

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 303.72.4

До захисту допущено
Завідувач кафедри ММСА
_____ Оксана ТИМОЩУК
«__» _____ 2024 р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою
«Системний аналіз фінансового ринку»
зі спеціальності 124 «Системний аналіз»
на тему: «Методи машинного навчання для вирішення задачі Human
Activity Recognition»

Виконав:
студент 2 курсу, групи КА-32мп
Каленіченко Денис Олександрович _____

Науковий керівник:
професор кафедри ШІ,
д.т.н., професор Данилов Валерій Якович _____

Рецензент:
директор НН ФТІ,
д.т.н., професор, чл.-кор. НАНУ
Новіков Олексій Миколайович _____

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

Студент: _____

Київ – 2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)

Спеціальність — 124 «Системний аналіз»

Освітньо-професійною програмою «Системний аналіз фінансового ринку»

ЗАТВЕРДЖУЮ

Завідувач кафедри ММСА

_____ Оксана ТИМОЩУК

«__» _____ 2024 р.

ЗАВДАННЯ

**на магістерську дисертацію студенту
Каленіченку Денису Олександровичу**

1. **Тема дисертації:** «Методи машинного навчання для вирішення задачі Human Activity Recognition», науковий керівник д.т.н., професор кафедри ШІ, Данилов Валерій Якович, затверджені наказом по університету від «07» листопада 2024 р. №5001-с.
2. **Строк подання студентом дисертації:** 06.12.2024.
3. **Об'єкт дослідження:** поведінкові дані, які отримані з носимих людиною датчиків - акселерометру та гіроскопу.
4. **Предмет дослідження:** генеративні мережі у розв'язанні задачі розпізнавання активності людини.
5. **Перелік завдань, які потрібно розробити:**
 - 1) аналіз носимих пристроїв та отриманих ними даних поведінкових характеристик людини;
 - 2) аналіз алгоритмів, методів і підходів глибокого машинного навчання для розв'язання задачі розпізнавання активності людини;

- 3) розробка нового алгоритму та архітектури розпізнавання активності людини на основі методів машинного навчання з використанням генеративної методології;
- 4) формулювання вимог до програмного забезпечення та розробка відповідного програмного рішення для розпізнавання активності людини;
- 5) проведення налаштування архітектури з розпізнавання активностей людини на основі бази даних значень з давачів носимих пристроїв.
- 6) Оцінка якості розпізнавання, аналіз та інтерпретація результатів.

6. Перелік графічного (ілюстративного) матеріалу:

- 1) Графіки часових рядів, які відповідають за активності об'єкта 1 (рис. 3.1);
- 2) Статистика персональних даних учасників датасету (таблиця 3.1);
- 3) Архітектура системи розпізнавання активності (рис. 3.2);
- 4) Архітектура кодувальника з використанням LSTM (рис. 3.3 (а));
- 5) Архітектура декодувальника з використанням LSTM (рис. 3.3 (б));
- 6) Пораховані метрики для кожної з активностей з використанням LSTM (рис. 3.4);
- 7) Матриця невідповідностей для кожної з активностей для LSTM (рис. 3.5);
- 8) Архітектура кодувальника з використанням GRU (рис. 3.6 (а));
- 9) Архітектура декодувальника з використанням GRU (рис. 3.6 (б));
- 10) Пораховані метрики для кожної з активностей з використанням GRU (рис. 3.7);
- 11) Матриця невідповідностей для кожної з активностей з використанням GRU (рис. 3.8).

7. Орієнтовний перелік публікацій:

- 1) Каленіченко Д.О., Данилов В.Я. Методи вирішення задачі розпізнавання людської активності за допомогою генеративних мереж. *International Science Journal of Engineering & Agriculture*. 2024. 3(6), С. 10–15. URL: <https://doi.org/10.46299/j.isjea.20240306.02>;
- 2) Каленіченко Д.О., Данилов В.Я. Методи машинного навчання для вирішення задачі Human Activity Recognition. Збірник доповідей II Всеукраїнської науково-практичної конференції «Системні науки та інформатика» 25–29 листопада 2024 року, Київ. К., НН ІПСА КПІ ім. Ігоря Сікорського, 2024, 6 с.

8. Дата видачі завдання: 02 вересня 2024.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Отримання завдання на магістерську дисертацію	02.09.2024 – 15.09.2024	Виконано
2	Концептуальний вступ дисертації: формулювання об'єкта, предмета, мети, завдань.	16.09.2024 – 01.10.2024	Виконано
3	Огляд літератури за темою магістерської дисертації.	01.10.2024 – 15.10.2024	Виконано
4.	Розділ перший. Розгляд поняття «носимі пристрої», їх використання та важливість розпізнавання активності.	15.10.2024 – 20.10.2024	Виконано
5	Розділ другий. Огляд методів машинного навчання та методології їх розробки.	20.10.2024 – 29.10.2024	Виконано
6	Розділ третій. Розробка архітектури нейронної мережі глибокого навчання з використанням генеративного підходу. Створення програмного застосунку.	29.10.2024 – 4.11.2024	Виконано
7	Тестування та вибір архітектури нейронної мережі глибокого навчання.	4.11.2024 – 12.11.2024	Виконано
8	Четвертий розділ. Розробка власного стартап-проекту	12.11.2024 – 20.11.2024	Виконано
9	Формування висновків	20.11.2024 – 21.11.2024	Виконано
10	Оформлення пояснювальної записки	21.11.2024 – 29.11.2024	Виконано

Студент

Науковий керівник дисертації

Денис КАЛЕНІЧЕНКО

Валерій ДАНИЛОВ

РЕФЕРАТ

Магістерська дисертація: 90 с., 11 рис., 21 табл., 1 дод., 32 джерела.

Об'єктом дослідження є поведінкові дані людини з носимих пристроїв, які отримані з давачів – акселерометру та гіроскопу. В дослідження також входять способи обробки отриманих поведінкових даних та архітектур розпізнавання активності методами машинного навчання на базі цих даних.

Предметом дослідження є підгрупа методів машинного навчання, а саме генеративні мережі, їх властивості та можливості у розпізнаванні поведінкової активності людини.

Методи дослідження – це методи глибокого навчання з генеративним підходом, їх можливості для розв'язання проблеми розпізнавання поведінкової активності людини.

Метою роботи є розробка системи розпізнавання активності людини на базі поведінкових даних з давачів носимих пристроїв на основі методів машинного навчання з генеративним підходом на базі автокодувальників.

Магістерська дисертація складається зі вступу, чотирьох розділів, переліку джерел посилання та одного додатку. У першому розділі наведений розгляд понять носимих пристроїв, їх використання, актуальність та важливість. У другому розділі наведені види машинного навчання, методи та архітектури, які підходять до вирішення задачі розпізнавання активності. В третьому розділі приводиться розробка архітектури системи, тренування отриманих моделей глибокого машинного навчання і результати. Четвертий розділ включає розробку стартап-проєкту. У додатку наведено код розробленої програми.

ГЛИБОКЕ НАВЧАННЯ, ГЕНЕРАТИВНІ МЕРЕЖІ, ВАРІАЦІЙНИЙ АВТОКОДУВАЛЬНИК, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, НАПІВКЕРОВАНЕ НАВЧАННЯ.

ABSTRACT

Master's thesis: 90 p., 11 fig., 21 tables, 1 appendix, 32 references.

The object of the study is data from wearable devices that are obtained from sensors – accelerometer and gyroscope. The study also includes methods for processing them and methods for recognizing activity by machine learning methods on this data.

The subject of the study is a subgroup of machine learning methods, namely generative networks, their properties and capabilities in recognizing human activity. Research methods are generative networks, their capabilities in solving the problem of activity recognition.

The purpose of the work is to develop systems for recognizing human activity based on data from sensors of wearable devices based on machine learning methods, namely generative networks.

The master's thesis consists of an introduction, Sections 1-4, references and Appendix A. Section 1 provides an overview of the concepts of wearable devices, their use, relevance and importance. Section 2 presents the types of machine learning, methods and architectures that are suitable for solving the problem of activity recognition. Section 3 provides the development of the system architecture, training of the obtained deep machine learning models and the results. Section 4 includes the development of a start-up project. Appendix A contains the code of the developed program.

DEEP LEARNING, GENERATIVE NETWORKS, VARIATIONAL AUTOENCODER, RECURRENT NEURAL NETWORKS, SEMI-GUIDED LEARNING.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1 ОГЛЯД ТЕОРЕТИЧНИХ ПОНЯТЬ ТА ЗАСТОСУВАННЯ РОЗПІЗНАВАННЯ АКТИВНОСТІ.....	12
1.1 Носимі пристрої.....	12
1.1.1 Види носимих пристроїв.....	12
1.1.2 Використання носимих пристроїв у різних сферах.....	13
1.2 Розпізнавання активності людини	16
1.3 Актуальність та важливість задачі розпізнавання активності людини.....	18
1.4 Висновки до розділу 1	19
РОЗДІЛ 2 ОГЛЯД МЕТОДІВ МАШИННОГО НАВЧАННЯ.....	21
2.1 Методи розв’язання задачі розпізнавання активності людини	21
2.2 Опис поняття машинного навчання	21
2.3 Основні типи машинного навчання	22
2.4 Опис поняття глибокого навчання	23
2.5 Некероване навчання в розпізнаванні активності людини.....	23
2.6 Варіаційний кодувальник.....	24
2.6.1 Варіаційний автокодувальник у задачі розпізнавання активності людини	24
2.6.2 Означення варіаційного автокодувальника	25
2.6.3 Опис латентного простору варіаційного автокодувальника	27
2.6.4 Проміжні підсумки вибору архітектури варіаційного автокодувальника.....	28
2.7 Нейронні мережі для вирішення задачі розпізнавання активності	29

2.7.1	Згорткові нейронні мережі	29
2.7.2	Рекурентні нейронні мережі.....	32
2.7.2.1	Базові рекурентні нейронні мережі	32
2.7.2.2	Довга короткочасна пам'ять (LSTM)	33
2.7.2.3	Вентильні рекурентні вузли (GRU).....	34
2.7	Використання автокодувальника для виявлення аномалій в даних	35
2.8	Висновки до розділу 2.....	36
РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ АКТИВНОСТІ ЛЮДИНИ		38
3.1	Аналіз даних	38
3.2	Підготовка даних для тренування моделей	40
3.3	Вибір архітектури моделі	41
3.4	Система розпізнавання активності людини.....	42
3.5	Метрики.....	42
3.6	Рекурентний варіаційний автокодувальник на базі LSTM нейронних мереж.	44
3.6.1	Архітектура мережі.....	44
3.6.2.	Тренування моделей.....	46
3.6.3.	Аналіз результатів моделі	48
3.7	Рекурентний варіаційний автокодувальник на базі GRU нейронних мереж	50
3.7.1	Архітектура мережі.....	50
3.7.2	Тренування моделей.....	52
3.7.3	Аналіз результатів моделі.....	54
3.8	Порівняння метрик моделей	55

3.9 Висновки до розділу 3	56
РОЗДІЛ 4 РОЗРОБКА ВЛАСНОГО СТАРТАП-ПРОЄКТУ	58
4.1 План розробки стартапу та масштабування його на ринок	58
4.2 Опис ідеї стартап-проекту	60
4.3 Технологічний аудит ідеї проекту	61
4.4 Аналіз ринкових можливостей запуску стартап-проекту	63
4.5 Розроблення ринкової стратегії стартап-проекту	73
4.6 Розроблення маркетингової програми стартап-проекту	75
4.7 Висновки до розділу 4	78
ВИСНОВКИ	79
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	80
ДОДАТОК А	83

ВСТУП

У сучасному світі носимі пристрої, такі як розумні годинники, збирають і передають значні обсяги даних, отриманих від різноманітних сенсорів — акселерометрів, гіроскопів, магнітометрів тощо. Ці дані можуть бути використані для розпізнавання активності людини, що має велике значення в галузях здоров'я, фітнесу та безпеки. Проте в більшості випадків ці дані є нерозміченими, оскільки користувачі не здійснюють їх ручну анотацію, а залучення експертів для розмітки є складним та витратним процесом.

У зв'язку з цим виникає потреба в застосуванні методів навчання без нагляду, які можуть виявляти структуру та закономірності в нерозмічених даних. Варіаційні автокодувальники є потужним інструментом у цій сфері. Вони поєднують можливості автокодувальників з варіаційними методами байєсівської статистики, що дозволяє моделювати складні розподіли даних та витягувати приховані характеристики.

Класичні методи обробки даних, такі як стандартні автокодувальники або інші алгоритми навчання без нагляду, мають обмеження у здатності моделювати складні ймовірнісні розподіли та генерувати нові дані. Вони часто не враховують невизначеність у латентному просторі та можуть призводити до менш змістовного представлення даних. Варіаційні автокодувальники, навпаки, дозволяють моделювати дані з урахуванням ймовірнісних розподілів, що забезпечує більш гнучке та потужне представлення інформації. Це робить VAE більш придатними для задачі розпізнавання активності на основі нерозмічених даних.

Важливість варіаційних автокодувальників полягає в їх здатності навчатися компактному та змістовному представленню даних у латентному просторі. Це особливо корисно для аналізу даних з носимих пристроїв, де важливо виявити приховані патерни та особливості в сигналах сенсорів без

попередньої розмітки. VAE дозволяють генерувати нові зразки даних, що схожі на вихідні, а також виявляти аномалії та кластеризувати дані за схожістю.

РОЗДІЛ 1 ОГЛЯД ТЕОРЕТИЧНИХ ПОНЯТЬ ТА ЗАСТОСУВАННЯ РОЗПІЗНАВАННЯ АКТИВНОСТІ

1.1 Носимі пристрої

Носимі пристрої (Wearable devices) — це електронні пристрої, які інтегруються в одяг або аксесуари та носяться безпосередньо на тілі. Вони оснащені датчиками та технологіями зв'язку, що дозволяють збирати, обробляти та передавати дані про фізичний стан користувача та його оточення.

1.1.1 Види носимих пристроїв

Існують такі види носимих пристроїв [1]:

- **Фітнес-трекери:** пристрої, призначені для моніторингу фізичної активності, такі як кількість кроків, пройденої відстані, пульс, спалені калорії та якість сну. Приклад: Fitbit, Xiaomi Mi Band.
- **Розумні годинники:** поєднують функції фітнес-трекерів з можливостями смартфона, включаючи прийом повідомлень, дзвінків та використання додатків. Приклад: Apple Watch, Samsung Galaxy Watch.
- **Розумні окуляри:** пристрої, що використовують технології доповненої (AR) або віртуальної реальності (VR) для надання інформації або створення інтерактивного досвіду. Приклад: Google Glass, Microsoft HoloLens.
- **Носимі медичні пристрої:** спеціалізовані пристрої для моніторингу здоров'я, такі як глюкометри, монітори артеріального тиску, пристрої для ЕКГ. Вони допомагають у діагностиці та лікуванні медичних станів.

