

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

Навчально-науковий інститут прикладного системного аналізу

Кафедра системного проектування

«На правах рукопису»

УДК _____

До захисту допущено:

Завідувач кафедри

_____ **Вадим МУХІН**

«__» _____ 2024 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальністю 122 Комп'ютерні науки

освітня програма (спеціалізація) Інтелектуальні сервіс-орієнтовані розподілені
обчислювання

на тему: «Самонавчальні системи для розпізнавання людської діяльності
Reinforcement learning for human activity recognition»

Виконав (-ла): студент (-ка) II курсу, групи ДА-21мп

Цибін Максим Дмитрович _____

Науковий керівник: доцент, к.т.н. Кислий Р.В. _____

Рецензент: доцент кафедри цифрових технологій

в енергетиці НН ІАТЕ, к.т.н., доц Шаповалова С. І. _____

Консультант з нормконтролю: доцент, к.т.н. Кирюша Б.А. _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____

Київ – 2024

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Інститут/факультет Прикладного системного аналізу

(повна назва)

Кафедра Системного проектування

(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність 122 Комп'ютерні науки

(код і назва)

Освітня програма (спеціалізація) Інтелектуальні сервіс-орієнтовані розподілені обчислювання

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Вадим МУХІН

«__» _____ 2023 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Цибіна Максима Дмитровича

1. Тема дисертації «Самонавчальні системи для розпізнавання людської діяльності Reinforcement learning for human activity recognition», науковий керівник дисертації Кислий Роман Володимирович, кандидат технічних наук, затверджені наказом по університету від __ листопада 2023 р. № _____

2. Термін подання студентом дисертації – 15 грудня 2023 р.

3. Об'єкт дослідження

Об'єктом дослідження є системи машинного навчання для розпізнавання людської діяльності з використанням навчання з підкріпленням

4. Вихідні дані

Предметом дослідження є системи машинного навчання для розпізнавання людської діяльності, що базуються на навчанні з підкріпленням, а також дані про активності людей та зворотній зв'язок, що використовується для навчання цих систем.

5. Перелік завдань, які потрібно розробити

- пошук датасетів з видами і даними про людську діяльність
- дослідження систем машинного навчання для оцінювання і прогнозування людської діяльності
- дослідження методів створення систем машинного навчання з використанням навчання з підкріпленням
- аналіз отриманих результатів
- Орієнтовний перелік графічного (ілюстративного) матеріалу
- графіки з результатами прогнозування

9. Дата видачі завдання – 25 червня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Початок практики	01 вересня 2023	
2	Ознайомлення з нормативними документами	07 вересня 2023	

3	Пошук датасетів з видами і даними про людську діяльність	16 вересня 2023	
4	Дослідження систем машинного навчання для оцінювання і прогнозування людської діяльності	21 вересня 2023	
5	Дослідження методів створення систем машинного навчання з використанням навчання з підкріпленням	27 вересня 2023	
6	Аналіз отриманих результатів	6 жовтня 2023	
7	Орієнтовний перелік графічного (ілюстративного) матеріалу	13 жовтня	
8	Графіки з результатами прогнозування	20 жовтня 2023	
9	Звіт з практики	25 жовтня 2023	
10	Тестування системи та аналіз ефективності розроблених систем	30 жовтня 2023	
11	Розробка стартап-проекту	07 грудня 2023	
12	Підготовка дисертації до захисту	13 грудня 2023	
13	Захист дисертації	08 січня 2024	

Студент _____ М. Д. Цибін
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Р. В. Кислий
(підпис) (ініціали, прізвище)

РЕФЕРАТ

магістерської дисертації Цибіна Максима Дмитровича

на тему «Самонавчальні системи для розпізнавання людської діяльності

Reinforcement learning for human activity recognition»

Загальний обсяг роботи с. 00, рис. 00, 00 таблиць, додатки 0, джерел 00

Актуальність теми магістерської дисертації полягає у тому, що розробка системи розпізнавання людської діяльності з використанням навчання з підкріпленням може сильно допомогти людям відстежувати свої показники коректно, та на основі отриманих даних отримувати актуальні рекомендації від лікарів або тренерів, наприклад якщо мова йде про спортивну чи побутову активність. Особливо це актуально для тих чия рухова поведінка може відрізнитися від загальноприйнятої, це можуть бути люди з певними вадами, або ті що знаходяться у процесі реабілітації після травм, операцій. Таким чином темою дослідження є покращення адаптивності систем розпізнавання людської діяльності(HAR) під конкретного користувача, враховуючи його особливості рухів та зміни рухової поведінки з часом за допомогою методів Reinforcement Learning.

Мета та задачі дослідження. Метою цього дослідження є дослідити та покращити адаптивність систем розпізнавання людської діяльності(HAR) під конкретного користувача, враховуючи його особливості рухів та зміни рухової поведінки з часом за допомогою методів Reinforcement Learning.

Для досягнення мети були поставлені такі задачі:

- 1) Аналіз методів та систем розпізнавання людської діяльності на основі сенсорів з особистих пристроїв.

- 2) Аналіз методів створення систем машинного навчання з використанням reinforcement learning.
- 3) Розробка власної системи з використанням навчання з підкріпленням для розпізнавання людської діяльності.
- 4) Аналіз роботи даної системи, визначення можливостей покращити її.

Об'єкт дослідження: системи машинного навчання для розпізнавання людської діяльності з використанням навчання з підкріпленням.

Предмет дослідження: системи машинного навчання для розпізнавання людської діяльності, що базуються на навчанні з підкріпленням, а також дані про активності людей та зворотній зв'язок, що використовується для навчання цих систем.

Наукова новизна. Більшість з існуючих систем розпізнавання людської діяльності є закритими та не дозволяють вільно використовуватися та доповнюватися сторонніми розробниками. Також ці системи часто не коректно розпізнають людську діяльність і не адаптуються під конкретну людину, й навпаки можуть з часом деградувати. Перевагою нашої розробленої системи є сильна адаптивність під конкретного користувача, навіть якщо з часом його рухова поведінка змінюється, система підлаштовується автоматично, а також можливість доповнювати і покращувати систему та впроваджувати її у інші додатки.

Також дуже нетиповим є використання навчання з підкріпленням для нашої задачі класифікації. Це є поштовхом для подальших досліджень і покращень таких систем.

Приклади можливих застосувань та практичної цінності результатів магістерської роботи:

- 1) Розроблену систему можливо інтегрувати у мобільний додаток смартфона, або смарт годинника для розпізнавання спортивних та щоденних

активностей. Також можна інтегрувати з медичними системами для аналізу показників.

- 2) Розроблена система гнучка, її можна доповнювати і покращувати для майбутніх наукових досліджень.
- 3) Є показовим результатом успішного застосування reinforcement learning у задачах класифікації.

Публікація:

1. Самонавчальна система розпізнавання людської діяльності з використанням навчання з підкріпленням /Цибін М. Д., Кислий Р. В. // Системні науки та інформатика: збірник доповідей II науково-практичної конференції «Системні науки та інформатика», 4–8 грудня 2023 року, Київ. – К., НН ІІСА КПІ ім. Ігоря Сікорського, 2023. – с. 382 - 387

Ключові слова: розпізнавання людської діяльності, навчання з підкріпленням, машинне навчання, штучний інтелект.

ABSTRACT

master's thesis by Tsybin Maksym Dmytrovych

on the topic "Reinforcement Learning for Human Activity Recognition"

The total volume of the work is 00 pages, Fig. 00, 00 tables, 0 appendices, and 00 sources.

The relevance of the master's thesis topic lies in the development of a human activity recognition system using reinforcement learning, which can significantly assist individuals in tracking their metrics accurately. Based on the obtained data, it can provide relevant recommendations from doctors or trainers, especially in sports or daily activities. This is particularly relevant for individuals whose movement behavior may differ from the norm, such as people with specific disabilities or those undergoing rehabilitation after injuries or surgeries. Therefore, the aim was to investigate and enhance the adaptability of Human Activity Recognition (HAR) systems for specific users, considering their movement patterns and changes over time using Reinforcement Learning methods.

Research Objective and Tasks. The aim of this research is to investigate and enhance the adaptability of Human Activity Recognition (HAR) systems for specific users, considering their movement patterns and changes over time using Reinforcement Learning methods.

To achieve this aim, the following tasks were outlined:

- 1) Analysis of methods and systems for human activity recognition based on personal device sensors.
- 2) Analysis of methods for creating machine learning systems using reinforcement learning.
- 3) Development of a proprietary system using reinforcement learning for human activity recognition.

- 4) Analysis of the developed system's performance and identification of possibilities for improvement.

Research Object: Machine learning systems for human activity recognition using reinforcement learning.

Research Subject: Machine learning systems for human activity recognition based on reinforcement learning, as well as data on human activities and feedback utilized for training these systems.

Scientific Novelty. Most existing human activity recognition systems are closed and do not allow easy integration or supplementation by third-party developers. Additionally, these systems often fail to accurately recognize human activities and do not adapt to individual users. Conversely, they might degrade over time. The advantage of our developed system lies in its strong adaptability to individual users. Even if their movement behavior changes over time, the system adjusts automatically. Furthermore, it allows for supplementation, enhancement, and integration into other applications. The use of reinforcement learning for our classification task is also highly unconventional, serving as a catalyst for further research and improvements in such systems.

Examples of Possible Applications and Practical Value of Master's Work

- 1) The developed system can be integrated into a smartphone's mobile application or a smartwatch for recognizing sports and daily activities. It can also be integrated with medical systems for analyzing health indicators.
- 2) The developed system is flexible and can be augmented and improved for future scientific research.
- 3) It serves as a demonstrative success of applying reinforcement learning in classification tasks.

Publication:

1. "Self-learning Human Activity Recognition System using Reinforcement Learning / Tsybin M. D., Kyslyi R. V. // Systems Sciences and Informatics: Proceedings of the II Scientific-Practical Conference "Systems Sciences and Informatics", December 4-8, 2023, Kyiv. – K., NN IPSA Igor Sikorsky KPI, 2023. – pp. 382-387

Keywords: human activity recognition, reinforcement learning, machine learning, artificial intelligence.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	12
РОЗДІЛ 1. МЕТОДИ ТА СИСТЕМИ РОЗПІЗНАВАННЯ ЛЮДСЬКОЇ ДІЯЛЬНОСТІ НА ОСНОВІ СЕНСОРІВ З ОСОБИСТИХ ПРИСТРОЇВ.....	13
1.1 Збір даних.....	13
1.2 Попередня Обробка Даних.....	15
1.3 Техніки Машинного Навчання.....	16
1.4 Навчання з Підкріпленням (RL).....	23
1.5 Існуючі Системи Розпізнавання Людської Діяльності.....	27
1.6 Проблеми та майбутнє систем розпізнавання людської діяльності.....	29
1.7 ВИСНОВКИ.....	31
РОЗДІЛ 2. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЛЮДСЬКОЇ ДІЯЛЬНОСТІ.....	33
2.1 Початковий набір даних.....	33
2.2 Мобільний додаток для збору даних з датчиків.....	35
2.3 Початкова обробка даних.....	41
2.4 Реалізація базових моделей для класифікації активостей.....	41
2.5 Реалізація алгоритму навчання з підкріпленням Монте Карло.....	44
2.6 Опис реалізації механізмів адаптивності системи розпізнавання людської діяльності.....	45
2.7 ВИСНОВКИ.....	48
РОЗДІЛ 3. ОПИС РЕЗУЛЬТАТІВ ТЕСТУВАННЯ СИСТЕМИ.....	50
3.1 Тестування при обраній базовій моделі LSTM.....	50
3.2 Тестування при обраній базовій моделі LGBM.....	55
3.3 ВИСНОВКИ.....	59
РОЗДІЛ 4. РОЗРОБКА СТАРТАП ПРОЄКТУ «Самонавчальні системи для розпізнавання людської діяльності Reinforcement learning for human activity recognition».....	60
4.1 Опис ідеї проєкту.....	60
4.2 Технологічний аудит проєкту.....	64
4.3 Аналіз ринкових можливостей.....	65
4.4 Розробка ринкової стратегії проєкту.....	74
4.5 Розробка маркетингової програми.....	79
4.6 Висновки до розділу.....	84
ВИСНОВКИ.....	86
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	88
ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	89

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

RL – навчання з підкріпленням

RNN – рекурентна нейронна мережа

LSTM – Long Short-Term Memory

LGBM – Light Gradient Boosting Machine

DDPG – Deep Deterministic Policy Gradient

HAR – розпізнавання людської діяльності

РОЗДІЛ 1. МЕТОДИ ТА СИСТЕМИ РОЗПІЗНАВАННЯ ЛЮДСЬКОЇ ДІЯЛЬНОСТІ НА ОСНОВІ СЕНСОРІВ З ОСОБИСТИХ ПРИСТРОЇВ

Розпізнавання людської активності (HAR) - це міждисциплінарна концепція, яка зосереджується на ідентифікації та класифікації ряду людських активностей через дані, отримані з різних датчиків. Ця область перетинається з галузями комп'ютерних наук, аналітики даних та технології датчиків. Системи HAR покликані розпізнавати такі активності, як ходьба, біг, сидіння або більш складні жести, і стають все більш розповсюдженими у нашому технологічно-орієнтованому світі. Вони відіграють ключову роль у покращенні інтерактивних технологій, найбільш розповсюджені у таких сферах, як охорона здоров'я для моніторингу пацієнтів, спортивна наука для аналізу продуктивності та розумні системи для підвищення стандартів життя. Основа HAR полягає у здатності обробляти дані датчиків за допомогою складних алгоритмів, перетворюючи сирі дані на значущі інсайти про людську поведінку та взаємодію з оточенням.

1.1 Збір даних

Збір даних у сфері Розпізнавання Людської Активності (HAR) здійснюється переважно через масив датчиків, вбудованих у особисті пристрої. Наприклад у смартфон, як бачимо на рис. 1.1

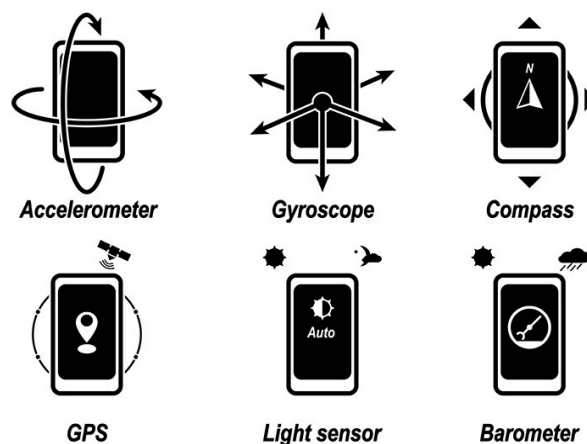


Рисунок 1.1 Приклади датчиків у смартфоні [1]

Ці датчики безперервно записують дані, які відображають фізичні рухи користувача та його взаємодію з навколишнім середовищем [1]. Наприклад, акселерометри та гіроскопи у смартфонах та носимих пристроях фіксують дані, пов'язані з рухом, тоді як монітори серцевого ритму та GPS-датчики відстежують фізіологічні реакції та діяльність, пов'язану з місцезнаходженням, відповідно. Ці дані зазвичай збираються в режимі реального часу та можуть зберігатися локально на пристрої або передаватися на сервер для більш складної обробки. Ключовим аспектом цього етапу є високочастотний запис даних датчиків, що забезпечує багатий набір даних, точно відображаючи широкий спектр людських активностей.

Основні датчики включають:

Акселерометри: Ці датчики вимірюють сили прискорення. У HAR вони важливі для виявлення шаблонів руху, таких як ходьба, біг або раптові рухи.

Гіроскопи: Гіроскопи доповнюють акселерометри, вимірюючи орієнтацію та обертання. Вони важливі для розуміння постави та обертальних рухів користувача.

Магнітометри: Призначені для визначення розташування у просторі. Визначає положення об'єкту відносно магнітного полюсу землі. Важливі для активностей пов'язаними з поворотами.

Датчики GPS (Глобальна Система Визначення Місцезнаходження): Датчики GPS надають дані географічного розташування, які є критично важливими для діяльності, яка включає відстеження місцезнаходження, такої як біг на відкритому повітрі або навігаційна допомога.

Датчики Частоти Серцевих Скорочень: Ці датчики моніторять серцевий ритм користувача, надаючи цінні дані для діяльності, пов'язаної зі здоров'ям та відстеженням спортивних занять.

Крокоміри: Вбудовані у багато фітнес-трекерів, крокоміри рахують кроки, допомагаючи кількісно оцінити фізичні активності, такі як ходьба або піші походи.

Барометри: Ці датчики вимірюють атмосферний тиск і корисні для визначення висоти, що може бути важливим для таких діяльностей, як піші походи або виявлення поверху в багатоповерхових будівлях.

Кожен з цих датчиків вносить унікальний набір даних, дозволяючи системам HAR створювати всебічну картину активностей користувача. Об'єднання даних з декількох типів датчиків часто призводить до більш точного та надійного розпізнавання активності, підвищуючи загальну ефективність системи HAR. В даній роботі використовуються переважно дані з акселерометрів та гіроскопів, також в розробленому додатку можливо отримати дані магнітометра.

1.2 Попередня Обробка Даних

Після збору даних вони проходять попередню обробку, яка є важливою для підвищення якості та корисності даних для HAR.

Етапи попередньої обробки включають [2]:

Фільтрація Шумів: Дані датчиків часто містять шуми через різні фактори, такі як неточності датчиків або зовнішні впливи навколишнього середовища. Техніки фільтрації шумів застосовуються для очищення даних.

Нормалізація: Цей процес передбачає масштабування даних у певному діапазоні, роблячи їх послідовними та легшими для порівняння. Нормалізація особливо важлива, коли дані з різних датчиків поєднуються.

Сегментація: Неперервний потік даних датчиків сегментується на менші, керовані фрагменти. Це допомагає ізолювати певні активності в даних для більш зосередженого аналізу.

Масштабування Ознак: Ознаки можуть масштабуватися, щоб гарантувати, що жодна з них не домінує у аналізі через свій масштаб, підвищуючи продуктивність наступних моделей машинного навчання.

Виділення ознак: Виділення ознак є критичним кроком, де з попередньо оброблених даних отримується значуща інформація. Ознаки - це специфічні вимірювання або характеристики, видобуті з даних датчиків, які є важливими для розпізнавання активностей. Вони можуть включати статистичні ознаки, такі як середнє значення, варіація та пікові значення з часових рядів, або більш складні ознаки, такі як представлення у частотній області. Вибір ознак значно впливає на точність та ефективність системи HAR і часто вимагає знань у певній області для визначення найбільш важливих ознак для конкретних активностей.

1.3 Техніки Машинного Навчання

Техніки машинного навчання є невід'ємною частиною HAR, дозволяючи класифікувати та прогнозувати людські активності на основі даних датчиків. Основні техніки машинного навчання, що використовуються в HAR, включають:

Дерева Рішень: Використовується для класифікації або регресії. Їх основний принцип полягає у розбитті даних на все менші групи за допомогою послідовності простих правил або "рішень". Цей процес можна порівняти з грою в "20 питань", де кожне наступне питання ставиться на основі відповідей на попередні [3].

