

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

«На правах рукопису»
УДК 004.891.2

До захисту допущено:
В.О. Завідувача кафедр
_____ Михайло Новатарський
« ____ » _____ 2024 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Інженерія програмного забезпечення комп'ютерних систем» зі спеціальності 121 “Інженерія програмного забезпечення” на тему: «Інформаційна система підтримки прийняття рішень щодо спорядження військовослужбовця»

Виконав:

студент II курсу, групи ІМ-31мп
Шморгуненко Іван Олександрович _____

Керівник:

Ас. кф. ОТ, д.філ.,
Молчанова Анастасія Анатоліївна _____

Консультант з нормоконтролю:

проф. каф. ОТ, д.т.н., професор
Жабін Валерій Іванович _____

Рецензент: _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2024 року

5. Перелік завдань, які потрібно розробити: аналіз існуючих методів прийняття рішень, дослідження сучасних методів обробки історичних даних і використання нейронних мереж у військовій сфері, створення програмного забезпечення для збору та зберігання історичних даних, розробка нейронної мережі для аналізу історичних даних і визначення впливу спорядження на успішність місій, розробка програми для рекомендацій

6. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1.	Молчанова.А.А., ас.каф.ОТ		
2.	Молчанова.А.А., ас.каф.ОТ		
3.	Молчанова.А.А., ас.каф.ОТ		
4.	Молчанова.А.А., ас.каф.ОТ		

7. Дата видачі завдання 1.09.2024

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів дисертації	Примітка
1.	Затвердження теми дослідження. Визначення предмету дослідження	1.09.21-7.09.21	
2.	Аналіз існуючих підходів у сфері розробки систем прийняття рішень	8.09.21-21.09.21	
3.	Розробка архітектурного рішення системи	22.09.21-28.09.21	
4.	Розробка модуля збору історичних даних ("База знань")	29.09.21-05.10.21	
5.	Розробка модуля аналізу даних та формування таблиці відповідностей ("Підготовка експерта")	06.10.21-19.10.21	
5.	Розробка модуля рекомендацій спорядження ("Помічник у комплектуванні")	20.10.21-09.11.21	
6.	Тестування моделі та аналіз її ефективності.	10.11.21-13.11.21	
7.	Розробка стартап-проекту. Оформлення магістерської дисертації.	14.11.21-30.11.21	

Студент

Іван ШМОРГУНЕНКО

(підпис)

Науковий керівник дисертації

Анастасія МОЛЧАНОВА

(підпис)

РЕФЕРАТ

На магістерську дисертацію

Виконану на тему: Інформаційна система підтримки прийняття рішень щодо спорядження військовослужбовця
студентом: Шморгуниенком Іваном Олександровичем

Робота складається із вступу та чотирьох розділів. Загальний обсяг роботи: 82 аркушів основного тексту, 8 ілюстрацій, 6 таблиці. При підготовці використовувалася література з 19 різних джерел.

Сучасна військова сфера вимагає постійного вдосконалення підходів до підготовки, оснащення та забезпечення військовослужбовців. Успішне виконання завдань часто залежить не лише від професійної підготовки персоналу, але й від оптимального спорядження, що відповідає конкретним умовам місії. В умовах збройної агресії, яку переживає Україна, питання ефективності використання ресурсів набуває критично важливого значення.

Використання універсального набору спорядження не завжди виправдане, оскільки різні місії вимагають різних підходів залежно від типу завдання, погодних умов, рельєфу місцевості, а також фізичних характеристик солдатів. Оптимізація спорядження допомагає знизити зайве навантаження на військовослужбовця та підвищити ефективність виконання завдань.

Бойові дії, які відбуваються на території України, вимагають максимальної ефективності використання людських і матеріальних ресурсів. Розроблена система дозволить краще підготувати солдатів до виконання завдань, мінімізувати ризики та підвищити шанси на успішне завершення місії.

Метою магістерської роботи є підвищення ефективності експертної системи для оптимального вибору військового спорядження під різні умови задач.

Об'єктом дослідження є обчислювальні процеси в експертних системах.

Предметом дослідження є методи та алгоритми оптимізації процесу

вибору військового спорядження з використанням машинного навчання та паралельних обчислювальних підходів.

Науковою новизною роботи є спосіб суміщення нейронних мереж з паралельними обчислювальними підходами для створення інтелектуальної системи підтримки прийняття рішень у військовій сфері. Запропонований алгоритм дозволяє ефективно обробляти великі обсяги даних та надавати точні рекомендації щодо вибору спорядження під різні бойові задачі.

Було розроблено та впроваджено новий алгоритм аналізу даних, інтегровано спосіб суміщення нейронних мереж нейронні мережі з паралельними обчислювальними підходами, а також проведено експериментальні дослідження для верифікації ефективності запропонованої системи. Крім того, було здійснено розробку програмного забезпечення, яке реалізує запропонований підхід у вигляді інтелектуальної системи підтримки прийняття рішень.

Практична цінність роботи полягає у створенні інноваційної інформаційної системи, яка може бути впроваджена в реальних умовах для оптимізації планування військових операцій. Система дозволяє аналізувати великі обсяги даних, пов'язаних із виконанням місій, та формувати рекомендації щодо спорядження, яке підвищує шанси на успішне виконання завдань. Це сприяє не лише покращенню оперативної ефективності, але й підвищенню рівня безпеки військовослужбовців.

Результати роботи можуть бути використані для навчання військового персоналу, зокрема в моделюванні різних сценаріїв бойових операцій. Система також має потенціал для масштабування та адаптації під нові вимоги та умови, що робить її універсальним інструментом для підтримки рішень у військовій сфері. Додатково, вона може слугувати основою для подальших досліджень у галузі застосування штучного інтелекту в оборонних технологіях.

Ключові слова: Підтримка прийняття рішень, нейронна мережа, інформаційна система, аналіз ефективності, база знань, таблиця відповідностей, машинне навчання.

ABSTRACT

For the Master's Degree Certification

Prepared on the topic: **Information system for decision support in the equipment of military personnel.**

By student: **Ivan Oleksandrovysh Shmorhunenko**

The work consists of an introduction and four chapters. The total volume of the work is 82 pages of main text, 8 illustrations, and 6 tables. Literature from 19 different sources was used in its preparation.

The modern military sphere requires continuous improvement of approaches to the training, equipping, and provision of military personnel. Successful task execution often depends not only on the professional training of personnel but also on optimal equipment that meets the specific mission conditions. In the context of the armed aggression experienced by Ukraine, the efficiency of resource utilization becomes critically important.

The use of a universal set of equipment is not always justified, as different missions require different approaches depending on the type of task, weather conditions, terrain, as well as the physical characteristics of soldiers. Optimizing equipment helps reduce unnecessary burden on military personnel and increases the effectiveness of task execution.

Combat operations taking place on the territory of Ukraine demand maximum efficiency in the use of human and material resources. The developed system will allow for better preparation of soldiers for mission execution, minimize risks, and increase the chances of successful mission completion.

The aim of this master's thesis is to enhance the efficiency of an expert system for the optimal selection of military equipment under various task conditions.

The object of the research is computational processes in expert systems.

The subject of the research is methods and algorithms for optimizing the process of selecting military equipment using machine learning and parallel computational approaches.

The scientific novelty of the work lies in the method of combining neural networks with parallel computational approaches to create an intelligent decision support system in the military field. The proposed algorithm allows for efficient processing of large volumes of data and provides accurate recommendations for equipment selection for various combat tasks.

I have developed and implemented a new data analysis algorithm, integrated the method of combining neural networks with parallel computational approaches, and conducted experimental studies to verify the effectiveness of the proposed system. Additionally, software development was carried out to implement the proposed approach in the form of an intelligent decision support system.

The practical value of the work lies in the creation of an innovative information system that can be implemented in real conditions to optimize military operation planning. The system allows for the analysis of large volumes of data related to mission execution and forms recommendations for equipment that increases the chances of successful task completion. This contributes not only to improving operational efficiency but also to enhancing the safety level of military personnel.

The results of the work can be used for training military personnel, particularly in modeling various combat operation scenarios. The system also has the potential for scaling and adaptation to new requirements and conditions, making it a universal tool for decision support in the military sphere. Additionally, it can serve as a basis for further research in the field of artificial intelligence applications in defense technologies.

Keywords: Decision Support, Neural Network, Information System, Efficiency Analysis, Knowledge Base, Correspondence Table, Machine Learning.

ЗМІСТ

РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ	15
1.1 Поняття експертних систем	15
1.2 Огляд наукового здобутку в галузі експертних систем	16
1.2.1 Історія розвитку експертних систем.....	16
1.2.2 Теоретичні основи експертних систем.....	17
1.2.3 Методи представлення знань	17
1.2.4 Сучасні підходи до розробки експертних систем	18
1.2.5 Застосування експертних систем у різних галузях	18
1.2.6 Висновки з огляду наукового здобутку	19
1.3 Аналіз розробок та практичних реалізацій експертних систем у світовій практиці	20
1.3.1 Медичні експертні системи	20
1.3.2 Експертні системи у бізнесі	22
1.3.3 Промислові експертні системи	23
1.3.4 Військові експертні системи	25
1.4 Формулювання напрямів подальших досліджень та конкретизація завдань	27
1.4.1 Розробка бази знань експертної системи	27
1.4.2 Моделювання та навчання нейронної мережі	28
1.4.3 Розробка алгоритмів для формування рекомендацій.....	28
1.4.4 Адаптація системи до нових даних і масштабування.....	29
1.4.5 Валідація та тестування системи	30
Висновки	31
РОЗДІЛ 2 ІНФОРМАЦІЙНА СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО СПРОЯДЖЕННЯ ВІЙСЬКОВОСЛУЖБОВЦЯ	32
2.1 Обґрунтування вибору напрямку досліджень	32
2.1.1 Актуальність розробки системи підтримки прийняття рішень щодо спорядження військовослужбовців.....	32
2.1.2 Особливості функціонування системи та її компонентів.....	33
2.1.3 Визначення цілей і завдань дослідження.....	33

2.2	Методотика проведення дослідження	34
2.2.1	Опис підходу до розробки програми (Водоспадна модель).....	34
2.2.2	Етапи розробки: збір вимог, проєктування, реалізація, тестування.....	35
2.2.3	Взаємодія між програмами та її важливість	36
2.3	Вибір нейронної мережі для обробки даних	36
2.3.1	Огляд різних архітектур нейронних мереж	36
2.3.2.	Обґрунтування вибору багатошарової персептронної мережі (MLP)	39
2.4	Методи навчання та перевірки коректності роботи нейронної мережі.	40
2.4.1	Пояснення методів навчання нейронних мереж	40
2.4.2	Вибір оптимізаційного методу (Adam) для навчання мережі.....	43
2.4.3	Опис методів перевірки коректності	44
2.4.4	Вибір основних метрик ефективності для оцінки якості системи.....	45
2.5	Стек технологій для реалізації системи	46
2.5.1	Опис технологій для реалізації кожної з програм.....	46
2.5.2	Вибір фреймворку PyTorch для тренування нейронної мережі.....	47
2.5.3	Порівняння фреймворків та їх переваги для кожного етапу розробки	47
2.6	Оцінка похибок та методи покращення точності	48
2.6.1	Оцінка впливу якості даних на результати системи та методи боротьби з похибками.....	48
2.6.2	Аналіз результатів навчання нейронної мережі та візуалізація впливу спорядження на успішність місій	49
2.6.3	Покращення моделей на основі аналізу помилок і налаштування параметрів системи для підвищення точності	50
	Висновки	52
	РОЗДІЛ 3 ТЕХНІЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	53
3.1	Розробка архітектури	53
3.2	Загальна структура системи	55
3.2.1	Архітектура системи: єдність та модульність	55
3.2.2.	Взаємодія між модулями.....	55
3.3	Реалізація модулів системи	57

3.3.1 Модуль "База знань"	57
3.3.2 Модуль "Підготовка експерта"	60
3.3.3 Модуль "Помічник у комплектуванні"	65
Висновки	68
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЄКТУ	70
4.1 Загальна характеристика стартап-проєкту	70
4.2 Технологічний аудит ідеї проєкту	72
4.3. Аналіз ринкових можливостей запуску стартап-проєкту.....	75
4.4. Розробка ринкової стратегії проєкту	77
4.5 Розробка стратегії виходу проєкту на ринок.....	80
4.4.1 Визначення цільової аудиторії	80
4.4.2 Позиціонування продукту	80
4.4.4 Цінова політика	81
4.4.6 Технологічна підтримка.....	83
4.6. Розроблення маркетингової програми стартап-проєкту	84
Висновки	87
ВИСНОВКИ	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	90
ДОДАТОК 1	92

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

MLP – Multilayer Perceptron (багатошарова персептронна мережа), нейронна мережа, що складається з вхідного, одного або декількох прихованих та вихідного шарів нейронів.

SQL – Structured Query Language (структурована мова запитів), стандартна мова для управління та маніпулювання даними в реляційних базах даних.

CSV – Comma-Separated Values (значення, розділені комами), текстовий формат для подання табличних даних.

WPF – Windows Presentation Foundation (платформа представлення Windows), інструмент для створення графічного інтерфейсу користувача у додатках на базі Windows.

API – Application Programming Interface (прикладний програмний інтерфейс), набір засобів для створення програмного забезпечення, що дозволяє взаємодіяти з іншими програмами або сервісами.

PyTorch – гнучкий і популярний фреймворк для розробки нейронних мереж та застосування методів глибинного навчання.

Adam – Adaptive Moment Estimation (адаптивна оцінка моментів), метод оптимізації для навчання нейронних мереж, який об'єднує переваги моменту та адаптивних методів.

RNN – Recurrent Neural Network (рекурентна нейронна мережа), нейронна мережа, що використовує зворотні зв'язки для обробки послідовних даних.

CNN – Convolutional Neural Network (згорткова нейронна мережа), спеціалізована нейронна мережа, зазвичай використовується для обробки зображень.

NN – Neural Network (нейронна мережа), модель машинного навчання, що імітує роботу біологічного мозку.

EF – Entity Framework, ORM (Object-Relational Mapping) бібліотека для роботи з реляційними базами даних у .NET додатках.

.NET – програмна платформа від компанії Microsoft, яка використовується для розробки додатків різних типів.

MSE – Mean Squared Error (середньоквадратична похибка), метрика оцінки якості моделей машинного навчання.

AI – Artificial Intelligence (штучний інтелект), галузь комп'ютерних наук, що займається створенням інтелектуальних систем, здатних виконувати завдання, які зазвичай вимагають людського інтелекту.

UI – User Interface (інтерфейс користувача), частина програмного забезпечення, що взаємодіє з користувачем.

GPU – Graphics Processing Unit (графічний процесор), апаратний пристрій для обробки графіки, також використовується для обчислень у машинному навчанні.

SQL Server – система управління базами даних (СУБД), створена компанією Microsoft для роботи з реляційними базами даних.

ВСТУП

Сучасний розвиток військових технологій та інформаційних систем стає ключовим чинником забезпечення ефективності та безпеки операцій у військовій сфері. Успіх бойових завдань залежить від багатьох факторів, включаючи належний вибір спорядження, адаптованого до конкретних умов місії, характеристик солдата та зовнішніх обставин. У зв'язку з цим виникає потреба у створенні інноваційних систем підтримки прийняття рішень, які ґрунтуються на аналізі історичних даних і застосуванні сучасних технологій штучного інтелекту.

Повномасштабна війна в Україні стала суворим випробуванням для всієї військової системи країни. Збройні Сили України зіткнулися з необхідністю оперативно адаптуватися до швидко змінюваних умов ведення бойових дій. Забезпечення солдатів необхідним спорядженням у потрібний час стало одним із критичних завдань. У цих умовах автоматизація вибору спорядження та адаптація до специфіки місії і характеристики військовослужбовців є важливими для зменшення втрат і підвищення ефективності виконання завдань.

Розробка інформаційної системи прийняття рішень щодо спорядження військовослужбовців спрямована на вирішення низки ключових завдань. Система повинна забезпечувати збирання та аналіз історичних даних, генерувати таблиці відповідностей між характеристиками місії, параметрами солдатів та ефективністю спорядження. Вона також має надавати рекомендації щодо найоптимальнішого спорядження для підвищення ймовірності успішності завдань.

Актуальність цієї роботи зумовлена як загальними тенденціями розвитку військових технологій, так і конкретними викликами, які виникають у контексті оборони України. У сучасних реаліях автоматизація та оптимізація вибору спорядження дозволяють не лише зекономити ресурси, але й значно покращити підготовку до бойових дій, зменшити ризики для військовослужбовців та підвищити їхню бойову спроможність.

У процесі виконання цієї роботи були враховані сучасні методи та інструменти машинного навчання, зокрема нейронні мережі, а також технології

обробки та аналізу даних. Було обрано багат шарову перцептронну мережу, оптимізовану за допомогою алгоритму Adam, що дозволяє ефективно обробляти різноманітні дані та робити точні прогнози. Для інтеграції даних використовувалася база даних SQL Server, що забезпечує структуроване зберігання інформації про місії, солдатів та спорядження.

Метою роботи є підвищення ефективності експертної системи для оптимального вибору військового спорядження під різні умови задач, сам система допоможе автоматизувати вибір спорядження для військовослужбовців, підвищуючи ефективність місій.

Завдання роботи включають:

- Аналіз існуючих методів підтримки прийняття рішень у військовій сфері.
- Розробку архітектури інформаційної системи.
- Реалізацію модулів системи, що включають збір даних, аналіз та генерацію таблиці відповідностей, а також надання рекомендацій.
- Оцінку точності та ефективності розробленої системи.

Наукова новизна дослідження полягає у синергічному використанні штучного інтелекту та паралельної обробки для автоматизації вибору спорядження, враховуючи широкий спектр факторів, що впливають на успішність місій. Практична значущість полягає у можливості застосування системи у військових підрозділах для підвищення ефективності їхньої діяльності.

Ця робота відкриває нові можливості для інтеграції інформаційних систем у військовій галузі, сприяючи підвищенню їхньої оперативності, ефективності та адаптивності, що є особливо важливим у сучасних умовах війни.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

1.1 Поняття експертних систем

Експертні системи є важливим компонентом інформаційних систем підтримки прийняття рішень (СППР) і відіграють ключову роль у багатьох галузях, таких як медицина, бізнес, промисловість та військова сфера. Вони забезпечують автоматизацію процесу прийняття рішень шляхом використання бази знань і логічних правил для обробки великої кількості даних та надання рекомендацій або конкретних рішень у складних умовах. Особливо це актуально для військової сфери, де швидкість і точність прийняття рішень мають вирішальне значення для забезпечення безпеки і успіху місії.

Експертні системи дозволяють моделювати процеси прийняття рішень на основі історичних даних та знань, отриманих від фахівців. Вони можуть враховувати безліч факторів, таких як характеристики солдата, умови місії, погодні фактори, та визначати, які елементи спорядження найбільше впливають на успішність виконання завдання. Використання таких систем у військовій галузі має на меті підвищити точність рекомендацій, мінімізувати ризики та забезпечити оптимальний підбір спорядження для солдатів у різних умовах.

