

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

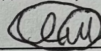
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

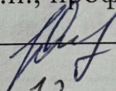
«Інформаційна технологія прогнозування
рентабельності, валового прибутку та валового
збитку підприємства»

Виконав: студент 2-го курсу, групи ЗКН-23м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)



Семенов О. М.
(прізвище та ініціали)

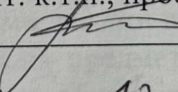
Керівник: к.т.н., професор каф. КН



Колесницький О. К.
(прізвище та ініціали)

« 05 » 12 2024 р.

Опонент: к.т.н., професор каф. КСУ



Биков М. М.
(прізвище та ініціали)

« 05 » 12 2024 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

« 06 » 12 2024 р.

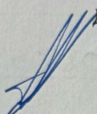
Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

д.т.н., проф. Яровий А. А.

 11. 10 2024 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Семенову Олександрю Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Інформаційна технологія прогнозування рентабельності, валового прибутку та валового збитку підприємства»

керівник роботи к.т.н., проф. каф. КН Колесницький О.К.

затверджені наказом вищого навчального закладу від «17» 09 2024 року № 310

2. Строк подання студентом роботи 11. 11 2024 року

3. Вихідні дані до роботи:

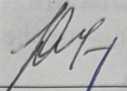
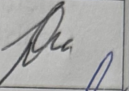
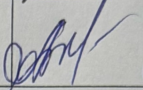
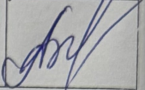
Мова програмування - об'єктно-орієнтована; похибка прогнозування – не вище 10%; можливість роботи з вхідними даними в форматі .xlsx.

4. Зміст текстової частини: вступ; аналіз предметної області прогнозування рентабельності, валового прибутку та валового збитку підприємства; проектування інформаційної технології прогнозування рентабельності, валового прибутку та валового збитку підприємства; програмна реалізація інформаційної технології прогнозування рентабельності, валового прибутку та валового збитку підприємства; тестування та аналіз результатів роботи програми; економічна частина, висновки, список використаних джерел, додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): Структурна схема інформаційної технології прогнозування; Структура нейронної мережі інформаційної технології прогнозування; Схема алгоритму роботи інформаційної технології прогнозування; UML-діаграма класів інформаційної технології прогнозування; Інтерфейс та тестування інформаційної технології.

6. Консультанти розділів роботи

Таблиця 1 - Консультанти роботи

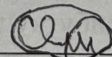
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Колесницький О.К., к.т.н., проф. каф. КН		
5	Адлер О.О., к.т.н., доц. каф. ЕПВМ		

7. Дата видачі завдання 02.09.2024

КАЛЕНДАРНИЙ ПЛАН

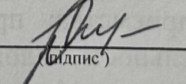
№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Прим
1	Аналіз предметної області прогнозування рентабельності підприємства	02.09.24 - 14.09.24	
2	Підготовка навчальної та тестової вибірки нейронної мережі	15.09.24 - 25.09.24	
3	Проектування інформаційної технології прогнозування рентабельності підприємства	26.09.24 - 02.10.24	
4	Програмна реалізація інформаційної технології прогнозування рентабельності підприємства	03.10.24 - 18.10.24	
5	Тестування та аналіз результатів роботи розробленої програми	19.10.24 - 28.10.24	
6	Підготовка економічної частини	29.10.24 - 04.11.24	
7	Оформлення матеріалів до захисту МКР	05.11.24 - 11.11.24	

Студент


(підпис)

Семенов О. М.
(прізвище та ініціали)

Керівник роботи


(підпис)

Колесницький О
(прізвище та ініціали)

АНОТАЦІЯ

УДК 004.8

Семенов О.М. Інформаційна технологія прогнозування рентабельності, валового прибутку та валового збитку підприємства. Магістерська кваліфікаційна робота зі спеціальності 122 – «Комп'ютерні науки», освітня програма – «Системи штучного інтелекту». Вінниця: ВНТУ, 2024. 114 с.

Укр. мовою. Бібліогр.: 26 назв; рис.: 18; табл. 8.

В даній магістерській кваліфікаційній роботі досліджено процес прогнозування рентабельності, валового прибутку та збитку підприємства. В роботі проведено порівняльний аналіз методів прогнозування та обґрунтовано використання нейронних мереж виду багатошаровий перцептрон. Інформаційну технологію було реалізовано мовою програмування C# в середовищі Visual Studio 2022. Інтерфейс створено за допомогою графічної підсистеми .Net Windows Presentation Foundation. Навчання мережі було проведено на основі даних сайту Агенства з розвитку інфраструктури фондового ринку України (АРІФРУ). Навчальна вибірка складалась з 1000 підприємств, тестова вибірка – зі 100 підприємств. Розроблена інформаційна технологія має середню похибку прогнозування рентабельності 3.64% та прогнозування валового прибутку/збитку 3.27%. Це перевищує точність прогнозування програм-аналогів GMDH Streamline та SAP IBP.

Графічна частина складається з 12 плакатів.

У економічному розділі визначено, що комерційний потенціал розробки вище середнього. Термін окупності становить 0,87 р., що менше 3-х років, що свідчить про комерційну привабливість розробки для потенційного інвестора.

Ключові слова: прогнозування, рентабельність, валовий прибуток, валовий збиток, нейронні мережі, багатошаровий перцептрон.

ABSTRACT

Semenov O.M. Information Technology for Forecasting Profitability, Gross Profit, and Gross Loss of an Enterprise. Master's Qualification Thesis in the specialty 122 – "Computer Science," educational program – "Artificial Intelligence Systems." Vinnytsia: VNTU, 2024. 114 pages. In Ukrainian. Bibliography: 26 titles; figures: 18; tables: 8.

This master's thesis explores the process of forecasting profitability, gross profit, and gross loss of an enterprise. The study includes a comparative analysis of forecasting methods and justifies the use of neural networks of the multilayer perceptron type. The information technology was implemented in C# programming language within the Visual Studio 2022 environment. The interface was developed using the .Net Windows Presentation Foundation (WPF) graphics subsystem. The network was trained using data from the website of the Ukrainian Securities and Stock Market Infrastructure Development Agency (SSMIDA). The training dataset consisted of 1000 enterprises, while the test set included 100 enterprises. The developed intelligent module demonstrates an average forecasting error for profitability of 3.64% and for gross profit/loss of 3.27%, surpassing the forecasting accuracy of the GMDH Streamline and SAP IBP software counterparts.

The graphic part consists of 12 posters.

The economic section establishes that the commercial potential of the development is above average. The payback period is 0.87 years, which is less than three years, indicating the commercial attractiveness of the development for potential investors.

Keywords: forecasting, profitability, gross profit, gross loss, neural networks, multilayer perceptron.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПРОГНОЗУВАННЯ РЕНТАБЕЛЬНОСТІ, ВАЛОВОГО ПРИБУТКУ ТА ВАЛОВОГО ЗБИТКУ ПІДПРИЄМСТВА	12
1.1 Постановка задачі прогнозування економічних показників підприємства.....	12
1.2 Огляд відомих методів прогнозування.....	15
1.3 Обґрунтування вибору аналогу до інформаційної технології прогнозування рентабельності підприємства	21
1.3 Висновок до розділу 1	25
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ РЕНТАБЕЛЬНОСТІ, ВАЛОВОГО ПРИБУТКУ ТА ВАЛОВОГО ЗБИТКУ ПІДПРИЄМСТВА	26
2.1 Обґрунтування вибору типу нейронної мережі.....	26
2.2 Розробка структури багатошарового перцептрону	35
2.3 Обґрунтування алгоритму навчання багатошарового перцептрону та підготовка навчальної вибірки.....	38
2.4 Розробка алгоритму роботи інформаційної технології прогнозування рентабельності підприємства	44
2.5 Розробка UML-діаграми класів	48
2.6 Висновок до розділу 2	50
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ РЕНТАБЕЛЬНОСТІ, ВАЛОВОГО ПРИБУТКУ ТА ВАЛОВОГО ЗБИТКУ ПІДПРИЄМСТВА.....	51
3.1 Обґрунтування вибору мови та середовища програмування.....	51
3.2 Реалізація програмних засобів прогнозування рентабельності, валового прибутку та валового збитку	56
3.3 Висновок до розділу 3	61

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ.....	62
4.1 Тестування роботи програми.....	62
4.2 Аналіз роботи інформаційної технології.....	67
4.3 Висновок до розділу 4	69
5 ЕКОНОМІЧНА ЧАСТИНА	70
5.1 Проведення комерційного та технологічного аудиту інформаційної технології прогнозування рентабельності, валового прибутку та валового збитку підприємства	70
5.2 Розрахунок витрат на здійснення науково-дослідної роботи	72
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	78
5.4 Висновок до розділу 5	82
ВИСНОВКИ.....	84
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	86
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	89
Додаток Б (обов'язковий) Лістинг програми.....	90
Додаток В (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА.....	101
Додаток Г (довідниковий) Інструкція користувача.....	110

ВСТУП

Актуальність теми. В сучасному бізнес-середовищі прогнозування рентабельності, прибутку та збитку підприємства є одними із найважливіших задач управління підприємством. Одним з ключових аспектів ефективного управління підприємством є здатність прогнозувати його економічні показники. В середовищі зростаючої конкуренції та нестабільності ринкових умов, здатність передбачити майбутні фінансові результати набуває великого значення для підприємства. Точний прогноз може допомогти підприємству визначити оптимальні стратегії та прийняти раціональні рішення, що сприятимуть його фінансовій стабільності та збільшенню прибутковості.

Традиційні методи прогнозування, такі як статистичні моделі та експертні оцінки, часто мають обмежену точність і не враховують всіх факторів, що впливають на рентабельність. У такому контексті, розвиток інтелектуальних модулів прогнозування, заснованих на штучному інтелекті та аналізі даних, є актуальним та перспективним напрямком досліджень.

Інформаційні технології прогнозування, що використовують штучний інтелект та аналіз даних, мають потенціал перевершити традиційні методи у прогнозуванні рентабельності підприємств. Вони можуть ефективно обробляти великі обсяги даних, враховувати складні взаємозв'язки та залежності між різними факторами, що впливають на рентабельність. Крім того, вони можуть автоматично навчатися на основі історичних даних та адаптуватися до змін у середовищі, що дозволяє забезпечувати більш точні прогнози.

Дослідження цієї області має велике значення для розвитку сучасного бізнесу. Інформаційна технологія прогнозування рентабельності дозволить підприємствам ефективніше приймати рішення щодо стратегічного планування, фінансового управління та ресурсного розподілу. Тому даний напрямок є актуальним для проведення подальших досліджень.

Зв'язок роботи з науковими програмами, планами, темами.

Магістерська кваліфікаційна робота виконана відповідно до напряму наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження.

Метою дослідження магістерської кваліфікаційної роботи є підвищення точності прогнозування рентабельності, валового прибутку та валового збитку підприємства за рахунок використання нейронних мереж.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз проблеми розв'язання задачі прогнозування рентабельності, валового прибутку та валового збитку підприємства;
- розглянути існуючі методи вирішення задачі прогнозування та обрати й обґрунтувати вибір методу, який задовольняє мету магістерської кваліфікаційної роботи;
- розробити структуру нейронної мережі згідно поставленої задачі;
- розробити структуру та алгоритм роботи інформаційної технології;
- розробити UML-діаграму класів програмного засобу;
- виконати програмну реалізацію запропонованої інформаційної технології прогнозування;
- провести тестування програмного продукту та виконати аналіз отриманих результатів.

Об'єкт дослідження – процес прогнозування економічних показників ефективності роботи підприємства.

Предмет дослідження – інформаційна технологія і програмні засоби для прогнозування економічних показників ефективності роботи підприємства та точність прогнозування.

Методи дослідження. У роботі були використані такі методи наукових досліджень: методи системного аналізу; теорія штучних нейронних мереж для прогнозування числових значень; теорія моделювання; методи об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів.

– набула подальшого розвитку інформаційна технологія розв'язання задач прогнозування рентабельності, валового прибутку та валового збитку підприємства, яка відрізняється від відомих використанням нейронної мережі виду багатошаровий перцептрон, що дозволяє підвищити точність прогнозування;

Практичне значення одержаних результатів полягає у наступному:

1. Розроблено алгоритм функціонування інформаційної технології прогнозування рентабельності, валового прибутку та збитку підприємства.
2. Адаптовано структуру нейронної мережі для підвищення точності прогнозування рентабельності, валового прибутку та збитку підприємства.
3. Удосконалено алгоритм навчання мережі для поставленої задачі за рахунок вибору оптимальної функції активації.
4. Розроблено UML-діаграму для проектування програмного засобу для інформаційної технології прогнозування рентабельності.
5. Розроблено програмний засіб для розв'язання задачі прогнозування рентабельності, валового прибутку та валового збитку підприємства на основі нейронної мережі багатошаровий перцептрон.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки задачі, коректним

застосуванням математичних методів під час доведення наукових положень, експериментальними дослідженнями тестування програмної реалізації інформаційної технології прогнозування рентабельності, валового прибутку та збитку підприємства.

Особистий внесок здобувача. Усі результати, що наведені у магістерській кваліфікаційній роботі, отримані самостійно. У роботах, опублікованих у співавторстві, автору належать такі результати: [1] – аналіз видів нейронних мереж для прогнозування рентабельності підприємства.

Апробація результатів. Результати дослідження були апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)», ВНТУ, Вінниця, 2024 [1].

Публікації. За результатами виконання бакалаврської дипломної роботи було опубліковано тези доповіді на науково-технічній конференції [1] та подано заявку на видачу авторського свідоцтва на твір-програму.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПРОГНОЗУВАННЯ РЕНТАБЕЛЬНОСТІ, ВАЛОВОГО ПРИБУТКУ ТА ВАЛОВОГО ЗБИТКУ ПІДПРИЄМСТВА

1.1 Постановка задачі прогнозування економічних показників підприємства

Прогнозування полягає у передбаченні майбутніх станів, значень або подій на основі наявних даних та залежностей між ними. Основною метою прогнозування є створення обґрунтованих передбачень щодо можливих майбутніх результатів або подій [2].

Воно широко використовується в багатьох галузях, таких як економіка, метеорологія, логістика, сільське господарство, тощо. Наприклад, прогнозування може бути корисним у розробці стратегій управління запасами для підтримання оптимального рівня запасів товарів у магазинах або на складах. А у фінансовій сфері фінансовій сфері прогнозування допомагає інвесторам приймати рішення щодо інвестицій на основі прогнозів щодо ринкових тенденцій та ризиків [2].

У даному дослідженні потрібно розв'язати задачу прогнозування рентабельності, валового прибутку та валового збитку підприємства, що передбачає визначення майбутніх значень цих величин на основі наявних економічних показників.

Рентабельність – це показник, що обчислюється шляхом поділу прибутку компанії на її витрати і виражається у відсотках. Цей показник має велике значення для аналізу економічної діяльності підприємства, оскільки він вказує на те, яку частину прибутку отримано з кожної витраченої одиниці. Порівняння рентабельності дозволяє визначити, яке з підприємств ефективніше використовує свої ресурси [3].

Рентабельність, як показник, що відображає ефективність діяльності підприємства, є результатом взаємозв'язку між його прибутком і витратами. Цей показник непрямо залежить від різноманітних факторів, що впливають на обсяги прибутку та витрат. Серед таких факторів можна виокремити вартість основних фондів, середньорічну заробітну плату працівників, кількість працівників та рівень кредитної та дебіторської заборгованості [3].

Основні фонди – це засоби виробництва, що використовуються протягом тривалого періоду виробництва, зберігаючи свою фізичну форму та властивості. Вони включають у себе матеріальні активи, такі як будівлі, машини, обладнання, транспортні засоби та інші засоби виробництва, які використовуються для створення готового продукту або надання послуг [3].

Вартість основних фондів знаходять за такою формулою [3]:

$$B = B_{\text{пр}} + B_{\text{т}} + B_{\text{м}} + B_{\text{с}} + B_{\text{ін}}, \quad (1.1)$$

де $B_{\text{пр}}$ – це ціна придбання основних фондів;

$B_{\text{т}}$ – це витрати на їх транспортування;

$B_{\text{м}}$ – витрати на монтаж, налагодження основних фондів;

$B_{\text{с}}$ – це витрати на страхування ризиків доставки;

$B_{\text{ін}}$ – це інші витрати, пов'язані з доведенням основних фондів до стану, в якому вони придатні до використання.

Середньорічна заробітна плата – це середня заробітна плата, яку працівники отримують за рік своєї роботи.

Вона знаходиться за наступною формулою [3]:

$$C_{\text{зп}} = \frac{B_{\text{на}} * B_{\text{оп}}}{K}, \quad (1.2)$$

де $B_{\text{на}}$ – це вартість нематеріальних активів;

$V_{\text{на}}$ – це витрати на оплату праці;

K – це кількість працівників на підприємстві.

Кредитна заборгованість – це сума грошових коштів або фінансових зобов'язань, які підприємство повинно повернути своїм кредиторам. Це включає позики, кредити, облігації або будь-які інші зобов'язання, які підприємство отримало з метою фінансування своєї діяльності або придбання активів [3].

Кредитна заборгованість є важливим аспектом фінансового аналізу підприємства, оскільки вона впливає на його фінансову стійкість, платоспроможність та здатність здійснювати свою діяльність. Аналіз кредитної заборгованості допомагає оцінити ризики пов'язані з платежами, процентними платежами та можливістю погашення зобов'язань підприємства у встановлені строки.

Дебіторська заборгованість підприємства відображає суму грошових коштів або фінансових зобов'язань, які інші особи (клієнти або покупці) мають перед підприємством за надані товари або послуги. Це означає, що підприємство очікує отримання платежів від своїх клієнтів за поставлені товари чи послуги в майбутньому [3].

Тобто, основна різниця між дебіторською заборгованістю та кредитною заборгованістю полягає в тому, що дебіторська заборгованість виникає внаслідок зобов'язань клієнтів перед підприємством за надані товари або послуги, тоді як кредитна заборгованість виникає внаслідок зобов'язань підприємства перед його кредиторами або постачальниками.

Валовий прибуток — це різниця між виручкою від реалізації продукції (товарів, робіт, послуг) та собівартістю цієї продукції. Він показує, скільки грошей залишається після покриття основних витрат на виробництво або придбання товарів, але до вирахування операційних витрат, податків та інших витрат [3].

Валовий збиток виникає, коли собівартість реалізованої продукції перевищує виручку від її продажу. Його значення є різницею між собівартістю продукції та виручкою. Це означає, що компанія витрачає більше на виробництво або придбання товарів, ніж отримує від їх продажу [3].

Якщо враховувати ці показники під час прогнозування, то можна врахувати приховані залежності, які не відображені на попередніх значеннях прибутку і витрат. Наприклад, якщо підприємство виплатило свою заборгованість, то в подальшому кількість витрат зменшиться.

Тому використання вище перелічених величин в якості вхідних даних додатково до значень рентабельності, прибутку і витрат дозволить підвищити точність прогнозування рентабельності, валового прибутку та валового збитку підприємств.

1.2 Огляд відомих методів прогнозування

У сфері прогнозування відомі два основні підходи: кількісний та якісний, кожен з яких має свої унікальні характеристики та можливості для використання у різноманітних завданнях.

Кількісний підхід ґрунтується на використанні математичних моделей та аналізі історичних даних. Цей підхід використовує статистичні методи та алгоритми для визначення майбутніх тенденцій та патернів на основі кількісних даних [4].

Якісний підхід, натомість, ґрунтується на експертній думці, освіченій інтуїції та досвіду фахівців у відповідній галузі. Цей підхід використовується у випадках, коли недостатньо кількісних даних або коли майбутні події не можуть бути точно передбачені на основі статистики. Він дозволяє залучити експертне знання та суб'єктивні оцінки для формування прогнозів [4].

Залежно від ступеня формалізації, методи прогнозування можна поділити на дві категорії: формальні та неформальні. Формальні методи

ґрунтуються на чітких правилах та алгоритмах, використовуючи точні математичні моделі та статистичні методи для прогнозування. Ці методи відрізняються високим рівнем об'єктивності та повторюваності результатів, оскільки вони базуються на конкретних правилах і процедурах, які можуть бути легко відтворені та перевірені [4, 5].

Неформальні методи прогнозування, натомість, оперують нечіткими правилами та евристиками. Вони базуються на експертному знанні, особистому досвіді та інтуїції фахівців у конкретній галузі. Ці методи використовуються у випадках, коли формалізація є складною або неможливою через недостатність даних або складність моделювання проблеми. Неформальні методи можуть бути корисними у ситуаціях, де важливі суб'єктивні аспекти, і вони надають можливість враховувати експертну думку та контекстуальні фактори, які не завжди можуть бути виражені у формальних правилах [4, 5].

Серед інтелектуальних методів прогнозування можна виділити наступні [5]:

- 1) часові ряди;
- 2) програмні агенти;
- 3) генетичні алгоритми;
- 4) нейронні мережі.

Часові ряди є потужним інструментом у сфері аналізу даних і прогнозування, що набули широкого визнання завдяки своїй здатності враховувати часову залежність даних. Їх застосування варіюється від економічного аналізу та фінансового прогнозування до метрології і технічного обслуговування [6].

Часовий ряд можна визначити як послідовність даних, зібраних або записаних у часовій послідовності. Основною характеристикою часових рядів є те, що дані мають природний порядок, визначений часом. Аналіз часових рядів передбачає вивчення даних з метою виявлення закономірностей і трендів, які можуть бути використані для майбутнього прогнозування [6].

Перевагами часових рядів є [6, 7]:

- 1) простота моделювання;
- 2) автокореляція (враховується взаємозв'язок між значеннями в різні моменти часу);
- 3) висока гнучкість моделей;
- 4) стійкість до короткотривалих змін.

Тоді як недоліками є [6, 7]:

- 1) для точного прогнозування потрібні довгі часові ряди;
- 2) висока чутливість до змін в даних (раптові зміни можуть значно впливати на точність прогнозів);
- 3) методи часових рядів часто базуються на лінійних припущеннях, що може бути недостатнім для моделювання складних нелінійних взаємозв'язків;
- 4) обмеженість у врахуванні великої кількості факторів.

Порівняно з іншими методами прогнозування часові ряди можуть мати перевагу у випадках, коли [7]:

- дані мають явну часову структуру;
- прогнозування потребує простоти та швидкості виконання;

Однак, часові ряди менше підходять в таких випадках [7]:

- дані мають складні, нелінійні залежності;
- прогнозування потребує врахування великої кількості факторів.

Програмні агенти представляють собою комп'ютерні програми, які працюють у самостійному та інтерактивному режимі в певному середовищі. Вони можуть виконувати дії, приймати рішення та взаємодіяти з іншими агентами та компонентами системи. Програмні агенти мають здатність використовувати різноманітні методи інтелектуального аналізу даних, навчання, планування та інші, залежно від конкретних вимог та завдань [8].

Застосування програмних агентів може бути корисним для прогнозування динамічних та розподілених процесів у визначеному середовищі. Вони

дозволяють аналізувати та прогнозувати зміни, що відбуваються в системі протягом часу та в розподілених мережах. Це може включати прогнозування майбутніх станів системи, визначення оптимальних стратегій дій, розробку планів та сценаріїв розвитку, а також виявлення складних зв'язків та закономірностей у великому обсязі даних [8].

Серед переваг програмних агентів можна виділити [8, 9]:

- 1) автономність та інтерактивність дії;
- 2) здатність до реагування на зміни у середовищі.

Недоліки програмних агентів [8, 9]:

- 1) складність моделювання та верифікації поведінки агентів;
- 2) необхідність стандартизації протоколів комунікації між агентами

Порівняно з іншими методами прогнозування програмні агенти можуть мати перевагу у випадках, коли [8]:

- дані є розподілені по різних джерелах та потребують локальної обробки;
- прогнозування потребує врахування контексту та ситуативності;
- прогнозування потребує групового рішення з участю різних зацікавлених сторін.

Однак, програмні агенти менше підходять в таких випадках [8]:

- дані однорідні і централізовані;
- прогнозування потребує високої точності та інтерпретованості.

Генетичні алгоритми використовують концепцію природного відбору для розв'язання оптимізаційних задач. Вони моделюють еволюційні процеси, використовуючи механізми відбору, схрещування та мутації для пошуку оптимальних рішень [10].

Ці алгоритми можуть застосовуватися для вирішення складних та багатокритеріальних задач, де потрібно знайти набір рішень, що відповідає різноманітним обмеженням та вимогам. Вони дозволяють шукати ефективні

рішення у просторі можливих варіантів та знаходити компроміси між конфліктуючими цілями [10].

Генетичні алгоритми можуть застосовуватися для прогнозування найкращих стратегій та сценаріїв в різних областях, таких як економіка, фінанси, логістика та інші. Вони дозволяють виявляти оптимальні рішення, адаптуватися до змінних умов та прогнозувати наслідки різних варіантів дій [11].

Переваги генетичних алгоритмів [10]:

- 1) здатність до глобального пошуку оптимальних розв'язків у складних та багатокритеріальних задачах;
- 2) здатність до паралельного обчислення та адаптації до зміни умов задач.

Недоліки генетичних алгоритмів [10]:

- 1) висока обчислювальна складність та часовитрата алгоритму;
- 2) необхідність попереднього визначення параметрів алгоритму;
- 3) можливість застрягання у локальних оптимумах.

