.

У дев'ятому підрозділі відбувалось інвертування перетворення, це зроблено для того щоб була можливість розрахувати оцінки помилок та графіки, які можна порівняти з іншими моделями, таких як прогноз стійкості.

Десятий підрозділ являв собою перевірку результатів, а саме друкування значень RMSE помилки для кожного з тимчасових кроків, після чого друкувався графік з прогнозуванням.

У одинадцятому підрозділі виконувалась перевірка результатів, для цього було змінено дані у наборі, надруковано RMSE помилки та графік.

У дванадцятому підрозділі проводилось тестування інтелектуальної системи, зокрема зміна наборів даних та проведення прогнозувань на їх основі.

## ВИСНОВКИ

Сьогодні інформаційні технології дуже швидко розвиваються. Розвиток комп'ютерних технологій сприяв збільшенню продажів, що в свою чергу почало ускладнювати життя недостатністю якості прогнозування.

У даній роботі було досліджено основні проблеми процесу прогнозування продажу товарів.

У результаті проведеної роботи по розробці застосунку були вирішені наступні завдання:

- проаналізовано системи-аналоги;
- обрано та використано тип нейронної мережі для прогнозування продажів медикаментів;
- спроектовано інтелектуальну систему для прогнозування продажів медикаментів;
- проведено тестування на різних наборах даних для визначення якості роботи системи, точності її прогнозування та швидкості її роботи.

Система з допомогою людини, що використовує запропонований метод, значно зменшить обсяг ручної роботи, збільшить точність прогнозів та пришвидшить процес. LSTM технологія надає змогу прогнозувати на багато кроків вперед, що дає змогу спланувати свій продаж максимально ефективно.

Тому завершення перерахованих завдань свідчить про успішне досягнення поставленої мети даної роботи.

У методичній частині розроблялись лабораторні роботи за допомогою нейронних мереж в галузі знань 12 «Інформаційні технології». Частина має 2 лабораторні роботи, а саме «Прогнозування часових рядів за допомогою LSTM у TensorFlow» та «Класифікація зображень за допомогою нейронної мережі CNN».

У спеціальній частині з охорони праці було розглянуто питання охорони праці на робочому місці та цивільного захисту працівників ФОП Губарев М. О.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Підсумки загального продажу медикаментів: вебсайт. URL: <https://medzakupivli.com/uk/patsientam/perelik-tovariv-do-zakupivli/2022-rik> (дата звернення: 22.10.2022).
2. Продаж ліків та впливаючі проблеми: вебсайт. URL: <https://rau.ua/novyni/aptechnij-ritejl-pid-chas-vijni/> (дата звернення: 22.10.2022).
3. Aye, G.C., Balcilar, M., Gupta, R., Majumdar, A.: Forecasting aggregate retail sales. *IJPE* 160, 2015, pp. 66–79.
4. Hasin, M.A.A., Ghosh, S., Shareef, M.A.: An ANN approach to demand forecasting in retail trade in Bangladesh. *IJTEF* 2(2), 2011, pp. 154.
5. Lee, W.I., Chen, C.W., Chen, K.H., Chen, T.H., Liu, C.C.: A comparative study on the forecast of fresh food sales using logistic regression, moving average and BPNN methods. *JMST* 20(2), 2012, pp. 142–152.
6. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: *Time Series Analysis: Forecasting and Control*. Wiley, Hoboken, 2015.
7. Adamowski J, Fung Chan H, Prasher SO, Ozga-Zielinski B, Sliusarieva A Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in Montreal, Canada. *Water Resour Res* 48:25, 2012.
8. Breiman, L.: *Random forests*, 2011.
9. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with LSTM. *Neural Comput.* 12, 2000, pp. 2451–2471.
10. Бідюк П. І. Ймовірно-статистичні методи моделювання і прогнозування : [монографія] / П. І. Бідюк, О. П. Гожий. – Миколаїв : Чорноморський державний університет ім. Петра Могили, 2014, 440 с.
11. Ho, T. K. The random subspace method for constructing decision forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8), 2011, pp. 832–844.
12. Muzaffar S., Afshari A., Short-term load forecasts using LSTM networks. *Energy Procedia* 158:2922–2927, 2019.