Метою цього розділу є аналіз наукових, інноваційних та практичних досягнень у сфері експертних систем, зокрема їхнього використання у військовій сфері та суміжних галузях. Проведення такого аналізу дозволить визначити основні тенденції розвитку технологій, оцінити їх переваги та недоліки, а також виявити проблеми, що залишаються невирішеними. Це, своєю чергою, дасть можливість конкретизувати напрямки подальших досліджень і визначити завдання магістерської роботи, спрямовані на розробку експертної системи для підтримки прийняття рішень щодо спорядження військовослужбовців.

1.2 Огляд наукового здобутку в галузі експертних систем

Експертні системи (ЕС) є однією з найстаріших і найрозвинутіших галузей штучного інтелекту (ШІ), що виникла у 1960-х роках і продовжує розвиватися до сьогодні. Вони дозволяють автоматизувати процес прийняття рішень на основі знань та правил, що моделюють логіку експертів. У цьому розділі розглянемо основні етапи розвитку наукової думки в галузі експертних систем, їх теоретичні основи, підходи до реалізації та застосування в різних галузях.

1.2.1 Історія розвитку експертних систем

Розробка експертних систем почалася в 1960-х роках, коли дослідники почали розглядати можливість автоматизації процесу прийняття рішень, використовуючи обчислювальні потужності та знання експертів. Перші експертні системи, такі як Dendral (система для аналізу хімічних сполук) та MYCIN (медична експертна система для діагностики інфекційних захворювань), стали прикладами успішного застосування знань у вузьких галузях.

Етапи розвитку експертних систем:

- 1960-ті – 1970-ті роки: Поява перших ЕС, таких як Dendral та MYCIN, які використовували прості правила і бази знань для прийняття рішень у вузькоспеціалізованих галузях[1].
- 1980-ті роки: Бурхливий розвиток ЕС, впровадження знань у різні галузі: медицина, хімія, промисловість. З'явилися системи з більш складними базами знань і механізмами виведення.
- 1990-ті роки – до сьогодні: Інтеграція експертних систем із новими технологіями, такими як нейронні мережі та машинне навчання, для покращення точності і швидкості рішень. Розробка гібридних систем, які комбінують кілька методів для досягнення найкращих результатів.

1.2.2 Теоретичні основи експертних систем

Основні теоретичні основи експертних систем зосереджені на структурі та принципах роботи, що дозволяють їм обробляти знання та застосовувати їх у реальних умовах.

Основні компоненти експертної системи:

- База знань: Це основний компонент, що містить інформацію та правила, необхідні для прийняття рішень. Знання можуть бути представлені у вигляді правил, логічних висловів, фактів або онтологій. Бази знань можуть поповнюватися або оновлюватися, що дозволяє експертній системі адаптуватися до нових умов.
- Механізм виведення: Це алгоритм або набір алгоритмів, що використовуються для обробки даних з бази знань і прийняття рішень. Він порівнює поточну ситуацію з інформацією в базі знань та застосовує відповідні правила для генерації результатів.
- Інтерфейс користувача: Інтерактивний компонент, який забезпечує зручну взаємодію з системою. Користувач вводить дані та отримує рекомендації у зрозумілому вигляді.
- Модуль пояснення: Деякі експертні системи містять модуль, що пояснює користувачу, як було отримано конкретне рішення, що підвищує довіру до системи та забезпечує прозорість процесу прийняття рішень.

1.2.3 Методи представлення знань

У експертних системах існує кілька методів представлення знань, кожен з яких має свої переваги та недоліки. Найбільш поширеними є:

- Правила (Rule-based systems): Правила є найпростішим і найпоширенішим способом представлення знань. Вони виглядають як "якщо-то" (IF-THEN) висловлювання, що визначають, як повинна реагувати система на певні умови. Правила є ефективними для вузькоспеціалізованих систем, але можуть стати неефективними при великій кількості знань, оскільки їх складно оновлювати та підтримувати.

- **Об'єктно-орієнтовані підходи:** Знання представляються у вигляді об'єктів із властивостями і методами. Цей підхід дозволяє легше інтегрувати експертні системи з іншими програмними рішеннями та забезпечує більшу гнучкість у представленні складних знань.
- **Онтології:** Використовуються для представлення структурованих знань і визначення відносин між різними об'єктами та концепціями. Онтології часто застосовуються у сучасних експертних системах, особливо в галузях, де важливо розуміти контекст і взаємозв'язки між даними (наприклад, медицина чи управління ресурсами).

1.2.4 Сучасні підходи до розробки експертних систем

У сучасних експертних системах активно використовуються новітні технології, такі як штучний інтелект, нейронні мережі та машинне навчання[2]. Ці підходи дозволяють експертним системам навчатися на основі історичних даних, що значно підвищує їхню точність і адаптивність.

Гібридні експертні системи:

- **Поєднання експертних систем та нейронних мереж:** Нейронні мережі використовуються для аналізу великих обсягів даних і виявлення складних закономірностей, які складно формалізувати за допомогою правил. Наприклад, у медичних експертних системах нейронні мережі можуть аналізувати історію захворювань та виявляти ризики захворювань на основі даних пацієнта.
- **Інтеграція з алгоритмами машинного навчання:** Системи, що базуються на машинному навчанні, можуть аналізувати дані в режимі реального часу та адаптувати свої рішення на основі нових даних. Це особливо корисно у військових та промислових системах, де умови можуть змінюватися дуже швидко[3].

1.2.5 Застосування експертних систем у різних галузях

Експертні системи широко застосовуються для діагностики захворювань та рекомендацій щодо лікування. Системи, такі як Watson for Health,

використовують базу знань, що містить наукові дослідження та клінічні дані, для надання точних рекомендацій лікарям.

В бізнесі ж вони допомагають приймати рішення щодо управління запасами, планування виробництва та стратегічного планування. SAP BusinessObjects є прикладом такої системи, яка інтегрує аналіз даних і надає прогнози для прийняття бізнес-рішень.

У військових умовах експертні системи допомагають планувати операції, аналізувати загрози та приймати рішення в умовах обмеженого часу. Системи, такі як JCAS (Joint Conflict and Tactical Simulation), моделюють бойові ситуації, аналізуючи дані про противника, погодні умови та спорядження військ[4].

1.2.6 Висновки з огляду наукового здобутку

Експертні системи демонструють високу ефективність у багатьох галузях, але вони стикаються з рядом викликів. Одним з основних недоліків є залежність від бази знань: системи не можуть бути ефективними без актуальної та повної інформації. Крім того, складність інтеграції та підтримки таких систем зростає зі збільшенням обсягів знань.

Незважаючи на ці проблеми, сучасні дослідження та розробки показують, що інтеграція експертних систем із технологіями машинного навчання та штучного інтелекту значно підвищує їхню ефективність і адаптивність. Це особливо актуально для військової сфери, де рішення мають бути максимально точними та своєчасними. Подальший розвиток експертних систем потребує уваги до таких аспектів, як гнучкість баз знань, можливість швидкої адаптації до нових умов та зручність використання.

Таким чином, науковий здобуток у галузі експертних систем забезпечує фундамент для розробки ефективної системи підтримки прийняття рішень щодо спорядження військовослужбовців, що і є метою даного дослідження.

1.3 Аналіз розробок та практичних реалізацій експертних систем у світовій практиці

Експертні системи (ЕС) знаходять широке застосування в різних галузях, таких як медицина, промисловість, логістика та військова сфера. Вони використовуються для автоматизації процесу прийняття рішень, підвищення точності аналізу даних та оптимізації ресурсів. У цьому розділі буде розглянуто найвизначніші світові розробки експертних систем, а також проведено їх аналіз з точки зору переваг і недоліків[5].

1.3.1 Медичні експертні системи

Медицина є однією з найпопулярніших галузей для впровадження експертних систем. ЕС у цій сфері допомагають лікарям приймати рішення щодо діагностики та лікування пацієнтів, враховуючи симптоми, історію хвороб і результати аналізів.

IBM Watson for Oncology

Ця система використовує потужний обсяг медичних даних, наукових статей та клінічних випадків для аналізу та надання рекомендацій лікарям щодо лікування онкологічних захворювань. Watson for Oncology враховує індивідуальні особливості пацієнта та генерує декілька варіантів лікування з оцінкою ефективності кожного з них.

Переваги:

- Швидкість аналізу: Система обробляє величезні обсяги медичних даних за короткий час, що дозволяє лікарям отримувати рекомендації швидше, ніж при ручному пошуку інформації.
- Персоналізація лікування: Враховує індивідуальні особливості пацієнта, що підвищує ефективність лікування.
- Доступ до останніх досліджень: Система регулярно оновлює свою базу знань, що дозволяє користувачам працювати з актуальними медичними даними.

Недоліки:

- Висока вартість впровадження та обслуговування: Системи на кшталт Watson потребують значних фінансових ресурсів для налаштування та підтримки.
- Необхідність великих обсягів даних: Для ефективної роботи необхідно мати доступ до великої кількості клінічних даних та наукових досліджень.
- Потреба у кваліфікованому персоналі: Для ефективного використання системи потрібні лікарі з високим рівнем підготовки, які можуть інтерпретувати результати та рекомендації.

InferMed's Arezzo

InferMed's Arezzo — це експертна система для клінічної підтримки прийняття рішень, яка допомагає лікарям у діагностиці та виборі лікування. Система використовує базу знань, яка містить клінічні протоколи та правила, для автоматизації процесу прийняття рішень. Вона підтримує лікарів на кожному етапі лікування, від постановки діагнозу до моніторингу ефективності лікування.

Переваги:

- Стандартизація лікування: Система допомагає стандартизувати лікування відповідно до затверджених протоколів.
- Покращення якості обслуговування пацієнтів: Лікарі отримують доступ до найновіших клінічних рекомендацій.
- Інтерактивність: Користувачі можуть коригувати рекомендації на основі нових даних, що підвищує гнучкість системи.

Недоліки:

- Залежність від бази знань: Необхідно регулярно оновлювати базу знань для актуальності рекомендацій.
- Обмежена персоналізація: Система не завжди враховує унікальні особливості кожного пацієнта.
- Високі вимоги до впровадження: Вимагає інтеграції з електронними медичними записами та іншими клінічними системами.

1.3.2 Експертні системи у бізнесі

Бізнес-середовище активно використовує експертні системи для аналізу ринку, прогнозування, управління запасами та прийняття стратегічних рішень. Такі системи допомагають компаніям підвищити ефективність управління ресурсами та покращити взаємодію з клієнтами.

Oracle Hyperion

Oracle Hyperion — це система бізнес-аналітики, яка використовується для фінансового прогнозування, планування бюджету та стратегічного планування. Система аналізує фінансові дані компанії, використовуючи складні алгоритми для створення звітів та рекомендацій щодо оптимізації витрат і ресурсів.

Переваги:

- Розширені можливості аналізу даних: Система підтримує складні алгоритми прогнозування та аналізу, що дозволяє приймати обґрунтовані рішення.
- Інтеграція з ERP-системами: Hyperion легко інтегрується з іншими корпоративними системами, що дозволяє працювати з централізованими даними.
- Гнучкість у налаштуванні звітів: Користувачі можуть створювати звіти, які відповідають конкретним потребам їхнього бізнесу.

Недоліки:

- Складність налаштування: Вимагає кваліфікованих фахівців для налаштування та підтримки системи.
- Висока вартість: Ліцензії та впровадження системи є дорогими, що може обмежити її доступність для малого та середнього бізнесу.
- Чутливість до якості даних: Якщо в базі даних є помилки або неповні дані, результати можуть бути некоректними.

SAP BusinessObjects

Ця система є однією з провідних платформ для бізнес-аналітики, що забезпечує підприємства можливістю збирати, обробляти та аналізувати дані в режимі реального часу. Вона інтегрується з різними бізнес-процесами,

забезпечуючи повну картину діяльності компанії та надаючи рекомендації для прийняття обґрунтованих рішень.

Переваги:

- Інтеграція з іншими системами: SAP BusinessObjects легко інтегрується з іншими бізнес-додатками, що дозволяє централізовано керувати даними.
- Швидкий аналіз даних у реальному часі: Система може працювати з великими обсягами даних, надаючи актуальну інформацію для прийняття рішень.
- Гнучкість у налаштуванні: Користувачі можуть налаштовувати систему під конкретні потреби свого бізнесу, створюючи власні звіти та аналітичні панелі.

Недоліки:

- Висока вартість ліцензій та підтримки: Впровадження системи потребує значних фінансових витрат, що може бути проблемою для малого бізнесу.
- Складність налаштування та використання: Система потребує кваліфікованого персоналу для налаштування та обслуговування, що може бути викликом для компаній без технічних фахівців.
- Чутливість до якості даних: Неефективне управління даними або наявність помилок у них може призвести до неправильних висновків і рішень.

1.3.3 Промислові експертні системи

У промисловості експертні системи застосовуються для оптимізації виробничих процесів, управління автоматизованими лініями та моніторингу технічного стану обладнання. Вони допомагають підвищити ефективність виробництва та зменшити витрати на технічне обслуговування.

Siemens XHQ

Siemens XHQ — експертна система для управління виробничими процесами в реальному часі. Система забезпечує аналіз ефективності обладнання, управління виробничими лініями та моніторинг технічного стану. XHQ інтегрується з промисловими датчиками та системами моніторингу, що дозволяє швидко реагувати на зміни в процесах.

Переваги:

- Управління виробництвом у реальному часі: Система дозволяє швидко коригувати виробничі процеси на основі даних, що надходять у реальному часі.
- Підвищення ефективності виробництва: Завдяки аналізу даних і оптимізації процесів система допомагає зменшити витрати і підвищити продуктивність.
- Інтеграція з іншими промисловими системами: XHQ легко інтегрується з існуючими системами управління виробництвом.

Недоліки:

- Висока вартість впровадження: Для інтеграції з існуючими виробничими процесами потрібні значні інвестиції.
- Потреба в кваліфікованому персоналі: Обслуговування та налаштування системи вимагають спеціалізованих навичок.
- Вразливість до збоїв в обладнанні: Якщо сенсори або обладнання не працюють належним чином, система може генерувати некоректні рекомендації.

GE Predix

GE Predix є індустріальною експертною системою для моніторингу стану обладнання, прогнозування технічних несправностей та управління виробничими процесами. Вона обробляє дані з датчиків, встановлених на виробничих лініях, і надає рекомендації щодо технічного обслуговування та оптимізації роботи обладнання.

Переваги:

- Прогнозування несправностей: Система виявляє потенційні проблеми до того, як вони вплинуть на роботу виробничої лінії, що знижує витрати на ремонт.
- Оптимізація процесів у реальному часі: Система аналізує дані від обладнання в режимі реального часу і може адаптувати процеси для підвищення продуктивності.
- Зниження витрат: Використання предиктивного обслуговування знижує кількість позапланових зупинок та витрати на технічне обслуговування.

Недоліки:

- Високі вимоги до обладнання: Для ефективного використання системи потрібне встановлення додаткових датчиків і систем моніторингу, що може бути дорогим.
- Складність інтеграції: Інтеграція системи з існуючими виробничими процесами може бути складною, особливо на підприємствах зі старим обладнанням.
- Потреба в кваліфікованому персоналі: Система потребує технічного персоналу для налаштування, управління та інтерпретації результатів.

1.3.4 Військові експертні системи

Військова сфера також активно впроваджує експертні системи для планування операцій, управління ресурсами та аналізу бойових ситуацій. Такі системи дозволяють командирам швидко приймати рішення, враховуючи дані з різних джерел, зокрема про розташування підрозділів, рельєф місцевості та погодні умови.

Sitreps (Situational Reports)

Sitreps — це військова експертна система для моніторингу та аналізу оперативної обстановки в реальному часі. Вона обробляє дані з різних джерел, включаючи безпілотні літальні апарати (БПЛА), супутники та радіопередавачі, для створення звітів та прогнозів щодо розвитку бойових ситуацій[3].

Переваги:

- Аналіз у реальному часі: Система забезпечує швидкий аналіз бойових умов і надає рекомендації щодо подальших дій у режимі реального часу.
- Інтеграція даних із різних джерел: Sitreps інтегрує інформацію з різних джерел, що дозволяє командуванню отримувати повну картину бойової обстановки.
- Моделювання можливих сценаріїв: Система дозволяє змоделювати різні варіанти розвитку подій і обрати оптимальну стратегію.

Недоліки:

- Висока складність інтеграції: Для коректної роботи система потребує інтеграції з великою кількістю зовнішніх джерел даних, що ускладнює її налаштування.
- Залежність від зовнішніх джерел: Якщо канали зв'язку або джерела даних недоступні, система не зможе забезпечити актуальну інформацію.
- Потреба у високому рівні безпеки: Оскільки система обробляє критично важливі дані, вона потребує високого рівня захисту від кібератак.

JCAS (Joint Conflict and Tactical Simulation)

JCAS — це експертна система, яка використовується для моделювання та аналізу бойових дій, планування військових операцій та навчання військових. Система враховує різні фактори, такі як рельєф місцевості, чисельність сил, озброєння та погодні умови, для моделювання бойових сценаріїв та надання рекомендацій командуванню[3].

Переваги:

- Моделювання бойових ситуацій: Система дозволяє моделювати різні сценарії бою, що допомагає підготуватися до реальних умов.
- Аналіз у реальному часі: Система може працювати з даними в реальному часі, надаючи актуальні рекомендації під час бойових дій.
- Підвищення рівня навчання військових: Система використовується для навчання командирів та солдатів, покращуючи їхню готовність до різних бойових ситуацій.

Недоліки:

- Висока вартість впровадження: Системи такого типу потребують значних інвестицій у розробку та підтримку.
- Складність інтеграції з іншими військовими системами: Для ефективного використання система повинна інтегруватися з іншими компонентами військової інфраструктури, що може бути складним процесом.
- Потреба в оновленні даних: Ефективність системи залежить від актуальності даних, тому її необхідно регулярно оновлювати.

Світові розробки експертних систем демонструють високу ефективність у різних галузях, однак вони стикаються з рядом проблем. Основні з них — це висока вартість впровадження, складність інтеграції та необхідність у великій кількості даних для навчання системи. Водночас, інтеграція експертних систем з новими технологіями, такими як машинне навчання та нейронні мережі, дозволяє значно підвищити їхню точність та адаптивність.

Аналіз розробок і практичних реалізацій підтверджує, що експертні системи мають величезний потенціал у військовій сфері, де рішення мають бути швидкими, точними та враховувати безліч змінних. Подальший розвиток цих систем у контексті військового спорядження потребує гнучкості у налаштуванні, адаптивності до нових умов та можливості інтеграції з іншими військовими системами.

1.4 Формулювання напрямів подальших досліджень та конкретизація завдань

На основі аналізу сучасних експертних систем, виявлених проблем та специфічних вимог до розробки системи для підтримки прийняття рішень щодо спорядження військовослужбовців, формуються наступні напрями подальших досліджень та конкретизуються завдання, які необхідно виконати для досягнення цілей магістерської роботи.

1.4.1 Розробка бази знань експертної системи

Основним завданням є створення бази знань, що включатиме історичні дані про бойові операції, оцінені експертами, та відповідні рівні успішності місій. База знань повинна бути гнучкою та підлягати регулярному оновленню, щоб забезпечити актуальність системи.