- **Розумний одяг:** одяг з вбудованими датчиками, що можуть вимірювати фізіологічні параметри, наприклад, температуру тіла, частоту серцевих скорочень, м'язову активність. Приклад: сорочки з електроміографією, взуття з датчиками тиску.
- **Носимі камери:** пристрої для запису відео та аудіо з точки зору користувача. Використовуються в спорті, правоохоронних органах та для особистих потреб. Приклад: GoPro, поліцейські боді-камери.
- **Носимі аудіопристрої:** безпроводні навушники та гарнітури з додатковими функціями, такими як шумозаглушення, біометричний зворотний зв'язок. Приклад: Apple AirPods, Jabra Elite.

1.1.2 Використання носимих пристроїв у різних сферах

Носимі пристрої використовуються у багатьох сферах, наприклад:

Сфера здоров'я та фітнес

- **Моніторинг фізичної активності:** фітнес-трекери та розумні годинники відстежують кількість кроків, пройдену відстань, спалені калорії та активні хвилини. Це допомагає користувачам встановлювати цілі, слідкувати за прогресом та підтримувати мотивацію до фізичної активності.
- **Контроль серцевого ритму:** датчики серцевого ритму дозволяють відстежувати пульс у реальному часі, що корисно для оптимізації тренувань та контролю здоров'я серцево-судинної системи.
- **Аналіз сну:** носимі пристрої можуть відстежувати фази сну (глибокий, легкий, REM), виявляти порушення сну та надавати рекомендації щодо його покращення.
- **Моніторинг стресу та відновлення:** вимірювання варіабельності серцевого ритму та інших показників допомагає оцінити рівень стресу та ефективність відновлення після фізичних навантажень.

- Персональні тренування: пристрої можуть пропонувати індивідуальні тренувальні плани на основі зібраних даних та цілей користувача.

Сфера медицини

- Постійний моніторинг пацієнтів: носимі медичні пристрої дозволяють віддалено відстежувати життєві показники пацієнтів, такі як артеріальний тиск, рівень глюкози в крові, сатурацію кисню та ЕКГ.

- Раннє виявлення захворювань: аналіз зібраних даних може допомогти у виявленні ранніх ознак захворювань або загострень хронічних станів.

- Телемедицина: дані з пристроїв можуть передаватися лікарям для дистанційної діагностики та коригування лікування.

- Моніторинг пацієнтів після операцій: сприяє швидкому виявленню ускладнень та забезпечує безперервний догляд.

Сфери промисловості та бізнесу

- Безпека працівників: носимі пристрої можуть відстежувати стан здоров'я працівників у небезпечних умовах (наприклад, у шахтах або на хімічних заводах) та попереджати про потенційні ризики.

- Оптимізація робочих процесів: відстеження місцезнаходження та діяльності працівників допомагає в управлінні персоналом та ресурсами.

- Доступ до інформації: розумні окуляри можуть надавати працівникам доступ до інструкцій, схем та даних без необхідності відриватися від роботи.

Сфери безпеки та моніторингу

- Відстеження місцезнаходження: пристрої для дітей, літніх людей або домашніх тварин, які дозволяють визначати їхнє місцезнаходження та отримувати сповіщення при виході за задані межі.

- Системи екстреного виклику: Можливість швидко викликати допомогу у випадку надзвичайної ситуації, наприклад, падіння або раптового погіршення здоров'я.

- Домашня безпека: керування системами сигналізації та відеоспостереження за допомогою носимих пристроїв.

Сфери спорту та екстремальних видів діяльності

- Аналіз спортивної техніки: датчики, вбудовані в одяг або обладнання, відстежують рухи спортсмена, допомагаючи покращити техніку та продуктивність.

- Безпека: моніторинг життєвих показників та оточуючих умов (температура, висота, тиск) під час екстремальних заходів.

- Соціальний аспект: порівняння результатів з іншими спортсменами, участь у віртуальних змаганнях.

Сфера соціальної взаємодії

- Комунікація: прийом дзвінків, повідомлень та сповіщень безпосередньо на носимий пристрій.

- Соціальні додатки: використання спеціальних платформ для обміну досвідом, участі в спільнотах та групових заходах.

- Емоційне здоров'я: деякі пристрої можуть відстежувати емоційний стан користувача та пропонувати підтримку або рекомендації.

Сфери розумного дому та Інтернету речей (IoT)

- Керування домашніми пристроями: регулювання освітлення, температури, побутової техніки за допомогою голосових команд або жестів.

- Автоматизація: налаштування сценаріїв, які виконуються автоматично на основі місцезнаходження або часу.

Сфери персоналізованого маркетингу та реклами

- Цільові пропозиції: аналіз даних про активність та вподобання користувача для надання релевантних рекламних матеріалів.

- Інтерактивний досвід покупок: використання AR для віртуальної примірки одягу або перегляду товарів у домашніх умовах.

Сфера дослідження та наука

- Збір великих даних: носимі пристрої надають можливість зібрати великі обсяги даних про поведінку та здоров'я людей для наукових досліджень.
- Експерименти в реальному часі: відстеження впливу різних факторів на фізичний та психічний стан учасників досліджень.

Сфера військових та спецслужб

- Тактична підтримка: носимі пристрої можуть надавати інформацію про місцезнаходження союзників, стан боєзапасів та навколишнє середовище.
- Навчання та симуляції: використання VR для тренувань у безпечному та контрольованому середовищі.

Сфери фінансів та платежів

- Безконтактні платежі: носимі пристрої з NFC-чипами дозволяють здійснювати платежі без використання карток чи смартфонів.
- Фінансовий менеджмент: відстеження витрат, бюджетування та отримання фінансових сповіщень.

Сфера особистого асистенту

- Голосове керування: інтеграція з голосовими помічниками для виконання команд, пошуку інформації та керування іншими пристроями.
- Нагадування та планування: встановлення подій у календарі, отримання нагадувань та організація завдань.

1.2 Розпізнавання активності людини

Розпізнавання активності людини (Human Activity Recognition, HAR) — це процес автоматичного визначення та класифікації фізичних дій або поведінки людини на основі даних, зібраних з різних сенсорів. Ці сенсори можуть бути вбудовані в носимі пристрої, смартфони або розміщені в

оточуючому середовищі. Мета HAR полягає у створенні систем, які можуть точно ідентифікувати такі активності, як ходьба, біг, сидіння, підйом по сходах, їзда на велосипеді тощо.

Однак через низку причин вирішення цієї задачі може бути дуже складним:

Складність та різноманітність даних:

- Високорозмірність даних: датчики генерують великі обсяги багатовимірних даних з високою частотою, що ускладнює їх обробку та аналіз;
- Гетерогенність сенсорів: різні типи сенсорів (акселерометри, гіроскопи, магнітометри) мають різні характеристики та можуть надавати неоднорідні дані;
- Шум та артефакти: дані можуть містити шуми через зовнішні фактори, неточності сенсорів або помилки користувача (наприклад, неправильне носіння пристрою).

Індивідуальні відмінності:

- Фізіологічні різниці: зріст, вага, статура та інші фізичні параметри впливають на сенсорні показники;
- Стиль руху: кожна людина має унікальний спосіб виконання певних дій, що ускладнює створення універсальних моделей.

Контекстна залежність:

- Зміна оточення: різні умови (поверхня, погода, одяг) можуть впливати на характер рухів;
- Перехідні стани: люди часто переходять між активностями без чітких меж, створюючи складність для класифікації.

Реальний час та обмежені ресурси:

- Обчислювальна ефективність: носимі пристрої мають обмежену обчислювальну потужність та енергоспоживання, що вимагає оптимізації алгоритмів;

- Затримки в обробці: для деяких застосувань критично важлива обробка даних у реальному часі.

Конфіденційність та етичні аспекти:

- Збір персональних даних: обробка даних про активність може розкрити конфіденційну інформацію про особу;
- Етичні питання: необхідність отримання згоди та забезпечення прозорості використання даних.

1.3 Актуальність та важливість задачі розпізнавання активності людини

Розпізнавання активності людини (Human Activity Recognition, HAR) є процесом автоматичного визначення та класифікації фізичних дій або поведінки людини на основі даних, зібраних з різних сенсорів, зокрема носимих пристроїв [1]. Ця задача набуває все більшої актуальності завдяки широкому поширенню носимих технологій та зростаючій потребі в інтелектуальних системах, які можуть взаємодіяти з користувачами на більш персоналізованому рівні [2].

Однією з основних проблем у HAR є обробка великих обсягів даних, які генеруються сенсорами в реальному часі [3]. Поведінкові дані можуть бути шумними та варіюватися через різні фактори, такі як індивідуальні відмінності між користувачами, зміни в оточенні та різні умови експлуатації пристроїв [4]. Це ускладнює розробку універсальних моделей, які можуть точно розпізнавати активність різних людей у різних контекстах.

Крім того, обмежені обчислювальні ресурси та енергоспоживання носимих пристроїв створюють додаткові виклики для реалізації складних алгоритмів машинного навчання на самих пристроях [5]. Це вимагає оптимізації алгоритмів для роботи в режимі реального часу без втрати точності.

Актуальність задачі розпізнавання активності людини зумовлена її широким застосуванням у різних сферах. У медицині HAR використовується для моніторингу пацієнтів, реабілітації та раннього виявлення аномалій у поведінці [1]. У фітнесі та спорті вона допомагає відстежувати прогрес та оптимізувати тренування [2]. У сфері розумних будинків та міст HAR сприяє створенню адаптивних систем, які підлаштовуються під потреби користувачів [3].

Задача розпізнавання активності людини має потенціал суттєво покращити якість життя та здоров'я населення. У медичному контексті вона дозволяє здійснювати безперервний моніторинг пацієнтів з хронічними захворюваннями, забезпечуючи своєчасне втручання у разі потреби [1]. Це особливо важливо для літніх людей та пацієнтів з обмеженими можливостями, яким необхідний постійний нагляд.

У сфері фітнесу HAR мотивує людей до активного способу життя, надаючи зворотній зв'язок та персоналізовані рекомендації [2]. Це сприяє запобіганню захворюванням, пов'язаним із малорухливим способом життя, та покращує загальний стан здоров'я населення.

У промисловості та бізнесі HAR може підвищити безпеку працівників, виявляючи небезпечні дії або умови [4]. У розумних містах та будинках вона сприяє підвищенню енергоефективності та комфорту, автоматично адаптуючи оточення під активність мешканців [3].

Науково-технологічна важливість HAR полягає у стимулюванні розвитку передових алгоритмів машинного та глибокого навчання [5]. Це відкриває нові можливості для інновацій, міждисциплінарних досліджень та створення нових продуктів і послуг, що можуть суттєво вплинути на економічний розвиток.

1.4 Висновки до розділу 1

Розпізнавання активності людини виступає як одна з ключових задач сучасної науки та технологій, що має значний вплив на різноманітні сфери життя. Зі зростанням популярності носимих пристроїв та необхідністю більш інтерактивних і персоналізованих систем, актуальність HAR продовжує збільшуватися.

Основні проблеми, з якими стикається HAR, включають обробку великих та варіативних обсягів даних, шум та неточності, пов'язані з різними умовами експлуатації та індивідуальними відмінностями користувачів. Обмежені ресурси носимих пристроїв додають складності в реалізації ефективних та точних алгоритмів у реальному часі.

Попри ці виклики, важливість HAR важко переоцінити. У медичній сфері вона сприяє покращенню якості догляду за пацієнтами, дозволяючи здійснювати постійний моніторинг та своєчасне втручання. У галузі фітнесу та спорту HAR мотивує людей до активного способу життя та допомагає досягати кращих результатів. Впровадження HAR у розумні будинки та міста підвищує комфорт, енергоефективність та безпеку мешканців.

Крім того, розвиток HAR стимулює прогрес у галузі машинного та глибокого навчання, відкриваючи нові можливості для інновацій, міждисциплінарних досліджень та економічного розвитку. Це підкреслює необхідність подальших досліджень та розробок у цій сфері, а також вирішення існуючих проблем для повної реалізації потенціалу розпізнавання активності людини.

РОЗДІЛ 2 ОГЛЯД МЕТОДІВ МАШИННОГО НАВЧАННЯ

2.1 Методи розв'язання задачі розпізнавання активності людини

Розпізнавання активності людини ефективно вирішується за допомогою методів машинного навчання, особливо глибокого навчання. Машинне навчання дозволяє створювати моделі, які можуть аналізувати великі обсяги даних, зібраних з носимих пристроїв, і робити точні передбачення щодо типів активності на основі цих даних.

Машинне навчання використовує алгоритми, які навчаються на прикладах і можуть розпізнавати закономірності в даних. У контексті розпізнавання активності, алгоритми машинного навчання аналізують сигнали від сенсорів (наприклад, акселерометрів, гіроскопів) для класифікації дій користувача, таких як ходьба, біг, сидіння тощо.

Глибоке навчання, як підрозділ машинного навчання, базується на використанні багат шарових нейронних мереж, які можуть моделювати складні нелінійні залежності в даних. Глибокі нейронні мережі, такі як згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), особливо ефективні для обробки часових рядів і просторово-часових даних, які характерні для сигналів сенсорів носимих пристроїв.

2.2 Опис поняття машинного навчання

Машинне навчання (ML) — це розділ штучного інтелекту, який фокусується на розробці алгоритмів і моделей, що дозволяють комп'ютерам вчитися з даних і робити прогнози або приймати рішення без явного програмування для кожного конкретного завдання. Основна ідея полягає в тому, щоб система могла покращувати свою продуктивність на основі досвіду.

«Комп'ютерна програма вважається здатною навчатися з досвіду E щодо деякого класу задач T і показника ефективності P , якщо її ефективність у задачах з T , виміряна за допомогою P , покращується з досвідом E .» [6].

2.3 Основні типи машинного навчання

Навчання з учителем (Supervised Learning): алгоритм навчається на позначених даних, де кожен вхідний приклад має відповідний вихід. Метою є навчитися прогнозувати виходи для нових, невідомих вхідних даних.

Приклади задач:

- **Класифікація:** визначення категорії, до якої належить вхідний приклад (наприклад, розпізнавання зображень).
- **Регресія:** прогнозування безперервного значення (наприклад, прогнозування цін на нерухомість).

Навчання без учителя (Unsupervised Learning): алгоритм працює з непозначеними даними, намагаючись виявити приховані структури або закономірності.

Приклади задач:

- **Кластеризація:** групування схожих прикладів разом (наприклад, сегментація клієнтів).
- **Зниження розмірності:** спрощення даних при збереженні важливої інформації (наприклад, аналіз головних компонент).

Приклади алгоритмів: K-means, ієрархічна кластеризація, автоенкодери.

Напів-контрольоване навчання (Semi-Supervised Learning) [7]: поєднує елементи навчання з учителем та без учителя, використовуючи невелику кількість позначених даних разом з великою кількістю непозначених.

Застосування: коли позначення даних є дорогим або трудомістким процесом.

2.4 Опис поняття глибокого навчання

Глибоке навчання (Deep Learning) [8] — це підгалузь машинного навчання, яка використовує штучні нейронні мережі з багатьма прихованими шарами для моделювання та виявлення складних закономірностей у даних. Воно базується на навчанні представлень даних на різних рівнях абстракції, що дозволяє системі вчитися від простих до більш складних ознак.

