

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: «Обґрунтування моделі прогнозування тривалості лікування цукрового діабету у дітей із використанням нейронних мереж»

Виконав: студент групи Іт-61

Спеціальності 126 «Інформаційні системи та технології»

(шифр і назва)

Кураш Андрій Васильович

(Прізвище та ініціали)

Керівник: д.т.н., професор Тригуба А.М.

(Прізвище та ініціали)

Рецензент: к.т.н., доцент Кригуль Р.Є.

(Прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

д.т.н., проф. А.М. Тригуба

« ____ » _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Курашу Андрію Васильовичу

1. Тема роботи: «Обґрунтування моделі прогнозування тривалості лікування цукрового діабету у дітей із використанням нейронних мереж»

Керівник роботи Тригуба Анатолій Миколайович, професор
затверджені наказом по університету від 12.09.2024 року № 616/к-с.

2. Строк подання студентом роботи 10.01.2024 р.

3. Вихідні дані до роботи: дані для оцінення тривалості лікування цукрового діабету у дітей; методи використання технологій обчислювального інтелекту.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити) _____

Вступ.

Аналіз стану використання методів обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей.

Особливості прогнозування тривалості лікування цукрового діабету у дітей та вибір методів обчислювального інтелекту.

Результати створення та визначення точності нейромережових моделей для прогнозування тривалості лікування цукрового діабету у дітей.

Охорона праці та безпека у надзвичайних ситуаціях.

Визначення показників ефективності.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових слайдів): аналіз стану використання методів обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей; особливості прогнозування тривалості лікування цукрового діабету у дітей та вибір методів обчислювального інтелекту; результати створення та визначення точності нейромережесевих моделей для прогнозування тривалості лікування цукрового діабету у дітей; економічна ефективність.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	<i>Тригуба А.М., зав. кафедри інформаційних технологій</i>		
4	<i>Городецький І.М., доцент кафедри фізики, інженерної графіки та безпеки виробництва</i>		

7. Дата видачі завдання

12 вересня 2024 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	Примітка
1	<i>Написання першого розділу</i>	<i>12.09-20.09.24</i>	
2	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>21.09-14.10.24</i>	
3.	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>15.10-10.11.24</i>	
4.	<i>Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»</i>	<i>11.11-20.11.24</i>	
5.	<i>Оцінення ефективності запропонованої системи</i>	<i>21.11-30.30.24</i>	
6.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	<i>01-04.12.24</i>	
7.	<i>Завершення роботи в цілому</i>	<i>05-10.12.24</i>	

Студент _____ Кураш А.В.
(підпис)

Керівник роботи _____ Тригуба А.М.
(підпис)

УДК 008.64: 004.89

Обґрунтування моделі прогнозування тривалості лікування цукрового діабету у дітей із використанням нейронних мереж.

Кураш А.В. Кафедра інформаційних технологій – Дубляни, ЛНУП, 2024.

Кваліфікаційна робота: 75 с. текст. част., 23 рис., 4 табл., 12 арк. ілюстраційного матеріалу, 50 джерел.

Охарактеризовано роль прогнозування тривалості лікування цукрового діабету у дітей. Проведено аналіз існуючих підходів та досліджень у галузі застосування обчислювального інтелекту в медицині. Подано особливості прогнозування цукрового діабету за допомогою машинного навчання. Сформульовано завдання кваліфікаційної роботи.

Подано особливості прогнозування тривалості лікування цукрового діабету у дітей та вибір методів обчислювального інтелекту. Здійснено вибір методів обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей. Здійснено опис вибраних нейромережевих моделей.

Подано результати підготовки даних для навчання нейромережевих моделей. Наведено результати навчання нейромережевих моделей для прогнозування тривалості лікування цукрового діабету у дітей. Подано результати навчання моделі з використанням long short-term memory (lstm) мережі, моделі рекурентної нейронної мережі (recurrent neural network - rnn) та моделі автоенкодер (autoencoder). Вибрано раціональну нейромережеву модель для прогнозування тривалості лікування цукрового діабету у дітей.

Розроблено заходи стосовно охорони праці. Проведено визначення показників ефективності від прогнозування тривалості лікування цукрового діабету у дітей із використанням нейромережевої моделі.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ СТАНУ ВИКОРИСТАННЯ МЕТОДІВ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ ДЛЯ ПРОГНОЗУВАННЯ ТРИВАЛОСТІ ЛІКУВАННЯ ЦУКРОВОГО ДІАБЕТУ У ДІТЕЙ	8
1.1. Роль прогнозування тривалості лікування цукрового діабету у дітей	8
1.2. Аналіз існуючих підходів та досліджень у галузі застосування обчислювального інтелекту в медицині	11
1.3. Прогнозування цукрового діабету за допомогою машинного навчання..	16
1.4. Завдання кваліфікаційної роботи	22
РОЗДІЛ 2. ОСОБЛИВОСТІ ПРОГНОЗУВАННЯ ТРИВАЛОСТІ ЛІКУВАННЯ ЦУКРОВОГО ДІАБЕТУ У ДІТЕЙ ТА ВИБІР МЕТОДІВ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ	24
2.1. Особливості прогнозування тривалості лікування цукрового діабету у дітей та вибір методів обчислювального інтелекту	24
2.2. Вибір методів обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей	27
2.3. Опис вибраних нейромережових моделей.....	31
2.3.1. Проста штучна нейронна мережа (simple neural network)	31
2.3.2. Рекурентна нейронна мережа (recurrent neural network – rnn).....	32
2.3.3. Автоенкодер (autoencoder – ae).....	34
2.3. Збір даних та вибір інструментарію для прогнозування тривалості лікування цукрового діабету у дітей та вибір методів обчислювального інтелекту.....	35
РОЗДІЛ 3. РЕЗУЛЬТАТИ СТВОРЕННЯ ТА ВИЗНАЧЕННЯ ТОЧНОСТІ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ ДЛЯ ПРОГНОЗУВАННЯ ТРИВАЛОСТІ ЛІКУВАННЯ ЦУКРОВОГО ДІАБЕТУ У ДІТЕЙ	37
3.1. Результати підготовки даних для навчання нейромережових моделей ...	37

3.2. Результати навчання нейромережових моделей для прогнозування тривалості лікування цукрового діабету у дітей.....	42
3.2.1. Результати навчання моделі з використанням long short-term memory (lstm) мережі	42
3.2.2. Результати навчання моделі рекурентної нейронної мережі (recurrent neural network - rnn).....	44
3.2.3. Результати навчання моделі автоенкодер (autoencoder)	47
3.3. Результати вибору раціональної нейромережової моделі для прогнозування тривалості лікування цукрового діабету у дітей.....	50
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ	52
4.1. Аналіз шкідливих чинників під час розробки моделі	52
4.2. Розробка заходів щодо покращення умов праці програмістів	52
4.3. Розробка логічно-імітаційної моделі травматизму під час ураження електричним струмом	53
4.4. Розробка заходів щодо безпеки у надзвичайних ситуаціях.....	56
РОЗДІЛ 5. ВИЗНАЧЕННЯ ПОКАЗНИКІВ ЕФЕКТИВНОСТІ ВІД ПРОГНОЗУВАННЯ ТРИВАЛОСТІ ЛІКУВАННЯ ЦУКРОВОГО ДІАБЕТУ У ДІТЕЙ ІЗ ВИКОРИСТАННЯМ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ.....	58
ВИСНОВКИ І ПРОПОЗИЦІЇ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66
ДОДАТКИ.....	72
Додаток А.1. Код простої штучної нейронної мережі (simple neural network – SNN)	73
Додаток А.2. Код рекурентної нейронної мережі (recurrent neural network – RNN)	74
Додаток А.3. Код Автоенкодера (Autoencoder – AE)	75

ВСТУП

У сучасному світі де наука і технології розвиваються зі сприянням обчислювального інтелекту, медична галузь не залишається осторонь цього технологічного прогресу [15]. Один із напрямків, які набувають особливого значення – це використання методів обчислювального інтелекту для аналізу та оптимізації лікування важких хвороб. Серед них особливе місце займає цукровий діабет, що стає великим соціальним та медичним викликом, особливо в контексті його поширення серед дітей.

Наша робота не тільки спрямована на вдосконалення медичної практики, але й відкриває нові можливості для розвитку персоналізованої медицини, де підходи до лікування адаптуються до індивідуальних характеристик користувача [17]. Отже, використання обчислювального інтелекту в аспекті прогнозування тривалості лікування цукрового діабету у дітей стає не лише технологічним викликом, але й ключовим кроком у бік підвищення якості медичної допомоги та збереження здоров'я наших майбутніх поколінь.

Об'єктом нашого дослідження є вплив методів обчислювального інтелекту, зокрема машинного навчання та аналізу великих обсягів медичних даних, на прогнозування тривалості лікування цукрового діабету у дітей.

Предметом дослідження є вдосконалення процесів діагностики, прогнозу та лікування цієї хвороби, зокрема через точніші та індивідуалізовані прогнози тривалості лікування.

У результаті виконання нашої роботи забезпечується досягнення високої ефективності лікування, підвищення якості життя хворих дітей та оптимізація ресурсів у галузі педіатричної ендокринології.

РОЗДІЛ 1.

АНАЛІЗ СТАНУ ВИКОРИСТАННЯ МЕТОДІВ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ ДЛЯ ПРОГНОЗУВАННЯ ТРИВАЛОСТІ ЛІКУВАННЯ ЦУКРОВОГО ДІАБЕТУ У ДІТЕЙ

1.1. Роль прогнозування тривалості лікування цукрового діабету у дітей

Цукровий діабет (ЦД) – це хронічне захворювання, яке характеризується порушенням обміну вуглеводів, білків і жирів. У дітей найчастіше зустрічається 1-й тип ЦД, який розвивається в результаті аутоімунного ураження бета-клітин підшлункової залози.

Лікування ЦД у дітей полягає в підтримці рівня глюкози в крові в межах нормальних значень за допомогою інсулінотерапії, дієти та фізичних вправ. Інсулінотерапія є основним методом лікування ЦД 1-го типу і є довічною [12].

Тривалість лікування ЦД у дітей залежить від багатьох факторів, включаючи:

✓ *Вік дитини на момент встановлення діагнозу.* У дітей, які були діагностовані з ЦД у віці до 5 років, ризик досягнення ремісії нижчий, ніж у старших дітей.

✓ *Вид ЦД.* У дітей з 1-м типом ЦД ремісія зустрічається рідше, ніж у дітей з 2-м типом ЦД.

✓ *Стабільність глікемічного контролю.* Діти з хорошим глікемічним контролем мають більший шанс досягти ремісії.

✓ *Наявність ускладнень ЦД.* Діти з ускладненнями ЦД, такими як діабетична ретинопатія або нефропатія, мають менший шанс досягти ремісії.

Прогнозування тривалості лікування ЦД у дітей є важливим для розробки індивідуального плану лікування. Діти з високим ризиком досягнення ремісії можуть отримувати менш агресивне лікування, яке може полегшити їхній повсякденний життя.

В даний час існує кілька методів прогнозування тривалості лікування ЦД у дітей. Одним з найбільш поширених методів є визначення рівня глікозильованого гемоглобіну (HbA1c) на початку лікування. Діти з HbA1c <7% мають більший шанс досягти ремісії, ніж діти з HbA1c >8%.



Рисунок 1.1 – Фактори ризику щодо цукрового діабету у дітей [12]

Інші методи прогнозування тривалості лікування ЦД у дітей включають:

- ✓ *Аналіз генів дитини.* Деякі гени пов'язані з підвищеним ризиком досягнення ремісії.
- ✓ *Вимірювання рівня інсуліну в крові дитини.* Діти з високим рівнем інсуліну в крові мають більший шанс досягти ремісії.

✓ *Вивчення стану підшлункової залози дитини. Діти з хорошою функцією підшлункової залози мають більший шанс досягти ремісії.*



Рисунок 1.2 – Синптоми та лікування цукрового діабету у дітей [12]

Дослідження в галузі прогнозування тривалості лікування ЦД у дітей тривають. Розробка більш точних методів прогнозування дозволить покращити результати лікування і якість життя дітей з ЦД.

1.2. Аналіз існуючих підходів та досліджень у галузі застосування обчислювального інтелекту в медицині

Оскільки генерація медичних даних постійно зростає, потреба в їх аналізі продовжує зростати [42].

Наука про дані створює інструменти та покращує доступ до наборів даних для дослідників. Бачення науки про великі дані полягає в тому, щоб вона підтримувала та приносила користь спільноті серцево-судинних захворювань у цілому. Розглядаємо це як заміну фундаментальних досліджень, що взаємодіє з фундаментальними дослідженнями. Багато інструментів науки про дані створюються для підтримки окремих дослідників, які проводять дослідження на основі гіпотез. До них належать інструменти аналізу даних Omics, а також інструменти аналізу тексту та інструменти шляхів анотації. Наука про дані керується даними, інструментами та користувачами, а не гіпотезами (рис. 1.3).

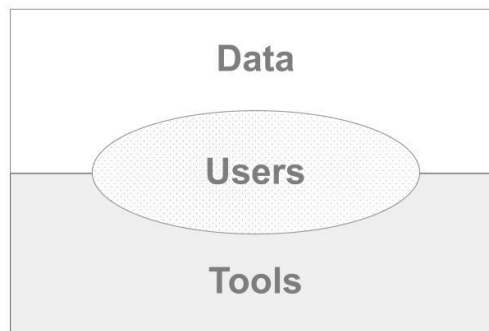


Рисунок 1.3 – Складові науки про дані (дані, інструменти та користувачі)

Наприклад, медичні дані, які походять від діагностики захворювання, можна аналізувати, щоб заощадити значну кількість витрат. McKinsey повідомляє, що 80% американських витрат на охорону здоров'я припадає на лікування хронічних захворювань, а 50% американців страждають від хронічних захворювань [25].

У роботі [21] запропоновано новий мультимодальний алгоритм прогнозування ризику захворювання на основі згорткової нейронної мережі (CNN), використовуючи структуровані та неструктуровані дані з лікарні.

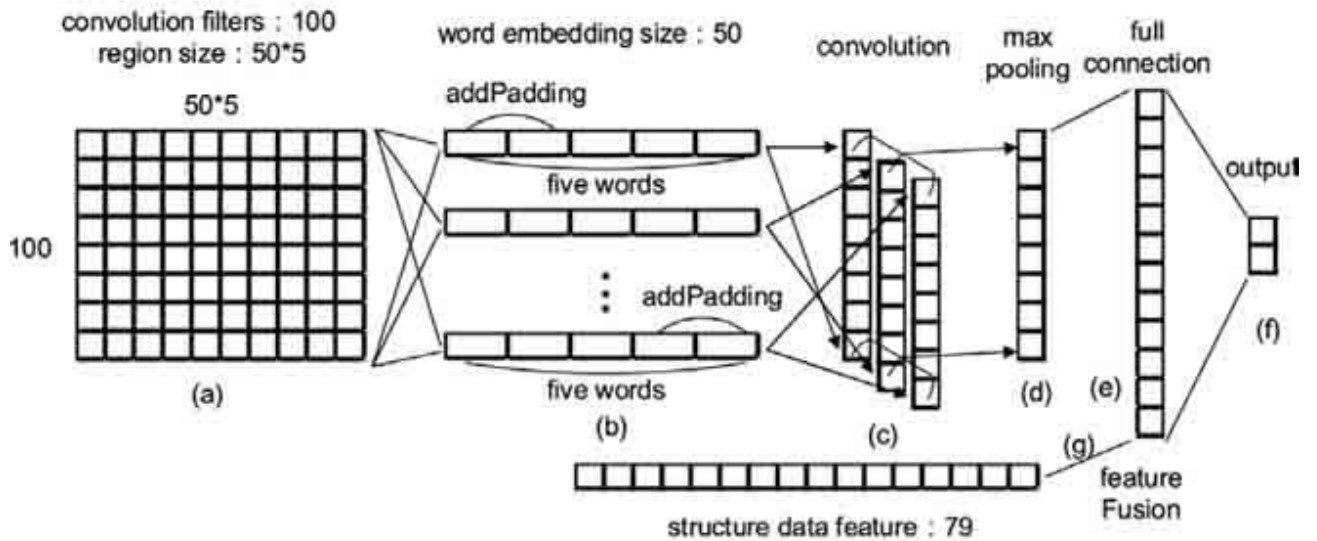


Рисунок 1.4 – Прогнозування захворювань за допомогою машинного навчання із великими даними від медичних спільнот [21]

Наскільки нам відомо, жодна з існуючих робіт не зосереджена на обох типах даних у сфері аналітики медичних великих даних. Порівняно з кількома типовими алгоритмами прогнозування, точність прогнозу запропонованого нами алгоритму досягає 94,8% зі швидкістю конвергенції, що є вищим, ніж у унімодального алгоритму прогнозування ризику захворювання на основі CNN.

Поширеність хронічних захворювань в Америці та інших країнах з хорошим рівнем життя зростає. У Китаї 86% смертей, пов'язаних із захворюваннями, пов'язані з хронічними захворюваннями [21]. Раннє виявлення захворювань у процесі діагностики та прогнозування має вирішальне значення для охорони здоров'я.

З величезним потоком медичних даних, які розробляються та зберігаються для досліджень [20], медичні дані зберігаються в електронних медичних записах (EHR) [21].

Моделі машинного навчання (ML) можуть автоматично отримувати відповідні характеристики з величезних наборів даних пацієнтів, які зберігаються в EHR [20]. Використовуючи відповідні функції з EHR, модель

ML може потім додатково допомогти у виявленні захворювань шляхом моніторингу даних і прогнозування потенційних захворювань.

Машинне навчання та глибоке навчання (DL) допомагають комп'ютерній моделі «адаптуватися до нових обставин, а також виявляти та екстраполювати шаблони» [41] шляхом навчання з необроблених даних. Кажуть, що модель навчається, якщо міра продуктивності цієї моделі збільшується для завдання з досвідом [38] і якщо модель може бути більш корисною для виконання завдання [37].

Для конкретного завдання модель ML буде подаватися прикладами, що містять різні функції як вхідні дані та мітки як бажані результати. Наприклад, пікселі з оцифрованих зображень людського мозку, зчитані патологоанатомами, перетворюються на особливості, а сліди ракових клітин, присутні в мозку, встановлюються як мітки [40].