Так для вирішення цієї задачі потрібно провести аналіз методів структуризації знань для формування багатовимірної таблиці відповідностей, яка буде використовуватися як основа для нейронної мережі. Необхідно дослідити, як

найкраще організувати дані для ефективного навчання нейронної мережі та забезпечення точності рекомендацій. Сюди входить задача по створенню структури бази даних для збереження історичних місій, спорядження, та оцінок успішності місій, враховуючи суб'єктивні оцінки експертів і задача по впровадженню механізм для оновлення бази знань з використанням нових даних, внесених експертами за допомогою програми 1("База знань").

1.4.2 Моделювання та навчання нейронної мережі

Для забезпечення гнучкості та адаптивності системи необхідно розробити та навчити нейронну мережу на основі історичних та симульованих даних. Мережа повинна навчитися визначати вплив спорядження, кліматичних умов та інших факторів на рівень успішності місії.

Напрямок дослідження: Дослідження архітектур нейронних мереж, які підходять для аналізу багатовимірних даних, і їхньої адаптації для роботи з військовими сценаріями. Також необхідно розглянути способи інтеграції симульованих даних у навчання мережі для підвищення точності рекомендацій.

Конкретні завдання:

- Розробити алгоритм для навчання нейронної мережі з використанням багатовимірної матриці відповідностей.
- Проводити навчання мережі на історичних даних і даних, отриманих із симуляцій бойових тренувань, щоб забезпечити точність і адаптивність рекомендацій.
- Реалізувати механізм повного перенавчання мережі для оновлення системи у випадку значних змін у структурі даних або параметрах спорядження.

1.4.3 Розробка алгоритмів для формування рекомендацій

Третя програма повинна використовувати дані нейронної мережі для надання солдатам рекомендацій щодо спорядження з урахуванням місії та умов. Основною метою є створення алгоритму, який би максимально точно визначав оптимальне спорядження для досягнення найвищого рівня успішності.

Задача покладена на цей пункт це - визначення найкращих методів для поєднання результатів нейронної мережі та таблиці відповідностей для формування рекомендацій. Особливу увагу слід приділити врахуванню кліматичних умов, тривалості місії та інших факторів, які можуть вплинути на вибір спорядження. Так реалізуючи цей пункт потрібно:

- Розробити алгоритм, який буде використовувати дані про місію (тип місії, кліматичні умови, тривалість, особисті характеристики) для надання оптимальних варіантів спорядження.
- Створити систему, яка дозволяє експерту оцінювати рекомендації та вносити корективи, що допоможе вдосконалити алгоритм на основі зворотного зв'язку.
- Перевірити алгоритм на наборі тестових сценаріїв, щоб оцінити його ефективність і точність рекомендацій.

1.4.4 Адаптація системи до нових даних і масштабування

Для того щоб система залишалася актуальною і ефективною, необхідно впровадити механізми адаптації до нових даних. Це включає як можливість оновлення бази знань, так і масштабування системи для врахування нових параметрів або додаткових даних.

Напрямок дослідження: Дослідження методів масштабування нейронних мереж та адаптації експертних систем до нових даних. Особлива увага приділяється інтеграції нових характеристик місій або спорядження з мінімальною потребою у перепроєктуванні системи.

Конкретні завдання:

- Реалізувати механізм автоматичного оновлення бази знань та навчання нейронної мережі на нових даних.
- Перевірити можливість додавання нових параметрів (наприклад, додаткового спорядження або характеристик місій) без повного перенавчання мережі, а з використанням часткових оновлень моделі.

1.4.5 Валідація та тестування системи

Тестування системи на реальних та симульованих даних є критично важливим для оцінки її ефективності. Основною метою є перевірка того, чи може система надавати точні рекомендації в різних умовах та з урахуванням різноманітних факторів.

Основна задача що ставиться перед цим пунктом це визначення ефективних методів тестування експертних систем у реальних і симульованих умовах. Необхідно розробити сценарії тестування, які відображатимуть реальні бойові ситуації. Так при вирішенні цієї задачі потрібно розробити тестовий план, який включатиме різноманітні сценарії місій і типів спорядження для оцінки точності та надійності системи, також потрібно провести тести на симульованих даних із бойових тренувань та реальних даних, отриманих з бойових дій, для оцінки точності рекомендацій;

Запропоновані напрями досліджень і конкретизовані завдання забезпечують комплексний підхід до розробки та вдосконалення експертної системи підтримки прийняття рішень щодо спорядження військовослужбовців. Вони включають створення гнучкої бази знань, навчання нейронної мережі на реальних і симульованих даних, розробку алгоритмів для формування рекомендацій, адаптацію до нових даних, а також валідацію системи. Реалізація цих завдань дозволить створити ефективну та надійну систему, яка забезпечуватиме високий рівень точності рекомендацій у складних і динамічних умовах.

Висновки

У першому розділі було проведено аналіз сучасних наукових, інноваційних та практичних здобутків у галузі експертних систем. Розглянуто основні етапи розвитку експертних систем та їхнє застосування в різних галузях, таких як медицина, промисловість, бізнес та військова сфера. Було показано, що експертні системи є потужним інструментом для автоматизації процесів прийняття рішень на основі знань і досвіду експертів.

Зокрема, було підкреслено, що ефективність експертних систем залежить від якості бази знань і правил, що використовуються для прийняття рішень. Основними проблемами залишаються оновлення та підтримка бази знань, адаптивність систем до нових даних, а також складність інтеграції з іншими системами та високі витрати на впровадження й підтримку.

У контексті розробки системи підтримки прийняття рішень щодо спорядження військовослужбовців, можна зробити висновок, що використання нейронних мереж для автоматичного створення таблиці відповідностей на основі історичних даних та експертних знань є перспективним підходом. Така система дозволить ефективно надавати рекомендації на основі аналізу різноманітних факторів, таких як умови місії, клімат, тривалість та особисті характеристики військових.

Проведений аналіз підтверджує актуальність дослідження та обґрунтовує необхідність подальших розробок у цій сфері. Подальші дослідження мають бути спрямовані на розробку гнучкої бази знань, інтеграцію нейронної мережі для створення таблиці відповідностей та забезпечення адаптивності системи до нових даних.

РОЗДІЛ 2

ІНФОРМАЦІЙНА СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО СПОРЯДЖЕННЯ ВІЙСЬКОВОСЛУЖБОВЦЯ

2.1 Обґрунтування вибору напрямку досліджень

Розробка інформаційної системи підтримки прийняття рішень щодо спорядження військовослужбовців є важливою і актуальною задачею, яка відповідає потребам сучасних військових підрозділів. На сучасному етапі розвитку військової техніки та тактики основною метою стає забезпечення ефективності та безпеки військовослужбовців під час виконання бойових завдань. Одним з ключових факторів у цьому процесі є правильний вибір спорядження для солдата, яке дозволить досягти максимального рівня успішності місії за мінімальних ризиків.

2.1.1 Актуальність розробки системи підтримки прийняття рішень щодо спорядження військовослужбовців

В умовах сучасних бойових дій та асиметричних конфліктів ефективність операцій значною мірою залежить від підготовки солдатів та їхнього оснащення. Оскільки ситуації на полі бою можуть змінюватися динамічно, а кількість можливих варіантів спорядження зростає, командири і самі військовослужбовці потребують системи, яка могла б швидко і точно підказати, яке спорядження обрати для конкретних умов.

Крім того, у зв'язку з розвитком сучасних технологій та можливістю використання великих обсягів даних, з'являється можливість автоматизувати процес аналізу даних про минулі бойові дії, спорядження та умови, у яких вони проходили, для підвищення точності рекомендацій. Це дозволяє створювати інноваційні рішення, які раніше були неможливі або обмежувалися ручним аналізом та досвідом експертів.

2.1.2 Особливості функціонування системи та її компонентів

Запропонована система складається з трьох основних компонентів:

- "База знань": Програма для збору та збереження історичних даних про місії, які містять інформацію про характеристики солдатів, погодні умови, тривалість місій, типи спорядження та їхній вплив на результат. Ця програма є основою для створення бази даних, яка забезпечує необхідну інформацію для аналізу.
- "Підготовка експерта": Програма для тренування нейронної мережі, яка на основі зібраних даних створює таблицю відповідностей, що показує вплив різних типів спорядження на успішність місій. Це дозволяє автоматизувати процес визначення оптимальних варіантів спорядження для різних умов.
- "Помічник в комплектуванні": Програма, яка на основі таблиці відповідностей та введених даних про місію надає рекомендації щодо спорядження для військовослужбовця, забезпечуючи максимально можливий рівень успішності місії.

Кожен із компонентів системи відіграє ключову роль у забезпеченні її ефективності та надійності. Взаємодія між цими компонентами забезпечує точний та своєчасний аналіз і видачу рекомендацій, що підвищує рівень підготовки військовослужбовців.

2.1.3 Визначення цілей і завдань дослідження

Основною метою дослідження є підвищення ефективності експертної системи, що автоматизує процес підбору спорядження для військовослужбовців з урахуванням багатьох факторів, включаючи погодні умови, тип місії, тривалість завдання, фізичні характеристики військовослужбовця та доступне спорядження. Система має аналізувати історичні дані та надавати рекомендації, які максимізують рівень успішності місій.

Для досягнення цієї мети необхідно вирішити наступні завдання:

- Створити програму для збору та структурування історичних даних ("База знань").

- Розробити та навчити нейронну мережу для аналізу зібраних даних та визначення впливу різних факторів на результати місій ("Підготовка експерта").
- Інтегрувати результати роботи нейронної мережі в програму, яка надаватиме рекомендації щодо спорядження ("Помічник в комплектуванні").
- Провести тестування системи на контрольних місіях для оцінки її ефективності.

Запропонований напрямок дослідження має велике практичне значення та є інноваційним у сфері військових технологій, що обґрунтовує актуальність та доцільність розробки цієї системи.

2.2 Методика проведення дослідження

Розробка системи підтримки прийняття рішень щодо спорядження військовослужбовців базується на структурованому підході до розробки програмного забезпечення та аналізу даних. У дослідженні використовується водоспадна модель розробки, яка забезпечує послідовне виконання етапів проєкту та дозволяє досягти високої якості продукту шляхом ретельного планування, проєктування та тестування кожного з компонентів системи.

2.2.1 Опис підходу до розробки програми (Водоспадна модель)

Для розробки системи використовується водоспадна модель (Waterfall Model). Ця модель передбачає послідовне виконання кожного етапу розробки, де кожен етап завершується повністю перед тим, як перейти до наступного. Це дозволяє створювати чіткий план розробки та гарантує, що кожен компонент системи буде розроблений, протестований та інтегрований відповідно до вимог.

Основні етапи водоспадної моделі:

- Збір вимог: Визначення всіх функціональних і нефункціональних вимог до системи на основі потреб користувачів та експертів. Це дозволяє сформулювати повну картину майбутньої системи і фіксує очікування від її роботи.

- **Проектування:** Розробка архітектури системи та створення загального плану взаємодії між компонентами. На цьому етапі формується бачення, як дані будуть передаватися між програмами та якими інтерфейсами вони будуть оснащені.
- **Реалізація:** Написання коду для кожної програми згідно з планом, визначеним на етапі проектування. Реалізація відбувається відповідно до вибраного технологічного стеку, що буде обґрунтований у наступних пунктах.
- **Тестування:** Перевірка роботи кожної програми окремо, а також тестування всієї системи на предмет коректності взаємодії компонентів. Цей етап включає тестування на контрольних місцях і перевірку точності результатів.
- **Впровадження і підтримка:** Після успішного тестування система впроваджується в експлуатацію, де також здійснюється моніторинг її роботи та оновлення бази даних за допомогою програми "База знань".

2.2.2 Етапи розробки: збір вимог, проектування, реалізація, тестування

Аналіз вимог до функціональності системи, включаючи збір даних, аналіз зібраної інформації, та видачу рекомендацій. Оцінка критеріїв ефективності та точності системи.

Визначення архітектури системи, включаючи структуру бази даних для зберігання історичних даних, архітектуру нейронної мережі для аналізу даних, а також загальні інтерфейси для взаємодії з користувачем.

Реалізація кожного компонента системи відповідно до вибраного технологічного стеку. Цей пункт не деталізує вибір фреймворків і технологій, а зосереджується на послідовності дій у процесі реалізації.

Оцінка функціональності кожної програми і тестування інтеграції між компонентами системи. Використання контрольних місць для перевірки точності рекомендацій.

2.2.3 Взаємодія між програмами та її важливість

Кожна з програм у системі має свою унікальну роль:

- "База знань": Програма для збору даних та збереження їх у централізованій базі. Це є основою системи, оскільки на основі цих даних будуються всі наступні етапи.
- "Підготовка експерта": Програма для аналізу зібраних даних за допомогою нейронної мережі та створення таблиці відповідностей. Цей компонент дозволяє визначити взаємозв'язок між умовами місій та спорядженням.
- "Помічник в комплектуванні": Програма, яка використовує створену таблицю відповідностей для надання рекомендацій щодо спорядження. Цей компонент є кінцевим елементом системи, що забезпечує користувачів необхідною інформацією для ефективного виконання місії.

Система побудована таким чином, що кожна програма може працювати незалежно після попередньої підготовки даних і таблиці відповідностей, що дозволяє забезпечити гнучкість і ефективність у використанні.

2.3 Вибір нейронної мережі для обробки даних

У процесі розробки експертної системи для підтримки прийняття рішень щодо спорядження військовослужбовців важливо було обрати таку нейронну мережу, яка зможе ефективно обробляти вхідні дані та генерувати таблицю відповідностей. Враховуючи особливості завдання, було детально проаналізовано кілька типів нейронних мереж, щоб визначити найбільш підходящу архітектуру для програми "Підготовка експерта".

2.3.1 Огляд різних архітектур нейронних мереж

Для вирішення поставленої задачі було розглянуто декілька варіантів нейронних мереж:

Багатошарова перцептронна мережа (Multilayer Perceptron, MLP):

Це класична архітектура, яка складається з декількох шарів: вхідного, одного або кількох прихованих, а також вихідного шару. Кожен шар складається з нейронів, що застосовують активаційні функції до вхідних даних[6].

До переваг цієї архітектури можна віднести те що MLP підходить для задач, де дані представлені у вигляді таблиць і не мають тимчасової структури. Вона відносно проста у реалізації та добре інтегрується з фреймворками на кшталт PyTorch. Така архітектура забезпечує високу точність для аналізу статичних даних.

Основними ж недоліками що було виділено є те що цей тип мереж не підходить для аналізу даних, що мають послідовність або тимчасові залежності, оскільки MLP не зберігає попередній контекст під час обробки даних.

Рекурентна нейронна мережа (Recurrent Neural Network, RNN):

RNN має зворотні зв'язки, що дозволяють зберігати інформацію про попередні стани та враховувати їх під час обробки. Це робить її корисною для аналізу даних, де важлива послідовність подій, таких як часові ряди[6].

Цей тип нейронних мереж добре підходить для задач, де потрібно зберігати і враховувати попередні стани або працювати з послідовностями, але враховуючи, що дані у моєму випадку є статичними і не мають послідовної структури, використання RNN не є оптимальним рішенням. Крім того, RNN має проблеми з затуханням градієнта, що може ускладнити процес навчання.

Згортова нейронна мережа (Convolutional Neural Network, CNN):

CNN, зазвичай, використовується для роботи з зображеннями, оскільки вона вміє виділяти просторові особливості та шаблони в даних. Проте її також можна адаптувати для роботи з табличними даними[6].

Такий тип нейронних мереж має один позитивний момент – він ефективно обробляє дані, що мають просторову структуру або шаблони. Але в цей же час адаптація CNN для роботи з табличними даними у моєму випадку потребує додаткових зусиль, що ускладнює процес розробки. Крім того, використання CNN не дасть особливих переваг для табличних даних, які є у завданні.

Глибока нейронна мережа (Deep Neural Network, DNN):

Це багат шарова нейронна мережа, що складається з більшої кількості шарів, ніж звичайна MLP. Завдяки цьому вона здатна моделювати складні та багатовимірні закономірності в даних, які не можуть бути ефективно виявлені простішими моделями. Висока глибина дозволяє DNN опрацьовувати великі обсяги інформації, враховуючи навіть найменші залежності між змінними, що особливо корисно для аналізу зображень, відео, тексту та інших складних структурованих даних.

Висока точність, яку забезпечують глибокі нейронні мережі, обумовлює їх широке застосування в таких галузях, як комп'ютерний зір, розпізнавання мови.

Попри свої переваги, DNN має й обмеження. Зокрема, використання глибоких нейронних мереж може бути надмірним для простих задач, де вхідні дані мають табличну структуру і містять лінійні або прості нелінійні залежності[7].

У таблиці нижче наведено основні характеристики, переваги та недоліки різних типів нейронних мереж, що дає змогу обрати найбільш відповідну архітектуру для конкретної задачі.

Таблиця 2.3.1

Порівняльна характеристика архітектур нейронних мереж

Характеристика	MLP	RNN	CNN	DNN
Тип даних	Табличні дані без тимчасової структури	Послідовні дані (часові ряди)	Зображення або дані з просторовою структурою	Складні структури даних
Підтримка послідовних даних	-	+	-	-
Збереження контексту	-	+	-	-
Простота реалізації	+	-	-	-

Таблиця 2.3.1 (продовження)

Порівняльна характеристика архітектур нейронних мереж

Характеристика	MLP	RNN	CNN	DNN
Потреба в ресурсах та складність навчання	Низька	Висока	Середня	Висока
Використання у статичних задачах	+	-	+	+
Використання для складних задач	-	+	+	+
Проблеми навчання	Легке перенавчання	Затухання градієнта	Потребує налаштувань	Вимагає великого обсягу даних

2.3.2. Обґрунтування вибору багатошарової перцептронної мережі (MLP)

Після аналізу всіх варіантів було вирішено зупинитися на багатошаровій перцептронній мережі (MLP), оскільки вона найкраще відповідає вимогам мого завдання. Основними причинами такого вибору стали:

- Зручність роботи з табличними даними: Оскільки вхідні дані представлені у вигляді таблиць, де кожен рядок — це набір характеристик місії та спорядження, MLP є оптимальним варіантом, адже вона ефективно обробляє саме такі дані.
- Простота реалізації: MLP легко реалізується за допомогою PyTorch, який було обрано як фреймворк для цієї програми. Це дозволить швидко налаштувати модель та протестувати її на історичних даних.

- Ефективність для статичних даних: У задачі, де немає залежності між подіями у часі, MLP забезпечує стабільну точність[7] та ефективність, що робить її найкращим вибором.

2.4 Методи навчання та перевірки коректності роботи нейронної мережі.

Важливим етапом розробки програми "Підготовка експерта" є навчання нейронної мережі та перевірка її коректності для забезпечення точних і надійних рекомендацій. У цьому розділі розглядаються методи навчання, які використовуються для тренування нейронної мережі, а також підходи до перевірки її коректності та оцінки якості роботи.

2.4.1 Пояснення методів навчання нейронних мереж

Для навчання нейронних мереж використовуються різні методи оптимізації, які дозволяють мережі мінімізувати похибки та адаптувати свої ваги таким чином, щоб покращити точність передбачень. Нижче наведені основні методи навчання, які були розглянуті для моєї програми:

Гرادієнтний спуск (Gradient Descent):

Основний метод навчання нейронних мереж, що використовує градієнт функції помилки для оновлення ваг. Метод корегує ваги у напрямку, який зменшує похибку моделі. Зазвичай він використовується у трьох основних варіантах[7]:

Batch Gradient Descent: Оновлює ваги після обчислення градієнта на всьому наборі даних. Це дає точні оновлення, але вимагає великих обчислювальних ресурсів.