Основні компоненти глибокого навчання:

Багатошарові нейронні мережі: включають вхідний шар, кілька прихованих шарів та вихідний шар. Кількість шарів визначає «глибину» моделі.

Активуючі функції: нелінійні функції (наприклад, ReLU, сигмоїдна, tanh), які додають нелінійність до моделі та дозволяють мережі вивчати складні взаємозв'язки.

Алгоритми навчання: використовують метод зворотного поширення помилки для оновлення вагових коефіцієнтів мережі шляхом мінімізації функції втрат.

2.5 Некероване навчання в розпізнаванні активності людини

Розпізнавання активності людини за допомогою машинного навчання є ефективним підходом до аналізу даних, отриманих з носимих пристроїв. Зокрема, некероване навчання стає все більш популярним у цій галузі, оскільки воно дозволяє виявляти приховані структури та патерни в даних без потреби в мічених зразках [9]. Це особливо важливо, оскільки процес збирання та анотування мічених даних є трудомістким і дорогим.

Некероване навчання краще підходить для задач розпізнавання активності через здатність працювати з великими обсягами немічених даних, які легко збираються з сенсорів носимих пристроїв [10]. Методи некерованого навчання, такі як кластеризація, автоенкодери та генеративно-змагальні мережі (GAN), дозволяють моделювати складні патерни в даних та виявляти аномалії або нові види активності [11].

Наприклад, автоенкодери можуть бути використані для зменшення розмірності даних та виділення важливих ознак, що покращує продуктивність подальших алгоритмів класифікації [12]. Генеративно-змагальні мережі можуть навчатися генерувати нові дані, подібні до оригінальних, що сприяє кращому розумінню структури даних та підвищенню стійкості моделі [13].

Крім того, некероване навчання дозволяє системам адаптуватися до індивідуальних особливостей користувачів та змін у їх поведінці [9]. Це робить моделі більш гнучкими та узагальненими, що є критично важливим для застосувань у реальному світі.

2.6 Варіаційний кодувальник

2.6.1 Варіаційний автокодувальник у задачі розпізнавання активності людини

Розпізнавання активності людини (HAR) за допомогою некерованого навчання набуває все більшої популярності, оскільки воно дозволяє аналізувати великі обсяги немічених даних, які легко отримати з носимих сенсорів [9]. Варіаційні автокодувальники (VAE) є потужним інструментом некерованого навчання, які краще підходять для задачі розпізнавання активності людини з кількох причин:

Моделювання складних розподілів даних: VAE можуть моделювати складні та багатомодальні розподіли даних, характерні для сигналів сенсорів

носимих пристроїв [14]. Це дозволяє ефективно витягувати приховані патерни та структури з даних.

Обробка немічених даних: оскільки процес анотації даних є трудомістким, VAE, що працюють у некерованому режимі, можуть навчатися на великих обсягах нерозмічених даних, зменшуючи потребу в розмічених зразках [15].

Структурований латентний простір: VAE вивчають неперервний та гладкий латентний простір, де подібні активності проєктуються в близькі області. Це дозволяє здійснювати інтерполяцію між активностями та виявляти аномалії [16].

Адаптивність до індивідуальних відмінностей: VAE можуть адаптуватися до індивідуальних особливостей користувачів та змін у їх поведінці, що покращує здатність моделей до узагальнення [10].

2.6.2 Означення варіаційного автокодувальника

Варіаційний автокодувальник (VAE) — це тип глибокої генеративної моделі, яка поєднує властивості автокодувальників та варіаційного байєсового виведення [17]. VAE складається з двох основних компонентів:

- **Енкодер (Кодувальник):** перетворює вхідні дані x у параметри латентного розподілу $q_{\phi}(z | x)$: де z — латентні змінні, а ϕ — параметри енкодера.
- **Декодер (Декодувальник):** генерує відновлені дані $\hat{x}$ з латентних змінних z , моделюючи розподіл $p_{\theta}(x | z)$, де θ — параметри декодера.

На відміну від класичних автокодувальників, варіаційник автокодувальник використовують стохастичне кодування, де енкодер виробляє параметри розподілу (середнє μ та дисперсія σ^2) латентних змінних:

$$\mu = f_{\phi}^{\mu}(x), \quad \ln \sigma^2 = f_{\phi}^{\sigma}(x)$$

Латентні змінні z вибираються шляхом семплювання з нормального розподілу:

$$z = \mu + \sigma \odot \epsilon, \quad \epsilon \sim N(0, I)$$

Функція втрат VAE складається з двох частин:

Реконструкційна втрата: вимірює, наскільки добре декодер може відновити вхідні дані з латентних змінних.

$$\mathbb{L}_{recon} = -\mathbb{E}_{q_{\phi}(z|x)}[\ln p_{\theta}(x|z)]$$

Відхилення Кульбака-Лейблера (англ. *Kullback-Leibler divergence*): регуляризує латентний простір, наближаючи апостеріорний розподіл $q_{\phi}(z|x)$ до апіорного розподілу $p(z)$ (зазвичай стандартного нормального розподілу).

$$\mathbb{L}_{KL} = D_{KL}(q_{\phi}(z|x)||p_{\theta}(z))$$

Загальна функція втрат:

$$\mathbb{L} = \mathbb{L}_{recon} + \mathbb{L}_{KL} \quad (2.6.1)$$

Відхилення Кульбака-Лейблера є мірою того, наскільки один розподіл ймовірності відрізняється від іншого. Воно використовується в теорії інформації для вимірювання «відстані» між двома розподілами ймовірності. У контексті машинного навчання та, зокрема, варіаційних автокодувальників, відхилення Кульбака-Лейблера служить як регуляризаційний термін, що змушує апостеріорний розподіл латентних змінних бути близьким до апіорного розподілу.

Для двох дискретних розподілів ймовірності $P(x)$ та $Q(x)$, визначених на одному і тому ж просторі подій, Відхилення Кульбака-Лейблера $D_{KL}(P \parallel Q)$ визначається як:

$$D_{KL}(P \parallel Q) = \sum_x P(x) \ln \frac{P(x)}{Q(x)}$$

Для неперервних розподілів ймовірності $P(x)$ та $Q(x)$:

$$D_{KL}(P \parallel Q) = \int_{-inf}^{+inf} P(x) \ln \frac{P(x)}{Q(x)} dx$$

У варіаційному автокодувальнику це відхилення Кульбака-Лейблера змушує апостеріорний розподіл $q_\phi(z | x)$ бути близьким до стандартного нормального розподілу $p(z)$. Це допомагає уникнути ситуації, коли модель використовує лише обмежену частину латентного простору або створює розриви в ньому.

Значення відхилення Кульбака-Лейблера для варіативного автокодувальника:

- **Регуляризація:** запобігає перенавчанню, забезпечуючи здатність моделі до узагальнення.
- **Гладкість латентного простору:** сприяє тому, щоб близькі точки в латентному просторі відповідали схожим реконструкціям у вихідному просторі.
- **Генеративні можливості:** дозволяє семплювати з апіорного розподілу $p(z)$ для генерації нових зразків.

2.6.3 Опис латентного простору варіаційного автокодувальника

Латентний простір у VAE є неперервним та структурованим простором, який моделює приховані фактори варіації у даних [18]. У контексті розпізнавання активності людини, латентний простір має такі властивості:

Кластеризація активностей: подібні активності проєктуються в близькі області латентного простору. Це дозволяє виявляти природні кластери без розмічених даних [16].

Інтерполяція між активностями: завдяки неперервності латентного простору, можна здійснювати плавні переходи між різними активностями, що допомагає розуміти взаємозв'язки між ними [19].

Виявлення аномалій: аномальні або рідкісні зразки розташовуються в областях латентного простору з низькою ймовірністю, що дозволяє їх ефективно виявляти [20].

Зменшення розмірності: латентний простір зазвичай має меншу розмірність, ніж вхідні дані, що спрощує подальший аналіз та візуалізацію [12].

2.6.4 Проміжні підсумки вибору архітектури варіаційного автокодувальника

Варіаційний автокодувальник є генеративною моделлю, яка має явний математичний формалізм для роботи з латентним простором. Це дозволяє отримувати компактне та інформативне представлення вхідних даних. У контексті аналізу даних акселерометрів, VAE може навчитися розподілу «нормальних» патернів руху користувача, дозволяючи детектувати аномалії, якщо поточні дані значно відрізняються від тих, які були представлені в латентному просторі.

На відміну від класичного автокодувальника, VAE враховує невизначеність у латентному просторі через використання розподілів (наприклад, багатовимірного нормального). Це є критично важливим для задачі верифікації, оскільки рухи людини завжди мають певну природну варіативність.

Вбудована регуляризація у вигляді мінімізації відхилення Кульбака-Лейблера між латентним розподілом та апіорним (зазвичай нормальним) забезпечує стійкість моделі до шуму і випадкових змін у вхідних даних акселерометра.

VAE оптимізує параметри таким чином, щоб мінімізувати різницю між вхідними та відновленими даними (функція відновлення). Якщо відновлення виконується з низькою похибкою, це свідчить про відповідність активності «звичайному» профілю користувача. Натомість високі похибки сигналізують про невідповідність (аномалії), які можуть бути викликані іншою активністю, зміною користувача або навіть маніпуляцією.

2.7 Нейронні мережі для вирішення задачі розпізнавання активності

Для ефективної роботи з часовими рядами, які характерні для сигналів носимих сенсорів, необхідно використовувати архітектури, здатні враховувати тимчасові залежності та патерни в даних. Для цього в моделі варіаційних автокодувальників інтегруються згорткові нейронні мережі та рекурентні нейронні мережі, які спеціально розроблені для обробки часових та послідовних даних.

2.7.1 Згорткові нейронні мережі

Згорткові нейронні мережі це тип нейронної мережі, який використовує операцію згортки для витягування ознак з даних. Основна ідея полягає в тому, щоб навчитися фільтрам (ядрам згортки), які здатні виявляти локальні патерни вхідних даних.

Архітектура CNN

CNN зазвичай складається з таких основних шарів:

Згорткові шари (Convolutional Layers): виконують згортку вхідних даних з ядрами (фільтрами), генеруючи карти ознак. Формально, вихід згорткового шару для одного каналу можна записати як:

$$y_{i,j} = (X * W)_{i,j} + b = \sum_{m=1}^M \sum_{n=1}^N X_{i+m,j+n} W_{m,n} + b$$

де:

- X – вхідний сигнал (наприклад, часовий ряд),
- W – ядро згортки (фільтр),
- b – зміщення,
- $y_{i,j}$ – вихідне значення.

Активаційні функції: застосовуються після згортки для введення нелінійності. Популярні функції: ReLU (Rectified Linear Unit), сигмоїда, tanh.

Шари підвибірки (Pooling Layers): зменшують розмірність даних, агрегуючи інформацію в локальних областях. Найпоширеніші операції: максимальне підвибіркування та середнє підвибіркування.

Повнозв'язні шари (Fully Connected Layers): використовуються в кінці мережі для класифікації або регресії, де кожен нейрон пов'язаний з усіма нейронами попереднього шару.

Принципи роботи CNN

Локальні зв'язки: ядра згортки взаємодіють лише з невеликими частинами вхідних даних, що дозволяє моделі виявляти локальні патерни.

Спільні ваги (Shared Weights): ті самі ядра використовуються по всьому вхідному простору, що зменшує кількість параметрів і покращує узагальнення.

Інваріантність до зсувів: завдяки спільним вагам і підвибірці, CNN можуть розпізнавати ознаки незалежно від їхнього положення.

У задачі розпізнавання активності людини поведінкові дані зазвичай представлені у вигляді часових рядів, отриманих з сенсорів (акселерометри, гіроскопи тощо). Згорткові нейронні мережі можуть бути адаптовані для обробки одновимірних (1D) часових рядів:

1D-CNN: згортка виконується вздовж тимчасової осі. Ядра згортки рухаються по часу, виявляючи тимчасові патерни.

Багатоканальні входи: дані з різних сенсорів або осей (X, Y, Z) можуть бути об'єднані як різні канали вхідного сигналу.

Використання згорткових нейронних мереж для вирішення задачі розпізнавання активності людини несе такі переваги:

Витягування локальних тимчасових ознак: CNN ефективно виявляють локальні патерни в часових рядах, такі як піки, імпульси чи повторювані сигнали, характерні для певних видів активності [3]. У цьому ж дослідженні було доведено, що згорткові нейронні мережі перевершують традиційні методи навчання за точністю.

Інваріантність до шуму та зміщень: завдяки підвибірці та спільним вагам, CNN стійкі до шуму та невеликих варіацій у даних [21].

Ефективність обчислень: CNN мають менше параметрів порівняно з повно-зв'язними мережами, що зменшує ризик перенавчання та покращує швидкість навчання [22].

У дослідженні [5] використовували глибокі згорткові нейронні мережі для автоматичного витягування ознак з сирих даних сенсорів без попередньої обробки, досягаючи високої точності класифікації.

Згорткові нейронні мережі є потужним інструментом для розпізнавання активності людини завдяки їх здатності ефективно витягувати локальні тимчасові ознаки з даних сенсорів. Вони демонструють високу точність, стійкість до шуму та ефективність обчислень, що робить їх популярним вибором у цій галузі.

2.7.2 Рекурентні нейронні мережі

Рекурентні нейронні мережі є класом нейронних мереж, спеціально розроблених для обробки послідовних даних. Вони здатні моделювати тимчасові залежності та запам'ятовувати попередню інформацію, що робить їх особливо корисними для задач, де порядок даних має значення, таких як розпізнавання активності людини.

2.7.2.1 Базові рекурентні нейронні мережі

Рекурентні нейронні мережі відрізняються від традиційних нейронних мереж тим, що вони мають рекурентні зв'язки, які дозволяють їм зберігати інформацію про попередні стани. На кожному кроці часу мережа отримує вхідний сигнал та попередній прихований стан, обчислюючи новий прихований стан.

Нехай x_t — вхідний вектор у момент часу t , h_t — прихований стан, а y_t — вихід мережі. Тоді оновлення прихованого стану та обчислення виходу задаються формулами:

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

$$y_t = W_{hy}h_t + b_y$$

де:

- W_{xh}, W_{hh}, W_{hy} — вагові матриці,
- b_h, b_y — вектори зміщень,
- $\tanh$ — функція активації гіперболічного тангенсу.

Однак, базові рекурентні нейронні мережі мають ряд проблем:

Зникнення та вибух градієнтів: при навчанні довгих послідовностей градієнти можуть експоненціально зменшуватися або збільшуватися, що ускладнює навчання мережі [23].

Обмежена довготривала пам'ять: базові RNN неефективні в зберіганні інформації на великих відстанях у часі.

Для вирішення цих проблем були винайдені та розроблені ще два типи рекурентних нейронних мереж: LSTM та GRU.