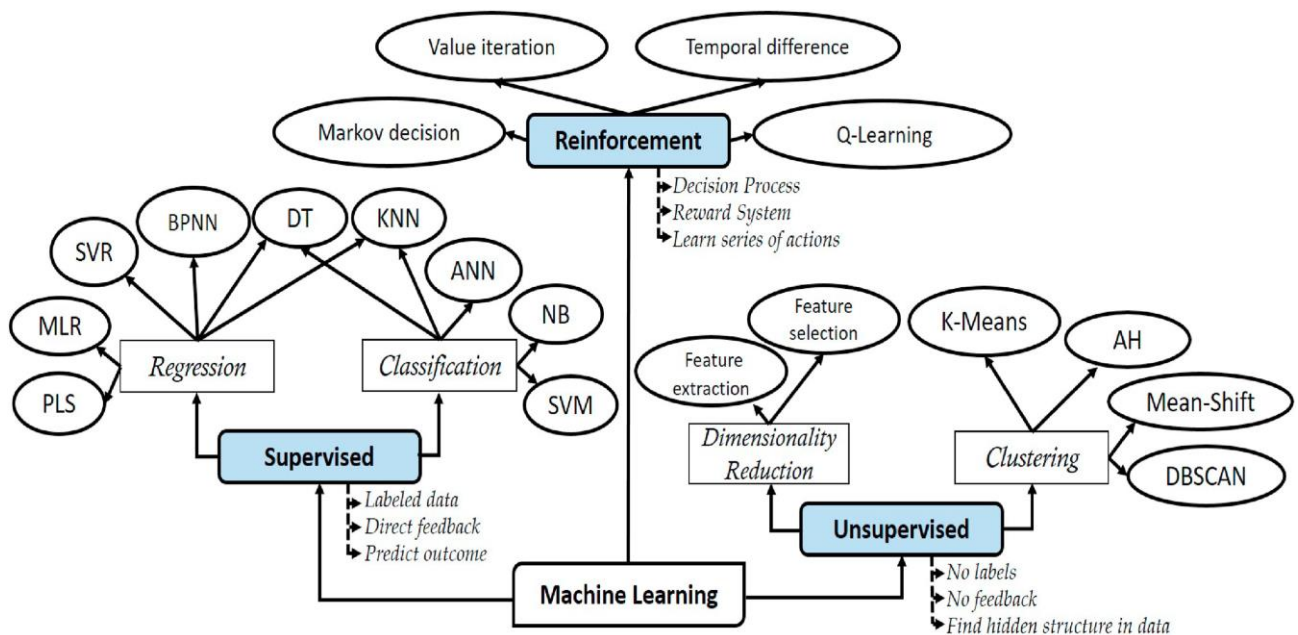


Рисунок 1.5 – Види машинного навчання [40]

Існують деякі варіації щодо того, як визначаються типи методів ML (тобто на основі проблеми та змісту необхідних результатів). На рис. 1.5 показано детальні зв'язки між типами, методами та техніками ML.

Потім модель навчання тренується з використанням різних типів методів навчання, у цьому випадку методів виявлення об'єктів, щоб надійно передбачити ракові клітини на основі нових, ніколи не бачених вхідних даних людського мозку.

Як згадувалося вище, машинне навчання вважається популярним додатком штучного інтелекту, у якому пристрої, програмне забезпечення та комп'ютери працюють через когнітивні функції (тобто дуже близько до принципу людського мозку). Останнім часом майже всі поля містять принаймні один із методів машинного навчання, як-от фільтрування електронного спаму та зловмисного програмного забезпечення, з якими ми маємо справу щодня, не усвідомлюючи цього. Додаткові приклади додатків ML показані на рис. 1.6. У цьому огляді обговорюється робота ML у галузі медицини, зокрема в охороні здоров'я, обробці зображень і діагностиці.

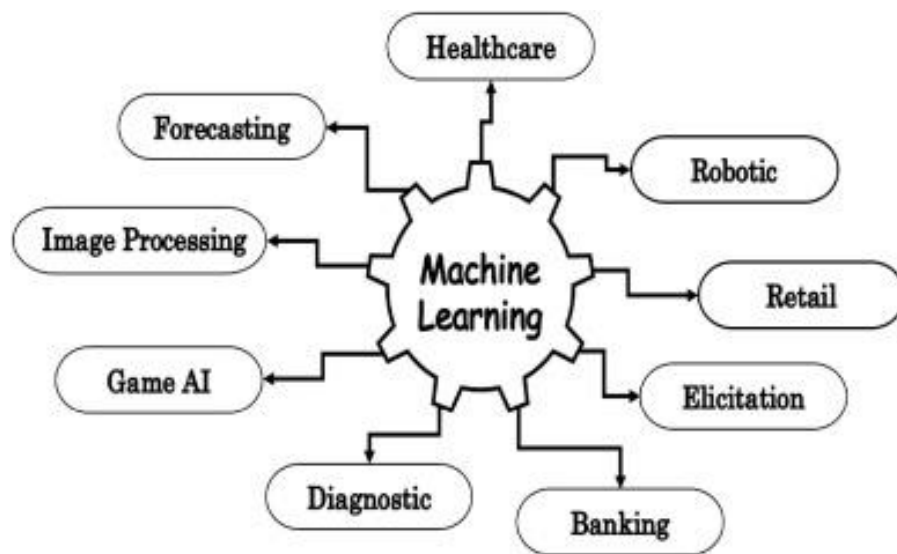


Рисунок 1.6 – Програми машинного навчання

У галузі медицини моделі ML вивчають знання предметної області з даних будь-де, від спостережень на рівні лікаря (нотатки лікаря) до медичної візуалізації (комп'ютерна томографія або КТ) [39]. Алгоритми аналізують усі дані, щоб виділити лише дескриптори (ознаки), які допоможуть точно передбачити результат. Візуалізацію процесу машинного навчання, починаючи з підготовки даних для оцінки моделі, можна побачити на рис. 1.7.

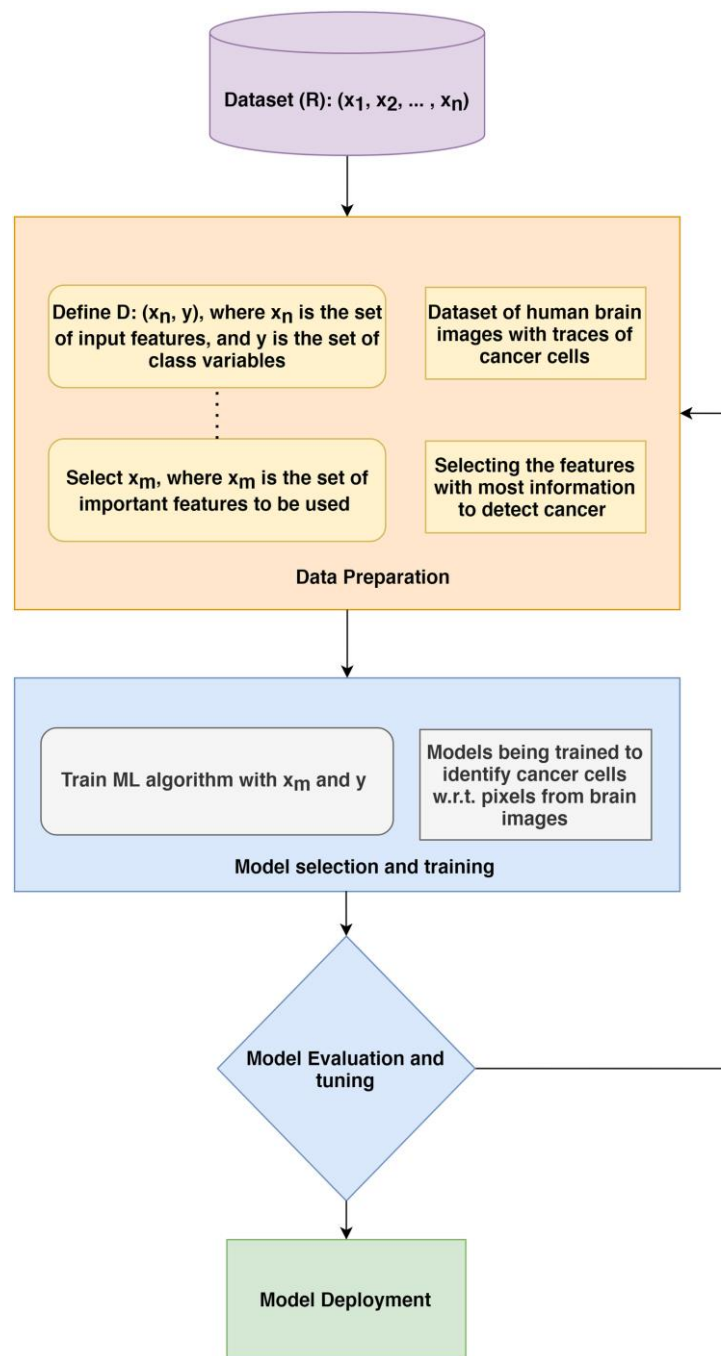


Рисунок 1.7 – Процес розробки моделі машинного навчання в галузі медицини

Машинне навчання адаптує кілька підходів до навчання, які є контрольованим і неконтрольованим навчанням, які диференціюються на основі наявності зворотного зв'язку.

Навчання під наглядом – це коли моделі пропонується набір прикладів із відповідними мітками, і модель вчиться ідентифікувати правильну мітку [22]. Класифікація та регресія є найпоширенішими типами навчання під наглядом.

Моделі класифікації приймають дані як вхідні дані для класифікатора та призначають дані в певний клас, зазвичай між двома класами [14] .

Класифікаційні моделі, такі як машини опорних векторів, постійно демонструють точні результати для різних завдань діагностики захворювань [43] . Тоді як неконтрольоване навчання - це коли мітки для вхідних даних не надаються. Модель вивчає подібність серед вхідних даних і намагається передбачити мітку як результат на основі подібності [22] .

Кластеризація є прикладом неконтрольованого навчання, коли алгоритм групує схожі вхідні дані в кластер. Як приклад, кластеризація нечітких середніх успішно використовується для діагностики раку молочної залози [32] .

Велика кількість методів машинного навчання, які використовують будь-який із цих підходів, використовуються в медицині та охороні здоров'я. Глибоке навчання, вдосконалена форма машинного навчання, реалізує більшу кількість рівнів, ніж типова модель машинного навчання, щоб витягувати глибші характеристики з даних, що дозволяє генерувати точніші прогнози.

Останні приклади використання методів штучного інтелекту в галузі медицини вказують на явний перехід від методів машинного навчання до методів глибокого навчання.

1.3. Прогнозування цукрового діабету за допомогою машинного навчання

В останні роки численні дослідження вивчали прогнозування діабету за допомогою кількох моделей машинного навчання (ML). У роботі [29] автори показали модель на основі нейронної мережі (NN) з точністю 88,6%. У своєму дослідженні вони використовували набір даних, отриманий з набору даних Pima Indian Diabetes (PID).

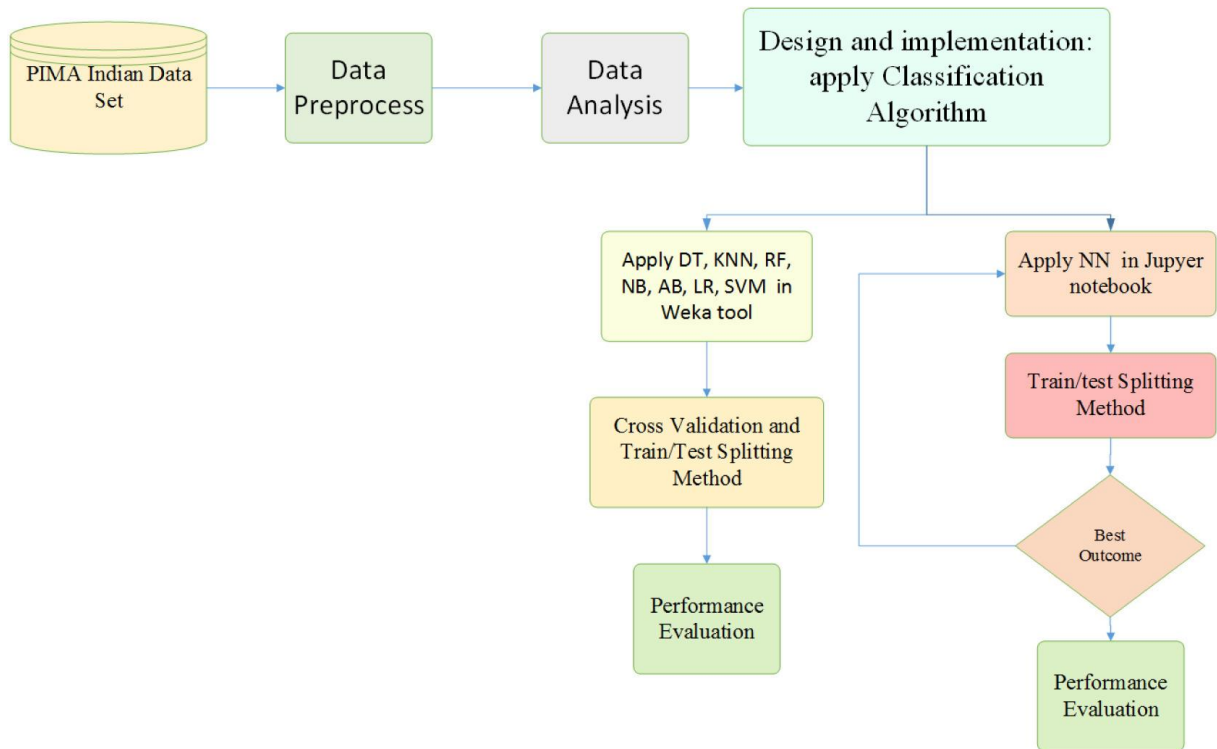


Рисунок 1.8 – Модель на основі нейронної мережі (NN) [29]

Незважаючи на те, що вони побудували модель за допомогою NN, вони не показали впливу функцій на цю модель у своїх дослідженнях, і точність моделі також була недостатньою.

У роботі [47] застосували методи інтелектуального аналізу даних і виявили, що випадковий ліс (RF) дає найкращі результати з точністю 97,4% при 10-кратній перехресній перевірці та 99% точністю при поєднанні тестування.



Рисунок 1.9 – Пропонована архітектура системи [47]

Вони використали набір даних, зібраний під час усних інтерв'ю з пацієнтами лікарні Sylhet Diabetes Hospital у Силхеті, Бангладеш. Вони показали хорошу точність, але їхній набір даних був незбалансованим. Вони не використовували жодних методів балансування даних/

У роботі [31] автори створили структуру для прогнозування діабету під назвою інтелектуальна структура прогнозування цукрового діабету (IDMPF) з точністю 83%. Вони використали набір даних під назвою індійський набір даних Pima Diabetes (PID). Результат їхньої моделі все ще недостатньо хороший, і є можливості для його покращення [31].

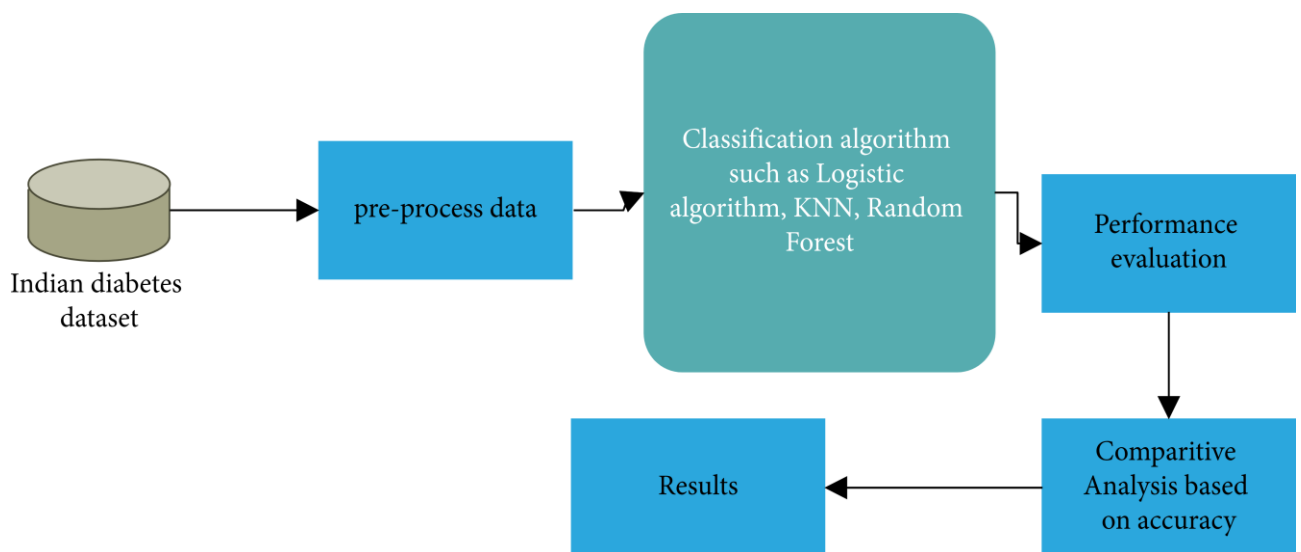


Рисунок 1.10 – Структура методів ML [31]

Запропонована структура розділена на різні фази. Блок-схема показана на рис. 1.9. Для всієї реалізації використовувався Python Jupyter Note. Для аналізу даних використовувалися різні пакети, такі як NumPy, pandas, scikit і Matplotlib. Завдання, що виконується на кожному етапі, і відповідні функції, досліджені за допомогою наборів інструментів Python.

Іслам та ін. запропонували модель для прогнозування діабету 2 типу в майбутньому, і вони досягли 95,94% точності [28]. Зібраний набір даних, використаний у їхньому дослідженні, був із дослідження серця в Сан-Антоніо, широко призначеного дослідження. Дослідження було успішним, але кількість

використаних функцій становила лише 11, що є недостатньою кількістю для створення та перевірки моделі машинного навчання.

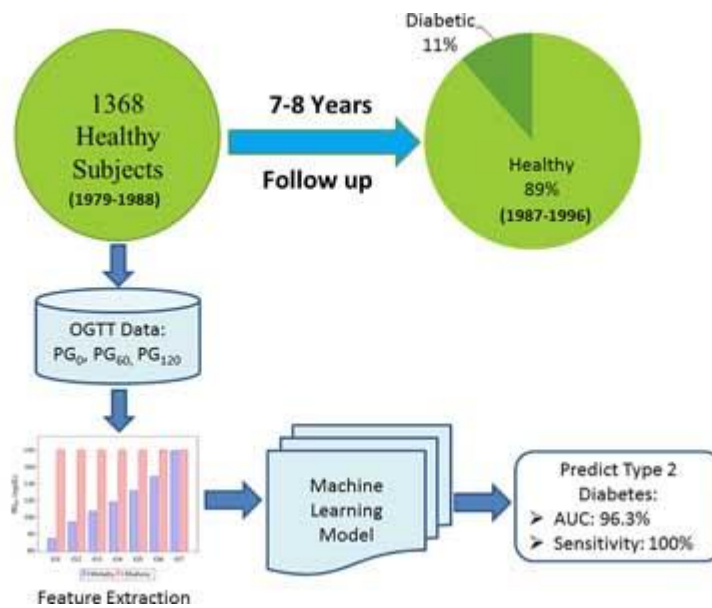


Рисунок 1.11 – Сучасні методи прогнозування майбутнього прогресування діабету 2 типу [28]

У цьому дослідженні автори пропонуємо два нових підходи до виділення ознак для пошуку найкращих факторів ризику з подальшим застосуванням конвеєра машинного навчання для довгострокового прогнозування діабету 2 типу. Запропоновані методи були оцінені з використанням даних довгострокового клінічного дослідження, відомого як San Antonio Heart Study. Запропонована ними модель змогла досягти 95,94% точності в прогнозуванні того, чи захворіє людина на діабет 2 типу протягом наступних 7-8 років чи ні.

Хасан та ін. зібрали класифікатори, щоб запропонувати модель з AUC 0,95. У цій роботі вони використовували набір даних Pima Indian Diabetes (PID) [24].

