

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.048:520.88

«До захисту допущено»
в.о. завідувача кафедри ММСА
_____ О.Л. Тимощук
« ____ » _____ 2021 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 122 Комп'ютерні науки
на тему: «Застосування методів машинного навчання для визначення та
класифікації астрономічних подій»

Виконав:

студент II курсу, групи КА-93 мн
Стець Сергій Юрійович _____

Керівник:

доцент кафедри математичного
моделювання економічних систем
КПІ ім. Ігоря Сікорського, к.ф.-м.н., доц.
Пишнограєв І. О. _____

Рецензент:

доцент кафедри математичного
моделювання економічних систем
КПІ ім. Ігоря Сікорського, к.ф.-м.н., доц.
Лазаренко І. С. _____

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

Київ
2021

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)
Спеціальність — 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
в.о. завідувача кафедри
О. Л. Тимошук
«__» _____ 2021 р.

ЗАВДАННЯ
на магістерську дисертацію студенту Стецю Сергію Юрійовичу

1. Тема дисертації: «Застосування методів машинного навчання для визначення та класифікації астрономічних подій», науковий керівник дисертації Пишнограєв Іван Олександрович, канд. фіз.-мат. наук, доц., затверджено наказом по університету від 17 березня 2021 р № 851-с

2. Термін подання студентом дисертації: 14 травня 2021 р.

3. Об'єкт дослідження: класифікація астрономічних подій

4. Предмет дослідження: алгоритми машинного навчання для задачі класифікації та розпізнання гравітаційних хвиль

5. Перелік завдань, які потрібно розробити:

- Дослідити гравітаційні хвилі та загальну теорію відносності Ейнштейна;
- Провести огляд та обрати кращі методи машинного навчання для вирішення задачі;
- Побудова моделі;
- Проаналізувати ринкових можливостей запуску стартап-проекту;
- Зробити висновки за результатами наукового дослідження.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

- Приклади даних;

- Схеми побудови моделей;
- Таблиці та графіки результатів.

7. Орієнтовний перелік публікацій: опублікувати 1 наукову статтю у міжнародній науково-практичній конференції “ACTUAL TRENDS OF MODERN SCIENTIFIC RESEARCH”. Мюнхен, Німеччина

8. Дата видачі завдання: 10 березня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1	Оформлення концептуального вступу	10.03.2021—15.03.2021
2	Огляд технічної літератури за темою	16.03.2021—20.03.2021
3	Збір статистичних даних	21.03.2021—01.04.2021
4	Вибір методів аналізу	02.04.2021—07.04.2021
5	Розробка програмного комплексу	08.04.2021—25.04.2021
6	Аналіз отриманих результатів	26.05.2021—30.04.2021
7	Оформлення звіту, формулювання висновків	01.05.2021—05.05.2021

Студент

Сергій СТЕЦЬ

Науковий керівник дисертації

Іван ПИШНОГРАЄВ

РЕФЕРАТ

Магістерська дисертація: 86 с., 26 рис., 15 табл., 27 бібл. найм.

Об'єктом дослідження є зображення гравітаційних хвиль з обсерваторії LIGO.

Предметом дослідження є методи класифікації та розпізнавання зображень методами машинного навчання.

Мета роботи – проаналізувати зображення гравітаційних хвиль з обсерваторії LIGO, дослідження вже створених підходів для класифікації і розпізнавання та розробка програмного забезпечення.

Виконано аналіз методів автоматичної класифікації та розпізнавання. Було розроблено модель нейронної мережі для класифікації та розпізнавання гравітаційних хвиль.

В результаті роботи було розроблено програмний продукт, який дозволяє класифікувати та розпізнавати гравітаційні хвилі.

**МАШИННЕ НАВЧАННЯ, ГРАВІТАЦІЙНІ ХВИЛІ, РОЗПІЗНАННЯ
ЗОБРАЖЕННЯ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА**

ABSTRACT

Master's dissertation: 86 pp., 26 figs., 15 tables., 27 bibl. hire.

The object of the study is the image of gravitational waves from the LIGO observatory.

The subject of research is the methods of classification and recognition of images by machine learning methods.

The purpose of the work is to analyze the images of gravitational waves from the LIGO observatory, to study the already created approaches for classification and recognition, and to develop software.

The analysis of automatic classification and recognition methods is performed. A neural network model was developed for the classification and recognition of gravitational waves.

As a result, a software product was developed that allows to classify and recognize gravitational waves.

MACHINE LEARNING, GRAVITATIONAL WAVES, IMAGE RECOGNITION, CONVOLUTIONAL NEURAL NETWORK.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Огляд літератури	10
1.2 Гравітаційні хвилі	13
1.3 Перші спроба реєстрації гравітаційні хвилі	21
1.4 LIGO гравітаційно-хвильова обсерваторія	22
1.5 Висновки до розділу	27
РОЗДІЛ 2. МЕТОДИ ДЛЯ ВИЗНАЧЕННЯ ТА КЛАСИФІКАЦІЇ АСТРОНОМІЧНИХ ПОДІЙ	28
2.1. Огляд методів машинного навчання	28
2.1.1 Навчання із учителем	29
2.1.2 Навчання без вчителя	30
2.1.3 Часткове навчання	30
2.1.4 Навчання із підкріпленням	31
2.2 Глибоке навчання	32
2.3 Нейронні мережі для задачі розпізнавання	33
2.3.1 Архітектура нейронних мереж	35
2.4 Згорткові нейронні мережі	39
2.5 Висновки до розділу	44
РОЗДІЛ 3. ОПИС ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ	45
3.1 Опис вхідних даних	45
3.2 Попередня обробка даних	47
3.3 Опис частин програми	51
3.4 Порівняння результатів	52
3.5 Висновки до розділу	61
РОЗДІЛ 4 РОЗРОБКА СТАРТАП ПРОЕКТУ	62
4.1 Опис ідеї проекту	62
4.2 Технологічний аудит проекту	63
4.2.1 Технологічна здійсненність ідеї проекту	65
4.2.2 Попередня характеристика потенційного ринку	65
4.2.3 Характеристика потенційних клієнтів	66
4.2.4 Фактори загроз	67

4.2.5 Обґрунтування факторів конкурентоспроможності	70
4.2.6 SWOT- аналіз стартап-проекту	70
4.3 Розроблення ринкової стратегії проекту.....	71
4.3.1 Вибір цільових груп потенційних споживачів	71
4.3.2 Визначення базової стратегії конкурентної поведінки.....	73
4.3.3 Визначення стратегії позиціонування	73
4.4 Розроблення маркетингової програми	74
4.4.1 Ключові переваги	74
4.4.2 Формування систем збуту	75
4.5 Висновки до розділу	76
ВИСНОВКИ.....	77
ПЕРЕЛІК ПОСИЛАНЬ	78
ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ	81

ВСТУП

Актуальність проблеми. Гравітаційні хвилі - це «брижі» у просторі-часі, спричинені деякими найбільш бурхливими та енергійними процесами у Всесвіті. Альберт Ейнштейн передбачив існування гравітаційних хвиль у 1916 р. У своїй загальній теорії відносності. Математика Ейнштейна показала, що масивні прискорювані об'єкти (наприклад, нейтронні зірки або чорні діри, що обертаються навколо один одного) порушують простір-час таким чином, що хвилі хвилеподібного простору-часу поширяться у всіх напрямках від джерела. Ці космічні брижі рухались би зі швидкістю світла, несучи із собою інформацію про своє походження, а також підказки про природу самої сили тяжіння. Найсильніші гравітаційні хвилі виробляються внаслідок катаклізмів, таких як зіткнення чорних дір, наднових (масивні зірки, що вибухають в кінці свого життя), і зіткнення нейтронних зірок. Інші хвилі, як передбачається, будуть викликані обертанням нейтронних зірок, які не є ідеальними сферами, і, можливо, навіть залишками гравітаційного випромінювання, створеного Великим вибухом. Оцінка та класифікація гравітаційних хвиль є важливим завданням для астрономії та фізики в сучасних умовах.

Важливою частиною науки про дані є алгоритми та методи машинного навчання. Вони є сучасним інструментом для роботи не лише в астрофізиці, а й в багатьох інших галузях. Основна задача, що стоїть перед астрофізиками є задача класифікації гравітаційних хвиль з обсерваторії LIGO. Завдяки сучасному прогресу в області програмного та апаратного забезпечення стає можливо використовувати алгоритми машинного навчання та їх моделі для допомоги та збільшення швидкості роботи, а також покращення результату загалом.

Таким чином, метою дисертації є створення ефективної моделі для розв'язання задачі класифікації гравітаційних хвиль з обсерваторії LIGO.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд літератури

Попри всю свою величезну пустоту, Всесвіт «гуде» від активності у вигляді гравітаційних хвиль. Вироблені екстремальними астрофізичними явищами, ці «відбиття» або хвилі струшують тканину простору та часу. Виявлення гравітаційних хвиль (ГХ) спричинило справжню бурю у всіх розділах фізики. Загальна теорія відносності Альберта Ейнштейна каже, що космос має поводити себе як деяка субстанція, яку можна розтягнути або навіть стиснути. Знаходження вченими гравітаційних хвиль напряду підтверджує цю теорію та відкриває нову еру в астрономії та астрофізиці.

У 2017 р. науковці змогли побачити гравітаційні хвилі завдяки події, яку також вдалося спостерігати за допомогою наземних телескопів. Близько 150 мільйонів років тому, дві нейтронні зірки наблизилися одна до одної по спіралі і злилися в один об'єкт. Таке явище називають «кілонова» (астрономічне явище злиття двох нейтронних зір або нейтронної зорі та чорної діри.) [1]. 17 серпня 2017 р. ці зірки були знайдені та зареєстровані на Землі за допомогою лазерної інтерферометричної гравітаційно-хвильової обсерваторії (Laser Interferometer Gravitational-Wave Observatory) LIGO та обсерваторії Virgo. Дані які отримали науковці змінили погляд людини на Всесвіт. Огляд літератури проведено в таблиці 1.1:

Таблиця 1.1 – Огляд літератури про гравітаційні хвилі

Автори, рік	Назва статті	Короткий опис
Іван Крячко (2017)	“Гравітаційні хвилі відкрили в 2017 році нову еру в астрономії” [1]	Виявлення гравітаційних хвиль в 2017 році. Підтвердження теорії Ейнштейна.

Продовження таблиці 1.1

Phys. Rev. Lett (2016)	“Observation of Gravitational Waves from a Binary Black Hole Merger” [2]	Спостереження. Детектори. Минулі дослідження та пошуки.
Леонід Чорногор, Світогляд, (2016)	“Сенсація XXI сторіччя: Гравітаційні хвилі знайдено!” [3]	Типи взаємодії гравітаційних хвиль. Важливі шляхи до відкриття. Як можна виявити гравітаційні хвилі. Що дадуть людству гравітаційні хвилі.
Davide Castelvechi& Alexandra Witze (2016)	“Einstein's gravitational waves found at last” [4]	Через сто років після того, як Альберт Ейнштейн передбачив існування гравітаційних хвиль, вчені нарешті помітили ці невловимі брижі у просторі-часі.
Kostas D. Kokkotas , Department of Physics, Aristotle University of Thessaloniki Thessaloniki, Greece, (2002)	“Gravitational Wave Physics” [5]	Лінеаризована теорія. Властивості гравітаційних хвиль. Потік енергії, який несуть гравітаційні хвилі. Випромінювання від гравітаційного колапсу. Випромінювання від двійкових систем. Випромінювання від обертаючих нейтронних зірок
Jorge L. Cervantes-Cota, Salvador Galindo-Uribarri and George F. Smoot, (2018)	“A Brief History of Gravitational Waves” [6]	Відкриття та дослідження гравітаційних хвиль
Alan Weinstein, Caltech (2019)	“Gravitational Waves and LIGO” [7]	Що таке гравітаційні хвилі? Звідки вони беруться? Як ми можемо їх виявити? LIGO та інші її проекти

Кінець таблиці 1.1

Ana Caramete, Andrei-Ieronim Constantinescu, Laurent, iu-Ioan Caramete, Traian Popescu, Razvan- Alexandru Balasov, Daniel Felea, Mircea-Victor Rusu, Petruta Stefanescu Ovidiu-Mircea Tintareanu (2020)	“Characterization of Gravitational Waves Signals Using Neural Networks” [8]	Ідентифікація та класифікація сигналів гравітаційних хвиль з використанням нейронних мереж
Timothy D. Gebhard, Niki Kilbertus, Ian Harry, and Bernhard Schölkopf	“Convolutional neural networks: A magic bullet for gravitational- wave detection?” [9]	Існуючі підходи на базі CNN. Приклад використання для глибокого навчання.
Kyungmin Kim, Ian W Harry, Kari A Hodge, Young-Min Kim, Chang-Hwan Lee, Hyun Kyu Lee, John J Oh, Sang Hoon Oh, Edwin J Son (2013)	“Application of artificial neural network to search for gravitational-wave signals associated with short gamma-ray bursts” [10]	Застосування штучних нейронних мереж для пошуку CBC-GRB
M. Carrillo, J. A González, M. Gracia-Linares, F. S. Guzmán (2015)	“Time series analysis of Gravitational Wave signals using neural networks” [11]	Алгоритм, заснований на нейронних мережах, який передбачає відношення маси двійкового зіткнення чорних дір із заданих сигналів Гравітаційної Хвилі (ГВ).
Yuting LIU, Kentaro SOMIYA (2019)	“Detection and Localization of Gravitational Waves Using Convolutional Neural Networks” [12]	Лінійна класифікація з використанням одного перцептрону. Архітектура та робочий процес згорткової нейронної мережі.