Випадки, коли варто використовувати генетичний алгоритм [11]:

- потрібно працювати з великими обсягами даних або високовимірними просторами;
- треба знайти оптимальне або наближене розв'язання задач оптимізації або пошуку.

Нейронні мережі є математичними моделями, які намагаються емулювати роботу нейронів людського мозку. Ці моделі можуть навчатися на основі доступних даних та пристосовуватися до змін у навколишньому середовищі. Вони відрізняються своєю здатністю виявляти складні закономірності та нелінійні зв'язки у вхідних даних, що забезпечується їхньою структурою, що складається зі штучних нейронів, які взаємодіють один з одним через зв'язки з вагами. Під час навчання нейронні мережі коригують ці ваги, оптимізуючи їх для досягнення певного критерію, такого як точність прогнозування [13].

Нейронні мережі знаходять широке застосування у багатьох галузях, таких як обробка сигналів, комп'ютерне зоро, мовленнєвий аналіз, розпізнавання образів, біоінформатика та фінансовий аналіз, оскільки вони можуть працювати з різними типами даних, такими як текст, зображення, аудіо та числові дані [13].

З урахуванням їх гнучкості та здатності до самоорганізації, нейронні мережі відкривають нові можливості для розв'язання складних задач прогнозування та моделювання. Хоча їхній потенціал ще не повністю досліджений, дослідники продовжують працювати над вдосконаленням їхніх алгоритмів та структур для досягнення ще більшої точності та ефективності прогнозування [13].

Переваги використання нейронних мереж [13]:

- 1) висока точність прогнозування;
- 2) здатність до самонавчання та адаптації;
- 3) універсальність застосування;
- 4) швидко обробляють нові дані після навчання;
- 5) можуть моделювати складні нелінійні залежності та знаходити шаблони у даних

Недоліки використання нейронних мереж [13]:

- 1) низька інтерпретативність;
- 2) потребують багато часу і ресурсів для навчання;
- 3) чутливі до вибору гіперпараметрів.

Для прогнозування в цій роботі було вибрано метод нейронних мереж, оскільки в нього висока точність та вони можуть швидко обробляти дані після навчання.

Часові ряди краще підходять у випадках, коли невелика кількість факторів та дані мають лінійні зв'язки. В даному випадку, коли враховується багато факторів між якими можуть бути нелінійні залежності, краще підходять нейронні мережі завдяки їх здатності моделювати складні, нелінійні зв'язки.

Програмні агенти не є ефективними для поставленої задачі через низьку точність і однорідність та централізованість даних в цій задачі. У той же час генетичні алгоритми мають нижчу точність і оптимальніші для більших обсягів даних.

1.3 Обґрунтування вибору аналогу до інформаційної технології прогнозування рентабельності підприємства

На сьогоднішній день існує обмежена кількість програм із функцією прогнозування рентабельності, валового прибутку та збитку підприємств. Однією з найбільш популярних серед них є Streamline [14].

Streamline є провідною платформою для спільного бізнес-планування, яка використовує штучний інтелект та динамічне моделювання для оптимізації ланцюгів постачання. Ця програма дозволяє підприємствам прогнозувати фінансові показники, планувати закупівлі та управляти запасами з високою точністю [14].

Основні функції GMDH Streamline включають [14]:

1. Прогнозування фінансових величин: Автоматичне прогнозування показників на основі історичних даних.
2. Оптимізація запасів: Забезпечення оптимального рівня запасів для уникнення надлишкових запасів та дефіциту товарів.
3. Планування поповнення: Планування поповнення запасів для системи складів, враховуючи функції розподільчих центрів та прямих поставок.
4. Інтеграція з ERP-системами: Двостороння інтеграція з будь-якими джерелами даних та ERP-системами, такими як SAP, Oracle, Microsoft Dynamics, QuickBooks та іншими.

Інтерфейс програми Streamline зображено на рисунку 1.1.

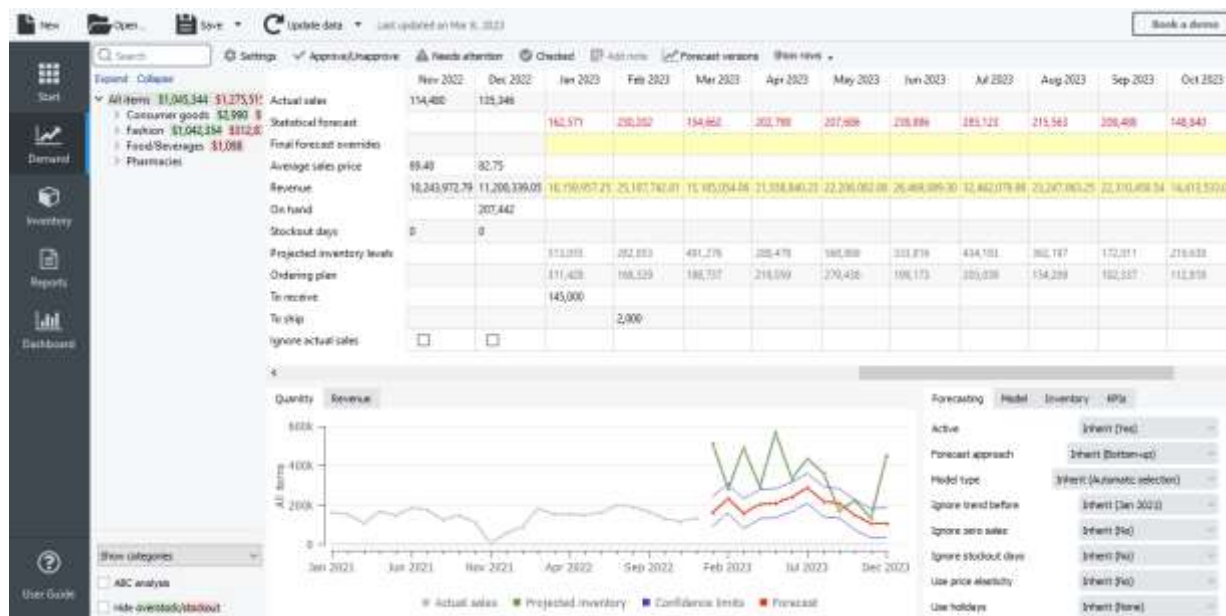


Рисунок 1.1 – Інтерфейс програми Streamline

Переваги програми Streamline [14]:

- 1) наявні функції управління запасами, такі як оптимізація та планування поповнення запасів;
- 2) автоматичне статистичне прогнозування;
- 3) наявний щомісячний і щотижневий прогноз;
- 4) інтеграція з ERP системами;
- 5) наявна безкоштовна версія.

Недоліки програми Streamline [14]:

- 1) складний для розуміння інтерфейс;
- 2) функціонал заплутаний для початківців;
- 3) для точного прогнозування потрібні довгі часові ряди;
- 4) не враховує факторів впливу на фінансові показники.

SAP Integrated Business Planning (SAP IBP) є комплексним хмарним рішенням, розробленим для управління стратегічними, тактичними та операційними цілями компанії в інтегрованому середовищі. Ця платформа, заснована на технології SAP HANA, об'єднує різні аспекти планування ланцюгів постачання, включаючи планування продажів та операцій (S&OP),

прогнозування та управління попитом, управління запасами та планування постачання [15].

Основні функції SAP IBP включають [15]:

1. Прогнозування економічних показників та управління попитом: Використання алгоритмів штучного інтелекту для підвищення точності прогнозів, статистичного моделювання, аналізу часових рядів та автоматичної корекції аномалій у даних продажів.
2. Планування постачання: Моделювання постачання на різних рівнях, включаючи багаторівневі специфікації матеріалів, що дозволяє створювати ефективні плани постачання для всієї мережі.
3. Планування продажів та операцій: Інтеграція фінансового та операційного планування в єдиний процес S&OP, що дозволяє проводити симуляції змін попиту та постачання для підготовки до можливих збоїв у ланцюзі постачання.
4. Управління запасами: Оптимізація рівнів запасів для зменшення надлишкових запасів та підвищення рівня обслуговування клієнтів.

Інтерфейс програми SAP IBP зображено на рисунку 1.2.

Переваги SAP IBP [13, 15]:

- 1) широкий спектр функцій;
- 2) можливість прогнозування різноманітних фінансових показників;
- 3) порівняно простий для розуміння інтерфейс;
- 4) багато способів візуалізації мережі постачання та інших економічних даних;
- 5) поповнення запасів залежно від попиту зменшує вплив похибки прогнозу на оптимальність стратегії відновлення ресурсів;

Недоліки програми SAP IBP [14]:

- 1) програма платна, хоча має тимчасову демо версію;
- 2) застарілий інтерфейс;
- 3) потребує спеціалізованих знань для ефективного використання;

- 4) невисока точність прогнозування деяких величин;
- 5) під час прогнозування не враховує факторів крім минулих значень величини.

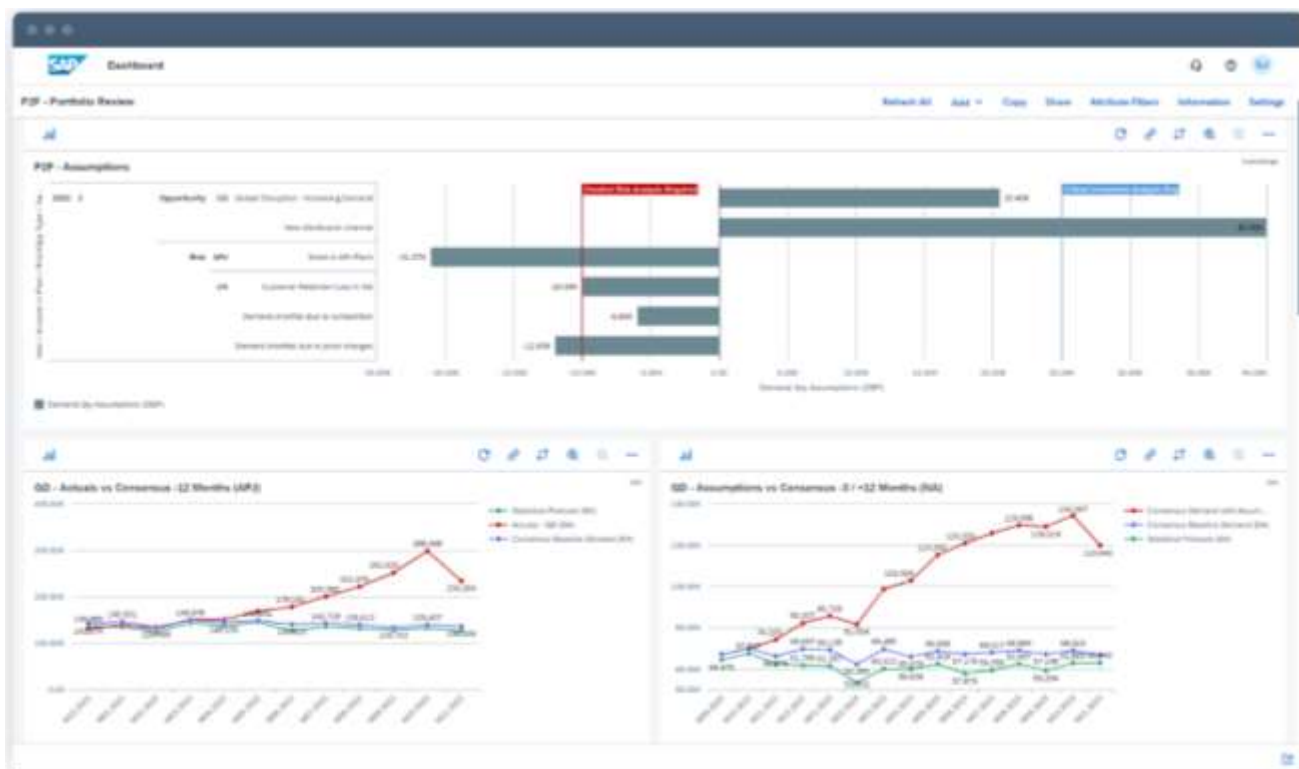


Рисунок 1.2 – Інтерфейс програми SAP IBP

Розглянуті програми мають велику кількість функцій, включаючи прогнозування різних фінансових величин. Проте вони має невисоку точність прогнозування показників, що залежать від високої кількості факторів. Це спричинено тим, що програма не враховує фактори, які впливають на зміну значення величини.

Тому необхідно розробити нові методи і програмні засоби, які зможуть усунути цей недолік та забезпечити точніше прогнозування рентабельності та валового прибутку/збитку. В даному дослідженні буде досягнуто кращих показників точності прогнозування за допомогою нейронної мережі, що враховує фактори впливу на ці показники. Крім того, розроблювана програма буде мати простий і зрозумілий інтерфейс, що дозволить будь-якому

користувачеві з доступом до вхідних даних використовувати її без особливих знань у галузі економіки.

1.3 Висновок до розділу 1

В даному розділі було проведено постановку задачі прогнозування рентабельності, валового прибутку та збитку підприємства. Розглянуто класифікацію методів прогнозування та здійснено порівняльний аналіз інтелектуальних методів розв'язання цієї задачі, таких як часові ряди, програмні агенти, генетичні алгоритми та метод штучних нейронних мереж. На основі проведеного аналізу для реалізації поставленої задачі було обрано нейромережевий метод як найбільш ефективний.

Крім того була проведена оцінка програмних аналогів, які використовуються для прогнозування рентабельності, валового прибутку та збитку. Такими аналогами є GMDH Streamline та SAP Integrated Business Planning. Головними недоліками було визначено низьку точність прогнозування величин, що залежать від великої кількості факторів, та складний для розуміння інтерфейс.

У зв'язку з цим, доцільно розробити нову програму для прогнозування рентабельності з вищою точністю прогнозування та більш інтуїтивним інтерфейсом.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ РЕНТАБЕЛЬНОСТІ, ВАЛОВОГО ПРИБУТКУ ТА ВАЛОВОГО ЗБИТКУ ПІДПРИЄМСТВА

2.1 Обґрунтування вибору типу нейронної мережі

Для вибору оптимального виду нейронної мережі необхідно провести порівняльний аналіз різних типів нейронних мереж. На сучасному етапі розвитку є велике розмаїття видів нейронних мереж, які відрізняються за архітектурою та використанням. Кожен з них має свій набір особливостей, переваг та недоліків, основні з яких будуть розглянуті в цьому підрозділі.

Згорткові нейронні мережі представляють собою тип мереж, спеціалізованих у виявленні та аналізі зображень та відео. Ці мережі використовують операцію згортки, яка дозволяє виокремлювати ключові характеристики зображень, такі як кольори, контури та текстур. Архітектура згорткових мереж включає кілька шарів, кожен з яких виконує різні функції, зокрема згортковий шар, агрегаційний шар, повнозв'язний шар та шар втрат. Після навчання згорткові мережі здатні розпізнавати об'єкти, обличчя, сцени, емоції та інші візуальні елементи [16].

Підґрунтям для розробки згорткових мереж послужив біологічний процес в зоровій корі тварин, де окремі нейрони реагують на стимули в обмеженій області зорового поля. Згорткові мережі використовують цю концепцію, шляхом локального з'єднання та спільного використання параметрів між штучними нейронами, щоб імітувати процес виявлення ознак візуальної стимуляції [15, 17].

Згортковий шар є одним з ключових компонентів згорткових нейронних мереж, використовуваних для обробки зображень і відео. Цей шар відіграє важливу роль у виділенні та виявленні основних ознак зображення шляхом застосування операції згортки [17].

Згортковий шар складається з набору фільтрів або ядер, які рухаються по вхідному зображенню. Кожен фільтр виконує локальну операцію згортки, в результаті чого створюється карта ознак, яка відображає виявлені характеристики в конкретних областях зображення. Ці ознаки можуть включати різні аспекти, такі як ребра, кути, текстури тощо [17].

Операція згортки полягає в обчисленні скалярного добутку між вагами фільтра та вхідними даними. Цей процес сприяє виявленню локальних шаблонів та ознак у зображенні. Після застосування кожного фільтра отримується окрема карта ознак, що представляє собою двовимірний шар числових значень [17].

Агрегувальний шар є компонентом згорткової нейронної мережі, призначеним для зменшення розміру вхідних даних. Цей шар отримує на вхід матрицю даних, таку як зображення, і застосовує до неї специфічну операцію агрегації, таку як максимум, середнє або суму, в кожному підвікні фіксованого розміру. В результаті отримується менша матриця даних, що містить тільки найважливіші ознаки з вихідної матриці [17].

Агрегувальний шар виконує кілька функцій для згорткової нейронної мережі [17]:

1) Він дозволяє зменшити обсяг передаваних даних до наступних шарів, тим самим скорочуючи час обчислень і кількість параметрів. Це сприяє підвищенню ефективності мережі та зменшенню її складності.

2) Агрегувальний шар забезпечує інваріантність до незначних зміщень або деформацій вхідних даних. Це означає, що він нечутливий до незначних варіацій у положенні або формі об'єктів на зображенні. Така інваріантність покращує роботу мережі при обробці зображень з різних ракурсів або в умовах невеликих змін.

3) Цей шар допомагає уникнути перенавчання, оскільки втрачаються менш суттєві деталі вхідних даних. Це дозволяє зберегти загальні ознаки та глобальну структуру об'єктів, сприяючи загальній узагальненості мережі і

знижуючи ризик перенавчання на конкретні особливості навчального набору даних.

Повнозв'язний шар є складовою частиною згорткової нейронної мережі, яка забезпечує зв'язок між кожним нейроном одного шару та кожним нейроном наступного шару. Він виконує функцію високорівневого розуміння, що означає аналіз та інтерпретацію ознак, виявлених попередніми шарами. Повнозв'язний шар схожий на традиційний багатошаровий перцептрон, але відрізняється тим, що він приймає матрицю даних, а не вектор [17].

Шар втрат є складовою частиною згорткової нейронної мережі, яка вимірює різницю між очікуваним та фактичним виходом мережі. Функція шару втрат полягає в оцінці якості навчання мережі, тобто визначенні, наскільки добре мережа виконує поставлену задачу. Цей шар знаходиться на останньому етапі згорткової нейронної мережі після всіх інших шарів. Він приймає два вхідні аргументи: очікуваний вихід та фактичний вихід. Шар втрат обчислює значення функції втрат, яка відображає помилку мережі. Вибір конкретної функції втрат залежить від типу задачі та характеру вхідних даних. Наприклад, для задач багатокласової класифікації часто використовується перехресна ентропія, а для задач регресії — середньоквадратична помилка [17].

Серед переваг згорткових нейронних мереж можна назвати [15, 17]:

автоматичне витягування ознак з вхідних даних без необхідності ручної інженерії ознак або попередньої обробки;

ефективне використання параметрів мережі шляхом спільного використання ваг та з'єднання локально сусідніх нейронів;

висока точність у задачах розпізнавання зображень, відео, мови тощо, яка перевершує багато інших алгоритмів;

легка масштабованість до великих датасетів та складних архітектур за допомогою графічних процесорів та хмарних обчислень.

В той же час в них є наступні недоліки [15, 17]:

навчання потребує великих затрат часу та ресурсів, особливо при великих датасетах чи складних архітектурах;

схильність до перенавчання, коли мережа добре пристосовується до тренувальних даних, але гірше працює для нових даних;

вразливість до шуму та аномалій у вхідних даних, які можуть негативно вплинути на якість вихідних результатів.

Рекурентні нейронні мережі (РНМ) представляють собою клас штучних нейронних мереж, в яких зв'язки між вузлами утворюють граф, орієнтований у часі. Це означає, що РНМ мають здатність зберігати інформацію про попередні вхідні дані у своїх внутрішніх станах, що дозволяє їм ефективно обробляти послідовні дані, такі як тексти, мова, аудіо або відео. РНМ можуть виконувати широкий спектр завдань, включаючи класифікацію, генерацію, переклад, розпізнавання та інші [16, 18].

Структурно РНМ складаються з трьох основних типів шарів: вхідного, прихованого та вихідного. Вхідний шар приймає послідовність векторних представлень вхідних даних, таких як слова або звуки, та передає їх до прихованого шару. Прихований шар має рекурентну структуру, що означає, що кожен його вузол отримує як поточний вхід, так і попередній стан (або контекст) інших вузлів. Прихований шар обчислює новий стан, використовуючи поточний вхід та попередній стан, за допомогою функції активації, яка може бути, наприклад, сигмоїдою або гіперболічним тангенсом. Вихідний шар отримує стани прихованого шару та генерує послідовність вихідних значень, також за допомогою функції активації [18].

Приклади структури рекурентних нейронних мереж зображено на рисунку 2.1.

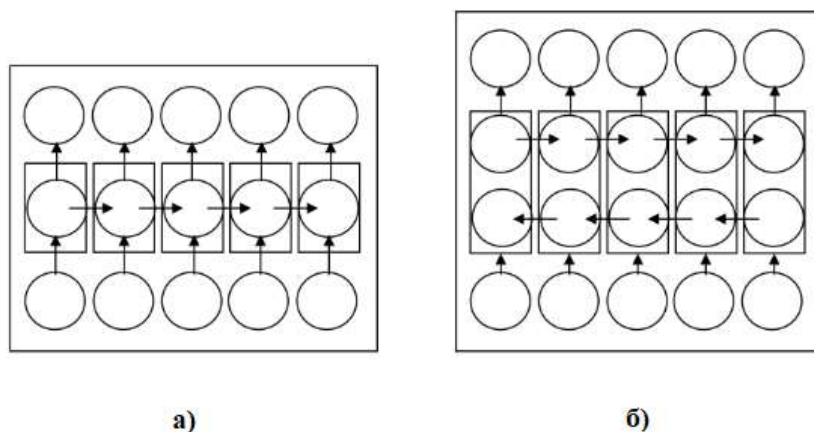


Рисунок 2.1 – Приклади структури РНМ: а) однонаправлена РНМ; б) двонаправлена РНМ

Однонаправлені рекурентні нейронні мережі (ОРНМ) є класом штучних нейронних мереж, в яких зв'язки між вузлами утворюють граф, що прямує в одному напрямку у часі. Це особливість, що дозволяє ОРНМ використовувати внутрішню пам'ять для ефективної обробки послідовностей вхідних даних. ОРНМ здатні використовувати інформацію з попередніх станів, але не мають доступу до майбутніх станів [18].

Двонаправлені рекурентні нейронні мережі (ДРНМ) є розширенням ОРНМ, у яких два прихованих шари з однаковим вхідним сигналом з'єднані в протилежних напрямках. Ця архітектура ДРНМ дозволяє вихідному шару рекурентної нейронної мережі отримувати інформацію одночасно з попередніх (минулих) та майбутніх (наступних) станів [18].

Основна відмінність між ОРНМ та ДРНМ полягає в тому, що ДРНМ здатні використовувати контекст вхідних даних з обох напрямків часу, тоді як ОРНМ обмежені використанням контексту лише з попереднього напрямку часу. Ця особливість ДРНМ робить їх більш ефективними для задач, які вимагають симетричної обробки даних, таких як розпізнавання мовлення, машинний переклад, розпізнавання рукописного тексту та інших подібних завдань [18].

В той же час, ОРНМ є більш простими у реалізації та навчанні порівняно з двонаправленими рекурентними нейронними мережами. Вони показують

кращу придатність для задач, де потрібно враховувати лише минулий контекст, а не майбутній. Наприклад, вони можуть бути використані для прогнозування наступного слова в тексті або наступного значення в часовому ряді [18].

До переваг рекурентних мереж можна віднести [16, 18]:

- РНМ ефективно обробляє послідовні дані з різною довжиною та структурою, оскільки вони використовують свою внутрішню пам'ять для збереження історії;
- вони виявляють та використовують часові залежності між входами та виходами, що дозволяє їм ефективно моделювати складну динаміку та контекст;
- такі мережі можуть генерувати послідовності з різною довжиною та структурою, що дозволяє їм створювати новий контент або перетворювати один тип даних на інший.

Рекурентні нейронні мережі мають такі недоліки [16, 18]:

- навчання РНМ потребує багато часу та обчислювальних ресурсів;
- РНМ можуть мати проблеми з градієнтами, які можуть швидко зростати або зменшуватися, ускладнюючи оптимізацію мережі під час навчання;
- вони можуть мати труднощі з запам'ятовуванням довготривалих залежностей, оскільки інформація про попередні входи може бути витиснена або розмита через багато кроків.

Мережа радіальних базисних функцій (РБФ) є одним з видів штучних нейронних мереж. Такі мережі використовують радіальні базисні функції як функцію активації. Вихід мережі формується шляхом лінійної комбінації радіальних базисних функцій вхідних сигналів та параметрів нейрона. Мережі РБФ мають широкий спектр застосувань, зокрема, в області апроксимації функцій, прогнозуванні часових рядів, вирішенні задач класифікації та керуванні системами [16].

Типова архітектура РБФ мереж включає три шари: вхідний шар, прихований шар з використанням нелінійної радіальної базисної функції активації та лінійний вихідний шар. Вхід мережі можна представити як вектор дійсних чисел, що відображає вхідні сигнали. Вихід мережі є скалярною функцією вхідного вектора і визначається як лінійна комбінація радіальних базисних функцій вхідних сигналів та параметрів нейрона [16].