Дерево рішень складається з вузлів та гілок:

1. **Кореневий вузол (Root Node):** Точка, з якої починається дерево. Він містить усю вибірку даних.
2. **Вузли рішень (Decision Nodes):** Вузли, де проводяться рішення, які ділять дані на дві або більше гілок за певним атрибутом.

3. **Гілки (Branches):** Лінії, які відділяють вузли, показуючи можливі результати рішення.
4. **Листові вузли (Leaf Nodes) або кінцеві вузли:** Кінцеві вузли, що представляють результат класифікації або регресії.

Процес побудови дерева рішень включає визначення найкращих питань або "рішень" на кожному етапі. Це зазвичай відбувається за допомогою критеріїв, таких як приріст інформації або індекс Джині, які вимірюють як добре конкретне розбиття розділяє дані [4].

Support vector machine (SVM): SVM гарно підходять для класифікації складних активностей, оскільки вони можуть обробляти багатовимірні дані та показують хороші результати за чіткого розділення між активностями [3]. Використовується для класифікації або регресійних задач. Основний принцип роботи SVM полягає в пошуку так званої гіперплощини (або межі рішення) в просторі ознак, яка найкращим чином розділяє різні класи даних. Ось ключові моменти у принципі роботи SVM:

1. **Максимізація відстані між класами:** SVM шукає гіперплощину, яка максимізує відстань між найближчими точками різних класів, відомими як опорні вектори (support vectors).
2. **Робота в багатовимірному просторі:** Хоча гіперплощина є плоскою поверхнею, SVM може працювати в дуже багатовимірних просторах, навіть якщо ознаки не є лінійно роздільними.
3. **Ядрові функції (Kernel Trick):** Якщо дані не можна ефективно розділити лінійно, SVM використовує ядрові функції для перетворення даних в більш багатовимірний простір, де вони можуть бути лінійно розділені [5].
4. **Регуляризація:** SVM включає параметри регуляризації, які допомагають уникнути перенавчання, збалансувавши складність моделі та її здатність правильно класифікувати тренувальні дані.

Нейронні Мережі: Особливо глибокі навчальні моделі, такі як Згорткові Нейронні Мережі (CNNs) та Рекурентні Нейронні Мережі (RNNs), вмілі у роботі з великими та складними наборами даних, характерними для HAR, пропонуючи високу точність у розпізнаванні активностей. Саме тому й було обрано удосконалену рекурентну нейронну мережу для поставленої задачі [6].

Recurrent Neural Networks (RNN) є типом нейронної мережі, що особливо ефективні для обробки послідовних даних, що дуже актуально саме для розпізнавання людської діяльності. Основною особливістю RNN є їх здатність "запам'ятовувати" попередній стан при обробці нових даних, що робить їх ідеальними для завдань, де контекст є важливим. Схематично структура зображена на рисунку 1.2.

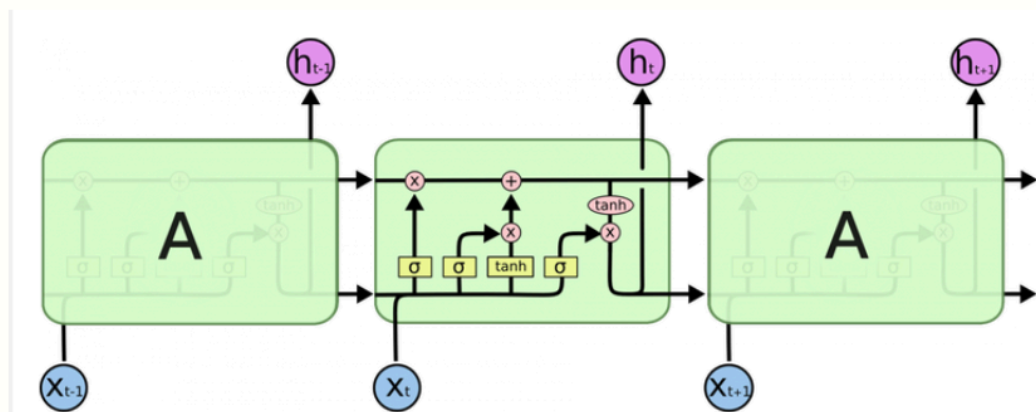


Рисунок 1.2 Схематична структура RNN [6]

Математично роботу RNN можна описати так:

1. Основні Рівняння RNN

RNN оновлює свій внутрішній стан h_t на кожному кроці t , враховуючи поточний вхід x_t та попередній стан h_{t-1} . Це можна виразити як: $h_t = f(Ux_t + Wh_{t-1} + b)$ де:

- h_t - це новий стан на момент часу t ,

- x_t - це вхід на момент часу t ,
- h_{t-1} - це попередній стан,
- U та W - матриці ваг, які необхідно навчити,
- b - вектор зсуву,
- f - активаційна функція, як-от \tanh або ReLU .

2. Вихідний Шар

Для отримання виходу RNN, останній стан h_t може бути перетворений на вихід y_t : $y_t = g(Vh_t + c)$ де:

- y_t - це вихід на момент часу t ,
- V - матриця ваг для вихідного шару,
- c - вектор зсуву для вихідного шару,
- g - активаційна функція, як-от softmax для задач класифікації.

3. Проблеми та Удосконалення

Традиційні RNN часто стикаються з проблемами зникнення або вибуху градієнтів при тренуванні через довгі послідовності. Для вирішення цих проблем були розроблені більш складні версії RNN, такі як LSTM (Long Short-Term Memory) і GRU (Gated Recurrent Units), які використовують спеціальні механізми "запам'ятовування" та "забування" інформації для підтримки довготривалих залежностей. Таким чином обрали як одну з основних моделей для розпізнавання саме LSTM.

4. Принцип роботи LSTM

LSTM (Long Short-Term Memory) - це спеціалізований тип RNN (Recurrent Neural Network), розроблений для вирішення проблеми зникнення градієнтів, яка часто виникає при тренуванні традиційних RNN на довгих послідовностях даних. LSTM здатні зберігати інформацію на довгі періоди часу та визначати,

коли її слід зберегти, змінити або видалити [7]. Ось основні компоненти та принципи роботи LSTM:

1. Структура Клітини LSTM

Кожна клітина LSTM містить три основні компоненти: ворота забуття (forget gate), ворота вводу (input gate), та ворота виводу (output gate). Ці ворота визначають, як інформація буде оброблятися та пересуватися через клітину [7].

2. Ворота Забуття (Forget Gate)

Ворота забуття вирішує, яку інформацію викинути з клітинного стану. Це робиться за допомогою сигмоїдної функції:

$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$ де f_t - вентиль забуття на часі t ,

W_f - ваги ворота забуття,

h_{t-1} - попередній вихідний стан,

x_t - поточний вхід, і b_f - зсув.

3. Ворота Вводу (Input Gate)

Ворота вводу визначають, яку нову інформацію слід додати до клітинного стану. Це складається з двох частин: сигмоїдної функції, яка вирішує, які значення оновити, і тангенса гіперболічного, який створює новий вектор кандидатів C_t для додавання до стану:

$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$

$C_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$ де i_t - ворота вводу,

W_i, W_C - ваги, а b_i, b_C - зсуви.

4. Оновлення Клітинного Стану

Клітинний стан оновлюється шляхом поєднання старого стану C_{t-1} з новим станом C_t :

$C_t = f_t * C_{t-1} + i_t * C_t$ де C_t - новий клітинний стан.

5. Ворота Виводу (Output Gate)

На останньому етапі, ворота виводу визначають, який наступний прихований стан слід вивести. Це залежить від клітинного стану, пропущеного через тангенс гіперболічний (щоб нормалізувати значення між -1 та 1), та сигмоїдної функції: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

$h_t = o_t * \tanh(C_t)$ де o_t - ворота виводу,

h_t - вихідний стан,

W_o - ваги воріт виводу, і b_o - зсув.

LSTM ефективно здійснює обробку даних зі складними, довготривалими залежностями, такими як мова, текст або часові ряди, завдяки своїм унікальним вентилям та механізмам оновлення стану.

Ансамблеві методи:

LightGBM (Light Gradient Boosting Machine) є одним з найкращих ансамблевих методів, що належить до категорії бустингових алгоритмів. Він використовує градієнтний бустинг над деревами рішень і відрізняється високою швидкістю та ефективністю, особливо при роботі з великими об'ємами даних.

Принцип Роботи LightGBM

1. **Ітеративний Підхід:** LightGBM побудовує ансамбль послідовно, додаючи нові дерева, які покращують прогнози попередніх дерев. На кожному

кроці бустингу алгоритм мінімізує функцію втрат, фокусуючись на прикладах, що були неправильно класифіковані раніше [8].

2. **Функція Втрат:** Враховуючи функцію втрат L , виправлення кожного нового дерева визначається як:
3. $L(y, F_{t-1}(x) + h_t(x))$ де y - це істинне значення, $F_{t-1}(x)$ - прогноз попереднього ансамблю, $h_t(x)$ - внесок нового дерева.
4. **Вибір Точки Розбиття:** На відміну від традиційного градієнтного бустингу, LightGBM використовує стратегію вибору точки розбиття, засновану на гістограмах, що дозволяє ефективно обробляти великі набори даних. Під час розбиття вузла дерева, LightGBM агрегує значення ознак у фіксовану кількість "кошиків" для створення гістограми, що знижує обчислювальну складність.
5. **Оптимізація з Використанням Градієнта:** Оновлення ваг в ансамблі виконується з використанням градієнта функції втрат:

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \frac{\partial L}{\partial w} \text{ де } \eta - \text{це швидкість навчання, } w - \text{ваги моделі.}$$

6. **Обрізка Дерев:** Для запобігання перенавчанню LightGBM використовує стратегію обрізки дерев, обмежуючи їх глибину або кількість листків.
7. **Використання Підмножини Даних (Subsampling):** LightGBM може використовувати підмножини даних для побудови кожного дерева в ансамблі, що додатково знижує обчислювальну складність.

Враховуючи свою швидкість та ефективність, LightGBM є відмінним вибором для завдань HAR (Human Activity Recognition), особливо коли потрібно обробляти великі об'єми даних або забезпечувати швидке реагування в реальному часі [8]. LightGBM здатний ефективно використовувати велику кількість ознак і впоратися зі складною структурою даних, що часто зустрічається в задачах аналізу людської активності.

1.4 Навчання з Підкріпленням (RL)

Навчання з підкріпленням (RL) - це парадигма машинного навчання, в якій агент навчається приймати рішення, виконуючи дії в середовищі та отримуючи зворотний зв'язок у вигляді винагород або штрафів. Цей процес навчання аналогічний навчання дитини через винагороди та виправлення. У контексті RL агент навчається вибирати оптимальний набір дій для максимізації сукупної винагороди з часом. На відміну від традиційного навчання з учителем, яке залежить від маркованих наборів даних, RL особливо ефективний у сценаріях, де необхідно адаптуватися до динамічних середовищ або де оптимальні рішення змінюються з часом. Його значення для Розпізнавання Людської Активності (HAR) полягає в здатності неперервно адаптуватися та вдосконалювати алгоритми розпізнавання на основі зворотного зв'язку від користувачів та зміни шаблонів людської активності.

В даній роботі були розглянуті метод навчання з підкріпленням Монте Карло та DDPG (Deep Deterministic Policy Gradient) [9] як найбільш вдалі для задачі класифікації людських активностей.

1. Метод Навчання з Підкріпленням Монте Карло

Метод Монте-Карло в контексті навчання з підкріпленням використовується для оцінки значень стратегій або дій на основі випадкових зразків з історії взаємодій агента з оточенням. У контексті розпізнавання людської діяльності (Human Activity Recognition, HAR) цей метод може бути застосований для визначення оптимальної стратегії агента, наприклад, у виборі найбільш вірогідної активності на основі спостережень [9].

Основні Концепції:

1. **Стан:** У контексті HAR, стан може бути представлений через набір датчикових читань або характеристик, що відображають поточну активність людини (наприклад, ходьба, біг, сидіння).

2. **Винагорода:** Визначається на основі точності ідентифікації активності. Наприклад, агент може отримувати позитивну винагороду за правильне розпізнавання та негативну - за помилку.
3. **Політика:** Стратегія або правило, яке визначає, яку дію агент повинен виконати в певному стані.

Принцип Роботи Методу Навчання з Підкріпленням Монте-Карло:

Метод Монте-Карло для RL оцінює значення політики на основі середнього виграшу з великої кількості епізодів. Ось як це можна виразити математично:

1. **Оцінка Значення Дії (Action Value Estimation):**

$$Q(s,a) = \frac{1}{N(s,a)} \sum_{i=1}^{N(s,a)} G_i \text{ де } Q(s,a) - \text{ це оцінка значення дії } a \text{ у стані } s,$$

$N(s,a)$ - кількість разів, коли дія a була вибрана у стані s , і G_i - виграш в i -му епізоді.

2. **Оцінка Значення Стану (State Value Estimation):** Якщо політика агента детермінована, можна оцінити значення стану:

$$V(s) = \frac{1}{N(s)} \sum_{i=1}^{N(s)} G_i$$

де $V(s)$ - це оцінка значення стану s , а $N(s)$ - кількість разів, коли стан s був відвіданий.

3. **Удосконалення Політики:** На основі оцінок значень дій агент може удосконалити свою політику, вибираючи дії з найвищими значеннями $Q(s,a)$ для кожного стану.

Цей підхід дозволяє агенту навчатися і вибирати найбільш ймовірні активності на основі історичних даних, постійно покращуючи свою здатність до

розпізнавання. Такий метод особливо корисний в динамічних середовищах, де поведінка людини може змінюватися, і агенту необхідно адаптуватися до цих змін.

2. Метод Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient (DDPG) є алгоритмом підсиленого навчання (Reinforcement Learning, RL), який поєднує концепції з глибокого навчання і градієнта політики. Хоча DDPG зазвичай використовується для навчання агентів у безперервних дієвих просторах, його можна адаптувати для застосування в задачах розпізнавання людської діяльності (Human Activity Recognition, HAR), де агент навчається вибирати оптимальні дії на основі спостережень про людську активність [10].

Основні Концепції DDPG:

1. **Актор-Критик Архітектура:** DDPG використовує дві нейронні мережі: одну для визначення політики (актор) і іншу для оцінки значення (критик).
2. **Офф-політика Навчання:** Агент навчається на основі політики, яка відрізняється від тієї, що використовується для генерації даних.
3. **Використання Replay Buffer:** Це дозволяє ефективно використовувати минулі досвіди, надаючи більш стабільне навчання.

Математично DDPG можна описати так:

1. Політика Актора: Актор визначає дію a на основі поточного стану s за допомогою політики π :

$a = \pi(s | \theta^\pi)$ де θ^π - параметри нейронної мережі актора.

2. Функція Втрат Критика: Критик оцінює значення пари стан-дія за допомогою Q-функції:

$$L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i | \theta^Q))^2$$

де $y_i = r_i + \gamma Q(s_{i+1}, \pi(s_{i+1} | \theta^\pi) | \theta^Q)$ - цільове значення,

γ - коефіцієнт зниження,

θ^Q, θ^π - параметри нейронних мереж критика.

3. Оновлення Політики: Актор оновлює свою політику, використовуючи градієнт політики:

$$\nabla_{\theta^\pi} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a | \theta^Q) |_{s=s_i, a=\pi(s_i)} \nabla_{\theta^\pi} \pi(s | \theta^\pi) |_{s_i}$$

4. Soft Update: Для забезпечення стабільності, параметри цільових мереж θ^Q та θ^π оновлюються за допомогою "soft update":

$\theta' \leftarrow \tau \theta + (1 - \tau) \theta'$ де τ - малий параметр, що забезпечує плавне оновлення.

У контексті HAR, DDPG може бути застосований для тренування агента, що навчається визначати оптимальні дії (наприклад, вибір активності або регулювання датчиків) на основі спостережень про поведінку людини. Завдяки своїй здатності ефективно обробляти безперервні дієві простори і використовувати досвід з минулих інтеракцій, DDPG є потужним інструментом для рішення складних завдань у галузі HAR.

Можливі проблеми з застосуванням навчання з підкріпленням

Застосування RL в HAR також має свої виклики. Одним з основних є потреба у добре визначеній системі винагород, яка точно представляє бажані результати. Крім того, моделі RL можуть бути складними та обчислювально вимогливими, що може створювати обмеження з точки зору обробки в реальному часі та розгортання на особистих пристроях з обмеженими обчислювальними ресурсами.

Баланс Між Дослідженням та Використанням: Одним з ключових моментів у RL є баланс між дослідженням (спробами нових дій) та використанням (застосуванням відомої інформації). У HAR цей баланс є критичним для забезпечення того, що система вчиться новим шаблонам та надійно розпізнає відомі активності.

Включення Навчання з Підкріпленням у системи HAR пропонує перспективний шлях для розробки більш адаптивних, персоналізованих та постійно вдосконалюючих рішень для розпізнавання активностей. Однак це вимагає ретельного врахування конструкції системи, структур винагород та обчислювальних обмежень для ефективного використання потенціалу RL у цій сфері.

1.5 Існуючі Системи Розпізнавання Людської Діяльності

Вибір систем розпізнавання людської активності (HAR) зазнав значного зросту з розповсюдженням розумних особистих пристроїв. Ці системи використовують вбудовані датчики в пристроях, таких як смартфони, розумні годинники та фітнес-трекери, для моніторингу та розпізнавання широкого спектру людських активностей. Декілька помітних прикладів включають:

Системи HAR на Базі Смартфонів: Сучасні смартфони, оснащені акселерометрами, гіроскопами та GPS, мають додатки, які можуть відстежувати такі активності, як ходьба, біг, та навіть виявляти падіння [11]. Системи, такі як Google Fit та Apple Health, є яскравими прикладами, пропонуючи надійний моніторинг активностей та функції контролю здоров'я.

Носимі Фітнес-Трекери: Пристрої, такі як Fitbit та Garmin, використовують алгоритми HAR для відстеження конкретних вправ, підрахунку кроків, моніторингу шаблонів сну та навіть оцінки інтенсивності тренувань через датчики серцевого ритму та руху.

Розумні Годинники з Розширеними Можливостями HAR: Розумні годинники, такі як Apple Watch та Samsung Galaxy Watch, еволюціонували не тільки для відстеження базових фізичних активностей, але й для розпізнавання більш тонких активностей, таких як стилі плавання, пози йоги, а також для надання моніторингу ЕКГ.

Моніторингові Пристрої в Охороні Здоров'я: У сфері охорони здоров'я системи HAR інтегровані в носимі пристрої для моніторингу пацієнтів, допомагаючи відстежувати рухи пацієнтів з хронічними захворюваннями, прогрес фізіотерапії та догляд за літніми людьми.