1.2 Гравітаційні хвилі

Раніше гравітаційні хвилі спостерігалася лише побічно, через їх вплив на періоди пульсарів у подвійних системах. Форма сигналу, яка була спостережена обсерваторіями LIGO та Virgo, збігається з передбаченою загальною теорією відносності Альберта Ейнштейна для гравітаційної хвилі, що походить зі спірале-подібного руху, злиття і завершального об'єднання двох чорних дір в одну. Сигнал отримав від вчених назву GW150914 («Gravitational Wave 2015/09/14» — «Гравітаційна хвиля 2015/09/14») [2]. На той момент це було першим реальним спостереженням процесу злиття двох чорних дір. Ця подія доводить, що існують системи подвійних чорних дір зіркової маси і що їх злиття вже траплялися в минулому.

Спостереження було одразу освітлено по всьому світу. Понад п'ятдесяти років вчені з усіх куточків Землі намагались довести існування таких хвиль. Навіть сам Альберт Ейнштейн сумнівався про можливість їх знаходження. Хвилі, що з'явилися внаслідок катаклізму злиття GW150914, досягли Землі як брижі в просторі-часі, які змінили довжину чотирікілометрового рукава LIGO на одну десятитисячну частку ширини протона, що є пропорційною еквівалентно зміні відстані до найближчої зірки за межами сонячної системи до ширини волоска [13]. Енергія під час кінцевих секунд такої масштабної події є еквівалентною приблизно трьом масам Сонця. Вона перетворилася на гравітаційні хвилі менш ніж за половину секунди і випромінювалася з піковою швидкістю близько $3,6 \times 10^{49}$ Вт (Рис. 1.1). Такий рівень перевищує об'єднану потужність усього світла, випромінюваного всіма зірками в спостережуваній частині космосу.

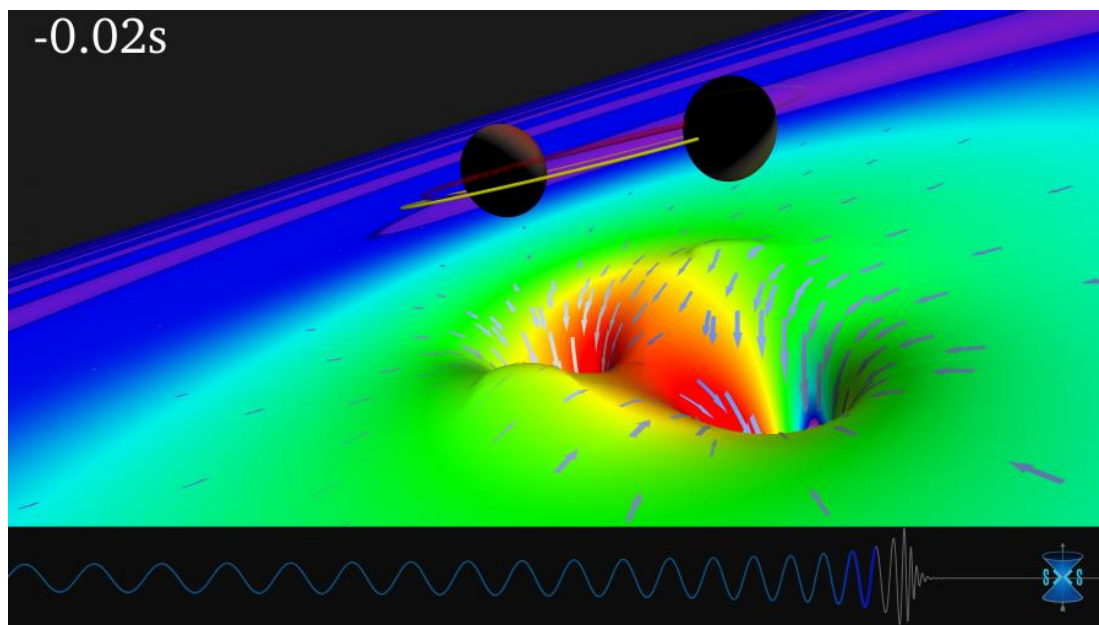


Рисунок 1.1 – Секунди до зіткнення чорних дір

Ця подія була проголошена як підтвердження останнього неперевіреного передбачення загальної теорії відносності Ейнштейна, і як перевірка цих гіпотетичних очікувань щодо спотворення простору-часу в контексті великомасштабних космічних подій (відомі як передбачення або тести суцільної теорії відносності), а також таким, що відкриває нову еру гравітаційно-хвильової астрономії. Це дає здатність досліджувати катастрофічні астрофізичні події, неспостережувані досі, а також безпосередньо спостерігати дуже ранню історію всесвіту перед тим, як спостереження на основі світла, радіо та інших електромагнітних хвиль стали б можливими.

На сьогодні звісно чотири типи взаємодій: сильна, електромагнітна, слабка і гравітаційна. Головні параметри цих взаємодій наведено в Табл. 1.2. Відомо, що потужна взаємодія утримує нуклони (протони і нейтрони) в ядрі атома, а також кварки (складові нуклонів) усередині нуклонів. Сильна взаємодія близькодійюча, вона істотна лише в межах ядра, тобто на відстанях близько 10^{-15} м. Електромагнітна взаємодія відповідальна за утворення атомів і молекул. Електромагнітні сили належать до сил далекої дії. Слабка взаємодія своєрідна для лептонів (класу елементарних частинок, яким не властива

сильна взаємодія, — електронів, мюонів, нейтрино та ін.) і баріонів (класу важких елементарних частинок — протонів, нейтронів, гіперонів та ін.). Приклад взаємодії протонів наведено в таблиці 1.2.

На відмінність від інших взаємодій, слабка взаємодія відповідає не за «зв'язок» речовини, а за її руйнування (наприклад, радіоактивний розпад). Зокрема, через слабку взаємодію в надрах Сонця та інших зірок відбуваються ядерні реакції з виділенням величезної кількості енергії. Гравітаційна взаємодія є дальньою взаємодією. Це стає конститутивним для великих мас у планетарному та космічному масштабі. Гравітаційні сили надзвичайно слабкі. Зокрема, сила електричного відштовхування двох електронів у $4,2 \cdot 10^{42}$ рази перевищує силу їх гравітаційного тяжіння. Фізики давно мріяли створити єдину теорію поля, яка поєднує всі чотири типи взаємодії.

Спочатку вдалося побудувати теорію електро-слабкої взаємодії, потім теорію Великого об'єднання, яка поєднує електричні, слабкі та сильні взаємодії. У теорії «супертеорії» повинні бути присутніми всі типи взаємодій. Проте гравітаційна взаємодія вперто "не погоджується бути залученою" в єдину теорію поля. Вперше гравітаційні сили та взаємодія були відкриті за часів Кеплера та Ньютона. Досі не зовсім зрозуміло, чи існує квант гравітаційного поля. Гравітаційна частинка має абсолютно такий самий заряд. Спостерігається тільки притягання саме таких частинок.

На відміну від електричного диполя, гравітаційний диполь неможливий. Однак слід зазначити, що близько 20 років тому було відкрито прискорене розширення Всесвіту, що пов'язано з темною енергією [3].

Таблиця 1.2 – Взаємодія протонів (із енергією 1 ГэВ $\sim 10^{-10}$ Дж) [3]

Взаємодія	Відносна величина	Дальність взаємодії, м	Тривалість взаємодії, с	Квант поля
сильна	1	10^{-15}	$10^{-23} \text{ — } 10^{-22}$	глюон
електромагнітна	1/137	будь-яка	$10^{-20} \text{ — } 10^{-18}$	фотон
слабка	10^{-5}	$2 \cdot 10^{-18}$	$10^{-10} \text{ — } 10^{-8}$	проміжний бозон
гравітаційна	10^{-38}	будь-яка	$> 10^{18}$	гравітон

Всі типи взаємодії мають скінченну швидкість її поширення. Не є винятком і гравітаційна взаємодія. Вільне гравітаційне поле, не зв'язане з тілом, становить собою гравітаційну хвилю, яка поширюється зі швидкістю світла (Рис. 1.2).

Гравітаційні хвилі породжуються тілами, які рухаються зі змінним прискоренням.

При досить слабких збуреннях ГБ є поперечним, тобто коливання відбуваються в області, перпендикулярній напрямку поширення. ГБ містять дві некеровані компоненти, розташовані під кутом 45° один до одного, які визначають два стани поляризації хвилі. ГБ несуть енергію та імпульс. Діючи на тіла, ці хвилі викликають деформацію тіл. На цьому атрибуті базуються принципи виявлення (виявлення) гравітаційних хвиль.

Спроби знайти гравітаційні хвилі не мали успіху майже 50 років. Це пов'язано з тим, що самі хвилі на краю слабкі і їх важко зафіксувати.

Для лабораторних джерел ГБ їх потужність не перевищує 10-20 Вт. Для порівняння зазначимо, що ця інтенсивність у тисячу разів менша за потужність світла, на яке реагує людське око. Для збільшення потужності ГХ у

лабораторних умовах необхідно збільшувати масу випромінювальних тіл і швидкість зміни прискорення (ривок).

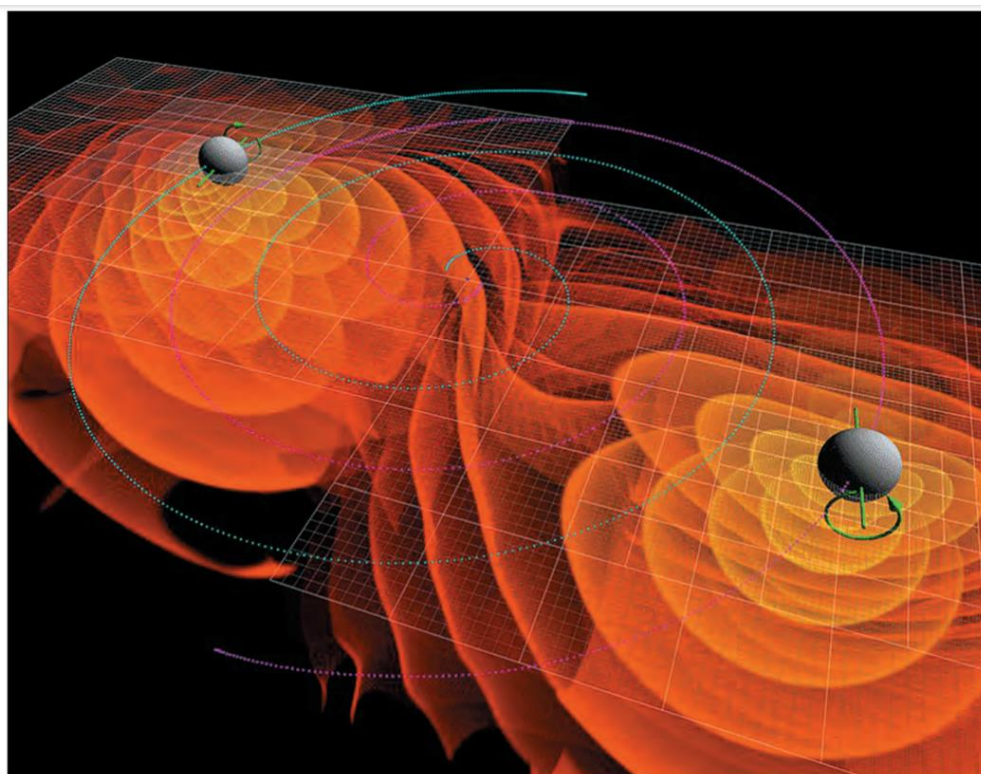


Рисунок 1.2 – Дві моделі злиття бінарних чорних дір, які випромінюють гравітаційні хвилі

Потужність випромінювання ГХ для двох взаємодіючих тіл збільшується пропорційно п'ятій інтенсивності тіл, а запас енергії співвідносний масі в першій потужності. Це означає, що при високих масах ефективність ГХ-випромінювання різко зростає. Потужність, яку випромінюють гравітаційні хвилі для повсякденних предметів, абсолютно незначна (Рис. 1.3). Навіть загальна потужність, випромінювана в гравітаційних хвилях Юпітером під час обертання навколо Сонця, становить лише кілька кіловат. Насправді існує лише кілька бінарних систем, де ніколи не було свідчень про випромінювання гравітаційних хвиль. Найвідоміший приклад - бінарний пульсар PSR B1913 + 16, який за допомогою методів хронометражу пульсарів показав, що його орбіта зменшується на 3 міліметри за орбітальний період через випромінювання гравітаційних хвиль. За це досягнення Халс і Тейлор були нагороджені в 1993 році Нобелівською премією з фізики.