2.7.2.2 Довга короткочасна пам'ять (LSTM)

LSTM (Long Short-Term Memory) — це спеціальний вид RNN, розроблений для вирішення проблеми зникнення градієнтів та покращення здатності мережі зберігати довготривалу інформацію [24]. LSTM використовують комірки пам'яті та механізми вентилів для контролю потоку інформації.

Архітектура LSTM

LSTM-комірка містить три основні «вентилі»:

Вентиль забування f_t : визначає, яку інформацію з попереднього стану слід забути.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

Вентиль введення i_t : визначає, яку нову інформацію слід додати до стану.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

Вентиль виходу o_t : визначає, яку інформацію з комірки пам'яті слід використовувати для обчислення виходу.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

Оновлення стану комірки та виходу:

$$\widetilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t$$

$$h_t = o_t \odot \tanh(c_t)$$

де:

- σ – сигмоїдальна функція активації,
- $\odot$ – по-компонентне множення.

Така архітектура надає LSTM декілька переваг:

- **Збереження довготривалої інформації:** механізми вентилів дозволяють LSTM ефективно зберігати та маніпулювати інформацією на великих відстанях у часі.
- **Вирішення проблеми зникнення градієнтів:** завдяки спеціальній архітектурі градієнти можуть ефективно поширюватися назад у часі.

2.7.2.3 Вентильні рекурентні вузли (GRU)

GRU — це спрощена версія LSTM, запропонована для зменшення кількості параметрів та спрощення моделі без значної втрати продуктивності [25].

Архітектура GRU

GRU поєднує вентиль забування та вентиль введення в один вентиль оновлення (z_t) і використовує вентиль скидання (r_t).

Вентиль оновлення:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

Вентиль скидання:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

Оновлення прихованого стану:

$$\widetilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h}_t$$

GRU має декілька переваг відносно LSTM:

- **Менше параметрів:** GRU має менше вентилів та параметрів, що зменшує обчислювальну складність.

- **Спрощена архітектура:** легше налаштовується та швидше навчається порівняно з LSTM.

2.7.2.4 Порівняння RNN, LSTM та GRU в вирішення задачі розпізнавання активності людини

- **Базові RNN:** хоча вони можуть моделювати послідовні дані, проблема зникнення градієнтів обмежує їх ефективність у задачах з довгими послідовностями [23].

- **LSTM:** завдяки вентилям та коміркам пам'яті, LSTM демонструють високу точність у HAR, особливо при розпізнаванні складних та довготривалих активностей [26].

- **GRU:** мають продуктивність, подібну до LSTM, але з меншим обсягом обчислень, що робить їх придатними для застосувань з обмеженими ресурсами [27].

2.7 Використання автокодувальника для виявлення аномалій в даних

Стаття [28] фокусується на застосуванні рекурентних автокодувальників для задачі біометричної верифікації на основі даних акселерометрів, отриманих з носимих пристроїв. Аналіз використання автокодувальників в задачах розпізнавання активності та верифікації біометричних характеристик демонструє перспективність цього підходу.

Основна увага приділяється рекурентним автокодувальникам, оскільки такі архітектури добре пристосовані для роботи з часовими рядами, які є характерними для даних акселерометрів. Було показано, що автокодувальники здатні точно відновлювати патерни руху та забезпечувати ефективне виявлення аномалій.

Результати, отримані в статті, показують високу ефективність автокодувальників для задач біометричної верифікації, однак використання VAE може ще більше підвищити точність та стабільність роботи системи.

2.8 Висновки до розділу 2

Варіаційний автокодувальник підходить для вирішення задачі розпізнавання активності людини з наступних причин.

Виявлення аномалій через відновлення даних: основна ідея роботи автокодувальників, у тому числі VAE, полягає в мінімізації різниці між вхідними та відновленими даними. У задачі розпізнавання активності це означає:

- Якщо різниця між відновленими даними та оригінальними є низькою, модель правильно ідентифікує активність як знайому.
- Якщо різниця висока, активність може бути іншою (аномальною), оскільки вона не відповідає шаблонам, закодованим у латентному просторі.

Здатність до роботи з часовими рядами: дані акселерометрів є часовими рядами, які характеризуються кореляцією між сусідніми вимірюваннями. Завдяки рекурентній структурі, яка може бути поєднана з VAE (наприклад, через LSTM або GRU блоки), модель здатна враховувати ці часові залежності.

Стійкість до шуму: через дані акселерометра зазвичай виникає багато шуму (наприклад, через зовнішні вібрації чи помилки сенсора). Завдяки ймовірнісній природі VAE модель здатна ефективніше ігнорувати такі випадкові збурення.

Зважаючи на це, варіаційний автокодувальник є обґрунтованим вибором для задачі розпізнавання активності на основі даних акселерометрів. Завдяки ймовірнісному підходу до роботи з латентним простором, стійкості до шуму

та здатності виявляти аномалії через аналіз похибки відновлення, VAE може забезпечити точніше і стабільніше розпізнавання в порівнянні з традиційними автокодувальниками. Ця архітектура також відкриває можливості для більш глибокого аналізу даних і створення адаптивних біометричних систем.

Також вибір LSTM-мережі для реалізації варіаційного автокодувальника у задачах аналізу даних акселерометрів та біометричної верифікації є обґрунтованим і ефективним з кількох причин, які пов'язані із природою вхідних даних та архітектурними особливостями LSTM. Дані акселерометрів, що є послідовностями часових рядів, вимагають врахування довгострокових залежностей та контексту, щоб забезпечити точне відновлення та розуміння структури руху. LSTM-мережі, завдяки своїй унікальній архітектурі, чудово підходять для цього.

На відміну від звичайних нейронних мереж, LSTM включає механізм збереження та оновлення пам'яті через «комірки пам'яті» і спеціальні вентиля-структури (вхідний, вихідний і забуваючий вентиля). Ці механізми дозволяють моделі зберігати релевантну інформацію протягом тривалих проміжків часу, ігноруючи неважливі чи шумові аспекти даних. У контексті роботи з сигналами акселерометру це означає, що модель може «запам'ятовувати» особливі патерни руху протягом кількох секунд чи навіть хвилин, що є критичним для точного розпізнавання активності чи виявлення аномалій.

РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ АКТИВНОСТІ ЛЮДИНИ

3.1 Аналіз даних

У цьому розділі буде розглянуто обраний нами набір даних (датасет) та його детальна статистика.

Для цього експерименту був обраний датасет [MotionSense](#) [33]. Цей датасет включає у себе часові ряди даних, згенеровані датчиками акселерометра та гіроскопа (положення, гравітація, користувацьке прискорення та швидкість обертання). Інформація збиралася з використанням iPhone 6s, розташованого в передньому кишені учасника за допомогою SensingKit, який збирає дані з фреймворку Core Motion на пристроях iOS. Всі дані були зібрані з частотою 50 Гц. Всього 24 учасники різної статі, віку, ваги та зросту виконали 6 видів активностей у 15 випробуваннях в однаковому середовищі та умовах: спуск по сходах, підйом по сходах, ходьба, біг, сидіння та стояння. За допомогою цього датасету ми сподіваємося знайти особисті відмітки атрибутів у часових рядах даних з датчиків, тобто патерни, які є характерними для певних атрибутів та можуть бути використані для визначення статі чи особистості суб'єктів даних, крім їхніх активностей.

Для кожного учасника дослідження спочатку були зібрані їхні демографічні дані (вік та стать) та інформація, пов'язана з фізичним станом (зріст і вага). Потім їм надали спеціальний смартфон (iPhone 6) і попросили зберігати його в передній кишені штанів під час експерименту. Всі учасники були попрошені вдягати плоскі взуття. Після цього їх попросили виконати 6 різних видів активностей (спуск по сходах, підйом по сходах, сидіння, стояння та біг) на території університетського кампусу Королеви Марії в Лондоні в районі Майл-Енд.

Графіки активності для учасника №7 для різних активностей наведені на малюнку 3.1:

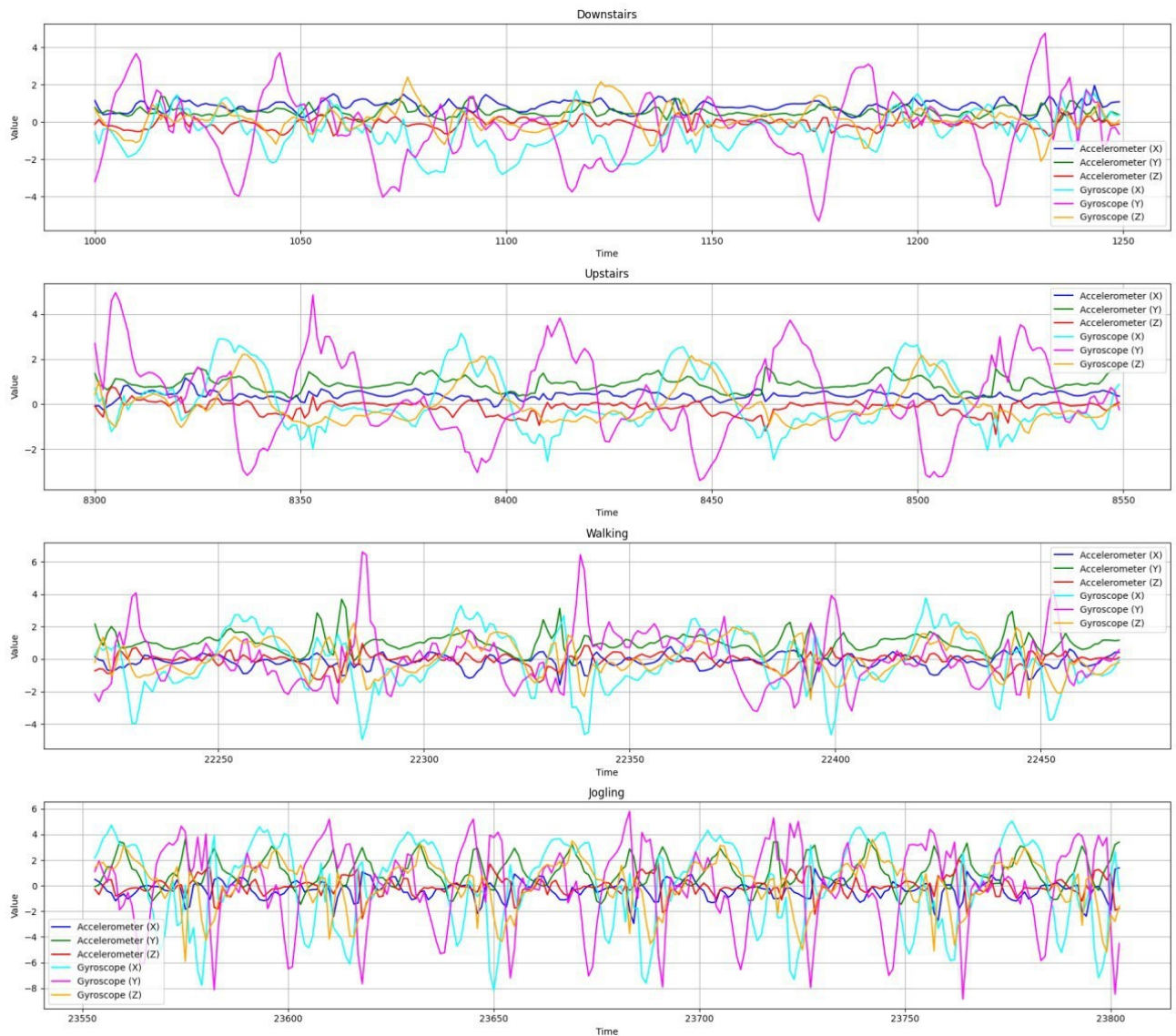


Рисунок 3.1 часові ряди, які відповідають за різні активності об'єкта 1.

Для кожної спроби дослідник налаштовував телефон і вручав його поточному учаснику, після чого дослідник стояв у кутку. Потім учасник натискував кнопку запуску додатка Crowdsense і клав її в передню кишеню штанів, виконуючи вказану активність. Їх просили робити це як найбільш природньо, схоже на їхнє щоденне життя. В кінці кожної спроби вони виймали телефон з кишені та натискали кнопку зупинки.

Статистика даних наведена в таблиці 3.1.

Таблиця 3.1. Статистика персональних даних

	mean	std	min	25%	75%	max
Вага	72.1	16.02	48.0	60.0	80.5	102.0
Зріст	174.2	8,90	161.0	164.8	180.0	190.0
Вік	28,80	5,40	18.0	25.0	31.2	46.0
Стать	0.6	0.5	0.0	0.0	1.0	1.0

Отже, нам доступні поведінкові дані акселерометра та гіроскопа, які представлені у вигляді часових рядів з вимірюваннями прискорення та обертання. З цими даними ми будемо визначати активність людини, тобто визначати, яку конкретну фізичну діяльність вона виконує у певний момент часу. Ми можемо це зробити завдяки тому, що давачі акселерометра та гіроскопа реєструють зміни в прискоренні та обертанні пристрою. Різні активності, такі як ходьба, біг, сидіння та інші, супроводжуються характерними шаблонами руху та коливаннями. Наприклад, при ходьбі можливі ритмічні коливання, а при сидінні – мінімальний рух.

3.2 Підготовка даних для тренування моделей

У процесі підготовки даних з датасету MotionSense було реалізовано наступні кроки:

1. Розподіл даних

Дані були випадковим чином розподілені на тренувальну та тестову вибірки. Навчальна вибірка включала поведінкові дані від 20 учасників, тоді як тестова вибірка містила дані 4 осіб, з яких 2 – чоловіки, а 2 – жінки.

2. Обробка аномальних даних

На початку та в кінці кожної сесії запису активності були виявлені аномальні викиди даних. Це зумовлено тим, що в ці моменти на мобільному пристрої запускалася програма для запису даних з датчиків, а сам пристрій

поміщався в кишеню. Для усунення цього впливу з кожного запису було відкинуто по 2 секунди на початку та в кінці.

3. Розбиття даних на сегменти

Кожна сесія була поділена на відрізки, що склалися зі 100 одиниць вимірювань. З огляду на частоту дискретизації 50 Гц, такий відрізок відповідає 2 секундам запису.

4. Розширення вибірки

Для підвищення різноманітності даних було застосовано метод вибору семплів часових рядів за допомогою ковзного вікна. При цьому 75% попереднього семплу перекривалися з новим, що дозволило забезпечити більшу кількість перекриваючих відрізків у вибірці.

Цей підхід забезпечив якісну підготовку даних для подальшого використання в моделі машинного навчання.

3.3 Вибір архітектури моделі

У розділі 2 ми проаналізували існуючі методи навчання та моделей машинного навчання та розібрали найпопулярніші архітектури та нейронні мережі. В результаті було вирішено, що варіаційний автокодувальник може бути використаний для вирішення задачі розпізнавання активності людини. Також в результаті аналізу робіт ми побачили, що LSTM – одна з найкращих нейронних мереж для вирішення задачі розпізнавання активності людини.

Також у [28] авторами було використано варіаційний рекурентний автокодувальник на базі LSTM мереж, а також був отримана низька похибка для задачі біометричної верифікації.