13. Anirudh A., Mathematical modeling and the transmission dynamics in predicting the Covid-19 - what next in combating the pandemic. *Infect Dis Model* 5:366–374, 2020.
14. Alguliyev R.M., Aliguliyev R.M., Sukhostat L.V., Efficient algorithm for big data clustering on single machine. *CAAI Trans Intell Technol* 5(1):9–14, 2020.
15. Chen Z., Zhao B., Wang Y., Multitask learning and GCN-based taxi demand prediction for a traffic road network. *Sensors* 20:3776, 2020.
16. Wang L., Zou H., Su J., An ARIMA-ANN hybrid model for time series forecasting. *Syst Res Behav Sci* 30:244–259. <https://doi.org/10.1002/sres.2179>, 2013.
17. Arunraj, N.S., Ahrens, D., Fernandes, M.: Application of SARIMA model to forecast daily sales in food retail industry. *IJORIS* 7(2), 2016, pp. 1–21.
18. Lee, M.H., Hamzah, N.A.: Variation model based on ARIMA for forecasting sales data. In: *Proceedings of the RCSS, 2010*, pp. 349–361.
19. Zadeh N.K., Sepehri M.M., Farvareh H., Intelligent sales prediction for pharmaceutical distribution companies: a data mining based approach. *Math Probl Eng* 2014:1–15., 2014.
20. Alpaslan F., Eǧrioǧlu E., AladaǧÇH T.E., An statistical research on feed forward neural networks for forecasting time series. *Am J Intell Syst* 2:21–25, 2012.
21. Yu L., Wang S., Lai K. A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates. *Comput Oper Res* 32:2523–2541, 2005.
22. Hofmann E., Rutschmann E., Big data analytics and demand forecasting in supply chains: a conceptual analysis. *Int J Logist Manag* 29:739–766, 2018.
23. Chen Y., Shen L., Li R., Quantification of interfacial energies associated with membrane fouling in a membrane bioreactor by using BP and GRNN artificial neural networks. *J Colloid Interface Sci* 565:1–10, 2020.
24. Azar A.T., Bastan M., Habibifar N., Hamid M., Performance optimisation of a pharmaceutical production line by integrated simulation and data envelopment analysis. *Int J Simul Process Model* 14:360, 2019.

25. Carrasco R., Vargas M., Soto I., Fuertes G., Alfaro M., Copper metal price using chaotic time series forecasting. *IEEE Lat Am Trans* 13(6):1961–1965, 2015.
26. Muralitharan K., Sakthivel R., Vishnuvarthan R., Neural network based optimization approach for energy demand prediction in smart grid. *Neurocomputing* 273:199–208, 2018.
27. PereaRG PEC, Montesinos P., Díaz JAR., Optimisation of water demand forecasting by artificial intelligence with short data sets. *Biosys Eng* 177:59–66, 2019.
28. Candan G., Taskin M., Yazgan H.R., Demand forecasting in pharmaceutical industry using neuro-fuzzy approach. *J Mil Inf Sci* 2:41, 2014.
29. Bandara K., Shi P., Bergmeir C., Sales demand forecast in e-commerce using a long short-term memory neural network methodology. *Neural Inf Process Lect Note Comput Sci*, 2019.
30. Silva CFD, Almeida T., Barbosa RDM, New trends in drug delivery systems for veterinary applications. *Pharm Nanotechnol*, 2020.
31. Kumar M., Thenmozhi M., Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models. *Int J Bank Acc Financ* 5:284, 2014.
32. Takahashi Y., Aida H., Saito T., ARIMA model's superiority over f-ARIMA model. In: *WCC 2000 - ICCT 2000 international conference on communication technology proceedings (Cat No00EX420)*, 2000.
33. Прогнозування продажу з використанням програмного засобу MatLab: вебсайт. URL: <https://naukarus.com/prognozirovanie-vyruchki-ot-prodazh-na-osnove-sovremennyh-instrumentov-adaptivno-neuro-nechetkoj-seti-sistemy-matlab> (дата звернення: 06.11.2022).
34. Прогнозування продажу з використанням програмного засобу Deductor: вебсайт. URL: <https://basegroup.com/solutions/tasks/demand-forecast> (дата звернення: 07.11.2022).
35. Програмний засіб для вирішення комплексних процесів бізнес-процесів: вебсайт. URL: <https://basegroup.com/solutions/ready-solutions/demand-planning> (дата звернення: 07.11.2022).

36. GMDH Shell: вебсайт. URL: <https://gmdhsoftware.com/ru/neural-network-software/> (дата звернення: 09.11.2022).
37. STATISTICA Automated Neural Networks (SANN): вебсайт. URL: [http://statsoft.ru/products/STATISTICA\\_Neural\\_Networks/](http://statsoft.ru/products/STATISTICA_Neural_Networks/) (дата звернення: 10.11.2022).
38. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning (Adaptive Computation and Machine Learning series), 2016.
39. Francois Chollet, Deep Learning with Python, 2021.
40. Scikit-Learn: вебсайт. URL: <https://scikit-learn.org/stable/> (дата звернення: 22.10.2022).
41. Pandas: вебсайт. URL: <https://pandas.pydata.org/> (дата звернення: 22.10.2022).
42. NumPy: вебсайт. URL: <https://numpy.org/> (дата звернення: 22.10.2022).
43. Matplotlib: вебсайт. URL: <https://matplotlib.org/> (дата звернення: 22.10.2022).
44. Recurrent neural network (RNN): вебсайт. URL: <https://www.ibm.com/topics/recurrent-neural-networks> (дата звернення: 22.10.2022).
45. Long short-term memory (LSTM): вебсайт. URL: <https://databasecamp.de/en/ml/lstms> (дата звернення: 22.10.2022).
46. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин.
47. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу //Охорона праці. – 2001. –№ 12. – с. 12-20.
48. НПАОП 0.00-1.28-10. Правила охорони праці під час експлуатації електронно-обчислювальних машин.
49. Кодекс з цивільного захисту URL: <https://zakon.rada.gov.ua/laws/show/5403-17#Text> (дата звернення 28.01.2023)