Великі маси властиві космічним об'єктам. Прикладами космічних джерел ГХ є систематичність зірка-планета, двійкові зірки, вибухи наднової, зіткнення нейтронних зірок, злиття чорних дір та асиметричний колапс (швидке стиснення) зірок. Саме з таких джерел гравітаційні хвилі шукали протягом останніх 40-45 років.

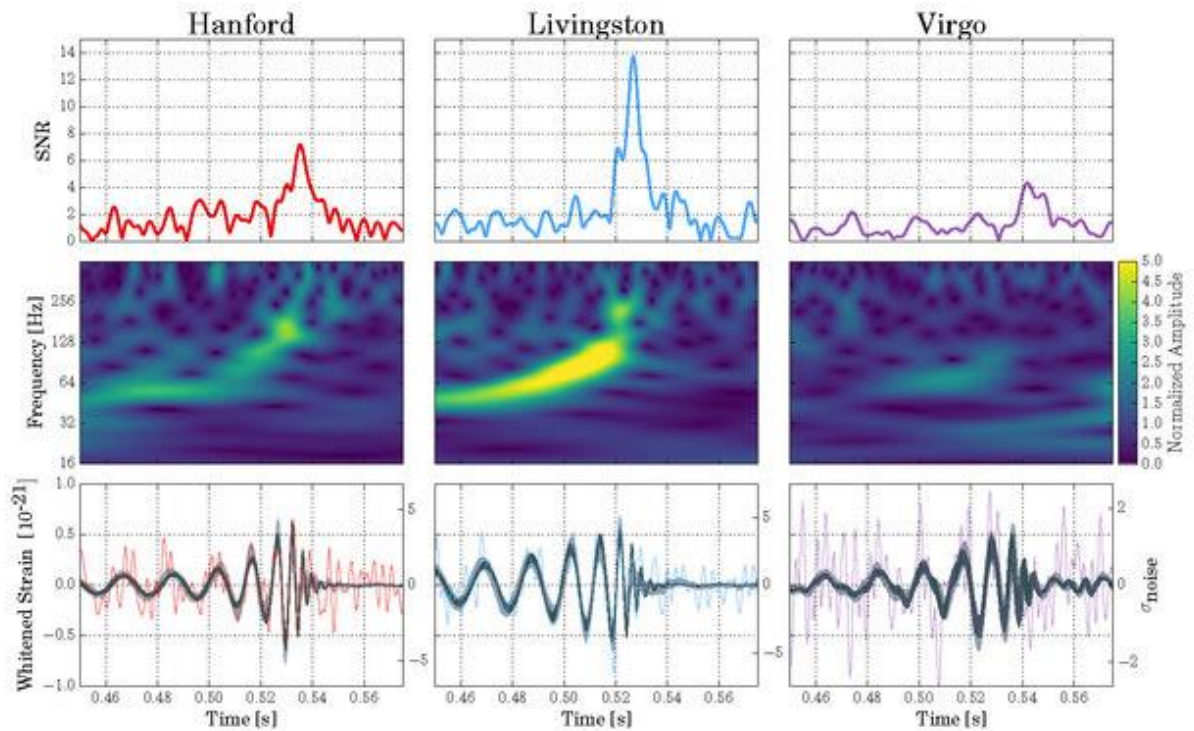


Рисунок 1.3 – «Підписи» гравітаційних хвиль, виявлені кожною обсерваторією

Результати розрахунків. Обертальний хід планет навколо Сонця є джерелом ГХ. Для найбільшої планети Сонячної системи — Юпітера — потужність випромінювання становить 5.2 кВт. При цьому обидва тіла витрачають свою енергію і наближаються у радіальному напрямі зі швидкістю $2.5 \cdot 10^{-20}$ м/с. Тобто, через $2.5 \cdot 10^{23}$ років Юпітер, витративши частину своєї енергії, міг би впасти на Сонце. Проте цього не трапитиметься, бо Сонцю залишилось існувати близько 5 млрд років [13].

Розрахунки показують, що бінарна зоря з масою кожної зорі 1000 мас Сонця, які розміщені на віддалі 100 млн км, випромінює гравітаційні хвилі потужністю $3 \cdot 10^{29}$ Вт. Стрімкість їх зближення становить 2.5 мм/с. Через

мільйон років ці зорі мають злитись. При цьому в енергію ГХ перейде лише близько $3 \cdot 10^{-8}$ енергії зір.

Гравітаційні хвилі забезпечують новий захоплюючий спосіб вивчення Всесвіту. Вони були вперше виявлені в 2015 році, коли гравітаційно-хвильова обсерваторія (LIGO) виявила гравітаційно-хвильовий сигнал (GW150914) від двох чорних дір, що стикаються (Рис. 1.4). З тих пір було виявлено 10 сигналів від злиття чорних дір, а також сигнал, що походить від коалесценції двох нейтронних зірок. Цей останній сигнал, який отримав назву GW170817, був вражаюче спостеріганий у всьому спектрі ЕМ, забезпечуючи перше спостереження астрономічного джерела як з ЕМ, так і з гравітаційними хвилями.

Зовсім інша картина ждється, коли дві чорні діри обертаються довкола спільного центру мас на віддалі одна від одної, приблизно однаковій радіусу тяжіння, і з орбітальною швидкістю, недалекою від швидкості світла. У цьому випадку для тіл із безліччю Сонця гравітаційний радіус є 3 км, кутова швидкість близько 16 кГц.

Гравітаційні хвилі під наглядом фізика-теоретика. ГС - збурення просторово-часових метрик, які відриваються від ключа, тобто гравітаційного тіла, і розбігаються хвилями, викликаючи "брижі простору-часу". Мова теоретиків фізики сильно відрізняється від науково-популярної. І все ж ми спробуємо пояснити, про що йде мова. Метрика - математичне поняття, що означає правило (формулу) для визначення відстані між будь-якими двома точками даного простору або частинками цього набору. Отже, в евклідовій геометрії найкоротша відстань між двома пунктами - це проміжок між прямими, що їх з'єднують. У 1900 р. Х.А. Лоренц припускав, що гравітаційна взаємодія "може йти зі швидкістю, не більшою за швидкість світла".

Термін "гравітаційна хвиля" був введений Пуанкаре ще в 1905 р. Він також затверджував, що існування ГК не заважає спостереженню рухомих тіл.

У 1916 р. А. Ейнштейн створив загальну теорію відносності і показав, що механічна система, що рухається зі змінним прискоренням, випромінює ГХ.

20-30-ті роки були роками вагань. Обговорювалось питання: чи справді можуть бути гравітаційні хвилі? Навіть сам автор сумнівався в їх існуванні.

Друга світова війна надовго перервала спір про реальність ГХ.

У 1978 р. Дж. Тейлор анонсує про знаходження випромінювання, викликаного бінарною системою пульсара.



Рисунок 1.4 – Одна зі секцій системи LIGO

Передусім, ніж опинитися в радіоприймач, радіохвиля проходить через антенний апарат. Тут відбувається трансформація енергії хвилі в енергію електричного сигналу. Для приймання ГХ також відповідна своєрідна антена і приймач.

Гравітаційною антеною може бути будь-яка пара або одне видовжене тіло. Гравітаційні приймачі слід таскати в просторі на віддалі 1-10 тис. Км. Ця система називається гравітаційним інтерферометром. Гравітаційний приймач може бути чутливим пристроєм, який реєструє надзвичайно малі умовні переміщення мас або сили, що їх спричиняють. Прийом ГХ заснований на тому, що ці хвилі взаємодіють з масивними тілами. Чим більша маса

гравітаційної антени, тим сильніший сигнал на вході та виході гравітаційного радіоприймача.

На даній відстані між двома тестовими тілами зміни цієї відстані дорівнюють добутку відстані та амплітуди хвилі. Наприклад, на відстані 5 км і амплітуді ГК 11-20 амплітуда варіацій становить $5 * 10^{-18}$ м, що в 2000 разів менше діаметра атомного ядра. І робиться це за допомогою багатокілометрової гравітаційної антени. Це основна складність при виявленні (ідентифікації) ГБ. Друге ускладнення полягає в тому, що амплітуда ГК обернено пропорційна відстані від джерела хвилі. Іншим ускладненням є невелика тривалість процесу, протягом якого генерується ГБ. Наприклад, при вибуху наднової тривалість сплеску становить 1 мс, а коли дві чорні діри зливаються. Нарешті, прийому ГБ заважають будь-які поштовхи гравітаційної антени. І це не просто далекі землетруси, від яких планета постійно трясеться, а й перешкоди наземного і навіть повітряного транспорту, а також промислових підрозділів і т. Д. Крім зовнішнього втручання важливу роль відіграють і внутрішні перешкоди - шум у гравітаційній антени. Щоб уповільнити їх дію, антену потрібно охолодити до температури в одиницях кельвіна або менше. І це коли розмір гравітаційної антени 1-4 м, а її маса 4-10 т.

1.3 Перші спроба реєстрації гравітаційні хвилі

Джозеф Вебер сконструював на той час перший резонансний детектор, що складався з єдиного двометрового алюмінієвого циліндру з чутливими п'єзо датчиками з сторін і віброізоляцією. При проходженні гравітаційної хвилі циліндр повинен резонувати в такт спотворень простору-часу, що і зобов'язані зареєструвати сенсори. Вчений збудував кілька таких детекторів, і в 1969 році, вилив, що зареєстрував «звучання гравітаційних хвиль» відразу в

декількох детекторах, рознесених один від одного на три кілометри. Заявлена їм амплітуда коливань позначилася неправдоподібно великою, в мільйон разів більше очікуваного значення. Протилежні експериментальні групи, зі схожими детекторами, не зуміли в наступному зловити ні одного подібного сигналу. Однак Вебер дав мотив всій цій області досліджень (Рис. 1.5.)

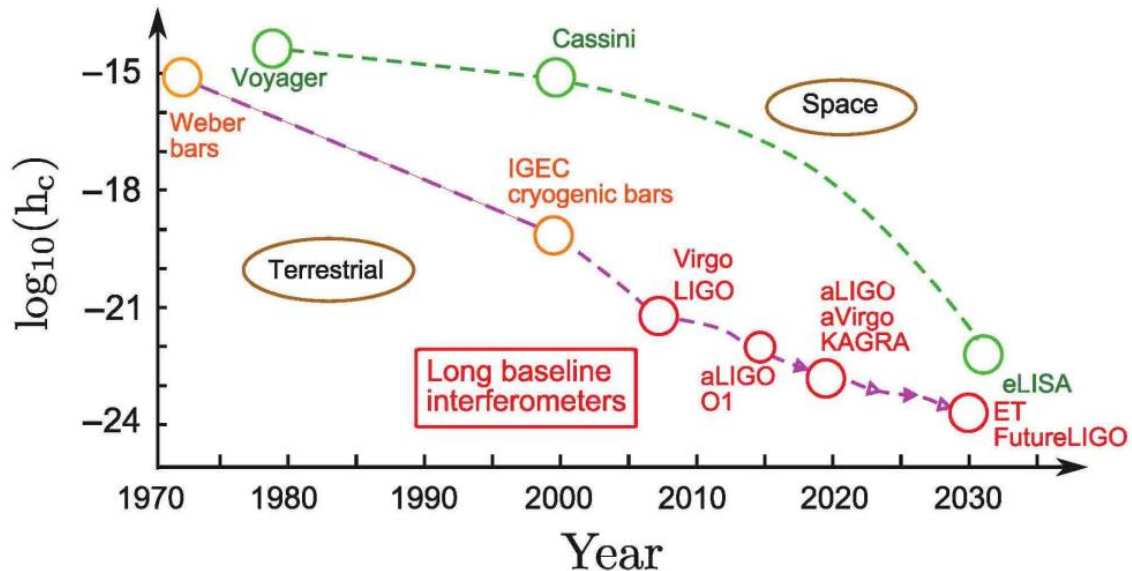


Рисунок 1.5 – Прогрес у розвитку гравітаційних детекторів [13]

За 45 років чутливість наземних ГВ детекторів покращилася на 7 порядків.

1.4 LIGO гравітаційно-хвильова обсерваторія

Сьогодні найдосконалішим лазерним інтерферометром є американська система LIGO (Лазерний інтерферометр Гравітаційний "Хвильовий спостерігач" - Лазерна інтерферометрична обсерваторія гравітаційних хвиль). Лівінгстон, штат Луїзіана: У Ханфордській обсерваторії є два інтерферометри, а в Лівінгстоні - один, і кожен інтерферометр складається з двох взаємно перпендикулярних плечей довжиною 4 км з контрольними масами на кінцях кожного плеча (рис. 1.6).