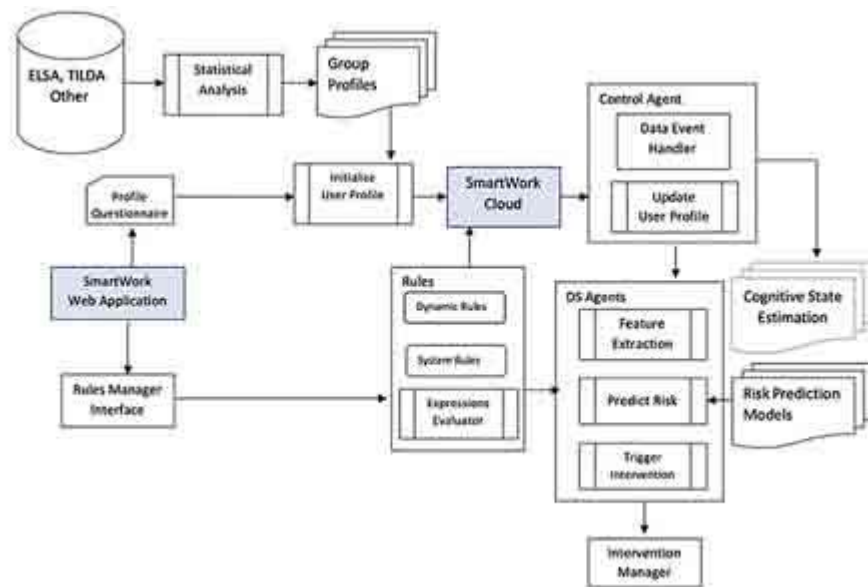


Рисунок 1.12 – Інструменти машинного навчання для довгострокового прогнозування ризику діабету 2 типу [24]

У зазначеній роботі розглядається створення набору даних, вибір функцій і класифікація з використанням різних моделей керованого машинного навчання (ML). Пропонується модель ML WeightedVotingLRRFs ансамблю для покращення прогнозування діабету, оцінюючи площу під кривою ROC (AUC) 0,884. Що стосується зваженого голосування, то оптимальні ваги оцінюються відповідною чутливістю та AUC моделі ML на основі двооб'єктивного генетичного алгоритму. Крім того, представлено порівняльне дослідження між Фінською оцінкою ризику діабету (FINDRISC) і Лестерською системою оцінки ризику та кількома моделями ML з використанням індуктивного та трансдуктивного навчання. Експерименти проводилися з використанням даних, отриманих з бази даних English Longitudinal Study of Aging (ELSA).

Фазакіс та ін. запропонували модель LRRFs ML із зваженим голосуванням із AUC 0,884 для прогнозування діабету 2 типу [24]. Використали набір даних щодо діабету, проведений у 2009–2012 роках, отриманий із Національного опитування щодо здоров'я та харчування. Набір даних складається з 6561 респондента з 657 діабетиків і 5904 контрольної групи. Модель LR демонструє, що 7 факторів із 14 таких як вік, освіта, ІМТ, систолічний АТ, діастолічний АТ, прямий холестерин і загальний холестерин є

факторами ризику діабету. Загальний ACC системи на основі ML становить 90,62% . Поєднання вибору функцій на основі LR і класифікатора на основі радіочастот дає 94,25% ACC і 0,95 AUC для протоколу K10.

Встановлено, що поєднання класифікатора на основі LR і RF працює краще. Ця комбінація буде дуже корисною для прогнозування пацієнтів з діабетом.

Набір даних, зібраний у їхньому дослідженні, був з бази даних English Longitudinal Study of Aging (ELSA). У цьому дослідженні не було жодних методів аналізу ознак.

Ахмед та ін. запропонував засновану на машинному навчанні (ML) модель Fused Model for Diabetes Prediction (FMDP) і отримав точність 94,87% [13].

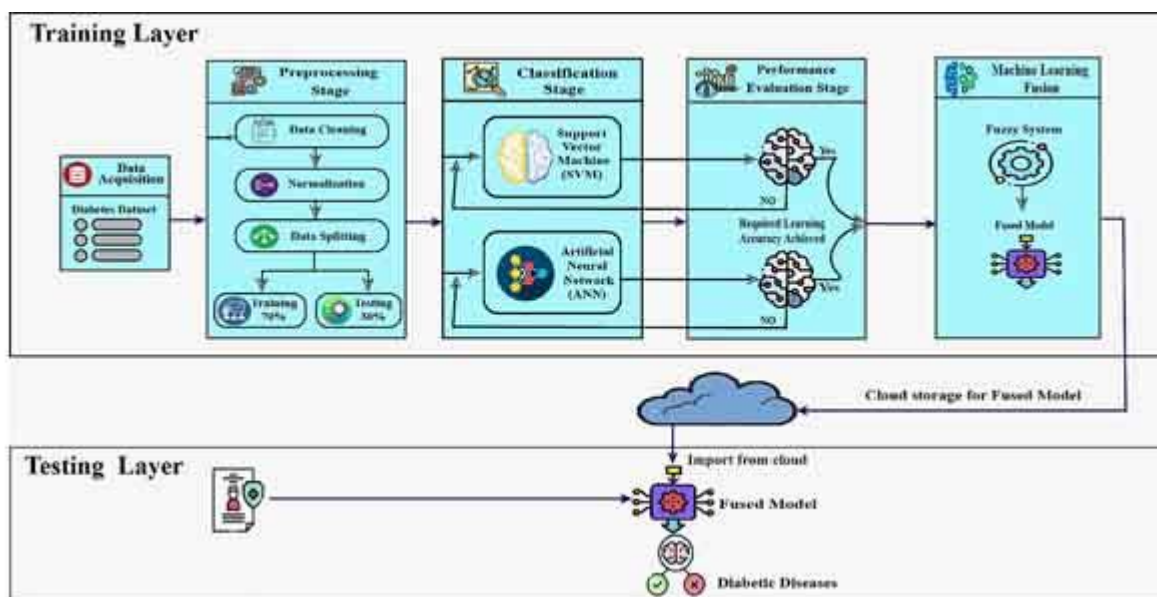


Рисунок 1.13 – Прогнозування діабету за допомогою Fused Machine Learning

Для цього дослідження вони використали набір даних, зібраний у лікарні Силхет, Бангладеш. Їхній метод дослідження був добре розроблений, але вони не показали жодного впливу функцій на модель і не проводили аналіз функцій.

Maniruzzaman та ін. (2020) представив модель, що поєднує вибір ознак логістичної регресії (LR) і випадковий ліс (RF), яка дала точність 94,25% і AUC 0,95 [36]. У їх дослідженні використовували Національне дослідження здоров'я

та харчування, яке проводилося з 2009 по 2012 рік. Набір даних містить лише 14 ознак.

Barakat et al., у 2010 році, провели дослідження для прогнозування цукрового діабету за допомогою опорного вектора (SVM) з точністю 94%, чутливістю 93% і специфічністю 94% [16]. У цьому дослідженні вони не вводили жодних методів аналізу ознак.

Тому в цьому дослідженні ми запропонували модель на основі машинного навчання (ML) для прогнозування діабету на ранній стадії. В останні роки ML виявився дуже ефективним методом прогнозування захворювань. Сьогодні ML відіграє важливу роль у біомедичному секторі, щоб подолати традиційні методи діагностики, прогнозування захворювань та лікування. Отже, немає жодних сумнівів щодо використання моделі прогнозування на основі ML для прогнозування діабету. Наші внески згадуються наступним чином:

Побудова моделі ML, яка прогнозуватиме діабет, використовуючи соціально-демографічні характеристики, а не клінічні ознаки. Це актуально, оскільки не всі люди, особливо з країн з низьким рівнем доходу, мають доступ до їхніх клінічних ознак.

1.4. Завдання кваліфікаційної роботи

Прогнозування тривалості лікування ЦД у дітей є важливим для розробки індивідуального плану лікування. Діти з високим ризиком досягнення ремісії можуть отримувати менш агресивне лікування, яке може полегшити їхній повсякденний життя. В даний час існує кілька методів прогнозування тривалості лікування ЦД у дітей.

Отже, виконання кваліфікаційної роботи на тему «Використання методів обчислювального інтелекту для прогнозування тривалості лікування цукрового

діабету у дітей» є достатньо актуальним на даний час. Відповідно до цього нами сформульовано наступні завдання роботи:

- ✓ виконати аналіз існуючих методів прогнозування тривалості лікування цд у дітей;
- ✓ вибрати алгоритми ML для прогнозування тривалості лікування цукрового діабету у дітей;
- ✓ провести збір та виконати підготовку даних для навчання ML-моделі;
- ✓ обґрунтувати параметри ML-моделі та здійснити оцінку точності ML-моделі;
- ✓ розробити заходи з охорони праці та безпеки у надзвичайних ситуаціях.

Результати роботи можуть бути використані для розробки більш ефективних методів лікування цукрового діабету у дітей.

РОЗДІЛ 2.

ОСОБЛИВОСТІ ПРОГНОЗУВАННЯ ТРИВАЛОСТІ ЛІКУВАННЯ ЦУКРОВОГО ДІАБЕТУ У ДІТЕЙ ТА ВИБІР МЕТОДІВ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ

2.1. Особливості прогнозування тривалості лікування цукрового діабету у дітей та вибір методів обчислювального інтелекту

Цукровий діабет є хронічним захворюванням, що характеризується порушенням обміну глюкози в організмі. У дітей цукровий діабет найчастіше розвивається у вигляді 1-го типу, який є аутоімунним захворюванням. Лікування цукрового діабету у дітей полягає в інсулінотерапії, а також у корекції способу життя.

Тривалість лікування цукрового діабету у дітей є важливим фактором, що впливає на якість життя дитини та її сім'ї. Довге лікування може призвести до розвитку ускладнень від цукрового діабету, таких як діабетична ретинопатія, нефропатія, нейропатія тощо.

Прогнозування тривалості лікування цукрового діабету у дітей є непростюю задачею. Це пов'язано з тим, що на тривалість лікування впливає безліч факторів, включаючи:

- вік дитини на момент встановлення діагнозу;
- тип цукрового діабету;
- ступінь тяжкості захворювання;
- співіснування інших захворювань;
- прихильність до лікування.

Використання методів обчислювального інтелекту може допомогти у прогнозуванні тривалості лікування цукрового діабету у дітей. Методи обчислювального інтелекту дозволяють обробляти великі обсяги даних та виявляти закономірності, які неможливо виявити за допомогою традиційних методів.

Дослідження показали, що методи обчислювального інтелекту можуть бути ефективними для прогнозування тривалості лікування ЦД у дітей. Наприклад, у дослідженні, проведеному в Ізраїлі, було показано, що метод обчислювального інтелекту на основі нейронних мереж може прогнозувати тривалість лікування цукрового діабету 1-го типу у дітей з точністю 80% [50].

У цьому дослідженні автори використовували нейронну мережу для прогнозування тривалості лікування цукрового діабету 1-го типу у дітей. Набір даних для навчання мережі включав інформацію про 100 дітей з цукрового діабету 1-го типу. Мережа була навчена прогнозувати тривалість лікування на основі таких факторів, як вік дитини на момент встановлення діагнозу, рівень глюкози в крові, рівень глікозильованого гемоглобіну (HbA1c) тощо. Автори дослідження показали, що нейронна мережа може прогнозувати тривалість лікування цукрового діабету 1-го типу у дітей з точністю 80%. Це означає, що мережа може правильно прогнозувати, чи дитина буде лікуватися від цукрового діабету протягом 5 років, з точністю 80%. Це дослідження є важливим кроком у розробці методів прогнозування тривалості лікування цукрового діабету у дітей. Методи обчислювального інтелекту, такі як нейронні мережі, можуть бути ефективними для прогнозування тривалості лікування цукрового діабету у дітей, що може допомогти в розробці індивідуальних планів лікування та покращенні якості життя дітей з цією хворобою.

Вибір методів обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей залежить від кількох факторів, включаючи:

- доступність даних;
- складність задачі;
- необхідність уточнення прогнозу.

Для задач прогнозування тривалості лікування цукрового діабету у дітей можуть бути використані такі методи обчислювального інтелекту, які представлені на рис. 2.1.



Рисунок 2.1 – Методи обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей

Методи ML дозволяють навчити модель на основі набору даних, що містить інформацію про дітей з цукровим діабетом. Навчена модель може потім використовуватися для прогнозування тривалості лікування для інших дітей.

Методи NLP можуть бути використані для обробки медичної документації, що містить інформацію про дітей з цукровим діабетом. Наприклад, NLP-методи можуть використовуватися для визначення типу цукрового діабету, ступеня тяжкості захворювання тощо.

Методи CV можуть бути використані для обробки зображень, отриманих при медичних обстеженнях дітей з цукровим діабетом. Наприклад, CV-методи можуть використовуватися для визначення рівня глюкози в крові, наявності ускладнень цукрового діабету тощо.

Вибір конкретного методу обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей повинен здійснюватися з урахуванням всіх факторів, зазначених вище.

У перспективі методи обчислювального інтелекту можуть бути використані для розробки індивідуальних планів лікування цукрового діабету у дітей. Це дозволить поліпшити якість життя дітей з цукровим діабетом та знизити ризик розвитку ускладнень захворювання.

2.2. Вибір методів обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей

Впровадження обчислювального інтелекту (ОІ) у медичні практики стає все більш актуальним завданням, особливо у сфері прогнозування тривалості лікування цукрового діабету у дітей. Діабет у дітей є серйозним індивідуальним захворюванням, і точне прогнозування тривалості лікування може значно поліпшити якість надання медичної допомоги та підвищити результати лікування.

Таблиця 2.1 – Методи обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей та вибір методів обчислювального інтелекту

Методи обчислювального інтелекту	Опис
Нейромережеві моделі	
Глибокі нейронні мережі	Використання ГНМ забезпечує можливість автоматичного вивчення характеристик пацієнтів та їх взаємодій з лікуванням. Зокрема, рекурентні нейронні мережі можуть враховувати динаміку хвороби та зміни показників з часом.
Глибокі згорткові нейронні мережі	Ці моделі ефективно аналізують зображення та структуровані дані, наприклад, медичні знімки чи результати аналізів. Застосування ГЗНМ дозволяє враховувати важливі візуальні аспекти здоров'я та визначати зміни в організмі.
Методи глибинного навчання	
Transfer Learning	Використання методів transfer learning дозволяє використовувати заздалегідь навчені моделі на великих

	наборах даних, що поліпшує ефективність прогнозування, особливо у випадках, коли доступ до обширних медичних даних обмежений.
Методи ансамблювання	
Random Forests та Gradient Boosting Machines	Використання ансамблевих методів дозволяє об'єднати прогнози з різних моделей, покращуючи точність та стабільність прогнозування.

Вибір нейромережових моделей для прогнозування тривалості лікування цукрового діабету у дітей обумовлений їхньою високою адаптивністю та здатністю до автоматичного вивчення складних зв'язків у даних. Глибоке навчання дозволяє автоматично виділяти важливі ознаки та робити точні прогнози навіть у складних клінічних випадках. Застосування нейромережових моделей дозволить оптимізувати індивідуальні підходи до кожного пацієнта, що є ключовим чинником для успішного лікування цукрового діабету у дітей.

Штучний інтелект (AI) має на меті емуляцію людського інтелекту в машинах. У комп'ютерних науках AI відноситься до вивчення «інтелектуальних агентів», тобто об'єктів, здатних сприймати навколишнє середовище і вживати заходів, щоб максимізувати свої шанси на досягнення певних цілей [48].

Машинне навчання (ML) – це галузь, яка фокусується на розробці та застосуванні методів, здатних навчатися на основі наборів даних. ML знаходить широке застосування в різних сферах, таких як розпізнавання мови, комп'ютерний зір, аналіз тексту, відеоігри, медичні науки та кібербезпека.

Глибоке навчання (DL) – це підмножина машинного навчання, яка досягає успіху в обробці неструктурованих даних. В даний час методи глибокого навчання перевершують традиційні підходи до машинного навчання. Моделі глибинного навчання черпають натхнення в структурі та функціональності людської нервової системи та мозку.

Моделі глибинного навчання черпають натхнення в структурі та функціональності людської нервової системи та мозку. Ці моделі використовують вхідні, приховані та вихідні шари для організації блоків обробки. У межах кожного шару вузли або блоки з'єднані з вузлами або блоками на рівні нижче, і кожному з'єднанню присвоюється вагове значення. Блоки підсумовують вхідні дані після множення їх на відповідні ваги.

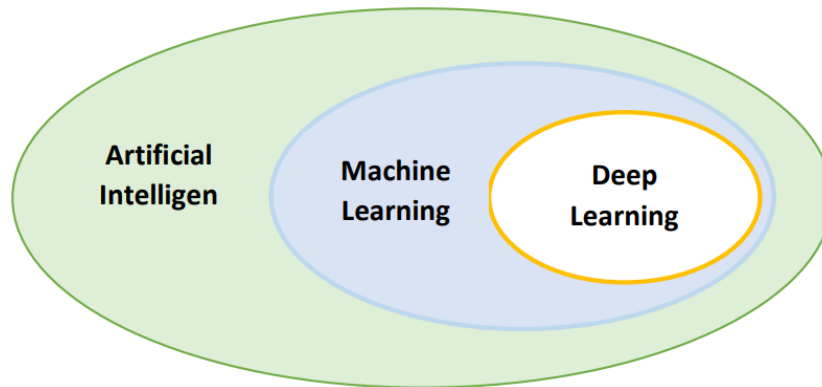


Рисунок 2.2 – Взаємозв'язок між AI, ML та DL

На рис. 2.2 представлено взаємозв'язок між AI, ML та DL. Слід зазначити, що машинне навчання та глибоке навчання є підгалуззями штучного інтелекту.

Глибоке навчання (DL) передбачає процес вивчення ієрархічних представлень даних за допомогою архітектур з декількома прихованими шарами. З розвитком високопродуктивних обчислювальних засобів методи глибокого навчання з використанням глибоких нейронних мереж набувають все більшої популярності [5]. В алгоритмі глибокого навчання дані проходять через декілька шарів, кожен з яких поступово виокремлює ознаки та передає інформацію наступному шару. Початкові шари виділяють низькорівневі характеристики, які потім об'єднуються наступними шарами для формування всебічного представлення [3].

У традиційних методах машинного навчання завдання класифікації зазвичай передбачає послідовний процес, який включає попередню обробку, вилучення ознак, ретельний відбір ознак, навчання та класифікацію. Ефективність методів машинного навчання значною мірою залежить від

точного вибору ознак, оскільки упереджений вибір ознак може призвести до неправильної класифікації класів. На противагу цьому, моделі глибокого навчання дозволяють проводити одночасне навчання і класифікацію, усуваючи необхідність в окремих етапах. Ця можливість робить глибоке навчання особливо вигідним для автоматизації вивчення ознак у різноманітних завданнях.