Приклади використання систем розпізнавання людської активності

Виявлення Падіння в Догляді за Літніми Людьми: Одним з прикладів застосувань HAR є системи виявлення падінь для літніх людей. Пристрої, оснащені акселерометрами та гіроскопами, можуть виявляти падіння та автоматично сповіщати доглядачів або медичні служби. Наприклад, функція виявлення падінь Apple Watch широко відома за свій потенціал забезпечити негайну допомогу після інцидентів падіння.

Моніторинг Реабілітації: Системи HAR все частіше використовуються в реабілітації. Дослідження, в якому брали участь пацієнти після інсульту, використовувало носимі датчики для моніторингу та аналізу рухів пацієнтів під час реабілітаційних вправ, надаючи терапевтам цінні дані для адаптації програм відновлення.

Аналіз Спортивної Продуктивності: У спорті системи HAR використовуються для аналізу продуктивності та запобігання травмам. Наприклад, професійні футбольні команди використовують носимі трекери для моніторингу фізичної активності гравців під час тренувань та матчів, допомагаючи оптимізувати тренувальні режими та знижувати ризики травм.

Моніторинг Сну: Пристрої, такі як Fitbit та Withings, використовують алгоритми HAR для відстеження шаблонів сну. Вони моніторять рухи під час сну, щоб надати інсайти щодо якості сну, його тривалості та стадій, допомагаючи в управлінні особистим здоров'ям.

Дослідження Поведінки: HAR також використовується в поведінкових дослідженнях для вивчення шаблонів щоденних активностей. Дослідження, в якому датчики смартфонів використовувались для відстеження таких активностей, як ходьба, біг, та сидячий спосіб життя, надало інсайти щодо шаблонів життя та їхнього впливу на психічне здоров'я.

1.6 Проблеми та майбутнє систем розпізнавання людської діяльності

Галузь Розпізнавання Людської Активності (HAR) стикається з кількома викликами, які критично важливо вирішити для її розвитку:

Конфіденційність Даних: Оскільки системи HAR збирають чутливі особисті дані, такі як місцезнаходження, рух та інформацію, пов'язану зі здоров'ям, забезпечення конфіденційності користувачів є надзвичайно важливим [12]. Існує потреба у надійному шифруванні даних та політиках конфіденційності для захисту даних користувачів від несанкціонованого доступу та зловживань.

Точність Системи: Досягнення високої точності у розпізнаванні активностей, особливо в складних та різноманітних реальних умовах, залишається викликом. Фактори, такі як різноманітність поведінки користувачів, обмеження датчиків та вплив навколишнього середовища, можуть впливати на точність систем HAR.

Обробка Даних у Реальному Часі: Для багатьох застосувань, особливо тих, що стосуються охорони здоров'я та безпеки, критично важлива обробка даних та розпізнавання активностей у реальному часі [13]. Однак це вимагає

ефективних алгоритмів та достатньої обчислювальної потужності, що може бути складно, особливо в невеликих пристроях, що працюють від батареї.

Майбутні Тенденції

Дивлячись у майбутнє, галузь HAR готується до захоплюючих проривів:

Покращення в технологіях датчиків: Майбутні покращення у технології роботи датчиків, наприклад збільшення чутливості та енергоефективності датчиків, можуть значно покращити можливості систем HAR. Інтеграція нових датчиків, таких як датчики шкірної провідності та розширені датчики руху, може забезпечити ще більш детальні інсайти про людські активності.

Інновації у Машинному Навчанні та ШІ: Постійний розвиток алгоритмів машинного навчання, особливо глибокого навчання та навчання з підкріпленням, ймовірно, підвищить точність та ефективність систем HAR. Персоналізація розпізнавання активностей за допомогою ШІ, що адаптується до індивідуальних шаблонів поведінки користувачів, підвищить досвід користувача та відповідність системи.

Обчислення на кінці мережі: Для вирішення викликів з обробки даних у реальному часі, обчислення на кінці мережі, де обробка даних відбувається ближче до джерела даних (наприклад, на смартфоні або носимому пристрої), ймовірно, стане більш поширеною [14]. Цей підхід може знизити затримку та залежність від обробки даних на базі хмари.

Інтеграція з Доповненою та Віртуальною Реальністю: HAR може відіграти значну роль у доповненій та віртуальній реальності (AR/VR), забезпечуючи більш занурювальні та інтерактивні досвіди завдяки точному відстеженню та інтерпретації рухів користувача.

Застосування у Різних Галузях: Очікується, що застосування HAR розширяться на нові галузі, такі як розумні будинки для покращення житлового

середовища, автомобільна промисловість для моніторингу водіїв, та навіть у моніторингу довкілля, розуміючи взаємодію людей з оточенням.

Етичний ШІ та Технології, Що Підвищують Конфіденційність:

Оскільки конфіденційність даних залишається актуальною проблемою, розвиток рамок етичного штучного інтелекту та технологій, що підвищують конфіденційність (PETs), буде важливим. Це включає техніки анонімізації даних та забезпечення того, щоб системи HAR були прозорими та відповідали регуляціям щодо захисту даних.

1.7 ВИСНОВКИ

Розпізнавання людської активності (HAR) стоїть на краю інновацій у технологіях та аналітиці даних, поєднуючи передові техніки машинного навчання та широкий спектр датчиків для точного відстеження та аналізу людської поведінки. Використання таких методів, як дерева рішень, SVM, RNN, а також передових алгоритмів, таких як LSTM і LightGBM, дозволяє точно і глибоко дослідити розв'язання складних задач класифікації та прогнозування активностей. Також значну роль у цьому процесі відіграє навчання з підкріпленням, зокрема методи Монте-Карло та DDPG, що відкривають нові горизонти для адаптації та оптимізації алгоритмів в динамічних середовищах.

Наслідки цих методів та систем мають велике значення для майбутнього технологій особистих пристроїв та розпізнавання людської активності. Неперервна еволюція систем HAR готова глибоко вплинути на спосіб нашої взаємодії з технологіями, роблячи наші пристрої більш інтуїтивними та чутливими до наших потреб та поведінки.

У сфері особистого здоров'я та фітнесу системи HAR прокладають шлях для більш персоналізованих та профілактичних рішень у галузі охорони здоров'я, дозволяючи користувачам точніше моніторити свої показники здоров'я та рівні активності.

У сфері безпеки та догляду за літніми людьми потенціал систем для виявлення надзвичайних ситуацій, таких як падіння або раптові зміни здоров'я, може запропонувати спокій та своєчасну допомогу.

HAR відкриває нові можливості для жестових керувань та контекстно-чутливих застосунків, підвищуючи досвід користувача та доступність.

Більше того, оскільки технології датчиків та алгоритми ШІ продовжують розвиватися, можна очікувати, що системи HAR стануть більш складними, точними. Це призведе до створення розумніших, більш чутливих середовищ, де пристрої можуть передбачати та адаптуватися до людських потреб в реальному часі, безперервно інтегруючи технологію в наше повсякденне життя.

Однак ці досягнення також супроводжуються відповідальністю. Забезпечення конфіденційності даних, усунення упередженості та розвиток етичних практик ШІ буде важливим у формуванні майбутнього, де технології HAR широко впроваджуються та приносять користь.

РОЗДІЛ 2. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЛЮДСЬКОЇ ДІЯЛЬНОСТІ

Під час роботи над даною роботою було розроблено систему розпізнавання людської діяльності з використанням навчання з підкріпленням. А також мобільний додаток для збору інформації з датчиків смартфона для розпізнавання діяльності нашою системою [15]. Система розпізнавання людської діяльності написана мовою програмування Python 3.8.10. Для створення додатку була використана мова програмування Java у середовищі для розробки мобільних додатків Android Studio.

2.1 Початковий набір даних

За початковий набір даних для тренування моделей були обрані дані з дослідження, що проводилося з групою 30 добровольців у віці від 19 до 48 років. Кожна людина виконувала шість активностей (ХОДЬБА, ПІДЙОМ ПО СХОДАХ, СХОДЖЕННЯ ПО СХОДАХ, СИДІННЯ, СТОЯННЯ, ЛЕЖАННЯ), носивши смартфон (Samsung Galaxy S II) на поясі. Використовуючи вбудований акселерометр і гіроскоп, захоплювалися 3-осьове лінійне прискорення та 3-осьову кутову швидкість з постійною частотою 50 Гц. Також експерименти були записані на відео для ручного маркування даних. Отриманий набір даних був випадково розділений на дві частини, де 70% добровольців було вибрано для генерації тренувальних даних та 30% - тестових даних.

Сигнали датчиків (акселерометра та гіроскопа) були попередньо оброблені шляхом застосування фільтрів від шуму, а потім вибірково в межах фіксованих ковзаючих вікон шириною 2,56 секунди та з перекриттям 50% (128 вимірювань на вікно). Сигнал прискорення датчика, який має компоненти гравітації та руху тіла, був розділений за допомогою низькочастотного фільтра Баттерворта на прискорення тіла та гравітацію. Припускається, що гравітаційна сила має лише низькочастотні компоненти, тому був використаний фільтр з

частотою відсікання 0,3 Гц. З кожного вікна отримувався вектор ознак шляхом розрахунку змінних з часової та частотної області (Рисунок 2.1 та рисунок 2.2).

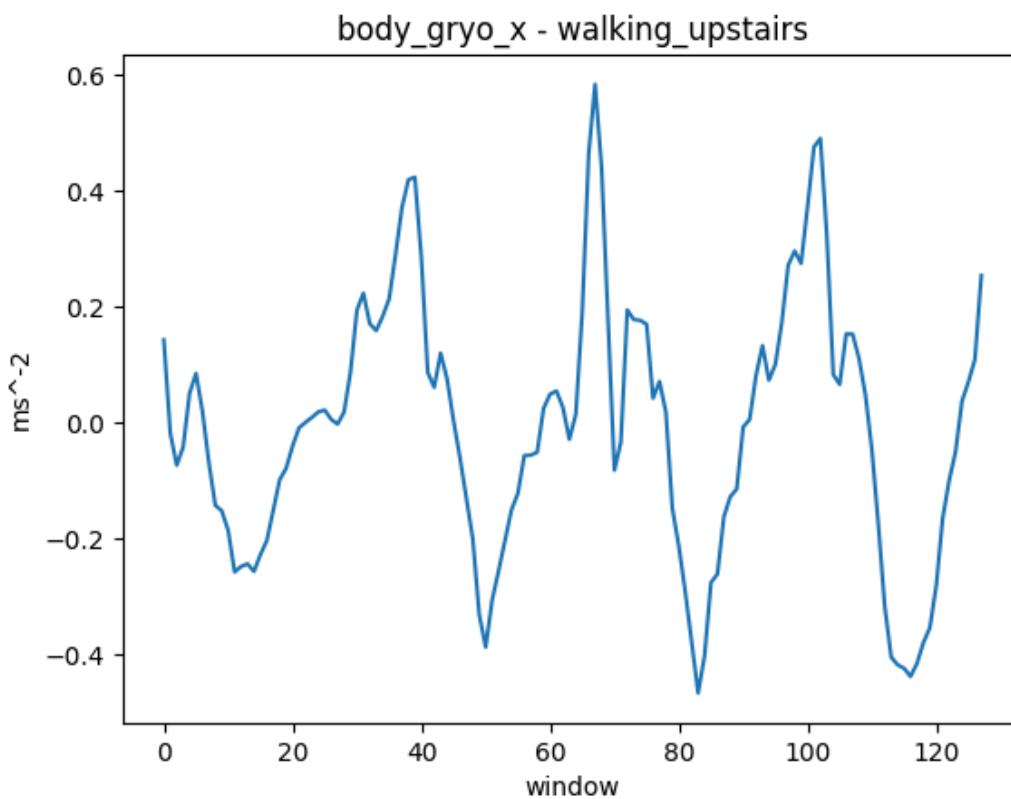


Рисунок 2.1 Приклад даних з гіроскопа по осі X під час підйому по сходах



Рисунок 2.2 Приклад даних з акселерометра по осі X під час підйому по сходах

2.2 Мобільний додаток для збору даних з датчиків

Ключовою ціллю було покращити адаптивність систем розпізнавання людської діяльності. Саме для збору індивідуальних даних і тестування був створений мобільний додаток для смартфонів на базі операційної системи Android.

Мобільний додаток використовує стандартні методи API операційної системи для отримання даних з датчиків з трьох осей акселерометра, гіроскопа та магнітометра.

При першому запуску додатку користувач надає дозвіл на показ сповіщень та доступ до сховища. Далі слід обрати за допомогою перемикачів датчики з яких ми будемо записувати дані. У полі вводу за бажанням слід

вказати назву активності або її ідентифікатор, у випадку, якщо дані будуть використовуватися для тренування (Рисунок 2.3).

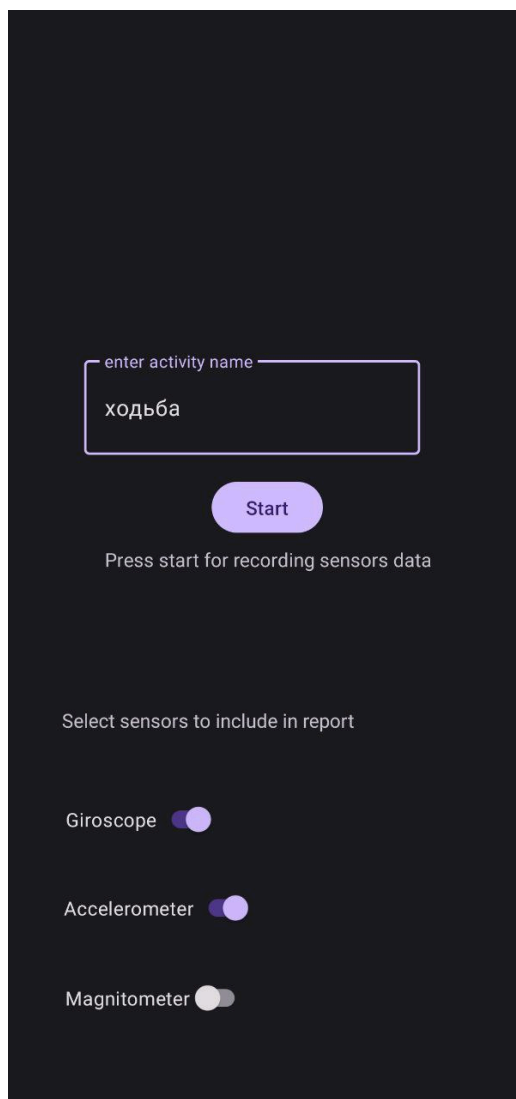


Рисунок 2.3 Початок запису активності в мобільному додатку.

Потім користувачу варто лише натиснути кнопку start і запис активності розпочнеться. У списку сповіщень буде відображатися інформація, що ведеться запис активності (Рисунок 2.4).

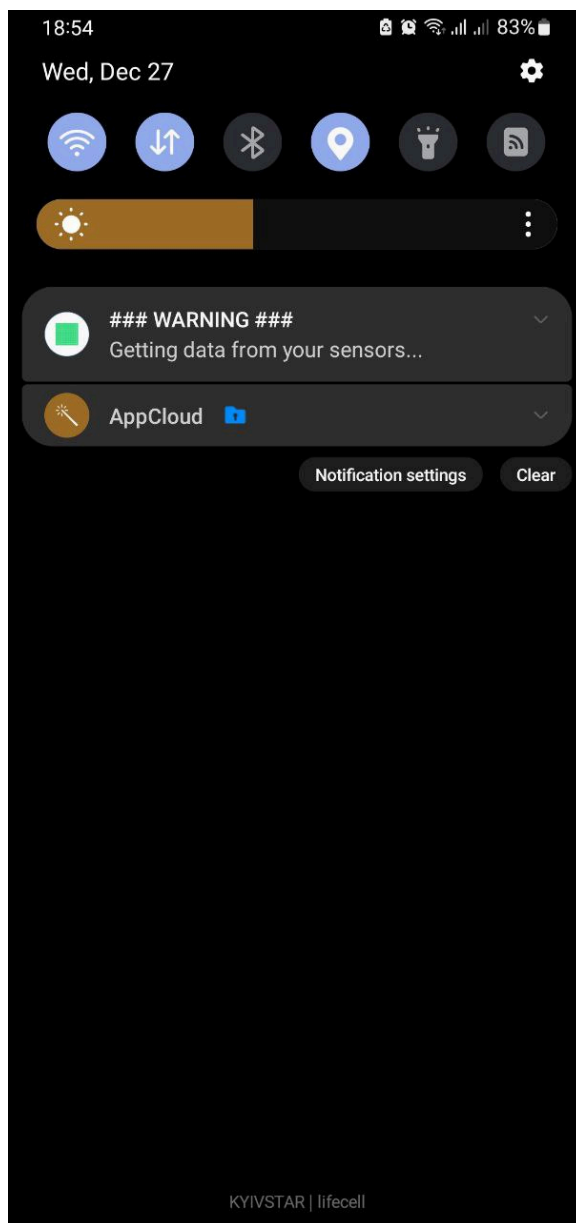


Рисунок 2.4 Сповіщення про запис активності.

Для закінчення запису слід натиснути кнопку stop. Після цього сповіщення про запис зникне і з'явиться кнопка для відкриття файлу (Рисунок 2.5).

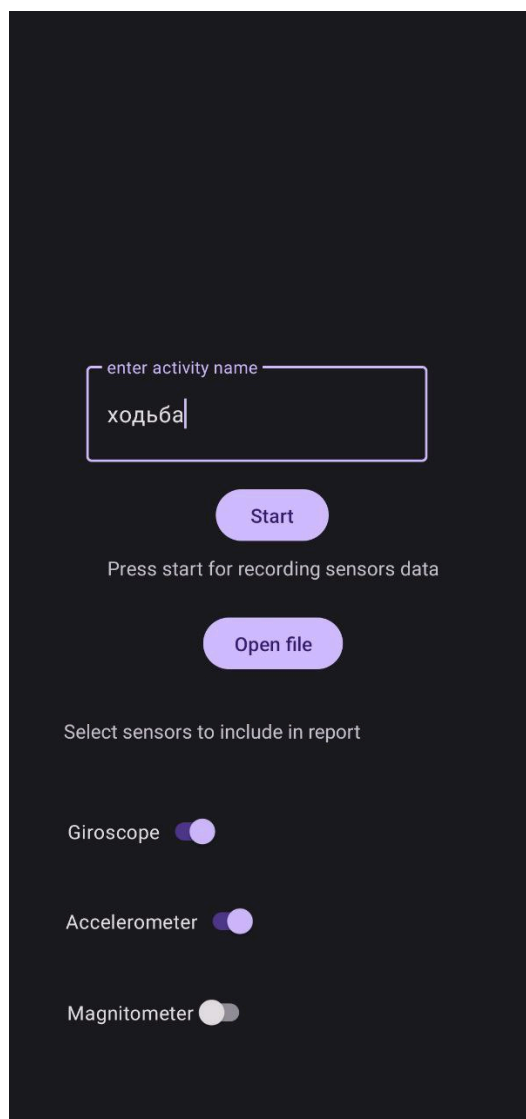


Рисунок 2.5 Завершення запису активності.