17 серпня космічний телескоп Фермі з використанням гамма-променів Фермі підняв імпульс високоенергетичного світла від потужного вибуху, про який негайно повідомили астрономи по всьому світу як короткий сплеск гамма-випромінювання. Вчені LIGO виявили гравітаційні хвилі, які отримали назву GW170817, від пари розбиваючих нейтронних зірок, прив'язаних до сплеску гамма-променів, спонукаючи астрономів шукати наслідки вибуху. Незабаром після цього сплеск було виявлено в рамках подальшого аналізу супутником ESA INTEGRAL. Пізніше місії Свіфт, Хаббл, Чандра та Спітцер НАСА зафіксували згасаюче сяйво розширюваного сміття вибуху.

Коли Свіфт незабаром після виявлення сплеску гамма-променів Фермі повернувся до галактики, він виявив яскраве та швидко згасаюче джерело ультрафіолету. Чандра чітко виявив рентгенівські промені через дев'ять днів після відкриття джерела. 22 серпня Хаббл розпочав знімати кілонову та фіксувати її ближній інфрачервоний спектр, який виявив рух та хімічний склад сміття, що розширюється. Спектр виглядав дуже схожим на те, як фізики-теоретики передбачали результат злиття двох нейтронних зірок. Це прив'язало цей об'єкт до джерела гравітаційних хвиль поза всяким розумним сумнівом.

Астрономи вважають, що видиме та інфрачервоне кілонове світло в основному генерується нагріванням внаслідок розпаду радіоактивних елементів, що утворюються в багатих нейтронами уламках. Падаючі нейтронні зірки можуть бути головним джерелом у Всесвіті багатьох важких елементів, включаючи платину та золото.

Лазерний інтерферометр використовується для вимірювання надзвичайно малих переміщень тестових мас антени. У цьому випадку розмір антени - це відстань між двома тестовими тілами. Він може досягати тисяч і більше кілометрів.

Декілька фізиків і астрофізиків засумнівалися в тому, що американській обсерваторії LIGO пощастило зафіксувати гравітаційні хвилі [7]. Дослідники

виявили так звану неординарну "синхронізацію шумів" в даних з двох детекторів цього наукового приладу.

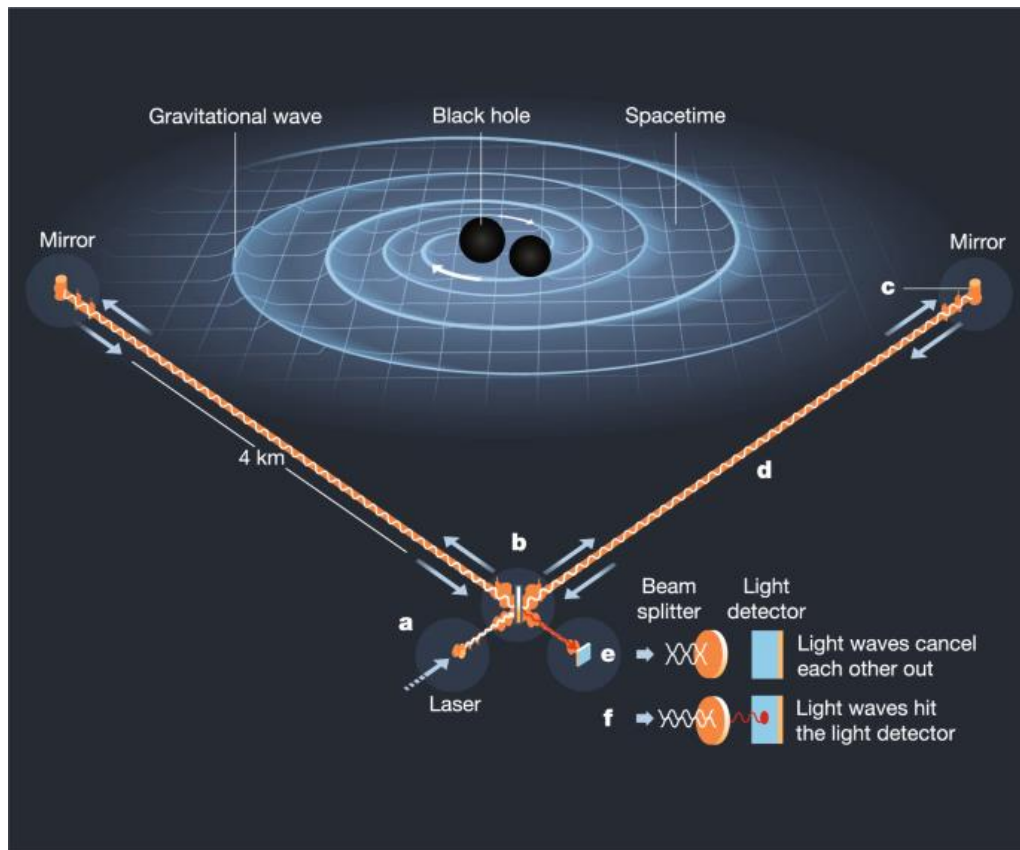


Рисунок 1.6 – Структура роботи обсерваторії LIGO [13]

В ході своїх досліджень вчені тренували не один сигнал, а шум, що генерується квантовими коливаннями в порожнечі та інших випадкових процесах, а також спеціальні калібрувальні системи, що використовуються для синхронізації "половин" обсерваторії LIGO.

Як бачать ті самі вчені, шуми заінтригували їх лише з тієї причини, що випадкова синхронізація шумів в обох детекторах протягом деякого часу може випадково генерувати сигнал, подібний до гравітаційних хвиль. Такий сигнал практично не буде розпізнаний з реального, але це якщо аналізувати лише сигнал, а не всі дані.

На аргументування дослідників, "синхронізація шуму" мала місце як при виявленні події GW150914 [1], так і в GW170104. Сигнал, ймовірно, надходив з Південної півкулі, приблизно в напрямку Магелланових Хмар. На їхню

думку, це нескінченно зменшує ймовірність того, що LIGO насправді зміг виявити злиття чорних дір.

Зразки з обсерваторії стверджують, що ці упущення присутні в розрахунках, проведених в рамках нового дослідження. На думку вчених, техніка, яку вони використовували для "очищення" самого отриманого сигналу, генерує такі кореляції або шуми. Як зазначалося, якщо приймати методологію датських фізиків для аналізу випадкових даних, отримується така ж сама картина, як і при аналізі результатів спостережень за допомогою LIGO, що викреслює розрахунки авторів статті (Рис. 1.7).

Більше того, як стверджують представники LIGO, сам сигнал можна побачити в даних обох детекторів беззбройним оком, не порівнюючи його з інформацією з другої "половини" обсерваторії і не очищуючи його (Рис. 1.8).

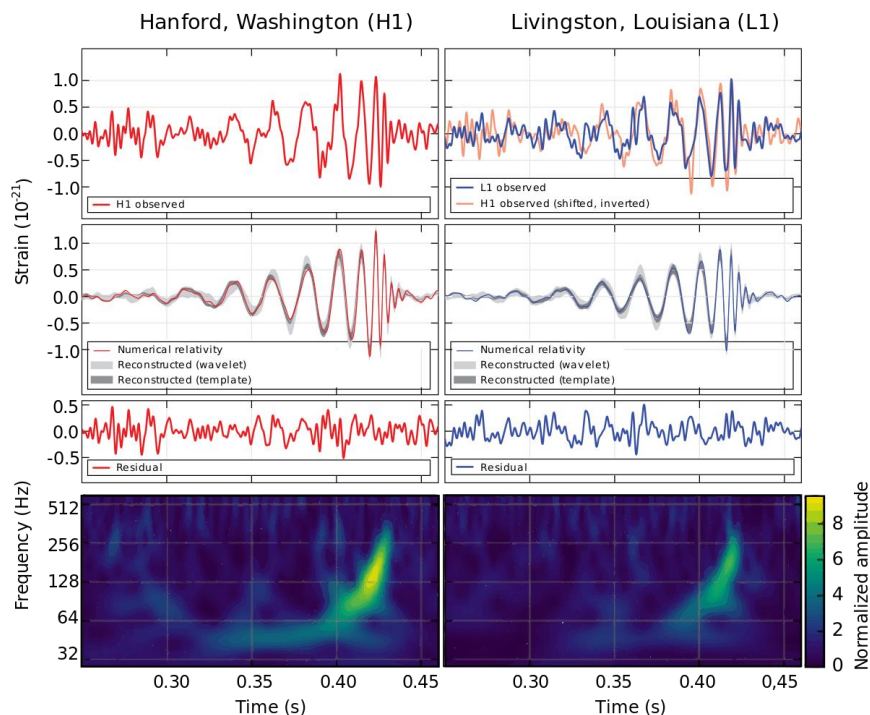
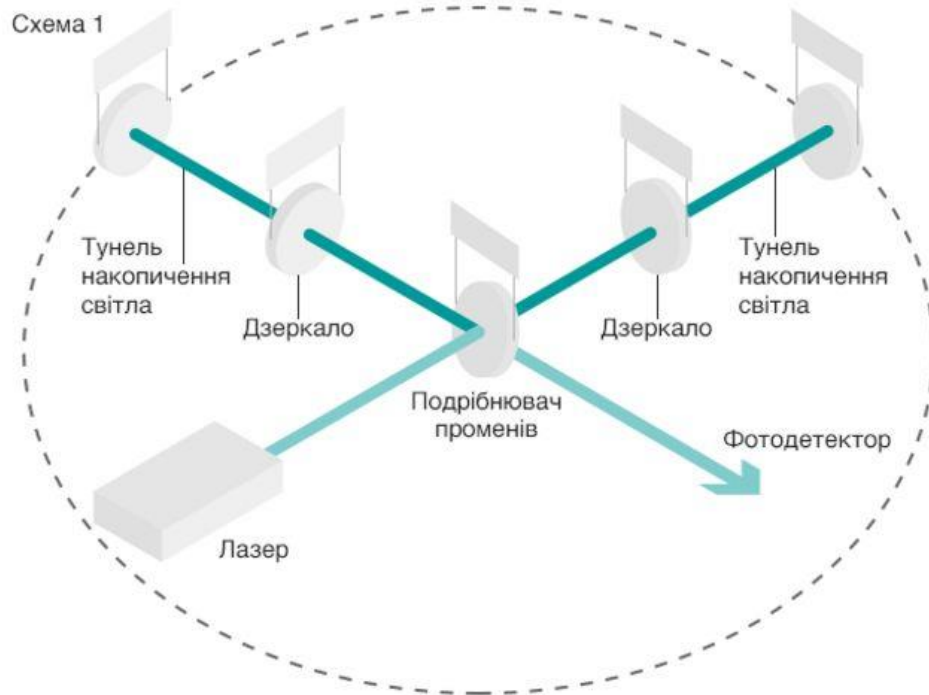
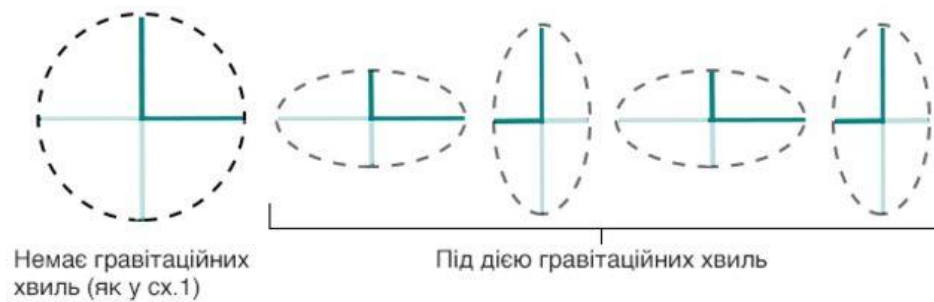


Рисунок 1.7 – Часова залежність сигналів гравітаційної хвилі GW150914, зареєстрованих в обсерваторіях Хенфорд (зліва) і Лівінгстон (справа) [2]

Інтерферометр: як працює детектор гравітаційних хвиль



Гравітаційні хвилі по чергово розтягують і стискають простір



Джерело: LIGO/NSF

BBC

Рисунок 1.8 – Схема роботи LIGO

LIGO зробила неймовірне відкриття, яке вчені досліджують по нинішній день. Немала обсерваторія завдала змогу дізнатися те, що неможливо було побачити та створила нову вітку в сучасній астрономії.

1.5 Висновки до розділу

Гравітаційні хвилі — новітній інструмент для вивчення Всесвіту і процесів у ньому. Астрономія справді стала всехвильовою. Століттями астрономія була оптичною. Все-таки хвилі видимого діапазону є лише нашим вузьким віконцем у Всесвіт. Плавна до оптичної астрономії приєднались радіоастрономія, інфрачервона, ультрафіолетова, рентгенівська і гамма-астрономія. Тепер прийшла черга і гравітаційно-хвильової астрономії. Усі вони великою мірою доповнюють одна одну, поглиблюючи наші знання про світ.