Stochastic Gradient Descent (SGD): Оновлює ваги після кожного зразка даних. Це значно зменшує час навчання, проте може спричиняти коливання і нестабільність у процесі навчання.

Mini-Batch Gradient Descent: Компроміс між двома попередніми методами, де ваги оновлюються після обробки невеликої підмножини даних. Це дозволяє збалансувати точність і швидкість навчання.

Цей метод є простим в реалізації та має високу ефективність для невеликих наборів даних але може потрапити у локальні мінімуми та потребує значних обчислювальних ресурсів для великих наборів даних.

Метод Adam (Adaptive Moment Estimation):

Метод Adam є адаптивною версією градієнтного спуску, який поєднує переваги адаптивного градієнта та моменту. Adam автоматично змінює швидкість навчання для кожного параметра на основі середнього значення та дисперсії градієнта. Це дозволяє досягти більш швидкої збіжності та стабільності[9].

До його плюсів можна віднести швидке збіження навіть для великих і складних моделей та стабільність під час навчання завдяки врахуванню попередніх градієнтів та моменту.

Проте він може бути менш стабільним, якщо параметри не налаштовані правильно, і потребує більше обчислень порівняно з простим градієнтним спуском.

RMSProp:

Root Mean Square Propagation) є одним із найпопулярніших адаптивних методів оптимізації, широко застосовуваних у навчанні нейронних мереж. Основна ідея цього методу полягає в тому, щоб автоматично змінювати швидкість навчання для кожного параметра моделі залежно від середньоквадратичної похибки градієнта. Це дозволяє більш ефективно управляти оновленням параметрів, особливо в задачах, де градієнти мають значні варіації або суттєво змінюються протягом навчання.

Однією з основних переваг RMSProp є його здатність добре працювати в ситуаціях, коли традиційні методи, такі як стандартний градієнтний спуск, зазнають труднощів.

Цей метод також добре зарекомендував себе у випадках, коли оптимізація проводиться на великих наборах даних, оскільки він забезпечує швидшу та стабільнішу конвергенцію порівняно з деякими іншими підходами.

Одним із мінусів є необхідність налаштування гіперпараметрів, таких як швидкість навчання та коефіцієнт згладжування. Неправильний вибір цих значень може призвести до повільної конвергенції або навіть до неможливості досягнення глобального мінімуму.

Нижче наведено порівняльну таблицю основних методів навчання нейронних мереж.

Таблиця 2.4.1

Порівняльна таблиця для методів навчання нейронних мереж

Характеристика	Гرادієнтний спуск	Метод Adam	RMSProp
Тип оновлення ваг	Після обчислення градієнта (Batch, SGD, Mini-Batch)	Адаптивне оновлення ваг на основі градієнта та моменту	Адаптивне оновлення на основі середньоквадратичної похибки градієнта
Швидкість збіжності	Середня	Висока	Середня
Стабільність навчання	Низька для SGD, висока для Batch	Висока завдяки моментам	Висока при правильному налаштуванні
Вимоги до ресурсів	Високі для Batch, середні для Mini-Batch, низькі для SGD	Високі	Середні
Адаптивність до градієнтів	Немає	Адаптивність швидкості навчання	Адаптивна зміна швидкості навчання

Таблиця 2.4.1(продовження)

Порівняльна таблиця для методів навчання нейронних мереж

Характеристика	Градiєнтний спуск	Метод Adam	RMSProp
Проблеми навчання	Може потрапити у локальні мінімуми	Може бути нестабільним при неправильних параметрах	Потребує налаштування параметрів для досягнення оптимальних результатів
Застосування для великих наборів даних	Високі витрати ресурсів (Batch)	Ефективний для великих наборів даних	Добре працює з великими наборами даних
Простота реалізації	Відносно проста	Відносно складна	Середня
Підтримка складних моделей	Обмежена	Підтримує складні моделі	Добре працює зі змінюваними градієнтами

2.4.2 Вибір оптимізаційного методу (Adam) для навчання мережі

Для навчання MLP було обрано метод оптимізації Adam, який дозволяє швидко й стабільно тренувати модель. Adam автоматично адаптує швидкість навчання для кожного параметра, що дозволяє моделі швидше знаходити оптимальні значення.

У порівнянні з іншими методами (наприклад, стохастичний градієнтний спуск), Adam швидше знаходить оптимальне рішення, що важливо при роботі з великими наборами даних[10].

Використання моментів дозволяє знижувати коливання та забезпечує стабільне навчання моделі. Adam ефективно працює з великими наборами даних,

що дозволяє нейронній мережі швидко адаптуватися до нових даних та оновлювати таблицю відповідностей.

Вибір MLP у поєднанні з методом оптимізації Adam дозволить ефективно створювати і оновлювати таблицю відповідностей для подальшого використання в програмі "Помічник в комплектуванні". Такий підхід забезпечить швидку і точну обробку даних та адаптацію системи до нових умов.

2.4.3 Опис методів перевірки коректності

Для забезпечення точності роботи нейронної мережі та коректності її висновків важливо використовувати різні підходи до перевірки, які допоможуть виявити та виправити потенційні проблеми. Основні методи перевірки коректності включають:

Тестування на контрольних місіях:

Після того як нейронна мережа навчиться на історичних даних, її роботу перевіряють на контрольних місіях, де результат уже відомий. Це дозволяє оцінити, наскільки модель коректно передбачає вплив спорядження на успішність місії.

Такий підхід дає можливість порівняти результати роботи моделі з реальними даними та оцінити її точність у реальних сценаріях.

Крос-валідація (Cross-Validation):

Використовується метод крос-валідації (наприклад, 5-fold cross-validation), де дані розділяються на кілька частин, і кожна з них по черзі використовується як тестовий набір, тоді як інші частини використовуються для навчання. Це дозволяє оцінити здатність моделі узагальнювати дані і знизити ризик перенавчання[11].

При використанні даного методу знижується ймовірність перенавчання і підвищується надійність оцінки точності моделі.

Порівняння з еталонними даними:

Використовуються дані, які не були задіяні під час навчання, щоб перевірити здатність моделі правильно передбачати результати. Такі дані можуть бути

отримані з симуляцій або нових місій. Порівняння результатів дозволяє оцінити точність і адекватність моделі. Вагомим плюсом методу можна назвати те що він оцінює здатність моделі адаптуватися до нових умов і даних, що важливо для подальшого використання.

Аналіз помилок:

Після тестування проводиться аналіз помилок моделі, щоб виявити слабкі місця та налаштувати модель для покращення її точності. Це дозволяє виявити закономірності або фактори, що можуть негативно впливати на результати моделі. Що дозволяє систематично покращувати модель та мінімізувати похибки, підвищуючи її надійність.

2.4.4 Вибір основних метрик ефективності для оцінки якості системи

Для оцінки якості роботи нейронної мережі обрано основну метрику — Індекс ефективності, яка вимірює, наскільки рекомендоване спорядження підвищує успішність місій у порівнянні з базовим або випадковим спорядженням. Метрика дозволяє визначити ефективність рекомендацій та оцінити, наскільки добре нейронна мережа узгоджує дані про місії з оптимальними наборами спорядження.

Індекс ефективності (Effectiveness Index): Порівнюється середній рівень успішності місій з рекомендованим спорядженням і з базовим, що дозволяє оцінити якісну зміну результатів місій та підтвердити коректність роботи нейронної мережі.

Для оцінки точності навчання використовуються методи крос-валідації та тестування на контрольних місіях, що гарантує надійність результатів та підтверджує адекватність рекомендацій, створених програмою "Помічник в комплектуванні".

Ці підходи забезпечують комплексну перевірку коректності та ефективності роботи нейронної мережі, дозволяючи вчасно виявляти та виправляти потенційні помилки для досягнення високої точності системи.

2.5 Стек технологій для реалізації системи

У цьому розділі обґрунтовується вибір технологічного стеку для реалізації кожної з програм системи підтримки прийняття рішень щодо спорядження військовослужбовців. Кожна програма системи виконує свою унікальну роль, тому технології були обрані таким чином, щоб забезпечити їх оптимальну роботу і взаємодію між компонентами.

2.5.1 Опис технологій для реалізації кожної з програм

Система складається з трьох основних програм: "База знань", "Підготовка експерта" і "Помічник в комплектуванні". Для кожної з них обрано найбільш відповідні технології:

"База знань":

- Мова програмування: C#
- Фреймворк: WPF (Windows Presentation Foundation)
- База даних: SQL Server

Опис і обґрунтування: Для створення "Бази знань" використовується мова програмування C# у поєднанні з WPF для побудови графічного інтерфейсу користувача. WPF обраний через його гнучкість у створенні інтерфейсів, а також можливість легкої інтеграції з базами даних. SQL Server забезпечує збереження великого обсягу даних про місії, що дозволяє легко їх витягувати та обробляти у подальшому.

"Підготовка експерта":

- Мова програмування: Python
- Фреймворк: PyTorch

Опис і обґрунтування: Для реалізації програми, яка відповідає за навчання нейронної мережі, обрано Python у поєднанні з фреймворком PyTorch. Python забезпечує простоту написання коду і можливість швидкого тестування, а PyTorch є зручним інструментом для створення та навчання нейронних мереж[12,13]. PyTorch надає велику гнучкість для налаштування моделі та

можливість швидкої адаптації до змін у даних. Це важливо, оскільки програма працює з великим обсягом історичних даних, які потребують ефективної обробки і навчання.

"Помічник в комплектуванні":

- Мова програмування: C#
- Фреймворк: WPF (Windows Presentation Foundation)
- База даних: SQL Server

Опис і обґрунтування: Як і "База знань", програма "Помічник в комплектуванні" написана на C# з використанням WPF для створення інтерфейсу, що дозволяє легко вводити параметри місії та отримувати рекомендації. WPF дозволяє створювати інтерактивні форми та легко підключати їх до бази даних SQL Server, де зберігається таблиця відповідностей, створена програмою "Підготовка експерта".

2.5.2 Вибір фреймворку PyTorch для тренування нейронної мережі

Після аналізу можливих варіантів фреймворків для тренування нейронної мережі, що генерує таблицю відповідностей, був обраний PyTorch адже він забезпечує інтуїтивний інтерфейс, що дозволяє легко створювати, налаштовувати та тренувати нейронні мережі.

Оскільки "Підготовка експерта" потребує адаптації до різних типів даних і ситуацій, PyTorch дозволяє швидко тестувати нові конфігурації та методи оптимізації, такі як Adam, що використовуватиметься у моделі[13].

Також Фреймворк оптимізований для роботи з великими обсягами даних, що дозволяє ефективно тренувати модель на історичних даних місій[14].

2.5.3 Порівняння фреймворків та їх переваги для кожного етапу розробки

Під час вибору фреймворків для розробки системи було порівняно декілька варіантів для кожної програми так для "Бази знань" і "Помічника в комплектуванні" розглядалися фреймворки Windows Forms та WPF. Було вирішено використовувати WPF, оскільки він забезпечує більшу гнучкість і дозволяє створювати сучасні, привабливі інтерфейси з використанням XAML.

Крім того, WPF забезпечує кращу інтеграцію з SQL Server для зберігання даних, що було важливим критерієм.

Для програми "Підготовки експерта" порівнювали TensorFlow та PyTorch. Вибір був зроблений на користь PyTorch, оскільки він більш гнучкий для розробки, зручний для дослідницьких експериментів і дозволяє швидко тестувати моделі. Крім того, PyTorch має простіший API, що дозволяє інтегрувати нові моделі або налаштовувати існуючі з меншими витратами часу[15,16].

Таким чином, вибір технологій для кожного з компонентів системи був обґрунтований на основі їхньої сумісності, ефективності та можливостей для швидкої адаптації та розвитку.

2.6 Оцінка похибок та методи покращення точності

Для розробки ефективної нейронної мережі у програмі "Підготовка експерта" необхідно постійно оцінювати якість її роботи та знаходити способи покращення точності. У цьому розділі детально розглянуто, як аналізуються похибки, що виникають під час навчання моделі, та які методи застосовуються для їх усунення.

2.6.1 Оцінка впливу якості даних на результати системи та методи боротьби з похибками

Похибки в нейронній мережі можуть виникати з різних причин, серед яких: низька якість даних, неповнота набору даних, або ж неправильне налаштування параметрів навчання. Для ефективної роботи програми важливо враховувати всі ці фактори і застосовувати методи, що мінімізують їх вплив.

Аналіз якості даних:

Дані, які використовуються для навчання нейронної мережі, повинні бути максимально повними і точними. Для цього під час збору даних у програмі "База

знань" важливо переконатися, що всі параметри місії введені правильно і відповідають реальності.

Методи контролю якості даних: Запровадження перевірок коректності введення даних у програмі "База знань" дозволяє зменшити ризик появи помилок через людський фактор.

Аналіз впливу розміру і варіативності набору даних:

Щоб нейронна мережа могла ефективно узагальнювати і працювати з різними ситуаціями, необхідно забезпечити великий і різноманітний набір даних. Недостатня кількість даних або надмірна схожість між ними може призвести до того, що модель стане перенавченою або, навпаки, недостатньо точною.

Методи боротьби з недостатньою кількістю даних: Додавання синтетичних даних, отриманих із симуляцій бойових тренувань, допомагає розширити вибірку та збільшити варіативність. Також можна застосовувати методи аугментації, які дозволяють змінювати існуючі дані для збільшення їхньої кількості.

2.6.2 Аналіз результатів навчання нейронної мережі та візуалізація впливу спорядження на успішність місії

Аналіз результатів навчання моделі допомагає виявити, які аспекти потребують покращення, і зрозуміти, як саме працює нейронна мережа. Для цього застосовуються такі підходи:

Візуалізація помилок моделі:

Після кожного етапу навчання проводиться оцінка точності на тестових даних, що дозволяє побачити, як часто модель помиляється і які саме фактори (наприклад, погодні умови, тип спорядження) викликають найбільше проблем.

Графічна візуалізація, наприклад, у вигляді графіків навчальної та тестової похибок, дозволяє оцінити, чи модель стабільно навчається, чи виникають проблеми з перенавчанням або недонавчанням.

Аналіз впливу спорядження на успішність місії:

Використовуючи таблицю відповідностей, створену нейронною мережею, можна візуалізувати, які типи спорядження найбільше впливають на успішність

місії в різних умовах. Це дозволяє зрозуміти, як мережа оцінює ефективність різних елементів спорядження.

Візуалізація у вигляді діаграм або теплових карток (heatmaps) допомагає побачити взаємозв'язок між змінними (наприклад, тип спорядження та погодні умови) та вплив на результат місії.

2.6.3 Покращення моделей на основі аналізу помилок і налаштування параметрів системи для підвищення точності

На основі аналізу помилок та результатів навчання моделі застосовуються коригування і налаштування параметрів для покращення якості роботи нейронної мережі:

Коригування параметрів навчання:

Якщо модель демонструє ознаки перенавчання, можна зменшити кількість епох навчання або використовувати техніки регуляризації, такі як Dropout. Це дозволяє моделі зберігати узагальнюючі властивості і не фокусуватися на особливостях конкретних прикладів з навчального набору даних.

Якщо, навпаки, модель показує недостатню точність, слід збільшити кількість епох або кількість нейронів у прихованих шарах, щоб збільшити здатність моделі до навчання.

Оновлення структури нейронної мережі:

У випадках, коли мережа не може адаптуватися до певних типів даних, можна змінити кількість шарів або використовувати інші активаційні функції. Це допомагає змінити архітектуру таким чином, щоб вона краще працювала з даними, специфічними для місій.

Також, у випадку виявлення надмірної кореляції між певними типами даних (наприклад, типами місій та погодними умовами), можна використовувати методи зменшення вимірності даних, такі як PCA (Principal Component Analysis).

Періодичне перенавчання моделі:

Для підтримки актуальності системи та врахування нових даних, що постійно збираються програмою "База знань", проводиться періодичне перенавчання

нейронної мережі. Це дозволяє інтегрувати нові дані про місії та відповідним чином оновлювати таблицю відповідностей, що використовується "Помічником в комплектуванні".

Застосування цих методів дозволяє підтримувати високу якість і точність роботи системи, а також оперативно адаптувати модель до змін і нових даних.

Висновки

У другому розділі магістерської роботи було детально розглянуто методологію проведення дослідження та обґрунтовано вибір архітектури нейронної мережі для реалізації системи підтримки прийняття рішень щодо спорядження військовослужбовців. Було обрано багат шарову перцептронну мережу (MLP) як оптимальну для аналізу статичних табличних даних, оскільки вона відповідає вимогам задачі і забезпечує високу точність. Для оптимізації навчання мережі використано метод Adam, що дозволяє ефективно і стабільно тренувати модель на великих наборах даних.

Також у розділі було обґрунтовано вибір технологічного стеку для кожної з програм, що складають систему. Програми "База знань" та "Помічник в комплектуванні" реалізовано на C# з використанням фреймворку WPF, а "Підготовку експерта" — на Python з використанням PyTorch. Вибір цих технологій забезпечує гнучкість у розробці та інтеграції, а також можливість швидкої адаптації до нових умов і даних.

Окрім цього, розділ включає детальний аналіз методів оцінки коректності роботи системи та нейронної мережі. Для перевірки точності було запропоновано використання тестування на контрольних місцях та крос-валідації, що дозволяє знизити ризик перенавчання та підвищити надійність оцінки результатів. Також було описано методи боротьби з похибками та шляхи покращення точності системи на основі аналізу помилок.

Таким чином, було обґрунтовано технологічні та методологічні рішення, необхідні для створення ефективної системи підтримки прийняття рішень. Це забезпечує основу для реалізації і тестування компонентів системи у наступних етапах дослідження.

РОЗДІЛ 3

ТЕХНІЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Розробка архітектури

Архітектура інформаційної системи, призначеної для підтримки прийняття рішень щодо спорядження військовослужбовця, є ключовим аспектом її розробки. Вона побудована на модульному підході, що дозволяє забезпечити гнучкість, масштабованість і зручність у використанні. Система складається з трьох основних компонентів: "База знань", "Підготовка експерта" та "Помічник у комплектуванні". Кожен із цих модулів виконує свою унікальну роль у загальній структурі, спрямованій на досягнення основної мети — оптимізацію спорядження військовослужбовців.

Перший модуль, "База знань", є серцем системи, оскільки він відповідає за збір та зберігання всіх історичних даних. Цей модуль надає користувачам інструменти для введення інформації про місії, характеристики солдатів та обладнання. Для реалізації цього компонента використовується мова програмування C# у поєднанні з WPF, що забезпечує зручний інтерфейс[17].

Дані зберігаються в централізованій базі даних на SQL Server для забезпечення цілісності та доступності для інших модулів. Завдяки "Базі знань" користувач може не тільки вводити дані, але і редагувати їх і перевіряти їх точність, що робить систему надійною і простий у використанні. Другий модуль, "Підготовка експерта", має інше завдання — це модуль, що аналізує історичні дані та формує таблицю відповідності спорядження. Саме цей компонент системи використовує сучасні методи машинного навчання.