Радіальна базисна функція представляє собою функцію, що приймає дійсні значення і залежить від відстані до початку координат або від відстані до певної центральної точки. Радіальна функція відповідає умовам радіальної симетрії, що означає, що її значення залежить тільки від відстані до центру, а не від конкретного напрямку. Зазвичай використовується евклідова відстань, але можна використовувати будь-які функції відстані.

Мережі радіально базисних функцій мають такий ряд переваг [16]:

- ефективна апроксимація складних функцій з різною кількістю вхідних та вихідних змінних;
- легка адаптація до нових даних шляхом додавання або видалення радіальних базисних функцій;
- вони можуть працювати з розрідженими даними та уникати проблеми перенавчання.

Але в той же час вони також мають набір недоліків [16]:

- потреба в значних обсягах пам'яті та обчислювальних ресурсів для збереження та оновлення матриці ваг;
- чутливість до шуму в даних, який може впливати на визначення центрів та ширин радіальних базисних функцій;
- можуть не мати достатньо гнучкості для моделювання деяких типів функцій, які не піддаються локальному наближенню;
- нестабільність, якщо радіальні базисні функції перекриваються або не охоплюють всю область входу.

Багатошаровий перцептрон є нейронною мережею, яка складається з кількох шарів, розташованих у спрямованій графовій структурі, де сигнал поширюється в одному напрямку через вузли мережі. Ця архітектура дозволяє багатошаровому перцептрону вирішувати складні завдання, такі як апроксимація функцій, прогнозування та розпізнавання образів [19].

Багатошаровий перцептрон має кілька шарів нейронів, які виконують обробку вхідних даних та передають отримані результати до наступного шару. Кожен нейрон у мережі має свої ваги та зсув, що визначають його активацію. Кожен шар має передавальну функцію, яка визначає вихідний сигнал для кожного нейрона у шарі [19].

Для навчання багатошарового перцептрона застосовується алгоритм зворотного поширення помилки, який коригує ваги та зсуви нейронів на основі різниці між очікуваним та отриманим виходом мережі. Багатошаровий перцептрон має здатність апроксимувати будь-яку функцію, при умові, що в ньому наявна достатня кількість нейронів у прихованому шарі. Завдяки своїй глибокій архітектурі та здатності до навчання за допомогою алгоритму зворотного поширення помилки, це потужний інструмент для розв'язання задач прогнозування, апроксимації функцій та розпізнавання образів [19].

Приклад типової структури багатошарового перцептрона зображено на рисунку 2.2.

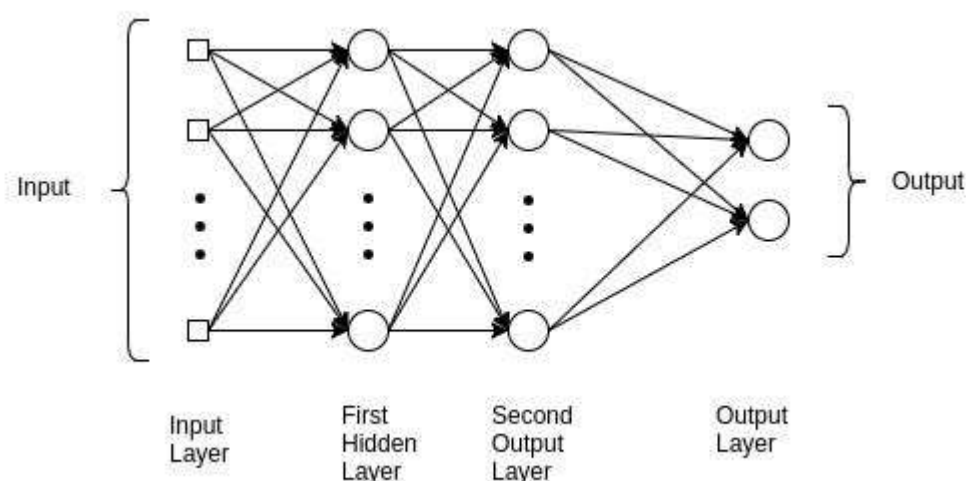


Рисунок 2.2 – Типова структура багатошарового перцептрона

Вихідний сигнал k -го нейрона вихідного шару розраховується за наступною формулою [19]:

$$y_k = f\left(\sum_{j=1}^m w_{kj}h_j + b_k\right), \quad (2.1)$$

де f – це функція активації;

w_{kj} – це вага зв'язку між j -м нейроном прихованого шару та k -м нейроном вихідного шару;

h_j – це вихідний сигнал j -го нейрона прихованого шару;

b_k – зсув k -го нейрона вихідного шару.

В багатошарового перцептрону є наступні переваги [18, 19]:

- здатність апроксимувати будь-яку функцію з бажаною точністю, якщо він має достатню кількість нейронів у своєму прихованому шарі;
- може використовувати різні функції активації для реалізації різних типів нелінійностей;
- ефективно працює з багатьма входами та виходами.

В той же час, у цього виду мережі є такі недоліки [18, 19]:

- може застрягати в локальних мінімумах функції помилки;
- ризик перенавчання мережі на тренувальних даних;
- не може ефективно обробляти динамічні або послідовні дані через відсутність внутрішньої пам'яті та зворотніх зв'язків.

Враховуючи переваги, недоліки та особливості розглянутих видів нейронних мереж, було обрано багатошаровий перцептрон як найоптимальніший для вирішення даної задачі.

Згорткові нейронні мережі краще підходять для роботи із зображеннями та відео, але вони не такі ефективні для розв'язку задач прогнозування даних у векторній формі. Крім того, вони вимагають більше обчислювальних ресурсів та пам'яті, ніж багатошаровий перцептрон. Рекурентні нейронні мережі набагато ефективніше працюють з даними, що мають послідовний характер, такими як аудіо чи мова. В задачі даного дослідження дані не мають такого характеру. РНМ також погано запам'ятовують довготривалі залежності через витіснення інформації. Радіально базисні мережі легко адаптуються до нових даних та добре апроксимують гладкі функції, але багатошаровий перцептрон може досягти вищої точності при достатній кількості нейронів прихованого шару. РБФ мережі потребують більше часу та ресурсів, більш чутливі до шуму в даних і вони мають меншу гнучкість. Це обґрунтовує вибір багатошарового перцептрону в даному дослідженні [1].

2.2 Розробка структури багатошарового перцептрону

Як уже було визначено у підрозділі вище, базова структура багатошарового перцептрону складається з кількох шарів, один з яких приймає вхідні дані на вхідні нейрони, приховані шари виконують подальшу обробку, а вихідний шар формує результат.

Кількість нейронів у вхідному шарі залежить від кількості вхідних параметрів. Відповідно до постановки задачі, вхідними даними є значення 10 економічних величин:

- рентабельність;
- дохід;
- собівартість
- валовий прибуток (збиток)
- витрати;
- вартість основних фондів;
- середньорічна заробітна плата працівників;
- кількість працівників;
- кредитна заборгованість;
- дебіторська заборгованість.

Для прогнозування рентабельності та валового прибутку (збитку) будуть використані відповідні значення цього та двох попередніх років. В результаті маємо 30 вхідних значень. Вони будуть приходити на 30 нейронів вхідного шару мережі.

Велика кількість прихованих шарів може призвести до перенавчання мережі, тобто до ситуації, коли мережа точно працює на навчальній вибірці, але вона не така ефективна для нових вхідних даних [19]. Тому не варто брати більше 2 прихованих шарів. При цьому 2 приховані шари можуть моделювати більш складні та нелінійні функції, порівняно з 1 прихованим шаром. Недоліком 2 прихованих шарів є ризик перенавчання мережі. Враховуючи можливі складні залежності між економічними величинами було прийнято рішення створити 2 приховані повнозв'язні шари.

Оскільки валовий прибуток та валовий збиток є протилежними значеннями [3], обидві величини можуть передаватися одним нейроном, що

прийматиме додатні (прибуток) або від'ємні (збиток) значення. Тобто вихідний шар буде складатися з двох вихідних нейронів.

У першому прихованому шарі було встановлено кількість нейронів, яка рівна середньому арифметичному між кількістю вхідних нейронів та кількістю вихідних нейронів, що складає 16 нейронів. Аналогічно, для другого прихованого шару було прийнято рішення про кількість нейронів, яка дорівнює середньому арифметичному між кількістю нейронів у першому прихованому шарі та кількістю вихідних нейронів, що становить 9 нейронів.

На основі проведеного аналізу було сформовано структуру нейронної мережі, яка зображена на рисунку 2.3.

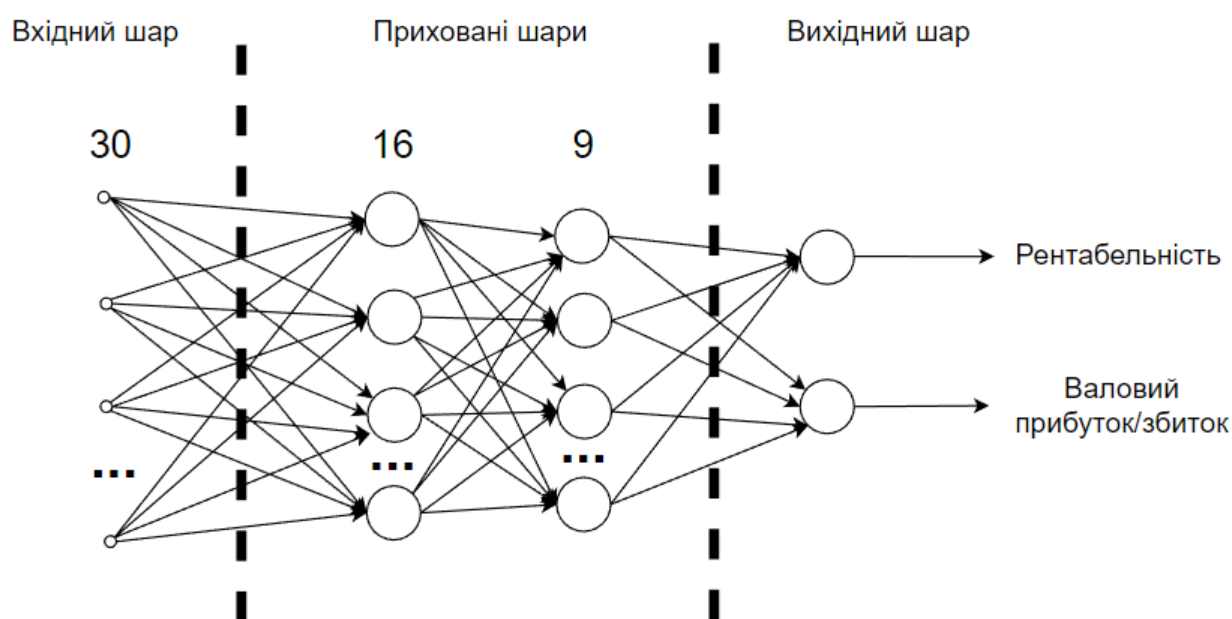


Рисунок 2.3 – Структура багатошарового перцептрону для прогнозування рентабельності та валового прибутку/збитку підприємства

Таким чином, було розроблено структуру нейронної мережі, що буде використовуватись для реалізації інформаційної технології прогнозування рентабельності, валового прибутку та валового збитку підприємства.

2.3 Обґрунтування алгоритму навчання багат шарового перцептрону та підготовка навчальної вибірки

Процес навчання багат шарового перцептрону полягає в адаптації ваг синаптичних зв'язків між нейронами різних шарів мережі з метою мінімізації помилки передбачення на виході мережі [19].

Методи навчання нейронних мереж поділяють на дві загальні категорії: навчання з учителем та без учителя.

Методи навчання з учителем базуються на наявності еталонних міток для кожного об'єкта в навчальній вибірці. Головною метою цих методів є встановлення залежності між вхідними ознаками та відповідями системи, що дозволяє прогнозувати відповіді для нових об'єктів. Наприклад, такі методи використовуються для класифікації, регресії, розпізнавання образів тощо [20].

Методи навчання без учителя, натомість, не потребують наявності еталонних міток для об'єктів у навчальній вибірці. Головна мета цих методів полягає в виявленні внутрішніх взаємозв'язків, залежностей та закономірностей між об'єктами. Наприклад, такі методи використовуються для кластеризації, узагальнення, пошуку асоціативних правил, зниження розмірності, візуалізації даних тощо [20].

Перевагами навчання з учителем можна назвати [20]:

- можливість вимірювати якість моделі за допомогою функції втрати або метрик точності;
- можливість навчити модель точно відповідати певним прикладам з попередньо заданими мітками або цільовими значеннями.

Серед недоліків навчання з учителем є [20]:

- потреба в достатній кількості попередньо розмічених даних, які можуть бути складними або дорогими для отримання;
- може спричинити перенавчання або недонавчання, якщо модель не налаштована належним чином.

Тим часом переваги навчання без учителя включають [20]:

- можливість виявляти складні закономірності та структури в даних без попереднього знання про правильну відповідь;
- не потребує попереднього розмічення даних, що зменшує час та витрати на підготовку даних.

А недоліки навчання без учителя такі [20]:

- може бути нестабільним або неефективним для деяких задач;
- не дозволяє вимірювати якість моделі за допомогою функції втрати або метрик точності, а лише за допомогою евристичних критеріїв або експертної оцінки.

Як правило, для навчання багатошарових перцептронів використовують метод зворотного поширення помилки [18, 19]. Цей метод відноситься до методів навчання з учителем, і тому він має відповіді переваги та недоліки. Крім вище названих, до переваг цього методу належать [19]:

- можливість навчити всі шари нейронної мережі;
- можливість використання різних функцій активації та оптимізації.

Серед його недоліків також є [19]:

- можливість потрапляння в локальні мінімуми;
- можливість паралічу мережі, тобто ситуації, коли більшість нейронів функціонують при дуже великих вихідних значеннях в області, де похідна стискаючої функції дуже мала.

Цей ітеративний градієнтний алгоритм застосовується з метою мінімізації помилки функціонування багатошарового перцептрону та досягнення очікуваних виходів. Основна концепція цього методу полягає в передачі сигналів помилки від виходів мережі до її входів у зворотному напрямку від прямого розповсюдження сигналів під час нормального функціонування. Для успішного використання методу зворотного поширення помилки, функція активації нейронів повинна бути диференційовною [19].

Даний метод характеризується таким алгоритмом [19]:

- 1) Вхідний сигнал надсилається на перший шар мережі, де кожен нейрон обчислює свій вихідний сигнал за допомогою функції активації та ваг зв'язків.
- 2) Порівнюється отриманий вихідний сигнал останнього шару мережі з очікуваним сигналом (міткою), і обчислюється помилка передбачення за допомогою функції втрати.
- 3) Помилка розповсюджується зворотно по шарах мережі, і обчислюються градієнти функції втрати по вагах зв'язків згідно з правилом ланцюгового диференціювання.
- 4) Ваги зв'язків оновлюються в напрямку, протилежному градієнту, за допомогою методу оптимізації, такого як стохастичний градієнтний спуск або його модифікації.
- 5) Цей процес повторюється для кожного прикладу з тренувального набору даних до тих пір, поки помилка передбачення не досягне заданого рівня або не стабілізується.

Корекція ваг у цьому алгоритмі відбувається за наступною формулою [19]:

$$w_{kj} = w_{kj} + \alpha(t_k - y_k)f'(y_k)h_j, \quad (2.2)$$

де α – коефіцієнт навчання;

t_k – цільовий сигнал k-го нейрона вихідного шару;

y_k – вихідний сигнал k-го нейрона вихідного шару;

f – функція активації;

h_j – вихідний сигнал j-го нейрона прихованого шару.

Загалом, можна виділити такі критерії для вибору функції активації [17]:

- 1) Функція активації повинна мати нелінійний характер, щоб забезпечити здатність мережі моделювати складні та нелінійні залежності між вхідними та вихідними даними.
- 2) Функція активації повинна бути диференційовною, щоб дозволити обчислення градієнта помилки та оновлення ваг мережі за допомогою градієнтного спуску.
- 3) Функція активації повинна мати прийнятний діапазон значень, щоб уникнути проблем з насиченням або зникненням градієнтів.

Оскільки це задача прогнозування значення числових характеристик, для неї потрібно вибрати таку функцію активації для вихідного шару мережі, яка не обмежує діапазон вихідних значень і показує лінійну залежність між входом і виходом. Крім того, варто врахувати, що мережі доведеться працювати із від'ємними даними. Такою функцією є ELU (Exponential Linear Unit), або експоненціально-лінійний зрізаний лінійний вузол. Вона приймає значення в діапазоні $(-\infty, +\infty)$ і показує кусково-лінійну перехідну характеристику [19].

Ця функція активації виглядає таким чином [19]:

$$\begin{cases} f(x) = x, \text{ якщо } x > 0; \\ f(x) = \alpha(e^x - 1), \text{ якщо } x \leq 0. \end{cases} \quad (2.3)$$

Як видно з її формули, вона приймає значення x для всіх додатніх значень x , та $\alpha(e^x - 1)$ для всіх інших значень. Параметр α визначає, наскільки сильно ELU зглажує або насичує від'ємні значення. Зазвичай цьому параметру надають значення 1, а тоді збільшують за потреби [19]. При цьому значенні параметра насичення не буде занадто різким. Графік функції ELU при $\alpha = 1$ зображено на рисунку 2.4.

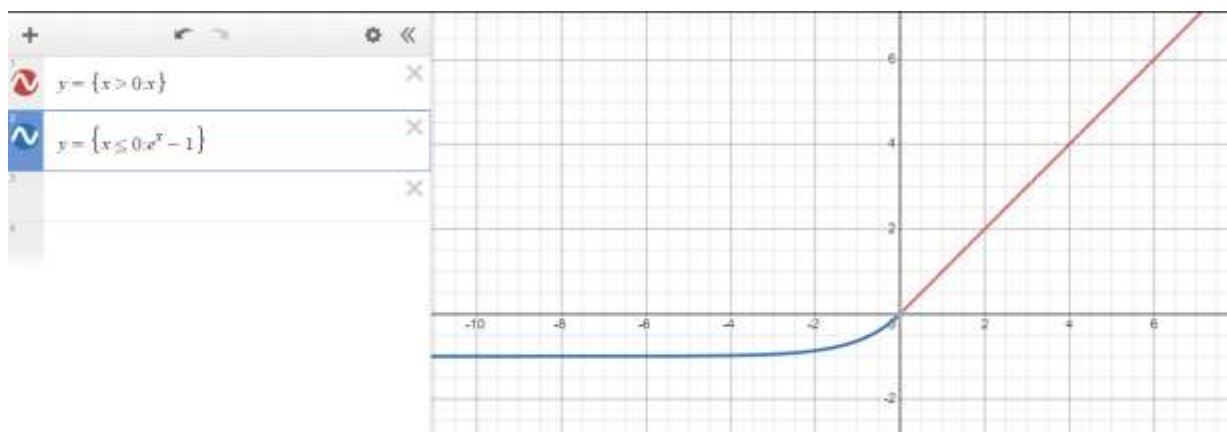


Рисунок 2.4 – Графік функції активації ELU ($\alpha = 1$)

Функція ELU має такі переваги над іншими функціями активації, що приймають від’ємні значення [18, 19]:

- 1) Швидша конвергенція під час навчання моделі, що допомагає уникнути проблем з градієнтами.
- 2) Зменшення зсуву активацій, що допомагає мережі краще працювати без додаткових налаштувань.
- 3) Менш чутлива до коливань у вхідних даних.
- 4) Обмежує надмірно великі негативні значення, що захищає від вибуху значень.
- 5) Менш схильна до перенавчання.

Проте, функція активації ELU також має деякі недоліки, які треба врахувати [18, 19]:

- 1) Вища обчислювальна складність порівняно з простішими функціями, як PReLU. Це збільшує час навчання.
- 2) Присутнє насичення негативних активацій, що може сповільнити навчання у глибоких мережах.
- 3) Ця функція не адаптивна, її форма залишається фіксованою під час навчання.

Зважаючи на її переваги та недоліки, було прийнято рішення використати функцію ELU в якості функції активації для навчання даної мережі.

Для формування навчальної та тестової вибірки буде використано дані офіційного сайту Агенства з розвитку інфраструктури фондового ринку України, що знаходиться за адресою <https://smida.gov.ua>. Цей сайт надає вільний доступ всім користувачам до фінансової звітності підприємств, що зареєстровані на території України. Приклад річного звіту підприємства зображено на рисунку 2.5.

Баланс (Звіт про фінансовий стан)
на 31.12.2017 р.

Актив	Код рядка	На початок звітного періоду	На кінець звітного періоду	На дату переходу на міжнародні стандарти фінансової звітності
1	2	3	4	5
I. Необоротні активи				
Нематеріальні активи:	1000	7483	8148	0
первісна вартість	1001	36668	38982	0
накопичена амортизація	1002	-29185	-30834	0
Незавершені капітальні інвестиції	1005	214	721	0
Основні засоби:	1010	214656	183335	0
первісна вартість	1011	667371	636620	0

Рисунок 2.5 – Приклад річного звіту підприємства

Вхідні дані мережі включають 10 фінансових показників:

- 1) рентабельність;
- 2) валовий прибуток (збиток);
- 3) витрати;
- 4) дохід;
- 5) собівартість;
- 6) вартість основних фондів;
- 7) середньорічна заробітна плата працівників;
- 8) кількість працівників;

- 9) кредитна заборгованість;
- 10) дебіторська заборгованість.

В річному звіті прибуток та витрати мають коди рядка 2090 та 2150 відповідно. Значення рентабельності не вказується, але його можна отримати поділивши дохід на витрати. Дохід та собівартість знаходяться на рядках з кодом 2000 та 2050. Вартість основних фондів можна знайти за кодом рядка 1010. Кількість працівників не вказана в фінансовому звіті, але вона є в основних відомостях про емінента. Середньорічну заробітну плату можна знайти за наступною формулою [3]:

$$\frac{\text{Вартість нематеріальних активів} \cdot \text{Витрати на оплату праці}}{\text{Кількість працівників}}, \quad (2.4)$$

де вартість нематеріальних активів і витрати на оплату праці мають коди рядка 1000 і 3105 відповідно.

Кредиторська заборгованість має код рядка 1610, а дебіторська – 1040.

Таким чином, для навчання мережі буде зібрано дані 1000 підприємств, а для її тестування – дані інших 100 підприємств. Для зручності, вони зберігатимуться в файлах програми Microsoft Excel.

2.4 Розробка алгоритму роботи інформаційної технології прогнозування рентабельності підприємства

Для розробки алгоритму інформаційної технології потрібно спочатку розробити його структуру. Структура інформаційної технології включатиме такі компоненти:

- модуль навігації;
- модуль навчання мережі;
- модуль тестування мережі;

– модуль прогнозування.

Модуль навігації забезпечує користувача можливістю перегляду та вибору різних функцій програми шляхом відображення меню на екрані. Модуль навігації створює інтуїтивно зрозуміле користувачеві середовище, де він може оглядати доступні функції програми та обирати ту, яка відповідає його потребам та вимогам.

Модуль навчання мережі використовує дані, вибрані користувачем, для оптимізації вагових коефіцієнтів мережі та налаштування її параметрів з метою забезпечення точності прогнозування рентабельності, валового прибутку та збитку підприємства.

Для оцінки якості прогнозування використовується модуль тестування мережі, який проводить тестування мережі на обраних користувачем тестових даних. Після завершення тестування цей модуль надає користувачеві інформацію про середню похибку прогнозування, що дозволяє оцінити ефективність розробленого модуля.

Останнім модулем є модуль прогнозування, який надає можливість користувачеві вручну ввести економічні показники свого підприємства і отримати прогноз значення рентабельності і прибутку/збитку на наступний рік. Цей модуль дозволяє використовувати розроблений інформаційна технологія для індивідуального аналізу та планування рентабельності підприємства.

Структура інформаційної технології зображена на рисунку 2.6.

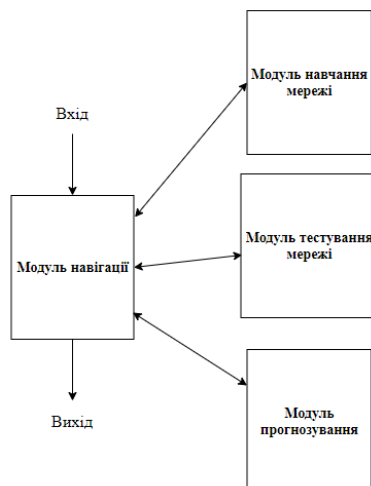


Рисунок 2.6 – Структура інформаційної технології прогнозування рентабельності підприємства

Інформаційна технологія працюватиме за таким алгоритмом:

- 1) Згенерувати головне меню програми та чекати на вибір користувача. Якщо користувач натиснув на кнопку «Прогнозувати величини», перейти до пункту 2. Якщо користувач вибрав кнопку тестування мережі, перейти до пункту 6. Якщо користувач обрав навчання мережі, перейти до пункту 11. Якщо ж користувач натиснув на кнопку «Вихід», завершити роботу програми.
- 2) Чекати на введення вхідних даних користувачем і натиснення кнопки прогнозування.
- 3) Нормалізувати введені дані для використання в мережі.
- 4) Виконати прогнозування рентабельності і валового прибутку(збитку) та вивести результат на екрані.
- 5) Якщо користувач натиснув на кнопку меню, перейти до пункту 1. Інакше перейти до пункту 2.
- 6) Очікувати на вибір файлу з тестовими даними.
- 7) Зчитати дані з файлу та нормалізувати їх.
- 8) Виконати тестування мережі.
- 9) Вивести результати тестування на екрані.