## ДОДАТОК А

### Критерії бальної оцінки умов праці

Таблиця А.1

№ п/п	Фактор умов праці на робочому місці	Оцінка, бали					
		1	2	3	4	5	6
1	Температура повітря на робочому місці (РМ) у виробничому приміщенні, °С: - теплий період - холодний період	23...25 21...23	26...28 18...20	29...32 15...17	33...35 12...14	35...37 Нижче +12	>37 -
2	Відносна вологість повітря на РМ, %	40...50	55...60	61...75	76...85	Понад 85	-
3	Швидкість руху повітря на РМ, м/с	Менше 0,2	0,2...0,5	0,6...0,7	0,8...1,2	1,3...1,7	Понад 1,7
4	Освітленість на РМ, лк	≥ 300	240...300	160...230	100...150	60...90	30...50
5	Мінімальний розмір об'єкта розпізнавання, мм	> 1,0	1...0,3	< 0,3	0,005...0,3	< 0,05	-
6	Виробничий шум, перевищення ГДР, дБА	< 1	Рівно ГДР	1...5	6...10	> 10	> 10 з вібрацією
7	Інтенсивність теплового випромінювання, Вт/м <sup>2</sup>	≤ 140	141..1000	1001...1500	1501...2000	2001...2500	>2500
8	Токсична речовина, озон, кратність перевищення ГДК	-	≤ 1	1...2,5	2,6...4,0	4,1...6	> 6,0
9	Виробничий пил (паперовий), кратність перевищення ГДК	-	≤ 1	1...5	6...10	11...30	> 30
10	Робоче місце (РМ), поза переміщення у просторі	РМ стаціонарне, поза вільна, маса переміщене вантажу ≤ 5 кг	РМ стаціонарне, поза вільна, маса переміщене вантажу > 5 кг	РМ стаціонарне, поза вільна, до 25 % часу зміни у нахиленому положенні до 30°	РМ стаціонарне, поза вільна, до 50 % робочої зміни	РМ стаціонарне, поза вимушене, незручне – більше 50 % робочої зміни	РМ стаціонарне, поза вимушене, незручне, нахили під кутом до 60 град більше 300 разів за робочу зміну
11	Кількість важливих об'єктів	Менше 5	5...10	11...25	Понад 25	-	-



спостереження							
---------------	--	--	--	--	--	--	--

## Закінчення таблиці А.1

№ п\п	Фактор умов праці на робочому місці	Оцінка, бали					
		1	2	3	4	5	6
12	Тривалість зосередженого спостереження, % часу зміни	Менше 25	25...50	51...75	76...85	86...90	Понад 90
13	Тривалість повторюваних операцій, с	Понад 100	31...100	20...30	10...19	5...9	1...4
14	Змінність роботи	Ранкова зміна	Дві зміни	Три зміни	Нерегулярні зміни	-	-
15	Тривалість безперервної роботи за добу, годин	-	< 8	< 12	> 12	-	-
16	Режим праці та відпочинку	Обґрунтований, з включенням музики та гімнастики	Обґрунтований, без включення музики та гімнастики	Відсутність обґрунтованого режиму праці та відпочинку	-	-	-
17	Нервово-емоційне навантаження	Прості дії за індивідуальним планом	Прості дії за заданим планом з можливістю корегування	Складні дії за заданим планом з можливістю корегування	Складні дії за заданим планом при дефіциті часу	Відповідальність за безпеку людей	Індивідуальний ризик
18	Кількість рухів пальців на годину	< 360	360...720	721...1080	1081...3000	> 3000	-
19	Монотонність, тривалість операцій, які повторюються, с	> 100	31...100	20...30	10...19	5...9	1...4

## ДОДАТОК Б

### Залежність категорії умов праці від величини інтегральної бальної оцінки

Таблиця Б.1

Діапазон інтегральної бальної оцінки	Категорія умов праці	Характер роботи
До 18	I	Роботи, що виконуються в оптимальних умовах
19...33	II	Роботи, що виконуються в умовах, які відповідають гранично допустимим концентраціям (ГДК) і рівням (ГДР) санітарно-гігієнічних елементів, а також допустимим рівням психофізіологічних факторів
34...45	III	Роботи, що відхиляються від ГДК і ГДР та допустимих рівнів психофізіологічних факторів
45,7...53,9	VI	Робота у несприятливих умовах праці
54...59	V	Роботи, що виконуються в екстремальних умовах
Понад 59	VI	Роботи, що виконуються в екстремальних умовах