Основою модуля є багатошарова перцептронна нейронна мережа, реалізована за допомогою мови Python та бібліотеки PyTorch. Ця мережа аналізує всі доступні дані, враховує характеристики місії, параметри солдатів і вибране спорядження, щоб створити таблицю, яка відображає вплив кожного елемента спорядження на

успішність місії. Результати роботи модуля зберігаються у форматі CSV, що дозволяє їх легко інтегрувати з іншими компонентами системи.

Третій модуль, "Помічник у комплектуванні", забезпечує кінцеву мету системи — надання користувачу рекомендацій щодо спорядження. Цей модуль також реалізований на мові C# із використанням WPF для створення інтуїтивного інтерфейсу[18]. Завдяки даним із таблиці відповідностей, отриманій від другого модуля, "Помічник у комплектуванні" може швидко надати користувачу список оптимального спорядження для конкретної місії, враховуючи параметри солдата. Крім того, цей модуль дозволяє експортувати отримані результати для подальшого використання.

Архітектура системи побудована таким чином, щоб забезпечити чітке розмежування функцій між модулями, що дозволяє кожному з них виконувати свої завдання незалежно. Це забезпечує високу гнучкість і масштабованість. Центральним елементом, що об'єднує всі компоненти, є база даних. Вона гарантує, що дані, введені у "Базу знань", можуть бути використані "Підготовкою експерта" для аналізу, а результати цього аналізу — застосовані у "Помічнику у комплектуванні".

Для наочності архітектуру системи можна уявити як багаторівневу структуру. На верхньому рівні працює "База знань", яка отримує дані від користувача. Ці дані потрапляють у централізовану базу SQL Server, яка є основою для роботи другого модуля. Далі "Підготовка експерта" аналізує зібрані дані, генерує таблицю відповідностей і зберігає її у вигляді CSV-файлу. Цей файл використовується "Помічником у комплектуванні", який працює безпосередньо з користувачем, забезпечуючи його необхідними рекомендаціями.

Модульна архітектура має багато переваг. По-перше, вона забезпечує гнучкість, дозволяючи оновлювати чи масштабувати окремі компоненти системи без втрати загальної функціональності. По-друге, така архітектура спрощує розробку та тестування, оскільки кожен модуль можна перевіряти окремо. По-третє, це забезпечує інтуїтивне використання системи для кінцевого користувача.

3.2 Загальна структура системи

Розробка інформаційної системи для підтримки прийняття рішень щодо спорядження військовослужбовця є складним, але надзвичайно важливим завданням. Вона охоплює всі ключові етапи аналізу, планування та реалізації технічного рішення, яке дозволяє не лише підвищити ефективність виконання бойових завдань, а й зменшити ризики для військових. Для досягнення цієї мети була створена інтегрована система, що складається з трьох основних модулів, кожен із яких виконує свою унікальну функцію.

3.2.1 Архітектура системи: єдність та модульність

Архітектура інформаційної системи будується на принципі модульності. Це означає, що кожен компонент виконує чітко визначену функцію, а всі модулі працюють у взаємодії, забезпечуючи злагоджену роботу системи.

Основу системи складають три модулі:

"База знань" – модуль для збору та збереження історичних даних про місії.

"Підготовка експерта" – аналітичний модуль, який здійснює аналіз зібраних даних та створює таблицю відповідностей.

"Помічник в комплектуванні" – рекомендаційний модуль, який надає солдатам персоналізовані рекомендації щодо спорядження для конкретної місії.

3.2.2. Взаємодія між модулями

Уся інформаційна система побудована таким чином, щоб забезпечити злагоджену взаємодію між її модулями. Головним принципом її архітектури є модульність, яка дозволяє кожному компоненту функціонувати автономно, водночас забезпечуючи ефективний обмін даними між ними. Це забезпечує не лише зручність у роботі, а й можливість подальшого вдосконалення системи без порушення її загальної функціональності.

Основу всієї системи складає модуль "База знань". Його можна назвати ядром, оскільки саме тут зберігаються всі дані, необхідні для роботи інших компонентів.

У "Базі знань" міститься інформація про місії: їхні характеристики,

тривалість, погодні умови, особливості місцевості. Окрім цього, тут є детальні відомості про солдатів, які брали участь у цих місіях, включаючи їхній вік, зріст, вагу, звання. Унікальною частиною цих даних є спорядження, яке використовувалося під час місії, разом із зазначенням його кількості та типу. Ця інформація є своєрідним "паливом" для роботи всієї системи, адже вона використовується як основа для аналізу в наступному модулі.

Другий модуль системи, що отримав назву "Підготовка експерта", працює безпосередньо з даними, які надходять із "Бази знань". Основне завдання цього модуля — аналіз історичних даних про місії та створення таблиці відповідностей, яка показує, як саме те чи інше спорядження впливає на успішність місій. Використовуючи сучасні методи машинного навчання, такі як багатосарова персептронна нейронна мережа та метод оптимізації Adam, модуль генерує результати, які є ключовими для подальшої роботи всієї системи. Таблиця відповідностей, створена другим модулем, містить детальну інформацію про вплив кожного елемента спорядження в контексті певного типу місії та характеристик солдата.

Фінальним компонентом системи є модуль "Помічник в комплектуванні". Саме він працює з кінцевим користувачем, забезпечуючи надання практичних рекомендацій. Користувач вводить дані про майбутню місію, такі як тип завдання, погодні умови, тривалість, а також інформацію про солдата, який виконуватиме це завдання. Використовуючи таблицю відповідностей, створену другим модулем, "Помічник в комплектуванні" генерує список спорядження, який є оптимальним для досягнення максимальної ймовірності успіху місії. Завдяки інтерактивному інтерфейсу користувач може отримати результат швидко та зручно, що підвищує ефективність прийняття рішень.

Однією з ключових особливостей системи є можливість постійного оновлення даних у "Базі знань". Це означає, що система може адаптуватися до нових умов і типів завдань, гарантуючи актуальність своїх рекомендацій. Наприклад, якщо з'являються нові типи спорядження або змінюються кліматичні умови в зоні виконання завдань, ці дані можна оперативно внести до бази. У результаті

"Підготовка експерта" зможе врахувати ці зміни під час формування нової таблиці відповідностей.

Крім того, модульна структура системи дозволяє здійснювати модернізацію кожного з компонентів окремо. Наприклад, якщо буде вдосконалено алгоритми нейронної мережі у другому модулі або зміниться формат представлення даних у "Базі знань", це не порушить роботи інших компонентів. Такий підхід забезпечує довгострокову гнучкість і масштабованість системи, що особливо важливо в умовах швидкої зміни технологій та потреб користувачів.

Для покращення розуміння взаємодії між модулями доцільно включити діаграму, яка відображатиме потік даних у системі. Наприклад, можна показати, як дані з "Базі знань" надходять у "Підготовку експерта", а результати роботи останнього використовуються в "Помічнику в комплектуванні". Також варто додати таблицю, яка ілюструє, як кожен із модулів взаємодіє з іншими, та включити приклади коду, що демонструють ці взаємозв'язки.

Таким чином, взаємодія між модулями побудована таким чином, щоб забезпечити їхню автономність і водночас ефективність у досягненні загальної мети системи — оптимізації спорядження військовослужбовців для максимальної ефективності виконання завдань.

3.3 Реалізація модулів системи

У цьому підрозділі буде детально розглянуто технічну реалізацію ключових модулів системи. Особливу увагу приділимо структурі модулів, їхньому функціоналу, а також використаним методам і технологіям. Реалізація базується на модульному підході, що забезпечує легкість масштабування та підтримки.

3.3.1 Модуль "База знань"

Модуль "База знань" є фундаментальною частиною системи, оскільки він відповідає за збір, обробку та збереження історичних даних. Реалізація даного модуля включає наступні етапи:

Дані про місії, солдатів та спорядження зберігаються у взаємопов'язаних таблицях SQL Server. Таблиці спроектовані з урахуванням нормалізації даних, що забезпечує їх цілісність.

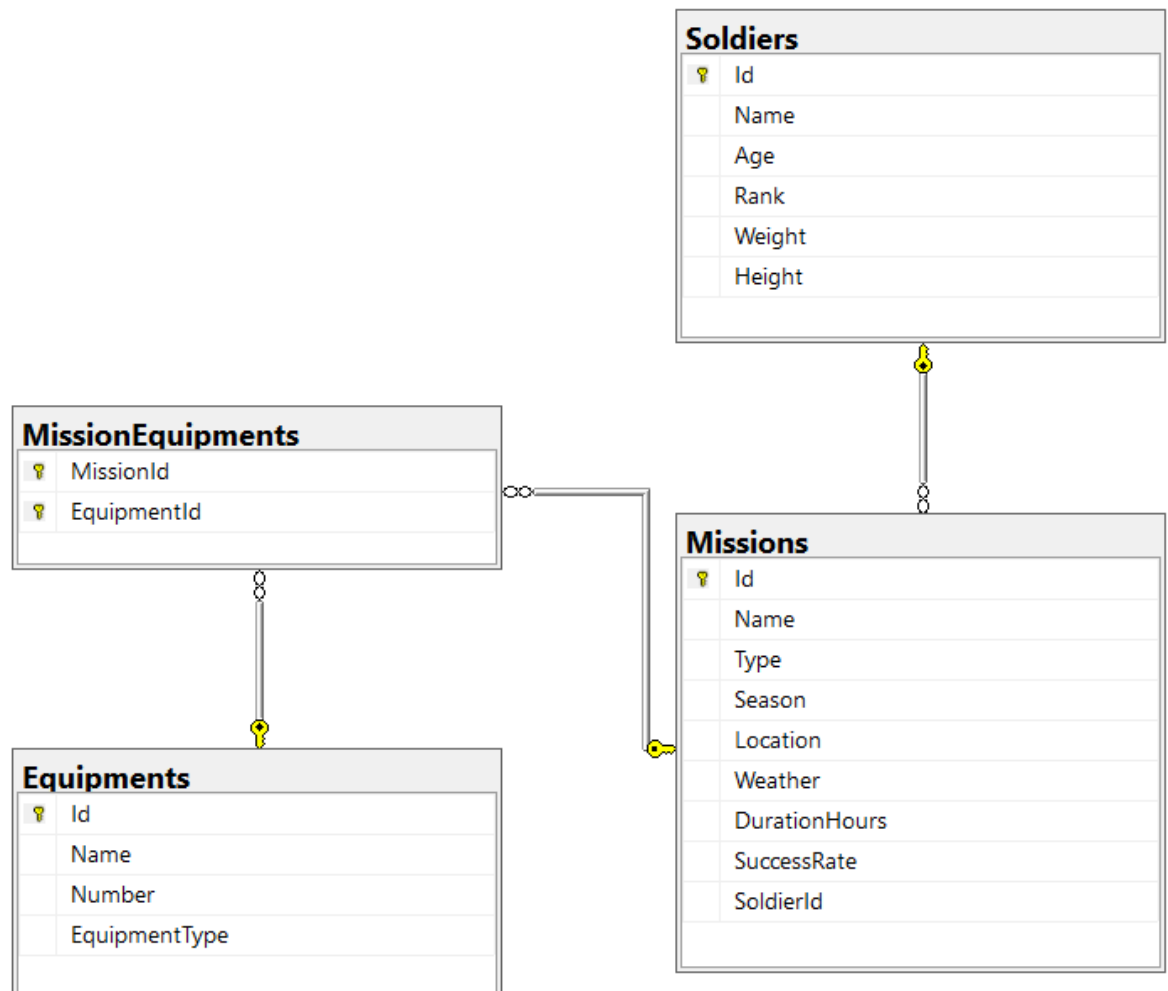


Рис.3.1. Діаграма бази даних для модуля "База знань".

Для введення даних було вирішено розробити WPF-додаток, який має змогу забезпечити просте та зручне додавання даних за допомогою форм, та

надає інструменти для кращої і швидшої перевірки коректності введення через вбудовані валідації. Нижче представлений приклад інтерфейсу користувача.

Солдати Спорядження Місії

Позивний:

Вік:

Звання:

Вага (кг):

Зріст (см):

Додати солдата

Рис.3.2. Форма для заповнення історичних даних про солдата.

Солдати Спорядження Місії

Додати нове спорядження

Назва спорядження:

Кількість:

Тип спорядження:

Додати спорядження

Рис.3.3. Форма для заповнення історичних даних про спорядження.

Солдати Спорядження Місії

Назва місії:

Тип місії:

Сезон:

Локація:

Погодні умови:

Тривалість (години):

Рівень успішності (0-100):

Солдат:

Спорядження:

Додати місію

Рис.3.4. Форма для заповнення історичних даних про перебіг місії.

3.3.2 Модуль "Підготовка експерта"

Модуль "Підготовка експерта" є ключовим компонентом системи підтримки прийняття рішень для військовослужбовців. Його основна функція полягає у наданні рекомендацій щодо оптимального спорядження для майбутніх місій, базуючись на аналізі історичних даних про попередні завдання.

Алгоритм базується на принципі групування місій за схожими параметрами та детального аналізу впливу кожного елемента спорядження на успішність місії. Основні етапи алгоритму включають:

- Збір та попередня обробка даних: Завантаження історичних даних про місії, солдатів та спорядження з бази даних. Виконується попередня обробка даних для приведення їх до необхідного формату для подальшого аналізу.
- Групування місій: Місії групуються за ключовими параметрами, такими як тип місії, сезон, локація, погодні умови, тривалість та вік солдата. Тривалість та вік солдата групуються за певними інтервалами для спрощення аналізу.
- Аналіз спорядження: Для кожної групи місій використовується нейронна мережа, яка визначає вплив кожного елемента спорядження на показник успішності місії. Елементи, що позитивно впливають на успішність, відзначаються для рекомендацій.
- Обробка випадків недостатньої інформації: Якщо для певної комбінації параметрів недостатньо даних, параметри поступово ігноруються у встановленому порядку (спочатку вік солдата, потім погодні умови) для збільшення розміру групи.
- Формування рекомендацій: На основі отриманих результатів формується таблиця рекомендацій щодо спорядження для кожної комбінації параметрів місії.

Реалізація модуля виконана на мові програмування Python з використанням бібліотек pandas, numpy та torch. Паралельна обробка здійснюється за допомогою

модуля `concurrent.futures`, що дозволяє оптимізувати час виконання аналізу при великих обсягах даних.

Першим кроком є завантаження даних з бази даних та їх об'єднання у єдиний `DataFrame` для подальшого аналізу. Історичні дані, що зберігаються в `SQL Server`, завантажуються через бібліотеку `PyODBC`[19].

Модуль отримує дані із бази, сформованої в модулі "База знань". Вхідні дані включають:

- Характеристики місії: тип, сезон, локація, погодні умови, тривалість, рівень успішності.
- Характеристики солдата: вік, зріст, вага.
- Список спорядження, яке використовувалося в місії, та його кількість.

Після завантаження даних здійснюється їх попередня обробка, що включає об'єднання різних таблиць даних для створення єдиного структурованого набору даних. На цьому етапі проводиться очищення даних, зокрема, видалення або заповнення пропущених значень, а також перетворення деяких змінних у відповідний формат для подальшого аналізу.

```
def load_data():
    print("Підключення до бази даних та завантаження даних...")
    conn_str = (
        'Driver={ODBC Driver 17 for SQL Server};'
        'Server=localhost;'
        'Database=InformationSystem;'
        'Trusted_Connection=yes;'
    )

    conn = pyodbc.connect(conn_str)

    missions = pd.read_sql_query('SELECT * FROM Missions', conn)
    soldiers = pd.read_sql_query('SELECT * FROM Soldiers', conn)
    equipment = pd.read_sql_query('SELECT * FROM Equipments', conn)
    mission_equipment = pd.read_sql_query('SELECT * FROM MissionEquipments', conn)

    conn.close()
    print("Дані успішно завантажено.")
    return missions, soldiers, equipment, mission_equipment
```

Рис.3.5. Приклад коду завантаження даних.

Для спрощення подальшого аналізу, параметри тривалості місії та віку солдатів групуються за визначеними інтервалами. Тривалість місії (DurationHours) розподіляється на три категорії: менш або дорівнює 12 годин, від 12 до 24 годин, та понад 24 години.

Аналогічно, вік солдата класифікується на три групи: молодший за 30 років, від 30 до 45 років, та старший за 45 років. Це дозволяє зменшити варіативність даних та сфокусуватися на значимих категоріях, що сприяє більш ефективному аналізу завдяки тому що данні сильно не розмазуються.

```
def map_duration(hours):  
    if hours <= 12:  
        return 6  
    elif 12 < hours <= 24:  
        return 18  
    else:  
        return 30  
  
def map_age(age):  
    if age < 30:  
        return 25  
    elif 30 <= age <= 45:  
        return 35  
    else:  
        return 50
```

Рис.3.6.Приклад коду групування деяких з даних.

Потім програмою проводиться обробка даних для зрозумілого оперування ними машиною. Так програма проводить такі дії:

- Нормалізація числових даних (наприклад, зріст, вага, успішність).
- Перетворення категоріальних даних, таких як тип місії чи погодні умови, за допомогою One-Hot Encoding.
- Формування матриці з ознаками, що передається в нейронну мережу.

```

def prepare_model_data(data, equipment_list):
    print("Підготовка даних для моделювання..")

    # Розгортаємо EquipmentList, щоб мати один елемент спорядження на рядок
    data_expanded = data.explode('EquipmentList')

    # Заповнюємо відсутні значення
    data_expanded['EquipmentName'] = data_expanded['EquipmentList'].fillna('NoEquipment')
    data_expanded.drop(columns=['EquipmentList'], inplace=True)

    # Виключення непотрібних колонок
    columns_to_drop = ['Id_Mission', 'Name_Mission', 'Id_Soldier', 'Name_Soldier', 'MissionId']
    data_expanded = data_expanded.drop(columns=[col for col in columns_to_drop if col in data_expanded.columns])

    # Збереження оригінальних колонок для групування
    grouping_cols = ['Type', 'Season', 'Location', 'Weather', 'Rank', 'DurationHours', 'Age']
    data_expanded[grouping_cols] = data_expanded[grouping_cols]

    # Кодування категоріальних змінних
    categorical_cols = ['Type', 'Season', 'Location', 'Weather', 'Rank', 'EquipmentName']
    data_encoded = pd.get_dummies(data_expanded, columns=categorical_cols)

    # Додавання збережених колонок
    for col in grouping_cols:
        data_encoded[col] = data_expanded[col]

    # Перетворення безперервних змінних
    numeric_cols = ['DurationHours', 'Age']
    data_encoded[numeric_cols] = data_encoded[numeric_cols].fillna(0)

    print("Підготовка даних для моделювання завершена.")
    return data_encoded

```

Активация

Рис.3.7. Кодування категоріальних змінних та підготовка даних для навчання нейронної мережі.

Для визначення впливу кожного елемента спорядження на показник успішності місії (SuccessRate) використовується нейронна мережа. Архітектура мережі складається з трьох шарів: вхідного шару, прихованого шару з 128 нейронами та функцією активації ReLU, другого прихованого шару з 64 нейронами та функцією активації ReLU, та вихідного шару, який видає прогнозований показник успішності місії.