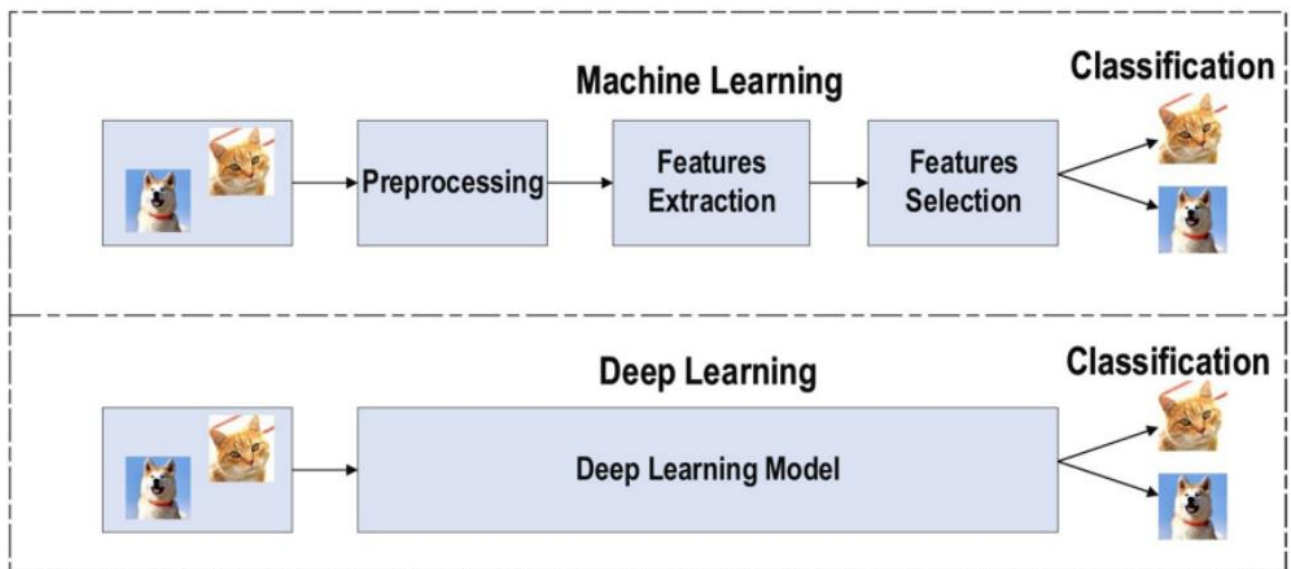


Рисунок 2.3 – Схема, яка ілюструє різницю між глибоким навчанням і традиційним машинним навчанням з точки зору вилучення ознак і навчання

В епоху глибокого навчання було розроблено широкий спектр методів та архітектур. Ці моделі можна умовно поділити на дві основні групи: дискримінативні (контрольовані) та генеративні (неконтрольовані) підходи. Серед дискримінативних моделей можна виділити дві великі групи: згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN). Крім того, генеративні підходи охоплюють різні моделі, такі як генеративні змагальні мережі (GAN) та автокодери (AE).

Для наших досліджень, які стосуються використання методів обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей та вибір методів обчислювального інтелекту, вибрано наступні 3 видів нейромережових моделей:

- 1) Проста штучна нейронна мережа (Simple Neural Network – SNN);

- 2) Рекурентна нейронна мережа (Recurrent Neural Network – RNN);
- 3) Автоенкодер (Autoencoder – AE).

2.3. Опис вибраних нейромережових моделей

2.3.1. Проста штучна нейронна мережа (Simple Neural Network)

Модель багат шарового перцептрона (MLP) – це тип штучної нейронної мережі (ANN) зі зворотним зв'язком, яка слугує базовою архітектурою для глибокого навчання або глибоких нейронних мереж (DNN). Вона працює як підхід до навчання під наглядом. MLP складається з трьох шарів: вхідного шару, вихідного шару та одного або декількох прихованих шарів. Це повністю пов'язана мережа, тобто кожен нейрон в одному шарі пов'язаний з усіма нейронами в наступному шарі.

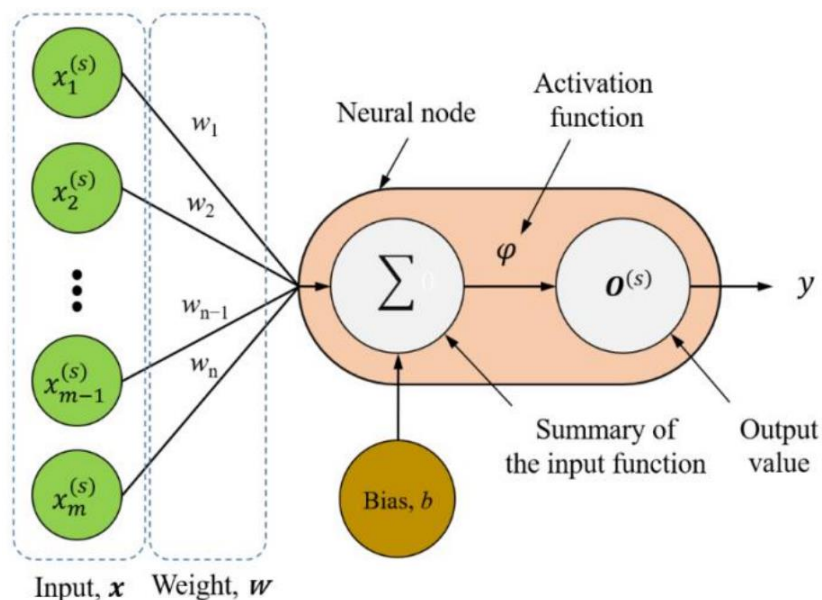


Рисунок 2.4 – Модель одонеуронного перцептрона

У MLP вхідний шар отримує вхідні дані і виконує нормалізацію ознак. Приховані шари, кількість яких може бути різною, обробляють вхідні сигнали. Вихідний шар приймає рішення або прогнози на основі обробленої інформації.

На рисунку 2.4 зображено модель однеї нейронного перцептрона, де функція активації φ (2.1) є нелінійною функцією, яка використовується для відображення функції підсумовування ($xw + b$) у вихідне значення y .

$$y = \varphi(xw + b), \quad (2.1)$$

У рівнянні (1) члени x , w , b та y , що представляють вхідний вектор, ваговий вектор, зсув та вихідне значення відповідно.

Рисунок 2.5 ілюструє структуру моделі багатозарового перцептрона (MLP).

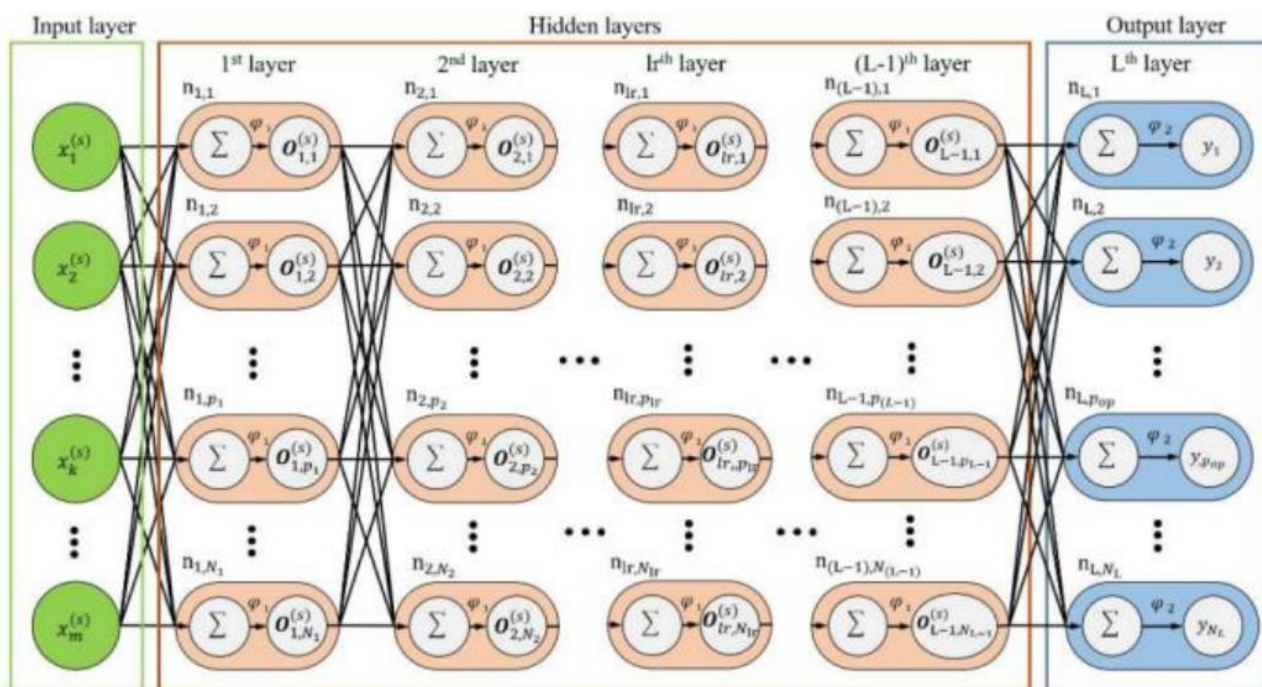


Рисунок 2.5 – Модель багатозарового перцептрона (MLP)

2.3.2. Рекурентна нейронна мережа (Recurrent Neural Network – RNN)

Рекурентні нейронні мережі (RNN) – це клас моделей глибокого навчання, які мають внутрішню пам'ять, що дозволяє їм фіксувати послідовні залежності. На відміну від традиційних нейронних мереж, які розглядають

вхідні дані як незалежні об'єкти, RNN враховують часовий порядок вхідних даних, що робить їх придатними для завдань, пов'язаних з послідовною інформацією. Використовуючи цикл, ANN застосовують одну і ту ж операцію до кожного елемента в серії, при цьому поточні обчислення залежать як від поточного входу, так і від попередніх обчислень.

Однак обмеженням простих ANN є їхня короткочасна пам'ять, що обмежує їхню здатність зберігати інформацію в довгих послідовностях. Для подолання цієї проблеми були розроблені більш досконалі варіанти ANN, серед яких Long ShortTerm Memory (LSTM), двонаправлений LSTM, Gated Recurrent Unit (GRU), двонаправлений GRU, Байєсівські ANN та інші.

На рисунку 2.6 зображено просту рекурентну нейронну мережу, де внутрішня пам'ять (ht) обчислюється за допомогою рівняння (2.2).

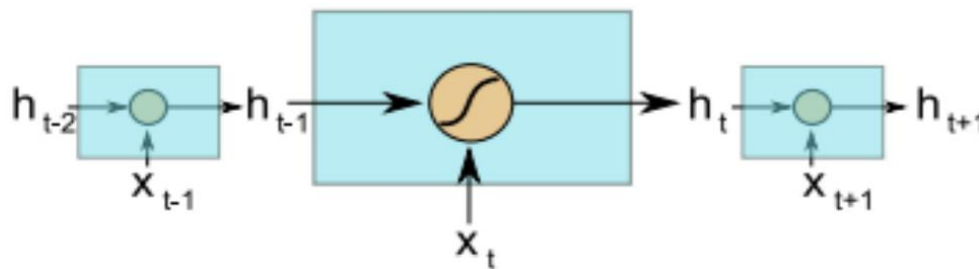


Рисунок 2.6 – Модель простої рекурентної нейронної мережі (RNN)

$$ht = g(Wxt + Uht + b), \quad (2.2)$$

У цьому рівнянні $g()$ представляє функцію активації (зазвичай гіперболічний тангенс), U і W – вагові матриці для прихованого стану (h), b – член зсуву, а x позначає вхідний вектор.

ANN виявилися потужними моделями для обробки послідовних даних, використовуючи їхню здатність фіксувати залежності в часі. Різні типи моделей ANN, такі як LSTM, двонаправлений LSTM, GRU та двонаправлений GRU, були розроблені для вирішення конкретних завдань у різних сферах застосування.

2.3.3. Автоенкодер (Autoencoder – AE)

Концепція автокодера виникла як нейронна мережа, призначена для реконструкції вхідних даних. Її основна мета полягає в тому, щоб навчитися осмисленому представленню даних у неконтрольований спосіб, що може мати різні застосування, включаючи кластеризацію. Автокодер – це нейронна мережа, яка прагне відтворити вхідні дані на виході. Вона складається з внутрішнього прихованого шару, який визначає код, що представляє вхідні дані.

Мережа автокодера складається з двох основних компонентів: функції кодера, що позначається як $z = f(x)$, і функції декодера, яка генерує реконструкцію, що позначається як $r = g(z)$.

Функція $f(x)$ перетворює точку даних x з простору даних у простір ознак, тоді як функція $g(z)$ перетворює z з простору ознак назад у простір даних для реконструкції вихідної точки даних x . У сучасних автокодерах ці функції:

$$z = f(x) \text{ і } r = g(z), \quad (2.3)$$

Вони розглядаються як стохастичні функції, представлені як *encoder* ($z|x$) та *decoder* ($r|z$), відповідно, де r позначає реконструкцію x . Рисунок 2.11 ілюструє модель автокодера.

Моделі автокодерів знаходять застосування в різних задачах некерованого навчання, таких як генеративне моделювання, зменшення розмірності, вилучення ознак, виявлення аномалій або викидів, а також деноаузація. Загалом, моделі автокодерів можна розділити на дві великі групи: Регулярні автокодери, які є цінними для навчання репрезентацій для подальших завдань класифікації, та варіаційні автокодери, які можуть функціонувати як генеративні моделі.

Прикладами регуляризованих моделей автокодерів є розріджений автокодер (SAE), скорочувальний автокодер (CAE) та деномінуючий автокодер (DAE).

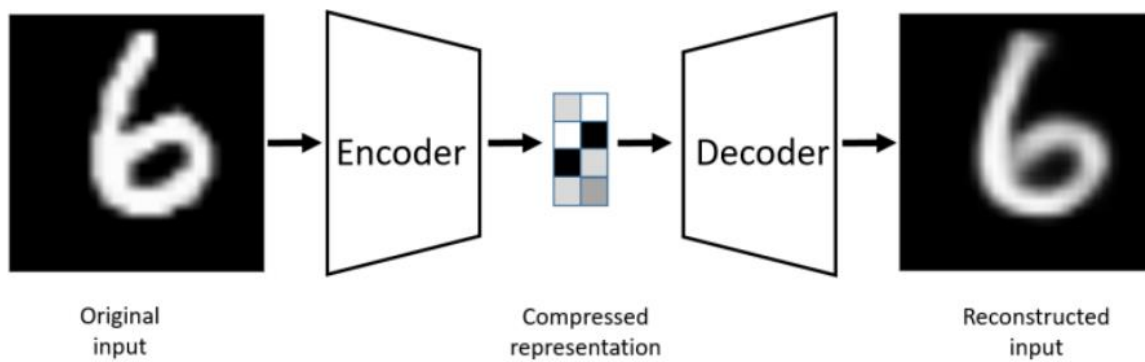


Рисунок 2.7 – Схема автоенкодера (Autoencoder – AE)

Варіаційний автокодер (VAE) – це генеративна модель, яка використовує імовірнісні розподіли, такі як середнє та дисперсія гаусівського розподілу, для генерації даних. VAE забезпечують принципову основу для вивчення моделей з глибокими латентними змінними та пов'язаних з ними моделей виведення. VAE складається з двох пов'язаних, але незалежно параметризованих моделей: кодера або моделі розпізнавання і декодера або генеративної моделі. Під час ітерацій навчання «максимізації очікування» генеративна модель отримує від моделі розпізнавання наближену апостеріорну оцінку своїх латентних випадкових величин, яку вона використовує для оновлення своїх параметрів. І навпаки, генеративна модель слугує каркасом для моделі розпізнавання, дозволяючи їй вивчати змістовні представлення даних, такі як потенційні мітки класів. З точки зору правила Байеса, модель розпізнавання є приблизно оберненою до генеративної моделі.

2.3. Збір даних та вибір інструментарію для прогнозування тривалості лікування цукрового діабету у дітей та вибір методів обчислювального інтелекту

Для нашого дослідження ми зібрали дані однієї із лікарень (ендокринологічного відділення клініки «Оксфорд Медікал») м. Львів. Вони

стосувалися лікування цукрового діабету 778 дітей віком до 19 років за 9 атрибутами. Визначено набір атрибутів, що включають дату госпіталізації, дату виписки, відділення, дату народження, температуру, зріст, стать, вагу, час перебування в лікарні та інші атрибути, які можуть впливати на тривалість лікування. Дані отримані із лікарні та містять інформацію про пацієнтів і їхнє лікування.

Після очищення та агрегації даних ми отримали 481 одиницю даних. Дані були введені в Jupyter – середовище інтерактивної розробки для блокнотів, коду та даних на мові Python.

Вибрано три типи нейромережових моделей, кожна з яких може допомогти вирішити конкретні аспекти задачі: 1) проста штучна нейронна мережа (SNN) – загально призначена модель для вирішення різноманітних задач; 2) рекурентна нейронна мережа (RNN) – корисна для обробки послідовностей та часових рядів; 3) автоенкодер (AE) – використовується для отримання компактної репрезентації вхідних даних. Ці моделі вибрані для того, щоб вони могли адаптуватися до різноманітності даних та виявляти складні залежності для точного прогнозування тривалості лікування.

Зважаючи на те, що кожен з вибраних методів використовує бібліотеку Keras для реалізації нейромережі, давайте детально опишемо параметри та компоненти, які використовуються для кожного методу.

Моделі побудовано із використанням бібліотеки Keras із її модулями Sequential та Dense. Її компонентами є Sequential – лінійний стек шарів, Dense – пов'язаний шар нейронів, LSTM – шар довготривалої короткочасної пам'яті для обробки послідовних даних, Conv2D – згортковий шар для обробки даних, input_dim – кількість входів на першому шарі, activation – функція активації (ReLU на прихованих шарах, linear на вихідному), activation – функція активації (tanh для LSTM, linear на вихідному), kernel_size – розмір ядра для згортки, Flatten – вирівнювання для подачі даних у Fully Connected шар, loss – функція втрат (середньоквадратична помилка для регресії), optimizer – оптимізатор для навчання моделі (Adam).

РОЗДІЛ 3.

РЕЗУЛЬТАТИ СТВОРЕННЯ ТА ВИЗНАЧЕННЯ ТОЧНОСТІ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ ДЛЯ ПРОГНОЗУВАННЯ ТРИВАЛОСТІ ЛІКУВАННЯ ЦУКРОВОГО ДІАБЕТУ У ДІТЕЙ

3.1. Результати підготовки даних для навчання нейромережевих моделей

Нами використано бібліотеку Pandas для завантаження даних з файлу Excel в об'єкт DataFrame та виведення перших кілька рядків цього DataFrame (рис. 3.1).

```
In [5]: # Виведення останніх кількох рядків
df.tail()
```

```
Out[5]:
```

	Date_hospitalization	Date_discharge	Department	Date_birth	Temperature	Height	Human_gender	Weight	In_hospital	Type_settlement	Bed_days
774	2023-11-15 13:50:00	2023-12-18 14:57:00	Педіатрія	2014-06-06	36.7	141.6	Ч	44.0	в плановому порядку	Село	33
775	2023-12-12 12:35:00	2023-12-23 13:44:00	Педіатрія	2011-11-26	0.1	0.0	Ч	0.0	за терміновими показаннями	Село	11
776	2023-12-15 09:11:00	2023-12-23 14:47:00	Педіатрія	2020-07-13	36.8	97.0	Ж	15.3	за терміновими показаннями	Село	8
777	2023-12-13 11:46:00	2023-12-28 13:19:00	Педіатрія	2021-01-20	37.1	0.0	Ж	0.0	за терміновими показаннями	Місто	15
778	2023-12-15 02:55:00	2023-12-30 13:12:00	Педіатрія	2009-01-03	0.1	0.0	Ж	47.7	за терміновими показаннями	Місто	15

Рисунок 3.1 – Код та результати завантаження даних з файлу Excel в об'єкт DataFrame та виведення останніх рядків

Основна ідея тут полягає в тому, що `pd.read_excel()` використовується для читання даних з файлу Excel, і весь набір даних зберігається у змінній `df` (DataFrame). Потім `df.head()` виводить перші кілька рядків цього DataFrame для перевірки структури та вмісту даних.