- 10) Якщо користувач натиснув на кнопку меню, перейти до пункту 1.
Інакше перейти до пункту 6.
- 11) Чекати на вибір файлу з навчальними даними.
- 12) Зчитати навчальну вибірку з файлу та нормалізувати її.
- 13) Провести навчання мережі.
- 14) Якщо користувач натиснув на кнопку меню, перейти до пункту 1.
Інакше перейти до пункту 11.

Описаний алгоритм зображено у вигляді блок-схеми на рисунку 2.7.

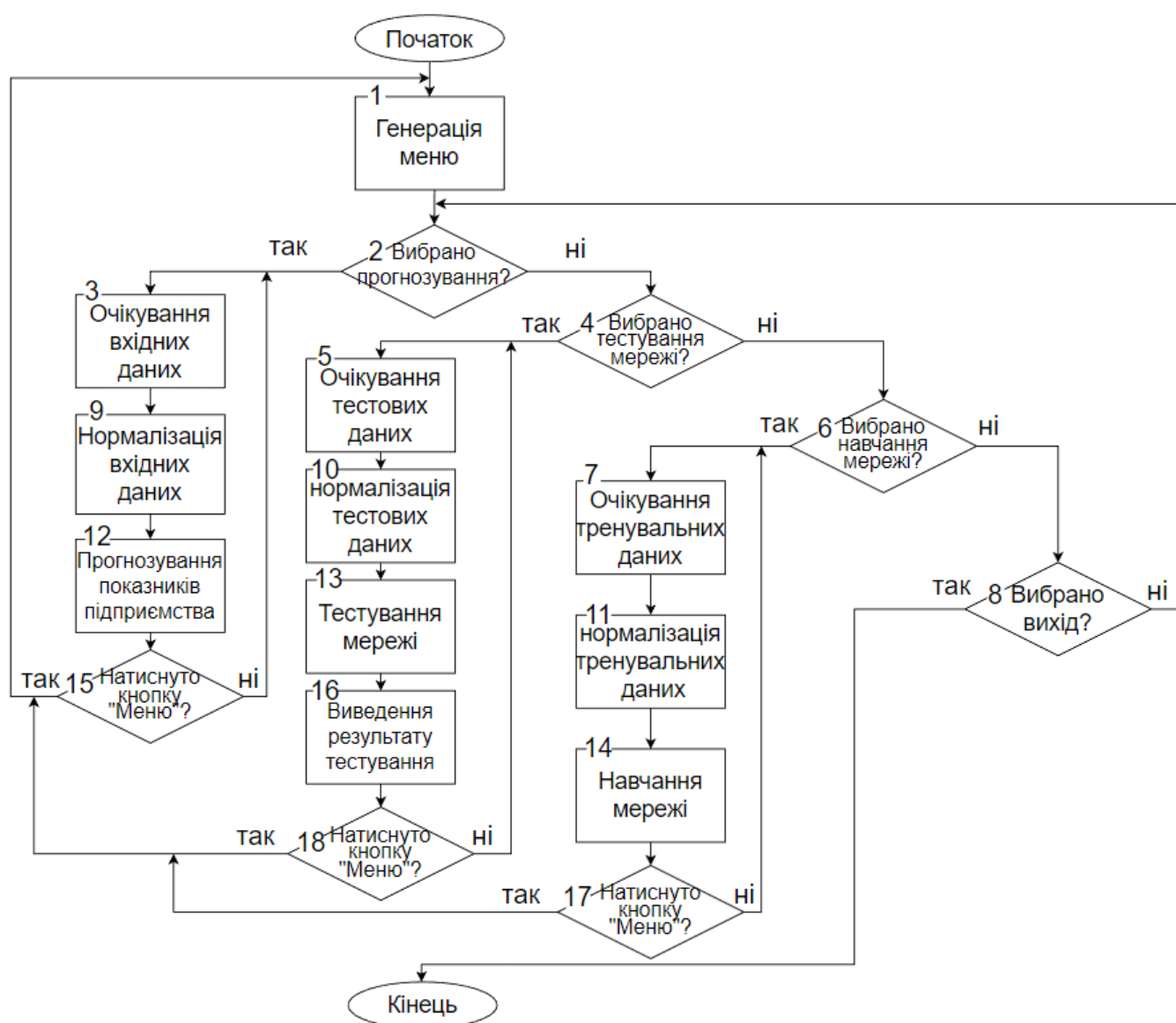


Рисунок 2.7 – Схема алгоритму роботи інформаційної технології прогнозування рентабельності та валового прибутку/збитку

В подальшому ця схема алгоритму використовується для розробки програмного забезпечення.

2.5 Розробка UML-діаграми класів

UML-діаграма класів є статичним відображенням структури моделі в UML в рамках об'єктно-орієнтованого програмування. Її призначенням є відображення статичної структури системної моделі, зокрема класів, інтерфейсів, об'єктів і кооперацій, а також їх взаємозв'язків [21].

Класи проектного у роботі інформаційної технології та їх зв'язки зображено на UML-діаграмі класів (рисунок 2.8). Взаємодія цих класів утворює комплексну програму для вирішення задачі прогнозування рентабельності з використанням нейронної мережі.

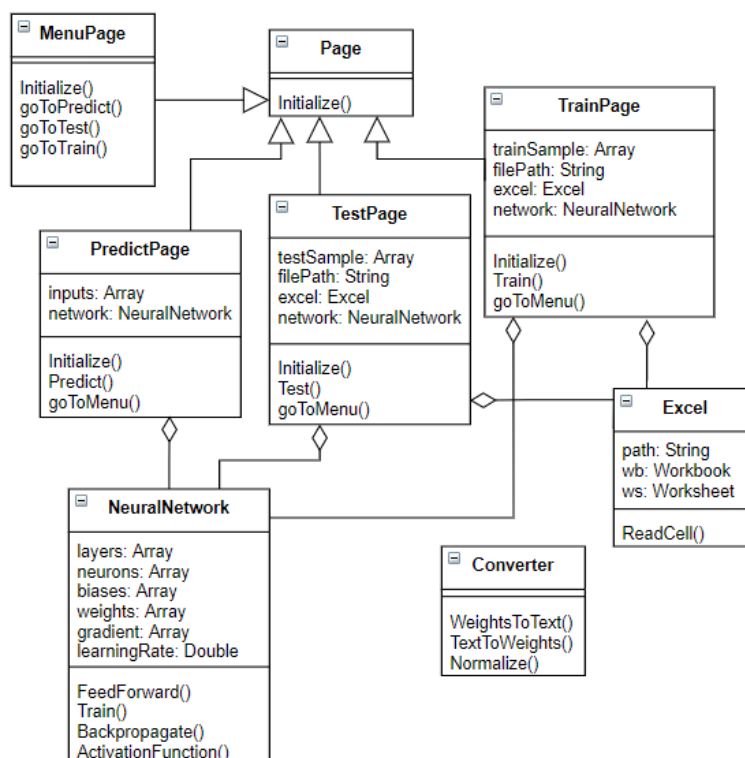


Рисунок 2.8 – Діаграма класів інформаційної технології прогнозування рентабельності

Клас Excel – це клас, що реалізує роботу з файлами програми Excel, в яких зберігаються тестові та навчальні дані. До атрибутів цього класу належать: адреса файлу, з яким ведеться робота, структура таблиць і власне таблиці файлу. Метод ReadCell() зчитує вказані комірки з вибраної таблиці. Цей клас пов'язаний зв'язками агрегації зі сторінками (активностями) тестування і навчання програми, оскільки ці сторінки містять екземпляри цього класу.

Клас Page – це батьківський клас до класів сторінок програми. Його метод ініціалізує сторінку та зображає її у вікні програми.

Класи MenuPage, PredictPage, TestPage та TrainPage мають зв'язок наслідування відносно класу Page. Методи goToMenu(), goToPredict, goToTest() та goToTrain() реалізують перехід програми до відповідної сторінки.

Клас PredictPage – це сторінка прогнозування рентабельності на основі введених користувачем вхідних даних. Вона зберігає масив вхідних даних та екземпляр нейронної мережі. Функція Predict() передає мережі вхідні дані, щоб вона виконала прогноз.

Клас TestPage – це сторінка тестування роботи мережі. Він зберігає адресу файлу з тестовим набором даних, екземпляр класу Excel для зчитування цього файлу, масив для збереження тестового набору і екземпляр мережі. Метод Test() виконує тестування мережі та виводить на екран результат проведених тестів.

Клас TrainPage – це сторінка навчання мережі. Серед атрибутів цього класу є адреса файлу з навчальною вибіркою, екземпляр класу Excel для роботи з цим файлом, масив для збереження тренувальних даних і екземпляр мережі. Метод Train() проводить навчання нейронної мережі на завантаженій вибірці.

Клас MenuPage є сторінкою головного меню, що містить методи для навігації до інших сторінок програми.

Клас Converter – це статичний клас, що реалізує методи для конвертації значень вагових коефіцієнтів мережі в текст для збереження в текстовому файлі і, навпаки, переведення тексту в масив ваг. Також він реалізує метод нормалізації чисел для їх подання в мережу. Методи цього класу використовуються в класах PredictPage, TestPage та TrainPage.

Клас `NeuralNetwork` відповідає за роботу нейронної мережі. Він зберігає розмірність шарів нейронів мережі, нейрони, упередження, вагові коефіцієнти, градієнт і коефіцієнт швидкості навчання. Серед методів класу мережі є:

- `FeedForward()`, що виконує прогнозування рентабельності на основі вхідних даних.
- `Train()`, що тренує мережу на основі навчальної вибірки.
- `Backpropagate()`, що реалізує метод зворотнього поширення помилки.
- `ActivationFunction()`, що виконує функцію активації ReLU.

Оскільки екземпляри цього класу є в класах `PredictPage`, `TestPage` та `TrainPage`, клас `NeuralNetwork` пов'язаний з ними зв'язками агрегації.

Таким чином, в даному підпункті було розроблено UML-діаграму класів та описано її компоненти, що буде використано під час реалізації інформаційної технології в наступному розділі роботи.

2.6 Висновок до розділу 2

В другому розділі було проведено порівняльний аналіз видів нейронних мереж та на основі цього аналізу було вибрано багатошаровий перцептрон.

Також було створено структуру цієї мережі, що містить 2 приховані шари нейронів. Вхідний шар складається з 30 нейронів. У першому прихованому шарі міститься 16 нейронів, а в другому – 9 нейронів. На виході мережі є 2 нейрони.

Крім того, було досліджено методи навчання з учителем та без учителя. Було вибрано метод зворотного поширення помилки з функцією активації ELU як метод навчання проектованої нейронної мережі. Було сформовано навчальну вибірку розміром в 1000 наборів даних.

Також було розроблено структуру та алгоритм роботи інформаційної технології. Структура складається з модуля навігації, модуля навчання мережі, модуля тестування мережі та модуля прогнозування. Алгоритм було наведено в описовому вигляді та у вигляді блок-схеми. Було розроблено UML-діаграму класів та описано самі класи, які вона представляє. Цими класами є `Excel`, `Page`, `MenuPage`, `PredictPage`, `TestPage`, `TrainPage`, `Converter` та `NeuralNetwork`.

З ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ РЕНТАБЕЛЬНОСТІ, ВАЛОВОГО ПРИБУТКУ ТА ВАЛОВОГО ЗБИТКУ ПІДПРИЄМСТВА

3.1 Обґрунтування вибору мови та середовища програмування

В даному дослідженні було розглянуто дві мови програмування в якості мови для реалізації спроектованого інформаційної технології, а саме C# та C++.

C# є об'єктно-орієнтованою мовою програмування, розробленою компанією Microsoft. Ця мова має широке застосування у розробці програмного забезпечення та входить до складу платформи .NET Framework. Вона характеризується сучасним синтаксисом, потужними інструментами розробки та розширеними можливостями, що дозволяють розробникам створювати різноманітні додатки [22].

Вона підтримує основні принципи ООП, такі як спадкування, поліморфізм, інкапсуляція та абстракція. Однією з головних переваг C# є її велика бібліотека класів та фреймворків, що дозволяють розробникам використовувати готові компоненти та функціональні можливості для швидкої реалізації проектів. Наприклад, платформа .NET Framework надає багато бібліотек для роботи з мережевими протоколами, базами даних, графікою та багато іншого.

C# підтримує багатопоточність, що дозволяє розробникам створювати паралельні та асинхронні програми для ефективної обробки даних та поліпшення продуктивності. Вона також має вбудовану систему збирання сміття, яка автоматично видаляє непотрібні об'єкти з пам'яті, спрощуючи управління ресурсами [22].

В порівнянні з деякими низькорівневими мовами, C# може мати деяке зниження продуктивності через використання високорівневих конструкцій та залежність від виконавчої платформи [22].

C++ є потужною мовою програмування загального призначення, що поєднує в собі можливості низькорівневого програмування з високорівневими конструкціями. Вона була розроблена як розширення мови C з метою надати програмістам більший контроль над системними ресурсами та оптимізацією продуктивності [22].

Основними рисами C++ є його підтримка об'єктно-орієнтованого програмування та загальної множини мови C, що дозволяє розробникам створювати класи, об'єкти та використовувати принципи ООП. Крім того, мова має велику стандартну бібліотеку, яка надає різноманітні функції для роботи зі строками, контейнерами даних, алгоритмами сортування та багато іншого [22].

Однією з найбільших переваг C++ є його прямий доступ до пам'яті та можливість використання низькорівневих конструкцій, таких як вказівники та управління пам'яттю. Це дає розробникам більшу свободу та контроль над ресурсами системи, що особливо корисно для вимогливих застосунків з високою продуктивністю, наприклад, вбудованих систем, графічних додатків або драйверів [22].

Проте, використання C++ також може бути складним і вимагати більшої експертизи в порівнянні з високорівневими мовами програмування. Це пов'язано з необхідністю вручну управляти пам'яттю та ресурсами, що може призвести до появи помилок, таких як витоки пам'яті або некоректна робота з показниками. Крім того, розробка програм на C++ може бути більш часо- та ресурсозатратною у порівнянні з іншими мовами вищого рівня [22].

Переваги C++ над C# [22]:

- продуктивність;
- крос-платформенність;
- контроль над пам'яттю;

Переваги C# над C++ [22]:

- простота використання;
- автоматичне управління пам'яттю;
- безпека;

Оскільки дана дослідна робота не передбачає необхідності прямого контролю над пам'яттю чи розробка крос-платформенного рішення та програма не вимагатиме високої продуктивності, для реалізації інформаційної технології було вибрано мову програмування C#. Вибір цієї мови обумовлений кращою підтримкою об'єктно-орієнтованого підходу до програмування, що сприятиме зручності в реалізації програмного рішення.

В якості можливого середовища програмування розглядалися Visual Studio 2022 та Visual Studio Code.

Visual Studio є одним з найпопулярніших та найрозповсюдженіших інтегрованих середовищ розробки, призначених для розробки програмного забезпечення на різних мовах програмування, включаючи мову C#. Розроблений компанією Microsoft, Visual Studio забезпечує розширені можливості та інструменти для зручної та ефективної розробки програм [23].

Visual Studio надає редактор коду з підсвічуванням синтаксису, автодоповненням та інтелектуальними функціями, що полегшують написання коду. Воно також має налагоджувач, який дозволяє здійснювати крок за кроком виконання коду для виявлення та усунення помилок. Однією з ключових переваг Visual Studio є його широкий функціонал, що включає інструменти для керування версіями, автоматичну генерацію коду, підтримку тестування, візуальний дизайнер і багато іншого. Крім того, Visual Studio інтегрується з іншими інструментами Microsoft, такими як Azure Cloud, SQL Server та Team Foundation Server, що полегшує розробку, відлагодження та розгортання програмного забезпечення [23].

Visual Studio також надає можливості для розробки крос-платформених додатків на платформі .NET Core, що дозволяє створювати програми, які

працюють на різних операційних системах, таких як Windows, macOS та Linux [23].

Visual Studio Code є безкоштовним текстовим редактором, розробленим компанією Microsoft, який зарекомендував себе як популярний інструмент для розробки програмного забезпечення на різних мовах програмування, включаючи мову C#. Він надає розширені можливості, гнучкість та широкий спектр інтеграційних можливостей [23].

Цей редактор вирізняється своєю легкістю та швидкістю, завдяки чому редактор запускається швидко і має малі вимоги до ресурсів системи. Він також має інтуїтивний інтерфейс та приємний користувацький досвід, що полегшує роботу розробників [23].

Однією з ключових переваг Visual Studio Code є його розширюваність. Редактор підтримує велику кількість розширень, які дозволяють налаштувати середовище розробки під власні потреби. Це включає розширення для підсвічування синтаксису, автодоповнення коду, налагоджувача, системи керування версіями і багато іншого. Завдяки цьому, розробники можуть налаштувати Visual Studio Code для своїх унікальних потреб та робочого процесу [23].

Він також відмінно інтегрується з іншими інструментами і сервісами, такими як Git, Docker, Azure Cloud та інші. Це дозволяє розробникам зручно працювати з різними технологіями та платформами [23].

Незважаючи на свою легкість та простоту, це середовище має високу функціональність і потужність, що задовольняє потреби різних типів розробників. Він підтримує роботу з багатьма мовами програмування, включаючи C#, і забезпечує розширені можливості розробки, налагодження та тестування програмного забезпечення [23].

Переваги Visual Studio Code над Visual Studio 2022 [23]:

- легкість та швидкість;
- гнучкість;

- багато інструментів для Web розробки.

Переваги Visual Studio 2022 над Visual Studio Code [23]:

- функціональність;
- автоматизація;
- багатий набір інструментів для роботи в .Net.

Враховуючи той факт, що розробка даної програми відбуватиметься на платформі .NET, було прийняте рішення вибрати Visual Studio 2022 в якості основного інструменту розробки.

Для реалізації інтерфейсу було вибрано графічну підсистему .Net WPF.

Windows Presentation Foundation (WPF) – це технологія розробки програмного забезпечення, яка надає можливості для створення графічних інтерфейсів користувача в операційній системі Windows. WPF включає набір інструментів та фреймворків, які дозволяють розробникам створювати багатофункціональні, стильні та інтерактивні додатки з високою ступенем контролю над візуальним представленням [24].

WPF базується на мові програмування XAML, яка дозволяє описувати візуальні елементи і їх взаємодію з кодом. Це дозволяє розробникам відокремити логіку програми від візуального представлення, що спрощує роботу з дизайном та підтримку розділення відповідальностей [24].

Однією з основних переваг WPF є можливість створювати багатошарові та гнучкі інтерфейси, використовуючи вбудовану систему стилів, анімацію, трансформації та ефекти. WPF також підтримує розширення взаємодії з користувачем, такі як введення даних, перетягування елементів і вбудовані механізми управління подіями [24].

Іншою важливою перевагою WPF є його можливості для масштабування. Завдяки використанню векторної графіки та роздільній здатності незалежній від пристрою, WPF дозволяє створювати додатки, які працюють як на роздільних здатностях екрану, так і на великих дисплеях з високою роздільною здатністю [24].

Крім того, WPF підтримує вбудовану систему зв'язування даних, що спрощує роботу з обміном даними між різними елементами і джерелами даних. Це дозволяє створювати динамічні інтерфейси, які автоматично оновлюються при зміні даних [24].

Загалом, WPF є потужним інструментом для розробки додатків з високоякісним графічним інтерфейсом в середовищі Windows. Його переваги полягають у великому наборі функцій, зручному розділенні візуального представлення і логіки програми, можливостях масштабування та вбудованих інструментах для роботи з даними [24].

Отже, в результаті проведеного аналізу було прийняте рішення реалізовувати інформаційна технологія прогнозування рентабельності підприємства мовою C# в середовищі Visual Studio 2022, використовуючи графічну підсистему WPF для реалізації інтерфейсу.

3.2 Реалізація програмних засобів прогнозування рентабельності, валового прибутку та валового збитку

Під час розробки програми було реалізовано класи та методи, описані в другому розділі роботи. Далі буде наведено приклади реалізації деяких з них.

Клас Excel, що реалізує роботу з файлами Excel, було запрограмовано таким чином:

```
internal class Excel
{
    string path = "";
    _Application excel = new _Excel.Application();
    // use wb.close() after working with file
    public Workbook wb;
    Worksheet ws;
    public Excel(string path, int sheet)
    {
        this.path = path;
        wb = excel.Workbooks.Open(path);
        ws = wb.Worksheets[sheet];
    }
}
```

```

    }
    public double ReadCell(int i, int j)
    {
        //because Excel counts from 1
        i++;
        j++;
        if (ws.Cells[i, j].Value2 != null)
        {
            return ws.Cells[i, j].Value2;
        }
        else return -1;
    }
}

```

Для реалізації цього класу була використана бібліотека Microsoft.Office.Interop.Excel. На вхід методу зчитування комірки таблиці Excel йдуть координати комірки, яку треба зчитати. На виході цей метод повертає число, що зберігається в цій комірці.

Робота меню і навігація між сторінками програми була реалізована так:

```

private void btnPredict_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new Predict());
}

private void btnTest_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new Test());
}

private void btnTrain_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new Train());
}

private void btnExit_Click(object sender, RoutedEventArgs e)
{
    System.Windows.Application.Current.Shutdown();
}

```

Після натиску відповідної кнопки запускається один з вище наведених методів. Функція NavigationService.Navigate створює новий екземпляр класу сторінки та виводить її на екран, видаливши з пам'яті екземпляр попередньої сторінки.

Так працює функція тестування мережі:

```
private void btnTest_Click(object sender, RoutedEventArgs e)
{
    for (int i = 0; i < samplesNum; i++)
    {
        tbTestLeft.Text = "Залишилось наборів: " + Convert.ToString(samplesNum - i);
        meanError1 += System.Math.Abs(network.FeedForward(samples[i])[0] - answers1[i]);
        meanError2 += System.Math.Abs(network.FeedForward(samples[i])[0] - answers2[i]);
        AllowUIToUpdate();
    }
    meanError1 /= samplesNum;
    MessageBox.Show("Тестування завершено!\nТестова вибірка складалась з " +
        Convert.ToString(samplesNum) + "наборів\nСередня похибка прогнозування рентабельності" +
        Convert.ToString(meanError1) + "%\nСередня похибка прогнозування валового прибутку/збитку" +
        Convert.ToString(meanError2) + "%");
}
```

Середня похибка знаходиться як середнє арифметичне похибок прогнозування. Похибка прогнозування це частка ділення різниці між цільовим та прогнозованим значенням рентабельності, поділена на цільове значення рентабельності.

Наведемо деякі методи класу NeuralNetwork, що реалізує роботу багатошарового перцептрона. Так була реалізована функція активації ELU та її похідна:

```
public double ActivationFunction(double x)
{
    if (x > 0)
    {
        return x;
    }
    else
    {
        return alpha * (Math.Exp(x) - 1);
    }
}
```



```

    }
}

public double ActivationFunctionDerivative(double x)
{
    if (x > 0)
    {
        return 1;
    }
    else
    {
        return alpha * Math.Exp(x);
    }
}

```

Для її реалізації було використано бібліотеку System.Math. Ця функція порівнює значення вхідного параметра з нулем. Таким чином, для всіх чисел більше нуля вона повертає саме це число, а для чисел менше нуля – різницю між значенням експоненти в цій точці та одиниці. Похідна цієї функції в таких випадках повертає 0 та 1 відповідно.

Далі наведена функція, що виконує прогнозування на основі вхідних даних:

```

public double[] FeedForward(List<double> input)
{
    for (int i = 0; i < input.Count; i++)
    {
        neurons[0][i] = input[i];
    }

    for (int layer = 1; layer < layers.Length; layer++)
    {
        for (int neuron = 0; neuron < neurons[layer].Length; neuron++)
        {
            double sum = 0;
            for (int weight = 0; weight < weights[layer][neuron].Length;
weight++)

```

```

        {
            double weightValue = weights[layer][neuron][weight];
            double neuronValue = neurons[layer - 1][weight];
            sum += weightValue * neuronValue;
        }
        sum += bias[layer][neuron];
        neurons[layer][neuron] = ActivationFunction(sum);
    }
}

int outputLayer = neurons.Length - 1;
return new double[] { neurons[outputLayer][0], neurons[outputLayer][2]
};}

```

На вхід цієї функції подається список з вхідними даними. Після обробки даних, вона повертає масив значень на 2 вихідних нейронах.

Так виглядає функція, що навчає мережу:

```

public void Train(List<double> input, List<double> desiredOutputs)
{
    double[] predictions = FeedForward(input);
    Backpropagate(predictions, desiredOutputs);
}

```

Параметрами цієї функції є список тренувальних даних і список цільових значення прогнозування. Ця функція викликає метод зворотного поширення помилки.