Модель навчається на історичних даних, де вхідними ознаками є закодовані параметри місії та елементи спорядження, а цільовою змінною — показник успішності місії. Під час навчання використовуються алгоритми оптимізації - Adam, та функція втрат - середньоквадратична помилка (MSE), для мінімізації різниці між прогнозованими та фактичними значеннями SuccessRate.

Архітектура нейронної мережі:

- Вхідний шар: включає параметри місії, солдата та спорядження.

- Приховані шари: оптимізують аналіз взаємозв'язків між вхідними параметрами.
- Вихідний шар: генерує оцінку ефективності спорядження.

```
class MissionModel(nn.Module):
    def __init__(self, input_size):
        super(MissionModel, self).__init__()
        self.fc1 = nn.Linear(input_size, 128)
        self.relu1 = nn.ReLU()
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, 1)

    def forward(self, x):
        out = self.fc1(x)
        out = self.relu1(out)
        out = self.fc2(out)
        out = self.relu2(out)
        out = self.fc3(out)
        return out
```

Рис.3.8. Приклад коду нейронної мережі моделі.

Після навчання модель прогнозує вплив кожного елемента спорядження на успішність місії. Ці дані використовуються для формування таблиці відповідностей у форматі CSV.

З метою оптимізації часу обробки великих обсягів даних, було застосовано паралельну обробку груп місій. Для цього використовується модуль `concurrent.futures` з бібліотеки Python, який дозволяє одночасно виконувати декілька задач у різних процесах. Кожна група місій, визначена за відповідними параметрами, обробляється окремим процесом, що значно зменшує загальний час виконання аналізу.

Для кожної групи місій, що містить достатню кількість записів (не менше двох місій), виконується аналіз впливу спорядження на `SuccessRate` за допомогою навченого нейронного мережевого моделі. Модель визначає, які елементи спорядження мають позитивний вплив на успішність місії, і ці елементи додаються до списку рекомендацій. Якщо для певної групи місій недостатньо

даних, параметри групування ігноруються послідовно згідно з визначеним порядком (спочатку вік солдата, потім погодні умови) для збільшення розміру групи. Якщо навіть після цього даних недостатньо, у полі рекомендованого спорядження ставиться прочерк, що означає відсутність рекомендацій.

Після завершення аналізу всіх груп формуються результати у вигляді таблиці, де кожен рядок відповідає певній комбінації параметрів місії та містить рекомендовані елементи спорядження. Таблиця також зберігається у форматі CSV для подальшого використання та аналізу.

Алгоритм базується на використанні сучасних методів машинного навчання, зокрема нейронних мереж, для аналізу великих обсягів даних та виявлення складних закономірностей між параметрами місії та успішністю її виконання. Групування даних за схожими параметрами дозволяє локалізувати аналіз та зосередитися на релевантних факторах, що впливають на результат місії. Використання нейронних мереж забезпечує високу гнучкість та здатність моделі виявляти нелінійні взаємозв'язки між ознаками.

Впровадження паралельної обробки значно покращує продуктивність системи, дозволяючи ефективно обробляти великі набори даних та скорочувати час аналізу. Це особливо важливо у військовому контексті, де швидкість прийняття рішень може мати критичне значення.

Технології та інструменти що використовуються для розробки цього модуля такі:

- Мова програмування: Python.
- Фреймворк: PyTorch для нейронної мережі.
- Бібліотеки: Pandas, PyODBC, Scikit-learn для обробки даних.

3.3.3 Модуль "Помічник у комплектуванні"

Модуль "Помічник у комплектуванні" є третьою частиною інформаційної системи, що забезпечує користувачу можливість отримати рекомендації щодо оптимального спорядження для виконання конкретної місії. Цей модуль інтегрується з таблицею відповідностей, згенерованою другим модулем

("Підготовка експерта"), і дозволяє користувачу вказати характеристики місії та солдата.

Основна мета модуля полягає у забезпеченні користувача інтуїтивно зрозумілим інтерфейсом для введення параметрів місії та солдата й отримання персоналізованого списку спорядження, яке максимально підвищить шанси на успішність виконання завдання.

Основний функціонал

1. Введення параметрів місії Користувач заповнює такі дані про місію:

- Тип місії (наприклад, оборона, розвідка).
- Сезон (зима, літо тощо).
- Локація (ліс, місто, поле).
- Погодні умови (сонячно, дощ тощо).
- Тривалість місії.

UI-елементи: Для кожного параметра використано комбобокси або текстові поля, які запобігають некоректному вводу.

2. Введення характеристик солдата Користувач вказує:

- Вік.
- Зріст.
- Вагу.

UI-елементи: Текстові поля з перевіркою введених даних (вік має бути числом, зріст і вага — в діапазонах, що відповідають реальності).

3. Формування рекомендацій Модуль звертається до таблиці відповідностей, враховує вказані параметри і виводить список спорядження, що включає:

- Назву спорядження.
- Кількість одиниць (наприклад, кількість магазинів до зброї).
- Відсотковий вплив спорядження на успішність місії.

UI-елементи: Вивід рекомендацій реалізований через таблицю, що дозволяє користувачу швидко переглянути список спорядження та його параметри.

Логіка модуля написана на C# із використанням LINQ для обробки даних із таблиці відповідностей. Дані завантажуються у вигляді CSV-файлу, обробляються, а потім відображаються в інтерфейсі користувача.

Інтерфейс побудований за допомогою WPF, що забезпечує:

- Інтуїтивно зрозумілий дизайн.
- Гнучкість у налаштуванні елементів UI.
- Можливість масштабування під різні роздільні здатності екранів.

Висновки

У розглянутому розділі дипломної роботи було представлено результати практичної реалізації інформаційної системи підтримки прийняття рішень щодо спорядження військовослужбовців. Було детально розглянуто розробку архітектури системи, взаємодію між її модулями та особливості програмної реалізації кожного компонента.

Розробка системи базувалася на модульному підході, що дозволило забезпечити незалежну роботу кожного модуля та їх ефективну взаємодію. Модуль "База знань" реалізовано для збору й збереження історичних даних про місії, спорядження та характеристики солдатів. Ці дані є основою для роботи модуля "Підготовка експерта", який, використовуючи сучасні методи машинного навчання, формує таблицю відповідностей, що демонструє вплив спорядження на успішність місій. Таблиця відповідностей слугує ключовим ресурсом для третього модуля — "Помічника в комплектуванні", який генерує рекомендації щодо спорядження для конкретних місій.

Особлива увага в розділі була приділена технічним аспектам реалізації. Описано використання WPF для створення інтуїтивного інтерфейсу в модулях "База знань" і "Помічник в комплектуванні", що забезпечує зручність роботи користувача. Для модуля "Підготовка експерта" було використано мову програмування Python і бібліотеку PyTorch, які дозволили ефективно реалізувати нейронну мережу та забезпечити якісний аналіз історичних даних.

Важливим аспектом розділу стало обґрунтування використання багатосарової перцептронної нейронної мережі та методу оптимізації Adam. Це дало змогу автоматизувати процес створення таблиці відповідностей, враховуючи всі характеристики місій, солдатів і спорядження.

Отримані результати підтверджують, що розроблена система здатна вирішувати поставлені завдання, забезпечуючи автоматизацію аналізу даних і формування рекомендацій. Такий підхід не лише оптимізує процес прийняття рішень, але й підвищує точність і швидкість виконання поставлених завдань.

Таким чином, цей розділ демонструє, як розроблені рішення на практиці забезпечують ефективну роботу інформаційної системи, підтверджуючи її відповідність заявленим вимогам та її актуальність для вирішення завдань оптимізації спорядження військовослужбовців.

РОЗДІЛ 4

РОЗРОБКА СТАРТАП-ПРОЄКТУ

4.1 Загальна характеристика стартап-проєкту

Розробка інформаційної системи підтримки прийняття рішень щодо спорядження військовослужбовця є унікальним проєктом, спрямованим на вирішення низки актуальних проблем у військовій сфері. У сучасному світі, де технології впливають на всі аспекти життя, включно з обороною та безпекою, інтеграція інноваційних підходів у планування військових операцій та забезпечення солдатів стає стратегічно важливою.

Основною метою розробки є створення програмного забезпечення, яке здатне надавати точні рекомендації щодо вибору спорядження для військовослужбовців. Це дозволить підвищити ефективність виконання місій, зменшити ризик для особового складу та оптимізувати використання ресурсів.

Завданнями проєкту є:

- Розробка системи, яка враховує унікальні характеристики місії та солдата.
- Інтеграція методів машинного навчання для аналізу історичних даних.
- Забезпечення можливості адаптації системи до змін у бойових умовах.
- Надання рекомендацій у зручному та зрозумілому форматі.

Система має низку переваг, що роблять її унікальною на ринку:

- Використання історичних даних для генерації рекомендацій.
- Інтеграція модулів на основі сучасних фреймворків (C# та PyTorch).
- Модульна структура, яка забезпечує легкість модернізації та масштабування.

Місце для малюнка: Схема загальної архітектури системи, що демонструє взаємодію трьох основних модулів: "База знань", "Підготовка експерта", "Помічник у комплектуванні".

Проєкт орієнтований на такі категорії користувачів:

- Державні військові структури: Збройні сили, підрозділи територіальної оборони, інші військові формування.
- Приватні компанії: Підрядники, які займаються логістикою та постачанням військового спорядження чи охоронні компанії.
- Військові навчальні заклади: Академії та центри підготовки, що використовують симуляції та навчальні моделі.

Таблиця 4.1.1

Характеристика цільової аудиторії

Категорія користувачів	Основні потреби	Очікувані вигоди
Державні військові структури	Оптимізація логістики та зменшення втрат	Підвищення ефективності виконання місій
Приватні компанії	Аналіз попиту на спорядження	Скорочення витрат та покращення постачання
Військові навчальні заклади	Інструменти для моделювання бойових сценаріїв	Підвищення якості підготовки особового складу

Розробка проекту поділена на кілька етапів:

1. Дослідження та аналіз: Вивчення потреб кінцевих користувачів, збір даних.
2. Розробка модулів: Поетапне створення "Бази знань", "Підготовки експерта" та "Помічника у комплектуванні".
3. Навчання моделі: Застосування багатошарової перцептронної мережі для аналізу історичних даних.
4. Тестування: Перевірка точності рекомендацій на контрольних місіях.
5. Впровадження: Інтеграція системи у робочі процеси кінцевих користувачів.

У результаті впровадження системи планується досягти:

- Підвищення рівня успішності місій: Рекомендації щодо спорядження зменшують ризик помилок у плануванні.
- Економія ресурсів: Зниження перевитрат на зайве спорядження.
- Адаптивність: Система може постійно оновлюватися відповідно до змін у тактичних умовах.

Таблиця 4.1.2

Прогнозовані показники ефективності.

Показник	Значення
Підвищення успішності місій	До 25%
Скорочення витрат на спорядження	До 15%
Час на підготовку рекомендацій	Менше 2 хвилин на одну місію

Проект має значний потенціал для розширення. У перспективі планується розширення аналізованих параметрів та збільшення глибини аналізу, інтеграція системи з платформами управління військовими операціями, а також адаптація для інших галузей, таких як рятувальні служби або гуманітарні організації.

4.2 Технологічний аудит ідеї проєкту

Технологічний аудит є важливою складовою розробки стартап-проєкту, оскільки дозволяє оцінити доцільність використання запропонованих технологій, їх ефективність, ризики впровадження та відповідність сучасним потребам. У цьому пункті буде детально розглянуто технічні аспекти створення інформаційної системи підтримки прийняття рішень щодо спорядження військовослужбовця.

Для реалізації системи було обрано сучасні технологічні рішення, які забезпечують високу продуктивність, надійність та гнучкість.

1. Мова програмування C# Використовується для розробки першого і третього модулів системи, зокрема "Бази знань" і "Помічника у

комплектуванні". Ця мова була обрана завдяки її потужним можливостям для роботи з базами даних, інтеграції з інтерфейсами користувача (WPF) і високій продуктивності.

2. Мова програмування Python та фреймворк PyTorchPython використовується у другому модулі — "Підготовка експерта". Фреймворк PyTorch дозволяє створювати нейронні мережі для аналізу історичних даних, що є ключовим елементом системи. PyTorch забезпечує простоту роботи з великими даними та ефективно навчання моделей.
3. База даних SQL Server Використовується для зберігання історичних даних про місії, солдатів та спорядження. Ця база даних обрана завдяки її стабільності, можливостям масштабування та інтеграції з іншими компонентами системи.

Інтерфейс WPF (Windows Presentation Foundation)

Забезпечує зручний графічний інтерфейс для роботи користувачів із системою. Його можливості дозволяють створити інтуїтивно зрозумілий та естетичний інтерфейс.

Таблиця 4.2.1

Порівняльний аналіз обраних технологій.

Технологія	Призначення	Основні переваги	Можливі недоліки
C#	Розробка модулів 1 і 3	Висока продуктивність, інтеграція з WPF	Залежність від платформи Windows
Python + PyTorch	Розробка модуля 2	Простота навчання моделей, гнучкість	Відносно повільна обробка великих даних
SQL Server	Зберігання історичних даних	Масштабованість, стабільність	Високі вимоги до конфігурації сервера
WPF	Інтерфейс користувача	Інтуїтивність, сучасний дизайн	Обмеження у кросплатформенності

Технології, обрані для проєкту, забезпечують повне покриття всіх необхідних функцій а саме:

- Швидкість обробки даних: Завдяки SQL Server та ефективним алгоритмам у Python, система обробляє великі обсяги даних без затримок.
- Адаптивність: Нейронні мережі PyTorch дозволяють оновлювати таблицю відповідностей відповідно до змін у даних.
- Зручність використання: Інтерфейс WPF робить взаємодію користувачів із системою простою та зрозумілою.

Незважаючи на значні переваги, існують певні ризики, які можуть вплинути на успішність реалізації проєкту:

1. Ризик недостатньої кількості даних. Якщо історичних даних для навчання моделі виявиться замало, результати можуть бути менш точними. Для вирішення цього ризику можна використовувати синтетичні дані або дані з симуляцій.

2. Ризик несумісності компонентів. Інтеграція модулів на різних технологіях (C# і Python) потребує ретельної перевірки. Використання чітко визначених форматів даних (наприклад, CSV) мінімізує цей ризик.
3. Ризик перевантаження бази даних. У разі значного збільшення обсягу даних база може працювати повільніше. Для вирішення цього ризику планується впровадження оптимізованих запитів та архівування старих даних.

На основі проведеного аудиту можна зробити висновок, що обрані технології та методи розробки забезпечують:

- Стабільну роботу системи.
- Високу точність рекомендацій завдяки використанню нейронних мереж.
- Простоту масштабування та оновлення кожного модуля.

На основі проведеного аналізу, у наступних розділах буде описано ринкову стратегію, маркетингову програму та фінансові аспекти стартапу. Це дозволить чітко визначити перспективи комерціалізації розробленого проєкту.

4.3. Аналіз ринкових можливостей запуску стартап-проєкту

Аналіз ринкових можливостей є ключовим етапом для оцінки потенціалу стартапу. У цьому пункті буде розглянуто ринкове середовище, можливі сегменти споживачів, конкурентне середовище та обсяги попиту.

Головним об'єктом розробленої системи є військові підрозділи, зокрема:

- Міністерства оборони різних країн.
- Приватні компанії, які надають послуги з військової підготовки та навчання.
- Центри симуляційних тренувань для військових.

Крім того, існує можливість адаптації системи для: служб безпеки, поліцейських підрозділів, рятувальних служб.

Цільова аудиторія поділяється на кілька основних груп:

1. **Військові структури:** Ці структури зацікавлені у вдосконаленні екіпірування своїх підрозділів для підвищення ефективності виконання завдань. Система дозволяє аналізувати місії та вдосконалювати спорядження.
2. **Навчальні центри:** Центри симуляції бойових дій можуть використовувати систему для формування сценаріїв місій із оптимальним екіпіруванням.
3. **Приватні компанії:** Компанії, які спеціалізуються на безпеці, можуть використовувати систему для планування спорядження своїх співробітників.

Таблиця 4.3.1

Сегментація цільової аудиторії.

Сегмент	Характеристика споживачів	Потенційна користь від системи
Військові структури	Міністерства оборони, армійські підрозділи	Оптимізація спорядження, підвищення ефективності місій
Навчальні центри	Центри бойових симуляцій	Планування сценаріїв із урахуванням спорядження
Приватні компанії	Безпекові організації	Забезпечення співробітників оптимальним спорядженням

Згідно з аналітичними дослідженнями, світовий ринок військових технологій щорічно зростає на 4-6%. Орієнтовний обсяг ринку військового спорядження у 2024 році сягає понад 200 мільярдів доларів США. Система підтримки прийняття рішень щодо спорядження може зайняти свою нішу на цьому ринку.

На даний момент на ринку існує кілька рішень для військових, що використовують елементи штучного інтелекту. Однак, прямих конкурентів із аналогічним функціоналом не виявлено. Основні переваги нашої системи:

- Комплексний підхід до аналізу даних місій.

- Використання історичних даних для генерації рекомендацій.
- Можливість адаптації системи до потреб конкретного замовника.

Для кращого розуміння потенціалу проекту наведемо SWOT-аналіз.

Таблиця 4.3.2

SWOT-аналіз стартапу.

Сильні сторони (Strengths)	Слабкі сторони (Weaknesses)
Висока точність рекомендацій	Потреба у великій кількості даних
Інтуїтивно зрозумілий інтерфейс	Залежність від якісного налаштування
Гнучкість та адаптивність	Потреба у висококваліфікованих спеціалістах
Можливості (Opportunities)	Загрози (Threats)
Розширення на інші сектори	Висока конкуренція у військовій сфері
Потенціал інтеграції з іншими системами	Ризики кібератак
Використання для симуляцій	Залежність від бюджетів замовників

Аналіз ринкових можливостей а також політичної ситуації в світі підтвердив, що стартап має значний потенціал на ринку військових технологій. Відсутність прямих конкурентів із подібним функціоналом створює сприятливі умови для виходу на ринок. У наступному пункті буде розглянуто ринкову стратегію та етапи просування продукту.

4.4. Розробка ринкової стратегії проекту

Для успішного запуску та просування стартапу необхідно розробити детальну ринкову стратегію. Ця стратегія охоплює позиціонування продукту, визначення цільової аудиторії, канали збуту та маркетингові інструменти.

Система підтримки прийняття рішень щодо спорядження військовослужбовців позиціонується як інноваційний інструмент для підвищення ефективності місій. Основними перевагами є:

- Автоматизована генерація рекомендацій спорядження.
- Використання історичних даних для прогнозування результатів.
- Простота інтеграції у військові процеси.

Це рішення підходить як для великих державних структур, так і для приватних компаній, що займаються безпекою та симуляціями.

Впровадження системи передбачає кілька основних етапів:

1. Демонстрація функціоналу: Проведення презентацій для потенційних замовників, таких як Міністерства оборони та навчальні центри.
2. Пілотне впровадження: Інтеграція продукту в роботу обраного підрозділу для тестування його можливостей.
3. Аналіз зворотного зв'язку: Оцінка ефективності та внесення необхідних коректив.
4. Масштабування: Повна інтеграція у військові структури та розширення на інші ринки.