Ідея гравітаційних хвиль вже була передбачена в спеціальній теорії відносності 1905 року з її граничною граничною швидкістю передачі інформації. Явне визначення гравітаційних хвиль загальної теорії відносності було висунуто Ейнштейном в 1916 і 1918 рр. Він засвідчив, що прискорення мас породжує залежні від часу гравітаційні поля, які поширюються від своїх джерел зі швидкістю світла як бойові сторінки простору-часу.

Гравітаційно-хвильова астрономія допускає нам заглянути на відстань в 13 млрд світлових років, тобто підкрастися впритул до моменту народження нашого Всесвіту. Відповідно до теорії, Великий вибух, який породив Всесвіт, і його розширення мають супроводжуватися надпотужним випромінюванням ГХ. Так зване реліктове, тобто залишкове, яке існує й донині, випромінювання ГХ несе інформацію про найперші миті життя нашого Всесвіту.

РОЗДІЛ 2. МЕТОДИ ДЛЯ ВИЗНАЧЕННЯ ТА КЛАСИФІКАЦІЇ АСТРОНОМІЧНИХ ПОДІЙ

2.1. Огляд методів машинного навчання

Машинне навчання (Machine Learning) — достатньо новий та обширний підрозділ штучного інтелекту, який вивчає методи побудови алгоритмів, здатних навчатися. Машинне навчання перебуває на стику математичної статистики, методів оптимізації та інших, древніх математичних дисциплін. Підрозділ має і власну специфіку, пов'язану з проблемами обчислювальної ефективності та перенавчання. Багато методів індуктивного навчання розроблялися як альтернатива класичним статистичним підходам. Багато методів міцно пов'язані з діставанням інформації та інтелектуальним аналізом даних[14].

Найбільш теоретичні розділи машинного навчання об'єднані в окрему область, яка називається теорією обчислювального навчання.

Машинне навчання - це не просто математика. Це практична інженерна дисципліна. Тільки теорія, як правило, не відразу призводить до методів та алгоритмів, які можна застосувати на практиці. Для роботи необхідно винайти додаткові механізми, щоб компенсувати розбіжність між висунутими в теорії припущеннями та умовами реальних проблем. Майже жодне дослідження машинного навчання не обходиться без експериментів на моделі чи реальних даних, що підтверджує практичність методу.

2.1.1 Навчання із учителем

У навчання з учителем, або Supervised learning, ідея полягає існуванні деякої залежності між відповідями і об'єктами, але вона невідома. Відома тільки кінцева сукупність прецедентів, або пар «об'єкт, відповідь», яка названа навчальною вибіркою [15]. На основі цих даних моделі потрібно відновити залежність, а саме побудувати алгоритм, здатний для будь-якого об'єкта видати досить точну відповідь. Для вимірювання точності відповідей певним чином вводиться функціонал якості.

Під учителем розуміється або сама навчальна вибірка, або той, хто вказав на заданих об'єктах правильні відповіді. (Рис. 2.1)

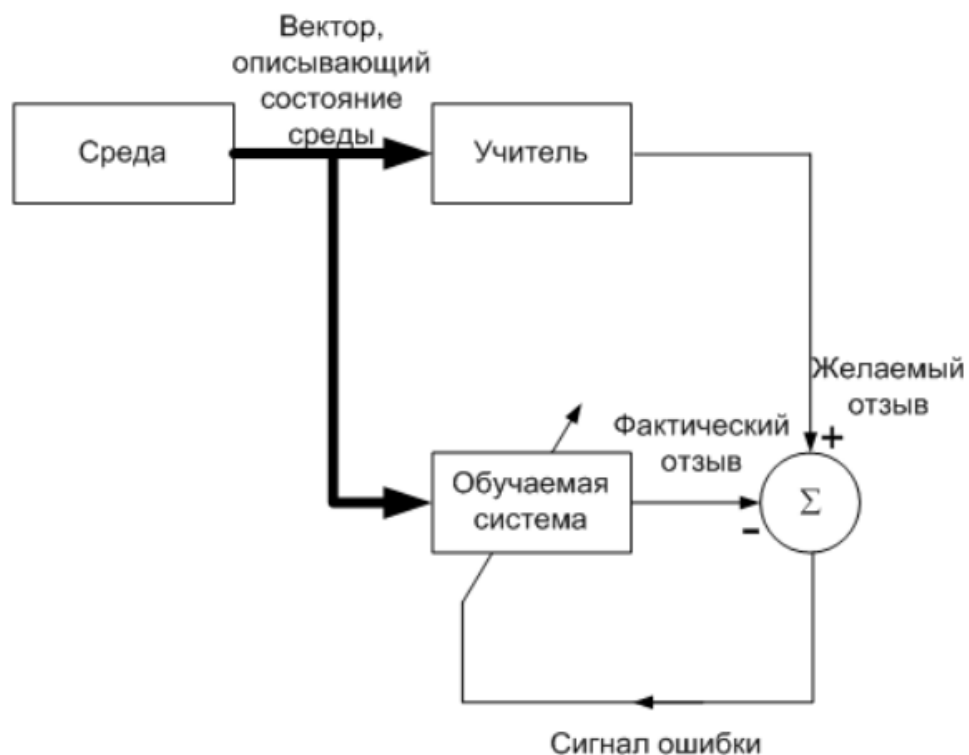


Рисунок 2.1 – Навчання з учителем

Функціонал якості, зазвичай, визначається як середня помилка відповідей, виданих алгоритмом, по всіх об'єктах вибірки.

2.1.2 Навчання без вчителя

Навчання без вчителя, або Unsupervised learning [15], вирішує в свою чергу задачі, у яких відомий тільки лише опис множини об'єктів. Потрібно знайти внутрішні взаємозв'язки, залежності які існують між об'єктами. Прикладами таких задач є кластеризація і візуалізація цих даних.

2.1.3 Часткове навчання

Часткове навчання, або навчання під наглядом [15], використовує як позначені, так і розділені дані в навчанні. Як правило, таке навчання використовує невелику кількість тегованих даних та значну кількість немаркованих даних. Часткове навчання, в свою чергу, є компромісом між навчанням без нагляду та навчанням під контролем. Було зазначено, що якщо ви використовуєте непомічені дані з невеликою кількістю тегованих даних, то це може забезпечити значне і високе підвищення якості навчання. Під самою якістю навчання ми маємо на увазі певний функціонал якості, як варіант, середньоквадратичну помилку.

Збір "позначених" даних для навчання дуже часто вимагає від кваліфікованої особи вручну або спеціальної класифікації навчальних об'єктів / даних [16]. Вартість самого процесу розмітки може бути настільки великою, що неможливо створити абсолютний набір даних для роботи, тоді як збір нерозподілених даних є відносно простим. У таких випадках важливість часткового тренування дуже важко переоцінити. Прикладом прикладної проблеми є самозаголовок великої кількості питань, за умови, що деякі з них уже присвоєні деяким заголовкам.

2.1.4 Навчання із підкріпленням

Навчання із підкріпленням, або reinforcement learning, вивчає, як агент повинен діяти в колі, щоб максимізувати деякий довготривалий прибуток. Алгоритми з частковим навчанням намагаються знайти стратегію. В економіці та теорії ігор навчання з підмогою розглядається в якості інтерпретації того, як може встановитися рівновага [17].

Загалом навчання із підкріпленням працює таким чином (Рис. 2.2):

1. До агенту приходить стан роботи.
2. Агент сам вибирає дію.
3. Середовище посилає тому ж агенту винагороду і наступний стан програми.
4. Агент опрацьовує цю нагороду і коректує свої дії.

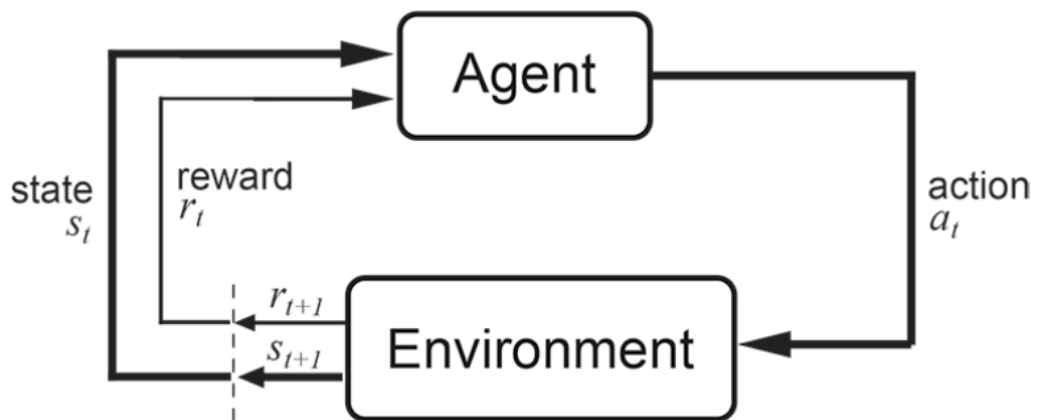


Рисунок 2.2 – Взаємодія агента та середовища

При вченні з підкріпленням не надаються вірні пари. Прийняття майже оптимальних рішень не обмежується [14]. Навчання з підкріпленням намагається знайти деякий компроміс між дослідженням невивчених областей і застосуванням наявних знань.

2.2 Глибоке навчання

Глибоке навчання - це досить нова галузь машинного навчання, яка використовує алгоритми, що імітують абстракції даних високого рівня шляхом багаторазової обробки шарів, що складаються з безлічі лінійних та нелінійних перетворень. У порядку гравітаційного агента використовується глибока модель із шарами гіпотонічної нейронної мережі (CNN), яка продемонструвала відмінні характеристики та вважається найсучаснішою у класифікації зображень. Іншою причиною використання глибокого навчання є його масштабованість [18]. Порівняно з традиційним машинним навчанням, такими методами, як векторна машинна підтримка (SVM), глибоке навчання може обробляти та використовувати великі обсяги даних.

Багато трактатів показали, що використання купу джерел інформації може поліпшити загальну ефективність систематизації гравітаційних хвиль. У цьому проекті використовуються багаторазові збої протягом тривалості, які також демонструються волонтерам на Zooniverse [13]. Ці тривалості поєднуються у квадратну форму, щоб ядра могли ковзати протягом будь-якої різної тривалості та вивчати схеми.

Висновок кожного ядра відомий як карта функцій. Карти об'єктів зазвичай вибирають із зразка з підтримкою макс. (Або середній). Тут використовується максспулінг для отримання - квадратна матриця ковзає по карті об'єктів і надає граничне значення серед елементів у ній. Для цього використовується шар активаційних функцій для визначення виходу даного нейрона. Модель Gravity Spy використовує популярну функцію активації, відому як випрямлена лінійна одиниця (ReLU), яка визначається як макс. Потім скріплений шар повністю наноситься [13]. Кожен вузол абсолютно пов'язаного шару з'єднаний з усіма вузлами попереднього шару.

Початковий шар - це шар «softmax» з 20 виходами. Softmax - це повністю зв'язаний шар з такою ж кількістю вузлів, що і кількість класів, і широко використовується як нижчий шар у багато класних завданнях з класифікації. Вихідна оцінка шару softmax, о використовується як розподіл ймовірностей, знайдений класифікатором зображень [19].

2.3 Нейронні мережі для задачі розпізнавання

Одним із найсучасніших способів вивести класифікатор машинного навчання є нейронні мережі.

Область нейронних мереж була натхненна метою моделювання реальних біологічних нейронних систем (Рис. 2.3), але з тих пір вона розійшлася і стала запитанням інженерії та звернення гарних результатів у завданнях машинного навчання (Рис. 2.4).

Ідея нейронних мереж взята з мозку людини. Дослідження Д. Хьюебеля та Т. Н. Візель у 1950–1960-х роках щодо мозку ссавців запропонували нову модель того, як ссавці візуально сприймають світ. Вони показали, що зорова кора кішок і мавп включає нейрони, які реагують на нейрони лише в їх безпосередньому оточенні.

Основною обчислювальною одиницею людського мозку є нейрон. Приблизно 85 000 000 000 нейронів можна знайти в нервовій системі людини. Ці нейрони зв'язуються приблизно з 10^{15} синапсами. Будь-який нейрон отримує вхідні дані від власних дендритів і виробляє висновки вздовж свого аксона. Аксон з часом розгалужується і через синапси з'єднується з дендритами інших нейронів. У обчислювальній моделі нейрона сигнали мультиплікативно взаємодіють з дендритами протилежного нейрона на основі синаптичної сили цього синапсу. Сама ідея полягає в тому, що синаптичні

сили можна вивчати і контролювати силу одного нейрона до іншого. У еталонній моделі дендрити несуть сигнал до тіла клітини, де всі вони отримують підсумки. Якщо невозмущенная кількість перевищує певний поріг, нейрон може спрацювати, посилаючи імпульс уздовж аксона [18].

У моделі розрахунку припускається, що тотожні терміни імпульсів не мають ваги, і що лише частота роботи передає інформацію. На основі цієї частотної інтерпретації моделюється тригерна частота нейрона, як функцію активації – f . Вона відображає частоту імпульсів уздовж аксона. Історично загальним вибором роботи активації є σ (сигмовидна функція), оскільки вона приймає істинне значення вхідного сигналу і розподіляє на відстані від 0 до 1.