Наступна функція реалізує метод зворотного поширення помилки:

```

private void Backpropagate(double[] generatedOutputs, List<double>
targetOutputs)
{
    error = CalcError(generatedOutputs, targetOutputs);
    CalcOutputLayerWeightGradient(generatedOutputs, targetOutputs);
    CalcHiddenLayerWeightGradient();
    UpdateWeights();
}

```

На вході цієї функції списки спрогнозованих та цільових значень. Вона обраховує помилку прогнозування та обчислює градієнт ваг вихідного та прихованих шарів. Тоді ця функція оновлює вагові коефіцієнти.

Таким чином було реалізовано інформаційну технологію з використанням бібліотек `Microsoft.Office.Interop.Excel` та `System.Math`. Після реалізації інформаційної технології можна перейти до етапу його тестування та аналізу роботи програми, що буде описано у наступному розділі.

3.3 Висновок до розділу 3

В третьому розділі роботи було проведено порівняльний аналіз об'єктно-орієнтованих мов програмування `C++` та `C#`. В результаті аналізу було прийнято рішення використати мову `C#` для подальшої реалізації інформаційної технології.

Також було порівняно переваги та недоліки середовищ розробки `Visual Studio Code` та `Visual Studio 2022`. Після проведеного порівняння було зроблено висновок, що середовище `Visual Studio 2022` є більш зручним для роботи над програмою.

Крім того, було вирішено побудувати інтерфейс системи за допомогою технології `WPF`.

Далі було реалізовано інформаційну технологію прогнозування рентабельності підприємства з використанням бібліотек мови `C#` `Microsoft.Office.Interop.Excel` та `System.Math` і наведено приклади роботи програмного коду.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ

4.1 Тестування роботи програми

Проведемо тестування розробленої програми, проаналізуємо її роботу та оцінимо точність прогнозування нейронною мережею.

Після запуску програми відкриється вікно головного меню, що зображено на рисунку 4.1.

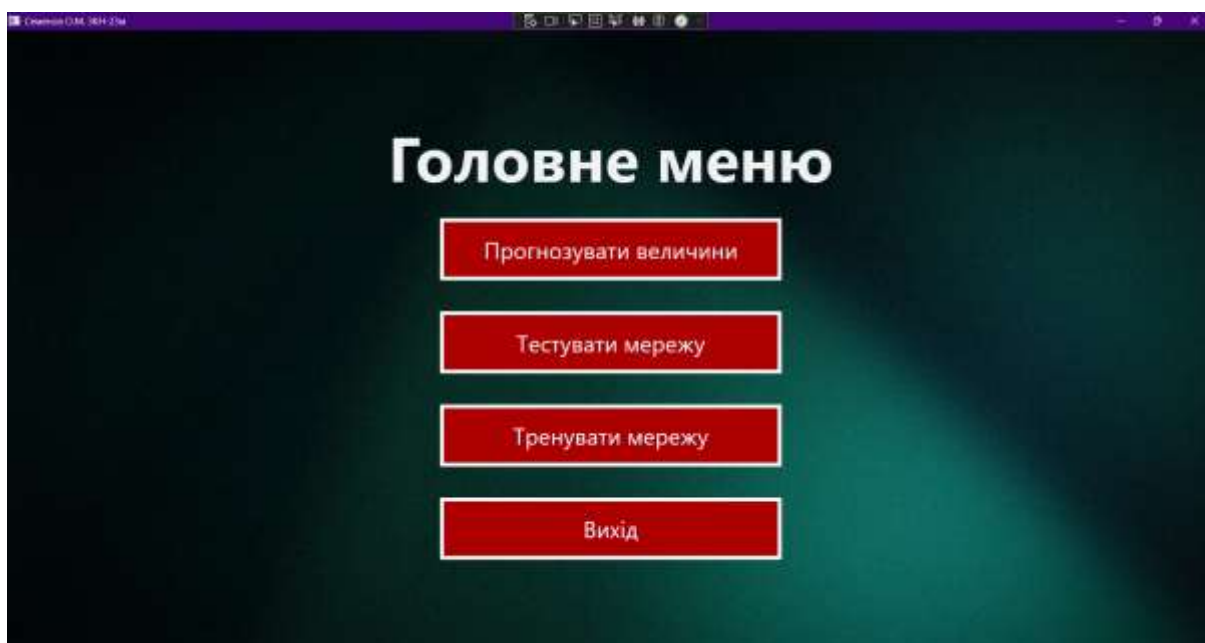


Рисунок 4.1 – Головне меню розробленої програми

На головному меню користувачу надаються 4 опції: розпочати тренування нейронної мережі, протестувати її, прогнозувати значення економічних показників свого підприємства або вийти з програми.

Якщо натиснути на кнопку під назвою «Тренувати мережу», відкриється вікно тренування мережі, яке зображено на рисунку 4.2.

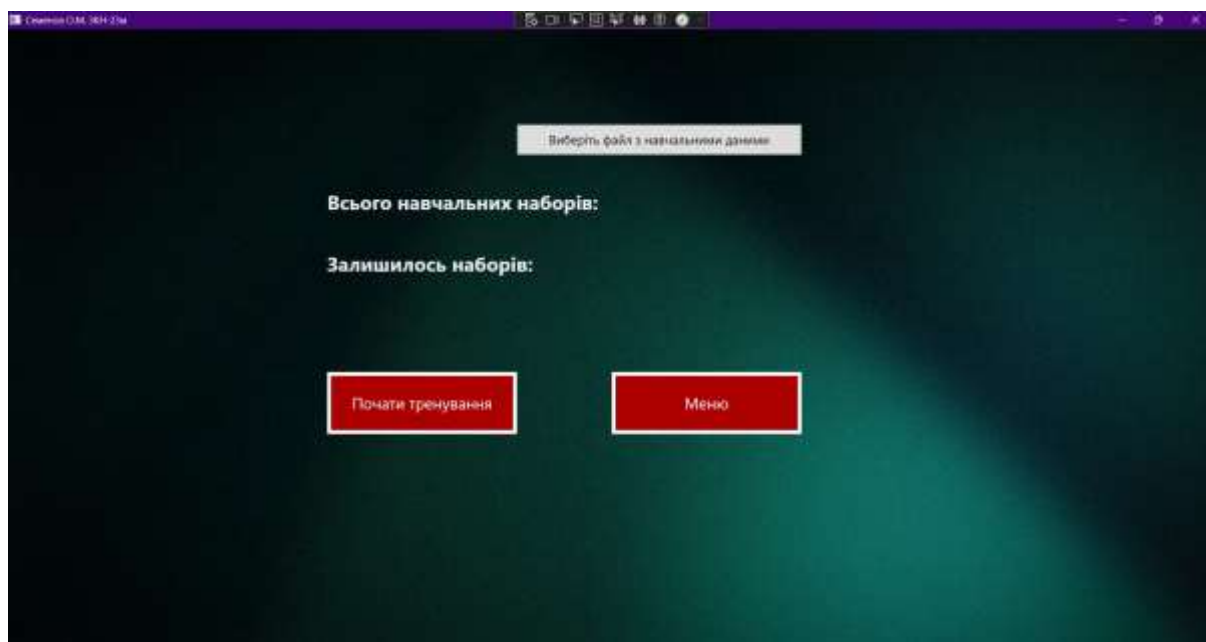


Рисунок 4.2 – Вікно тренування мережі

Щоб розпочати тренування, спочатку користувач має вибрати файл з навчальною вибіркою. Для цього треба натиснути на верхню кнопку під назвою «Виберіть файл з навчальними даними». Після цього відкриється файловий провідник, як зображено на рисунку 4.3. В ньому треба вибрати файл типу .xls або .xlsx з наборами тренувальних даних.

Після вибору файлу система автоматично підрахує кількість наборів даних в ньому та покаже її користувачу. Це зображено на рисунку 4.4.

Далі, щоб почати тренування нейронної мережі треба натиснути на кнопку з відповідною назвою. Біля напису «Залишилось наборів:» буде показано, скільки тренувальних наборів залишилось опрацювати мережі. Коли це число досягне 0, навчання мережі на даному наборі завершено. Вікно програми після навчання зображено на рисунку 4.5.

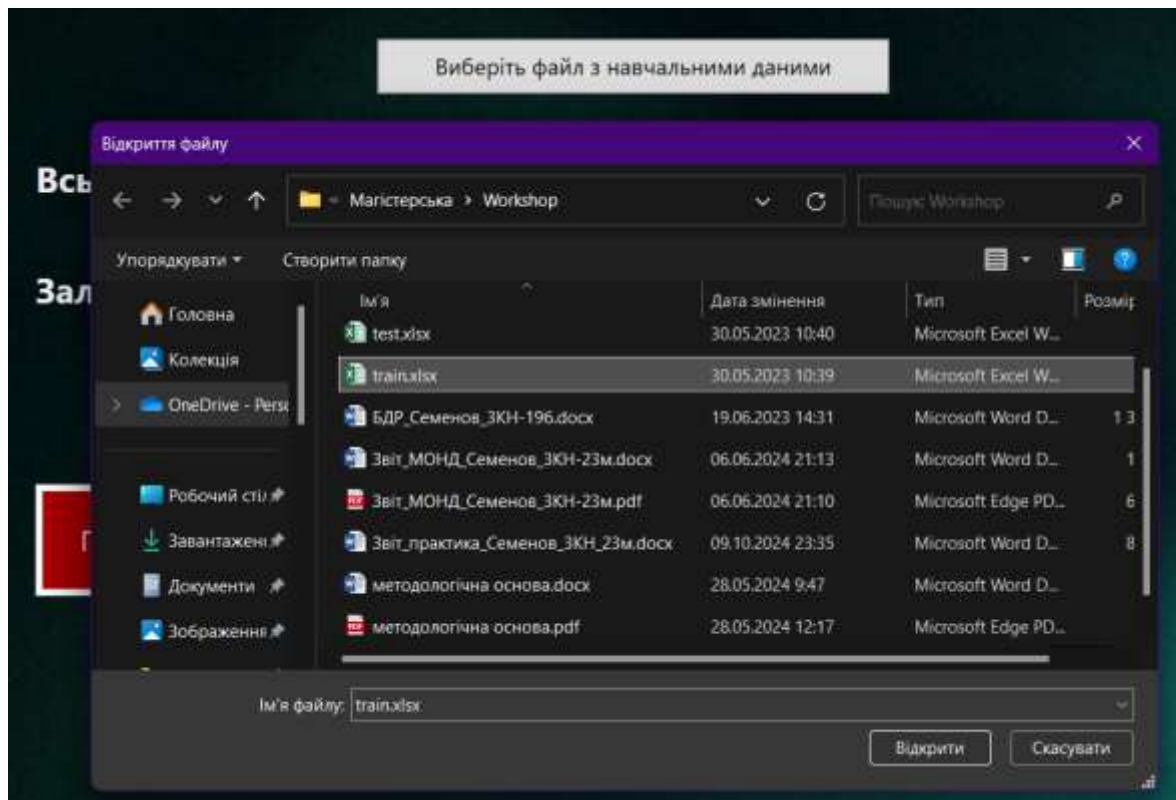


Рисунок 4.3 – Вибір файлу з навчальною вибіркою

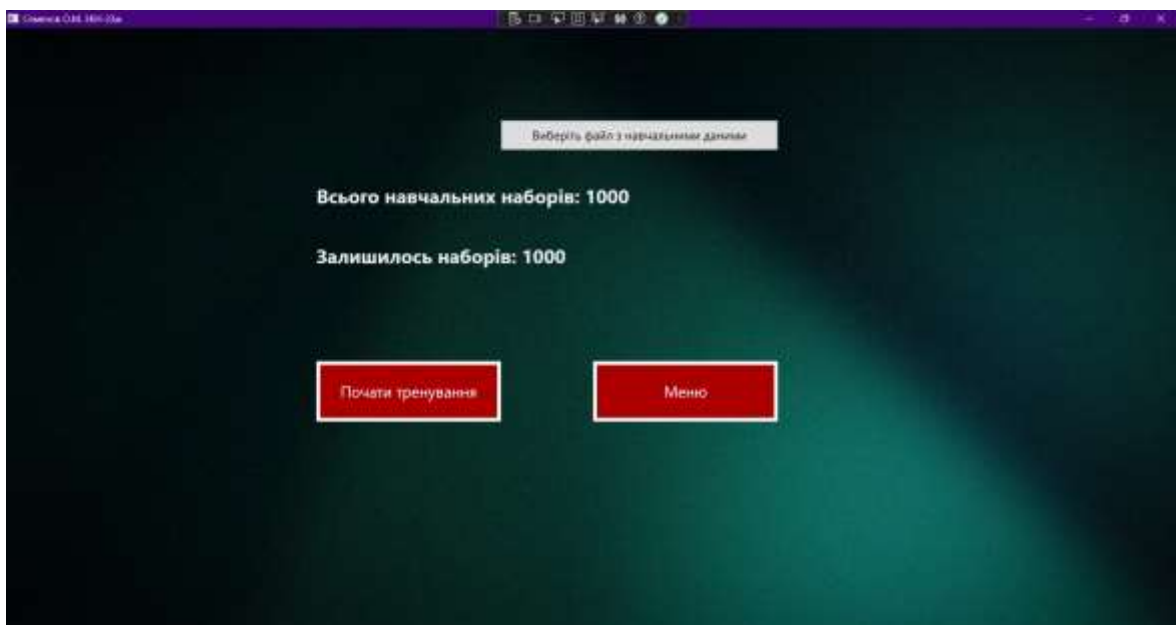


Рисунок 4.4 – Вигляд програми після вибору файлу

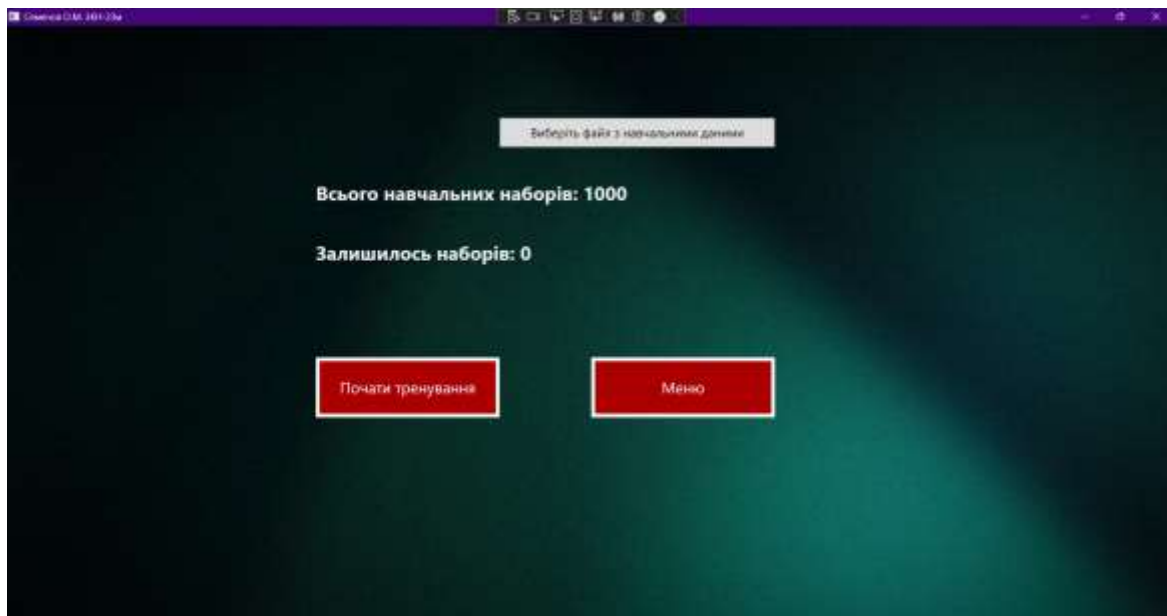


Рисунок 4.5 – Вигляд програми після завершення навчання

Після навчання користувач може повернутись в головне меню відповідною кнопкою. Після натискання кнопки тестування мережі, відкриється вікно, зображене на рисунку 4.6.

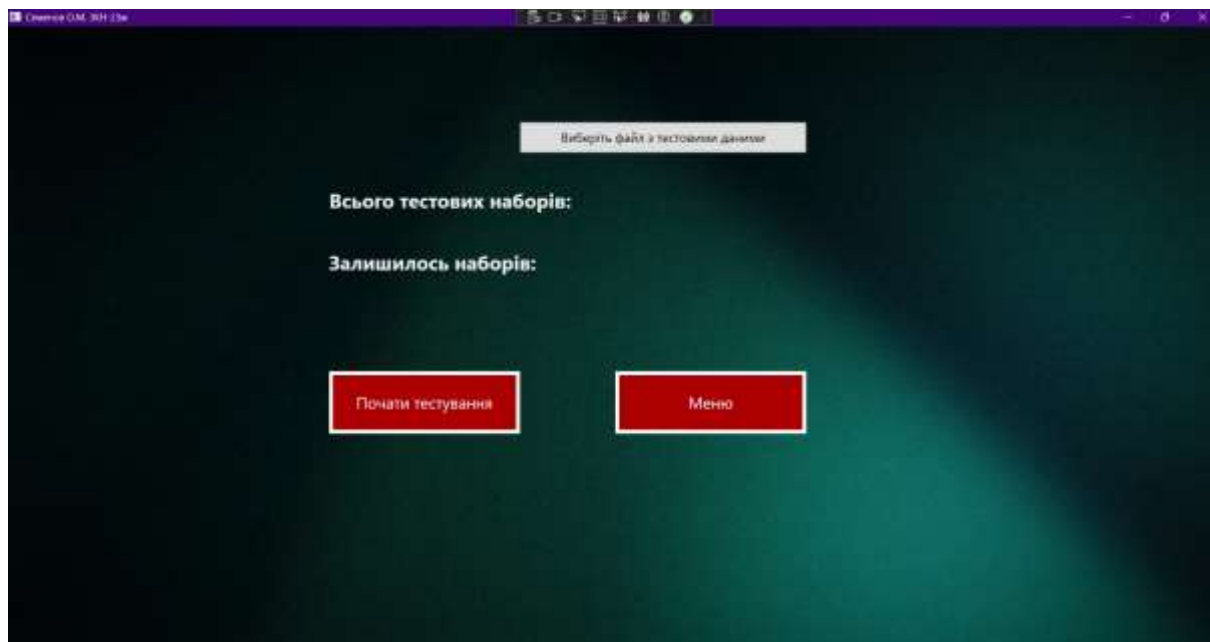


Рисунок 4.6 – Вікно тестування програми

Для тестування треба зробити аналогічні кроки, як у вікні навчання, тільки вибрати тестову вибірку та натиснути на кнопку «Почати тестування». Після тестування відкриється вікно результатів тестування, яке повідомить користувача, скільки було тестових наборів і яка середня похибка прогнозування. Це вікно видно на рисунку 4.7.

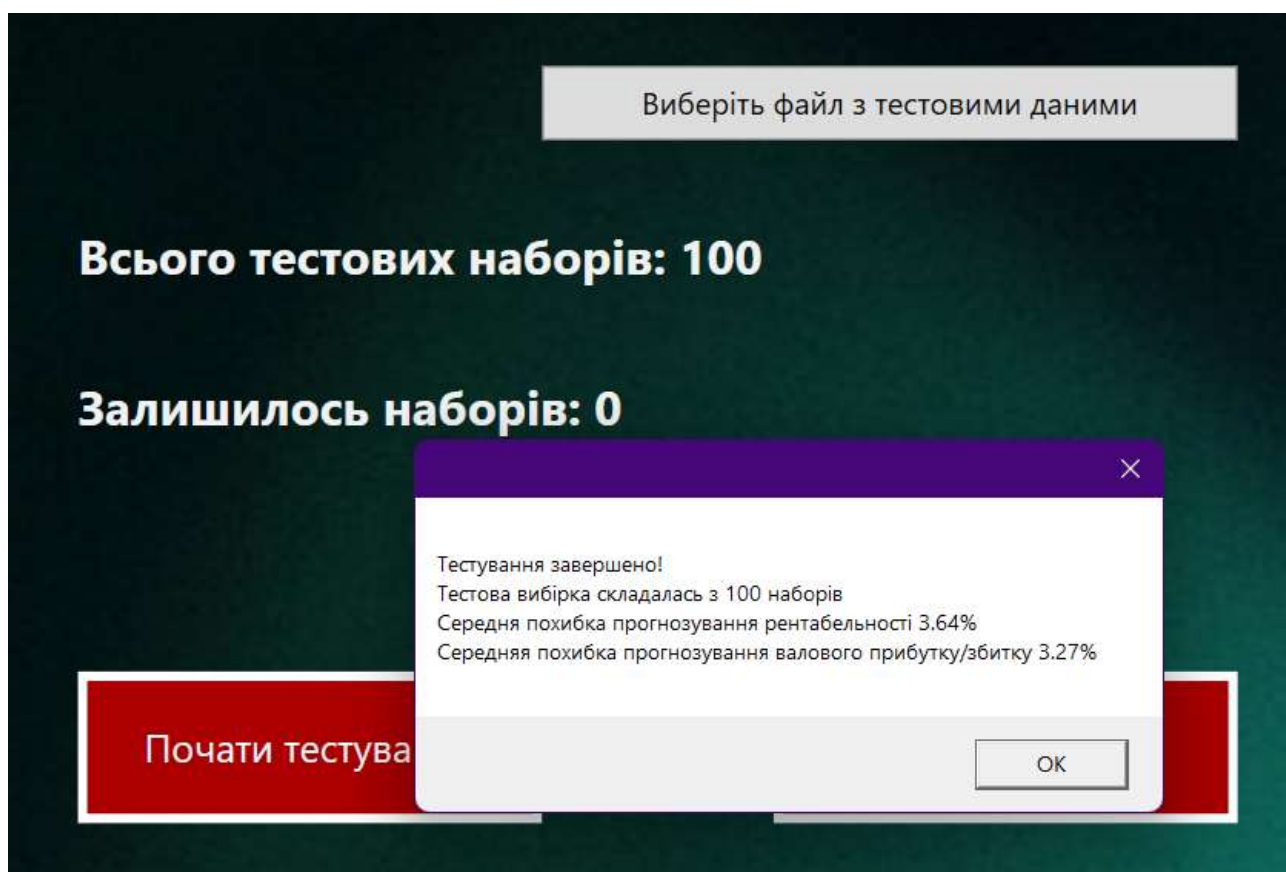


Рисунок 4.7 – Вікно результатів тестування мережі

Після натискання кнопки меню під назвою «Прогнозувати величини», відкриється вікно прогнозування. Для прогнозування треба ввести вхідні дані у відповідні клітинки та натиснути на кнопку «Прогнозувати». Результат зображено на рисунку 4.8.

Введіть дані за останні 3 роки

	Нинішній рік	Минулий рік	Позаминулий рік
Рентабельність:	0.106	0.174	0.341
Дохід:	9154	20316	39461
Собівартість:	85659	116779	115812
Валовий прибуток:	5662	8654	1443
Витрати:	3467	3571	4356
Вартість фондів:	31169	36523	38428
Серед. зарплата:	55.915	48.144	53.246
К-ть працівників:	190	270	271
Кредит. борг:	2936	2937	2447
Дебіт. борг:	0	0	0

Рентабельність: 0.057
Валовий прибуток/збиток: 2765

Прогнозувати
Меню

Рисунок 4.8 – Результат прогнозування рентабельності, валового прибутку та збитку

4.2 Аналіз роботи інформаційної технології

Проаналізуємо роботу розробленої технології та порівняємо точність прогнозування з програмами-аналогами. В другому розділі роботи було визначено навчальну і тестову вибірку для їх обробки багатошаровим перцептроном.

Мережа навчалася на вибірці розміром 1000 наборів, а тестувалася на вибірці розміром 100 наборів. Навчальні та тестові дані були зібрані з офіційного сайту Агенства з розвитку інфраструктури фондового ринку України. Результати тестування розробленого інформаційної технології подані у таблиці 4.1.

Таблиця 4.1 – Результати тестування інформаційної технології

Параметр	Розроблена інформаційна технологія	GSDH Streamline	SAP Integrated Business Planning
Розмір тестової вибірки	100	100	100
Середня похибка прогнозування рентабельності	3.64%	6.57%	5.38%
Середня похибка прогнозування валового прибутку/збитку	3.27%	6.14%	5.03%

Середня похибка прогнозування знаходилась як середнє арифметичне похибок прогнозування для кожного набору тестових даних.

Похибка прогнозування знаходилась діленням модуля різниці цільового значення і результату прогнозування на цільове значення.

Обчислимо, наскільки зменшилась похибка відносно аналогів, за наступної формулою:

$$\Delta E = \frac{E_{\text{аналог}} - E_{\text{розр}}}{E_{\text{аналог}}} \cdot 100\% \quad (4.1)$$

де ΔE – відносне зменшення похибки прогнозування;

$E_{\text{розр}}$ – середня похибка прогнозування розробленої інформаційної технології;

$E_{\text{аналог}}$ – середня похибка прогнозування програми-аналога.

Результати обчислень наведено у таблиці 4.2.

Таблиця 4.2 – Відносні зменшення похибки прогнозування розробленої технології

Параметр, що прогнозувався	GSDH Streamline	SAP Integrated Business Planning
Рентабельність	44.6%	32.34%
Валовий прибуток/збиток	46.74%	34.99%

Отже, за результатами тестування в розробленого інформаційної технології точність прогнозування вийшла вищою, ніж у програм-аналогів.