У директорії Документи користувач може знайти csv файли з записаними даними обраних датчиків і файл з назвами активності для кожного запису, якщо було заповнене поле вводу назви при записі (Рисунок 2.6).

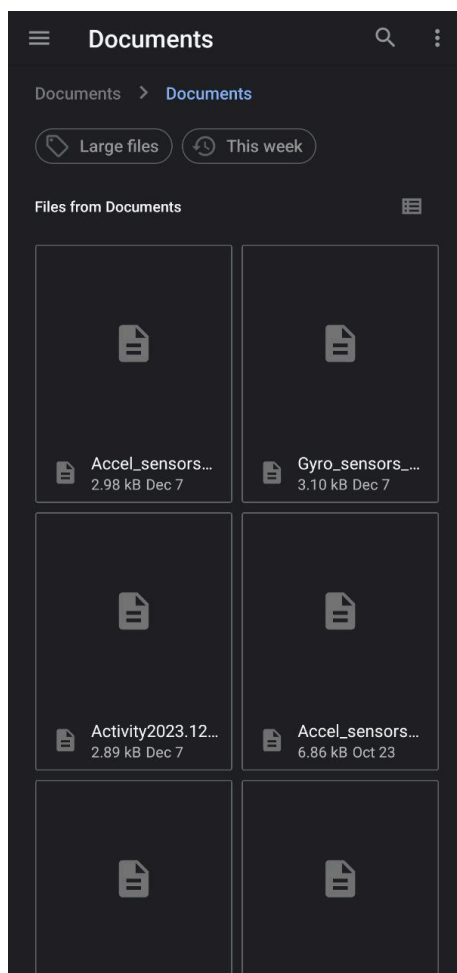


Рисунок 2.6 Файли даних отриманих з датчиків.

Ці файли з записами активностей й були використані для дослідження розробленої системи з розпізнавання людської діяльності.

Accel_sensors_dat...3.12.07.19.33.04 - Read-only

Read Only - To make changes, save a copy...

	A	B	C	D	E	F
1	LSM6DSL	-0.7733	9.48343	2.22182	8.66E+13	
2	LSM6DSL	-0.814	9.44991	2.34871	8.66E+13	
3	LSM6DSL	-0.4549	9.49062	1.71185	8.66E+13	
4	LSM6DSL	-0.419	9.26077	3.28245	8.66E+13	
5	LSM6DSL	0.09816	8.34619	5.93044	8.66E+13	
6	LSM6DSL	-1.7071	7.8051	3.28724	8.66E+13	
7	LSM6DSL	-1.4676	8.64786	3.69665	8.66E+13	
8	LSM6DSL	-1.5419	8.99502	3.28006	8.66E+13	
9	LSM6DSL	-1.1085	8.61673	4.89854	8.66E+13	
10	LSM6DSL	-0.577	9.09078	4.50349	8.66E+13	
11	LSM6DSL	-1.1013	8.34379	4.40773	8.66E+13	
12	LSM6DSL	-0.7853	8.78672	3.67031	8.66E+13	
13	LSM6DSL	-1.0582	8.85136	4.06535	8.66E+13	
14	LSM6DSL	-0.735	8.99741	3.82354	8.66E+13	
15	LSM6DSL	-0.7853	8.93756	3.71341	8.66E+13	
16	LSM6DSL	-0.6033	9.02614	3.41892	8.66E+13	
17	LSM6DSL	-0.9888	8.9591	3.8834	8.66E+13	
18	LSM6DSL	-0.6345	8.93037	3.73735	8.66E+13	
19	LSM6DSL	-0.814	8.82982	4.09648	8.66E+13	
20	LSM6DSL	-0.8595	8.94474	3.62482	8.66E+13	
21	LSM6DSL	-0.9457	8.89925	3.73017	8.66E+13	
22	LSM6DSL	-1.0894	8.86573	4.2138	8.66E+13	
23	LSM6DSL	-0.826	8.77475	3.65834	8.66E+13	
24	LSM6DSL	-0.8571	8.95192	3.33512	8.66E+13	
25	LSM6DSL	-0.7638	8.91122	4.13	8.66E+13	
26	LSM6DSL	-0.7661	8.89207	4.00311	8.66E+13	
27	LSM6DSL	-0.6919	8.80348	4.00311	8.66E+13	
28	LSM6DSL	-0.7159	8.96389	3.73735	8.66E+13	
29	LSM6DSL	-0.8404	8.93037	4.0869	8.66E+13	
30	LSM6DSL	-0.5531	8.79151	3.88579	8.66E+13	
31	LSM6DSL	-0.7326	8.7963	4.00071	8.66E+13	
32	LSM6DSL	-0.7063	8.73165	4.2545	8.66E+13	
33	LSM6DSL	-0.6608	8.75081	3.96719	8.66E+13	
34	LSM6DSL	-0.6201	8.81306	4.06535	8.66E+13	
35	LSM6DSL	-0.6177	8.82024	4.12521	8.66E+13	

fx LSM6DSL Accelerometer

Рисунок 2.7 Представлення даних з акселерометра.

У файлах зберігаються дані по трьом осям датчика, також назва і конкретна модель датчика мобільного пристрою і мітка часу на кожен запис у таблиці (Рисунок 2.7).

2.3 Початкова обробка даних

Для підвищення ефективності тренування моделей, слід підготувати дані. В нашому випадку дані розташовані у різних файлах, тому для зручності об'єднали їх в один файл з роширенням h5. Для зручного переводу в формат Dataframe з бібліотеки pandas, ми попередньо створимо словник з ключами inputs та labels. Також слід переконатися що дані нормалізовані, тобто дані з нашого набору отриманому з дослідження і особистого пристрою мають однакову розмірність, а також видалимо пропущені значення.

В результаті обробки отримали трьох вимірний масив з даними, що містить 10299 записів з активностями, де кожна активність має розмірність (128, 9). І відповідно одновимірний масив міток для кожного запису. Кожна мітка це ідентифікатор, що відповідає певній активності.

0: лежання

1: ходьба,

2: 'підйом сходами',

3: 'спуск сходами',

4: сидіння,

5: стояння,

2.4 Реалізація базових моделей для класифікації активностей

Як базові моделі було обрано LSTM та LGBM. Ці моделі добре підходять для розпізнавання людської діяльності, що є задачею класифікації. Моделі було протестовано на початковому наборі даних, отримана точність прогнозу 0.9455 для LSTM та 0.9845 для LGBM. Це гарний результат, що повністю задовольняє нашу ціль.

Наша модель LSTM реалізована за допомогою бібліотеки Keras у Python і складається з наступних елементів:

Вхідний Шар (Input): Цей шар приймає вхідні дані з формою, що відповідає формі першого виміру тренувальних даних (`self.x_train[0].shape`).

LSTM Шар: Ми використовуємо LSTM шар з 128 одиницями. Параметр `return_sequences=True` вказує, що шар повертає послідовність вихідних даних для кожного часового кроку, що дозволяє цій послідовності передаватися наступним шарам.

Повнозв'язний Шар (Dense): Цей шар містить 64 нейрони з активаційною функцією ReLU (Rectified Linear Unit). Він перетворює вихідні дані з LSTM шару на більш абстрактний рівень.

Глобальний Максимальний Пулінг 1D (GlobalMaxPooling1D): Цей шар зменшує розмірність даних, вибираючи максимальне значення з кожного виміру в послідовності.

Вихідний Шар: Останній повнозв'язний шар має кількість нейронів, що відповідає кількості класів (K). Він використовує активаційну функцію `softmax` для визначення ймовірності приналежності входу до одного з класів.

Компіляція Моделі: Модель компілюється з оптимізатором Adam (зі швидкістю навчання 0.001), функцією втрати `sparse_categorical_crossentropy` та метрикою `accuracy` для оцінки точності моделі.

Ця модель використовується для класифікації часових рядів, що містять декілька класів активностей, із зібраних даних. LSTM шари ефективно обробляють послідовні дані, забезпечуючи здатність моделі "запам'ятовувати" інформацію про попередні стани, що є важливим для задач, де контекст і послідовність даних мають велике значення.

LightGBM (LGBM) - в даній роботі цей алгоритм реалізовано наступним чином:

Підготовка Даних: дані розділені на тренувальний і тестовий набори за допомогою функції `train_test_split`. Це забезпечує, що модель буде тренуватися

на одній частині даних (X_{train} , y_{train}) і валідуватися на іншій (X_{test} , y_{test}), що допомагає уникнути перенавчання та перевіряє загальну здатність моделі до узагальнення.

Створення LGBM Датасетів: Дані для тренування та тестування обгортані в спеціальні об'єкти LGBM Dataset, які оптимізовані для ефективності та швидкості в LGBM.

Налаштування Параметрів: Визначені параметри моделі, включаючи тип задачі ('objective': 'multiclass' для багатокласової класифікації), кількість класів ('num_class'), тип бустингу, метрику для оцінки моделі та швидкість навчання. Ці параметри допомагають налаштувати модель для конкретних вимог задачі HAR.

Тренування Моделі: Модель тренується на даних за допомогою методу `lgb.train`, використовуючи задані параметри та датасети. Валідація виконується на тестовому наборі даних, що допомагає відстежувати і порівнювати ефективність моделі на необхідних даних.

Прогнозування та Оцінка Точності: За допомогою обученої моделі виконується прогнозування на тестових даних. Оцінка прогнозу здійснюється шляхом вибору класу з найвищою ймовірністю (`np.argmax`). Точність моделі (ассигасу) вимірюється за допомогою порівняння прогнозованих класів з фактичними мітками класів у тестовому наборі даних.

Виведення Точності: Отримана точність моделі виводиться для оцінки загальної продуктивності моделі в задачі розпізнавання людської діяльності.

Ця реалізація LGBM в контексті HAR забезпечує високу точність та надійність в розпізнаванні різних типів активностей.

2.5 Реалізація алгоритму навчання з підкріпленням Монте Карло

У розробленій системі моделі LSTM і навчання з підкріпленням методом Монте Карло поєднуються для досягнення більшої ефективності. Модель метода Монте Карло на основі тренувальних даних генерує нові дані схожі на попередні але вони мають більш виразні ознаки активностей. Далі ці дані приймає на вхід наша базова модель. Особливості реалізації саме навчання з підкріпленням:

- Агент використовує евклідову відстань для знаходження схожих станів, дозволяючи оцінити цінність кожного стану за допомогою методу Монте-Карло.
- При обчисленні функції цінності використовується нормалізація, що забезпечує однорідність даних.
- Прогнозування здійснюється шляхом вибору станів з тренувального набору, які найбільше схожі на тести.
- Модель використовується для тренування на основі сформованих прогнозів та виконує валідацію на тестовому наборі даних.

Якщо говорити більш конкретно, опис і методи класу такі:

- Клас MCAgent успадковує від базового класу BaseRLAgent.
- Ініціалізація включає модель, параметри для самовідновлення моделі (`model_healing_mode`), порогове значення точності, обсяг даних для тренування, історію активностей та міток, лічильник історії.
- Вектор значень V ініціалізується з нульовими значеннями для кожного стану в тренувальному наборі даних.
- Викликається функція `compute_value_function` для розрахунку функції цінності на основі тренувальних даних.
- Виконується прогнозування та тренування моделі.

Методи

- `extract_features`: статичний метод для перетворення послідовності даних у плоский вектор.
- `find_similar`: знаходить найбільш схожий стан у тренувальному наборі даних, використовуючи евклідову відстань.
- `compute_value_function`: обчислює функцію цінності, використовуючи евклідову відстань для визначення винагороди.
- `predict`: виконує прогнозування, використовуючи схожі стани з тренувального набору даних для тестового набору.
- `printModelStats`: виводить статистику моделі, включаючи архітектуру моделі, результати тренування та матрицю помилок.

2.6 Опис реалізації механізмів адаптивності системи розпізнавання людської діяльності

У створеній системі ефективно використовується комбінація моделі LSTM та методу Монте-Карло в рамках навчання з підкріпленням для підвищення загальної продуктивності. Метод Монте-Карло використовує тренувальні дані для створення нових даних, які мають чіткіші характеристики активностей. Потім ці дані подаються на вхід основної моделі, наприклад LSTM. Після тренування, LSTM модель готова обробляти нові дані активностей, щоб прогнозувати стани. Якщо із часом поведінка користувача змінюється і модель втрачає свою точність, вона автоматично перетреноується на нових даних, враховуючи збережену історію активностей для майбутнього удосконалення. Таке поєднання в системі забезпечує використання переваг обох методів, роблячи процес прогнозування більш точним та здатним адаптуватися до користувача (Рисунок 2.8).

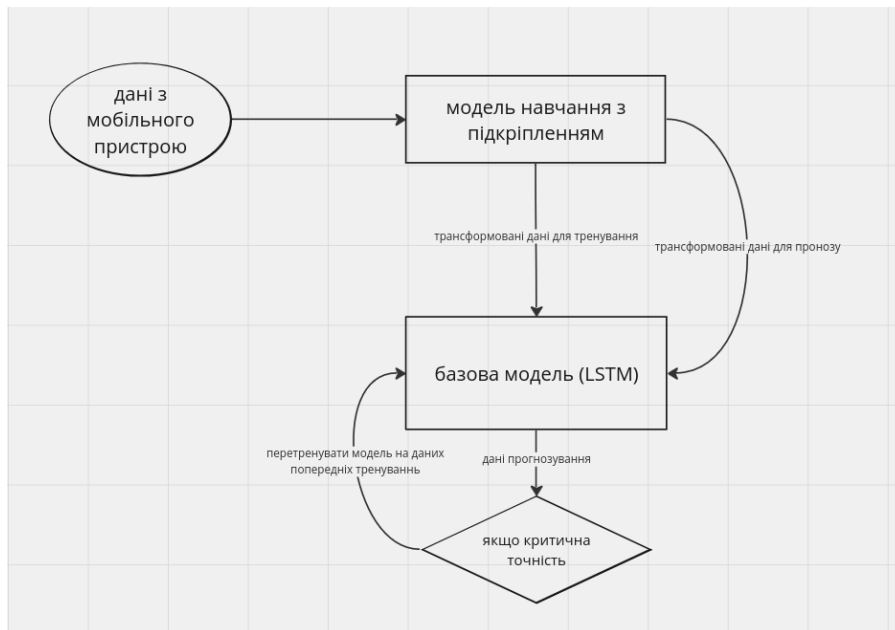


Рисунок 2.8. Схема роботи системи.

У кодї реалізовано клас `MCagent`, який є адаптивною системою для розпізнавання людської активності, використовуючи методи машинного навчання та навчання з підкріпленням. Ось як працює ця система:

Ініціалізація та Основні Властивості

- Клас `MCagent` ініціалізується з зовнішньою моделлю (наприклад, LSTM), з визначенням режиму виправлення моделі (`model_healing_mode`), який може бути "retrain" або інший.
- Встановлюється порогова точність (`threshold_accuracy`) та розмір навчальної вибірки (`trainable_size`).
- Створюються буфери для зберігання історії активностей та відповідних міток.

Обчислення Функції Цінності

- `compute_value_function` розраховує цінності станів на основі тренувальних даних, використовуючи евклідову відстань для оцінки схожості станів.

Прогнозування

- `predict` використовує розраховані цінності станів, щоб визначити найкращі дії для тестових даних.

Завантаження та Оцінка Нових Активностей

- `loadActivity` та `loadActivityCollection` використовуються для завантаження нових активностей і їх міток. Вони оцінюють точність моделі для цих нових даних та зберігають їх у історії.
- Якщо точність моделі падає нижче порогової точності, система автоматично переходить до перетренування (`retrain`) або вибірки (`resample`), залежно від режиму виправлення.

Перетренування Моделі

- `retrain` дозволяє системі перетренуватися на нових даних, коли точність падає. Це забезпечує адаптивність системи до змін у поведінці користувача або умовах даних.
- Під час перетренування система використовує збережену історію активностей та міток для створення нового тренувального та тестового наборів, на яких модель повторно тренується.

Цей підхід дозволяє системі динамічно адаптуватися до нових даних, забезпечуючи високу точність та актуальність моделі в довгостроковій перспективі. Завдяки постійному оновленню на основі нових даних, система здатна зберігати свою відповідність та ефективність у розпізнаванні активностей.

2.7 ВИСНОВКИ

Було розроблено передову систему для розпізнавання людської діяльності, що використовує методи навчання з підкріпленням, зокрема Монте-Карло, та супроводжується мобільним додатком для збору даних з датчиків смартфона. Система, написана на Python 3.8.10, та мобільний додаток, створений у Java через Android Studio, разом становлять комплексне рішення для ефективного розпізнавання та аналізу людської активності.

1. Збір та Обробка Даних

Початкові дані для тренування моделей були зібрані з дослідження, яке включало 30 добровольців, використовуючи смартфон Samsung Galaxy S II. Сигнали з акселерометра та гіроскопа були ретельно оброблені та використані для створення набору тренувальних та тестових даних.

2. Розробка Мобільного Додатку

Мобільний додаток було створено з метою підвищення адаптивності систем розпізнавання активностей. Додаток збирає індивідуальні дані активностей користувачів, що є критично важливим для точного та ефективного тренування моделей.

3. Використання LSTM та LGBM Моделей

Для класифікації активностей були обрані LSTM та LGBM моделі, які виявилися ефективними для задачі. Точність прогнозування цих моделей на початковому наборі даних була висока, що свідчить про їхню адекватність і ефективність у розпізнаванні різноманітних людських активностей.

4. Адаптивність та Ефективність Системи

Особливо важливою є адаптивність системи, яка досягається через використання алгоритму Монте-Карло в поєднанні з LSTM моделлю. Система здатна вчасно адаптуватися до змін у поведінці користувачів, автоматично

перетреновуючись на нових даних, що забезпечує її актуальність та високу точність у довгостроковій перспективі.

Розроблена система розпізнавання людської діяльності демонструє високу точність, адаптивність та ефективність. Використання передових технік навчання з підкріпленням разом із збором індивідуальних даних через мобільний додаток створює надійну основу для подальшого аналізу та розпізнавання людської активності у різних сценаріях. Таке поєднання підходів відкриває нові можливості для покращення якості досліджень та розробки в цій галузі.

РОЗДІЛ 3. ОПИС РЕЗУЛЬТАТІВ ТЕСТУВАННЯ СИСТЕМИ

Розроблена система є набором класів, методів та допоміжних функцій що дозволяють класифікувати людську активність та автоматично адаптуватися до зміни рухової поведінки.