Тому для нашого дослідження розпізнавання активності за допомогою методів машинного навчання ми вирішили використати генеративну мережу, а

саме варіаційний автокодувальник, який буде побудований на базі рекурентних нейронних мереж LSTM та GRU.

3.4 Система розпізнавання активності людини

Для вирішення завдання розпізнавання активності за допомогою варіаційного автокодувальника буде використано його здатність виявляти аномалії у процесі відновлення даних.

З цією метою для кожного типу активності, яку необхідно розпізнати, створюється окрема модель, спеціалізована саме на цій активності. Таким чином, коли система отримує вхідні поведінкові дані, найменша похибка відновлення спостерігатиметься у моделі, яка була навчена на даних, що відповідають відповідній активності.

Архітектура системи представлена на малюнку 3.2

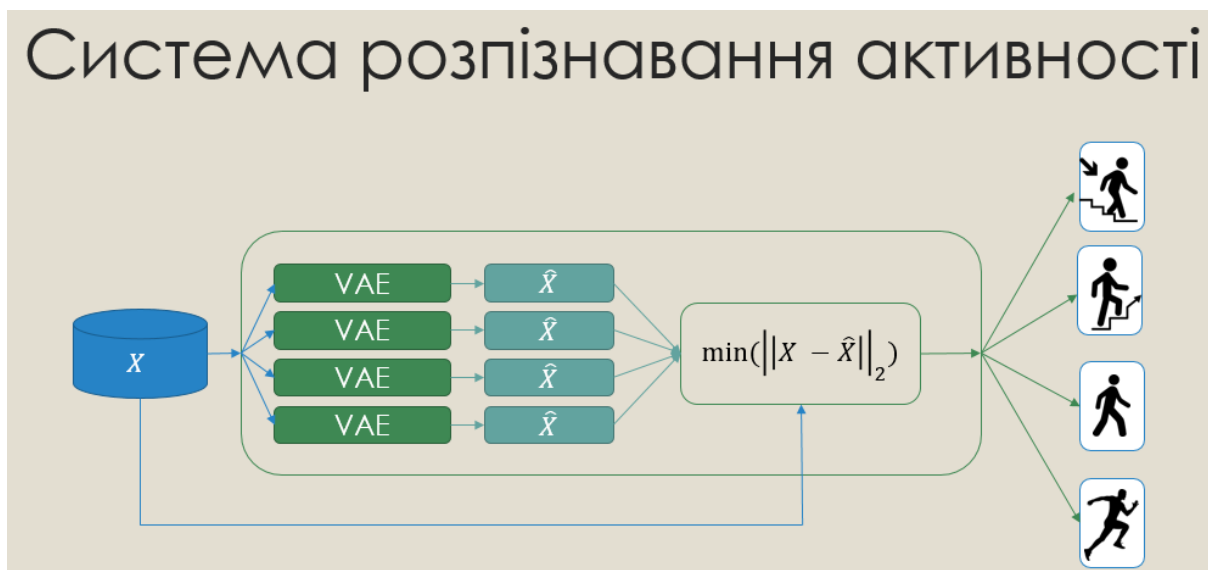


Рисунок 3.2 система розпізнавання активності

3.5 Метрики

Для того, щоб проаналізувати результати, нам потрібно визначити метрики, якими ми виміряємо модель.

Найкраще для оцінювання задачі бінарної класифікації підходить матриця невідповідностей. Її основні компоненти це – **TP**, **TN**, **FP** і **FN**. Вони описують, як модель класифікує поведінкові дані, і є основою для розрахунку багатьох метрик.

Опишемо, що означає кожна з компонент:

TP (True Positives, Істинно позитивні): це кількість випадків, коли модель правильно передбачила позитивний клас.

TN (True Negatives, Істинно негативні): це кількість випадків, коли модель правильно передбачила негативний клас.

FP (False Positives, Хибно позитивні): це кількість випадків, коли модель помилково передбачила позитивний клас, хоча насправді клас негативний.

FN (False Negatives, Хибно негативні): це кількість випадків, коли модель помилково передбачила негативний клас, хоча насправді клас позитивний.

Для оцінювання результатів задачі класифікації найчастіше використовуються наступні метрики:

Точність (Accuracy) – це частка правильно класифікованих прикладів серед усіх прикладів.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Її використовують коли класи збалансовані (наприклад, однакова кількість позитивних і негативних прикладів) і коли немає критичного значення, який тип помилок важливіше (FP чи FN).

Однак вона має свої недоліки: для **незбалансованих класів** метрика може вводити в оману, оскільки навіть модель, яка завжди прогнозує більший клас, може мати високий Accuracy.

Влучність (Precision) – це частка правильно зразків, правильно відмічених як належні до позитивного класу.

$$Precision = \frac{TP}{TP + FP}$$

Її використовують коли важливо зменшити кількість FP (помилкових позитивів). Однак вона має свої недоліки: ігнорує помилки, пов'язані з FN.

Повнота (Recall) – це сума істинно позитивних та хибно негативних, що є зразками, як не було відмічено як належні до позитивного класу, але повинно було бути відмічено.

$$Recall = \frac{TP}{TP + FN}$$

Її використовують коли важливо зменшити кількість FN. Однак вона має свої недоліки, схожі на недоліки влучності: ігнорує помилки, пов'язані з FP.

F1-Score – це гармонічне середнє влучністю та повнотою. Вона забезпечує баланс між цими двома метриками.

$$F1 - score = 2 \frac{Precision \times Recall}{Precision + Recall}$$

Її використовують коли існує незбалансованість між класами і коли важливо знайти баланс між влучністю та повнотою.

3.6 Рекурентний варіаційний автокодувальник на базі LSTM нейронних мереж.

3.6.1 Архітектура мережі

Архітектура цієї мережі зображена на малюнках 3.3 а – кодувальник та 3.3 (б) – декодувальник.

Кодувальник складається з двох послідовних LSTM-шарів:

Перший шар LSTM приймає вхідні дані розмірності 6×512 та обробляє їх, знижуючи розмірність.

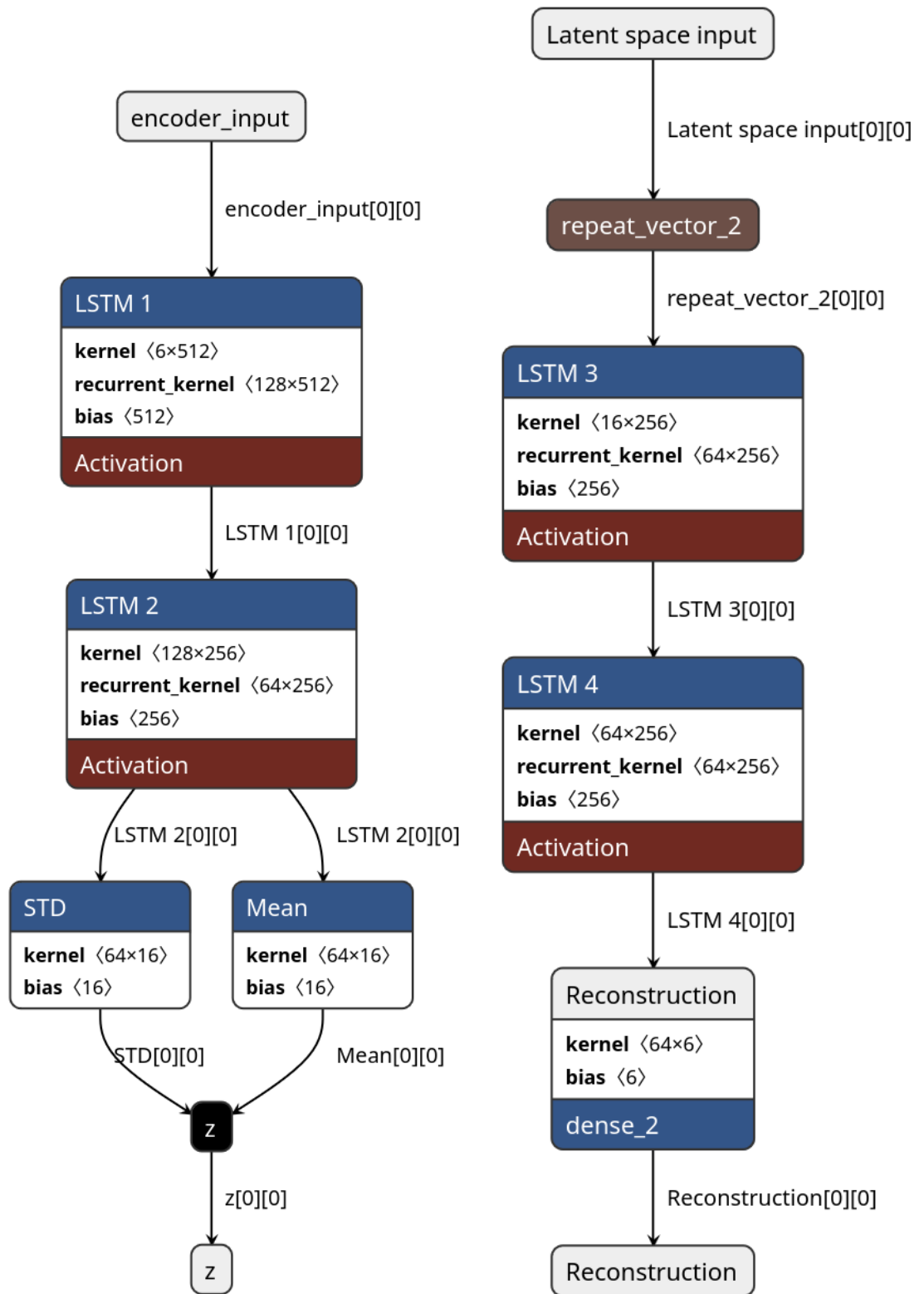


Рисунок 3.3 (а) Архітектура кодувальника, (б) архітектура декодувальника

Після цього використовується функція активації для подальшої обробки сигналу. Другий LSTM шар отримує дані від першого та додатково зменшує

розмірність до 128×256 . Він забезпечує компактніше приховане представлення даних.

Далі приховані стани розщеплюються на два окремі шари:

- **Mean:** обчислює середнє значення для латентного вектору.
- **STD:** обчислює стандартне відхилення для латентного вектору.

Ці значення використовуються для зразкового представлення в латентному просторі, де генерується вектор z шляхом репараметризації

Декодувальник складається з двох послідовних LSTM-шарів і шарів реконструкції:

Repeat Vector: розширює вектор z із латентного простору, щоб підготувати його до обробки LSTM-шарами.

Перший LSTM шар декодера приймає вектор z розмірності 16×256 і починає розгортати його у вигляді послідовності. Другий шар LSTM завершує формування прихованого представлення, яке відповідає оригінальній послідовності.

Останній це Dense-шар: на виході декодера використовується шар Dense, який реконструює вихідні дані на основі прихованих станів.

3.6.2. Тренування моделей

Навчання проводилося за допомогою оптимізатора Adam [30] протягом 100 епох. Ми використали навчання батчами, кожен батч мав у собі 32 зразка даних.

Також для кращого навчання ми модифікували функцію втрат варіаційного автоенкодера (2.6.1), додавши до неї константу:

$$\mathbb{L} = \mathbb{L}_{recon} + \alpha \times \mathbb{L}_{KL}$$

Ця константа α була додана через проблему з навчанням, коли відхилення Кульбака-Лейблера в один момент може стати дуже великим і

навчання моделі зупинялося. Ця проблема і її рішення були описані в роботі [30].

Навчання моделей зображено на малюнку 3.4:

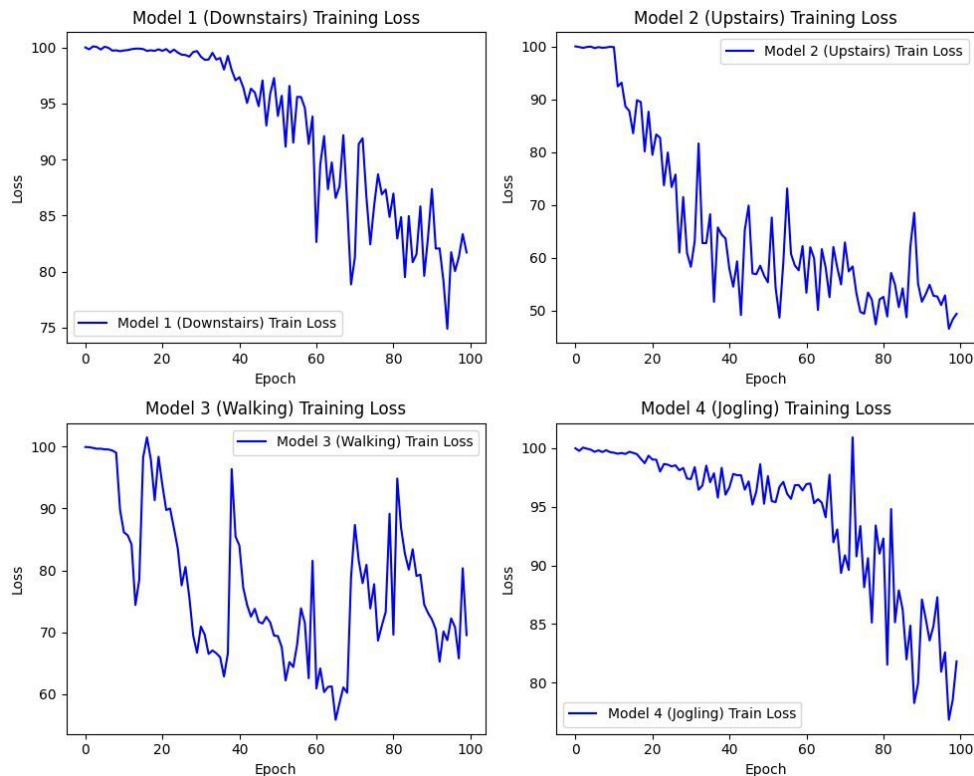


Рисунок 3.4 – навчання моделей для кожного типу активності.

Після навчання перевіряємо точність нашої моделі за наступним алгоритмом:

1. Обробка вхідних даних: кожен варіаційний автокодувальник відновлює данні.

2. Обчислення похибки реконструкції: для кожної моделі розраховується середньоквадратична похибка (MSE) між вхідними даними та їх реконструйованими версіями, отриманими на виході автокодувальника. Ця похибка характеризує якість відновлення даних.

3. Вибір моделі з найменшою похибкою: серед усіх отриманих похибок визначається найменша. Модель з найменшою похибкою реконструкції

вважається тією, що найбільш відповідно відображає вхідну активність, і, отже, ідентифікує її.

В результаті тестування було отримано такі метрики, які показані на малюнку 3.5:

	precision	recall	f1-score	support
Downstairs	0.46	0.85	0.60	2242
Upstairs	0.67	0.74	0.70	2691
Walking	0.93	0.65	0.76	5556
Jogling	0.88	0.68	0.77	2195
accuracy			0.71	12684
macro avg	0.73	0.73	0.71	12684
weighted avg	0.78	0.71	0.72	12684

Рисунок 3.5 Пораховані метрики для кожної з активностей

Також ми отримали матрицю невідповідностей, яка зображена на малюнку 3.6.