Після цього нами проаналізовано унікальні значення та їх кількостей у кожному стовпці. Для прикладу покажемо стовбець «Human_gender» нашого DataFrame (рис. 3.2).

```
In [10]: # Виведення унікальних значень та їх кількостей у стовпці "Human_gender"
gender_counts = df['Human_gender'].value_counts()

# Виведення результатів
print("Унікальні значення та їх кількості у стовпці 'Human_gender':")
print(gender_counts)

Унікальні значення та їх кількості у стовпці 'Human_gender':
Human_gender
Ж      323
Ч      300
Name: count, dtype: int64
```

Рисунок 3.2 – Код та результати аналізу унікальних значень та їх кількості у стовпці «Human_gender»

Результат аналізу унікальних значень та їх кількостей у стовпці «Human_gender» показує, що є два унікальних значення (жіноча стать – «Ж» та чоловіча стать – «Ч»). Кількість спостережень для кожної статі також вказана:

- Ж (жіноча стать) – 323 спостереження;
- Ч (чоловіча стать) – 300 спостережень.

```
In [13]: # Створення словника для заміни значень
department_mapping = {'Педіатрія': 1, 'Реанімація/Відділення анестезіології та інтенсивної терапії': 2}
gender_mapping = {'Ж': 0, 'Ч': 1}
in_hospital_mapping = {'в плановому порядку': 1, 'за терміновими показаннями': 2}
type_settlement_mapping = {'Село': 1, 'Місто': 2}

# Заміна категоріальних значень на числові у вказаних стовпцях
df['Department'] = df['Department'].replace(department_mapping)
df['Human_gender'] = df['Human_gender'].replace(gender_mapping)
df['In_hospital'] = df['In_hospital'].replace(in_hospital_mapping)
df['Type_settlement'] = df['Type_settlement'].replace(type_settlement_mapping)

# Виведення оновленого DataFrame
print("Оновлений DataFrame:")
print(df.head())

# Виведення тексту, які категорії були замінені на які числа
print("\nЗаміна категорій на числа:")
print(f"Department: {department_mapping}")
print(f"Human_gender: {gender_mapping}")
print(f"In_hospital: {in_hospital_mapping}")
print(f"Type_settlement: {type_settlement_mapping}")

Оновлений DataFrame:
   Age  Department  Temperature  Height  Human_gender  Weight  In_hospital  \
2     5           1           36.7   114.0            0    18.4           1.0
3    10           1           36.6   138.0            0    26.0           1.0
6     4           1           36.6   103.5            0    19.4           1.0
7    14           1           36.6   173.0            1    49.0           1.0
12    6           1           36.6   111.0            1    17.4           1.0

   Type_settlement  Bed_days
2                 1         22
3                 2         11
6                 1         13
7                 2          6
12                2          8

Заміна категорій на числа:
Department: {'Педіатрія': 1, 'Реанімація/Відділення анестезіології та інтенсивної терапії': 2}
Human_gender: {'Ж': 0, 'Ч': 1}
In_hospital: {'в плановому порядку': 1, 'за терміновими показаннями': 2}
Type_settlement: {'Село': 1, 'Місто': 2}
```

Рисунок 3.3 – Код та результати заміни категоріальних значень на числові у вказаних стовпцях DataFrame

Ці дані можуть бути корисні при подальшому аналізі або використанні їх у моделях машинного навчання.

Наступний код (3.3) виконує заміну категоріальних значень на числові у вказаних стовпцях DataFrame, а потім виводить оновлений DataFrame та текст, що показує, які категорії були замінені на які числа. Розглянемо результати аналізу:

Оновлений DataFrame передбачає наступні зміни:

- значення в стовпці «Department» (відділення) були замінені з текстових (Педіатрія, Реанімація/Відділення анестезіології та інтенсивної терапії) на числові (1, 2);
- значення в стовпці «Human_gender» (стать) були замінені з текстових (Ж, Ч) на числові (0, 1);
- значення в стовпці «In_hospital» (госпіталізація) були замінені з текстових ("в плановому порядку", "за терміновими показаннями") на числові (1, 2);
- значення в стовпці «Type_settlement» (тип поселення) були замінені з текстових (Село, Місто) на числові (1, 2).

Аналіз результатів полягає в тому, що тепер усі значення у відповідних стовпцях є числовими, що може бути корисним при використанні їх у моделях машинного навчання. Така конвертація робить дані придатними для використання в алгоритмах, які працюють із числовими даними.

```
In [15]: # Список стовпців, які потрібно перевірити на пропуски
columns_to_check = ['Temperature', 'Height', 'Weight', 'Department', 'Human_gender', 'In_hospital', 'Type_settlement', 'Age']

# Перевірка пропусків у вказаних стовпцях
for column in columns_to_check:
    missing_values_count = df[column].isnull().sum()
    print(f"Кількість пропусків у стовпці '{column}': {missing_values_count}")

Кількість пропусків у стовпці 'Temperature': 0
Кількість пропусків у стовпці 'Height': 0
Кількість пропусків у стовпці 'Weight': 0
Кількість пропусків у стовпці 'Department': 0
Кількість пропусків у стовпці 'Human_gender': 0
Кількість пропусків у стовпці 'In_hospital': 0
Кількість пропусків у стовпці 'Type_settlement': 0
Кількість пропусків у стовпці 'Age': 0
```

Рисунок 3.4 – Код та результати пропуски на пропуски у стовпцях DataFrame

Результати перевірки на пропуски (рис. 3.4) у показує, що в жодному з вказаних стовпців («Temperature», «Height», «Weight», «Department», «Human_gender», «In_hospital», «Type_settlement», «Age») немає пропущених значень. Усі кількості пропусків дорівнюють нулю, що свідчить про те, що дані в цих стовпцях є повними і можуть бути використані для аналізу та моделювання.

Нами здійснено побудову гістограми віку пацієнтів, які лікувалися від цукрового діабету (рис. 3.5).

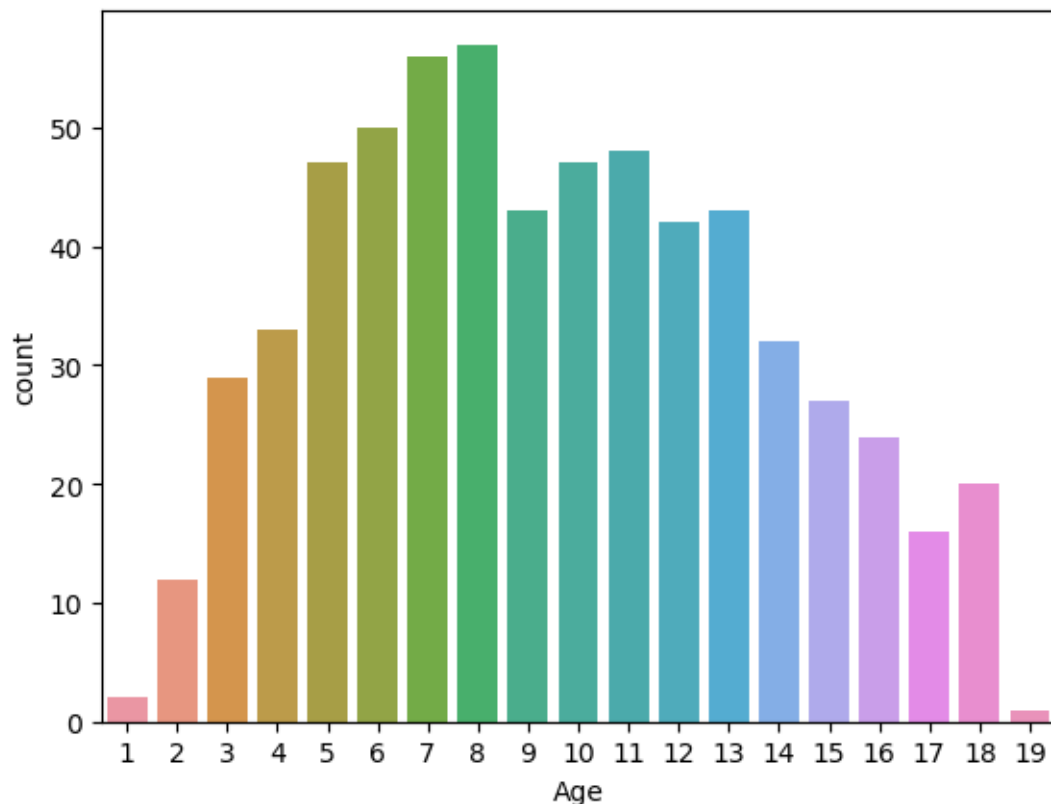


Рисунок 3.5 – Гістограма віку пацієнтів, які лікувалися від цукрового діабету

Для цього використали бібліотеку seaborn для побудови стовпчастої діаграми (countplot), де на вісі X відображаються унікальні значення стовпця "Age", а на вісі Y відображається кількість входжень кожного значення.

Цей графік дозволяє швидко оцінити, які вікові групи представлені у ваших даних та яка кількість спостережень припадає на кожну з них. Встановлено, що найбільше у віці від 5 до 13 років.

Також нами побудовано графік розподілу віку пацієнтів (стовпець "Age") з використанням гістограми та врахування тривалості лікування (стовпець "Bed_days") як відтінку кольору (використовуюючи параметр hue) (рис. 3.6).

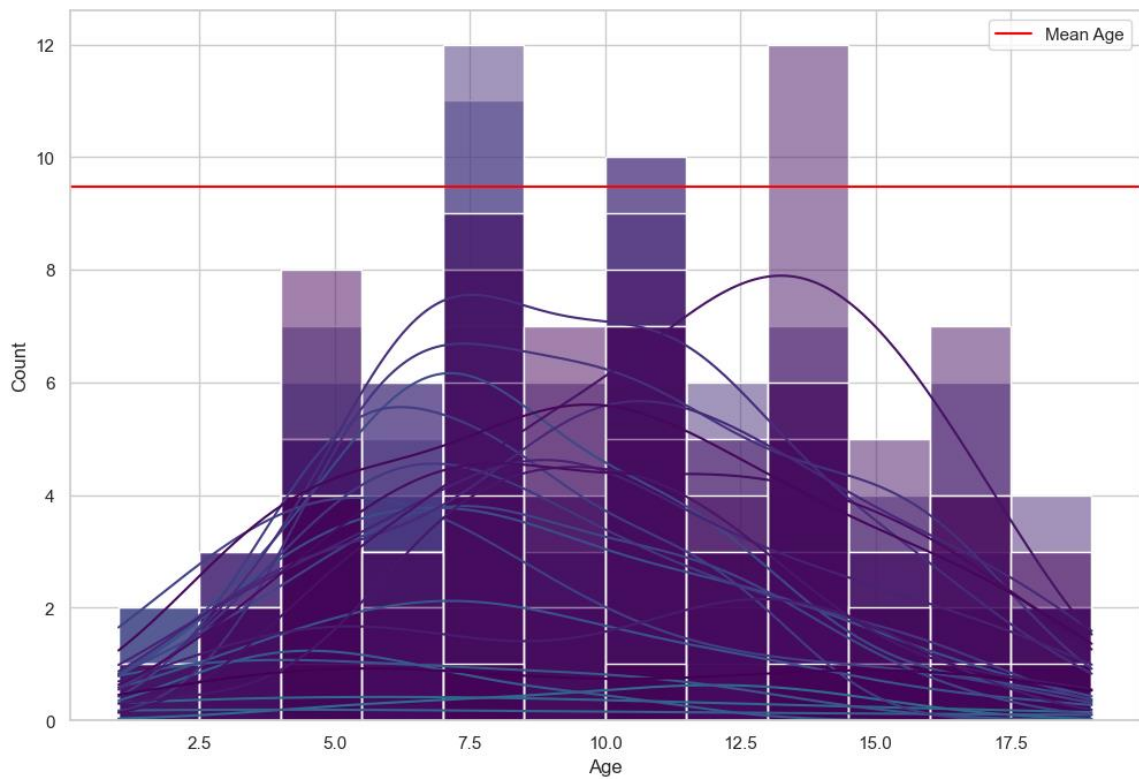


Рисунок 3.6 – Гістограма розподілу віку пацієнтів (стовпець "Age") із врахуванням тривалості лікування (стовпець "Bed_days")

Кожен стовпець у гістограмі відповідає конкретному віку, і кольори вказують на різні тривалості лікування для кожного віку. Графік дозволяє оцінити, як вік пацієнтів розподілений в наборі даних. З використанням кольорів можна оцінити, як тривалість лікування варіюється в залежності від віку. Встановлено, що найдовша тривалість лікування припадає на вік 7-8 років та 13-14 років. Лінія середнього віку (червона лінія) вказує на середній вік пацієнтів (9,5 років), що може бути важливою метрикою для загального розуміння даних.

3.2. Результати навчання нейромережових моделей для прогнозування тривалості лікування цукрового діабету у дітей

3.2.1. Результати навчання моделі з використанням Long Short-Term Memory (LSTM) мережі

Нами використано бібліотеки для машинного навчання, такі як Keras та scikit-learn, для навчання моделі з використанням Long Short-Term Memory (LSTM) мережі на задачі прогнозування тривалості лікування (рис. 3.7).

Проста штучна нейронна мережа (Simple Neural Network):

Одна або декілька прихованих шарів Функція активації: ReLU або Sigmoid Оптимізатор: Adam або SGD

```
In [46]: import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error
from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Оригінальні дані
X = train_clean.drop(columns=['Bed_days'])
y = train_clean['Bed_days']

# Нормалізація даних
scaler = StandardScaler()
X_normalized = pd.DataFrame(scaler.fit_transform(X), columns=X.columns)

# Розділення на навчальні та тестові дані
X_train, X_test, y_train, y_test = train_test_split(X_normalized, y, test_size=0.2, random_state=42)

# Перетворення даних для LSTM
X_train_lstm = X_train.values.reshape((X_train.shape[0], X_train.shape[1], 1))
X_test_lstm = X_test.values.reshape((X_test.shape[0], X_test.shape[1], 1))

# Створення та навчання LSTM моделі
model_rnn = Sequential()
model_rnn.add(LSTM(64, input_shape=(X_train_lstm.shape[1], 1), activation='tanh'))
model_rnn.add(Dense(1, activation='linear'))
model_rnn.compile(loss='mean_squared_error', optimizer=Adam())
history = model_rnn.fit(X_train_lstm, y_train, epochs=50, batch_size=32, validation_data=(X_test_lstm, y_test), verbose=1)

# Оцінка моделі на тестових даних
y_pred = model_rnn.predict(X_test_lstm)

# Виведення графіку з історією навчання
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

# Розрахунок метрик
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)

# Виведення тексту з результатами
print(f'Mean Squared Error: {mse:.3f}')
print(f'Mean Absolute Error: {mae:.3f}')
```

Рисунок 3.7 – Код навчання моделі з використанням Long Short-Term Memory (LSTM) мережі на задачі прогнозування тривалості лікування дітей із цукровим діабетом

Насамперед нами було вхідні дані нормалізовано за допомогою StandardScaler, щоб забезпечити стабільність та швидкодіючість навчання. Дані розділено на навчальний та тестовий набори для оцінки ефективності моделі. Здійснено перетворення даних для LSTM. Дані перетворюються для використання в LSTM, яка вимагає тривимірної форми (кількість прикладів, кількість ознак, часова вимірність). Для створення LSTM моделі використовується Sequential модель з одним шаром LSTM та одним вихідним Dense шаром з лінійною активаційною функцією. Оптимізатор – Adam, який адаптивно налаштовує швидкість навчання.

Модель компілюється та навчається на навчальних даних, історія навчання зберігається для подальшого аналізу. Модель оцінюється на тестових даних, і прогнозовані значення порівнюються зі справжніми за допомогою середньоквадратичної та середньоабсолютної помилок. Виводиться графік з історією навчання (втрати на тренувальних та валідаційних даних) (рис. 3.8).

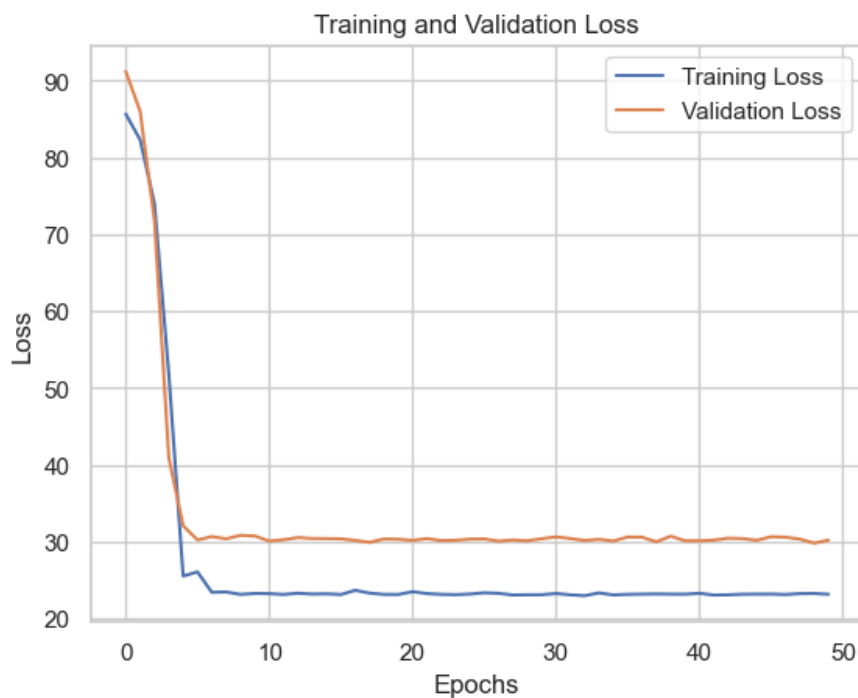


Рисунок 3.8 – Результати навчання моделі з використанням Long Short-Term Memory (LSTM) мережі на задачі прогнозування тривалості лікування дітей із цукровим діабетом

Оцінюючи результати навчання моделі за кожною із 50 епох, ми можемо зробити наступні спостереження. Функція втрат зменшується протягом перших епох і зберігає відносну стабільність пізніше. Це свідчить про те, що модель поступово навчається і адаптується до тренувальних даних.

Втрата на тренувальному та валідаційному наборах не показує суттєвого збільшення чи зменшення після певної кількості епох, що може свідчити про відсутність перенавчання. Втрата на валідаційному наборі трошки коливається, але загалом зберігається стабільною.

Кожна епоха займає приблизно 4 секунди для навчання. Це може бути важливим фактором, особливо якщо потрібно вирішити питання, чи можна дозволити більше епох для навчання. Функція втрат на тренувальному та валідаційному наборах стабілізується, що може свідчити про те, що оптимізатор (Adam) працює ефективно, і немає необхідності у великому числі епох.