3.1 Тестування при обраній базовій моделі LSTM

При ініціалізації системи вона навчається на початковому наборі даних. Навчання проходить у 5 епох, і сумарно тренується 79302 параметрів (Рисунок 3.1).

```
Epoch 1/5
97/97 [=====] - 13s 109ms/step - loss: 0.9960 - accuracy: 0.6388 - val_loss: 0.5872 - val_accuracy: 0.7841
Epoch 2/5
97/97 [=====] - 9s 97ms/step - loss: 0.5169 - accuracy: 0.8142 - val_loss: 0.3743 - val_accuracy: 0.8767
Epoch 3/5
97/97 [=====] - 10s 98ms/step - loss: 0.4208 - accuracy: 0.8521 - val_loss: 0.3778 - val_accuracy: 0.8706
Epoch 4/5
97/97 [=====] - 10s 99ms/step - loss: 0.4011 - accuracy: 0.8534 - val_loss: 0.3831 - val_accuracy: 0.8667
Epoch 5/5
97/97 [=====] - 10s 105ms/step - loss: 0.3610 - accuracy: 0.8738 - val_loss: 0.3042 - val_accuracy: 0.8971
init results
Model: "model_21"
```

Layer (type)	Output Shape	Param #
input_22 (InputLayer)	[(None, 128, 9)]	0
lstm_21 (LSTM)	(None, 128, 128)	70656
dense_42 (Dense)	(None, 128, 64)	8256
global_max_pooling1d_21 (GlobalMaxPooling1D)	(None, 64)	0
dense_43 (Dense)	(None, 6)	390
...		
Trainable params: 79302 (309.77 KB)		
Non-trainable params: 0 (0.00 Byte)		

Рисунок 3.1 Параметри моделі після початкового тренування.

Також проведемо аналіз тестування за графіками точності та функції втрат (Рисунок 3.2).

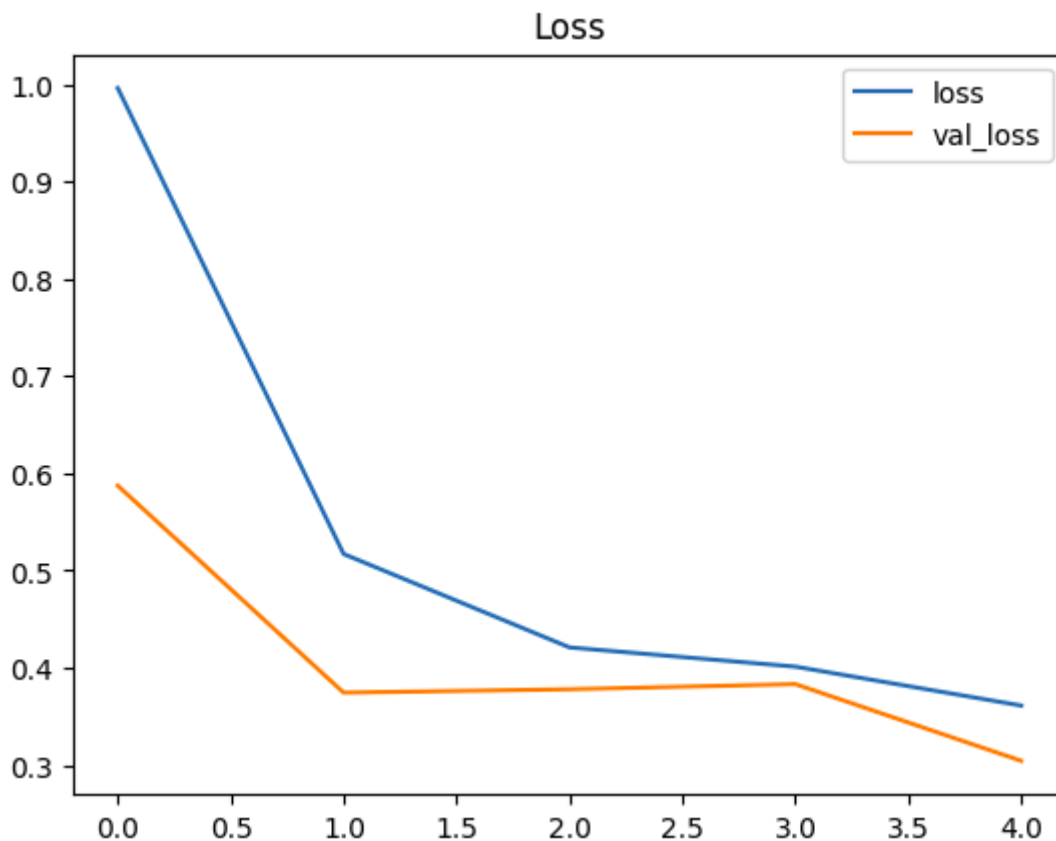


Рисунок 3.2 Графік функції втрат.

На графіку спостерігаємо значення функції втрат залежно від “епохи” тренування. На кожен наступний етап вона зменшувалась. Функція втрат мінімізувалася, її значення 0.36 на кінець тренування. Це показує що тренування пройшло успішно.

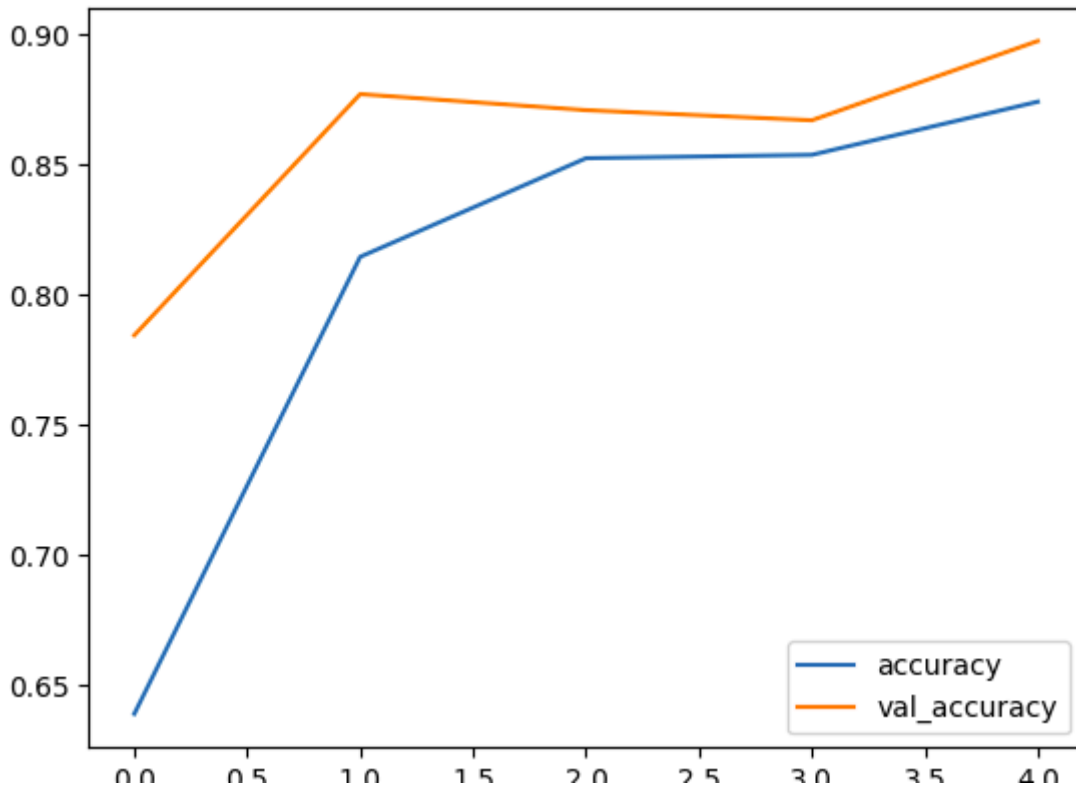


Рисунок 3.3 Графік точності моделі.

На графіку точності на рисунку 3.3 бачимо значення точності в залежності від “епохи”. Точність зростає з кожною наступною епохою. Нам вдалося досягти точності в 89% правильного визначення активності.

Далі розглянемо як поводитиме себе система при додаванні нових активностей користувачем. Ми згенерували активності. Пороговою точністю для початку перетренування стало 0.80 при наповненості в 60 активностей. Використаємо метод `loadActivityCollection()` класу `MCagent`. І передамо у нього масив активностей з мітками, тобто індексами назв активностей. При кожному розпізнаванні будемо виводити точність розпізнавання останніх ста активностей.

```

Model Accuracy: 0.8
1/1 [=====] - 0s 17ms/step
predicted activity is: [0]
1/1 [=====] - 0s 17ms/step
Model Accuracy: 0.8
1/1 [=====] - 0s 17ms/step
Model Accuracy: 0.79
Epoch 1/5
3/3 [=====] - 0s 75ms/step - loss: 0.6420 - accuracy: 0.7857 - val_loss: 0.1793 - val_accuracy: 0.9000
Epoch 2/5
3/3 [=====] - 0s 60ms/step - loss: 0.4561 - accuracy: 0.8571 - val_loss: 0.1788 - val_accuracy: 0.9667
Epoch 3/5
3/3 [=====] - 0s 62ms/step - loss: 0.4450 - accuracy: 0.8571 - val_loss: 0.2394 - val_accuracy: 0.9667
Epoch 4/5
3/3 [=====] - 0s 65ms/step - loss: 0.4299 - accuracy: 0.8143 - val_loss: 0.2492 - val_accuracy: 0.9667
Epoch 5/5
3/3 [=====] - 0s 64ms/step - loss: 0.3296 - accuracy: 0.9000 - val_loss: 0.2107 - val_accuracy: 0.9667
retrain results
Model: "model_21"

```

Рисунок. 3.4 Авто ретрейн при точності менше 0.8

```

Confusion matrix, without normalization
[[ 8  0  0  0  0]
 [ 0  4  0  0  0]
 [ 0  1  4  0  0]
 [ 0  0  0 11  0]
 [ 0  0  0  0  2]]

```

Рисунок 3.5 Матриця відповідностей моделі після ретрейну.

Як бачимо з рисунку 3.4 та рисунку 3.5, тестування системи показує, що після ретрейну точність з 0.79 виросла до 0.96. Продовжуючи тестування модель при деградації декілька раз перетреноувалась. Нижче наведемо приклад графіків функції втрат та точності.

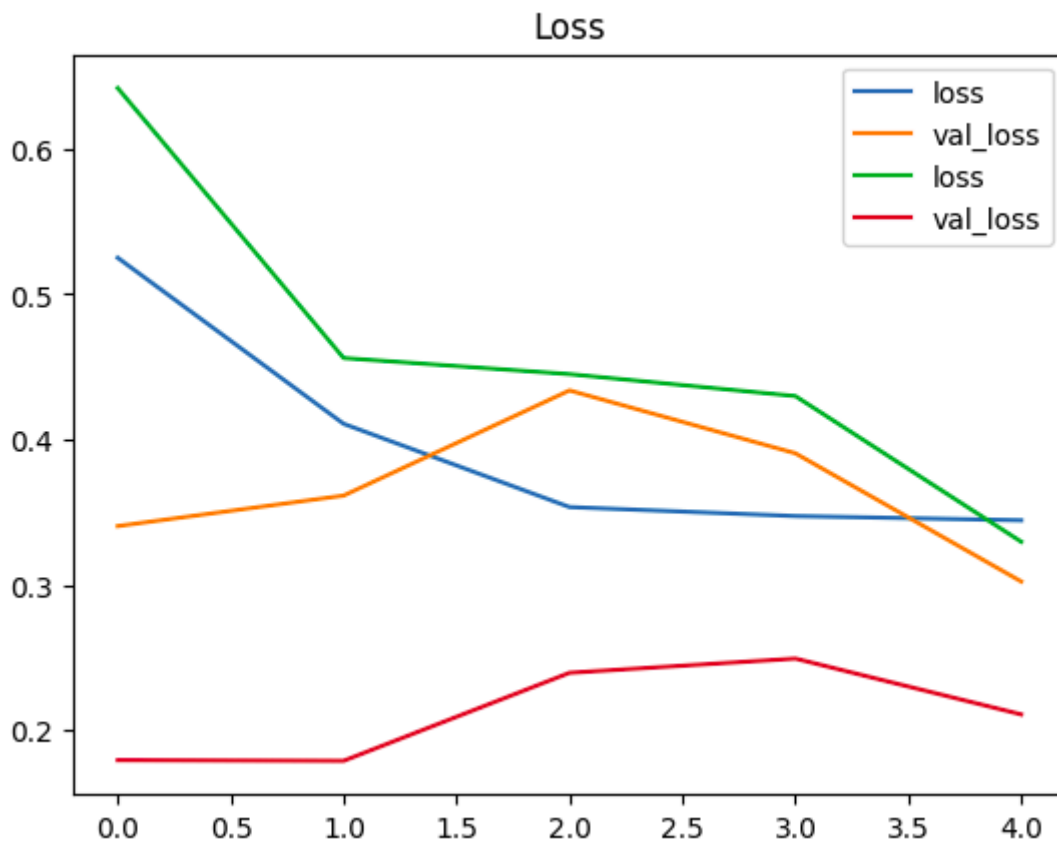


Рисунок 3.6 Графіки функції втрат при ретреїні моделі.

Як бачимо з рисунку 3.6, функція втрат кожного разу зменшується, що говорить про успішне навчання моделі.

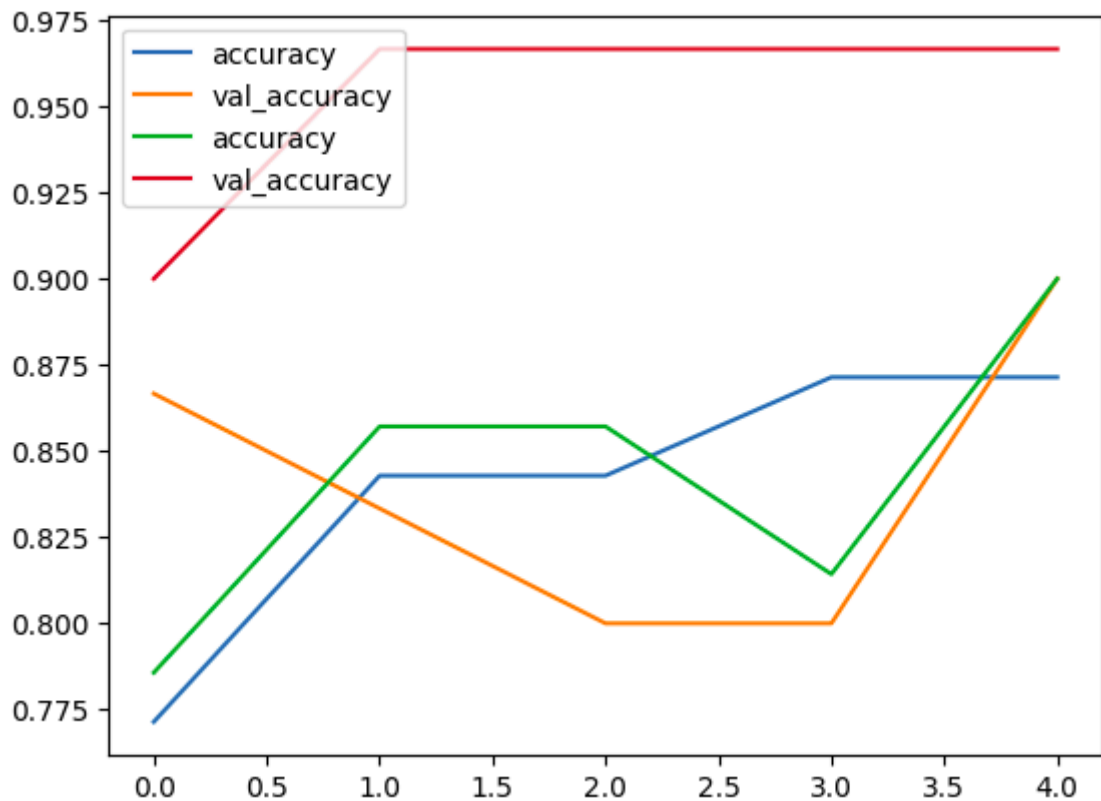


Рисунок 3.7 Графіки точності при ретрейнні моделі.

На рисунку 3.7 бачимо, що точність при кожному ретрейнні стабільно підвищувалась, що також доводить ефективність розробленого алгоритму.

3.2 Тестування при обраній базовій моделі LGBM

Також був проведений експеримент з базовою LGBM моделлю. Для сумісності з методами системи, що більше націлені під роботу з моделями keras, для нашої LGBM моделі був доданий клас обгортка з коректною імплементацією відповідних методів з keras моделей.

При початковому тренуванні точність моделі була 0.96 і матриця відповідностей показувала чудовий результат (Рисунок 3.8).

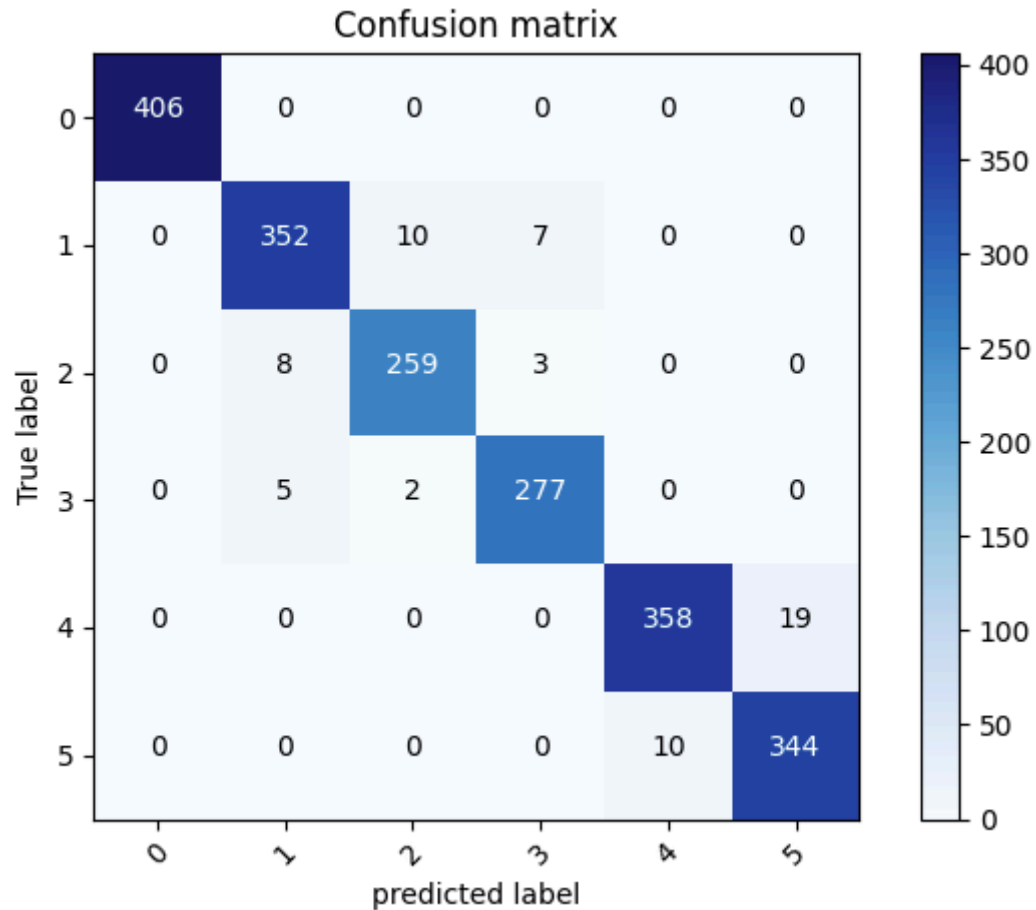


Рисунок 3.8 Матриця відповідностей для системи з LGBM

Далі розглянемо як поводитиме себе система при додаванні нових активностей користувачем. Ми згенерували активності. Пороговою точністю для початку перетренування стало 0.90 при наповненості в 60 активностей. Використаємо метод `loadActivityCollection()` класу `MCagent`. І передамо у нього масив активностей з мітками, тобто індексами назв активностей. При кожному розпізнаванні будемо виводити точність розпізнавання останніх ста активностей.

```

predicted activity is: [0]
Model Accuracy: 0.9
Model Accuracy: 0.9
Model Accuracy: 0.9
Model Accuracy: 0.89
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.000082 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1575
[LightGBM] [Info] Number of data points in the train set: 70, number of used features: 63
[LightGBM] [Info] Start training from score -34.538776
[LightGBM] [Info] Start training from score -1.415282
[LightGBM] [Info] Start training from score -1.415282
[LightGBM] [Info] Start training from score -1.609438
[LightGBM] [Info] Start training from score -1.945910
[LightGBM] [Info] Start training from score -1.763589
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf

```

Рисунок 3.9 Авторетрейн при точності менше 0.9

```

Confusion matrix, without normalization
[[ 3  0  0  0  0]
 [ 3  1  0  0  0]
 [ 2  0  3  0  0]
 [ 0  0  0 11  1]
 [ 0  0  0  2  4]]
/home/maksymtsybin/projects/diploma_work/MainU
f3.show()
Model Accuracy: 1.0
predicted activity is: [4]
predicted activity is: [4]
Model Accuracy: 1.0
predicted activity is: [4]
Model Accuracy: 1.0
Model Accuracy: 1.0
Model Accuracy: 1.0
Model Accuracy: 0.99
predicted activity is: [4]
Model Accuracy: 0.98
Model Accuracy: 0.98
Model Accuracy: 0.98

```

Рисунок 3.10 Матриця відповідностей після ретрейну.