3.6.3. Аналіз результатів моделі

Модель має низьку влучність (0.46) та високу повноту (0.85) для класу «спуск по сходах». Це означає, що модель виявляє 85% усіх реальних випадків спуску по сходах, але лише 46% її прогнозів цього класу є правильними.

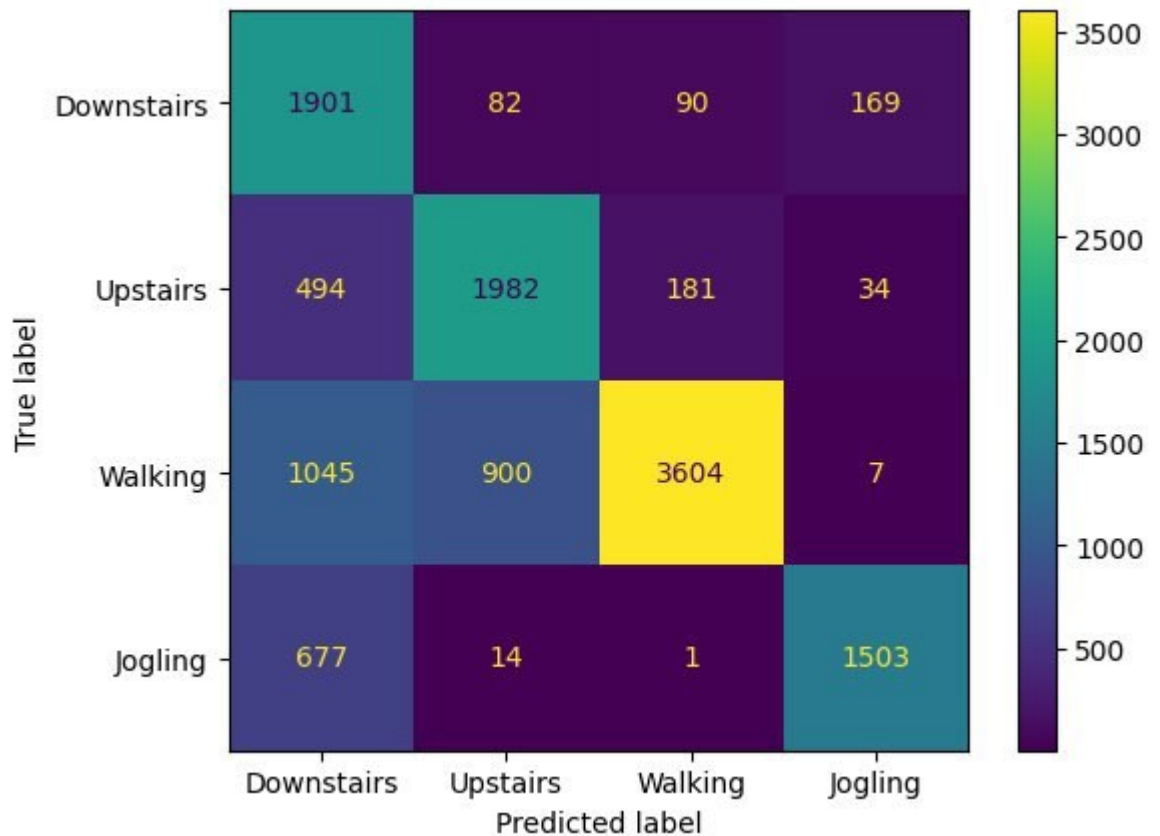


Рисунок 3.6. Матриця невідповідностей для кожної з активностей

Модель часто помилково класифікує інші активності як «спуск по сходах», що призводить до великої кількості хибнопозитивних результатів.

Для класу «підйом по сходах» модель демонструє середню влучність (0.67) та високу повноту (0.74). Це вказує на те, що модель добре виявляє реальні випадки підйому по сходах і менш схильна до хибнопозитивних помилок порівняно з «спуск по сходах».

Модель має дуже високу влучність (0.93) для класу «ходьби», що означає, що вона рідко помиляється при прогнозуванні цього класу. Однак повнота 0.65 свідчить про те, що модель пропускає 35% реальних випадків ходьби.

Для класу «біг» модель має високу точність (0.88) та помірну повноту (0.68). Це означає, що модель добре розпізнає біг підтюпцем, але пропускає близько 32% реальних випадків.

Модель на базі LSTM демонструє збалансовану продуктивність з трохи кращими результатами для класів «підйом по сходах" та «ходьба».

3.7 Рекурентний варіаційний автокодувальник на базі GRU нейронних мереж

3.7.1 Архітектура мережі

Архітектура цієї мережі зображена на малюнках 3.7 (а) – кодувальник та 3.7 (б) – декодувальник.

Кодувальник складається з двох послідовних GRU-шарів, які стискають вхідні дані до латентного простору:

Перший GRU-шар приймає вхідні дані розмірності 6×384 та знижує їх розмірність, зберігаючи важливу інформацію. Шар активується за допомогою функції активації.

Другий GRU-шар приймає вихідні дані від першого GRU та додатково зменшує розмірність до 128×192 .

Після проходження через GRU-шари, приховані стани розділяються на два окремі шари:

- **Mean:** обчислює середнє значення для латентного вектору.
- **STD:** обчислює стандартне відхилення для латентного вектору.

Ці два параметри використовуються для репараметризації, яка генерує латентний вектор z .

Декодувальник складається з двох послідовних GRU-шарів та шару реконструкції, який відновлює вихідні дані:

Repeat Vector: Розширює латентний вектор z до необхідного формату для подальшої обробки GRU.

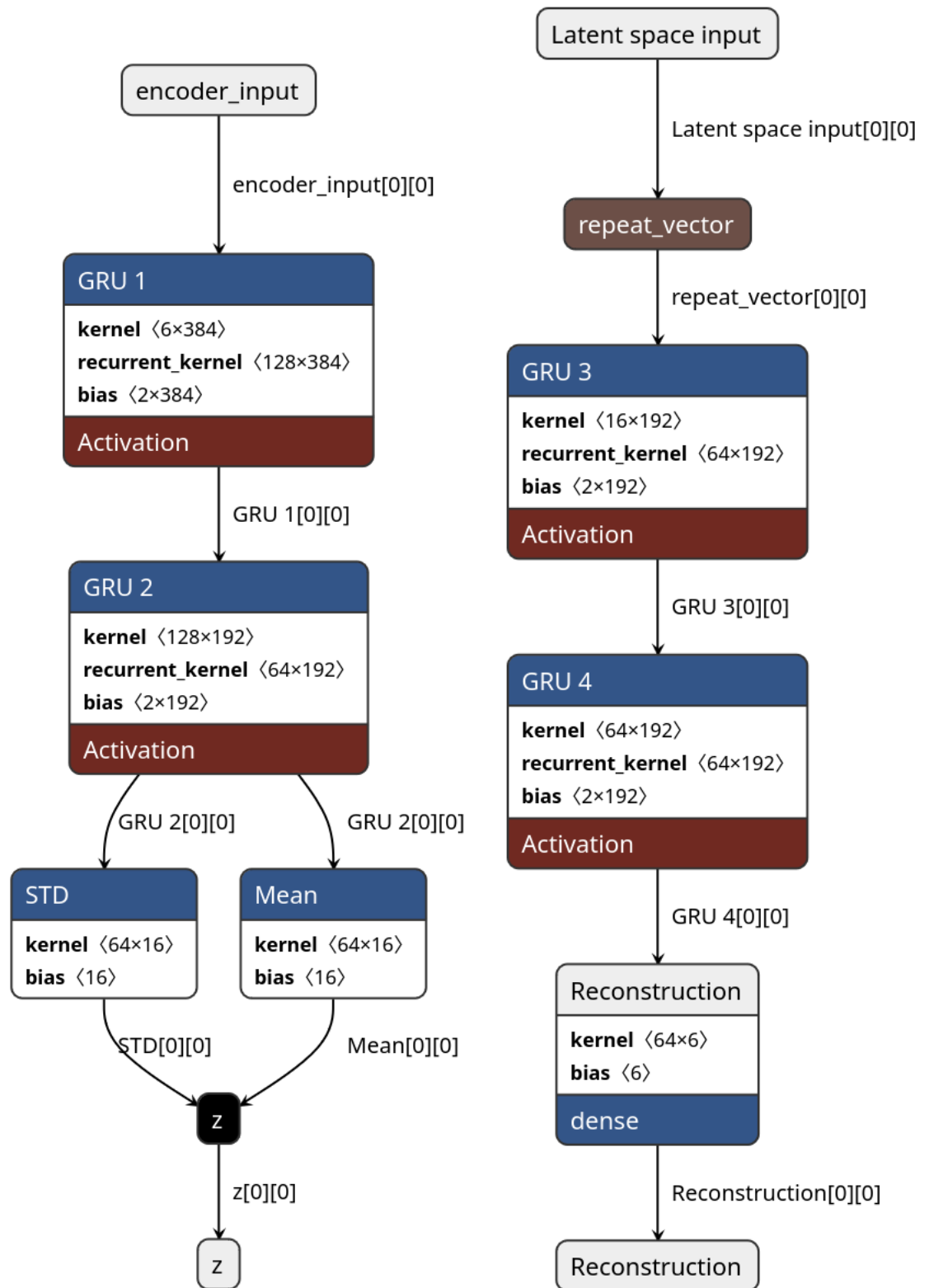


Рисунок 3.7. а) Архітектура кодувальника, б) архітектура декодувальника

Перший GRU-шар у декодері приймає розширений вектор z розмірності 16×192 та починає його розгортання.

Другий GRU-шар завершує обробку, перетворюючи приховані стани у формат, що відповідає вихідній послідовності.

Reconstruction Dense-шар із розмірністю 64×6 використовується для реконструкції вихідних даних із прихованих станів GRU.

3.7.2 Тренування моделей

Навчання проводилося за допомогою оптимізатора Adam [30] протягом 100 епох. Ми використали навчання батчами, кожен батч мав у собі 32 зразка даних.

Також для кращого навчання ми модифікували функцію втрат варіаційного автоенкодера (2.6.1), додавши до неї константу:

$$\mathbb{L} = \mathbb{L}_{recon} + \alpha \times \mathbb{L}_{KL}$$

Ця константа α була додана через проблему з навчанням, коли відхилення Кульбака-Лейблера в один момент може стати дуже великим і великим і навчання моделі зупинялося. Ця проблема і її рішення були описані в роботі [30].

Після навчання перевіряємо точність нашої моделі за наступним алгоритмом:

1.Обробка вхідних даних: кожен варіаційний автокодувальник відновлює данні.

2.Обчислення похибки реконструкції: для кожної моделі розраховується середньоквадратична похибка (MSE) між вхідними даними та їх реконструйованими версіями, отриманими на виході автокодувальника. Ця похибка характеризує якість відновлення даних.

3.Вибір моделі з найменшою похибкою: серед усіх отриманих похибок визначається найменша. Модель з найменшою похибкою реконструкції вважається тією, що найбільш відповідно відображає вхідну активність, і, отже, ідентифікує її.

В результаті тестування було отримано такі метрики, які показані на малюнку 3.8. Також ми отримали матрицю невідповідностей, яка зображена на малюнку 3.9.

precision	recall	f1-score	support	
Downstairs	0.55	0.88	0.68	2242
Upstairs	0.52	0.78	0.62	2691
Walking	0.94	0.59	0.72	5556
Jogling	0.94	0.69	0.80	2195
accuracy			0.70	12684
macro avg	0.74	0.73	0.71	12684
weighted avg	0.78	0.70	0.71	12684

Рисунок 3.8 Пораховані метрики для кожної з активностей



Рисунок 3.9. Пораховані метрики для кожної з активностей

3.7.3 Аналіз результатів моделі

Для класу «спускання по сходах» модель має низьку влучність (0.55) та високу повноту (0.88). Це означає, що модель правильно виявляє 88% усіх реальних випадків спуску по сходах, але коли вона прогнозує «спускання по сходах», лише в 55% випадків цей прогноз є правильним. Іншими словами, модель часто помилково класифікує інші активності як «спускання по сходах», що призводить до великої кількості хибнопозитивних результатів.

Для класу «підйом по сходах» модель також демонструє низьку влучністю (0.52) та відносно високу повноту (0.78). Це свідчить про те, що модель виявляє 78% усіх реальних випадків підйому по сходах, але коли вона прогнозує «підйом по сходах», лише в 52% випадків цей прогноз є правильним. Це означає, що модель часто помилково класифікує інші

активності як «підйом по сходах», збільшуючи кількість хибнопозитивних прогнозів.

Для класу «ходьба» модель має дуже високу влучність (0.94) та низьку повноту (0.59). Це означає, що коли модель прогнозує «ходьба», вона майже завжди права (94% прогнозів є правильними). Однак вона виявляє лише 59% усіх реальних випадків ходьби, пропускаючи 41% цих випадків. Це свідчить про те, що модель консервативна у прогнозуванні цього класу та часто не розпізнає «ходьба», коли це відбувається.

Для класу «біг» модель також має високу влучність (0.94) та помірну повноту (0.69). Це означає, що модель правильно прогнозує 94% випадків, коли вона вважає, що відбувається біг підтюпцем, але виявляє лише 69% усіх реальних випадків «біг». Модель пропускає 31% цих випадків, що вказує на можливість покращення виявлення цього класу.

Модель демонструє загальну точність 70%, що є прийнятним результатом, але показує дисбаланс між влучністю та повнотою для різних класів:

Для класів «спуск по сходах» та «підйом по сходах»: висока повнота та низька влучність означають, що модель добре виявляє реальні випадки цих активностей, але часто помилково класифікує інші дії як спуск або підйом по сходах.

Для класів «ходьба» та «біг»: висока влучність та низька повнота свідчать про те, що модель рідко помиляється, коли прогнозує ці класи, але пропускає значну кількість реальних випадків цих активностей.

3.8 Порівняння метрик моделей

Модель на базі GRU має вищу влучність (0.55 проти 0.46) та F1-міру для класу «спуск по сходах», хоча повнота трохи вища у LSTM моделі.

Модель на базі LSTM має трохи нижчу точність, але вищу повноту та F1-міру для класу «ходьба», що означає краще виявлення реальних випадків ходьби.

Модель на базі GRU перевершує LSTM модель за всіма метриками для класу «біг».

Загальний порівняльний висновок:

- GRU модель краще справляється з класами «спуск по сходах» та «біг».
- LSTM модель демонструє кращі результати для класів «підйом по сходах» та «ходьба».
- Загальна точність моделей майже однакова: 70% для GRU та 71% для LSTM.

Загальні висновки щодо методу:

Некероване навчання: VAE можуть навчатися на немічених даних, що знижує потребу в мічених зразках.

Витягування релевантних ознак: VAE здатні навчитися прихованим представленням даних, що можуть бути більш інформативними для класифікації.