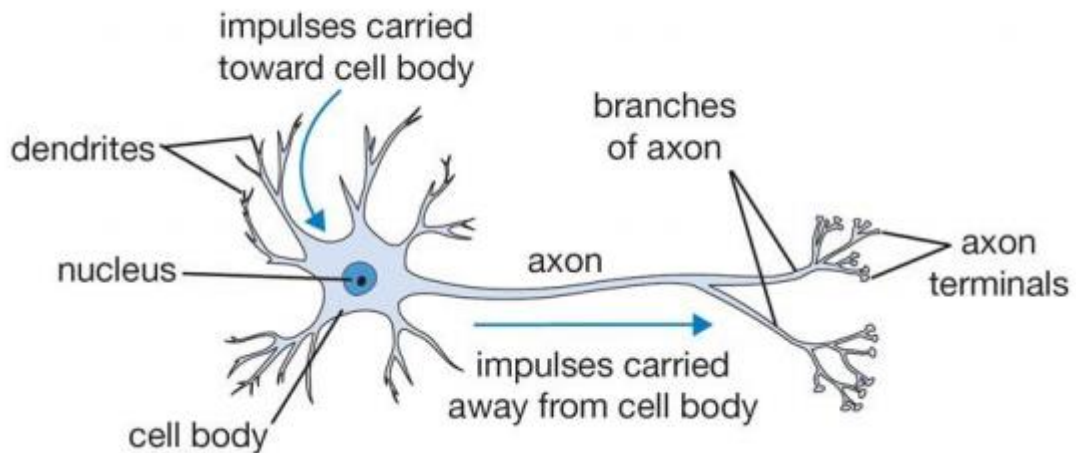


Рисунок 2.3 – Біологічна модель нейрона [16]

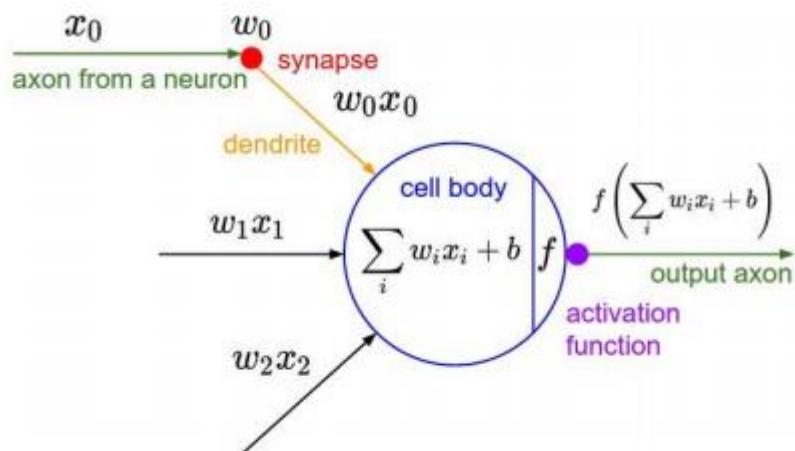


Рисунок 2.4 – Математична модель нейрона [16]

Інакшими мовами, будь-який нейрон виконує скалярне множення його входу із його вагами, додає зміщення та застосовує нелінійність (або функцію активації), в цьому випадку це сигмоїду (Рис. 2.4).

2.3.1 Архітектура нейронних мереж

Нейронні мережі - це нейрони, які згруповані в графі. Нейронні мережі моделюються як сукупність декількох нейронів, які з'єднані в ациклічному графіку [20]. Протилежними словами, вихід декотрих нейронів може бути входом інших нейронів [21]. Моделі нейронних мереж нерідко організовані в відмінні шари нейронів. Для звичайних нейронних мереж найпоширенішим видом шару є повністю зв'язаний шар, в якому нейрони між двома поміжними шарами повністю спарені, але нейрони в одному шарі не мають спільних зв'язків. У зорі перейнятливих поле одного сенсорного нейрона - це конкретна область сітківки, в якій щось впливатиме на випалення цього нейрона (тобто буде активувати нейрон). Кожна сенсорна клітка нейрона має схожі рецептивні поля, і їх поля перекриваються (Рис. 2.5).

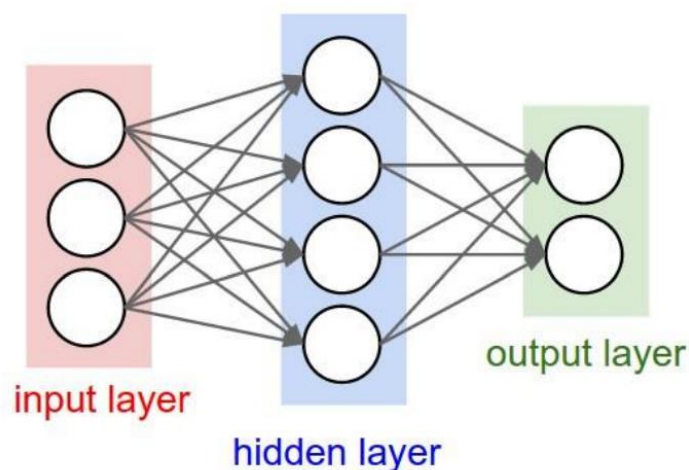


Рисунок 2.5 – Двошарова нейронна мережа (один прихований шар з чотирма нейронами і один вихідний шар з двома нейронами), та три входи [16]

На відміну від усіх шарів нейронної мережі, нейрони вихідного шару часто не виконують активації, тому що вони мають лінійну функцію активації ідентичності. Це, в свою чергу, прив'язано з тим, що останній рівень джерел зазвичай приймається для подань, які є довільними дійсними числами, або конкретної цілі для дійсних значень, одним з прикладів є регресія.

Прямий обхід нейронної мережі або обчислення зворотного зв'язку. Повторне множення матриці часто можна сплутати з функцією активації. Одним з головних слоїв, по якому нейронні мережі організовані в шари, є те, що структура робить оцінку нейронних мереж за допомогою матричних векторних операцій дуже продуктивним, простим і швидким. При роботі з 3-шаровою нейронною мережею вхідним сигналом буде вектор $[3 \times 1]$. Всі шари тягарів можна вільно зберігати в одній матриці. Пряма вага прихованого шару $W1$ має розмір $[4 \times 3]$, а зміщення для всіх нейронів буде у векторі $b1$, який має розмір $[4 \times 1]$. Будь-який з окремих нейронів має ваги в лінії $W1$, тому множення матричного вектора " $np.dot(W1, x)$ " оцінює активацію всіх нейронів у цьому конкретному шарі. Подібним чином, $W2$ буде матрицею $[4 \times 4]$, яка зберігає з'єднання другого прихованого шару, а $W3$ буде $[1 \times 4]$ матрицею для останнього вихідного шару. Загальна пряма послідовність цієї тришарової нейронної мережі має три множення матриць, переплетених з функцією активації [19].

Одним із багатьох засобів опрацювати нейронні мережі з повністю зв'язаними рівнями - визначення набору функцій параметризованих вагами мережі.

Нейронні мережі, що містять хоча б один невидимий шар є універсальними апроксиматорами. Тобто, це може бути показано, що за умови неперервної функції $f(x)$ та деякого $\epsilon > 0$ існує нейронна мережа $g(x)$ з одним прихованим шаром (з розумним вибором нелінійності, наприклад сигмовидної), такими, що $\forall x, |f(x) - g(x)| < \epsilon$ [19].

Використовують більше шарів тому, що той факт, що двошарова нейронна мережа є універсальним апроксиматором, при цьому є відносно слабким і не корисним твердженням на практиці. В одному параметрі функція, що зображена у формулі:

$$g(x) = \sum_i c_i 1(a_i < x < b_i)$$

де a , b , c є векторами параметрів, є також універсальним апроксиматором, але аніхто не вважає за потрібне використовувати цю функцію в машинному навчанні. Нейронні мережі дуже добре працюють на практиці, це є чудовим плюсом. Мотивом є те, що вони компактно виражають гладкі функції, які в свою чергу добре відповідають статистичним властивостям даних, які ми зустрічаємо на практиці [20]. Їх ще легко навчати за допомогою алгоритмів оптимізації (приклад: градієнтний спуск). Ти самим чином, той факт, що глибші мережі, які мають декілька прихованих шарів, можуть працювати краще, ніж мережі з одним-прихованим пластом, є емпіричним спостереженням, незважаючи на те, що їхня представницька сила рівна.

Як на відмінність, на практиці дуже часто буває, що 3-шарові нейронні мережі переважають двошарові мережі, проте виготовлення більш глибоких нейронних мереж (5, 6, 7 шарів) дуже рідко виручає [20]. Це в свою чергу суттєво контрастує зі згортковими нейронними мережами (Convolutional Neural Networks, CNN), де облічено, що саме глибина є надзвичайно важливим компонентом для гарної системи розпізнавання. Одинокий аргумент для цього спостереження є те, що зображення поміщають в ієрархічну структуру. Тому кілька рівнів обробки виготовлюють інтуїтивний глузд [20].

Зі зростом розміру та кількості шарів у нейронній мережі пропускна здібність мережі значно збільшується. Мається на увазі, що простір представлених функцій зростає, адже нейрони можуть співпрацювати для

вираження багатьох різних функцій. Можна обучити три окремі нейронні мережі, кожна з яких має самотній прихований шар певного розміру, і отримати класифікатори як на Рис. 2.6 [15]:

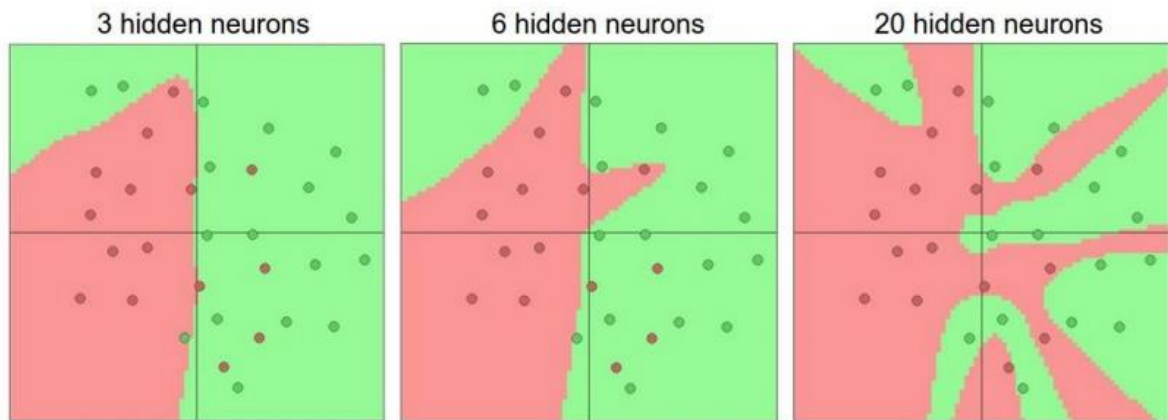


Рисунок 2.6 – Результати тренування мережі з 3, 6 та 20 нейронами у прихованому шарі [15]

Наведена вище діаграма ілюструє, що нейронні мережі з більшою кількістю нейронів можуть виражати більш складні функції. Це одночасно і добре, оскільки можна вивчитися класифікувати більш складні дані, і одночасно погано, бо набагато легше перевчити дані навчання. Перепідготовка відбувається в моментах коли модель великої ємності підходить для шуму даних, а не відношень даних як очікувалося. Наприклад, модель з 21 невидимими нейронами підходить для всіх навчальних даних, але ціною сегментації простору може неправильно диференціювати зелений сегмент та червоний сегмент. Модель же з 4 невидимими нейронами має лише репрезентативну спосібність класифікувати дані. Вона вдає дані у виді двох блоків і інтерпретує кілька червоних крапок усередині зеленого скупчення як деякі викиди [19]. На досвіді це може довести до кращого узагальнення на тестовому наборі.

Якщо дані замало складні для запобігання перенавчання, то прерогатива менших нейронних мереж може бути кращою в даному випадку. Проте це вважається неправильним. Існує купа інших переважних способів запобігання перенавчання в нейронних мережах. На досвіді, завше слід значніше

використовувати методи для інспектування перенавчання, а не за кількістю нейронів.

Тотожний мотив полягає в тому, що менші мережі набагато важче тренувати за допомогою таких місцевих(локальних) методів, як градієнтний спуск. І наспроти, більші нейронні мережі містять значно більше локальних мінімумів, але ці мінімуми виявляються безкінечно кращими з точки зору їх фактичної втрати. Якщо йде тренування маленької мережі, то остаточна втрата може відображати гарну кількість дисперсії. У деяких випадках повезе, але в деяких випадках можна потрапити в один з жахливих мінімумів. З іншого боку, якщо йде випробування великої мережі, почнеться знаходження великої кількості різних рішень, але різниця в остаточному досягненні втрат буде набагато меншою. Інакше кажучи, всі рішення більш-менш однаково гарні, і менше покладаються на успіх неочікуваної ініціалізації.