На основі аналізу ринкових можливостей визначено основні сегменти цільової аудиторії:

1. Державні організації: Міністерства оборони, національні гвардії.
2. Приватні компанії: Охоронні структури та безпекові організації.
3. Навчальні центри: Організації, що спеціалізуються на симуляції бойових дій.

Для реалізації продукту будуть використовуватися наступні канали збуту:

- Прямий продаж: Контакти з потенційними замовниками через тендери та договори.
- Партнерські угоди: Співпраця з компаніями, що займаються інтеграцією військових технологій.
- Цифровий маркетинг: Просування через тематичні сайти, блоги та соціальні мережі.

Таблиця 4.4.1

Основні канали збуту продукту.

Канал	Опис	Приклад
Прямий продаж	Презентації та комерційні переговори	Міністерства оборони
Партнерські угоди	Співпраця з інтеграторами	Навчальні центри, тренувальні бази
Цифровий маркетинг	Онлайн-просування через сайти та соцмережі	Реклама у професійних спільнотах

Для забезпечення впізнаваності продукту будуть застосовані наступні інструменти:

- **Рекламні кампанії:** Запуск відеороликів, презентацій та оголошень у спеціалізованих ЗМІ.
- **Участь у виставках:** Представлення продукту на військових і технологічних виставках.
- **Публікації:** Розробка тематичних статей та аналітичних звітів про ефективність системи.

Під час реалізації стратегії можливі наступні ризики:

- Низька довіра до інноваційних продуктів у державному секторі.
- Конкуренція з міжнародними розробниками.
- Ризики затримок через бюрократичні процедури.

Для мінімізації ризиків передбачено проведення широкої роз'яснювальної роботи та адаптації продукту до потреб замовника.

Ринкова стратегія розроблена з урахуванням особливостей цільової аудиторії та сучасних тенденцій ринку. Завдяки комплексному підходу до впровадження продукту, стартап має високі шанси на успішну реалізацію. У наступному пункті буде описано маркетингову програму, спрямовану на підвищення впізнаваності продукту.

4.5 Розробка стратегії виходу проекту на ринок

Ринкова стратегія є ключовим аспектом успішного впровадження стартапу, оскільки вона визначає, як продукт буде просуватись, позиціонуватись та адаптуватись до потреб споживачів. У цьому пункті викладено основні елементи ринкової стратегії для стартапу "Інформаційна система підтримки прийняття рішень щодо спорядження військовослужбовця".

4.4.1 Визначення цільової аудиторії

Цільова аудиторія продукту охоплює декілька сегментів, які мають безпосередню зацікавленість у впровадженні системи:

- Військові структури: підрозділи ЗСУ та інші силові структури.
- Приватні охоронні компанії: організації, які потребують оптимізації процесів спорядження персоналу.
- Міжнародні організації: НАТО, миротворчі місії, які активно впроваджують інноваційні технології.
- Навчальні центри: військові академії та тренувальні бази.

4.4.2 Позиціонування продукту

Наш продукт позиціонується як сучасне, інноваційне рішення для автоматизації та оптимізації процесів спорядження військовослужбовців.

Основні переваги:

- Інтеграція з існуючими системами: можливість взаємодії з іншими інформаційними платформами.
- Гнучкість: адаптація до різних умов експлуатації.
- Точність: використання нейронних мереж для прийняття рішень.

Порівняння з конкурентами підтверджує, що продукт має унікальні риси, які забезпечують йому конкурентну перевагу.

4.4.3 Канали збуту та комунікації

Для ефективного охоплення цільової аудиторії пропонуються наступні канали збуту:

1. Прямі продажі - переговори з військовими відомствами та урядовими структурами, демонстрація продукту на спеціалізованих виставках.
2. Партнерські продажі - співпраця з компаніями, які займаються впровадженням ІТ-рішень для військових.
3. Онлайн-продажі: Інформування через офіційний сайт стартапу та використання соціальних мереж та платформ для демонстрації функціональності продукту.

4.4.4 Цінова політика

Цінова стратегія продукту базується на його унікальності та адаптації до потреб користувачів:

- Базовий рівень: для невеликих організацій, які потребують базового функціоналу.
- Розширений рівень: для великих організацій, які використовують повний спектр можливостей продукту.
- Індивідуальні рішення: для клієнтів із специфічними вимогами.

Таблиця 4.4.2.

Рекомендована цінова політика

Пакет	Ціна(у.о.)	Основні можливості
Базовий	10 000	Основний функціонал без інтеграції.
Розширений	25 000	Додаткові модулі, інтеграція з іншими системами.
Індивідуальний	Від 50 000	Налаштування продукту під специфічні потреби.

4.4.5 Просування продукту

Для успішного просування продукту та підвищення його впізнаваності передбачено розробку комплексної стратегії, яка охоплює декілька ключових напрямків. Основна мета цієї стратегії — забезпечити широку обізнаність серед цільової аудиторії, підвищити інтерес до продукту та стимулювати його впровадження в практичну діяльність військових підрозділів.

Першим етапом просування стане організація масштабної рекламної кампанії. Одним із центральних елементів цієї кампанії буде створення професійного демонстраційного відео, яке висвітлюватиме ключові функціональні можливості системи. Відеоматеріал буде спрямований на показ того, як саме система допомагає підвищити ефективність бойових місій за рахунок оптимального вибору спорядження для солдатів. Це відео буде розміщено на спеціалізованих онлайн-платформах, орієнтованих на військову аудиторію, таких як тематичні форуми, сайти військових закупівель та платформи, присвячені новітнім військовим технологіям.

Окрім онлайн-реклами, особливу увагу буде приділено участі у спеціалізованих виставках і конференціях. Демонстрація продукту на таких заходах дозволить безпосередньо презентувати його потенційним клієнтам, включаючи представників військових відомств, командирів підрозділів, розробників спорядження та інших зацікавлених сторін. Участь у виставках забезпечить можливість отримати зворотний зв'язок від професіоналів, що дозволить краще адаптувати продукт до їхніх реальних потреб.

Окремим елементом стратегії стане публікація інформації про продукт у спеціалізованих медіа. Статті, що детально висвітлюють особливості системи, її переваги та практичне застосування, будуть розміщені у профільних виданнях, орієнтованих на військову аудиторію. Такі матеріали можуть містити не лише технічний опис, але й історії успішного використання системи в реальних умовах, що зробить її більш привабливою для потенційних клієнтів.

Крім того, планується випуск прес-релізів, які інформуватимуть про запуск продукту на ринок. У прес-релізах будуть детально описані унікальні переваги системи, основні проблеми, які вона вирішує, та її перспективи для військової сфери. Ці повідомлення будуть поширені через офіційні канали компанії-розробника, а також через профільні новинні ресурси.

Усі зазначені заходи спрямовані на те, щоб забезпечити максимальну обізнаність цільової аудиторії про продукт, підкреслити його унікальні можливості та переконати потенційних користувачів у доцільності його

впровадження. Комплексний підхід до просування дозволить ефективно позиціонувати продукт на ринку військових технологій, створити позитивний імідж системи та сприяти її активному використанню у військових структурах.

4.4.6 Технологічна підтримка

Технологічна підтримка є одним із ключових аспектів успішного впровадження та довготривалої співпраці із замовниками. Вона спрямована на забезпечення стабільного функціонування системи, адаптацію продукту до нових умов та підтримку користувачів на всіх етапах його використання. Комплексний підхід до технологічної підтримки включає кілька основних компонентів, які гарантують ефективність і надійність роботи системи.

Одним із головних елементів технологічної підтримки є регулярне оновлення програмного забезпечення. Це передбачає не лише виправлення можливих помилок у системі, але й розширення її функціональних можливостей відповідно до зворотного зв'язку від користувачів та нових вимог. Такі оновлення дозволять системі залишатися актуальною у швидко змінюваних умовах військової сфери. Зокрема, це може включати оптимізацію алгоритмів, адаптацію до нових типів спорядження чи сценаріїв місій, а також поліпшення інтерфейсу для користувачів.

Для оперативного вирішення будь-яких питань, що можуть виникнути у користувачів, передбачено підтримку через онлайн-чат і телефонну лінію. Онлайн-чат дозволить клієнтам отримувати консультації в режимі реального часу, не припиняючи роботи із системою. Телефонна лінія стане додатковим каналом для спілкування, особливо в тих випадках, коли потрібні більш детальні пояснення або технічна допомога. Обидва канали забезпечать користувачам доступ до кваліфікованих фахівців, які допоможуть швидко вирішити будь-які проблеми.

Окрім цього, важливим компонентом підтримки є проведення тренінгів для персоналу клієнтів. Це дозволить користувачам глибше розуміти функціонал системи, ефективно використовувати її можливості та самостійно вирішувати базові завдання. Тренінги можуть проводитися у формі семінарів, інтерактивних

занять або онлайн-курсів, що забезпечить гнучкість і доступність для різних категорій клієнтів. Крім того, навчальні матеріали можуть бути надані у вигляді посібників, відеоуроків чи інтерактивних інструкцій, що допоможе користувачам засвоїти інформацію у зручному форматі.

Важливим аспектом технологічної підтримки є також її доступність протягом усього життєвого циклу продукту. Це передбачає безперервний моніторинг роботи системи, своєчасне реагування на запити клієнтів та постійну готовність до адаптації продукту до нових умов. Такий підхід гарантує високу якість обслуговування та сприяє побудові довірливих відносин із замовниками.

4.6. Розроблення маркетингової програми стартап-проекту

Розроблення маркетингової програми є одним із ключових етапів у процесі реалізації стартап-проекту. Ця програма покликана забезпечити впізнаваність продукту, створити попит серед цільової аудиторії та сформувати довіру до системи. У сучасних умовах високої конкуренції на ринку військових технологій саме правильно побудована маркетингова стратегія здатна зробити продукт успішним і забезпечити його стабільний розвиток.

Маркетингова програма стартапу базується на кількох важливих елементах. По-перше, це створення чіткої стратегії позиціонування продукту, яка враховує його унікальні переваги. По-друге, це використання багатоканального підходу для охоплення різних сегментів цільової аудиторії. По-третє, це забезпечення постійного зворотного зв'язку з клієнтами, який дозволяє вдосконалювати продукт та підтримувати високий рівень їхньої задоволеності.

Маркетингова програма починається із визначення ключових завдань. Перш за все, необхідно забезпечити формування сильного бренду, який викликатиме довіру у клієнтів. Для цього створюється не лише унікальний логотип та слоган, а й детально розробляється концепція бренду, яка підкреслює його інноваційність, якість та українське походження. Наприклад, продукт може

позиціонуватись як технологічне рішення, здатне підвищити ефективність військових операцій та забезпечити додатковий рівень захисту.

Наступним важливим етапом є вибір каналів просування. Сучасний ринок вимагає багатоканального підходу. З одного боку, необхідно використовувати цифрові канали, такі як створення офіційного сайту продукту, який стане головним джерелом інформації про систему. Сайт має містити не лише опис функціональних можливостей, а й демонстраційні матеріали, включаючи відео та відгуки перших клієнтів. Крім того, важливу роль відіграють соціальні мережі, які дозволяють взаємодіяти з широкою аудиторією та оперативно реагувати на її потреби. З іншого боку, традиційні канали, такі як участь у виставках та конференціях, забезпечують безпосередній контакт із потенційними клієнтами.

Не менш важливою складовою маркетингової програми є проведення рекламних кампаній. Це може включати як таргетовану рекламу в інтернеті, так і публікації у спеціалізованих військових та технологічних виданнях. Під час цих кампаній необхідно акцентувати увагу на перевагах продукту, демонструючи його ефективність на прикладах реальних кейсів.

Ключовою особливістю маркетингової програми є її адаптивність. У сучасних умовах, коли технології швидко змінюються, важливо постійно аналізувати ефективність обраних стратегій і за потреби коригувати їх. Наприклад, якщо один із каналів просування виявиться менш ефективним, слід перерозподілити ресурси на більш успішні напрямки. У цьому процесі особливу роль відіграють метрики ефективності, які дозволяють оцінити результати кожного кроку програми. До таких метрик можуть належати кількість відвідувань сайту, показники залучення клієнтів, рівень задоволеності користувачів тощо.

Останнім, але не менш важливим етапом маркетингової програми є забезпечення постійного зворотного зв'язку. Проводячи опитування серед клієнтів, аналізуючи їхні відгуки та пропозиції, стартап отримує унікальну можливість вдосконалювати свій продукт і робити його максимально відповідним до потреб ринку.

Таким чином, розроблена маркетингова програма є комплексною та спрямованою на досягнення довгострокових результатів. Завдяки використанню сучасних цифрових інструментів, вдосконаленню стратегії бренду та активному залученню клієнтів, стартап-проект має всі шанси успішно конкурувати на ринку та стати одним із провідних гравців у своїй галузі.

Висновки

У цьому розділі було представлено повний цикл розробки стартап-проєкту, заснованого на інформаційній системі підтримки прийняття рішень щодо спорядження військовослужбовця. Здійснено глибокий аналіз ідеї, технологічних можливостей і ринкового середовища, які забезпечують фундамент для реалізації проєкту.

Проведений технологічний аудит підтвердив інноваційність запропонованої системи, а також її високу ефективність і доцільність використання у військовій сфері. Аналіз ринкових можливостей дозволив виділити конкурентні переваги продукту, зокрема його унікальність та здатність відповідати актуальним потребам оборонного сектора. Розроблена ринкова стратегія й маркетингова програма надали чітке бачення шляху впровадження продукту, що включає як використання цифрових платформ, так і співпрацю з потенційними споживачами.

Основною метою цього розділу було забезпечення комплексного підходу до планування запуску продукту, враховуючи всі аспекти, пов'язані з його просуванням та інтеграцією у реальне ринкове середовище. Описані методи та підходи формують практичну основу для успішного старту та подальшого розвитку системи. Висновки, отримані у цьому розділі, забезпечують необхідний зв'язок із загальною метою роботи та підтверджують практичну значущість запропонованого рішення.

ВИСНОВКИ

У роботі розглянуто проблему процесу вибору військового спорядження для виконання різних за умовами задач. На основі проведеного аналізу запропоновано та реалізовано ефективний алгоритм, який використовує комбінацію нейронних мереж і паралельних обчислювальних підходів для аналізу великих обсягів даних.

У роботі проведено аналіз існуючих експертних систем, їх переваг і недоліків, а також визначено сучасні тенденції в розробці подібних систем. Було розглянуто приклади успішного впровадження експертних систем у різних галузях, зокрема в оборонній сфері, що допомогло сформулювати основні вимоги до створюваної системи.

На основі проведених досліджень запропоновано архітектуру інформаційної системи, яка складається з трьох модулів: "База знань", "Підготовка експерта" та "Помічник в комплектуванні". Кожен із модулів реалізує окремий етап роботи системи, забезпечуючи збір даних, аналіз спорядження та формування рекомендацій для конкретних умов місії.

Науковою новизною роботи є спосіб суміщення нейронних мереж із паралельними обчислювальними підходами, що забезпечує швидке й точне формування рекомендацій щодо вибору спорядження залежно від умов місії. Запропонований алгоритм враховує широкий спектр змінних, зокрема характеристик місії, параметрів спорядження та фізіологічних особливостей військовослужбовців. Такий підхід дозволяє забезпечити високу адаптивність системи до специфічних умов бойових задач, що є важливим внеском у розвиток технологій підтримки прийняття рішень у військовій сфері.

Практична цінність роботи полягає в розробці інструменту, який дозволяє підвищити оперативність і точність процесу підготовки військовослужбовця до запланованих задач. Система здатна зменшити витрати часу та ресурсів, необхідних для планування бойових операцій. Крім того, запропонована

архітектура є простою для інтеграції в існуючі військові інформаційні системи, що робить її перспективною для впровадження у практику.

Результати роботи можуть бути впроваджені в інформаційно-аналітичні підрозділи Збройних Сил України. Використання системи в тренувальних центрах дозволить протестувати та адаптувати алгоритми, що стане першим етапом її інтеграції у військові операції. Пілотне впровадження може сприяти оптимізації підготовки військових до виконання бойових задач та покращенню загальної ефективності військових дій.

Подальші дослідження можуть бути спрямовані на розширення бази даних для врахування специфічних умов місій, таких як географічні особливості, погода та індивідуальні параметри військовослужбовців. Інтеграція додаткових модулів, таких як аналіз ризиків і прогнозування результатів місій, дозволить значно розширити функціональні можливості системи. Застосування методів підкріпленого навчання також є перспективним напрямом для забезпечення динамічного аналізу змінюваних умов задачі.

Таким чином, запропонована в роботі система демонструє високий потенціал для впровадження в реальні умови. Результати дослідження створюють основу для подальшого розвитку технологій підтримки прийняття рішень, спрямованих на підвищення ефективності виконання бойових задач у військовій сфері.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Buchanan, B. G., & Shortliffe, E. H. (1984). Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley.
2. Джарратано Дж., Райли Г. Экспертные системы: принципы разработки и программирование. Москва. 2007. 1152с
3. Nilsson, N. J. (2009). The Quest for Artificial Intelligence: A History of Ideas and Achievements. Cambridge University Press.
4. С. В. Мариніна, Н. І. Шабранська, "Аналіз світових наукових та технологічних досліджень за тематикою 'Інформаційні технології у військовій сфері'," УкрІНТЕІ, Київ, 2024, 85 с.
5. О. В. Шаповалова, А. В. Мажара, "Аналіз розробок та практичних реалізацій експертних систем у світовій практиці," Київ: Науково-технічне видавництво, 2021, 120 с.
6. S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed., Pearson, 2016.
7. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
8. Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives" IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.
9. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980, 2014.
10. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning" Nature, vol. 521, no. 7553, pp. 436–444, 2015.
11. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009.
12. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). "PyTorch: An Imperative Style, High-Performance Deep

- Learning Library." *Advances in Neural Information Processing Systems*, 32, 8024-8035.
13. Stevens, E. (2020). *Deep Learning with PyTorch: A 60 Minute Blitz*. Packt Publishing.
 14. Ketkar, N. (2017). *Deep Learning with Python and PyTorch*. Apress.
 15. PyTorch Documentation. [Электронный ресурс] – Режим доступа до ресурсу: <https://pytorch.org/docs/>
 16. Howard, J., & Gugger, S. (2020). *Deep Learning for Coders with Fastai and PyTorch: AI Applications Without a PhD*. O'Reilly Media.
 17. Nathan, A. (2008). *Windows Presentation Foundation Unleashed (WPF)*. Sams Publishing.
 18. MacDonald, M. (2020). *Pro WPF in .NET Core 3: Windows Presentation Foundation in .NET Core 3*. Apress.
 19. Esposito, D. (2020). *Python Programming for Beginners: Develop Your Own Database Applications with PyODBC*. Packt Publishing.