3.9 Висновки до розділу 3

Результати показують, що розроблена система розпізнавання активності людини за її поведінковими даними побудована на базі автокодувальників є перспективним підходом, оскільки ми отримали метрики, які вже дозволяють використовувати модель (точність 70%), а архітектура системи спроектована для розширення новими активностями без повного перетренування інших моделей.

Таким чином варіаційні автокодувальники на базі GRU та LSTM є ефективними інструментами для розпізнавання активності людини. Подальші дослідження та оптимізація можуть підвищити точність та надійність моделей у реальних застосуваннях.

РОЗДІЛ 4 РОЗРОБКА ВЛАСНОГО СТАРТАП-ПРОЄКТУ

У сучасному світі стає все більш важливим розробка систем, які здатні розпізнавати та аналізувати активності людини за допомогою отриманих даних з акселерометру та гіроскопу з wearable-пристроїв. Існує багато підходів до цього завдання, однак лише декілька систем задовольняють вимоги широкого кола користувачів та дослідників. Ця проблема вимагає використання складних алгоритмів та обчислювальних ресурсів, а також значної кількості даних для навчання та тестування. Незважаючи на це, попит на технології, які забезпечують точне розпізнавання активностей людини, залишається високим у сферах фітнесу, медицини, технологій для здоров'я та багатьох інших.

Це означає, що проблема, яку вирішує стартап, є надзвичайно актуальною та може стати важливим рішенням для широкої аудиторії, включаючи виробників wearable-пристроїв, розробників програмного забезпечення для фітнесу, а також у сферах медицини та здоров'я.

4.1 План розробки стартапу та масштабування його на ринок

Наведемо план розробки стартапу та виведення його на ринок.

Спочатку треба провести маркетинговий аналіз, який включає в себе:

- конкурентний аналіз, щоб зрозуміти, якими методами вирішення проблем вже користуються люди;
- формування ідеї самого проекту та виділення цільової аудиторії;
- розробити стратегію виведення товару на ринок, базуючись на аналізі ринкового середовища.

Наступним кроком являється організація самого стартапу. На цьому етапі мають бути:

- складений весь план та побудований таймлайн розробки та запуску продукту;
- запланований обсяг виробництва та оцінений потенційний обсяг ресурсу, який буде потрібен для виконання плану;
- розраховані витрати, необхідні для реалізації проекту, та витрати на запуск проекту.

Далі необхідно виконати фінансово-економічний аналіз та оцінити ризики стартап-проекту, в межах якого:

- визначити обсяг інвестиційних втрат;
- розрахувати основні фінансово-економічні показники проекту (собівартість, ціну продукту/послуги, податковий збір та чистий прибуток) та визначити показники інвестиційної привабливості проекту (рентабельність продажів, період окупності проекту);
- визначити основні ризики проекту та способи для їх запобігання.

Фінальним кроком являється розробка заходів з комерціалізації продукту. Цей крок являється важливим для масштабування та збільшення розмірів продукту.

Для того, щоб залучити інвесторів та знайти різні способи фінансування проекту, необхідно:

- провести дослідження на предмет інтересів потенційних інвесторів та бізнесів;
- скласти інвестиційну пропозицію, яка включає в себе як опис самого продукту та його теперішні розміри, так і можливі шляхи розширення та розвитку;
- обрати канали комунікації із потенційно зацікавленими персонами.

Далі наведемо результати виконання кожного з описаних кроків.

4.2 Опис ідеї стартап-проекту

Стартап-проект полягає у розпізнавання активності людини на основі даних з давачів, які можуть знаходитися в фітнес трекері. Суть продукту полягає в тому, що користувач носить фітнес-трекер, а на базі його активності наша програма вже може давати персональні рекомендації, наприклад «вам треба більше рухатися».

У таблиці 4.1 наведена інформаційна карта стартапу.

Таблиця 4.1 – Інформаційна карта стартап-проекту

Назва проекту	MotionInsight
Автори проекту	Каленіченко Денис Олександрович
Коротка анотація	MotionInsight - це інноваційний стартап, який використовує дані з акселерометру та гіроскопу, отримані з wearable-пристроїв, для розпізнавання та аналізу активностей людини. Наша технологія вирізняється від усіх інших тим, що ми використовуємо передові методи машинного навчання для точного визначення того, що саме робить користувач
Термін реалізації проекту	12 місяців

Продовження таблиці 4.1

Необхідні ресурси	– Приміщення з комп'ютерами, доступом до Інтернету, доступ до електромережі – Програмне забезпечення для розробки, хмарне сховище для даних, антивірусні програми – Фінансові кошти на оплату заробітної плати виконавцям на термін 12 місяців, а також на такі витрати як: оренда приміщення, комунальні послуги, оренда хмарного сховища тощо – Набір девайсів, які мають акселерометр та гіроскоп – Площадка для виконання різних дій
Опис проблеми, яку вирішує проект	Розпізнавання активності людини
Головні цілі та завдання проекту	Метою проекту є створення системи, яка буде автоматично обробляти данні з wearable-device або смартфона та визначати, чим саме займається людина
Очікувані результати	Високі результати розпізнавання активності

4.3 Технологічний аудит ідеї проекту

Тепер можна розібрати ідею стартапу та провести конкурентний аналіз. У таблиці 4.2 наведений опис ідеї стартапу.

Таблиця 4.2 – Опис ідеї стартапу

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка комплексної системи для обробки даних з акселерометру та гіроскопу та аналізу активностей людини Система збору та аналізу фізичної активності для подальшого вдосконалення та оновлення	Аналіз фізичної активності та здоров'я, фітнес-технології, медицина	Отримання точних та персоналізованих даних про активність
	Покращення алгоритмів розпізнавання активностей, оновлення системи з урахуванням потреб користувачів	Постійне покращення точності та персоналізованість у визначенні фізичної активності

Далі проведемо порівняльний аналіз конкурентів проекту та наведемо результати: наразі конкуренція відсутня.

Далі аналізуємо реальність технічно здійснити ідею проекту (таблиця 4.4).

Таблиця 4.4 – Технологічна здійсненність продукту

п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
	Розробка комплексної системи для обробки даних з акселерометру та гіроскопу та аналізу активностей людини	Використання мови програмування Python	Наявні	Доступні
	Система збору та аналізу фізичної активності для подальшого вдосконалення та оновлення	Використання мови програмування Rust	Наявні, необхідні допрацювання	Доступні
Обрана технологія реалізації ідеї проекту: Python				

4.4 Аналіз ринкових можливостей запуску стартап-проекту

Далі проведемо попередній аналіз ринку для запуску стартап-проекту (таблиця 4.5).

Таблиця 4.5 – Попередня характеристика потенційного ринку стартап-проекту

п/п	Показники ринку (найменування)	Характеристика
	Кількість головних гравців, од	3
	Загальний обсяг продаж, грн/ум.од	6000
	Динаміка ринку (якісна оцінка)	Позитивна, зростає
	Наявність обмежень для входу (вказати характер обмежень)	Відсутні
	Специфічні вимоги до стандартизації та сертифікації	Відсутні
	Середня норма рентабельності в галузі (або по ринку), %	12%

Тепер проведемо характеристику потенційних клієнтів, які можуть бути зацікавлені в проекті (таблиця 4.6).

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту

п/п	Потреби, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Розпізнавання активностей з даних акселерометру та гіроскопу	Фітнес-ентузіасти, медичні технології, розробники фітнес-додатків	Різні потреби у точності визначення та виділенні різних видів активностей	Точність визначення активностей та зручність в користуванні
	Аналіз активностей для відеоспостереження	Державні та правоохоронні органи	Особлива потреба в збереженні якості деталей та ідентифікації об'єктів	Висока якість відтворення та деталізація зображень
	Розпізнавання активностей з даних акселерометру та гіроскопу	Фітнес-тренери, спортивні заходи, медичні дослідження	Різні вимоги до точності визначення різних типів активностей	Точність визначення та адаптованість до різних потреб користувачів

Обрахуємо фактори загроз (таблиця 4.7) та можливостей (таблиця 4.8). Проаналізуємо загрози, щоб зрозуміти можливі перешкоди при запуску

продукт на ринок. Фактори можливостей же треба обрахувати, щоб знати усі сприятливі умови та по можливості ними скористатися.

Таблиця 4.7 – Фактори загроз

п/п	Фактор	Зміст загрози	Можлива реакція компанії
	Конкуренція	На ринку присутні великі гравці зі своєю власною цільовою аудиторією	Знаходження додаткових переваг та значущості для користувача
	Ціна збуту	Конкуренти можуть пропонувати менші ціни через можливу низьку якість продукту	Зосередження на якості продукту та розробка маркетингової стратегії
	Якість аналізу	Комплексні завдання можуть призвести до проблем у різних областях	Розширення штату та ресурсів для розробки моделей для різних областей

Таблиця 4.8 – Фактори можливостей

п/п	Фактор	Зміст можливості	Можлива реакція компанії
	Універсальність	Незалежність від апаратної платформи	Зосередження на маркетингу та розвиток як унікального продукту
	Простота у використанні	Легкість завантаження файлів зображень користувачами	Розробка зручного інтерфейсу завантаження файлів
	Якість та гарантії	Надання високоякісних послуг та сервісів	Пропозиція моделей з найкращими результатами та технічною підтримкою
	Безкоштовний сервіс при MVP	Швидкий набір клієнтів та встановлення позицій на ринку	Широкий маркетинг та активне залучення клієнтів від конкурентів

Далі розглянемо питання конкуренції, а саме визначимо її тип та рівень (таблиця 4.9).

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції: недосконала	Мало продуктів та експертів в цій сфері	Максимізувати продажі розробленого застосунку
2. Рівень конкурентної боротьби: міжнародний	Наявність проектів, доступних у всьому світі	Розширити цільову аудиторію, розробити інтерфейс на різних мовах
3. Галузева ознака: внутрішньогалузева	Можливість працювати з різними галузями	Підвищити рівень персоналізації
4. Конкуренція за видами товарів: товарно-родова	Конкуренція з іншими системами та експертами	Підтримка та покращення якості існуючих функцій
5. Характер конкурентних переваг: нецінова	Різна пропозиція від різних компаній	Розробка більш якісних алгоритмів та моделей
6.Інтенсивність конкуренції: марочна	Присутність компаній із визнаним брендом	Розвиток комунікаційної стратегії для формування власного бренду

Далі необхідно виконаємо аналіз конкуренції за моделлю 5 сил конкуренції Майкла Портера (таблиця 4.10).

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти у галузі	Потенційні конкуренти	Постачальники	Клієнти	Товарозамінники
	Інші існуючі системи та продукти	Якість, ціни, кількість користувачів, капіталовкладення	Фактори сили постачальників	Контроль якості, порівняння цін	Сила бренду, якість, ціна, масштаби
Висновки	Конкуренція з невеликою інтенсивністю, а також підігрітий ринок	Можливості входження на ринок, нові потенційні конкуренти	Постачальники відсутні	Клієнти не диктують умови роботи на ринку	Товарозамінники відсутні

Маючи результати аналізу конкуренції (таблиця 4.10), характеристики ідеї стартап-проекту (таблиця 4.5), характеристики потенційних клієнтів і їх вимоги до продукту (таблиця 4.6) та фактори ринкового середовища (таблиці 4.7 і 4.8) було сформульовано та обґрунтовано перелік факторів конкурентоспроможності (таблиця 4.11).

Таблиця 4.11 – Обґрунтування факторів конкурентоспроможності

п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
	Універсальність	Продукт не залежить від апаратної платформи як у більшості конкурентів
	Простота у використанні	Від користувача треба лише завантажити файл зображення
	Якість та гарантії	Надавати найбільш якісні послуги та сервіси
	Безкоштовний сервіс при MVP	Макимально швидко набрати базу своїх клієнтів та заявити про себе на ринку

Тепер можна провести аналіз сильних та слабких сторін продукту (таблиця 4.12).

Таблиця 4.12 – Порівняльний аналіз сильних та слабких сторін системи

№ п/п	Фактор конкурентоспроможності	Бали 1- 20	Рейтинг товарів-конкурентів						
			3	2	1				
1	Універсальність	20							
2	Простота у використанні	16							
3	Якість та гарантії	10							
4	Безкоштовний сервіс при MVP	17							

Далі проведемо SWOT-аналіз продукту (таблиця 4.13).

Таблиця 4.13 – SWOT-аналіз стартап-проекту

Сильні сторони Універсальність Простота у використанні Якість та гарантії Безкоштовний сервіс при MVP	Слабкі сторони Відсутність сильного бренду Не сформована база клієнтів Не підключені альтернативні канали маркетингу
Можливості Покращення системи Персоналізація Інтеграція з цифровими фоторедакторами чи засобами цифрової фотографії	Загрози Нові системи та експерти Збут

Дякуючи проведенню SWOT-аналізу, ми змогли визначити сильні та слабкі сторони, можливості та загрози, пов'язані з конкуренцією та плануванням стартап-проекту. Далі спроектуємо альтернативну ринкову поведінку для інтеграції стартап-проекту на ринок та приблизний час реалізації системного комплексу, з урахуванням потенційних проектів, що можуть бути виведені на ринок та наведемо результати у таблиці 4.14.

Таблиця 4.14 – Альтернативи ринкового впровадження стартап проекту

п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
	Вихід на ринок з нижче якістю	70%	4 місяці
	Пропонувати одразу платне використання	50%	6 місяців
	Представлення користувачам системи без інтерфейсу	60%	5 місяці

У даному пункті був проведений детальний аналіз ринку та продукту. Також відповідно до результатів проведеного конкурентного аналізу, визначених факторів ринку та його сприятливості, описання ідеї та характеристик стартап-проекту, робимо висновок, що існують дуже сприятливі умови для виходу продукту на ринок.

4.5 Розроблення ринкової стратегії стартап-проекту

Для розробки ринкової стратегії продукту, у першу чергу, необхідно проаналізувати цільову аудиторію проекту (таблиця 4.15).

Таблиця 4.15 – Вибір цільових груп потенційних споживачів

п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит у межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
	Персональні користувачі	Висока	25%	Висока	Середня
	Великі бізнеси	Середня	20%	Середня	Середня
	Малі та середні бізнеси	Середня	20%	Середня	Середня
	Держава	Низька	10%	Низька	Висока
Які цільові групи обрано: 1, 3					

Маючи аналіз цільових груп, далі визначимо базову стратегію розвитку продукту (таблиця 4.16).

Таблиця 4.16 – Визначення базової стратегії розвитку

п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку у*
	1 та 3	Диференційованого маркетингу	Масштабування та максимізація	Оптимізація витрат

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиці 4.17, 4.18).

Таблиця 4.17 – Визначення базової стратегії конкурентної поведінки

Чи є проект «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 конкурентної поведінки*
Ні	Так	Ні	Виклику лідера

Таблиця 4.18 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспро- можні позиції власного стартап- проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Універсальність Простота у використанні Якість результатів	Оптимальних витрат	Універсальність Простота у використанні Якість та гарантії Безкоштовне використання при MVP	Система, яка краще всіх покращує якість користувацьких зображень Система з простим інтерфейсом

4.6 Розроблення маркетингової програми стартап-проекту

Після проведеного комплексного аналізу, можемо повноцінно описати ключові переваги концепції потенційного товару (таблиця 4.19) та побудувати концепцію маркетингових комунікацій (таблиця 4.20).