2.4 Згорткові нейронні мережі

Convolutional neural network (CNN) схожі на звичайні нейронні мережі. Вони складаються з нейронів, які мають вагові коефіцієнти та зміщення. Вся мережа описує єдину розрізнявальну оціночну функцію. З вихідних пікселів зображення на одній межі мереж отримується ступінь приналежності до того, чи іншого класу. Також, цей тип мереж має функцію втрат. Одні із варіантів це Softmax / SVM на останньому повнозв'язному шарі [22].

Ваговитою відмінністю архітектури CNN є гіпотеза, що вхідними даними є зображення, що в свою чергу дозволяють змінити певні особливості цих зображень у мережеву архітектуру. Це служить роботі передачі більш дійовою у реалізації та дуже зменшує кількість параметрів у мережі [22].

Комп'ютери «бачать» не так, як ми. Їх світ складається лише з чисел. Усяке зображення може бути представлене у вигляді двовимірних масивів чисел, відомих як пікселі. Але той факт, що вони вловлюють зображення по-іншому, не означає, що ми не можемо навчити їх розпізнавати візерунки, як ми.

Ординарна нейронна мережа. Нейронні мережі отримують вхід, тобто один вектор, і перетворюють його через ряд невидимих шарів. Кожен прихований шар складається з деякого набору нейронів, де кожен нейрон безповоротно зв'язаний з іншими нейронами в попередньому шарі. Нейрони, що знаходяться в одному шарі, функціонально незалежні та не мають спільних зв'язків [16]. Останній всебічно зв'язний шар вживається "вихідний шар" і в налаштуваннях класифікації він представляє ступінь приналежності до того, чи іншого класу.

Звичайні нейронні мережі дуже негарно масштабуються, зазвичай, для надвеликих зображень. У «CIFAR-10», зображення мають розмір 32x32x3 (32 ширина і висота та 3 кольорові канали), так що один глибоко зв'язний нейрон в першому прихованому шарі звичайної нейронної мережі буде мати в сумі 3072 ваг, тобто перемноження його розмірів [21]. Ця сума не є дуже велетенською для керування нею, але видно, що ця всебічно зв'язна структура не масштабується для гігантських зображень, що може бути проблемою. Наприклад, зображення більшого розміру, 400x400x4, призведе до того, що в нейронній мережі буде 64000 ваг. Така повно зв'язність є марнотратною і велика кількість вимірів може швидко привести до перенавчання, що є поганим наслідком [19].

Тривимірна розмірність нейронів. Вирішальною перевагою CNN над іншими варіантами є те, що ходами є самі зображення, які визначають архітектуру найбільш рівним шляхом. Наприклад, на відміну від ординарної нейронної мережі, шари в «ConvNet» мають нейрони розташовані в трьох вимірах (висота, ширина, глибина). Нейрони в шарі будуть сполучаються

тільки з дрібною областю шару перед ним. Крім того, кінцевий вихідний шар для CIFAR-10 має розмір $[1 \times 1 \times 10]$, тому що в кордоні архітектури CNN повне зображення меншає в єдиний вектор класу оцінки [21].

Будь-який шар CNN змінює тривимірний вхід у тривимірний вихід функції активації. Червоний вхідний шар репрезентує зображення, тому ширина та висота буватимуть вимірюваннями зображення.

Шари, що використовуються для будови CNN. Кожен шар CNN змінює одну розмірність активацій в іншу, обходячи через диференціюючу функцію. Зазвичай використовуються 3 типи шарів для побудови архітектури CNN. Ці три шари є: «convolutional layer», «pooling layer», «fully-connected layer». Поєднання цих шарів і організовує архітектуру CNN [22].

Проста архітектура мережі для CIFAR-10 набору (Рис. 2.7) [21]:

1. INPUT буде містити вихідні значення пікселів зображення, в цьому випадку це ширина 32 пікселів, висота 32 пікселів та глибина, яка однака трьом по кількості колірних каналів [21];

2. CONV шар, (згортковий шар) буде виважати вихід нейронів, які з'єднані з місцевими областями на вході, де кожен обрахунок полягає в скалярному описі між вагами та областю входу, з котрим вони з'єднані [21];

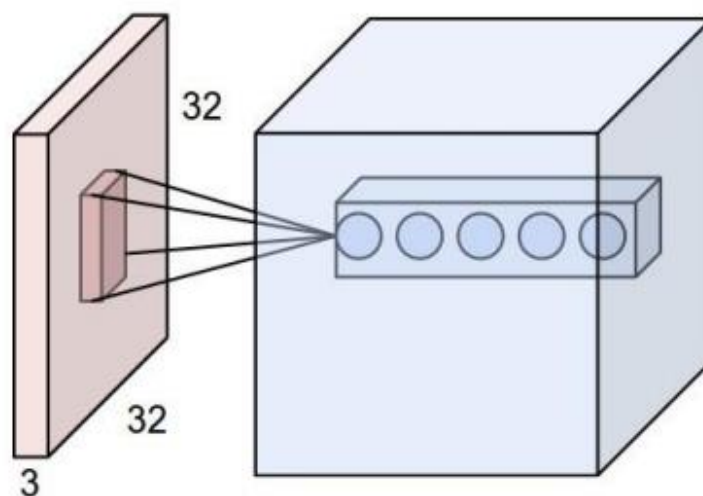


Рисунок 2.7 – Приклад розмірності входу ($32 \times 32 \times 3$) та зразок розмірності нейрону CONV шарі [17]

3. ReLu (Rectified Linear Units), шар по-елементно застосовує функцію активації, таку як $\max$ з порогом в нулі. Розмірність втримується без змін [21];

4. POOL шар буде виконує операцію субдискретизації просторових розмірів (ширина, висота), в результаті чого розмірність стає рівною $[16 \times 16 \times 12]$ [21];

5. Рівень FC обчислює ступінь приналежності до класу. Результат матиме розмірність $1 \times 1 \times 10$, в якій кожне з 10 остніх чисел відповідає ступеню приналежності до певного класу, наприклад, до 10 класів у «CIFAR- 10». Як і у звичайних нейронних мережах, будь-який нейрон цього шару буде повним чином зв'язаний з усіма нейронами [21].

Подібним чином CNN змінює пікселі вихідного зображення шар за шаром до остаточного значення оцінки. Деякі шари містять розміри, а інші - ні [22]. Рівні CONV та FC виконують перетворення, яке є результатом не тільки активації всього вхідного об'єму, але й вимірювань. З іншого боку, шари ReLu та POOL забезпечують передбачувану функціональність. Вимірювання в шарах CONV та FC вивчаються за допомогою методу градієнтного спуску, тому оцінка, яку CNN обчислює, відповідає міткам у навчальному наборі для кожного зображення [22].

Першою особливістю CNN є «local connectivity», або місцеве з'єднання. Якщо мірка входів дуже велика, наприклад, як в цьому випадку - зображення, недоцільно поєднувати нейрони за положенням "all for all" [23]. Натомість кожен нейрон підключається лише до локальної або місцевої області впровадження, лише до якоїсь частини. Просторова довжина зв'язку - це гіперпараметр, який вживається полем сприймання нейронів.

Згорткові нейронні мережі мають іншу архітектуру, ніж звичайні нейронні мережі [24]. Регулярні нейронні мережі перетворюють вхідні дані, прокладаючи їх через низку прихованих шарів. Кожен шар складається з набору нейронів, де кожен шар повністю зв'язаний з усіма нейронами в шарі

раніше. Нарешті, є останній повністю зв'язаний рівень - вихідний рівень - який представляє передбачення [25].

Backpropagation, або зворотне поширення. Зворотний прохід для операції згортки, як для даних і ваг, також є згорткою, але вже для фільтрів [24].

Всі нейрони в одному зрізі по глибині використовують один і той же вектор ваги, тому при проходженні CONV шару кожен зріз по глибині може бути обчислений як згортка ваг нейронів [26]. Таким чином, цей вектор є загальним для набору ваг в якості фільтра (або ядра), який є згорнутим з входом. Наслідком цієї згортки є карта activation map (карта особливостей) [27].

2.5 Висновки до розділу

Машинне навчання як знаряддя, дозволяє вирішувати проблеми, які важко вирішити за допомогою певного алгоритму чи програми, де потрібно постійно адаптуватися до непередбачених та нових входів. Цей клас завдань включає завдання розпізнавання, тобто присвоєння об'єкта відомому конкретному класу. Саме розпізнавання ділиться на кілька під задач: виділення об'єкта, обрання об'єктів на ньому, віднесення до певного класу.

Нейронна мережа може бути використана як класифікатор для присвоєння відібраного об'єкта певному класу. Це математична модель біологічних нейронних систем, що складаються з нейронів та зв'язків між ними. Ідея цієї моделі полягає в тому, що після тренування (регулювання ваги нейронів) така модель зможе працювати з новими, раніше невідомими даними. Спіральна нейронна мережа найчастіше використовується для розпізнавання зображень, оскільки її основною атрибутом є тривимірна розмірність нейронів, що вкорочує загальну кількість зв'язків між усіма нейронами (не всі шари повинні бути повністю зв'язані). Завдяки цьому ця мережа краще працює з такими ієрархічними даними, як зображення (тобто стає можливим вибір додаткових функцій із уже вибраних функцій).

РОЗДІЛ 3. ОПИС ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

Основним завданням цієї роботи та проекту було створення класифікатора інтерферометра для розпізнавання різних гравітаційних хвиль.

Робота проводилась у декілька етапів:

- 1) підбір методів машинного навчання для класифікації;
- 2) підготовка даних;
- 3) обробка даних;
- 4) реалізація методів;
- 5) запуск програми на підготовлених даних;
- 6) збір та аналіз результатів.

3.1 Опис вхідних даних

Обсерваторія гравітаційно-хвильових лазерних інтерферометрів (LIGO) була розроблена, щоб відкрити поле астрофізики гравітаційно-хвильових хвиль шляхом рівного виявлення гравітаційних хвиль, передбачених Загальною теорією відносності Ейнштейна. Багатокілометрові детектори гравітаційних хвиль LIGO використовують лазерну інтерферометрію для вимірювання хвильових хвиль у просторі-часі, спричинених проходженням гравітаційних хвиль від катаклізмичних космічних подій, таких як зіткнення нейтронних зірок або чорних дір, або наднових. LIGO складається з двох широко розділених інтерферометрів у Сполучених Штатах - одного в Ханфорді, штат Вашингтон, а іншого в Лівінгстоні, штат Луїзіана, - які працюють в унісон для виявлення гравітаційних хвиль [13].

Вхідні дані - цей набір даних складається із зображень LIGO, які люди класифікували як частину проекту Gravity Spy Zooniverse. Оригінальний набір

даних, був дещо адаптований для набору даних Kaggle, оскільки осі зображень були видалені та розміщені в папках train, validation та test, готових до машинного навчання (Рис. 3.1).

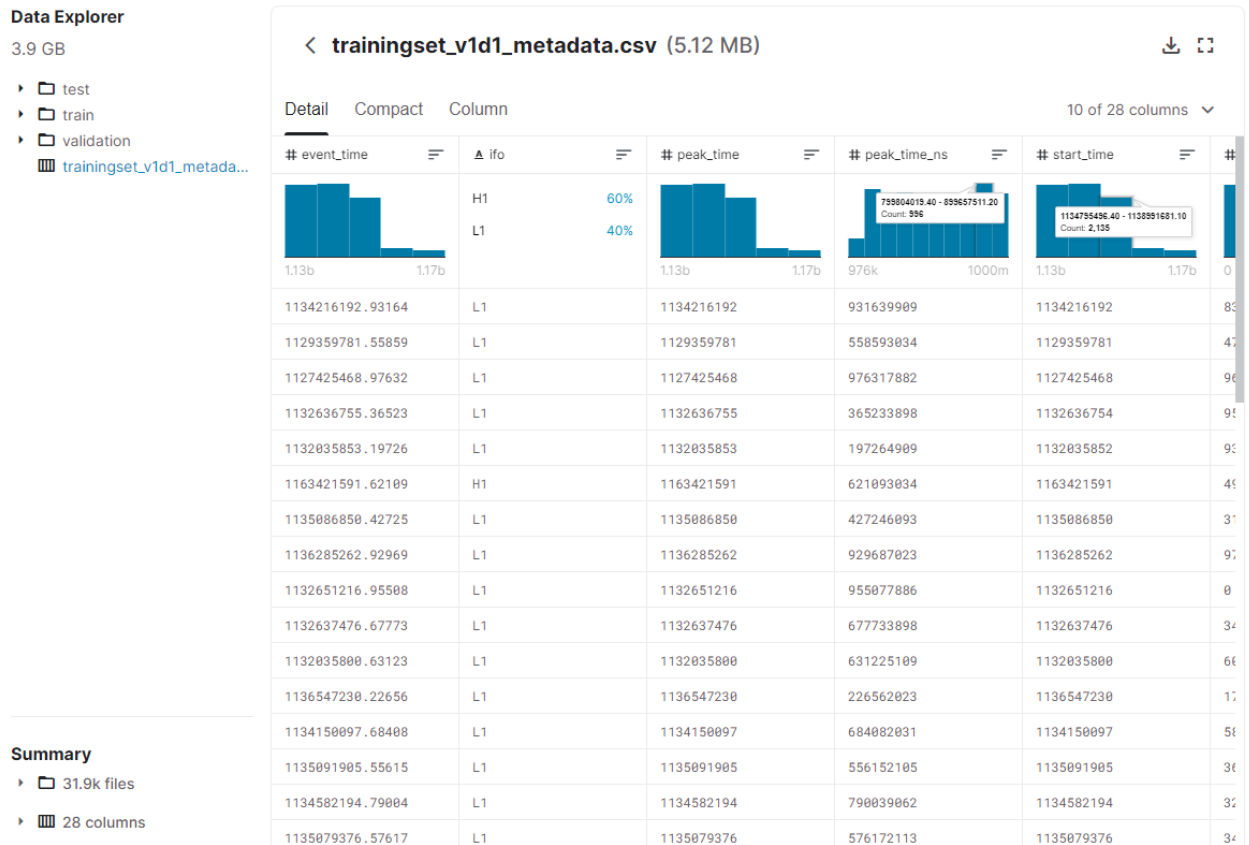


Рисунок 3.1 – Вхідні дані

Інформація зберігається у трьох папках:

- 1) test - папка з даними для тестування моделі;
- 2) train – папка з даними для тренування моделі;
- 3) validation – папка з даними для перевірки роботи моделі.