ДОДАТОК 1

Частина програмного коду “Бази знань”

```

<Window x:Class="KnowledgeBase.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:KnowledgeBase_1"
    mc:Ignorable="d"
    Title="Підготовка історичних даних" Height="600" Width="800">
<Grid>
    <TabControl>
        <TabItem Header="Солдати">
            <Grid Margin="10">
                <Grid.RowDefinitions>
                    <RowDefinition Height="Auto"/>
                    <RowDefinition Height="*/>
                </Grid.RowDefinitions>

                <!-- Форма додавання солдата -->
                <StackPanel Orientation="Vertical" Grid.Row="0">
                    <TextBlock Text="Позивний:"/>
                    <TextBox Name="SoldierNameTextBox" Width="200"/>

                    <TextBlock Text="Вік:"/>
                    <TextBox Name="SoldierAgeTextBox" Width="200"/>
                </StackPanel>
            </Grid>
        </TabItem>
    </TabControl>
</Grid>

```

```

<TextBlock Text="Звання:"/>
<TextBox Name="SoldierRankTextBox" Width="200"/>

<TextBlock Text="Вага (кг):"/>
<TextBox Name="SoldierWeightTextBox" Width="200"/>

<TextBlock Text="Зріст (см):"/>
<TextBox Name="SoldierHeightTextBox" Width="200"/>

<Button Content="Додати солдата" Click="AddSoldierButton_Click"
Margin="0,10,0,0" Width="200"/>
</StackPanel>
</Grid>
</TabItem>
<TabItem Header="Спорядження">
<Grid Margin="10">
<Grid.RowDefinitions>
<RowDefinition Height="Auto"/>
<RowDefinition Height="*/>
</Grid.RowDefinitions>
<!-- Форма додавання спорядження -->
<GroupBox Header="Додати нове спорядження">
<StackPanel Orientation="Vertical">
<TextBlock Text="Назва спорядження:"/>
<TextBox Name="EquipmentNameTextBox" Width="200"/>

```

```

<TextBlock Text="Кількість:"/>
<TextBox Name="EquipmentNumberTextBox" Width="200"/>

<TextBlock Text="Тип спорядження:"/>
<ComboBox Name="EquipmentTypeComboBox" Width="200"/>

<Button Content="Додати спорядження"
Click="AddEquipmentButton_Click" Margin="0,10,0,0" Width="200"/>
</StackPanel>
</GroupBox>
</Grid>
</TabItem>
<TabItem Header="Місії">
<Grid Margin="10">
<ScrollViewer VerticalScrollBarVisibility="Auto">
<StackPanel Orientation="Vertical">
<!-- Назва місії -->
<TextBlock Text="Назва місії:"/>
<TextBox x:Name="NameTextBox"/>

<!-- Тип місії -->
<TextBlock Text="Тип місії:" Margin="0,10,0,0"/>
<ComboBox x:Name="TypeComboBox"/>

<!-- Сезон -->
<TextBlock Text="Сезон:" Margin="0,10,0,0"/>
<ComboBox x:Name="SeasonComboBox"/>

```

```

<!-- Локація -->
<TextBlock Text="Локація:" Margin="0,10,0,0"/>
<ComboBox x:Name="LocationComboBox"/>

<!-- Погодні умови -->
<TextBlock Text="Погодні умови:" Margin="0,10,0,0"/>
<ComboBox x:Name="WeatherComboBox"/>

<!-- Тривалість -->
<TextBlock Text="Тривалість (години):" Margin="0,10,0,0"/>
<TextBox x:Name="DurationHoursTextBox"/>

<!-- Рівень успішності -->
<TextBlock Text="Рівень успішності (0-100):"
Margin="0,10,0,0"/>
<TextBox x:Name="SuccessRateTextBox"/>

<!-- Вибір солдата -->
<TextBlock Text="Солдат:" Margin="0,10,0,0"/>
<ComboBox x:Name="SoldierComboBox"
SelectedValuePath="Id">
    <ComboBox.ItemTemplate>
        <DataTemplate>
            <StackPanel Orientation="Horizontal">
                <TextBlock Text="{Binding Name}" Margin="5,0" />
                <TextBlock Text=" | " />
                <TextBlock Text="{Binding Age}" Margin="5,0" />
            </StackPanel>
        </DataTemplate>
    </ComboBox.ItemTemplate>
</ComboBox>

```

```

        <TextBlock Text="років | " />
        <TextBlock Text="{Binding Weight}" Margin="5,0" />
        <TextBlock Text="кг | " />
        <TextBlock Text="{Binding Height}" Margin="5,0" />
        <TextBlock Text="см" />
    </StackPanel>
</DataTemplate>
</ComboBox.ItemTemplate>
</ComboBox>

<!-- Вибір спорядження -->
<TextBlock Text="Спорядження:" Margin="0,10,0,0"/>
<ListBox x:Name="EquipmentListBox" SelectionMode="Multiple"
Height="100">
    <ListBox.ItemTemplate>
        <DataTemplate>
            <TextBlock>
                <TextBlock.Text>
                    <MultiBinding StringFormat="{0} ({1})">
                        <Binding Path="Name"/>
                        <Binding Path="Number"/>
                    </MultiBinding>
                </TextBlock.Text>
            </TextBlock>
        </DataTemplate>
    </ListBox.ItemTemplate>
</ListBox>

```

```

        <!-- Кнопка додавання місії -->
        <Button Content="Додати місію"
Click="AddMissionButton_Click" Margin="0,20,0,0"/>
    </StackPanel>
</ScrollViewer>
</Grid>
</TabItem>
</TabControl>
</Grid>
</Window>

```

```

public partial class MainWindow : Window
{
    private MissionContext db;
    private List<Equipment> selectedEquipments = new List<Equipment>();

    public MainWindow()
    {
        InitializeComponent();
        db = new MissionContext();

        LoadEnums();
        LoadSoldiers();
        LoadEquipments();
    }
}

```

```
private void AddSoldierButton_Click(object sender, RoutedEventArgs e)
{
    try
    {
        Soldier soldier = new Soldier
        {
            Name = SoldierNameTextBox.Text,
            Age = int.Parse(SoldierAgeTextBox.Text),
            Rank = SoldierRankTextBox.Text,
            Weight = SoldierWeightTextBox.Text,
            Height = SoldierHeightTextBox.Text
        };

        db.Soldier.Add(soldier);
        db.SaveChanges();

        MessageBox.Show("Солдата додано успішно!");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Помилка при додаванні солдата: " + ex.Message);
    }
}

private void AddEquipmentButton_Click(object sender, RoutedEventArgs e)
{
```

```

try
{
    Equipment equipment = new Equipment
    {
        Name = EquipmentNameTextBox.Text,
        Number = int.Parse(EquipmentNumberTextBox.Text),
        EquipmentType
(EnumEquipmentType)EquipmentTypeComboBox.SelectedValue
    };

    db.Equipment.Add(equipment);
    db.SaveChanges();

    MessageBox.Show("Спорядження додано успішно!");
}
catch (Exception ex)
{
    MessageBox.Show("Помилка при додаванні спорядження: " + ex.Message);
}
}

private void LoadEnums()
{
    // Завантаження перерахувань у ComboBox
    TypeComboBox.ItemsSource
Enum.GetValues(typeof(EnumMissionType)).Cast<EnumMissionType>();
    SeasonComboBox.ItemsSource
Enum.GetValues(typeof(EnumSeason)).Cast<EnumSeason>();
}

```

```
LocationComboBox.ItemsSource =  
Enum.GetValues(typeof(EnumLocationType)).Cast<EnumLocationType>();  
  
WeatherComboBox.ItemsSource =  
Enum.GetValues(typeof(EnumWeatherConditions)).Cast<EnumWeatherConditions>( );  
  
EquipmentTypeComboBox.ItemsSource =  
Enum.GetValues(typeof(EnumEquipmentType)).Cast<EnumEquipmentType>();  
}
```

```
private void LoadSoldiers()  
{  
    var soldiers = db.Soldier.ToList();  
    SoldierComboBox.ItemsSource = soldiers;  
}
```

```
private void LoadEquipments()  
{  
    var equipments = db.Equipment.ToList();  
    EquipmentListBox.ItemsSource = equipments;  
}
```

```
private void AddMissionButton_Click(object sender, RoutedEventArgs e)  
{  
    try  
    {  
        // Створення нової місії  
        Mission mission = new Mission  
    {
```

```
Name = NameTextBox.Text,
Type = (EnumMissionType)TypeComboBox.SelectedItem,
Season = (EnumSeason)SeasonComboBox.SelectedItem,
Location = (EnumLocationType)LocationComboBox.SelectedItem,
Weather = (EnumWeatherConditions)WeatherComboBox.SelectedItem,
DurationHours = int.Parse(DurationHoursTextBox.Text),
SuccessRate = int.Parse(SuccessRateTextBox.Text),
SoldierId = ((Soldier)SoldierComboBox.SelectedItem).Id,
Equipments = new List<Equipment>()
};

// Додавання обраного спорядження
foreach (Equipment equipment in EquipmentListBox.SelectedItems)
{
    mission.Equipments.Add(equipment);
}

db.Mission.Add(mission);
db.SaveChanges();

MessageBox.Show("Місію успішно додано!");

// Очищення полів
ClearFields();
}
catch (Exception ex)
{
```

```
        MessageBox.Show("Помилка при додаванні місії: " + ex.Message);
    }
}

private void ClearFields()
{
    NameTextBox.Clear();
    TypeComboBox.SelectedIndex = -1;
    SeasonComboBox.SelectedIndex = -1;
    LocationComboBox.SelectedIndex = -1;
    WeatherComboBox.SelectedIndex = -1;
    DurationHoursTextBox.Clear();
    SuccessRateTextBox.Clear();
    SoldierComboBox.SelectedIndex = -1;
    EquipmentListBox.SelectedItems.Clear();
}

}

public class MissionContext : DbContext
{
    public MissionContext() : base("name=MissionDB")
    {
    }

    public DbSet<Soldier> Soldier { get; set; }
    public DbSet<Mission> Mission { get; set; }
    public DbSet<Equipment> Equipment { get; set; }
}
```

```

protected override void OnModelCreating(DbModelBuilder modelBuilder)
{
    modelBuilder.Entity<Mission>()
        .HasMany(m => m.Equipments)
        .WithMany(e => e.Missions)
        .Map(me =>
        {
            me.ToTable("MissionEquipments");
            me.MapLeftKey("MissionId");
            me.MapRightKey("EquipmentId");
        });
}

```

Частина програмного коду програми “Підготовка експерта”

```

import pyodbc
import pandas as pd
import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim
import os
import warnings
warnings.filterwarnings('ignore')

from concurrent.futures import ProcessPoolExecutor, as_completed

```

```
# -----  
# 1. Підключення до бази даних та завантаження даних  
# -----  
  
def load_data():  
    print("Підключення до бази даних та завантаження даних...")  
    conn_str = (  
        'Driver={ODBC Driver 17 for SQL Server};'  
        'Server=localhost;'  
        'Database=InformationSystem;'  
        'Trusted_Connection=yes;'  
    )  
  
    conn = pyodbc.connect(conn_str)  
  
    missions = pd.read_sql_query('SELECT * FROM Missions', conn)  
    soldiers = pd.read_sql_query('SELECT * FROM Soldiers', conn)  
    equipment = pd.read_sql_query('SELECT * FROM Equipments', conn)  
    mission_equipment = pd.read_sql_query('SELECT * FROM MissionEquipments',  
    conn)  
  
    conn.close()  
    print("Дані успішно завантажено.")  
    return missions, soldiers, equipment, mission_equipment  
  
# -----  
# 2. Препроцесинг даних
```

```
# -----
```

```
def preprocess_data(missions, soldiers, equipment, mission_equipment):
    print("Початок препроцесингу даних...")

    data = missions.merge(soldiers, left_on='SoldierId', right_on='Id',
suffixes=('_Mission', '_Soldier'))

    mission Equip = mission_equipment.merge(equipment, left_on='EquipmentId',
right_on='Id', suffixes=('_MissionEquip', '_Equipment'))

    equip_per_mission = mission Equip.groupby('MissionId')['Name'].apply(list).reset_index(name='EquipmentList')

    data = data.merge(equip_per_mission, left_on='Id_Mission', right_on='MissionId',
how='left')

    data['EquipmentList'] = data['EquipmentList'].apply(lambda x: x if isinstance(x, list)
else [])

    data['Weight'] = data['Weight'].astype(float)
    data['Height'] = data['Height'].astype(float)
    data['Age'] = data['Age'].astype(float)
    data['DurationHours'] = data['DurationHours'].astype(float)
    data['SuccessRate'] = data['SuccessRate'].astype(float)

    print("Препроцесинг даних завершено.")
    return data
```

```

# -----
# 3. Підготовка даних для моделювання
# -----

def prepare_model_data(data, equipment_list):
    print("Підготовка даних для моделювання...")

    data_expanded = data.explode('EquipmentList')

    data_expanded['EquipmentName'] =
data_expanded['EquipmentList'].fillna('NoEquipment')
    data_expanded.drop(columns=['EquipmentList'], inplace=True)

    columns_to_drop = ['Id_Mission', 'Name_Mission', 'Id_Soldier', 'Name_Soldier',
'MissionId']

    data_expanded = data_expanded.drop(columns=[col for col in columns_to_drop if
col in data_expanded.columns])

    grouping_cols = ['Type', 'Season', 'Location', 'Weather', 'Rank', 'DurationHours',
'Age']

    data_expanded[grouping_cols] = data_expanded[grouping_cols]

    categorical_cols = ['Type', 'Season', 'Location', 'Weather', 'Rank', 'EquipmentName']
    data_encoded = pd.get_dummies(data_expanded, columns=categorical_cols)

    for col in grouping_cols:
        data_encoded[col] = data_expanded[col]

```

```
numeric_cols = ['DurationHours', 'Age']
data_encoded[numeric_cols] = data_encoded[numeric_cols].fillna(0)

print("Підготовка даних для моделювання завершена.")
return data_encoded

# -----
# 4. Навчання нейронної мережі
# -----

class MissionModel(nn.Module):
    def __init__(self, input_size):
        super(MissionModel, self).__init__()
        self.fc1 = nn.Linear(input_size, 128)
        self.relu1 = nn.ReLU()
        self.fc2 = nn.Linear(128, 64)
        self.relu2 = nn.ReLU()
        self.fc3 = nn.Linear(64, 1)

    def forward(self, x):
        out = self.fc1(x)
        out = self.relu1(out)
        out = self.fc2(out)
        out = self.relu2(out)
        out = self.fc3(out)
        return out
```

```
# -----  
# Додано функції на верхньому рівні для паралельної обробки  
# -----  
  
def map_duration(hours):  
    if hours <= 12:  
        return 6  
    elif 12 < hours <= 24:  
        return 18  
    else:  
        return 30  
  
def map_age(age):  
    if age < 30:  
        return 25  
    elif 25 <= age <= 45:  
        return 35  
    else:  
        return 50  
  
def analyze_group(args):  
    group_data, parameters, ignored_params, group_parameters, ignore_order = args  
  
    if len(group_data) < 2:  
        if len(ignored_params) == len(ignore_order):  
            result = {
```

```

        'Parameters': {param: group_data[param].iloc[0] if param in
group_data.columns else 'Ignored' for param in group_parameters},
        'RecommendedEquipment': '—'
    }
    return result
else:
    return None
else:
    categorical_cols = ['Rank']
    group_data_encoded = group_data.copy()
    group_data_encoded[categorical_cols] =
group_data_encoded[categorical_cols].astype(str)
    group_data_encoded = pd.get_dummies(group_data_encoded,
columns=categorical_cols)

    feature_cols = group_data_encoded.columns.drop('SuccessRate')

    X = group_data_encoded[feature_cols]
    y = group_data_encoded['SuccessRate']

    X_tensor = torch.tensor(X.values.astype(np.float32))
    y_tensor = torch.tensor(y.values.astype(np.float32)).unsqueeze(1)

    input_size = X.shape[1]
    model = MissionModel(input_size)
    criterion = nn.MSELoss()
    optimizer = optim.Adam(model.parameters(), lr=0.001)

```

```

num_epochs = 50
for epoch in range(num_epochs):
    model.train()
    optimizer.zero_grad()
    outputs = model(X_tensor)
    loss = criterion(outputs, y_tensor)
    loss.backward()
    optimizer.step()

with torch.no_grad():
    model.eval()
    weights = model.fc1.weight.data.numpy()

    equipment_columns = [col for col in X.columns if
col.startswith('EquipmentName_')]
    positive_equipment = []
    for col in equipment_columns:
        idx = X.columns.get_loc(col)
        weight = weights[0][idx]
        if weight > 0:
            equipment_name = col.replace('EquipmentName_', '')
            positive_equipment.append(equipment_name)

    if not positive_equipment:
        positive_equipment = ['—']

result = {

```

```

        'Parameters': {param: group_data[param].iloc[0] if param in
group_data.columns else 'Ignored' for param in group_parameters},

```

```

    'RecommendedEquipment': ', '.join(positive_equipment)

```

```

}

```

```

return result

```

```

# -----

```

```

# 5. Генерація таблиці відповідностей з набором спорядження (Новий алгоритм з
паралельною обробкою)

```

```

# -----

```

```

def generate_correspondence_table(data, equipment_list):

```

```

    print("Генерація таблиці відповідностей згідно з новим алгоритмом...")

```

```

    data['DurationHoursGrouped'] = data['DurationHours'].apply(map_duration)

```

```

    data['AgeGrouped'] = data['Age'].apply(map_age)

```

```

    group_parameters = ['Type', 'Season', 'Location', 'Weather', 'DurationHoursGrouped',
'AgeGrouped', 'Rank']

```

```

    ignore_order = ['AgeGrouped', 'Weather']

```

```

    parameter_sets = []

```

```

    for i in range(len(ignore_order)+1):

```

```

        ignored_params = ignore_order[:i]

```

```

        params = [param for param in group_parameters if param not in ignored_params]

```

```

        parameter_sets.append((params, ignored_params))

```

```
results = []

groups_to_process = []

for params, ignored_params in parameter_sets:
    grouped = data.groupby(params)
    for group_name, group_data in grouped:
        group_info = (group_data, params, ignored_params, group_parameters,
ignore_order)
        groups_to_process.append(group_info)

print("Обробка груп у паралельному режимі...")
with ProcessPoolExecutor() as executor:
    futures = [executor.submit(analyze_group, args) for args in groups_to_process]
    for future in as_completed(futures):
        result = future.result()
        if result:
            results.append(result)

unique_results = {}
for res in results:
    key = tuple(sorted(res['Parameters'].items()))
    if key not in unique_results:
        unique_results[key] = res

results = list(unique_results.values())
```

```

output_records = []

for res in results:
    record = {**res['Parameters'], 'RecommendedEquipment':
res['RecommendedEquipment']}
    output_records.append(record)

correspondence_table = pd.DataFrame(output_records)

correspondence_table.to_csv('equipment_correspondence_table_new.csv',
index=False)

print("Таблиця відповідностей згенерована та збережена.")

return correspondence_table

# -----
# 6. Основна функція
# -----

def main():
    # 1. Завантаження даних
    missions, soldiers, equipment, mission_equipment = load_data()

    # 2. Препроцесинг даних
    data = preprocess_data(missions, soldiers, equipment, mission_equipment)

    # 3. Отримання списку унікального спорядження
    equipment_list = equipment['Name'].unique().tolist()

```

```
# Переконаємося, що equipment_list містить рядки
equipment_list = [str(eq) for eq in equipment_list if pd.notnull(eq)]

# 4. Підготовка даних для моделювання
data_encoded = prepare_model_data(data, equipment_list)

# 5. Генерація таблиці відповідностей з новим алгоритмом
correspondence_table = generate_correspondence_table(data_encoded,
equipment_list)

if __name__ == '__main__':
    main()
```