Загалом, на основі наданих результатів можна сказати, що модель виявляється ефективною в навчанні, і додаткові налаштування або епохи можуть покращити її продуктивність.

3.2.2. Результати навчання моделі рекурентної нейронної мережі (Recurrent Neural Network - RNN)

Нами використано бібліотеку Keras для створення та навчання простої рекурентної нейронної мережі (RNN) для прогнозування значень тривалості лікування цукрового діабету у дітей на основі інших ознак (рис. 3.9).

Рекурентна нейронна мережа (Recurrent Neural Network - RNN):

LSTM або GRU шари Функція активації: Tanh або Sigmoid Оптимізатор: Adam або RMSprop

```
In [47]: import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error
from keras.models import Sequential
from keras.layers import SimpleRNN, Dense
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Оригінальні дані
X = train_clean.drop(columns=['Bed_days'])
y = train_clean['Bed_days']

# Нормалізація даних
scaler = StandardScaler()
X_normalized = pd.DataFrame(scaler.fit_transform(X), columns=X.columns)

# Розділення на навчальні та тестові дані
X_train, X_test, y_train, y_test = train_test_split(X_normalized, y, test_size=0.2, random_state=42)

# Перетворення даних для RNN
X_train_rnn = X_train.values.reshape((X_train.shape[0], 1, X_train.shape[1]))
X_test_rnn = X_test.values.reshape((X_test.shape[0], 1, X_test.shape[1]))

# Створення та навчання RNN моделі
model_rnn = Sequential()
model_rnn.add(SimpleRNN(64, input_shape=(X_train_rnn.shape[1], X_train_rnn.shape[2]), activation='tanh'))
model_rnn.add(Dense(1, activation='linear'))
model_rnn.compile(loss='mean_squared_error', optimizer=Adam())
history = model_rnn.fit(X_train_rnn, y_train, epochs=50, batch_size=32, validation_data=(X_test_rnn, y_test), verbose=1)

# Оцінка моделі на тестових даних
y_pred_rnn = model_rnn.predict(X_test_rnn)

# Виведення графіку з історією навчання
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

# Розрахунок метрик
mse_rnn = mean_squared_error(y_test, y_pred_rnn)
mae_rnn = mean_absolute_error(y_test, y_pred_rnn)

# Виведення тексту з результатами
print(f'Mean Squared Error (RNN): {mse_rnn:.3f}')
print(f'Mean Absolute Error (RNN): {mae_rnn:.3f}')
```

Рисунок 3.9 – Код навчання моделі рекурентної нейронної мережі (Recurrent Neural Network - RNN)

Вхідні дані нормалізуються за допомогою `StandardScaler`, щоб забезпечити стандартизацію і покращення навчання моделі. Дані розбиваються на тренувальний та тестовий набори зі співвідношенням 80/20. Вхідні дані для RNN перетворюються в формат (кількість зразків, кількість кроків, кількість ознак).

RNN модель створюється з одним шаром `SimpleRNN` та одним `Dense` шаром. Використовується функція активації 'tanh' для `SimpleRNN` та 'linear' для `Dense`.

Втрати обчислюються за допомогою середньоквадратичної помилки, і оптимізатор - Adam. Модель навчається протягом 50 епох з розміром партії 32. Після навчання модель оцінюється на тестових даних, і отримані прогнозовані значення зберігаються у `y_pred_rnn`.

Нами побудовано графік втрат на тренувальному та тестовому наборах даних, що забезпечує візуалізацію ефективності моделі під час навчання (рис. 3.10).

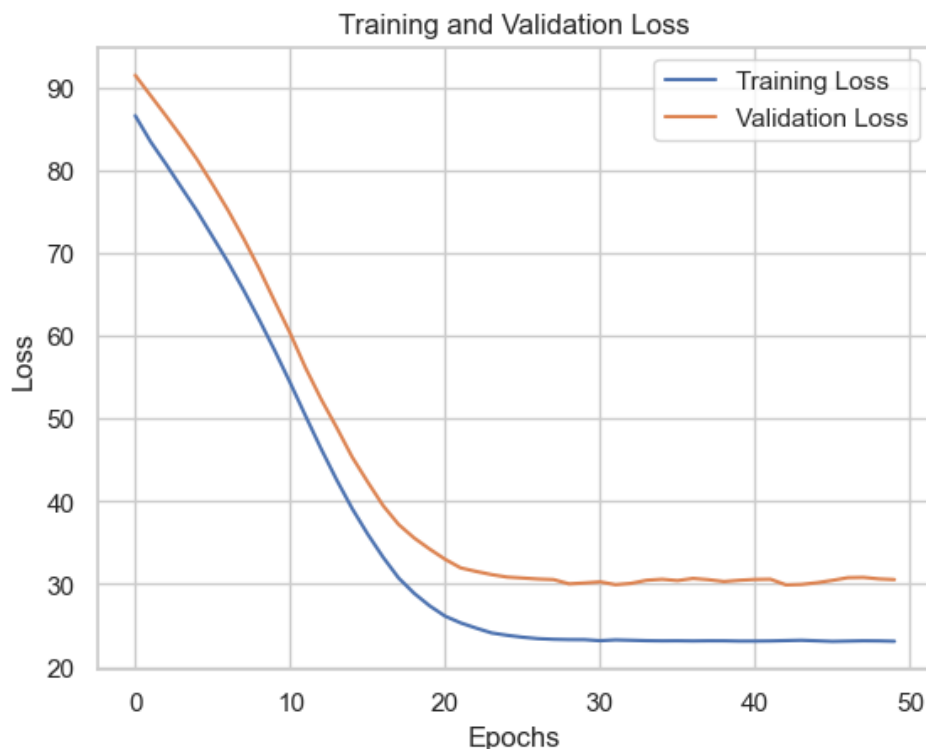


Рисунок 3.10 – Результати навчання моделі рекурентної нейронної мережі (Recurrent Neural Network - RNN)

Результати навчання можна проаналізувати, спостерігаючи за змінами втрат (loss) на тренувальному та валідаційному наборах даних протягом епох. Крива тренування (Training Loss) відображає, як змінюються втрати на тренувальному наборі під час навчання моделі. Крива валідації (Validation Loss) показує, як модель веде себе на тестових (валідаційних) даних.

Початкові втрати для тренування та валідації є високими, що може бути пов'язано з невідповідністю моделі до даних або неоптимальними параметрами. Протягом епохи втрати на обох наборах даних поступово зменшуються, що

свідчить про покращення навчання моделі. Після певної кількості епох може виникнути стійка тенденція до зменшення втрат або до їхнього стабілізування.

Початкові втрати (Epoch 1) на тренуванні – 86.6106 та валідації – 91.4965, а на останній епосі (Epoch 50) на тренуванні – 23.0831 та валідації – 30.5393.

Зменшення втрат свідчить про те, що модель навчається і покращується в прогнозуванні. Якщо бачимо, що втрати на валідаційному наборі даних також покращуються, це вказує на те, що модель уникне перенавчання. Для подальшого покращення можна розглянути збільшення кількості епох або оптимізацію гіперпараметрів моделі.

3.2.3. Результати навчання моделі автоенкодер (Autoencoder)

Нами використано бібліотеку Keras для створення автоенкодера, що забезпечує навчання моделі без нагляду (unsupervised learning) та використовує його для реконструкції оригінальних даних, що забезпечує прогнозування значень тривалості лікування цукрового діабету у дітей на основі інших ознак (рис. 3.11).

При цьому використовується простий автоенкодер з одним прихованим шаром (Dense шар з 32 нейронами), який використовує функцію активації ReLU на входному шарі та лінійну функцію активації на вихідному шарі. Цільовою функцією є середньоквадратична помилка (mean_squared_error), а оптимізатор - Adam.

Модель тренується протягом 50 епох, використовуючи збірку (batch) розміром 32. Зміни втрат відображаються на графіку для тренування та валідації протягом епох.

Здійснюється оцінка реконструкції на тестових даних, порівняння отриманих значень з оригінальними.

Отриманий графік втрат на тренуванні та валідації виводиться для візуальної оцінки процесу навчання (рис. 3.12).

Автоенкодер (Autoencoder):

Використання автоенкодера для навчання представлення даних Використання вивчених представлень для прогнозування Функція активації: ReLU або Sigmoid Оптимізатор: Adam або SGD

```
In [50]: import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error
from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Оригінальні дані
X = train_clean.drop(columns=['Bed_days'])
y = train_clean['Bed_days']

# Нормалізація даних
scaler = StandardScaler()
X_normalized = pd.DataFrame(scaler.fit_transform(X), columns=X.columns)

# Розділення на навчальні та тестові дані
X_train, X_test, y_train, y_test = train_test_split(X_normalized, y, test_size=0.2, random_state=42)

# Створення та навчання автоенкодера
model_autoencoder = Sequential()
model_autoencoder.add(Dense(32, input_dim=X_train.shape[1], activation='relu'))
model_autoencoder.add(Dense(X_train.shape[1], activation='linear'))
model_autoencoder.compile(loss='mean_squared_error', optimizer='adam')
history_autoencoder = model_autoencoder.fit(X_train, X_train, epochs=50, batch_size=32, validation_data=(X_test, X_test), verbose=1)

# Оцінка автоенкодера на тестових даних
X_pred_autoencoder = model_autoencoder.predict(X_test)

# Виведення графіку з історією навчання
plt.plot(history_autoencoder.history['loss'], label='Training Loss')
plt.plot(history_autoencoder.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss of Autoencoder')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

# Розрахунок метрик
mse_autoencoder = mean_squared_error(X_test, X_pred_autoencoder)
mae_autoencoder = mean_absolute_error(X_test, X_pred_autoencoder)

# Виведення тексту з результатами
print(f'Mean Squared Error (Autoencoder): {mse_autoencoder:.3f}')
print(f'Mean Absolute Error (Autoencoder): {mae_autoencoder:.3f}')
```

Рисунок 3.11 – Код навчання автоенкодера (Autoencoder)

Встановлено, що перші декілька епох показують значні зниження втрат як на тренуванні, так і на валідації. Тривалість навчання становить 50 епох, і здається, що процес навчання є стабільним та збіжним.

Початкові втрати на тренуванні та валідації становлять 0.4752 та 0.4017 відповідно. Після 50 епох вони значно зменшилися і становлять 0.0011 та 0.0011 відповідно. Зниження втрат свідчить про те, що автоенкодер успішно навчився реконструювати вхідні дані.

Валідаційні втрати швидко зменшуються, і процес здається стабільним. Втрати на валідації значно менші за втрати на тренуванні, що є позитивним сигналом і свідчить про здатність моделі узагальнювати для нових даних.

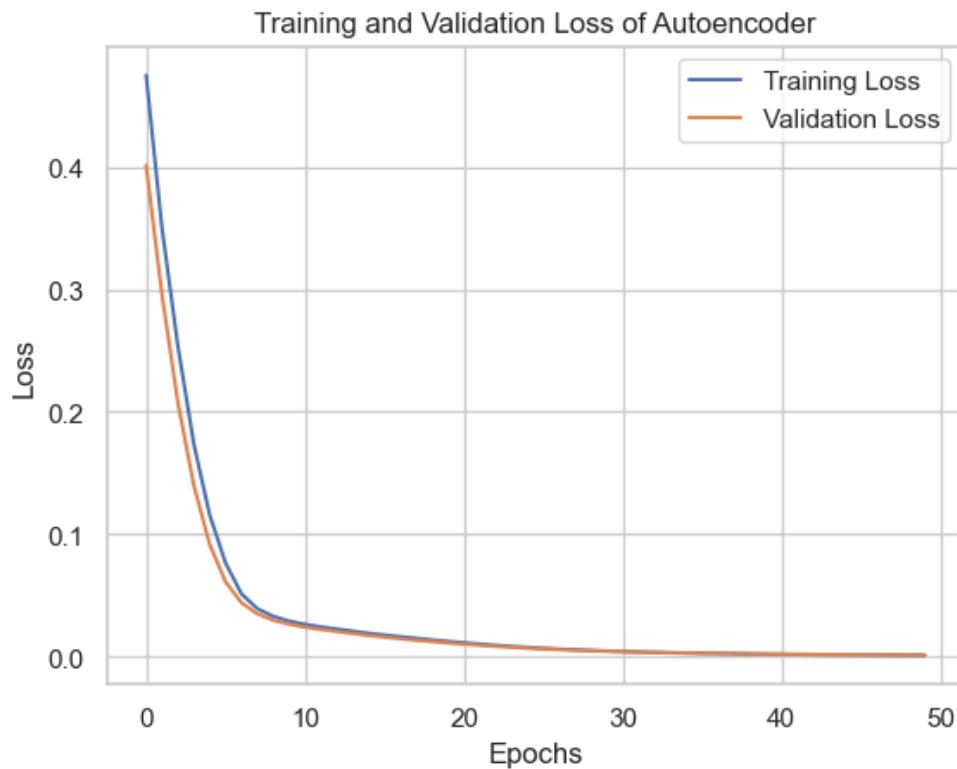


Рисунок 3.12 – Результати навчання автоенкодера (Autoencoder)

Загальна ефективність автоенкодера може бути визначена оцінкою на тестових даних та порівнянням реконструйованих даних із вхідними. Після тренування можна використовувати модель для реконструкції та генерації нових даних. Для покращення можна експериментувати із збільшенням розмірності прихованого шару, збільшенням кількості епох або зміною архітектури моделі. Загалом, виглядає, що автоенкодер успішно навчився реконструювати вхідні дані, і втрати на валідації підтверджують його здатність узагальнення.

3.3. Результати вибору раціональної нейромережевої моделі для прогнозування тривалості лікування цукрового діабету у дітей

Вибір раціональної нейромережевої моделі для прогнозування тривалості лікування цукрового діабету у дітей проводився з-поміж вище описаних та навчених моделей SNN, RNN і AE. Кожна із них має свої особливості. Якщо використано послідовні дані та важлива часова залежність, RNN може бути ефективним вибором. Якщо мета вилучення ознак чи зменшення розмірності даних, AE може бути корисним. SNN може використовуватися для стандартних задач класифікації та регресії.

Для порівняння отриманих моделей нами представлено показники, які характеризують результати навчання зазначених моделей (табл. 3.1).

Таблиця 3.1 – Результати визначення показників точності навчених моделей SNN, RNN і AE для прогнозування тривалості лікування цукрового діабету у дітей

Модель	Показники точності		
	Середня квадратична похибка (MSE)	Середня абсолютна похибка (MAE)	Середня тривалість, t , мс/епоху
Проста штучна нейронна мережа (Simple Neural Network – SNN)	30.215	4.394	5
Рекурентна нейронна мережа (Recurrent Neural Network – RNN)	30.539	4.408	4
Автоенкодер (Autoencoder – AE)	0.001	0.025	5

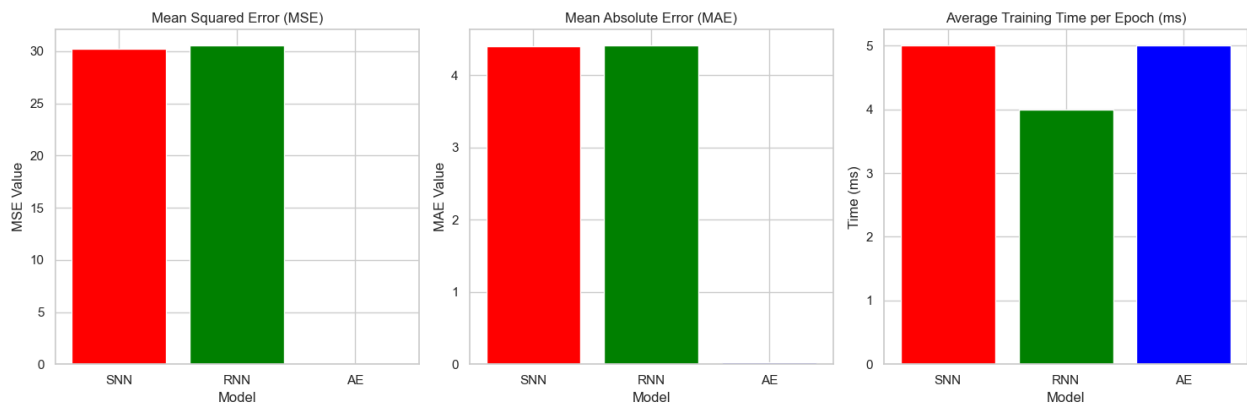


Рисунок 3.13 – Гістограми зміни показників точності моделей SNN, RNN і AE

Встановлено, що найнижча середня квадратична похибка (MSE) у моделі AE (0.001), що свідчить про високу точність прогнозування цієї моделі. SNN та RNN мають вищі значення MSE відповідно 30.215 та 30.539. Середня абсолютна похибка (MAE) найнижча за використання AE і становить 0.025, що є дуже точним результатом. SNN та RNN мають значення MAE відповідно 4.394 та 4.408, але вищі, ніж у моделі AE.

Час, витрачений на одну епоху навчання, вищий для SNN (5 мс) та AE (5 мс), ніж для RNN (4 мс). Проте, різниця у часі є невеликою, і для практичних застосувань вона може бути непомітною.

З таблиці 3.1 та рис. 3.13 видно, що найкращою моделлю за показником середньої квадратичної похибки (MSE) і середньої абсолютної похибки (MAE) є автоенкодер (AE). Його значення MSE дорівнює 0.001, а MAE – 0.025. Ці значення досить низькі, що свідчить про добру точність прогнозування моделі.

РОЗДІЛ 4.

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Аналіз шкідливих чинників під час розробки моделі

Потенційно шкідливим чинником кабінету для роботи програмістів, вважається небезпека враження його електричним струмом. Важливим, але менш ймовірним чинником являється пожежна небезпека під час аварійної ситуації. Хімічні та біологічні джерела практично не мають впливу.

Перелік шкідливих чинників кабінету для роботи програмістів подано у таблиці 4.1.

Таблиця 4.1 – Шкідливі чинники кабінету для роботи програмістів

Фізичні	Електробезпека, пожежа, шум, мікроклімат
Хімічні	Відсутні
Біологічні	Відсутні
Психофізіологічні	Відсутні

В приміщенні кабінету для роботи програмістів, присутні небезпечні чинники, та за умов дотримання заходів безпеки, вони не є критичним.

4.2. Розробка заходів щодо покращення умов праці програмістів

Програмісти є однією з найважливіших професій у сучасному світі. Вони розробляють програмне забезпечення, яке використовується в усіх сферах життя, від бізнесу до розваг. Однак умови праці програмістів часто залишають бажати кращого.

Програмісти часто проводять багато часу, сидячи за комп'ютером, що може призводити до проблем із поставою, очей, а також інших захворювань.

Програмісти часто працюють у стресових умовах, оскільки вони відповідають за якість і надійність програмного забезпечення, яке вони розробляють. Це може призводити до перевтоми, депресії та інших проблем зі здоров'ям.

Програмісти часто працюють наодинці, що може призводити до відчуття ізоляції та самотності.

Забезпечення комфортних робочих умов включає правильне освітлення, зручне крісло та стіл, а також можливість робити перерви протягом робочого дня. Слід передбачити запровадження програм з охорони здоров'я та безпеки праці. Ці програми повинні включати в себе регулярні медичні огляди, навчання з безпеки праці та заходи щодо запобігання стресу.