Як бачимо з рисунка 3.9 та рисунка 3.10, тестування системи показує, що після ретрейну точність з 0.89 виросла до 0.99. Продовжуючи тестування модель при деградації декілька раз перетреноувалась.

Але кожного разу все швидше починала деградувати і згодом точність при перетренуванні почала падати. Нижче на рисунку 3.11 наведемо приклад матриці відповідностей при великій кількості перетреновань.

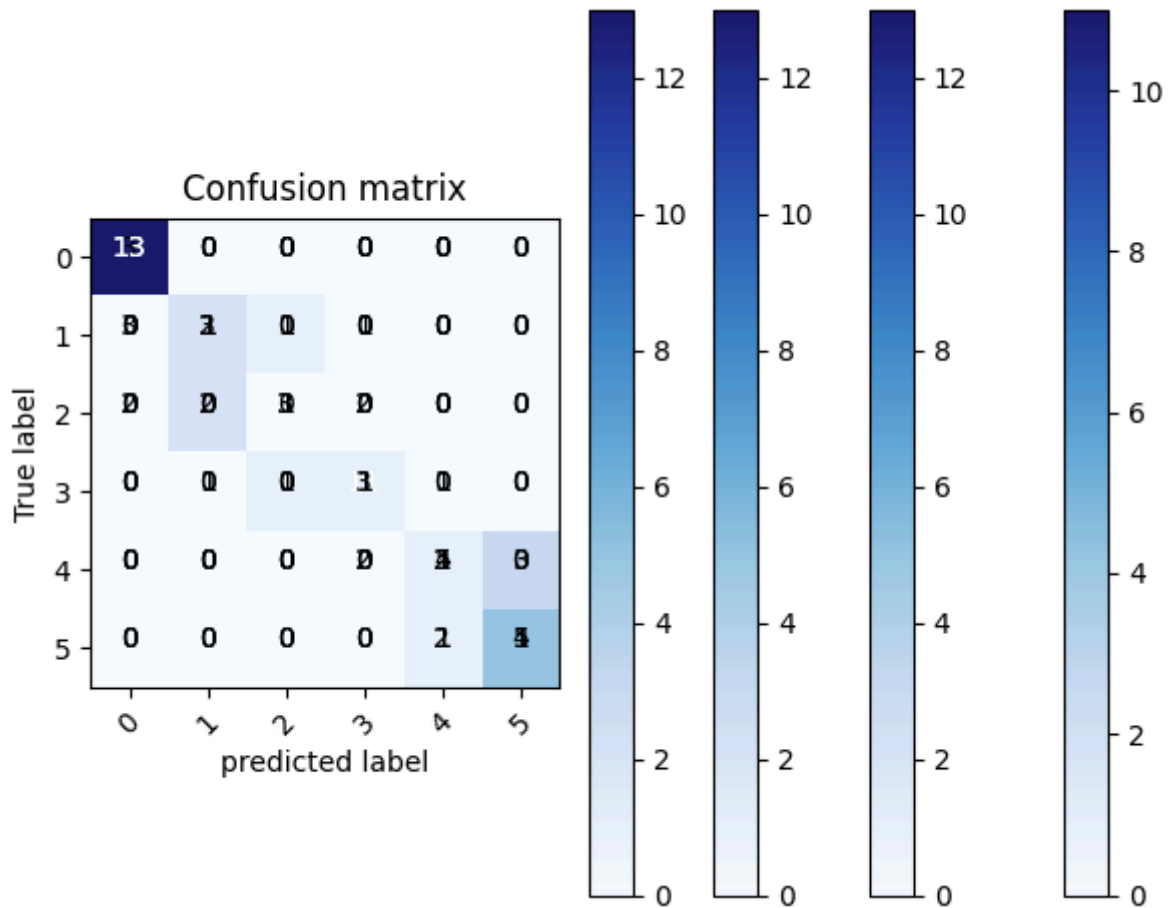


Рисунок 3.11 Матриці відповідностей при великій кількості перетреновань.

Нажаль точність моделі впала до 0.74, це може бути пов'язано з архітектурою LGBM моделі, або з неточно налаштованими гіперпараметрами.

3.3 ВИСНОВКИ

Ефективність LSTM моделі: Результати показують, що LSTM модель ефективно класифікує різні типи людської активності. Початкове навчання моделі привело до успішної мінімізації функції втрат до 0.36 та досягнення точності класифікації 89%. Це демонструє, що модель ефективно адаптується до навчальних даних.

Автоматичне перенавчання та адаптивність системи: Система показала високу адаптивність, успішно реагуючи на додавання нових активностей і автоматично перенавчаючись при падінні точності нижче порогового значення 0.80. Після ретрейну точність збільшилась до 0.96. Така динаміка свідчить про ефективність механізму перенавчання для підтримки актуальності моделі.

Тестування моделі LGBM: Експерименти з базовою LGBM моделлю, інтегрованою через спеціальний клас-обгортку, також показали високі результати. Початкова точність моделі склала 0.96. Система демонструвала гарні показники точності, навіть після додавання нових активностей і ретрейну. Така поведінка підтверджує гнучкість та ефективність використання LGBM в даній системі.

Проблеми зі стабільністю при частому перенавчанні: Незважаючи на первинно високі показники точності, було виявлено, що при частому перенавчанні LGBM моделі точність знижується, досягаючи 0.74. Це може свідчити про потенційні проблеми з архітектурою моделі або неоптимальні налаштування гіперпараметрів.

Рекомендації для подальших досліджень: Оптимізація гіперпараметрів та додаткові дослідження архітектури LGBM моделі можуть допомогти вирішити проблеми зі стабільністю моделі при частому перенавчанні. Також доцільно розглянути механізми для запобігання перенавчанню, які можуть підвищити загальну стабільність та надійність системи.

РОЗДІЛ 4. РОЗРОБКА СТАРТАП ПРОЄКТУ «Самонавчальні системи для розпізнавання людської діяльності Reinforcement learning for human activity recognition»

4.1 Опис ідеї проєкту

Розроблена система передбачає створення мобільного додатку для смартфонів, який здійснює автоматичний моніторинг фізичної активності користувача. Головною особливістю є його здатність адаптуватися до змін у руховій поведінці користувача. Цей додаток використовує технології навчання з підкріпленням для точного відстеження і аналізу рухової активності, а також для пропонування індивідуальних рекомендацій щодо підтримки фізичного здоров'я.

Головна мета цього розділу - показати процес реалізації цієї системи, а також провести маркетинговий аналіз, щоб оцінити ринкові можливості для використання цього додатку. Основні кроки включають:

1. Реалізація системи та додатку, описаних у попередніх розділах.
2. Розробка стратегії для виходу на ринок з конкурентоспроможним продуктом, та його подальший розвиток як стартапу.

Для повного розуміння сутності ідеї та ідентифікації ключових ринків, на яких слід зосередити пошук потенційних споживачів, важливо створити таблицю. У цій таблиці (табл. 4.1) має бути викладено детальний опис самої ідеї, різні можливі сфери застосування та основні переваги, які користувач може отримати від використання даного продукту.

Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення повноцінного мобільного додатку для автоматичного розпізнавання людської активності з високими показниками адаптації до змін рухової поведінки.	<p>Для індивідуальних користувачів: Додаток допомагає у підтримці здорового способу життя, відстежуючи фізичну активність та надаючи персоналізовані рекомендації.</p> <p>У сфері охорони здоров'я: Може бути використаний лікарями або фізіотерапевтами для моніторингу прогресу пацієнтів та адаптації програм реабілітації.</p> <p>У фітнес-індустрії: Фітнес-тренери можуть використовувати додаток для керування тренувальними планами своїх клієнтів та відстеження їхнього прогресу.</p>	<p>Здобувають здорові звички, підвищують фізичну активність, отримують персоналізовані вказівки для підвищення свого здоров'я.</p> <p>Лікарі мають точніші дані для оцінювання стану пацієнта та можуть краще керувати реабілітацією.</p> <p>Фітнес-тренери можуть більш ефективно відстежувати та налаштовувати тренування своїх клієнтів, що забезпечує кращі результати та підвищує задоволеність клієнтів.</p>

Цей проект має значний потенціал для різних груп користувачів. Використання технологій навчання з підкріпленням дозволяє індивідуалізувати підхід до кожного користувача, забезпечуючи високу адаптивність та в майбутньому персоналізовані рекомендації засновані на їхніх особистих даних.

Різноманітність напрямків застосування додатку, включаючи особисте використання для підтримки здорового способу життя, професійне використання у сфері охорони здоров'я та використання в фітнес-індустрії, розширює його цільову аудиторію. Це дозволяє додатку бути корисним як для індивідуальних користувачів, які прагнуть покращити своє здоров'я, так і для професіоналів, які можуть використовувати його для моніторингу та планування реабілітації пацієнтів чи тренувань клієнтів.

Основними вигодами для користувачів є підвищення якості здоров'я, точність в моніторингу та адаптації тренувань, а також зручність використання додатку. Таким чином, цей мобільний додаток має великий потенціал для залучення широкого спектру користувачів та може вважатися конкурентоспроможним та інноваційним продуктом на ринку мобільних додатків для здоров'я та фітнесу.

Складений перелік властивостей ідеї продукту, що включає як сильні, так і слабкі, а також нейтральні характеристики, служить основою для визначення його конкурентних переваг. У таблиці 4.2 представлено порівняння продукту з аналогічними на ринку виробами або їх заміниками, засноване на технічних та економічних характеристиках, що вказує на унікальні особливості та відмінності цього товару.

Визначення характеристик ідеї проекту

№ п/п	Техніко-ек ономічні характерис тики ідеї	(потенційні) товари/концепції конкурентів				W Слабка сторон а	N Нейтра льна сторон а	S Сильна сторон а
		Мій проект	Конкур ент 1	Конкур ент 2	Конкур ент 3			
1.	Форма виконання	Мобіль ний додато к	Додато к для смарт годинн ика	Додато к для смарт годинн ика	Мобіль ний додато к			+
2.	Собівартіс ть	Низька	Висока	Середн я	Середн я		+	
3.	Точність і адаптивніс ть розпізнава ння	Висока	Висока	Середн я	Низька			+
4.	Кросплатф орменість	Так	Ні	Ні	Ні			+
5.	Потреба в інтернеті	Ні	Ні	Ні	Ні			+

Сильними сторонами даного проекту, є його точність і адаптивність, також його форма виконання полягає у повноцінному кросплатформенному мобільному додатку, котрому не обов'язковий доступ до інтернету для того щоб користувач міг ним повноцінно користуватися. Ці сильні сторони неабияк

збільшують кількість повноцінних клієнтів, тому цей проект можна вважати конкурентоспроможним.

4.2 Технологічний аудит проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

Таблиця 4.3

Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології і реалізації	Наявність технологій	Доступність технологій
1.	Створення повноцінного мобільного додатку для автоматичного розпізнавання людської активності з високими показниками адаптації до змін рухової поведінки.	Tensorflow, Keras	Наявна	Безкоштовна, доступна
		Scikit learn	Наявна	Безкоштовна, доступна
		Light GBM	Наявна	Безкоштовна, доступна
		Android Studio	Наявна	Безкоштовна, доступна

Всі наведені в таблиці 4.3 технології мають бути використані у проекті. Вони доступні і зручні, мають величезну кількість інструментів для імплементації необхідних алгоритмів навчання з підкріпленням та методів машинного навчання. Тому проект можна вважати технологічно здійсненим.

4.3 Аналіз ринкових можливостей

Виявлення можливостей на ринку, які можуть бути використані для впровадження проекту, а також ідентифікація потенційних загроз, які можуть вплинути на його успішну реалізацію, є ключовими для планування стратегії розвитку проекту. Це передбачає врахування умов ринкового середовища, потреб потенційних споживачів та пропозицій конкуруючих проектів. Перший крок полягає у аналізі попиту, визначенні його наявності, обсягу та тенденцій розвитку ринку.

Таблиця 4.4

Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	100
2.	Загальний обсяг продаж. грн/ум.од.	1000
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу	Висока вартість початкового капіталу
5.	Специфічні вимоги до стандартизації та сертифікації	-
6.	Середня норма рентабельності в галузі. (або по ринку), %	$R = (3000000 * 100) / (1000000 * 12) = 25\%$

Таким чином, з таблиці 4.4 прибутковість даної галузі є нижчою за відсоткову ставку банківського депозиту. Це означає, що інвестування коштів у цей конкретний проект є обґрунтованим рішенням, оскільки проект не має спеціальних вимог до стандартизації чи сертифікації, адже він реалізується у вигляді програми для смартфона, доступної для завантаження на різних операційних системах.

Наступним кроком буде визначення потенційних груп клієнтів, їхніх особливостей та розробка приблизного списку вимог до продукту для кожної з цих груп. (табл 4.5)

Таблиця 4.5

Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	автоматичне розпізнавання людської активності з високими показниками адаптації до змін рухової поведінки.	а) Індивідуальні користувачі б) Лікарі в) Фітнес тренери	а) Націлені на швидку роботу додатку та точно визначення активностей впродовж дня б) Націлені на отримання дуже точних	а) Додаток: зручний у користуванні інтуїтивно зрозумілий інтерфейс, швидкодія, точність Постачальник додатку: Надає

			визначень активності і високу адаптивність до змін поведінки	технічну підтримку, покращує характеристик и розробленої системи
			в) Націлені на швидку роботу додатку та отримання дуже точних визначень активності і високу адаптивність до змін поведінки	

Були визначені ключові аспекти стартап-проекту: він задовольняє основну ринкову потребу автоматичному розпізнаванні людської активності з високими показниками адаптації до змін рухової поведінки.. Проект націлений на основні сегменти ринку, які включають індивідуальних користувачів, лікарів, фітнес тренерів. Відрізняючись у потребах і поведінці, ці цільові групи шукають швидке і точне визначення активності та високий рівень адаптивності. Додаток, розроблений у рамках проекту, має відповідати таким вимогам споживачів: бути швидким у роботі, зручним у використанні, з легко зрозумілим інтуїтивним інтерфейсом.

Наступним кроком є створення таблиць, в яких будуть визначені фактори, які підтримують введення проекту на ринок, а також фактори, які можуть стати перепорою для його ринкового запровадження. (табл. 4.6)

Таблиця 4.6

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Зростаюча вимогливість покупців	Споживачі в час технічних іновацій вимагають швидку роботу програми та високу точність прогнозування активностей	Пошук нових та модернізація старих методів для більш точного та швидкого визначення активності
2.	Зміна потреб і смаків покупців	Окрім служб підтримки/продажу користувачі можуть застосовувати додаток як частину більшої системи	Додавати більше функцій додатку під потреби користувачів
3.	Збільшення витрат на технічну підтримку	Невчасне реагування на сучасний ринок потреб користувачів	Вчасно оновлювати програмне забезпечення
4.	Конкуренція	Вихід на ринок великої компанії	Вихід з ринку, запропонувати великій компанії поглинути себе, передбачити додаткові переваги власного сервісу для того, щоб повідомити про них саме після виходу міжнародної компанії на ринок
5.	Зменшення	Зменшення зацікавленості	Постійні оновлення передових

	кількості замовників	замовників у наданих послугах	методів та підходів до визначення людської активності
--	-------------------------	----------------------------------	-------------------------------------------------------------

Були визначені ключові ризики для даного стартап-проекту. Однією з головних загроз є зміна у вимогах та потребах користувачів. Для мінімізації цих ризиків необхідно регулярно проводити дослідження для виявлення більш вдосконалених методів та алгоритмів розпізнавання людської діяльності, а також для поліпшення точності прогнозування та адаптивності. На даний момент, менш значущими ризиками є потенційне зростання витрат на технічну підтримку та можливе зниження числа користувачів-замовників.

Далі будуть визначені ключові характеристики конкурентного середовища на ринку. (табл. 4.7)

Таблиця 4.7

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентно спроможною)
1. Вказати тип конкуренції - олігополія	Існує мала кількість фірм конкурентів на ринку	Створення більш технологічно досконалих рішень ніж конкуренти
2. За рівнем конкурентної - інтернаціональна	Майже всі конкуренти - закордонні	Можна знайти більш дешевих дослідників.
3. За галузевою	Конкуренти мають сервіси	Добре описані рамки

ознакою - внутрішньогалузева	які використовуються лише всередині даної галузі	галузі та способи впливу на неї
4. Конкуренція за видами товарів - товарно-видова	Види товарів є однаковими, а саме - програмне забезпечення	Конкуренція між різними програмами розпізнавання діяльності
5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення додатку, щоб собівартість була нижчою	Поліпшення якості продукції
6. За інтенсивністю - не марочна	Бренди відсутні	Легше вийти на ринок молодій компанії

Було проведено детальний аналіз конкуренції на ринку, в рамках якого були визначені наступні аспекти: тип конкуренції як олігополія; міжнародний рівень конкурентної боротьби; внутрішньогалузева приналежність конкуренції; конкуренція в контексті товарних видів; конкуренція, заснована на нецінових конкурентних перевагах; і немаркована інтенсивність конкуренції. Також були окреслені потенційні стратегії для збереження конкурентоспроможності компанії, включаючи розробку більш технологічно вдосконалених рішень порівняно з конкурентами, залучення менш витратних дослідників, розвиток методів впливу на галузь, створення нових систем розпізнавання людської діяльності, та підвищення якості продукції.