Таблиця 4.19 – Ключові переваги концепції потенційного товару

п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
	Якісна генерація	Якісне збільшення зображення	Постійне покращення та перенавчання моделей, які зможуть аналізувати більше прикладних областей
	Універсальність	Система не залежить від апаратної платформи цифрової фотографії	Такого виду систему може використовувати будь який користувач
	Простий інтерфейс	Система дуже проста у використанні	Система із інтуїтивно зрозумілим інтерфейсом, який вимагає всього лише завантаження аудіо файлу

Таблиця 4.20 – Концепція маркетингових комунікацій

п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Пошук спеціалізованих систем	b2b продажі Зв'язок через теплі контакти Таргетована реклама у соціальних мережах Публікація в спеціалізованих виданнях, журналах	Точність Якість Універсальність	Поєднати повідомлення про те, що це якісна система, яка є незалежною	Таргетована реклама на цільову аудиторію
	Пошук доступного та дешевого продукту	Рекламні банери в Інтернеті, форуми, реклами від інфлюенсерів	Простота Безкоштовне використання MVP	Вселити довіру у бренд та продукт	Реклама у лідерів думок Вивіски в публічних місцях Таргетована реклама на цільову аудиторію

4.7 Висновки до розділу 4

Цей розділ стосується аналізу стартап-проекту, який представляє систему для аналізу активностей людини за допомогою даних з акселерометру та гіроскопу. Розглядалися стратегії виходу на ринок та маркетингу. У цій області конкуренція не настільки велика, оскільки існуючі системи надають лише обмежений функціонал. Оскільки наша система універсальна та доступна, у нашому стартапі є потенціал стати лідером на ринку.

Також були проаналізовані переваги та недоліки проекту, проведений SWOT-аналіз, аналіз конкурентів та цільової аудиторії. На основі цих досліджень розроблено концепцію маркетингової стратегії для обраної цільової аудиторії.

ВИСНОВКИ

Магістерська дисертація присвячена дослідженню генеративних мереж, а саме варіаційного автокодувальника, який використовується для вирішення задачі розпізнавання активності на базі поведінкових даних давачів з носимих пристроїв.

У роботі були детально проаналізовані існуючі рішення для розв'язання задачі розпізнавання активності людини в умовах нестачі розмічених даних. На основі попередніх досліджень була розглянута найбільш перспективна архітектура нейронної мережі – варіаційний автокодувальник на базі LSTM рекурентних нейронних мереж, зважаючи на їх низьку помилку в інших задачах при роботі з поведінковими характеристиками з давачів.

Розроблено нову архітектуру системи розпізнавання активності людини, яка базується на інтеграції декількох варіаційних автокодувальників з рекурентними нейронними мережами як складовими елементами.

Розроблена система була перевірена декількома різними архітектурами варіаційного автокодувальника (на базі LSTM та GRU рекурентних нейронних мереж). Система та варіаційні автокодувальники були реалізовані мовою програмування Python з використанням фреймворку для побудови моделей машинного навчання Tensorflow.

Слід зазначити, що розроблена система демонструє значний потенціал для покращення метричних показників, а її розширення та масштабування є простішими порівняно зі стандартними моделями класифікації. Це зумовлено тим, що система використовує декілька варіаційних автокодувальників, кожен з яких спеціалізується на певній активності. Модульна архітектура системи спрощує її розширення та масштабування шляхом додавання нових моделей без необхідності переробки всієї структури, що є перевагою над стандартними моделями класифікації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Lara O. D., Labrador, M. A.. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*. 2012. 15(3), P. 1192-1209.
2. Bulling A., Blanke U., Schiele B. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*. 2014. 46(3), P. 1-33.
3. Wang J., Chen Y., Hao S., Peng, X., Hu L. Deep learning for sensor-based activity recognition: A survey. *Pattern recognition letters*, 2019. 119, P. 3-11.
4. Preece S. J., Goulermas J. Y., Kenney L. P., Howard D. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering*, 2008, 56(3), P. 871-879.
5. Yang J., Nguyen M. N., San P. P., Li X., Krishnaswamy S. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Ijcai*, 2015. Vol. 15, P. 3995-4001.
6. *CSC380: Principles of Data Science Introduction to Machine Learning*. (n.d.). URL: http://www.pacheco.j.com/courses/csc380_fall21/lectures/mlintro.pdf
7. Zhu X., Goldberg A. B. *Introduction to semi-supervised learning*. Springer Nature, 2022.
8. Goodfellow I., Bengio Y., Courville A., Deep Learning URL: <http://www.deeplearningbook.org>
9. Chen Y., Xue Y. A deep learning approach to human activity recognition based on single accelerometer. In *2015 IEEE international conference on systems, man, and cybernetics*. P. 1488-1492.
10. Hammerla N. Y., Halloran S., Plötz T. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*, 2016.

11. Wang Z., Oates T. Imaging time-series to improve classification and imputation. *arXiv preprint arXiv:1506.00327*, 2015.
12. Baldi P. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, 2012,. JMLR Workshop and Conference Proceedings. P. 37-49.
13. Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Bengio Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
14. Kingma D. P. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
15. Bengio Y., Courville A., Vincent P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 2013, 35(8), P. 1798-1828.
16. Li F., Shirahama K., Nisar M. A., Köping L., Grzegorzec M. Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors*, 2018, 18(2), 679.
17. Doersch C. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
18. Higgins I., Matthey L., Pal A., Burgess C. P., Glorot X., Botvinick M. M., Lerchner A. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 2017, 3.
19. White T. Sampling generative networks. *arXiv preprint arXiv:1609.04468*, 2016.
20. An J., Cho S. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2015, 2(1), P. 1-18.
21. LeCun Y., Bengio Y., Hinton, G. Deep learning. *nature*, 2015, 521(7553), P. 436-444.
22. Yang J., Nguyen M. N., San P. P., Li X., Krishnaswamy S. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Ijcai*, 2015, Vol. 15, P. 3995-4001.

23. Bengio Y., Simard P., Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 1994, 5(2), P. 157-166.
24. Hochreiter S., Schmidhuber J. Long Short-term Memory. *Neural Computation MIT-Press*, 1997, 9(8), P. 1735-1780.
25. Cho K. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
26. Ordóñez F. J., Roggen D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 2016, 16(1), P. 115.
27. Chung J., Gulcehre C., Cho K., Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
28. Havrylovych M., Danylov V., Gozhyj A. Comparative analysis of using recurrent autoencoders for user biometric verification with wearable accelerometer. In *CEUR Workshop Proceedings*, 2020, P. 358-370.
29. MotionSense dataset. URL: <https://github.com/mmalekzadeh/motion-sense>
30. Kingma, D. P.. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
31. Kalenichenko D., Danylov V. Методи вирішення задачі розпізнавання людської активності за допомогою генеративних мереж. *International Science Journal of Engineering & Agriculture*, 2024, 3(6), P. 10-15.
32. Каленіченко Д.О., Данилов В.Я. Методи машинного навчання для вирішення задачі Human Activity Recognition. Збірник доповідей II Всеукраїнської науково-практичної конференції «Системні науки та інформатика» 25–29 листопада 2024 року, Київ. К., НН ІПСА КПІ ім. Ігоря Сікорського, 2024, 6 с.

ДОДАТОК А

Код програми:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf

from tensorflow.keras import layers, Model
from sklearn.metrics import classification_report,
confusion_matrix, ConfusionMatrixDisplay

CONFIG_PATH = "configs/"
DATA_PATH = "data/"
RESULT_PATH = "results/"

labels = ["Downstairs", "Upstairs", "Walking", "Jogling"]

def get_codes_and_labels():
    """
    Action labels and trial codes in MotionSense dataset
    """
    ACT_LABELS = ["dws", "ups", "wlk", "jog", "std", "sit"]
    TRIAL_CODES = {
        ACT_LABELS[0]: [1, 2, 11],
        ACT_LABELS[1]: [3, 4, 12],
        ACT_LABELS[2]: [7, 8, 15],
        ACT_LABELS[3]: [9, 16],
        ACT_LABELS[4]: [6, 14],
        ACT_LABELS[5]: [5, 13],
    }
```

```
return ACT_LABELS, TRIAL_CODES
```

```
def make_trial_to_act():
    """
    Returns dictionary to get activity from trial code for
    MotionSense dataset
    """
    trial_to_act = {1: 0, 2: 0, 11: 0, 3: 1, 4: 1, 12: 1, 7: 2, 8:
2, 15: 2, 9: 3, 16: 3, 6: 4, 14: 4, 5: 5, 13: 5}
    return trial_to_act
```

```
class Encoder(Model):
    def __init__(self, latent_dim):
        super(Encoder, self).__init__()
        self.lstm1 = layers.LSTM(128, return_sequences=True)
        self.lstm2 = layers.LSTM(64)
        self.dense_mean = layers.Dense(latent_dim)
        self.dense_log_var = layers.Dense(latent_dim)

    def call(self, x):
        x = self.lstm1(x)
        x = self.lstm2(x)
        z_mean = self.dense_mean(x)
        z_log_var = self.dense_log_var(x)
        return z_mean, z_log_var
```

Sampling Layer

```
class Sampling(layers.Layer):
    def call(self, inputs):
        z_mean, z_log_var = inputs
        epsilon = tf.random.normal(shape=tf.shape(z_mean))
```

```

z = z_mean + tf.exp(0.5 * z_log_var) * epsilon
return z

```

Define the Decoder

```
class Decoder(Model):
```

```

    def __init__(self, latent_dim, seq_len, num_features):
        super(Decoder, self).__init__()
        self.repeat_vector = layers.RepeatVector(seq_len)
        self.lstm1 = layers.LSTM(64, return_sequences=True)
        self.lstm2 = layers.LSTM(64, return_sequences=True)
        self.output_layer

```

=

```
layers.TimeDistributed(layers.Dense(num_features))
```

```

    def call(self, z):
        x = self.repeat_vector(z)
        x = self.lstm1(x)
        x = self.lstm2(x)
        x = self.output_layer(x)
        return x

```

Define the VAE with custom training step

```
class VAE(Model):
```

```

    def __init__(self, encoder, decoder, beta=1.0):
        super(VAE, self).__init__()
        self.encoder = encoder
        self.decoder = decoder
        self.sampling = Sampling()
        self.beta = beta # Initialize beta for KL divergence

```

scaling

```

    def call(self, x):
        z_mean, z_log_var = self.encoder(x)

```

```

        z = self.sampling((z_mean, z_log_var))
        reconstructed = self.decoder(z)
        # Add KL divergence loss
        kl_loss = -0.5 * tf.reduce_mean(tf.reduce_sum(1 + z_log_var
- tf.square(z_mean) - tf.exp(z_log_var), axis=1))
        self.add_loss(self.beta * kl_loss)
        return reconstructed

    def train_step(self, data):
        x, y = data
        with tf.GradientTape() as tape:
            reconstructed = self(x, training=True)
            # Compute reconstruction loss
            reconstruction_loss =
tf.reduce_mean(tf.reduce_sum(tf.keras.losses.mse(y, reconstructed),
axis=1))

            # Total loss = Reconstruction loss + KL divergence loss
            total_loss = reconstruction_loss + sum(self.losses) #
self.losses includes KL divergence loss

            grads = tape.gradient(total_loss,
self.trainable_variables)
            self.optimizer.apply_gradients(zip(grads,
self.trainable_variables))
            # Update metrics
            for metric in self.metrics:
                metric.update_state(x, reconstructed)
            # Return a dict mapping metric names to current value
            return {"loss": total_loss}

# Implement KL annealing using a custom callback
class BetaAnnealingCallback(tf.keras.callbacks.Callback):

```

```

def __init__(self, total_epochs, beta_start=0.0,
beta_end=1.0):
    super(BetaAnnealingCallback, self).__init__()
    self.total_epochs = total_epochs
    self.beta_start = beta_start
    self.beta_end = beta_end

def on_epoch_begin(self, epoch, logs=None):
    # Linearly increase beta
    beta = self.beta_start + (self.beta_end - self.beta_start)
    * (epoch / self.total_epochs)
    self.model.beta = beta
    print(f"Epoch {epoch+1}, updated beta to {beta:.4f}")

def evaluate_models(models, data):
    results = []
    for model in models:
        preds = model.predict(data, verbose=0)
        res =
tf.math.reduce_mean(tf.math.reduce_mean(tf.math.square(data - preds),
axis=2), axis=1)
        results.append(res)

    results = tf.stack(results)

    return tf.math.argmin(results)

if __name__ == "__main__":
    # Define parameters
    latent_dim = 32 # Increased latent dimension
    seq_len = 100
    num_features = 6

```

```

beta = 0.01 # Starting value for beta
max_beta = 1.0 # Maximum value for beta
beta_increment = 0.005 # Increment for beta at each epoch

lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=1e-3,          decay_steps=10000,
decay_rate=0.96, staircase=True
)
optimizer =
tf.keras.optimizers.Adam(learning_rate=lr_schedule)
mse_loss_fn = tf.keras.losses.MeanSquaredError()

dataset = Dataset("./data/full_dataset.h5", time_window=2,
overlapping=0.75, split_ratio=0.8, K=6, seed=42)
dataset.build_dataset(5, dataset.single_split)

X_train, Y_train = dataset.make_dataset("train")
X_train = np.reshape(X_train, (-1, 100, 6))
X_test, Y_test = dataset.make_dataset("test")
X_test = np.reshape(X_test, (-1, 100, 6))

unique_activities = np.unique(Y_train)
models = []
histories = []

for activity in unique_activities:
    activity_idx = Y_train == activity
    train_dataset = X_train[activity_idx]

    encoder = Encoder(latent_dim)
    decoder = Decoder(latent_dim, seq_len, num_features)
    vae = VAE(encoder, decoder)

    vae.compile(

```

```

        optimizer=tf.keras.optimizers.Adam(learning_rate=1e-
3), metrics=[tf.keras.metrics.MeanSquaredError()])

        lr_schedule =
tf.keras.optimizers.schedules.ExponentialDecay(
            initial_learning_rate=1e-3, decay_steps=10000,
decay_rate=0.96, staircase=True
        )
        optimizer =
tf.keras.optimizers.Adam(learning_rate=lr_schedule)

        # Train the VAE using model.fit()
        epochs = 100
        batch_size = 32

        beta_callback = BetaAnnealingCallback(total_epochs=epochs,
beta_start=0.01, beta_end=1.0)

        history = vae.fit(
            train_dataset,
            train_dataset,
            epochs=epochs,
            batch_size=batch_size,
            callbacks=[beta_callback],
            validation_split=0.1,
        )

        models.append(vae)
        histories.append(history)

    res = evaluate_models(models, X_test)

    print(classification_report(Y_test, res, target_names=labels))

```

```
cm = confusion_matrix(y_true=Y_test, y_pred=res)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=labels)
disp.plot()
```