Створення сприятливого психологічного клімату в колективі є запорукою здоров'я та безпеки програмістів. Це включає в себе заохочення соціальної взаємодії між програмістами, проведення корпоративних заходів та створення можливості для професійного розвитку.

Встановлення в робочих приміщеннях програмістів зручних стільців і столів, а також обладнання для фізичних вправ. Заохочення соціальної взаємодії між програмістами, наприклад, через проведення корпоративних заходів, таких як командні ігри, тренінги та вечірки.

Запровадження заходів щодо покращення умов праці програмістів є важливим для забезпечення їхнього здоров'я, безпеки та продуктивності.

4.3. Розробка логічно-імітаційної моделі травматизму під час ураження електричним струмом

Для побудови логіко-імітаційної моделі процесу, формування і виникнення аварії та травми під час ураження електричним струмом складемо список базових подій. Вони лежатимуть у основі даної моделі. Кожному пункту

списку присвоюємо певне значення ймовірності виникнення. Нижче подано сам список:

- | | |
|--|-------------------|
| 1. Стан контролю з охорони праці | $P_1 = 0,2;$ |
| 2. Несерйозне відношення до проходження ТО інструменту | $P_2 = 0,1;$ |
| 3. Відсутність комплектуючих установки..... | $P_3 = 0,2;$ |
| 4. Невисока міцність | $P_4 = 0,03;$ |
| 1. Використання застарілого обладнання..... | $P_6 = 0,02;$ |
| 2. Попадання сторонніх предметів | $P_7 = 0,4;$ |
| 3. Досвід роботи виконавця | $P_{12} = 0,35.$ |
| 4. Професійний рівень виконавця | $P_{13} = 0,5;$ |
| 5. Психофізіологічний стан виконавця..... | $P_{14} = 0,083;$ |

На основі даного списку будуємо матрицю логічних взаємозв'язків між окремими пунктами, графічне представлення якої зображено на рис. 4.1.

Розрахуємо ймовірності виникнення подій, що входять у дану логіко-імітаційну модель процесу виконання робіт програмістом (на прикладі ймовірності отримання травми через ураження електричним струмом).

Ймовірність виникнення події P_5 визначаємо наступним чином:

$$P_5 = 0,2 + 0,1 + 0,2 + 0,003 - 0,2 \cdot 0,1 - 0,2 \cdot 0,03 - 0,2 \cdot 0,03 - 0,1 \cdot 0,2 - 0,1 \cdot 0,03 - 0,2 \cdot 0,03 + 0,2 \cdot 0,1 \cdot 0,2 + 0,1 \cdot 0,2 \cdot 0,03 + 0,2 \cdot 0,1 \cdot 0,2 + 0,2 \cdot 0,1 \cdot 0,03 - 0,2 \cdot 0,1 \cdot 0,2 \cdot 0,03 = 0,314$$

Ймовірність виникнення події P_{10} визначаємо так:

$$P_{10} = 0,2 + 0,1 = 0,3.$$

Ймовірність виникнення події P_{11} визначаємо:

$$P_{11} = 0,02 \cdot 0,314 \cdot 0,4 \cdot 0,3 = 0,00075.$$

Ймовірність виникнення події P_{15} визначаємо наступним чином:

$$P_{15} = 0,35 \cdot 0,5 \cdot 0,083 = 0,0145.$$

Ймовірність події P_{18} :

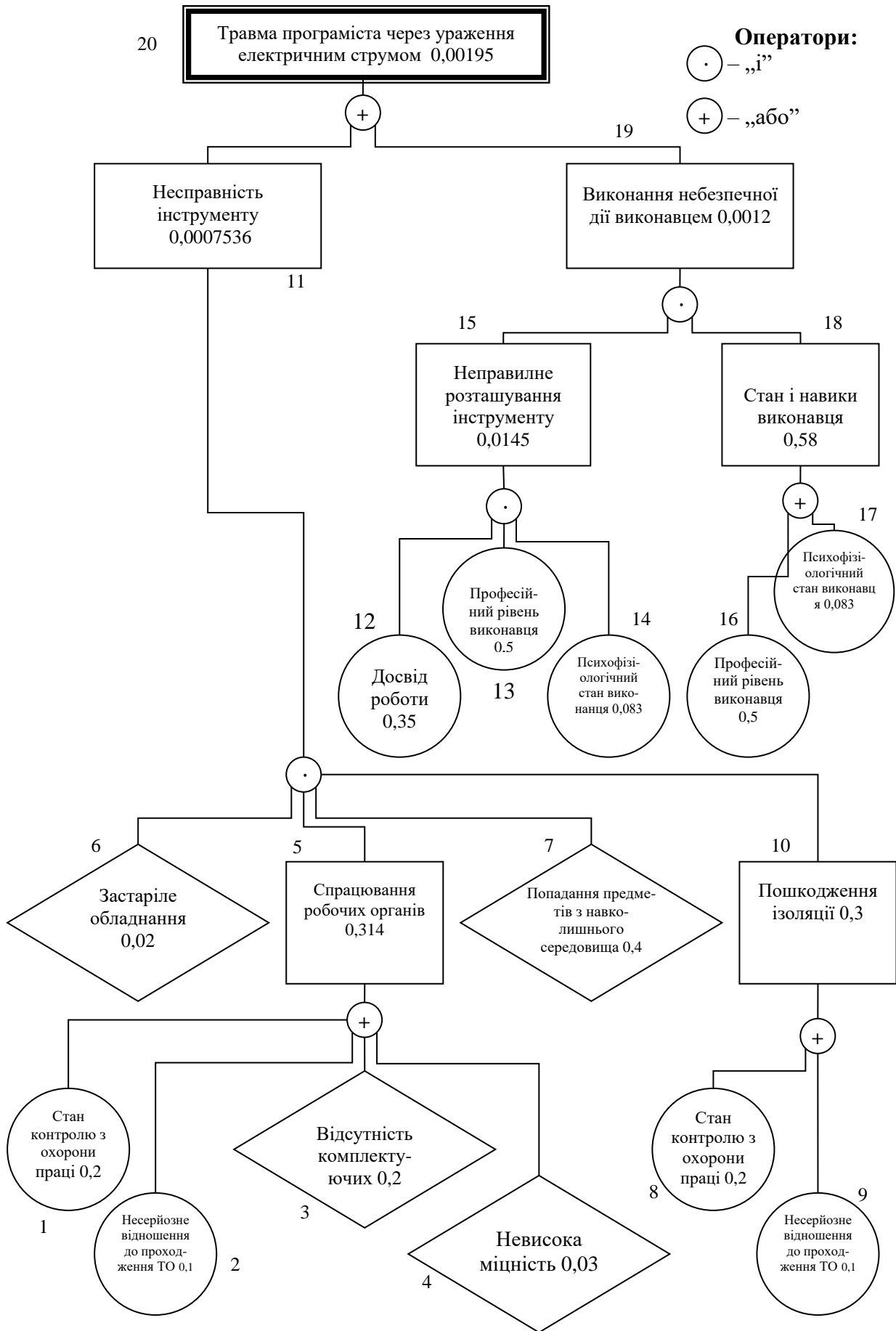


Рис. 4.1. Логіко-імітаційна модель процесу формування та виникнення аварії та травми під час ураження електричним струмом програміста

$$P_{18} = 0,5 + 0,083 = 0,58.$$

Ймовірність події P_{19} :

$$P_{19} = 0,0145 \cdot 0,083 = 0,0012.$$

Ймовірність події P_{20} :

$$P_{20} = 0,00075 + 0,0012 = 0,00195.$$

Ймовірність травми рівна ймовірності виникнення аварії, бо остання можлива лише за умови ураження електричним струмом програміста.

Після аналізу результатів моделювання ймовірність виникнення травми можна звести до дуже малої величини – достатньо зменшити вплив ймовірностей вихідних факторів, які до неї призводять.

4.4. Розробка заходів щодо безпеки у надзвичайних ситуаціях

Незалежно від того, чи перебуваєте в побуті або працюєте, важливо бути готовим до надзвичайних ситуацій. Це може бути природне лихо, техногенна катастрофа або терористичний акт.

Щоб бути готовим до надзвичайних ситуацій, важливо розробити план дій. Цей план повинен включати в себе такі елементи:

- Оцінка ризиків – першим кроком у розробці плану дій є оцінка ризиків, які можуть виникнути у вашому районі. Це допоможе вам визначити, які заходи безпеки слід вжити.
- План евакуації – у разі надзвичайної ситуації вам може знадобитися евакуюватися з вашого будинку або робочого місця. План евакуації повинен включати в себе маршрути евакуації, місця зустрічі та контактну інформацію.
- План дій на випадок надзвичайної ситуації – цей план повинен включати в себе конкретні дії, які ви та ваша сім'я будете виконувати в разі надзвичайної ситуації.

➤ План відновлення – цей план повинен включати в себе дії, які ви будете виконувати після надзвичайної ситуації, щоб відновити своє життя.

Слід забезпечити свій будинок або робоче місце засобами першої допомоги та аварійним освітленням. Зберігати запаси їжі, води та інших необхідних товарів на випадок надзвичайної ситуації. Ознайомитися з маршрутами евакуації та місцями зустрічі у вашому районі.

Важливо регулярно переглядати свій план дій та вносити в нього зміни, якщо це необхідно. Також повинні навчати своїх дітей та членів сім'ї діям у разі надзвичайної ситуації. Слід бути інформованими про поточні погодні умови та інші можливі загрози. Слідкувати за новинами та офіційними повідомленнями про надзвичайні ситуації.

У надзвичайних ситуаціях важливо зберігати спокій і тверезо мислити. Виконання цих заходів допоможе бути більш захищеними у разі надзвичайної ситуації.

РОЗДІЛ 5.

ВИЗНАЧЕННЯ ПОКАЗНИКІВ ЕФЕКТИВНОСТІ ВІД ПРОГНОЗУВАННЯ ТРИВАЛОСТІ ЛІКУВАННЯ ЦУКРОВОГО ДІАБЕТУ У ДІТЕЙ ІЗ ВИКОРИСТАННЯМ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ

Економічна ефективність від використання нейромережових моделей для прогнозування тривалості лікування цукрового діабету у дітей може бути визначена за допомогою різних показників. Оцінка економічної ефективності дозволяє визначити, наскільки вдалим є використання моделі з економічної точки зору.

Розглянемо показники економічної ефективності. Вартість прогнозування (Forecasting Cost) визначається за формулою:

$$FC = \sum(C_i), \quad (5.1)$$

де C_i - вартість прогнозування для кожного випадку, грн.

Сумарна вартість прогнозування для всіх випадків може бути визначена завдяки збільшенню ефективності лікування (Treatment Efficiency Gain) за формулою:

$$TEG = 1 - (TC_forecast / TC_baseline), \quad (5.2)$$

де TEG – зростання ефективності лікування (Treatment Effectiveness Gain) , грн; TC_baseline – вартість лікування без використання прогнозування (Treatment Cost without forecasting), грн; TC_forecast – вартість лікування з використанням прогнозування (Treatment Cost with forecasting), грн.

Також розраховується відносне зменшення вартості лікування завдяки використанню прогнозів. Зменшення тривалості лікування (Reduction in Treatment Duration) визначається за формулою:

$$RTD = 1 - (TD_forecast / TD_baseline), \quad (5.3)$$

де $TD_forecast$ – тривалість лікування з використанням прогнозів, днів; $TD_baseline$ – тривалість лікування без прогнозування, днів.

Припустимо, що маємо наступні показники економічної ефективності:

FC = 5000 грн. – припустима сумарна вартість прогнозування, грн;

TC_forecast = 30000 грн. – вартість лікування з використанням прогнозів, грн;

TC_baseline = 35000 грн. – вартість лікування без прогнозування, грн;

TD_forecast = 25 діб – тривалість лікування з використанням прогнозів, діб;

TD_baseline = 30 діб – тривалість лікування без прогнозування, діб.

Підставивши відповідні значення у формули (5.2) та (5.3) виконаємо розрахунок показників економічної ефективності.

$$TEG = 1 - (30000 / 35000) = 14.29\% .$$

$$RTD = 1 - (25 / 30) = 16.67\% .$$

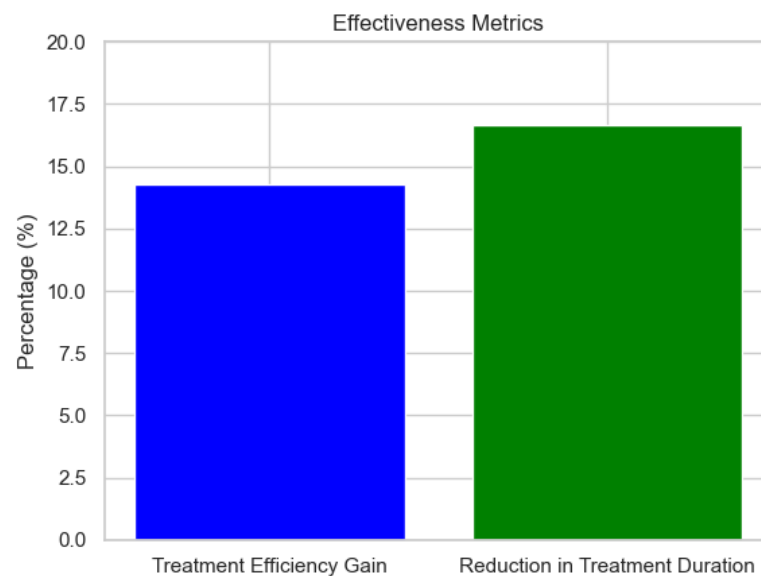


Рисунок 5.1 – Результати визначення показників економічної ефективності від використання нейромережових моделей для прогнозування тривалості лікування цукрового діабету у дітей

Отримані результати вказують на те, що застосування нейромережових моделей для прогнозування тривалості лікування цукрового діабету у дітей призвело до позитивних ефектів у сфері економічної ефективності.

Приріст ефективності лікування (TEG) становить 14,29%. Збільшення ефективності цього показника свідчить про те, що прогнозування тривалості

лікування за допомогою нейромереж допомагає оптимізувати процес та покращити результати. Скорочення тривалості лікування становить 16,67%. Зменшення на 16.67% означає, що використання нейромережевих моделей дозволило скоротити час лікування, що може мати позитивний вплив на ресурси та вартість проведення лікування. Такі результати свідчать про переваги та ефективність використання нейромережевих моделей у прогнозуванні тривалості лікування цукрового діабету у дітей з економічної точки зору.

ВИСНОВКИ І ПРОПОЗИЦІЇ

Ефективне прогнозування тривалості лікування є ключовим елементом управління цією хворобою. Низка науковців вказують на важливість використання обчислювального інтелекту для точніших та індивідуалізованих прогнозів. На підставі аналізу існуючих підходів досліджень та в галузі застосування обчислювального інтелекту в медицині проведено широкий огляд літератури, що вказує на розмаїття методів та їхню ефективність у медичних дослідженнях. Цей аналіз став основою для визначення перспективних напрямків застосування обчислювального інтелекту в прогнозуванні тривалості лікування цукрового діабету.

Виконаний аналіз стану науки із прогнозування цукрового діабету за допомогою машинного навчання, свідчить про те, що розглядаються конкретні методи машинного навчання для аналізу клінічних даних. Встановлено вплив цих методів на точність та достовірність прогнозів тривалості лікування.

Нами сформульовано завдання кваліфікаційної роботи. Результати розробки моделей із використанням нейромережевих моделей можуть бути використані як основа для подальших досліджень та впровадження інноваційних підходів у практику лікування цукрового діабету у дітей.

Для задач прогнозування тривалості лікування цукрового діабету у дітей можуть бути використані такі методи обчислювального інтелекту, які представлені на рис. 2.1. Методи обчислювального інтелекту можуть бути використані для розробки індивідуальних планів лікування цукрового діабету у дітей. Це дозволить поліпшити якість життя дітей з цукровим діабетом та знизити ризик розвитку ускладнень захворювання.

Нами здійснено вибір методів обчислювального інтелекту для прогнозування тривалості лікування цукрового діабету у дітей. Для цього на рис. 2.2 представлено взаємозв'язок між AI, ML та DL. Слід зазначити, що машинне навчання та глибоке навчання є підгалуззями штучного інтелекту. Для наших досліджень, які стосуються використання методів обчислювального

інтелекту для прогнозування тривалості лікування цукрового діабету у дітей та вибір методів обчислювального інтелекту, вибрано наступні 3 видів нейромережових моделей: 1) Проста штучна нейронна мережа (Simple Neural Network – SNN); 2) Рекурентна нейронна мережа (Recurrent Neural Network – RNN); 3) Автоенкодер (Autoencoder – AE).

Модель багат шарового перцептрона (MLP) – це тип штучної нейронної мережі (ANN) зі зворотним зв'язком, яка слугує базовою архітектурою для глибокого навчання або глибоких нейронних мереж (DNN). Вона працює як підхід до навчання під наглядом. MLP складається з трьох шарів: вхідного шару, вихідного шару та одного або декількох прихованих шарів. Це повністю пов'язана мережа, тобто кожен нейрон в одному шарі пов'язаний з усіма нейронами в наступному шарі.

Рекурентні нейронні мережі (RNN) – це клас моделей глибокого навчання, які мають внутрішню пам'ять, що дозволяє їм фіксувати послідовні залежності. На відміну від традиційних нейронних мереж, які розглядають вхідні дані як незалежні об'єкти, RNN враховують часовий порядок вхідних даних, що робить їх придатними для завдань, пов'язаних з послідовною інформацією. Використовуючи цикл, ANN застосовують одну і ту ж операцію до кожного елемента в серії, при цьому поточні обчислення залежать як від поточного входу, так і від попередніх обчислень.

Концепція автокодера виникла як нейронна мережа, призначена для реконструкції вхідних даних. Її основна мета полягає в тому, щоб навчитися осмисленому представленню даних у неконтрольований спосіб, що може мати різні застосування, включаючи кластеризацію. Автокодер – це нейронна мережа, яка прагне відтворити вхідні дані на виході. Вона складається з внутрішнього прихованого шару, який визначає код, що представляє вхідні дані.

Для нашого дослідження ми збрали дані однієї із лікарень (ендокринологічного відділення клініки «Оксфорд Медікал») м. Львів. Вони стосувалися лікування цукрового діабету 778 дітей віком до 19 років за 9

атрибути. Визначено набір атрибутів, що включають дату госпіталізації, дату виписки, відділення, дату народження, температуру, зріст, стать, вагу, час перебування в лікарні та інші атрибути, які можуть впливати на тривалість лікування. Дані отримані із лікарні та містять інформацію про пацієнтів і їхнє лікування.

Після очищення та агрегації даних ми отримали 481 одиницю даних. Дані були введені в Jupyter – середовище інтерактивної розробки для блокнотів, коду та даних на мові Python. Моделі побудовано із використанням бібліотеки Keras із її модулями Sequential та Dense.

Нами використано бібліотеку Pandas для завантаження даних з файлу Excel в об'єкт DataFrame та виведення перших кілька рядків цього DataFrame (рис. 3.1). Після цього нами проаналізовано унікальні значення та їх кількостей у кожному стовпці. Для прикладу покажемо стовбець «Human_gender» нашого DataFrame (рис. 3.2).