Далі буде сформовано перелік факторів, що впливають на конкурентоспроможність на ринку, заснований на аналізі моделі п'яти сил Майкла Портера. (табл. 4.9)

Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Garmin Google Samsung Apple	Сильні розробники та сильний аналітичний відділ	Amazon, Google, Garmin	Rozetka, Prom	
Висновки	Дуже інтенсивна боротьба за першість в технологіях машинного навчання та нейронних мереж	Потенційно може вийти будь-яка компанія з необхідними ресурсами	Нічого не дійтують, мають чималі ресурси для створення власних додатків	Диктують умови, але Вузькопрофільно, так як це не їх першочергова сфера діяльності	

Враховуючи конкурентне середовище, можна з упевненістю стверджувати, що проект має всі шанси на успіх у ринковій ніші, оскільки серед існуючих конкурентів не знайдено таких, які б могли значно вплинути на його позиції. Проект вирізняється завдяки тому, що пропоноване рішення допомагає та спрощує вести здоровий образ життя, та покращувати свою фізичну активність. Серед ключових переваг продукту, які забезпечують йому конкурентні переваги на ринку, можна відзначити його надійність, точність та високу адаптуємість.

На підставі проведеного аналізу та висновків, сформовано і обґрунтовано перелік факторів, які сприяють конкурентоспроможності проекту (табл. 4.9).

Таблиця 4.9

Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентно спроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Науково-технічний потенціал	Кращі дослідження = вища ефективність роботи додатку
2.	Кадровий потенціал	Кращі кадри = нові, точні та швидкі методи розпізнавання діяльності

Був проведений порівняльний аналіз конкурентоспроможності між сильними сторонами продуктів конкурентів та продуктів нашої компанії. У цьому аналізі проект нашої компанії особливо відзначився за такими факторами, як науково-технічний та кадровий потенціали.

В наступному етапі буде проведено SWOT-аналіз для подальшої оцінки проекту (табл. 4.10).

Таблиця 4.10

SWOT- аналіз стартап-проекту

Сильні сторони: – найкращі та найдосвідченіші фахівці в області, найбільш ефективні алгоритми і методи оцінювання і прогнозування	Слабкі сторони: слабкий маркетинговий відділ, відсутність репутації, молода компанія
Можливості: стрімкий ріст ринку та технологічності рішень у сфері машинного навчання, а саме навчання з підкріпленням та розпізнавання людської діяльності.	Загрози: конкуренти значно більш відомі і потужніші, мають більше ресурсів, з часом здатні створити найкраще поєднання існуючих методів для більш ефективної роботи додатку

Виведення продукту на ринок, враховуючи внутрішні ресурси та спроможності компанії, має такі сильні та слабкі сторони: до сильних належать наявність висококваліфікованих і досвідчених спеціалістів у галузі, а також використання передових технологій машинного навчання та навчання з підкріпленням; серед слабких сторін варто відзначити недостатньо сильний маркетинговий відділ, відсутність репутації та недовгий час існування компанії. З точки зору ринкового та зовнішнього середовища, можливості та загрози компанії характеризуються наступним чином: можливості полягають у швидкому зростанні ринку та технологічних рішень у сфері машинного навчання, особливо у контексті розпізнавання людської діяльності; загрозами є наявність більш відомих і ресурсно сильних конкурентів, які з часом можуть розробити більш ефективні комбінації існуючих методів для покращення продуктивності додатків.

На основі SWOT-аналізу будуть розроблені альтернативні стратегії для ринкового впровадження стартап-проекту (табл. 4.11).

Таблиця 4.11

Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтований комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Покращуємо систему настільки, щоб залишити конкурентів далеко позаду	60%	1.5 роки
2.	Виходимо на ринок з існуючою системою розпізнаванні людської діяльності, граємо на швидкому рості ринку.	40%	8 місяців

Таким чином, було обрано стратегію ринкового впровадження стартап-проекту, яка полягає у значному вдосконаленні алгоритму для того, щоб випередити конкурентів. Ця стратегія вибрана через більшу ймовірність залучення необхідних ресурсів і те, що часові рамки її реалізації не значно відрізняються від інших можливих альтернатив.

4.4 Розробка ринкової стратегії проекту

При створенні ринкової стратегії, першочерговим завданням є розробка плану охоплення ринку, що включає ідентифікацію та опис цільових сегментів потенційних клієнтів (Табл. 4.12).

Таблиця 4.12

Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів прийняти продукт	Орієнтований попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Лікарі що займаються реабілітацією й потребують якісне та точне визначення активності пацієнта та функції швидкої адаптації під зміни рухової поведінки	Готові, якнайшвидше	Попит - високий	Не дуже інтенсивна	Середня
2.	Фітнес тренери що займаються покращенням здоров'я й потребують якісне та точне визначення активності клієнта та функції швидкої адаптації під зміни рухової поведінки	Готові, якнайшвидше	Попит - високий	Не дуже інтенсивна	Середня

Які цільові групи обрано: обрано обидві групи, так як вони майже не відрізняються, а проект їх може задовольнити

Отже, було вибрано основні цільові групи: лікарі що займаються реабілітацією, яким потрібно якісне та точне визначення активності пацієнта та функції швидкої адаптації під зміни рухової поведінки та фітнес тренери що займаються покращенням здоров'я й потребують якісне та точне визначення активності клієнта та функції швидкої адаптації під зміни рухової поведінки, адже обидві групи мають схожі вимоги та готові сприйняти продукт.

Далі визначається базова стратегія розвитку (табл. 4.13).

Таблиця 4.13

Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1.	Створення нових систем з розпізнавання людської діяльності на основі навчання з підкріпленням, що є новою і унікальною властивістю	Створення продуктивних іноваций та їх маркетинг	Основна позиція - ефективне розпізнавання та висока адаптуємість, адже є	Стратегія дифференціації

			ключовими технологіями	
--	--	--	---------------------------	--

Таким чином, було вирішено зосередитися на розвитку проекту шляхом створення нових систем з розпізнавання людської діяльності на основі навчання з підкріпленням, що є новою і унікальною властивістю. Визначена основна стратегія розвитку полягає у диференціації, реалізація якої буде здійснюватися через створення інноваційних продуктів та їхній маркетинг, враховуючи, що ефективне розпізнавання активності є важливою технологією та перевагою нашого проекту.

Наступний етап передбачає визначення стратегії конкурентної поведінки на ринку (Табл. 4.14).

Таблиця 4.14

Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проєкт першопрохідцем на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента і які?	Стратегія конкурентно ї поведінки*
1.	Розпізнавання людської діяльності за допомогою навчання з підкріпленням - так	Шукати нових і забирати існуючих	Стабільність, надійність сервісів, підтримка	Оборонна

			інтерфейсу	
--	--	--	------------	--

Таким чином, було обрано основну стратегію конкурентної поведінки, яка носить оборонний характер, оскільки здатність розпізнавання людської діяльності за допомогою навчання з підкріпленням становить унікальну особливість продукту. Додатково, планується забезпечення технічної підтримки та розробка зручного інтерфейсу, що сприятиме створенню довіри та лояльності серед споживачів.

Далі на черзі є розробка стратегії позиціонування проекту, яка допоможе потенційним користувачам легше ідентифікувати і виокремити наш програмний продукт серед інших (табл. 4.15).

Таблиця 4.15

Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Коректна робота, надійність і підтримка	Стратегія диференціації	Основна позиція - розпізнавання людської діяльності з використанням навчання з	Швидкодія, адаптуємість, безпека

			підкріпленням, адже є ключовою технологією і перевагою нашого проєкту	
--	--	--	-----------------------------------------------------------------------------------	--

В результаті була розроблена стратегія позиціонування, яка включає визначення основних вимог цільової аудиторії до продукту: Коректна робота, надійність і підтримка. Основна стратегія розвитку обрана як диференціація, а ключові конкурентні переваги стартап-проєкту включають розпізнавання людської діяльності з використанням навчання з підкріпленням, що є вирішальною технологією та перевагою проєкту.

Також була розроблена комплексна позиція проєкту, яка акцентує на швидкодії, безпеці та адаптуємості.

4.5 Розробка маркетингової програми

Початковим етапом є розробка маркетингової стратегії продукту, яка буде представлена споживачам. В цьому контексті, в таблиці 4.18 необхідно агрегувати результати проведеного аналізу конкурентоспроможності продукту.

Таблиця 4.16

Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі що потрібно створити)
1.	Автоматизація	Не потрібно купувати	Використання навчання

	певного бізнес-процесу: ефективне розпізнавання людської діяльності	дорогі датчики і підлаштовувати їх під конкретну людину.	з підкріпленням для розпізнавання людської діяльності збільшує адаптуємість системи під зміни рухової поведінки.
--	---------------------------------------------------------------------------------	----------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------

Надалі розробляється трирівнева маркетингова модель товару (табл. 4.17).

Таблиця 4.17

Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом			
II. Товар у реальному виконанні	Властивості/характеристи ки	М/НМ	Вр/Тх/Т л/Е/Ор
	1. Надійність	М	Тл
	2. Точність	М	Тл
	прогнозування	М	Тл
	3. Адаптуємість		
	Відповідає очікуваним нормам розпізнавання, протестовано на наборі даних реальних людей		
	Повноцінний мобільний додаток з відкритим інтерфейсом		
	HARmony		
III. Товар з підкріпленням	Акції, знижки		
	Підтримка		
За рахунок чого потенційний товар буде захищено від копіювання: патент.			

Таким чином, було розглянуто три рівні моделі продукту: продукт буде представлений у формі додатку, який спрощуватиме бізнес-процеси користувачів, зокрема, забезпечуватиме ефективне розпізнавання людської діяльності.

Основні атрибути продукту включають ефективність розпізнавання та адаптуємість, високий рівень надійності та безпеки даних. Технологія продукту буде захищена від незаконного копіювання за допомогою патенту.

Наступним етапом є встановлення цінових рамок (згідно з таблицею 4.18), які будуть визначені на основі експертного аналізу.

Таблиця 4.18

Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення цін на товар/послугу
1.	125 грн. / 1000 викликів	250 грн. / 1000 викликів	> 20000 грн/місяць	275-500грн. / 1000 викликів

Були встановлені параметри ціноутворення для мобільного додатку, зокрема: ціновий рівень для товарів-замінників складає 125 грн за 1000 викликів програмного інтерфейсу; ціни на аналогічні товари становлять 250 грн; рівень доходів цільової аудиторії - 20000 грн. Встановлені верхня та нижня

межі ціни на продукт від 275 до 500 грн. Цей аналіз був здійснений за допомогою експертного методу.

Далі буде визначена оптимальна система збуту, в рамках якої буде прийнято відповідне рішення (табл. 4.19).

Таблиця 4.19

Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Клієнти зазвичай оціняють можливості платформи через безкоштовну пробну версію, а потім почнуть платити за кожні 1000 викликів сервісу протягом довгого часу	Фінансування витрат на функціонування каналу збуту, фінансування збутових операцій, обслуговування проданих товарів.	Канал нульового рівня	Проводити збут власними силами

Таким чином, була розроблена система збуту, яка передбачає щомісячну оплату за певну кількість доступу до сервісу, включаючи користувацьку підтримку. Продажі будуть здійснюватися безпосередньо через власні ресурси компанії.

Наступним етапом є створення концепції маркетингових комунікацій (табл. 4.20).

Таблиця 4.20

Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікації якими користуються цільові клієнти	Ключові позиції обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Це Молоді люди, провідні лікарі, фітнес тернери, які надають перевагу новітнім технологіям	E-mail, Таргетинг, контекста реклама конференції	Унікальна особливість - ефективне розпізнавання активності з високою адаптуемості під зміни рухової поведінки	Привернути максимум уваги до продукту, змусити цільову аудиторію спробувати продукт	Користувачі прагнуть покращити своє здоров'я, та свою ефективність в роботі з покращення стану клієнта

В результаті була створена концепція маркетингових комунікацій, де основні напрямки позиціонування включають ефективність розпізнавання людської діяльності з високим рівнем адаптуемості. Головна мета рекламного

повідомлення полягає в приверненні максимальної уваги до продукту і заохоченні користувачів до його випробування, акцентуючи на знаходженні балансу між потребами компаній та їхніх користувачів у рекламному меседжі.

4.6 Висновки до розділу

У цьому розділі було розглянуто ключові аспекти розробки та впровадження стартап-проекту, що передбачає створення мобільного додатку для моніторингу та аналізу фізичної активності користувачів. Основною інновацією проекту є використання технологій навчання з підкріпленням, які дозволяють додатку адаптуватися до змін у руховій поведінці користувача, забезпечуючи точне відстеження та персоналізовані рекомендації.

Аналіз ринкових можливостей виявив значний потенціал проекту серед різних груп користувачів, включаючи індивідуальних користувачів, медичних працівників, та фітнес-тренерів. Створення детального опису ідеї, потенційних напрямків застосування, та переваг для кінцевого користувача підкреслює універсальність та цінність додатку.

Технологічний аудит підтвердив доступність необхідних технологій та інструментів для реалізації ідеї. Використання вільно доступних технологій, таких як Tensorflow, Keras, Scikit learn, Light GBM та Android Studio, робить проект технологічно здійсненим.

Проведений SWOT-аналіз виявив сильні сторони проекту, такі як висока кваліфікація команди та ефективність запропонованих рішень, а також слабкі сторони, включаючи слабкий маркетинговий відділ та відсутність репутації. Оцінка можливостей та загроз на ринку дозволила визначити ключові напрямки для подальшого розвитку та позиціонування проекту.

Загалом, проект демонструє високий рівень конкурентоспроможності та інноваційного потенціалу завдяки своїй унікальній технологічній основі та адаптивності до потреб різних користувачів. Враховуючи всі викладені аспекти,

проект має всі перспективи для успішного впровадження на ринку та задоволення вимог цільової аудиторії.

Отже відповідно до проведених досліджень:

- Додаток можна вивести на ринок та комерціалізувати його роботу;
- Існують перспективи також враховуючи групи клієнтів, які будуть ним користуватися, так як бар'єр не є високим, а також проект має декілька власних переваг;
- Необхідно реалізувати повноцінну систему розпізнавання людської діяльності з зручним користувацьким інтерфейсом для мобільних пристроїв;
- Подальша розробка є доцільною.

ВИСНОВКИ

У дипломній роботі були досліджені інновації в галузі розпізнавання людської активності (HAR) з використанням навчання з підкріпленням з точки зору розвитку технологій, додатків, алгоритмічних методів та потенційного впливу на різні сфери життя. Значний акцент зроблено на використанні передових методів машинного навчання, таких як дерева рішень, SVM, RNN, а особливо LSTM та LightGBM в поєднанні з reinforcement learning алгоритмами, що дозволяють точно відстежувати та аналізувати людську поведінку. Було досліджено методи навчання з підкріпленням, зокрема Монте-Карло та DDPG, які відкривають нові можливості для оптимізації алгоритмів у динамічних середовищах.

У контексті практичного застосування, робота підкреслює значущість HAR у сферах особистого здоров'я та фітнесу, де вони допомагають створювати більш персоналізовані та профілактичні рішення. Також було розглянуто вплив HAR на безпеку та догляд за літніми людьми, зокрема, можливості виявлення надзвичайних ситуацій, таких як падіння або раптові зміни здоров'я. Робота також торкається інновацій в жестових керуваннях та контекстно-чутливих застосунках, що може підвищити загальний досвід користувача та доступність.

Детально описано розробку комплексного рішення для ефективного розпізнавання та аналізу людської активності, що включає систему з використанням навчання з підкріпленням на Python 3.8 та мобільний додаток, створений у Java через Android Studio. Також зосереджено увагу на зборі та обробці даних з датчиків смартфона, розробці системи з використанням навчання з підкріпленням для підвищення адаптивності системи, а також на ефективності LSTM та LGBM моделей для класифікації активностей.

Робота охоплює аналіз адаптивності та ефективності системи, включаючи використання алгоритму навчання з підкріпленням Монте-Карло у поєднанні з

LSTM моделлю для забезпечення актуальності та точності системи в довгостроковій перспективі. Вказана проблема зі стабільністю при частому перенавчанні LGBM моделі та рекомендовані подальші дослідження для оптимізації гіперпараметрів та архітектури моделі.

Окрім технічних аспектів, дипломна робота включає глибокий аналіз ринкових можливостей стартап-проекту. Визначено ключові напрямки розвитку, потенційних користувачів, а також проведено SWOT-аналіз для оцінки сильних і слабких сторін проєкту. Особливу увагу приділено ефективності запропонованих рішень, а також можливим загрозам і викликам на ринку.

Завершуючи, акцентовано на важливості етичних та конфіденційних зобов'язань, які супроводжують розробку та впровадження HAR технологій. Зазначено, що у міру розвитку та впровадження цих технологій, важливо зосередитись на забезпеченні конфіденційності даних, усуненні упередженості та розвитку етичних практик у галузі штучного інтелекту. Це має критичне значення для формування майбутнього, де такі технології будуть широко впроваджені та приносять користь суспільству.

Загалом, в роботі продемонстровано положення сучасних тенденцій та викликів у галузі HAR, та зроблено внесок у цю швидко розвиваючу область. Відображено не тільки технічний потенціал HAR з використанням навчання з підкріпленням, але й його широкомасштабні соціальні, етичні та комерційні аспекти.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. randomHAR: Improving Ensemble Deep Learners for Human Activity Recognition with Sensor Selection and Reinforcement Learning. [Електронний ресурс]. URL: <https://arxiv.org/pdf/2307.07770.pdf> (дата звернення:: 05.12.23).
2. Kotsiantis, Sotiris & Kanellopoulos, Dimitris & Pintelas, P. (2006). Data Preprocessing for Supervised Learning. International Journal of Computer Science. 1. 111-117.
3. Akash Takyar: A comprehensive exploration of various machine learning techniques. [Електронний ресурс]. URL: <https://www.leewayhertz.com/machine-learning-techniques/> (дата звернення:: 10.12.23).
4. Saloni G. Decision Tree [Електронний ресурс] / Gupta Saloni. – 2022. – [Електронний ресурс]. URL: <https://www.geeksforgeeks.org/decision-tree/>. (дата звернення:: 05.12.23).
5. Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, Asdrubal Lopez, A comprehensive survey on support vector machine classification: Applications, challenges and trends, Neurocomputing, Volume 408, 2020, Pages 189-215, ISSN 0925-2312, <https://doi.org/10.1016/j.neucom.2019.10.118>.
6. Alex Sherstinsky, Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network, Physica D: Nonlinear Phenomena, Volume 404, 2020, 132306, ISSN 0167-2789, <https://doi.org/10.1016/j.physd.2019.132306>.
7. Benjamin Lindemann, Timo Müller, Hannes Vietz, Nasser Jazdi, Michael Weyrich, A survey on long short-term memory networks for time series prediction, Procedia CIRP, Volume 99, 2021, Pages 650-655, ISSN 2212-8271, <https://doi.org/10.1016/j.procir.2021.03.088>.