4.3 Висновок до розділу 4

В ході виконання 4 розділу роботи було проведено тестування розробленої програми. Під час тестування програма запускалась успішно, не виникало помилок та затримок роботи. Елементи інтерфейсу реагували на взаємодію користувача та виконували свою функцію. Файли з тестовими та навчальними даними успішно завантажувались і оброблялися додатком. Програма правильно визначала кількість наборів даних у файлах та повідомляла користувача про кількість наборів, що залишилось обробити.

Після проведення навчання і тестування мережі, модуль тестування інформаційної технології показав середню похибку прогнозування, яка складала 3.64% для рентабельності та 3.27% для валового прибутку/збитку. Після введення вхідних даних та застосування функції прогнозування, технологія видала результат, близький до очікуваного. Відносно кращого аналога похибки прогнозування зменшились на 32.34% для рентабельності та на 34.99% для валового прибутку/збитку.

Отже, можна зробити висновок, що поставлену мету роботи було досягнуто.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Проведення комерційного та технологічного аудиту інформаційної технології прогнозування рентабельності, валового прибутку та валового збитку підприємства

Метою проведення комерційного та технологічного аудиту є визначення науково-технічного рівня й комерційного потенціалу розробки, створеної в межах науково-дослідної діяльності в процесі виконання магістерської кваліфікаційної роботи. Аудит допомагає виявити інноваційність і ступінь новизни технічних рішень, а також їхнє практичне застосування, що є ключовими критеріями оцінки науково-технічного рівня. Особлива увага приділяється рівню наукового освоєння проблеми, над якою працювала дослідницька група, та можливості інтеграції розробки у наявний науковий ландшафт [25].

Комерційний потенціал оцінюється з урахуванням ринкових перспектив, наявності та позицій потенційних конкурентів, стратегій маркетингу й способів просування розробки на ринок. До уваги беруться також економічна обґрунтованість проєкту та стійкість до можливих фінансових і ринкових ризиків, що може підвищити шанси розробки на успішну комерціалізацію. Результати такого аналізу створюють основу для вдосконалення технологій, формування стратегій впровадження та налагодження співпраці з потенційними партнерами [25].

Для проведення аудиту було залучено три незалежні експерти, які є викладачами Вінницького національного технічного університету та мають фахові знання у відповідній галузі. Оцінювання науково-технічного рівня та комерційного потенціалу здійснюється за п'ятибальною шкалою за 12 критеріями, а результати відображаються у таблиці 5.1.

Таблиця 5.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу інформаційної технології прогнозування рентабельності, валового прибутку та збитку

Критерії	Експерти		
	Колесницький О.К.	Озеранський В.С.	Іванчук Я.В.
	Бали виставлені експертами		
Технічна здійсненість концепції	3	3	4
Ринкові переваги (наявність аналогів)	4	3	3
Ринкові переваги (ціна продукту)	4	4	3
Ринкові переваги (технічні властивості)	2	3	3
Ринкові переваги (експлуатаційні витрати)	3	3	2
Ринкові перспективи (розмір ринку)	3	3	4
Ринкові перспективи (конкуренція)	3	2	2
Практична здійсненність (наявність фахівців)	4	3	3
Практична здійсненність (наявність фінансів)	3	4	3
Практична здійсненність (необхідність нових матеріалів)	4	4	4
Практична здійсненність (термін реалізації)	3	2	3
Практична здійсненність (розробка документів)	3	3	4
Сума балів	39	37	38
Середньоарифметична сума балів, СБ	38		

За результатами розрахунків, наведених в табл. 5.1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал інформаційної технології цифрової обробки фотографії – вище середнього.

5.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати на оплату праці. Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, 62 копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Основна заробітна плата дослідників. Витрати на основну заробітну плату дослідників (Z_o) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.1)$$

де k – кількість посад дослідників, залучених до процесу дослідження;

M_{ni} – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

T_p – число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні;

t_i – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 5.2.

Таблиця 5.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу інформаційної технології прогнозування рентабельності, валового прибутку та збитку

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	35000	1166	7	8162
Розробник	25000	833	22	18326
Консультанти	19000	633	3	1899
Всього:	28387			

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників ($З_p$) за відповідними найменуваннями робіт розраховують за формулою:

$$З_p = \sum_{i=1}^n C_i \cdot t_i, \quad (5.2)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год; t_i – час роботи робітника на виконання певної роботи, год. Погодинну тарифну ставку робітника відповідного розряду C і можна визначити за формулою:

$$C_i = \frac{M_m \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (5.3)$$

де M_m – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), у 2024 році $M_m=8000$ грн; K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду; K_c –

мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати; T_p – середня кількість робочих днів в місяці, приблизно $T_p = 21 \dots 23$ дні; $t_{зм}$ – тривалість зміни, год.

Проведені розрахунки зведемо до таблиці 5.3.

Таблиця 5.3 – Витрати на заробітну плату робітників

Найменування робіт	Трудовісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф коєф.	Величина, грн.
Програмування	15	5	73.9	1.7	1108.5
Підготовка тренувальних та тестових даних	50	3	58.7	1.35	2935
Налаштування та тех. обслуговування	8	4	65.2	1.5	521.6
Тестування	5	3	58.7	1.35	293.5
Всього					4858.6

Додаткова заробітна плата. Додаткова заробітна плата $З_d$ всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як $(10 \dots 12)\%$ від суми основної заробітної плати всіх розробників та робітників, тобто:

$$З_d = 0.1 \cdot (З_o + З_p) = 0.1 \cdot (28387 + 4858.6) = 3324.56 \quad (5.4)$$

Відрахування на соціальні заходи. Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$H_{зп} = \beta \cdot (З_д + З_о + З_р) = 0.22 \cdot (28387 + 4858.6 + 3324.56) = 8045.44 \quad (5.5)$$

де $З_о$ – основна заробітна плата розробників, грн.; $З_р$ – основна заробітна плата робітників, грн.; $З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Розрахунок витрат на матеріали. Витрати на матеріали M , що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$M = \sum_{i=1}^n H_i \cdot Ц_i \cdot K_i, \quad (5.6)$$

де H_i – кількість матеріалів i -го виду, шт.; $Ц_i$ – ціна матеріалів i -го виду, грн.; K_i – коефіцієнт транспортних витрат, $K_i = (1, 1 \dots 1, 15)$; n – кількість видів матеріалів. Проведені розрахунки зведемо до таблиці 5.4

Таблиця 5.4 – Матеріали, що використані на розробку

Найменування матеріалів	Ціна за одиницю, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Офісний папір А4	165	1	165
Тонер для принтера	200	1	200
Всього, з врахуванням коефіцієнта транспортних витрат			401.5

Програмне забезпечення. До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10...12% від вартості програмного забезпечення. Балансову вартість програмного забезпечення розраховують за формулою:

$$B_{\text{прг}} = \sum_{i=1}^K C_{\text{прг.}i} \cdot S_{\text{прг.}i} \cdot K_i, \quad (5.7)$$

де $C_{\text{прг.}i}$ – ціна придбання програмного забезпечення i -го виду, грн.; $S_{\text{прг.}i}$ – кількість одиниць програмного забезпечення відповідного виду, шт.; K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного забезпечення, $K_i = (1, 1 \dots 1, 12)$; k – кількість видів програмного забезпечення.

Проведені розрахунки зведемо до таблиці 5.5.

Таблиця 5.5 – Витрати на придбання програмного забезпечення

Найменування програмного забезпечення	Ціна за одиницю, грн.	Витрачено	Вартість програмного забезпечення, грн.
Visual Studio 2022 Pro	1235.1	1	1235.1
Microsoft 365	259	1	259
Всього, з урахуванням коефіцієнта інсталяції та налагодження			1653.51

Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи. У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою:

$$A = \frac{Ц_6}{T_в} \cdot \frac{t}{12}, \quad (5.8)$$

де $Ц_6$ – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; t – термін використання основного фонду, місяці; $T_в$ – термін корисного використання основного фонду, роки. Проведені розрахунки зведемо до таблиці 5.6.

Таблиця 5.6 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук Lenovo Ideapad Gaming 3	36000	2	2	3000
Принтер Pantum M6500	7218	2	2	601.5
Всього	3601.5			

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%} = (28387 + 4858.6) \cdot \frac{50}{100} = 16622.8, \quad (5.9)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат:

$$B_{\text{заг}} = Z_o + Z_p + Z_d + H_{\text{зп}} + K_v + B_{\text{прг}} + A_{\text{обл}} + I_v = 28387 + 4858.6 + 3324.56 + 8045.44 + 401.5 + 1653.51 + 3601.5 + 16622.8 = 66894.91 \text{ грн. (5.10)}$$

Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ZB = \frac{B_{\text{заг}}}{\eta} = \frac{66894.91}{0.9} = 74327.67 \text{ грн.,} \quad (5.11)$$

де η – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії впровадження, то $\eta=0,9$.

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В умовах ринкової економіки узагальненим позитивним результатом, який може отримати потенційний інвестор від впровадження результатів певної науково-технічної розробки, є зростання чистого прибутку.

У даному контексті, коли йдеться про розробку інструменту, основу майбутнього економічного ефекту формують такі показники:

- 1) ΔN — приріст кількості споживачів, які отримують відповідну інформаційну послугу у проаналізовані періоди;
- 2) N — кількість споживачів, яким була надана відповідна інформаційна послуга в році, що передував впровадженню результатів розробки;

- 3) Π_0 — вартість послуги у базовому році (до впровадження інформаційної системи);
- 4) $\pm \Delta \Pi_0$ — зміна вартості послуги (підвищення або зниження), спричинена впровадженням науково-технічної розробки у відповідні періоди.

Приріст чистого прибутку потенційного інвестора ($\Delta \Pi$) для кожного року, протягом якого очікується отримання позитивного економічного ефекту від впровадження і комерціалізації науково-технічної розробки, обчислюється за допомогою формули:

$$\Delta \Pi = (\pm \Delta \Pi_0 \cdot N + \Pi_0 \cdot \Delta N_i)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.12)$$

де $\pm \Delta \Pi_0$ — зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки);

N — основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки;

Π_0 — основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році;

Π_0 — основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів;

ΔN — зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки);

λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість.

У 2024 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$; ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho = 0,2 \dots 0,5$; θ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\theta = 25\%$.

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\begin{aligned} \Delta\P &= ((3000 - 2500) \cdot 1000 - (1500 - 1500) \cdot 2500) \cdot 0.8333 \cdot 0.3 \cdot \left(1 - \frac{25}{100}\right) \\ &= 93746.25 \text{ грн.} \end{aligned}$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\P_i}{(1+\tau)^t} = \frac{93746.25}{(1+0.1)^1} = 85223.86 \text{ грн.}, \quad (5.13)$$

де $\Delta\P$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо $T=1$ рік);

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках) від моменту початку впровадження науковотехнічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot 3B = 1 \cdot 74327.67 = 74327.67 \text{ грн.} \quad (5.14)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо. Зазвичай $k_{\text{інв}} = 1 \dots 5$, але може бути і більшим;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV = 85223.86 - 74327.67 = 10896.19, \quad (5.15)$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.;

PV – теперішня вартість початкових інвестицій, грн.

Оскільки $E_{\text{абс}} > 0$, то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_v або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій E_B , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_B = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} = \sqrt[1]{1 + \frac{10896.19}{74327.67}} = 1.15, \quad (5.16)$$

де $T_{\text{ж}}$ – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування $\tau_{\text{мін}}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_o = \frac{1}{E_B} = \frac{1}{1.15} = 0.87 \text{ року.} \quad (5.17)$$

Оскільки $T_o = 0.87 < 1 \dots 3$ -років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

5.4 Висновок до розділу 5

В ході виконання економічної частини було проведено комерційний та технологічний аудит інформаційної технології прогнозування рентабельності, валового прибутку та валового збитку підприємства. Під час оцінювання

науково-технічного рівня і комерційного потенціалу розробці було надано 38 балів, що відповідає комерційному потенціалу вище середнього.

Далі було розраховано витрати на здійснення науково-дослідної роботи. В результаті проведених обчислень було визначено загальні витрати на завершення науково-дослідної роботи та оформлення її результатів рівні 74327.67 грн.

Також було розраховано економічну ефективність науково-технічної розробки за її можливої комерціалізації потенційним інвестором. За проведеними розрахунками термін окупності становить 0.87 року, що менше на 3 роки та свідчить про комерційну привабливість даної розробки для фінансування потенційним інвестором її розробки та виведення на ринок.

Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи на тему «Інформаційна технологія прогнозування рентабельності, валового прибутку та валового збитку підприємства».

ВИСНОВКИ

В ході виконання магістерської роботи було розроблено інформаційну технологію прогнозування рентабельності, валового прибутку та валового збитку підприємства.

Було проведено аналіз проблеми розв'язання задачі прогнозування рентабельності, валового прибутку та валового збитку підприємства. В ході аналізу було виконано постановку задачі прогнозування цих фінансових показників та здійснено порівняльний аналіз методів прогнозування. Під час аналізу були розглянуті метод часових рядів, метод програмних агентів, генетичні алгоритми та метод нейронних мереж. За результатами аналізу було прийнято рішення використовувати нейронні мережі для розв'язання поставленої задачі. Також було проаналізовано існуючі програми-аналоги на прикладі GMDH Streamline та SAP Integrated Business Planning. Ці програми популярні в досліджуваній сфері і надають широкий асортимент функцій, проте у них не дуже висока точність прогнозування величин, що піддаються впливу великої кількості факторів. Таким чином була сформована мета підвищення точності прогнозування рентабельності, валового прибутку та валового збитку підприємства. Для досягнення цієї мети було розширено набір економічних характеристик, використовуваних для прогнозування цих величин.

Далі було проведено порівняльний аналіз різних видів нейронних мереж. Досліджені типи мереж включали рекурентну мережу, згорткову мережу, РБФ мережу та багатошаровий перцептрон. В результаті цього аналізу було вирішено використати багатошаровий перцептрон. Після цього було розроблено структуру цієї мережі та наведено її у графічній формі. Також проведено дослідження навчання нейронних мереж з учителем та без учителя, внаслідок чого обрано метод зворотного поширення помилки та функцію активації ELU як оптимальні для організації навчання мережі. Були наведені формули, за якими працює метод зворотного поширення помилки та графік функції активації. Далі була сформована навчальна вибірка розміром в 1000

наборів даних. Розроблено структуру та алгоритм інформаційної технології. Алгоритм було наведено в описовій формі та у вигляді блок-схеми. Після цього розроблено UML-діаграму класів та описано класи та зв'язки, що на ній зображені.

Після цього було проведено порівняльний аналіз мов та середовищ програмування, внаслідок якого було вирішено реалізовувати інформаційну технологію мовою C# в середовищі програмування Visual Studio 2022 з використанням графічної підсистеми WPF. Тоді було реалізовано програму та наведено приклади реалізації деяких класів та методів.

Тоді було проведено тестування розробленої технології на тестовій вибірці розміром 100 наборів тестових даних. В процесі тестування виявлено, що програма працює без помилок та затримок і має середню похибку прогнозування рентабельності 3.64%, та прибутку/збитку 3.27%.

В результаті тестування програми-аналога на даних тих же підприємств за такий же період часу середня похибка прогнозування рентабельності склала 6.57% та 5.38%, що відповідно на 2.93% та 3.3% більше, ніж у розробленої технології. Похибки прогнозування валового прибутку/збитку в програм-аналогів складають 6.14% та 5.03%, що перевищує похибку програми-аналога на 2.87% та 1.76% відповідно. Таким чином, відносно кращого аналога похибку прогнозування рентабельності було зменшено на 32.34%, а валового прибутку/збитку – на 34.99%.

Під час оцінювання науково-технічного рівня і комерційного потенціалу розробці було надано 38 балів, що відповідає комерційному потенціалу вище середнього.

За проведеними розрахунками термін окупності становить 0.87 року, що менше на 3 роки та свідчить про комерційну привабливість даної розробки для фінансування потенційним інвестором її розробки та виведення на ринок.

Отже, усі поставлені задачі було виконано та мету роботи було досягнуто.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Семенов О.М., Колесницький О. К. Аналіз видів нейронних мереж для прогнозування рентабельності підприємства. *КОНФЕРЕНЦІЇ ВНТУ електронні наукові видання, «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)»* [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/viewFile/21711/17912>
2. McCleary R., McDowall D., Bartos B. J. Forecasting. Oxford University Press, 2017. 185 p.
3. Elliott G., Timmermann A. Economic Forecasting. Princeton University Press, 2016. 568 p.
4. Ганчук А. А., Соловійов В. М., Чабаненко Д. М. Методи прогнозування. Брама-Україна, 2012. 140 с.
5. Sanders N. Forecasting Fundamentals. Business Expert Press, 2016. 122 p.
6. Бойко Н., Качмарик В. Побудова моделей для прогнозування часових рядів застосовуючи мережі довгострокової пам'яті. Науковий вісник Ужгородського університету. Серія: Математика і інформатика, 2022. Т. 40, № 1. 125 с.
7. Hamilton J. D. Time Series Analysis. Princeton University Press, 2020. 816 p.
8. Blokdyk G. Software Agents a Complete Guide - 2020 Edition. Emereo Pty Limited, 2020. 304 p.
9. Multi-Agent Systems / ed. by A. Rosenfeld, N. Talmon. Cham : Springer International Publishing, 2021. 281 p.
10. Hugo Van H., Alain V. Genetic Algorithms and Recognition Problems. Genetic Algorithms for Pattern Recognition, 2017. 166 p.
11. Amit K. Genetic Algorithms. Artificial Intelligence and Soft Computing, 2018. 475 p.

12. Галкін О. В. Нейронні мережі в економічному моделюванні. Комп'ютерна математика, 2013. 74 с.
13. 5 найкращих програм для бізнес-прогнозування у 2024 році [Електронний ресурс] – Режим доступу: <https://gmdhsoftware.com/ua/business-forecasting-software/>
14. SAP Integrated Business Planning for Supply Chain [Електронний ресурс] – Режим доступу: <https://www.sap.com/products/scm/integrated-business-planning.html>
15. Ventriglia F. Neural Modeling and Neural Networks. Elsevier Science & Technology Books, 2013. 526 p.
16. Artificial Neural Network Modelling / ed. by S. Shanmuganathan, S. Samarasinghe. Cham : Springer International Publishing, 2016. 472 p.
17. Doroshenko A. Y., R.V. K. Recurrent neural networks for the problem of improving numerical meteorological forecasts. Problems in programming, 2023. No. 4. 97 p.
18. Bianconi G. Multilayer Network Models. Oxford University Press, 2018. 225 p.
19. Manaswi N. K. Multilayer Perceptron. Deep Learning with Applications Using Python. Berkeley, CA, 2018. 56 p.
20. Іванчук Я. В., Месюра В. І., Яровий А. А., Манжілевський О. Д. Інтелектуальний аналіз даних та машинне навчання. Частина 1. Базові методи та засоби аналізу даних : навчальний посібник, Вінниця : ВНТУ, 2021. – 69 с.
21. Rumpe B. Modeling with UML. Cham : Springer International Publishing, 2016. 281 p.
22. C# vs C++: порівняння їх характеристик та ефективності [Електронний ресурс] – Режим доступу: <https://foxminded.ua/c-sharp-vs-c-plus-plus/>

23. Visual Studio vs Visual Studio Code – What's The Difference Between These IDE Code Editors? [Електронний ресурс] – Режим доступу: <https://www.freecodecamp.org/news/visual-studio-vs-visual-studio-code/>
24. Windows Presentation Foundation documentation [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/>
25. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки / уклад.: А. А. Яровий, О. К. Колесницький. – Вінниця : ВНТУ, 2023. – 58 с.
26. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с

Додаток А (обов'язковий)
Протокол перевірки кваліфікаційної роботи на наявність текстових
запозичень

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія прогнозування рентабельності
валового прибутку та валового збитку підприємства»

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Turnitin

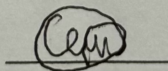
Оригінальність 87,00% Схожість 13,00%

Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

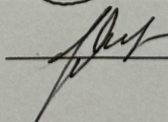
Ознайомлені з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи



Семенов О.М.

Керівник роботи

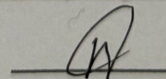


Колесницький О.К.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)**Лістинг програми**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using static System.Net.Mime.MediaTypeNames;

namespace Magisterska
{
    /// <summary>
    /// Interaction logic for Menu.xaml
    /// </summary>
    public partial class Menu : Page
    {
        public Menu()
        {
            InitializeComponent();

        }

        NeuralNetwork network = new NeuralNetwork([30, 16, 9]);

        private void btnPredict_Click(object sender, RoutedEventArgs e)
```



```

        {
            NavigationService.Navigate(new Predict(network));
        }

private void btnTest_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new Test(network));
}

private void btnTrain_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new Train(network));
}

private void btnExit_Click(object sender, RoutedEventArgs e)
{
    System.Windows.Application.Current.Shutdown();
}
    }
}

```

```

using System;
using System.Collections.Generic;

namespace Magisterska
{
    public class NeuralNetwork
    {
        public int[] layers;
        public double[][] neurons;
        public double[][] bias;
        public double[][][] weights;
        public double[][][] gradient;
        public double learnRate;
    }
}

```

```

private double error;

private readonly Random _random = new Random();

public NeuralNetwork(int[] layerSize, double learnRate = 0.1)
{
    this.learnRate = learnRate;
    InitLayers(layerSize);
    InitNeurons();
    Initbias();
    InitWeights();
    InitWeightAdjustments();
}

public NeuralNetwork(int[] layerSize, double[][][] weights, double
learnRate = 0.1)
{
    this.learnRate = learnRate;
    this.weights = weights;
    InitLayers(layerSize);
    InitNeurons();
    Initbias();
    InitWeightAdjustments();
}

private void InitLayers(int[] layerSize)
{
    layers = new int[layerSize.Length];
    Array.Copy(layerSize, layers, layerSize.Length);
}

private void InitWeights()
{
    weights = new double[layers.Length][][];
    for (int layer = 1; layer < layers.Length; layer++)

```

```

{
    weights[layer] = new double[neurons[layer].Length][];
    for (int neuron = 0; neuron < neurons[layer].Length; neuron++)
    {
        int prevLayerNeurons = neurons[layer - 1].Length;
        weights[layer][neuron] = new double[prevLayerNeurons];
        for (int weight = 0; weight < prevLayerNeurons; weight++)
        {
            weights[layer][neuron][weight] = SmallRandNumber();
        }
    }
}

private void InitWeightAdjustments()
{
    gradient = new double[layers.Length][][];
    for (int layer = 1; layer < layers.Length; layer++)
    {
        gradient[layer] = new double[neurons[layer].Length][];
        for (int neuron = 0; neuron < neurons[layer].Length; neuron++)
        {
            int prevLayerNeurons = neurons[layer - 1].Length;
            gradient[layer][neuron] = new double[prevLayerNeurons + 1];
        }
    }
}

private void InitNeurons()
{
    neurons = new double[layers.Length][];
    for (int layer = 0; layer < layers.Length; layer++)
    {
        neurons[layer] = new double[layers[layer]];
    }
}

```

```

    }

    private void Initbias()
    {
        bias = new double[layers.Length][];
        for (int layer = 1; layer < layers.Length; layer++)
        {
            bias[layer] = new double[neurons[layer].Length];
            for (int neuron = 0; neuron < bias[layer].Length; neuron++)
            {
                bias[layer][neuron] = SmallRandNumber();
            }
        }
    }

    private double SmallRandNumber() =>
        (2.4999 * _random.NextDouble() + 0.0001) * (_random.Next(2) == 0 ? -
1 : 1);

    public double[] FeedForward(List<double> input)
    {
        for (int i = 0; i < input.Count; i++)
        {
            neurons[0][i] = input[i];
        }

        for (int layer = 1; layer < layers.Length; layer++)
        {
            for (int neuron = 0; neuron < neurons[layer].Length; neuron++)
            {
                double sum = 0;
                for (int weight = 0; weight < weights[layer][neuron].Length;
weight++)
                {
                    double weightValue = weights[layer][neuron][weight];

```

```

        double neuronValue = neurons[layer - 1][weight];
        sum += weightValue * neuronValue;
    }
    sum += bias[layer][neuron];
    neurons[layer][neuron] = ActivationFunction(sum);
}

}

int outputLayer = neurons.Length - 1;
return new double[] { neurons[outputLayer][0],
neurons[outputLayer][1] };
}

public void Train(List<double> input, List<double> desiredOutputs)
{
    double[] predictions = FeedForward(input);
    Backpropagate(predictions, desiredOutputs);
}

public double CalcError(double[] predictions, List<double>
desiredOutputs)
{
    double totalError = 0;
    for (int i = 0; i < predictions.Length; i++)
    {
        totalError += System.Math.Abs(predictions[i] -
desiredOutputs[i]) / desiredOutputs[i];
    }
    return totalError / predictions.Length;
}

private void Backpropagate(double[] generatedOutputs, List<double>
targetOutputs)
{
    error = CalcError(generatedOutputs, targetOutputs);
    CalcOutputLayerWeightGradient(generatedOutputs, targetOutputs);
}

```