Наступний код (3.3) виконує заміну категоріальних значень на числові у вказаних стовпцях DataFrame, а потім виводить оновлений DataFrame та текст, що показує, які категорії були замінені на які числа. Результати перевірки на пропуски (рис. 3.4) у показує, що в жодному з вказаних стовпців («Temperature», «Height», «Weight», «Department», «Human_gender», «In_hospital», «Type_settlement», «Age») немає пропущених значень.

Нами здійснено побудову гістограми віку пацієнтів, які лікувалися від цукрового діабету (рис. 3.5). Цей графік дозволяє швидко оцінити, які вікові групи представлені у ваших даних та яка кількість спостережень припадає на кожну з них. Встановлено, що найбільше у віці від 5 до 13 років.

Також нами побудовано графік розподілу віку пацієнтів (стовпець "Age") з використанням гістограми та врахування тривалості лікування (стовпець "Bed_days") як відтінку кольору (використовуючи параметр hue) (рис. 3.6). Встановлено, що найдовша тривалість лікування припадає на вік 7-8 років та 13-14 років. Лінія середнього віку (червона лінія) вказує на середній вік

пацієнтів (9,5 років), що може бути важливою метрикою для загального розуміння даних.

Нами використано бібліотеки для машинного навчання, такі як Keras та scikit-learn, для навчання моделі з використанням Long Short-Term Memory (LSTM) мережі на задачі прогнозування тривалості лікування (рис. 3.7). Оцінюючи результати навчання моделі за кожною із 50 епох, ми можемо зробити наступні спостереження. Це свідчить про те, що модель поступово навчається і адаптується до тренувальних даних. Показано графік з історією навчання (втратами на тренувальних та валідаційних даних) (рис. 3.8).

Нами використано бібліотеку Keras для створення та навчання простої рекурентної нейронної мережі (RNN) для прогнозування значень тривалості лікування цукрового діабету у дітей на основі інших ознак (рис. 3.9). Вхідні дані нормалізуються за допомогою StandardScaler, щоб забезпечити стандартизацію і покращення навчання моделі. Дані розбиваються на тренувальний та тестовий набори зі співвідношенням 80/20. Нами побудовано графік втрат на тренувальному та тестовому наборах даних, що забезпечує візуалізацію ефективності моделі під час навчання (рис. 3.10).

Нами використано бібліотеку Keras для створення автоенкодера, що забезпечує навчання моделі без нагляду (unsupervised learning) та використовує його для реконструкції оригінальних даних, що забезпечує прогнозування значень тривалості лікування цукрового діабету у дітей на основі інших ознак (рис. 3.11). Початкові втрати на тренуванні та валідації становлять 0.4752 та 0.4017 відповідно. Після 50 епох вони значно зменшилися і становлять 0.0011 та 0.0011 відповідно. Зниження втрат свідчить про те, що автоенкодер успішно навчився реконструювати вхідні дані.

Нами представлено результати визначення показників точності навчених моделей SNN, RNN і AE для прогнозування тривалості лікування цукрового діабету у дітей. З таблиці 3.1 та рис. 3.13 видно, що найкращою моделлю за показником середньої квадратичної похибки (MSE) і середньої абсолютної похибки (MAE) є автоенкодер (AE). Його значення MSE дорівнює 0.001, а MAE

– 0.025. Ці значення досить низькі, що свідчить про добру точність прогнозування моделі.

Розроблено рекомендації щодо охорони праці дають можливість створити безпечні умови праці для розробників нейромережових моделей.

На підставі проведених розрахунків показників економічної ефективності від використання нейромережової моделі встановлено, що приріст ефективності лікування (TEG) становить 14,29% та скорочення тривалості лікування становить 16,67%. Такі результати свідчать про переваги та ефективність використання нейромережових моделей у прогнозуванні тривалості лікування цукрового діабету у дітей з економічної точки зору.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Введення в машинне навчання за допомогою Python и Scikit-Learn. URL: <https://habr.com/ua/company/mlclass/blog/247751/> (дата звернення: 26.08.2024).
2. Жидецький В.Ц., Джигирей В.С., Мельников О.В. Основи охорони праці. Підручник. Вид. 5-е, доповнене. Львів: Афіша, 2012. 350с.
3. Зайченко Ю.П. Дослідження операцій: підручник. Сьоме видання, перероблене та доповнене. К.: Видавничий Дім «Слово», 2006. 816 с.
4. Класифікація в Python з Scikit-Learn та Pandas. URL: <https://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/> (дата звернення: 21.07.2024).
5. Лехман С.Д., Рублев В.І., Рябцев Б.І. Запобігання аварійності і травматизму у сільському господарстві. К.: Урожай, 1993. 267 с.
6. Марценюк В. П., Семенець А. В. Медична інформатика. Інструментальні та експертні системи. Тернопіль: Укрмедкнига, 2004. 222 с.
7. Огляд методів класифікації у машинному навчанні за допомогою Scikit-Learn. URL: <https://tproger.ru/translations/scikit-learn-in-python/s://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/> (дата звернення: 15.09.2024).
8. Павлиш В. А. ., Гліненко Л. К, Шаховська Н. Б. Основи інформаційних технологій і систем: підручник. Львів: Львівська політехніка, 2018. 620 с.
9. Продеус А.М., Синєкоп Ю.С., Швець Є.Я., Кісельов Є.М., Баран М.М. Експертні системи в медицині: навч. посіб. Запоріжжя: Вид. ЗДІА, 2014. 332 с.
10. Ситнік Б.Т. Основи інформаційних систем і технологій: навч. посіб. Харків: УкрДУЗТ, 2018. 130 с.
11. Tryhuba, A., Boyarchuk, V., Tryhuba, I., Ftoma, O., Padyuka, R., Rudynets, M. Forecasting the Risk of the Resource Demand for Dairy Farms Basing

on Machine Learning. Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT+DS 2020). 2020; I, 327-340.

12. Цукровий діабет 2-го типу у дітей та підлітків. URL: <https://doctorthinking.org/2021/08/type-2-diabetes-child/>

13. Ahmed, U.; Issa, G.F.; Khan, M.A.; Aftab, S.; Khan, M.F.; Said, R.A.T.; Ghazal, T.M.; Ahmad, M. Prediction of diabetes empowered with fused machine learning. *IEEE Access* 2022, 10, 8529–8538.

14. Alpaydin E. Introduction to Machine Learning. MIT Press (2020)

15. American Diabetes Association. Management of type 1 diabetes in children and adolescents. *Diabetes Care*, 2022, 45(1). P.18-40.

16. Barakat, N.; Bradley, A.P.; Barakat, M.N.H. Intelligible support vector machines for diagnosis of diabetes mellitus. *IEEE Trans. Inf. Technol. Biomed.* 2010, 14, 1114–1120.

17. Barnett AH, Craig ME, Daneman D, et al. International consensus on the management of type 1 diabetes in children and adolescents. *Pediatr Diabetes*, 2018, 19(Suppl 2). P. 1-146.

18. Bashynsky, Coordination of dairy workshops projects on the community territory and their project environment”. International Scientific and Technical Conference on Computer Sciences and Information Technologies. <https://www.scopus.com/authid/detail.uri?authorId=57205218805>. 2019; 3: 51–54.

19. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* 2002, 16, 321–357.

20. Chen M., Mao S., Liu Y. Big data: A survey *Mob. Netw. Appl.*, 19 (2), (2014), pp. 171-209.

21. Chen M., Hao Y., Hwang K., Wang L., Wang L. Disease prediction by machine learning over big data from healthcare communities. *Ieee Access*, 5 (2017), pp. 8869-8879.

22. Dallora A.L., Eivazzadeh S., Mendes E., Berglund J., Anderberg P. Machine learning and microsimulation techniques on the prognosis of dementia: A systematic literature review. *PLoS One*, 12 (6) (2017), Article e0179804
23. Dataset. Available online: <https://www.kaggle.com/datasets/andrewmvd/early-diabetes-classification> (accessed on 17 November 2023).
24. Fazakis, N.; Kocsis, O.; Dritsas, E.; Alexiou, S.; Fakotakis, N.; Moustakas, K. Machine learning tools for long-term type 2 diabetes risk prediction. *IEEE Access* 2021, 9, 103737–103757.
25. Groves P., Kayyali B., Knott D., Kuiken S.V. The big Data revolution in Healthcare: Accelerating Value and Innovation. Center for US Health System Reform Business Technology Office. (2016).
26. Hasan, M.K.; Alam, M.A.; Das, D.; Hossain, E.; Hasan, M. Diabetes prediction using ensembling of different machine learning classifiers. *IEEE Access* 2020, 8, 76516–76531.
27. Islam, M.M.F.; Ferdousi, R.; Rahman, S.; Bushra, H.Y. Likelihood prediction of diabetes at early stage using data mining techniques. *Computer Vision and Machine Intelligence in Medical Image Analysis*; Springer: Singapore, 2020; pp. 113–125.
28. Islam, M.S.; Qaraqe, M.K.; Belhaouari, S.B.; Abdul-Ghani, M.A. Advanced techniques for predicting the future progression of type 2 diabetes. *IEEE Access* 2020, 8, 120537–120547.
29. Khanam; Jamal, J.; Foo, S.Y. A comparison of machine learning algorithms for diabetes prediction. *ICT Express* 2021, 7, 432–439.
30. Koval N., Tryhuba A., Kondysiuk I., Tryhuba I., Boiarchuk O., Rudynets M., Grabovets V., Onyshchuk V. Forecasting the Fund of Time for Performance of Works in Hybrid Projects Using Machine Training Technologies. *Proceedings of the 3rd International Workshop on Modern Machine Learning Technologies and Data Science Workshop*. Proc. 3rd International Workshop (MoMLLeT&DS 2021). 2021; I, 96-206.

31. Krishnamoorthi, R.; Joshi, S.; Almarzouki, H.Z.; Shukla, P.K.; Rizwan, A.; Kalpana, C.; Tiwari, B. A novel diabetes healthcare disease prediction framework using machine learning techniques. *J. Healthc. Eng.* 2022, 2022, 1684017.
32. Le T.-L. Fuzzy C-means clustering interval type-2 cerebellar model articulation neural network for medical data classification. *IEEE Access*, 7 (2019), pp. 20967-20973.
33. Lee C., Yoon H. Medical big data: promise and challenges. *Kidney Res. Clin. Pract.*, 36 (1) (2017), pp. 3-11.
34. Lillioja S, Moller DE, Sharp PS, et al. Genetic variation in insulin secretion and action predicts type 1 diabetes remission. *Nat Med*, 2008, 14(1). P. 106-111.
35. Malanchuk, O., Tryhuba, A., Tryhuba, I., Bandura, I. A conceptual model of adaptive value management of project portfolios of creation of hospital districts in Ukraine. *CEUR Workshop Proceedings*. 2023; 3453, 82–95.
36. Maniruzzaman, M.; Rahman, M.J.; Ahammed, B.; Abedin, M.M. Classification and prediction of diabetes disease using machine learning paradigm. *Health Inf. Sci. Syst.* 2020, 8, 7.
37. Michie D., Spiegelhalter D.J., Taylor C., *et al.* Machine learning. *Neural Stat. Classif.*, 13 (1994) (1994), pp. 1-298.
38. Mitchell T.M. *Machine Learning*. McGraw-Hill, New York (1997)ю
39. Obermeyer Z., Emanuel E.J. Predicting the future—big data, machine learning, and clinical medicine. *New England J. Med.*, 375 (13) (2016), p. 1216.
40. Rajkomar A., Dean J., Kohane I. Machine learning in medicine. *New Engl. J. Med.*, 380 (14) (2019), pp. 1347-1358.
41. Russell S., Norvig P. *Artificial intelligence: a modern approach*. (2002)
42. Scruggs S.B., Watson K., Su A.I., Hermjakob H., Yates J.R., Lindsey M.L., Ping P. Harnessing the heart of big data. *Circ. Res.*, 116 (7) (2015), pp. 1115-1119.

43. Shen L., Chen H., Yu Z., Kang W., Zhang B., Li H., Yang B., Liu D. Evolving support vector machines using fruit fly optimization for medical data classification. *Knowl.-Based Syst.*, 96 (2016), pp. 61-75.
44. Tryhuba, A., Boyarchuk, V., Tryhuba, I., Ftoma, O., Padyuka, R. & Rudynets, M. Forecasting the risk of the resource demand for dairy farms basing on machine learning. *CEUR Workshop Proceedings*. <https://www.scopus.com/authid/detail.uri?authorId=57205225539>. 2021; 2631: 327–340.
45. Tryhuba, A., Kondysiuk, I., Tryhuba, I., Boiarchuk, O., Tatomyr, A. Intellectual information system for formation of portfolio projects of motor transport enterprises. *CEUR Workshop Proceedings*. 2022; 3109, 44–52.
46. Tryhuba, A., Tryhuba, I., Ftoma, O. & Boyarchuk, O. Method of quantitative evaluation of the risk of benefits for investors of fodder-producing cooperatives. *International Scientific and Technical Conference on Computer Sciences and Information Technologies*, <https://www.scopus.com/authid/detail.uri?authorId=57205225539>. 2019; 3: 55–58.
47. Tryhuba, A., Koval, N., Tryhuba, I. & Boiarchuk, O. Application of sarima models in information systems forecasting seasonal volumes of food raw materials of procurement on the territory of communities. *CEUR Workshop Proceedings*. 2022; 3295: 64–75. <https://www.scopus.com/authid/detail.uri?authorId=57205225539>.
48. Tryhuba, A., Malanchuk, O., Tryhuba, I. Prediction of the Duration of Inpatient Treatment of Diabetes in Children Based on Neural Networks. *CEUR Workshop Proceedings*. 2023; 3426, 122–135.
49. Tryhuba, A., Vovk, M., Batyuk, B., Holomsha, O., Sava, A. Improving The Quality Of Management In The System Of Forecasting Milk Procurement In Communities Usage The Technology Of Neutron Networks. *Journal of Hygienic Engineering and Design*. 2022; 40, 201–209.

50. Ben-Bassat A., Barzilai N. E., Cohen E., Elbaz H. Y., Goldshtein A., and Shoham Y. Predicting the duration of type 1 diabetes in children using Machine Learning. *Journal of Diabetes Research* 2023, 20230104.

ДОДАТКИ

Додаток А.1

Код простої штучної нейронної мережі (Simple Neural Network – SNN)

```

import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error
from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
# Оригінальні дані
X = train_clean.drop(columns=['Bed_days'])
y = train_clean['Bed_days']
# Нормалізація даних
scaler = StandardScaler()
X_normalized = pd.DataFrame(scaler.fit_transform(X), columns=X.columns)
# Розділення на навчальні та тестові дані
X_train, X_test, y_train, y_test = train_test_split(X_normalized, y, test_size=0.2, random_state=42)
# Перетворення даних для LSTM
X_train_lstm = X_train.values.reshape((X_train.shape[0], X_train.shape[1], 1))
X_test_lstm = X_test.values.reshape((X_test.shape[0], X_test.shape[1], 1))
# Створення та навчання LSTM моделі
model_rnn = Sequential()
model_rnn.add(LSTM(64, input_shape=(X_train_lstm.shape[1], 1), activation='tanh'))
model_rnn.add(Dense(1, activation='linear'))
model_rnn.compile(loss='mean_squared_error', optimizer=Adam())
history = model_rnn.fit(X_train_lstm, y_train, epochs=50, batch_size=32,
validation_data=(X_test_lstm, y_test), verbose=1)
# Оцінка моделі на тестових даних
y_pred = model_rnn.predict(X_test_lstm)
# Виведення графіку з історією навчання
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title("Training and Validation Loss")
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
# Розрахунок метрик
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
# Виведення тексту з результатами
print(f'Mean Squared Error: {mse:.3f}')
print(f'Mean Absolute Error: {mae:.3f}')

```

Додаток А.2

Код рекурентної нейронної мережі (Recurrent Neural Network – RNN)

```

import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error
from keras.models import Sequential
from keras.layers import SimpleRNN, Dense
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
# Оригінальні дані
X = train_clean.drop(columns=['Bed_days'])
y = train_clean['Bed_days']
# Нормалізація даних
scaler = StandardScaler()
X_normalized = pd.DataFrame(scaler.fit_transform(X), columns=X.columns)
# Розділення на навчальні та тестові дані
X_train, X_test, y_train, y_test = train_test_split(X_normalized, y, test_size=0.2, random_state=42)
# Перетворення даних для RNN
X_train_rnn = X_train.values.reshape((X_train.shape[0], 1, X_train.shape[1]))
X_test_rnn = X_test.values.reshape((X_test.shape[0], 1, X_test.shape[1]))
# Створення та навчання RNN моделі
model_rnn = Sequential()
model_rnn.add(SimpleRNN(64, input_shape=(X_train_rnn.shape[1], X_train_rnn.shape[2]),
activation='tanh'))
model_rnn.add(Dense(1, activation='linear'))
model_rnn.compile(loss='mean_squared_error', optimizer=Adam())
history = model_rnn.fit(X_train_rnn, y_train, epochs=50, batch_size=32,
validation_data=(X_test_rnn, y_test), verbose=1)
# Оцінка моделі на тестових даних
y_pred_rnn = model_rnn.predict(X_test_rnn)
# Виведення графіку з історією навчання
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
# Розрахунок метрик
mse_rnn = mean_squared_error(y_test, y_pred_rnn)
mae_rnn = mean_absolute_error(y_test, y_pred_rnn)
# Виведення тексту з результатами
print(f'Mean Squared Error (RNN): {mse_rnn:.3f}')
print(f'Mean Absolute Error (RNN): {mae_rnn:.3f}')

```

Додаток А.3

Код автоенкодера (Autoencoder – AE)

```

import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error
from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
# Оригінальні дані
X = train_clean.drop(columns=['Bed_days'])
y = train_clean['Bed_days']
# Нормалізація даних
scaler = StandardScaler()
X_normalized = pd.DataFrame(scaler.fit_transform(X), columns=X.columns)
# Розділення на навчальні та тестові дані
X_train, X_test, y_train, y_test = train_test_split(X_normalized, y, test_size=0.2,
random_state=42)
# Створення та навчання автоенкодера
model_autoencoder = Sequential()
model_autoencoder.add(Dense(32, input_dim=X_train.shape[1], activation='relu'))
model_autoencoder.add(Dense(X_train.shape[1], activation='linear'))
model_autoencoder.compile(loss='mean_squared_error', optimizer='adam')
history_autoencoder = model_autoencoder.fit(X_train, X_train, epochs=50,
batch_size=32, validation_data=(X_test, X_test), verbose=1)
# Оцінка автоенкодера на тестових даних
X_pred_autoencoder = model_autoencoder.predict(X_test)
# Виведення графіку з історією навчання
plt.plot(history_autoencoder.history['loss'], label='Training Loss')
plt.plot(history_autoencoder.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss of Autoencoder')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
# Розрахунок метрик
mse_autoencoder = mean_squared_error(X_test, X_pred_autoencoder)
mae_autoencoder = mean_absolute_error(X_test, X_pred_autoencoder)
# Виведення тексту з результатами
print(f'Mean Squared Error (Autoencoder): {mse_autoencoder:.3f}')
print(f'Mean Absolute Error (Autoencoder): {mae_autoencoder:.3f}')

```