8. Debasmita Mishra, Bighnaraj Naik, Janmenjoy Nayak, Alireza Souri, Pandit Byomakesha Dash, S. Vimal, Light gradient boosting machine with optimized hyperparameters for identification of malicious access in IoT network, *Digital Communications and Networks*, Volume 9, Issue 1, 2023, Pages 125-137, ISSN 2352-8648, <https://doi.org/10.1016/j.dcan.2022.10.004>.
9. Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.
10. Sewak, Mohit. (2019). *Deterministic Policy Gradient and the DDPG: Deterministic-Policy-Gradient-Based Approaches*. 10.1007/978-981-13-8285-7_13.
11. Strączkiewicz, Marcin & James, Peter & Onnela, Jukka-Pekka. (2021). A systematic review of smartphone-based human activity recognition for health research.
12. Francesco Audrino & Simon D. Knaus (2016) Lassoing the HAR Model: A Model Selection Perspective on Realized Volatility Dynamics, *Econometric Reviews*, 35:8-10, 1485-1521, DOI: 10.1080/07474938.2015.1092801
13. Kekevi, Uğur & Aydin, Ahmet. (2022). *Real-Time Big Data Processing and Analytics: Concepts, Technologies, and Domains*. 7. 111-123. 10.53070/bbd.1204112.
14. Tong Zhang, Yikai Li, C.L. Philip Chen, *Edge computing and its role in Industrial Internet: Methodologies, applications, and future directions*, *Information Sciences*, Volume 557, 2021, Pages 34-65, ISSN 0020-0255, <https://doi.org/10.1016/j.ins.2020.12.021>.

15. Самонавчальна система розпізнавання людської діяльності з використанням навчання з підкріпленням /Цибін М. Д., Кислий Р. В. // Системні науки та інформатика: збірник доповідей II науково-практичної конференції «Системні науки та інформатика», 4–8 грудня 2023 року, Київ. – К., НН ІІСА КПІ ім. Ігоря Сікорського, 2023. – с. 382 - 387

ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

data_preprocessing.py

```

from matplotlib import pyplot as plt
import pandas as pd
import numpy as np
import math
import h5py
import os

activityIDdict = {
    1: 'walking',
    2: 'walking_upstairs',
    3: 'walking_downstairs',
    4: 'sitting',
    5: 'standing',
    0: 'laying',
}

colNames = ['body_acc_x', 'body_acc_y', 'body_acc_z',
            'body_gryo_x', 'body_gryo_y', 'body_gryo_z',
            'total_acc_x', 'total_acc_y', 'total_acc_z']

def read_files():
    list_of_Xs = ['./UCI HAR Dataset/test/Inertial Signals',
                 './UCI HAR Dataset/train/Inertial Signals']

    list_of_ys = ['./UCI HAR Dataset/test/y_test.txt',
                  './UCI HAR Dataset/train/y_train.txt']

    train_X_array = []
    files = os.listdir(list_of_Xs[1])
    for file in files:
        print(file, " is reading...")
        data = np.loadtxt(list_of_Xs[1]+'/'+file)
        train_X_array.append(data)

    test_X_array = []
    files = os.listdir(list_of_Xs[0])
    for file in files:
        print(file, " is reading...")
        data = np.loadtxt(list_of_Xs[0]+'/'+file)
        test_X_array.append(data)

```

```

train_X = np.dstack(train_X_array)
print(train_X.shape)

test_X = np.dstack(test_X_array)
print(test_X.shape)

train_y = np.loadtxt(list_of_ys[1])
test_y = np.loadtxt(list_of_ys[0])
print(train_y.shape)
print(test_y.shape)

X = np.vstack((train_X, test_X))
y = np.hstack((train_y, test_y)).astype(np.int64)
y = np.where(y==6, 0, y)

print("X shape is ", X.shape)
print("y shape is ", y.shape)
print(set(y))
return [X, y]

def save_data(arr, file_name):
    dict_ = {'inputs' : arr[0], 'labels': arr[1]}
    f = h5py.File(file_name, 'w')
    for key in dict_:
        print(key)
        f.create_dataset(key, data = dict_[key])
    f.close()
    print('Done. ')

def window_plot(X, y, col, y_index):
    unit='ms^-2'
    x_seq = X[y_index][:, col]
    plottitle = colNames[col]+' - '+ activityIDdict[y[y_index]]
    plt.plot(x_seq)
    plt.title(plottitle)
    plt.xlabel('window')
    plt.ylabel(unit)
    plt.show()

if __name__ == "__main__":
    file_name = 'uci_har.h5'
    arr = read_files()

```

```

window_plot(arr[0],arr[1],0,500)
save_data(arr,file_name)

```

MainUtils.py

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import tensorflow as tf
from sklearn import metrics
import h5py
import matplotlib.pyplot as plt
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import Input, Conv2D, Dense, Flatten, Dropout,
SimpleRNN, GRU, LSTM,
GlobalMaxPooling1D, GlobalMaxPooling2D, MaxPooling2D, BatchNormalization
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import itertools
import lightgbm as lgb
from sklearn.metrics import accuracy_score
from collections import defaultdict

def read_h5(path):
    f = h5py.File(path, 'r')
    X = f.get('inputs')
    y = f.get('labels')
    print(type(X))
    print(type(y))
    X = np.array(X)
    y = np.array(y)

    return X, y

def draw(r):
    f1 = plt.figure(1)
    plt.title('Loss')
    plt.plot(r.history['loss'], label = 'loss')
    plt.plot(r.history['val_loss'], label = 'val_loss')
    plt.legend()
    f1.show()

```

```

f2 = plt.figure(2)
plt.plot(r.history['accuracy'], label = 'accuracy')
plt.plot(r.history['val_accuracy'], label = 'val_accuracy')
plt.legend()
f2.show()

def con_matrix(model, x_test, y_train, y_test):
    K = len(set(y_train))
    y_pred = model.predict(x_test).argmax(axis=1)
    cm = confusion_matrix(y_test, y_pred)
    plot_confusion_matrix(cm, list(range(K)))

def plot_confusion_matrix(cm, classes, normalize = False, title='Confusion
matrix', cmap=plt.cm.Blues):
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print("Confusion matrix, without normalization")
    print(cm)
    f3 = plt.figure(3)
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max()/2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                horizontalalignment = "center",
                color = "white" if cm[i, j] > thresh else "black")
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('predicted label')
    f3.show()

```

har_rl.ipynb

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import tensorflow as tf

```

```

from sklearn import metrics
import h5py
import matplotlib.pyplot as plt
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import Input, Conv2D, Dense, Flatten, Dropout,
SimpleRNN, GRU, LSTM,
GlobalMaxPooling1D, GlobalMaxPooling2D, MaxPooling2D, BatchNormalization
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import itertools
import lightgbm as lgb
from sklearn.metrics import accuracy_score
from collections import defaultdict
from sklearn.metrics.pairwise import euclidean_distances
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Input, Concatenate
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import backend as K

from MainUtils import *

class LGBMWrapper(Model):
    def __init__(self, x_train, y_train, x_test, y_test, params):
        super(LGBMWrapper, self).__init__()
        self.params = params
        self.lgbm = None
        self.predicted_classes = None

    def fit(self, x_train, y_train, validation_data, epochs = 5, batch_size =
32):
        train_data = lgb.Dataset(x_train, label=y_train)
        test_data = lgb.Dataset(validation_data[0], label=validation_data[1],
reference=train_data)
        self.lgbm = lgb.train(self.params, train_data,
valid_sets=[test_data])
        self.y_test = validation_data[1]

    def predict(self, X):
        y_pred = self.lgbm.predict(X, num_iteration=self.lgbm.best_iteration)
        self.predicted_classes = [np.argmax(pred) for pred in y_pred]

```

```

return y_pred

def summary(self):
    accuracy = accuracy_score(self.y_test, self.predicted_classes)
    print(f"Model Accuracy: {accuracy}")

class ModelAgent():
    def __init__(self, model_name, X, Y):
        self.model_name = model_name

        if model_name == "lstm":
            x_train, x_test, y_train, y_test = train_test_split(X, Y,
test_size=0.3, random_state = 1)
            self.x_train = x_train
            self.y_train = y_train
            self.x_test = x_test
            self.y_test = y_test
            self.lstm_model()

        if model_name == "lgbm":
            X_features = np.array([self.extract_features(activity) for
activity in X])
            X_features = X_features.reshape(X_features.shape[0], -1)
            x_train, x_test, y_train, y_test = train_test_split(X_features,
Y, test_size=0.2, random_state=42)
            self.x_train = x_train
            self.y_train = y_train
            self.x_test = x_test
            self.y_test = y_test
            self.model = self.lgbm_model()
        else:
            self.lstm_model()

    def lstm_model(self):
        K = len(set(self.y_train))
        i = Input(shape = self.x_train[0].shape)
        x = LSTM(128, return_sequences=True)(i)
        x = Dense(64,activation = 'relu')(x)
        x = GlobalMaxPooling1D()(x)
        x = Dense(K,activation = 'softmax')(x)
        model = Model(i,x)
        model.compile(optimizer = Adam(learning_rate=0.001),

```

```

        loss = 'sparse_categorical_crossentropy',
        metrics = ['accuracy'])
self.model=model
return model

def lgbm_model(self):
    params = {
        'objective': 'multiclass',
        'num_class': 6,
        'boosting_type': 'gbdt',
        'metric': 'multi_logloss',
        'learning_rate': 0.05
    }

    model = LGBMWrapper(self.x_train, self.y_train, self.x_test,
self.y_test, params)

    return model

def extract_features(self, window):
    return [
        np.mean(window, axis=0),
        np.std(window, axis=0),
        np.min(window, axis=0),
        np.max(window, axis=0),
        np.median(window, axis=0),
        np.percentile(window, 25, axis=0),
        np.percentile(window, 75, axis=0)
    ]

class BaseRlAgent:
    def __init__(self, modelAgent):
        self.modelAgent = modelAgent

class MCagent(BaseRlAgent):
    def __init__(self, modelAgent, model_healing_mode="retrain"):
        BaseRlAgent.__init__(self, modelAgent)
        self.model_healing_mode = model_healing_mode
        self.threshold_accuracy = 0.90
        self.trainable_size = 60
        self.history_activities = np.zeros(shape=(100, 63,))
        self.history_predictions = np.zeros(shape=(100, ))
        self.history_labels = np.zeros(shape=(100, ))
        self.history_counter = 0

```

```

self.V = np.zeros((len(self.modelAgent.x_train),))
self.compute_value_function(self.modelAgent.x_train)
self.predict(self.modelAgent.x_train, self.modelAgent.x_test)
# self.r =
    self.modelAgent.model.fit(self.predictions, self.modelAgent.y_test,
validation_data = (self.modelAgent.x_test, self.modelAgent.y_test), epochs =
5, batch_size = 32 )
    print("init results")
    self.printModelStats(self.modelAgent.x_test, self.modelAgent.y_train,
self.modelAgent.y_test)

@staticmethod
def extract_features(sequence):
    return sequence.flatten()

def find_similar(self, state_features, x_train):
    dataset_resaped = x_train.reshape(len(x_train), -1)
        distances = euclidean_distances(dataset_resaped,
state_features.reshape(1, -1))
    return np.argmin(distances)

def compute_value_function(self, x_train):
    for i in range(len(x_train)):
        # if i % 100 == 0:
            #         print(f"{i / len(self.modelAgent.x_train) * 100}%
completed")
        state_features = self.extract_features(x_train[i])
        similar_idx = self.find_similar(state_features, x_train)
        reward = -euclidean_distances(x_train[i].reshape(1, -1),
x_train[similar_idx].reshape(1, -1))
        self.V[i] += reward

self.V -= np.min(self.V)
self.V /= np.max(self.V)

def predict(self, x_train, x_test):
    predictions = []
    for i in range(len(x_test)):
        state_features = self.extract_features(x_test[i])
        similar_idx = self.find_similar(state_features, x_train)
        predictions.append(x_train[similar_idx])
    self.predictions = np.array(predictions)

```

```

return self.predictions

def printModelStats(self, x_test, y_train, y_test):
    self.modelAgent.model.predict(x_test)
    print(self.modelAgent.model.summary())
    # draw(self.r)
    con_matrix(self.modelAgent.model, x_test, y_train, y_test)

def loadActivity(self, activity, label=None):
    self.last_activity = activity
    last_prediction =
self.modelAgent.model.predict(self.last_activity.reshape(1,
63)).argmax(axis=1)
    if label:

        self.history_activities[self.history_counter]=self.last_activity
        self.history_predictions[self.history_counter]=last_prediction
        self.history_labels[self.history_counter]= label
        accuracy = accuracy_score(self.history_labels,
self.history_predictions)
        print(f"Model Accuracy: {accuracy}")

        self.history_counter = (self.history_counter + 1) % 100
        if(accuracy < self.threshold_accuracy and self.history_counter
>= self.trainable_size):
            if(self.model_healing_mode == "resample"):
                self.resample()
            else:
                self.retrain()
                self.history_activities = np.zeros(shape=(100, 63,))
                self.history_predictions = np.zeros(shape=(100, ))
                self.history_labels = np.zeros(shape=(100, ))
                self.history_counter = 0

    else:
        print("predicted activity is:", last_prediction)

def loadActivityCollection(self, activityCollection, labels=None):
    if labels.any():
        for activity, label in zip(activityCollection, labels):
            self.loadActivity(activity, label)
    else:
        for activity in activityCollection:

```

```

        self.loadActivity(activity)

    def retrain(self):
        x_train, x_test, y_train, y_test =
train_test_split(self.history_activities, self.history_labels,
test_size=0.3, random_state = 1)
        self.V = np.zeros((len(x_train),))
        self.compute_value_function(x_train)
        self.predict(x_train, x_test)
        self.r = self.modelAgent.model.fit(x_train, y_train, validation_data
= (x_test, y_test), epochs = 5, batch_size = 32 )
        print("retrain results")
        self.printModelStats(x_test, y_train, y_test)

    def resample(self):
        pass

# Define the Actor and Critic networks
def create_actor(state_shape, action_shape):
    state_input = Input(shape=state_shape)
    x = Dense(64, activation='relu')(state_input)
    x = Dense(32, activation='relu')(x)
    action_output = Dense(action_shape, activation='tanh')(x)
    return Model(state_input, action_output)

def create_critic(state_shape, action_shape):
    state_input = Input(shape=state_shape)
    action_input = Input(shape=action_shape)

    reshaped_action = Dense(np.prod(state_shape),
activation='relu')(action_input)
    reshaped_action = tf.reshape(reshaped_action, (-1,) + state_shape)

    concatenated_input = Concatenate(axis=-1)([state_input, reshaped_action])

    x = Dense(64, activation='relu')(concatenated_input)
    x = Dense(32, activation='relu')(x)
    q_value = Dense(1)(x)

    return Model([state_input, action_input], q_value)

class DDPGagent(BaseRLAgent):
    def __init__(self, state_shape, action_shape):
        self.actor = create_actor(state_shape, action_shape)

```

```

self.target_actor = create_actor(state_shape, action_shape)
self.critic = create_critic(state_shape, action_shape)
self.target_critic = create_critic(state_shape, action_shape)

self.target_actor.set_weights(self.actor.get_weights())
self.target_critic.set_weights(self.critic.get_weights())

self.actor_optimizer = Adam(learning_rate=0.001)
self.critic_optimizer = Adam(learning_rate=0.002)

def train(self, states, actions, rewards, next_states, dones):

    next_states = next_states.astype(actions.dtype)

    target_actions = self.target_actor(next_states)
    target_actions = tf.expand_dims(target_actions, axis=-1)

    # Train critic
    with tf.GradientTape() as tape:
        target_q = rewards + 0.99 * self.target_critic([next_states,
target_actions]) * (1 - dones)
        predicted_q = self.critic([states, actions])
        critic_loss = tf.keras.losses.MSE(target_q, predicted_q)

    # Train actor
    with tf.GradientTape() as tape:
        predicted_actions = self.actor(states)
        actor_loss = -tf.reduce_mean(self.critic([states,
predicted_actions]))

        actor_grad = tape.gradient(actor_loss,
self.actor.trainable_variables)
        self.actor_optimizer.apply_gradients(zip(actor_grad,
self.actor.trainable_variables))

    # Update target networks
    self.update_target_networks(tau=0.01)

def update_target_networks(self, tau):
    actor_weights = self.actor.get_weights()
    target_actor_weights = self.target_actor.get_weights()
    critic_weights = self.critic.get_weights()

```

```

target_critic_weights = self.target_critic.get_weights()

for i in range(len(target_actor_weights)):
    target_actor_weights[i] = tau * actor_weights[i] + (1 - tau) *
target_actor_weights[i]
    self.target_actor.set_weights(target_actor_weights)

for i in range(len(target_critic_weights)):
    target_critic_weights[i] = tau * critic_weights[i] + (1 - tau) *
target_critic_weights[i]
    self.target_critic.set_weights(target_critic_weights)

# %% [markdown]
# model base training

# %%
X,Y = read_h5("./uci_har.h5")

# %%
agent_model = ModelAgent("lgbm", X, Y)

# %%
agent_mc = MCagent(agent_model)

# %% [markdown]
# workflow

# %% [markdown]
# LSTM

# %%
x_test[0].reshape(1, 128, 9).shape

# %%
agent_mc.loadActivity(x_test[0].reshape(1, 128, 9), y_test[0])

# %%
y_test.any()

# %%
agent_mc.loadActivityCollection(x_test, y_test)

# %% [markdown]
# LGBM

```

```
# %%
def extract_features(window):
    return [
        np.mean(window, axis=0),
        np.std(window, axis=0),
        np.min(window, axis=0),
        np.max(window, axis=0),
        np.median(window, axis=0),
        np.percentile(window, 25, axis=0),
        np.percentile(window, 75, axis=0)
    ]

# %%
X, y = read_h5("./uci_har.h5")
X_features = np.array([extract_features(activity) for activity in X])
X_features = X_features.reshape(X_features.shape[0], -1)
x_train, x_test, y_train, y_test = train_test_split(X_features, Y,
test_size=0.3, random_state=42)

# %%
x_test[0].reshape(1, 63).shape

# %%
agent_model.model.predict(x_test[1].reshape(1, 63)).argmax(axis=1)
```