```

        CalcHiddenLayerWeightGradient();
        UpdateWeights();
    }

    public void CalcOutputLayerWeightGradient(double[] generatedOutputs,
        List<double> targetOutputs)
    {
        int outputLayer = layers.Length - 1;
        for (int neuron = 0; neuron < neurons[outputLayer].Length; neuron++)
        {
            double dAdZ =
            ActivationFunctionDerivative(generatedOutputs[neuron]);

            double dCdA = CostFunctionDerivative(generatedOutputs[neuron],
            targetOutputs[neuron]);

            error = dAdZ * dCdA;

            for (int weight = 0; weight <
            weights[outputLayer][neuron].Length; weight++)
            {
                gradient[outputLayer][neuron][weight] = dCdA * dAdZ *
            neurons[outputLayer - 1][weight];
            }

            double dZdW = 1;
            gradient[outputLayer][neuron][^1] = dCdA * dAdZ * dZdW;
        }
    }

    private void CalcHiddenLayerWeightGradient()
    {
        for (int layer = layers.Length - 2; layer > 0; layer--)
        {
            for (int neuron = 0; neuron < neurons[layer].Length; neuron++)
            {
                double dCdA = Calc_dCdA(layer, neuron);

                double dAdZ =
            ActivationFunctionDerivative(neurons[layer][neuron]);

```

```

        error = dCdA * dAdZ;

        for (int weight = 0; weight < weights[layer][neuron].Length;
weight++)
        {
            gradient[layer][neuron][weight] = dCdA * dAdZ *
neurons[layer - 1][weight];
        }

        double dZdW = 1;
        gradient[layer][neuron][^1] = dCdA * dAdZ * dZdW;
    }
}

private void UpdateWeights()
{
    for (int layer = 1; layer < layers.Length; layer++)
    {
        for (int neuron = 0; neuron < neurons[layer].Length; neuron++)
        {
            for (int weight = 0; weight < weights[layer][neuron].Length;
weight++)
            {
                weights[layer][neuron][weight] -=
gradient[layer][neuron][weight] * learnRate;
            }
            bias[layer][neuron] -= gradient[layer][neuron][^1] *
learnRate;
        }
    }
}

public double CostFunctionDerivative(double prediction, double
targetOutput)
{
    return -(targetOutput - prediction);
}

```

```

}

public double Calc_dCdA(int layer, int neuron)
{
    double dCdA = 0;
    for (int weight = 0; weight < neurons[layer + 1].Length; weight++)
    {
        dCdA += error * weights[layer + 1][weight][neuron];
    }
    return dCdA;
}

double alpha = 1;
public double ActivationFunction(double x)
{
    if (x > 0)
    {
        return x;
    }
    else
    {
        return alpha * (Math.Exp(x) - 1);
    }
}

public double ActivationFunctionDerivative(double x)
{
    if (x > 0)
    {
        return 1;
    }
    else
    {
        return alpha * Math.Exp(x);
    }
}

```



```

        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.IO;
namespace Magisterska
{
    /// <summary>
    /// Interaction logic for Predict.xaml
    /// </summary>
    public partial class Predict : Page
    {
        public Predict(NeuralNetwork network)
        {
            InitializeComponent();

            if (!File.Exists("~/weights.txt"))
            {
                network = new NeuralNetwork(layerSizes);

                File.WriteAllText("~/weights.txt",
                    Converter.WeightsToText(network.weights));
            }
        }
    }
}

```

```

        else
        {
            network = new NeuralNetwork(layerSizes,
Converter.TextToWeights(File.ReadAllText("~/weights.txt")));
        }
    }

    int[] layerSizes = { 24, 13, 7 };
    NeuralNetwork network;
    List<double> input;
    bool isEmpty = true;
    double res;
    double rent1, rent2, rent3;

    private void btnCalc_Click(object sender, RoutedEventArgs e)
    {
        foreach (TextBox el in MainGrid.Children)
        {
            input.Add(Converter.Normalize(Convert.ToDouble(el.Text)));
        }

        tbRes.Text = "Результат: " +
Convert.ToString(network.FeedForward(input) [0]);
    }

    private void btnMenu_Click(object sender, RoutedEventArgs e)
    {
        NavigationService.Navigate(new Menu());
    }
}
}

```

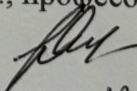
Додаток В (обов'язковий)**ІЛЮСТРАТИВНА ЧАСТИНА****«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ
РЕНТАБЕЛЬНОСТІ, ВАЛОВОГО ПРИБУТКУ ТА ВАЛОВОГО
ЗБИТКУ ПІДПРИЄМСТВА»**

Виконав: студент 2 курсу, групи 3КН-23м
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)



Семенов О.М.
(прізвище та ініціали)

Керівник: к.т.н., професор каф.КН



Колесницький О.К.
(прізвище та ініціали)

« 05 »

12

2024 р.

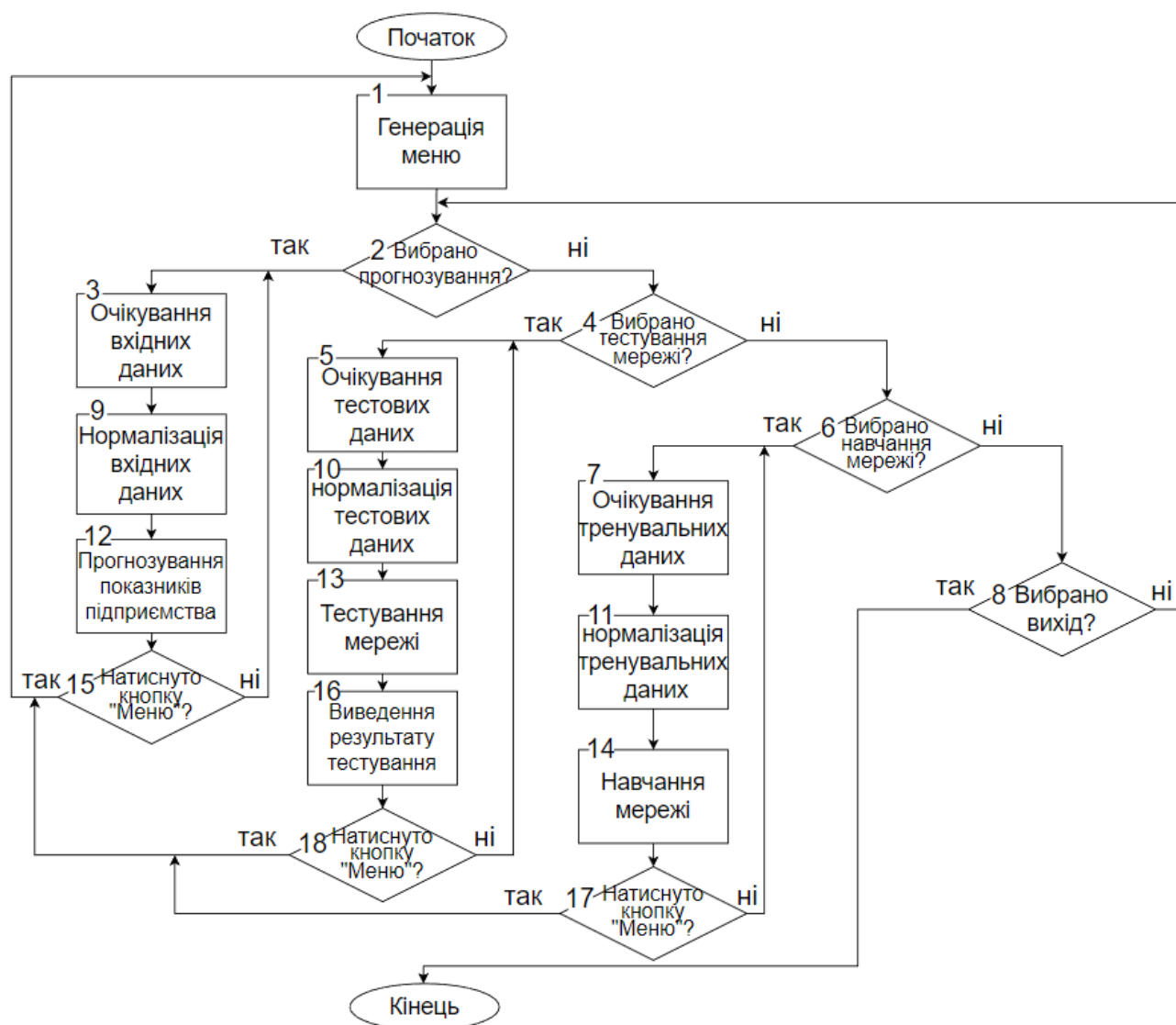


Рисунок В.1 – Блок-схема алгоритму роботи інформаційної технології прогнозування рентабельності, валового прибутку та валового збитку підприємства

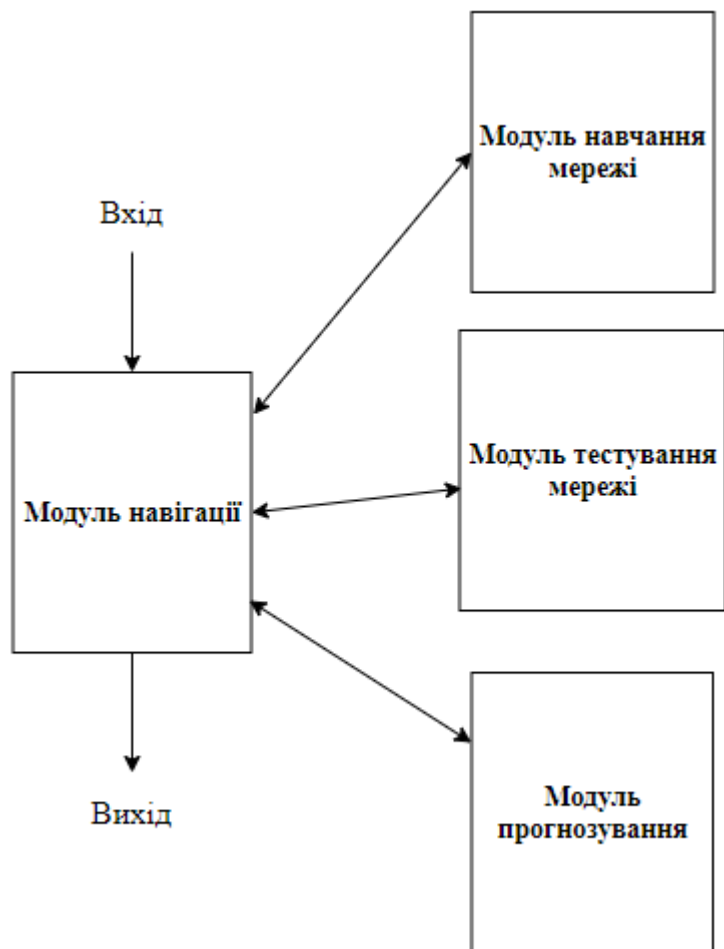


Рисунок В.2 – Структура інформаційної технології прогнозування рентабельності, валового прибутку та валового збитку підприємства

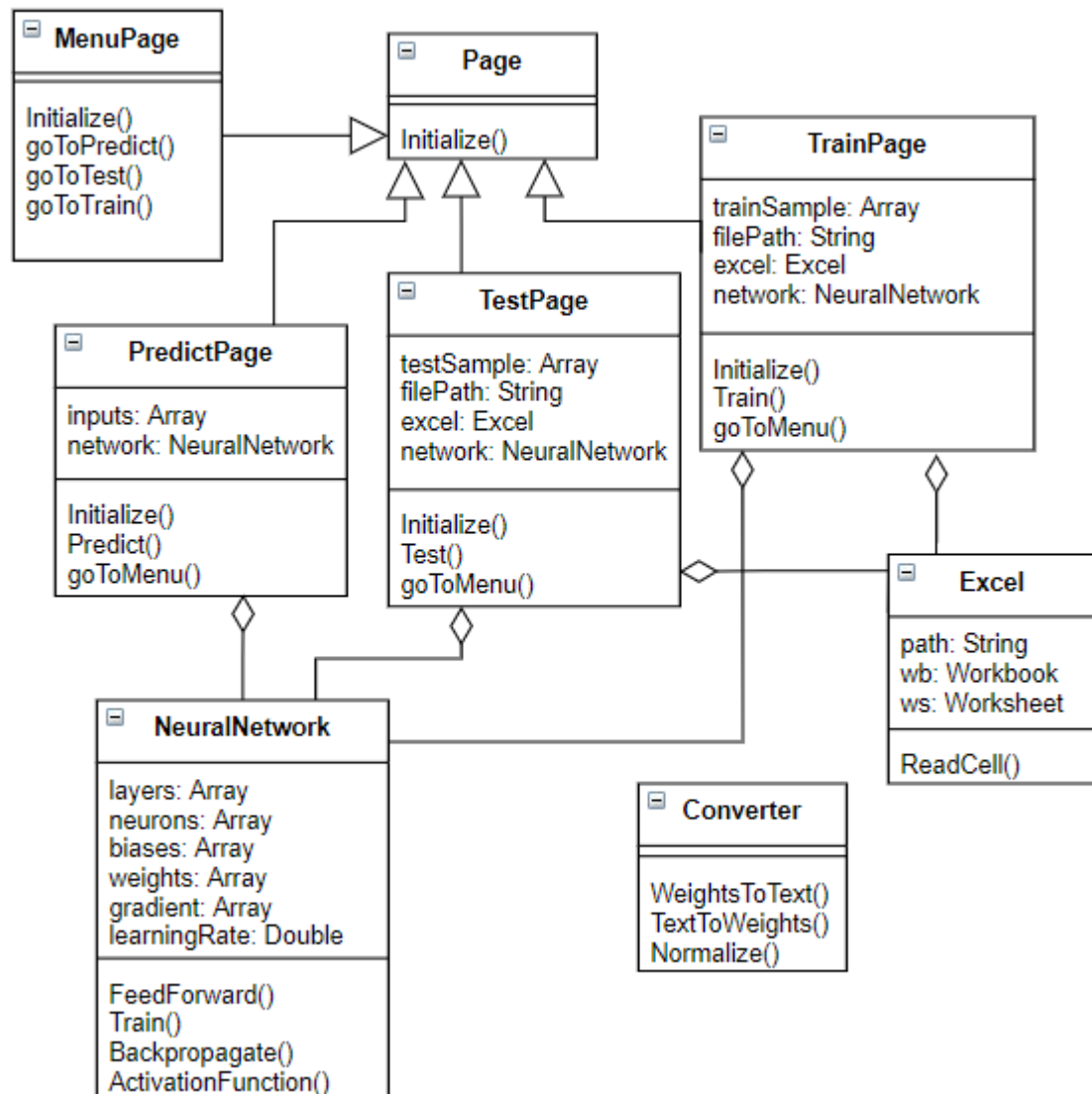


Рисунок В.3 – UML-діаграма класів інформаційної технології прогнозування рентабельності, валового прибутку та валового збитку підприємства

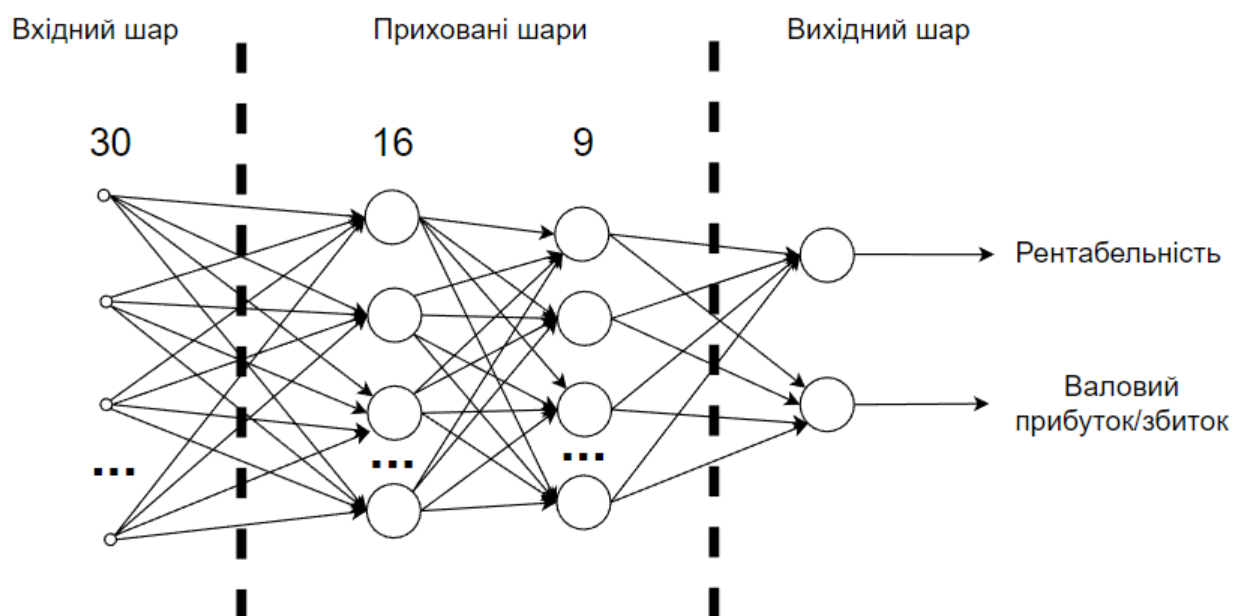


Рисунок В.4 Структура нейронної мережі для прогнозування рентабельності, валового прибутку та валового збитку підприємства

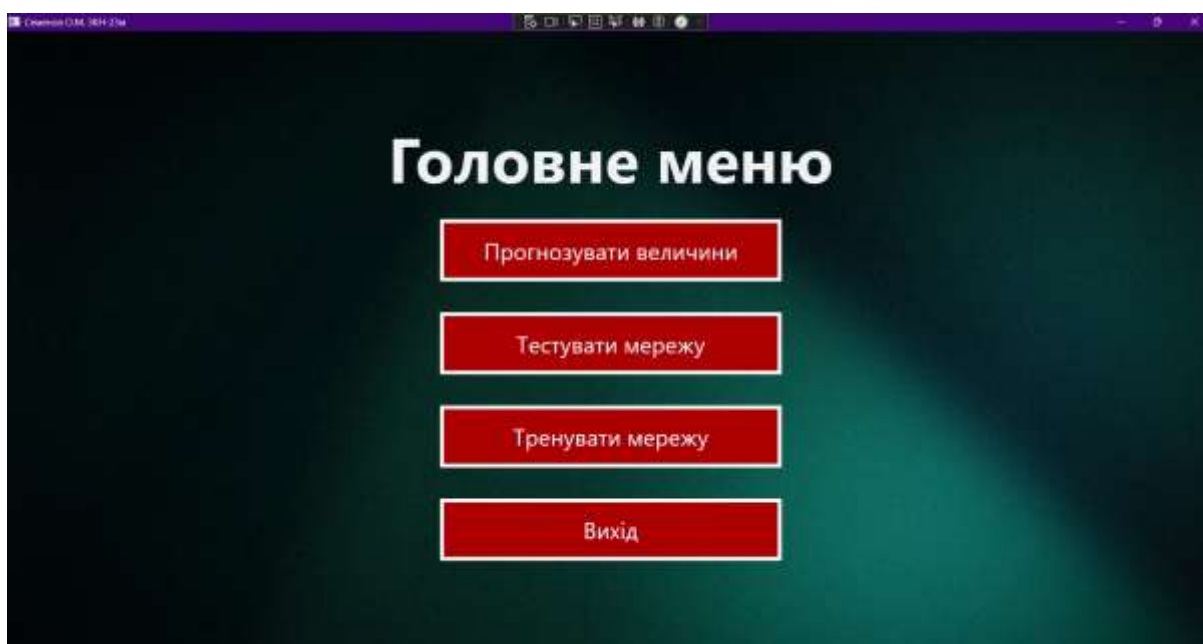


Рисунок В.5 – Головне меню розробленої програми

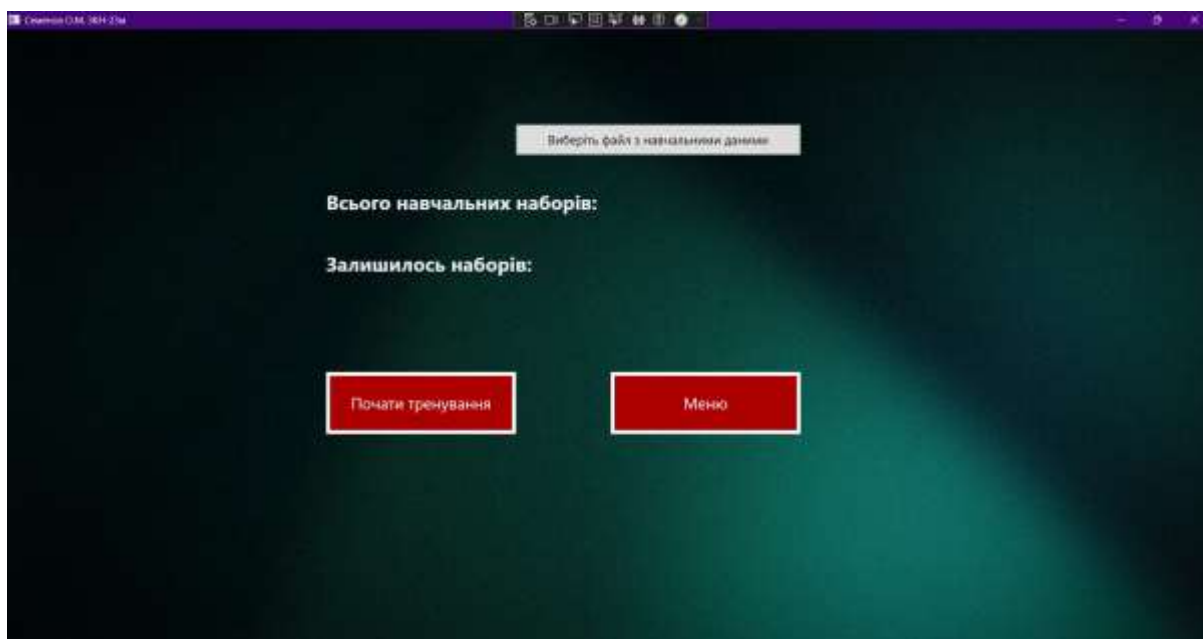


Рисунок В.6 – Вікно тренування мережі

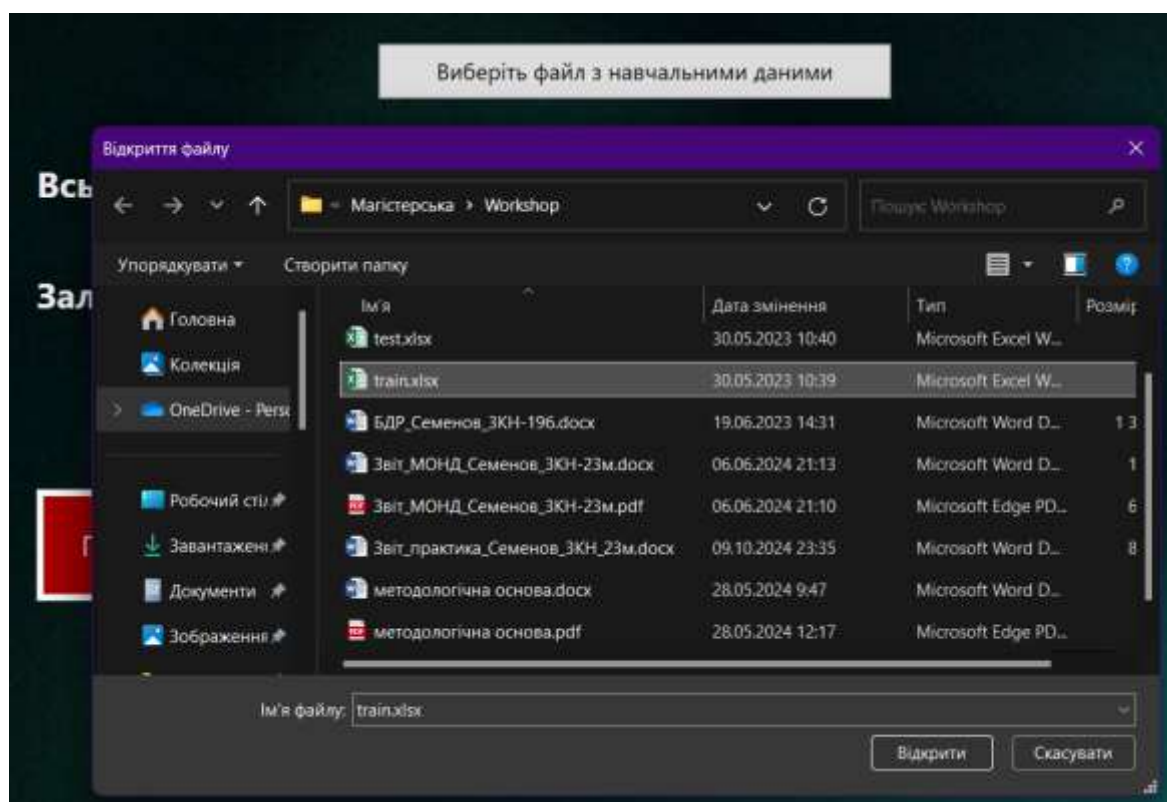


Рисунок В.7 – Вибір файлу з навчальною вибіркою

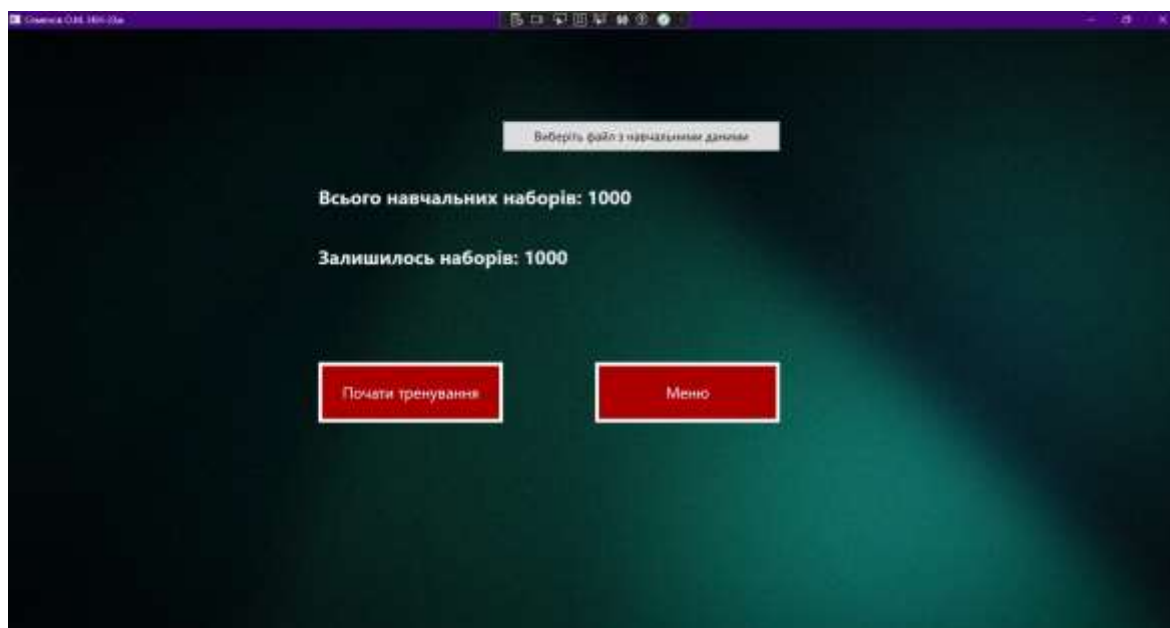


Рисунок В.8 – Вигляд програми після вибору файлу

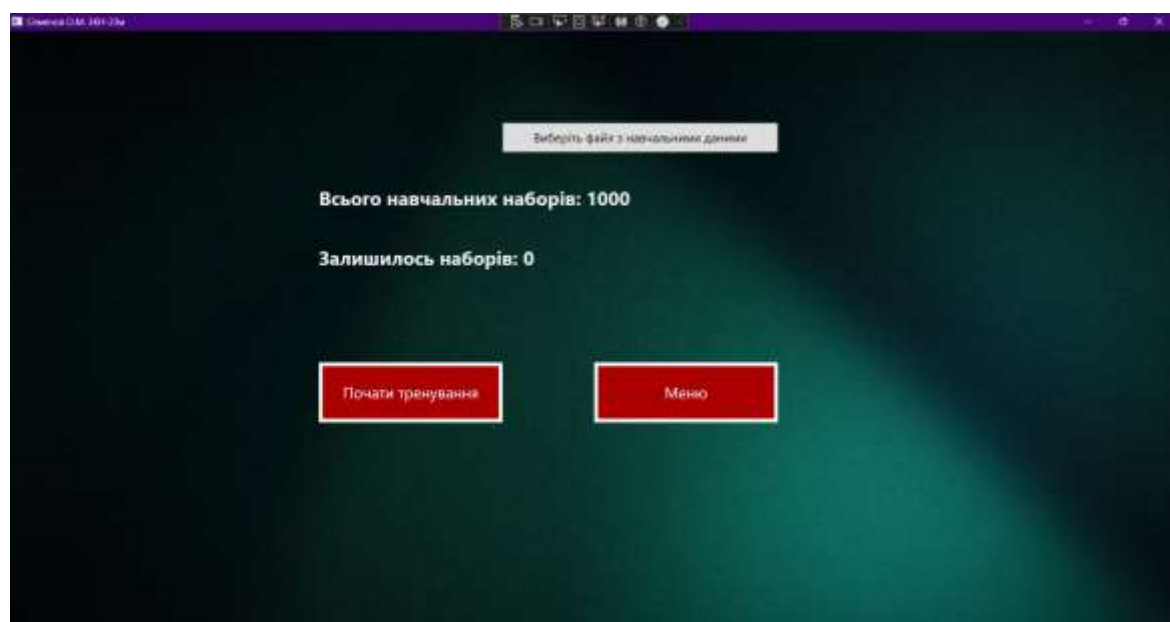


Рисунок В.9 – Вигляд програми після завершення навчання

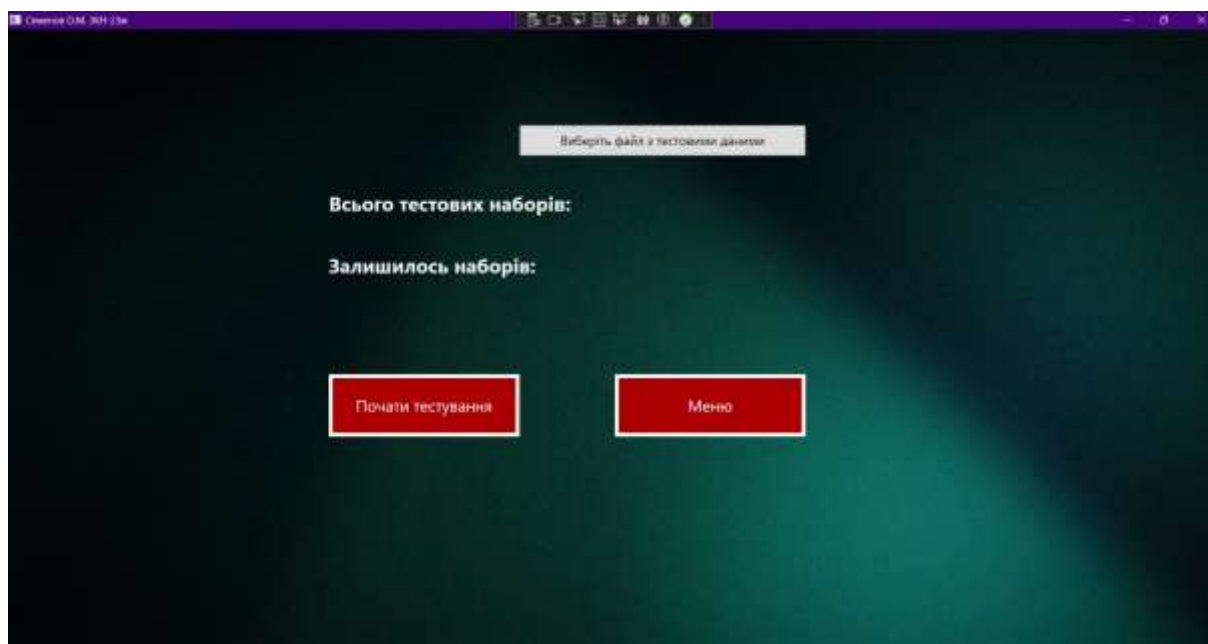


Рисунок В.10 – Вікно тестування програми

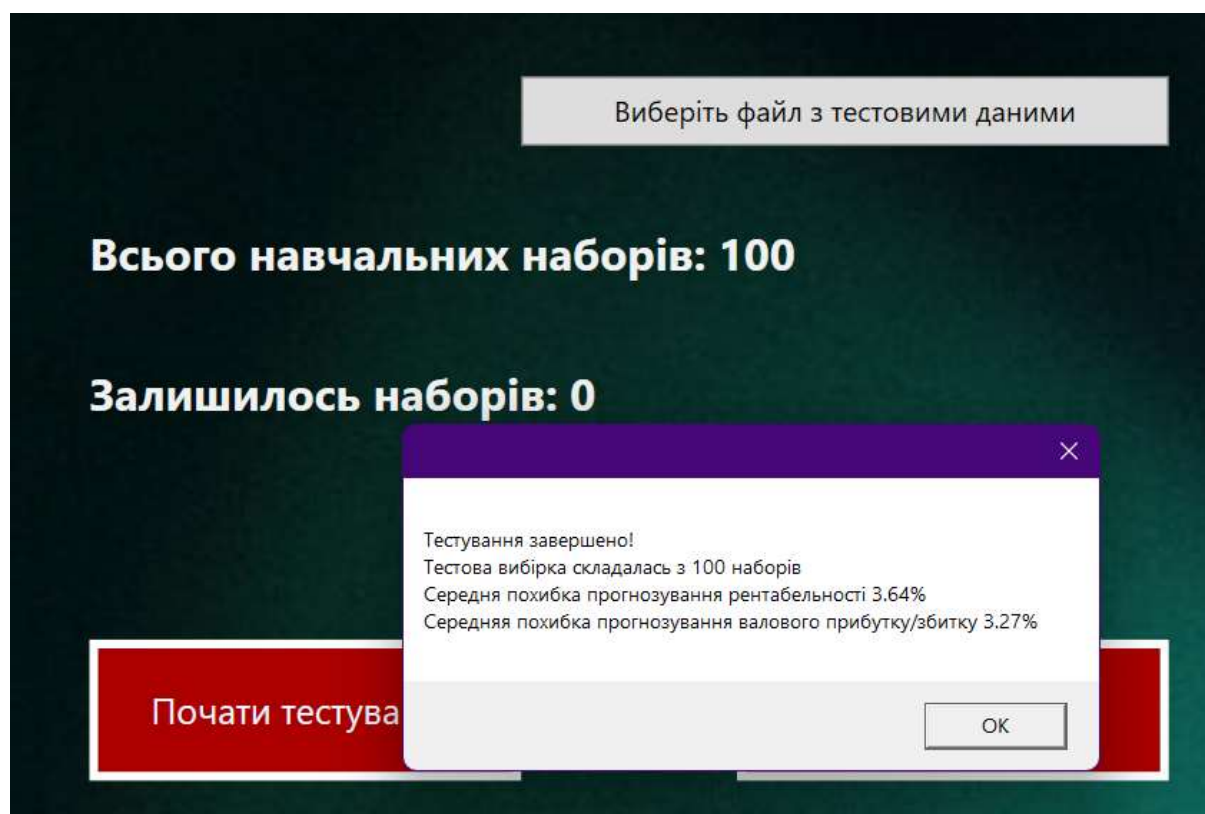


Рисунок В.11 – Вікно результатів тестування мережі

Введіть дані за останні 3 роки

	Нинішній рік	Минулий рік	Позаминулий рік
Рентабельність:	0.106	0.174	0.341
Лохіл:	9154	20316	39461
Собівартість:	85659	116779	115812
Валовий приб.	5662	8654	1443
Витрати:	3467	3571	4356
Вартість фондів:	31169	36523	38428
Серед. зарплата:	55.915	48.144	53.246
К-ть працівників:	190	270	271
Кредит. борг:	2936	2937	2447
Дебіт. борг:	0	0	0

Рентабельність: 0.057

Валовий прибуток/збиток: 2765

Прогнозувати
Меню

Рисунок В.12 – Результат прогнозування рентабельності, валового прибутку та збитку

Додаток Г (довідниковий)

Інструкція користувача

Програму необхідно запускати з комп'ютера, на якому встановлена операційна система Windows. Після запуску програми відкриється головне меню, що зображено на рисунку Г.1.

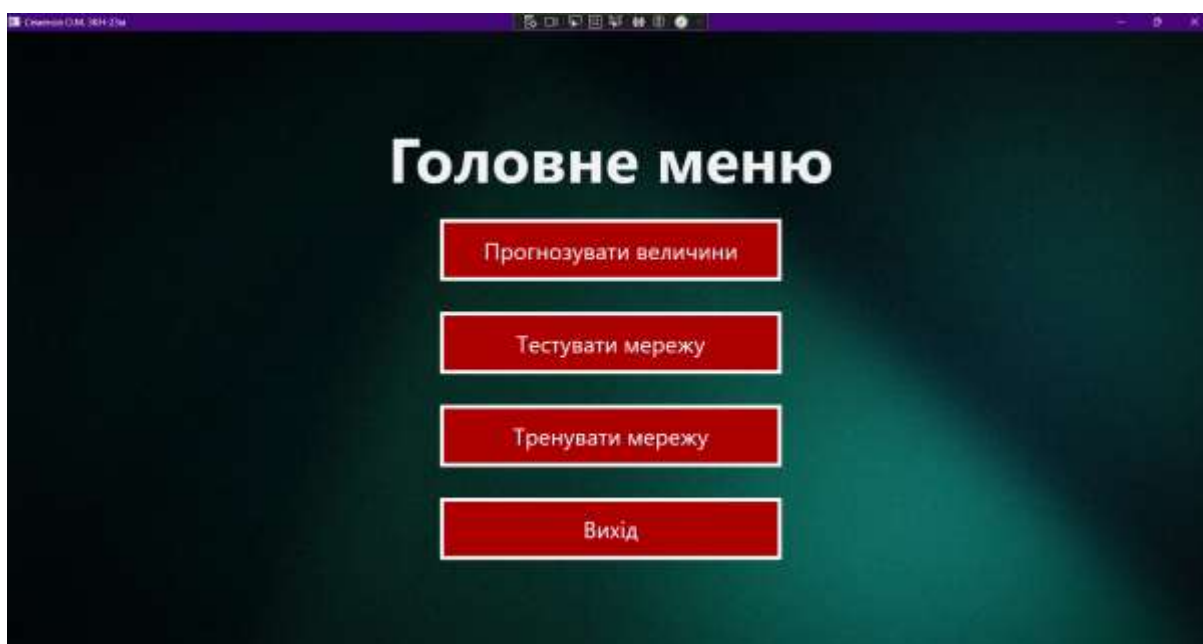


Рисунок Г.1 – Головне меню програми

Далі треба вибрати функцію програми за бажанням. Якщо програму на цьому комп'ютері запущено вперше, то спочатку треба провести навчання мережі. Для цього натисніть на кнопку під назвою «Тренувати мережу». Тоді відкриється вікно, зображене на рисунку Г.2.

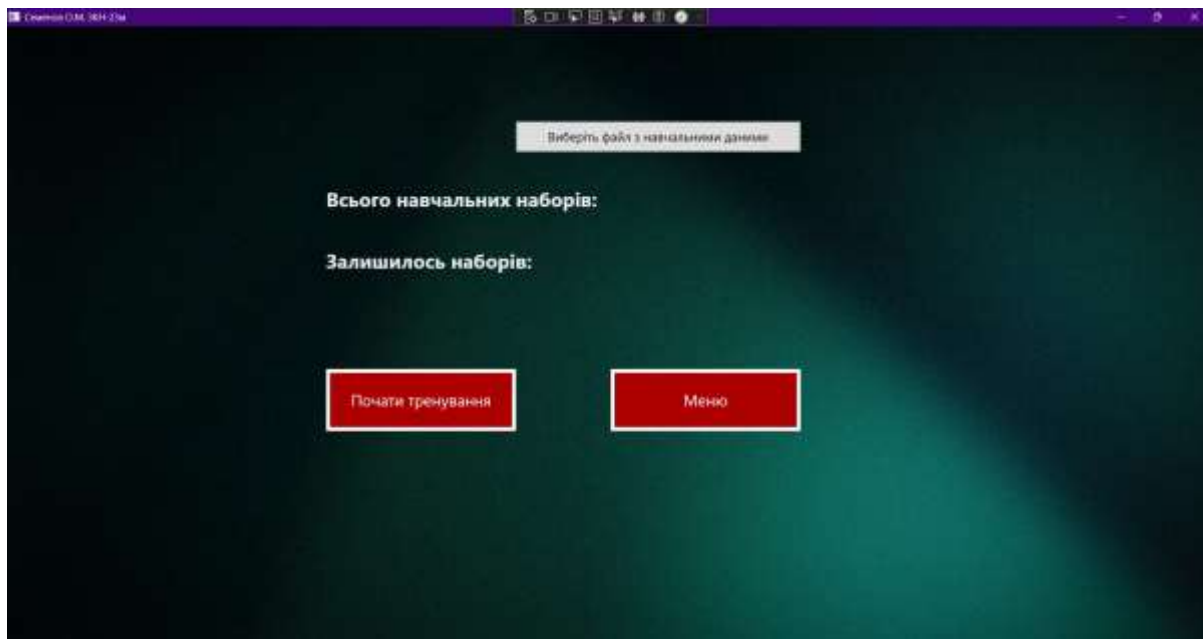


Рисунок Г.2 – Вікно тренування мережі

Перед початком навчання необхідно вибрати файл з навчальними даними. Це має бути файл типу Excel з 32 стовпчиками. В перших 30 стовпчиках мають бути вхідні дані, а в останніх – цільове значення рентабельності та прибутку/збитку відповідно. Тобто значення, які програма має видавати для таких вхідних даних.

Після вибору файлу з навчальними даними програма автоматично підрахує кількість наборів вхідних даних в ньому (кількість рядків). Для початку тренування треба натиснути на кнопку з відповідною назвою. Тоді, коли біля фрази «Залишилось наборів» з'явиться 0, навчання завершено. Можна вибрати новий файл з навчальними даними або повернутися в меню.

Якщо в меню вибрати кнопку під назвою «Тестувати мережу», відкриється вікно тестування мережі. Воно зображено на рисунку Г.3.

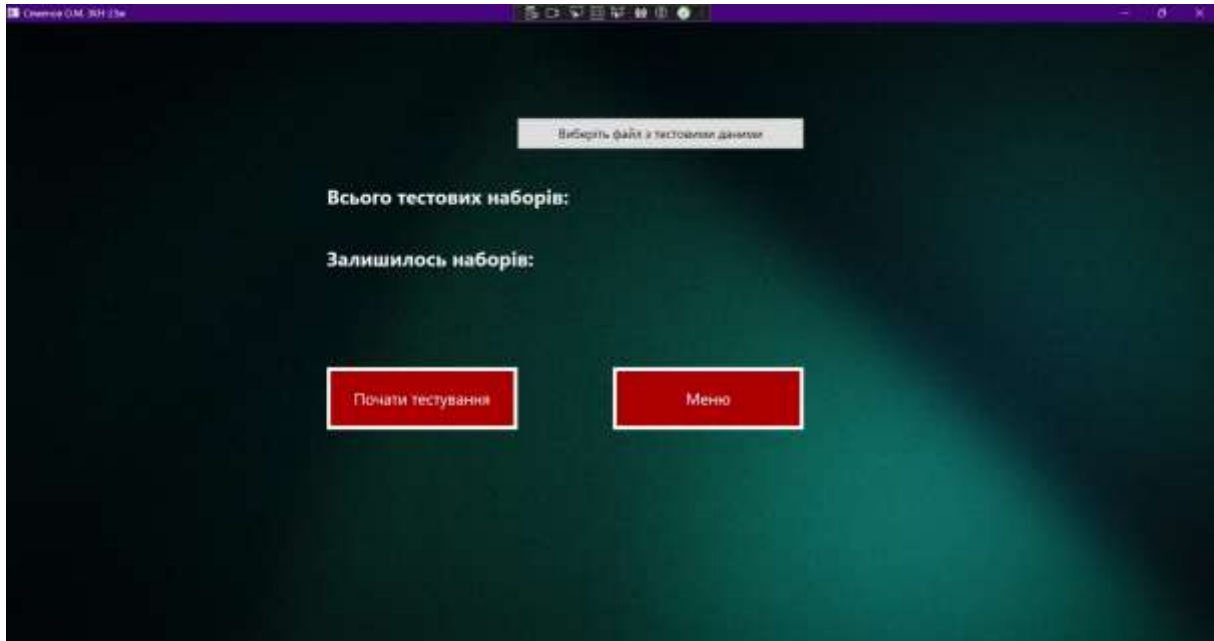


Рисунок Г.3 – Вікно тестування програми

Аналогічно до процедури навчання, треба вибрати файл з тестовими даними та натиснути на кнопку «Почати тестування». Формат і структура тестових даних така ж, як і в навчальних. Проте важливо не тестувати на таких же даних, на яких мережа навчається. Це спричинено тим, що в такому випадку точність прогнозування буде показана вищою, ніж вона є насправді. Після тестування відкриється невеличке вікно з результатами тестування. Воно зображено на рисунку Г.4.

В цьому вікні буде показана кількість тестових випадків (розмір тестової вибірки) та середні похибки прогнозування рентабельності та валового прибутку/збитку. Середня похибка прогнозування це середнє арифметичне похибок прогнозування всіх тестових випадків. Похибка прогнозування це результат ділення різниці між цільовим значенням і значенням, яке видала програма, на цільове значення. Вона показує, на скільки відсотків прогнозоване значення більше або менше, ніж воно має бути.

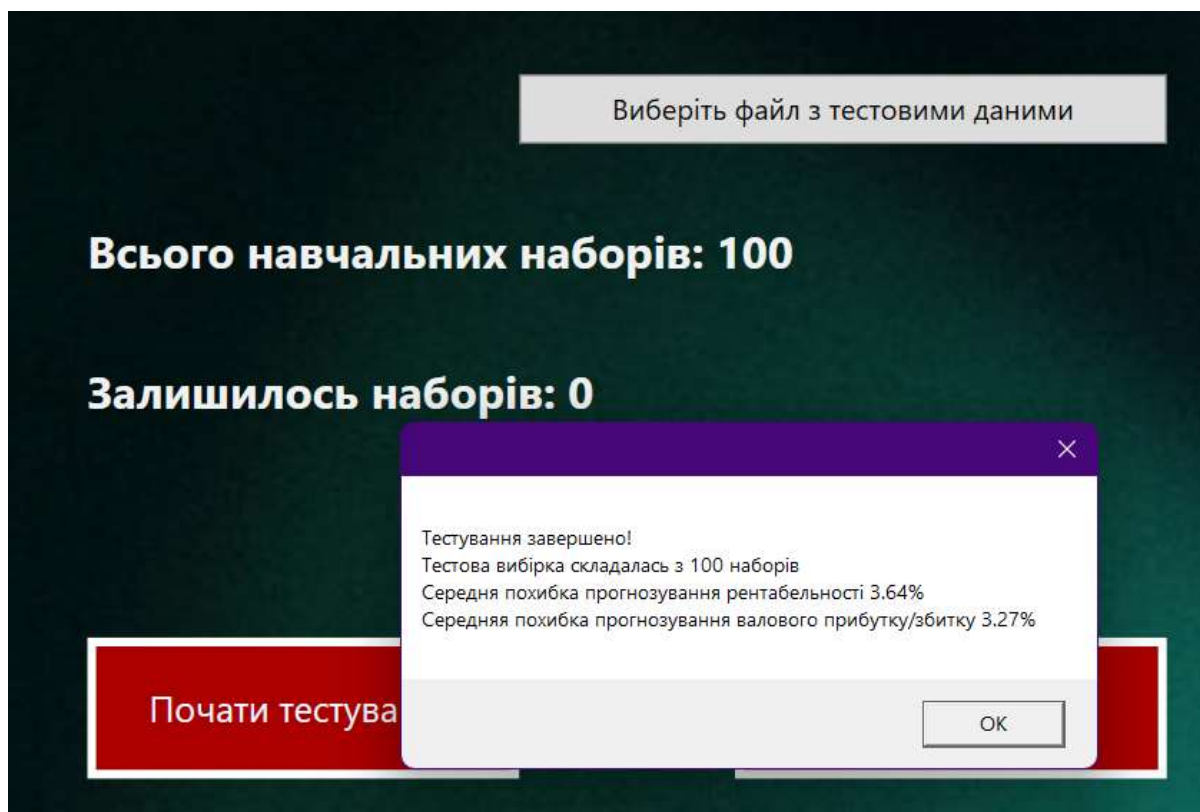


Рисунок Г.4 – Результат тестування мережі

Для прогнозування економічних показників свого підприємства треба натиснути на кнопку «Прогнозувати величини» у головному меню. Тоді відкриється вікно, зображене на рисунку Г.5.

На цьому вікні треба ввести власні фінансові показники за теперішній рік та за 2 минулі роки в порядку, поданому на інтерфейсі. Після заповнення всіх полів натисніть на кнопку «Прогнозувати». Програма видасть Вам результат.

Щоб вийти з програми, натисніть на хрестик в правому верхньому кутку вікна або на кнопку під назвою «Вихід» у головному вікні програми.

Введіть дані за останні 3 роки

	Нинішній рік	Минулий рік	Позаминулий рік
Рентабельність:	0.106	0.174	0.341
Лохіл:	9154	20316	39461
Собівартість:	85659	116779	115812
Валовий приб.	5662	8654	1443
Витрати:	3467	3571	4356
Вартість фондів:	31169	36523	38428
Серед. зарплата:	55.915	48.144	53.246
К-ть працівників:	190	270	271
Кредит. борг:	2936	2937	2447
Дебіт. борг:	0	0	0

Рентабельність: 0.057

Валовий прибуток/збиток: 2765

Прогнозувати
Меню

Рисунок Г.5 – Вікно прогнозування рентабельності, валового прибутку та збитку