Кожна папка містить види гравітаційних хвиль у вигляді зображень з сумарною кількістю у більше ніж 25000 png файлів, що розподілені на 22 класи (Рис. 3.2).

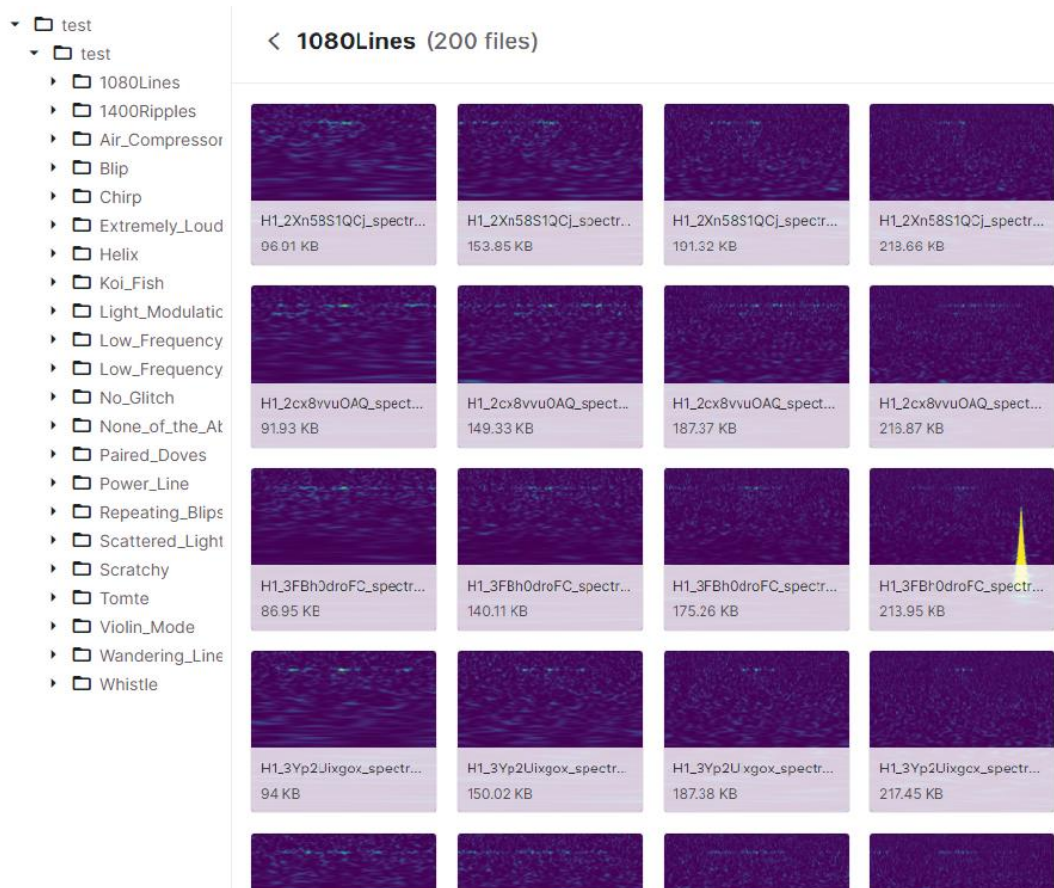


Рисунок 3.2 – Зображення виду «1080Line» гравітаційних хвиль у папці test

3.2 Попередня обробка даних

Перед початком роботи програми дані потрібно обробити.

Мінімальною одиницею даних є зображення гравітаційної хвилі яке має 4 канали та приблизні розміри (500x500) (Рис. 3.3).

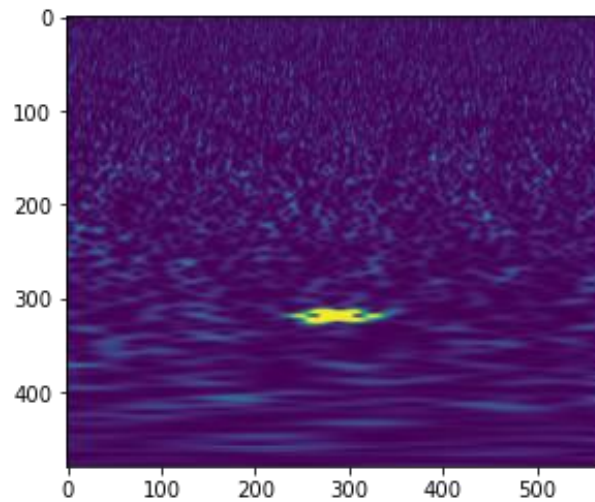


Рисунок 3.3 – Приклад зображення класу «Power Line»

Деякі зображення не мають чітку висоту чи ширину, тому була використана функція `rescale` для масштабування файлів (Рис. 3.4). Результат роботи (Рис. 3.5)

```
In [6]: TRAINING_DIR = '/kaggle/input/gravity-spy-gravitational-waves/train/train/'
training_datagen = ImageDataGenerator(rescale = 1./255)

VALIDATION_DIR = '/kaggle/input/gravity-spy-gravitational-waves/validation/validation/'
validation_datagen = ImageDataGenerator(rescale = 1./255)

In [7]: sample.shape[0]

Out[7]: 479
```

Рисунок 3.4 – Масштабування зображень

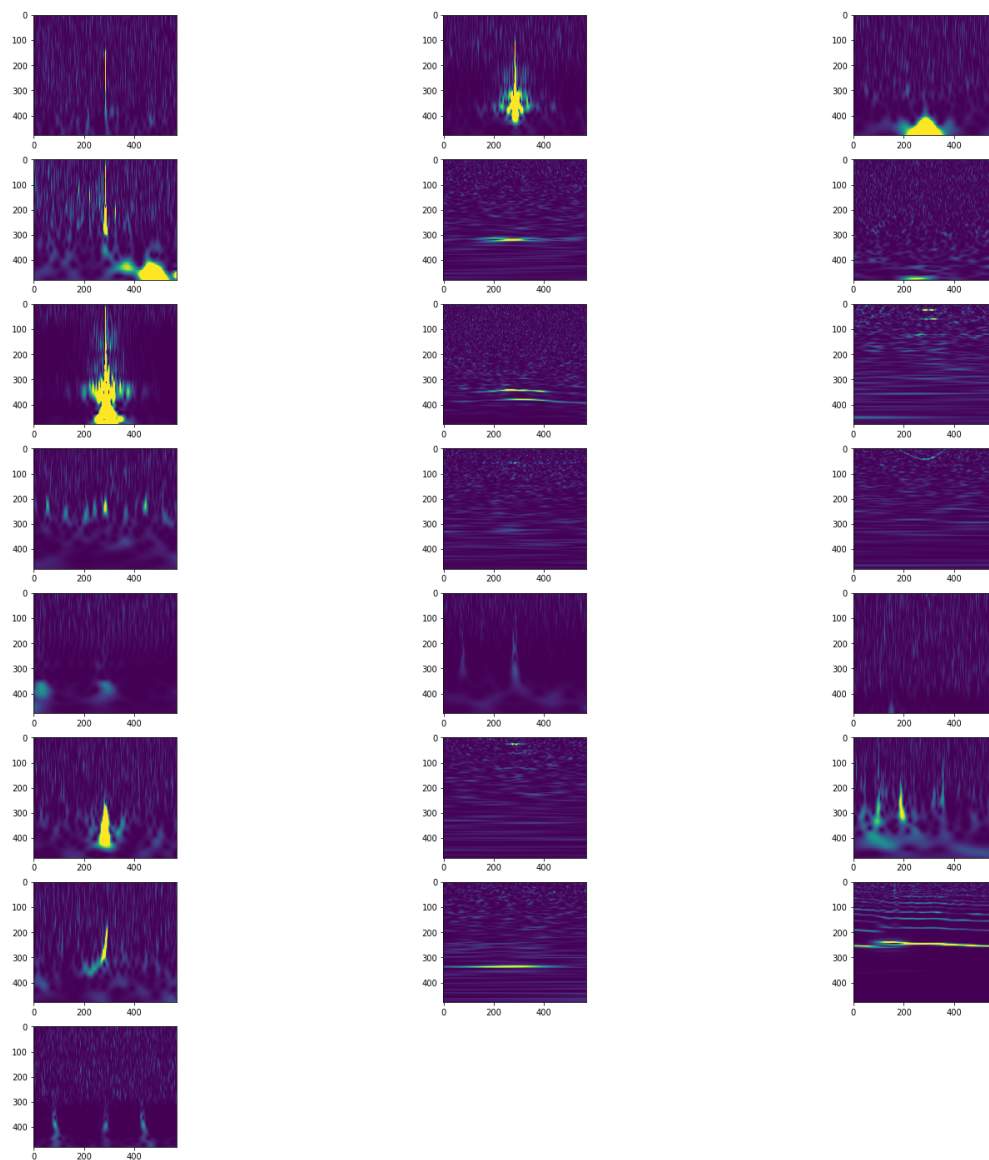


Рисунок 3.5 – Результати роботи rescale

Було проведено огляд класів та зображень директорій train та validation. Результати показали, що test зберігає 22348 зображень, а validation 4800. Обидва випадки мали 22 класи, що було очікувано (Рис. 3.6). Test директорія показала 4720 зображень та 22 класи.

```
In [8]: train_generator = training_datagen.flow_from_directory(
        TRAINING_DIR,
        target_size=(sample.shape[0], sample.shape[1]),
        batch_size = 32,
        class_mode='categorical',
        shuffle=True
    )

    validation_generator = validation_datagen.flow_from_directory(
        VALIDATION_DIR,
        target_size=(sample.shape[0], sample.shape[1]),
        batch_size = 32,
        class_mode='categorical',
        shuffle=True
    )

Found 22348 images belonging to 22 classes.
Found 4800 images belonging to 22 classes.
```

Рисунок 3.6 – Огляд train та validation

Було зроблено перерахування міток класу в цифри для зручної роботи моделі (Рис. 3.7).

```
In [9]: train_generator.class_indices

Out[9]: {'1080Lines': 0,
        '1400Ripples': 1,
        'Air_Compressor': 2,
        'Blip': 3,
        'Chirp': 4,
        'Extremely_Loud': 5,
        'Helix': 6,
        'Koi_Fish': 7,
        'Light_Modulation': 8,
        'Low_Frequency_Burst': 9,
        'Low_Frequency_Lines': 10,
        'No_Glitch': 11,
        'None_of_the_Above': 12,
        'Paired_Doves': 13,
        'Power_Line': 14,
        'Repeating_Blips': 15,
        'Scattered_Light': 16,
        'Scratchy': 17,
        'Tomte': 18,
        'Violin_Mode': 19,
        'Wandering_Line': 20,
        'Whistle': 21}
```

Рисунок 3.7 – Перерахування міток

3.3 Опис частин програми

Прикладна частина дослідження була проведена за допомогою мови програмування Python 3.9 та бібліотек TensorFlow, numpy та Pandas, оскільки ці інструментарій найкраще підходить для виконання поставленої задачі, а також для проектів машинного навчання та штучного інтелекту. Python надає доступ до бібліотек та фреймворків для машинного навчання (Machine Learning). Проекти на Python дуже легкі у своїй реалізації, та можуть бути запущені на будь-якому комп'ютері або на будь-якій сьогоdnішній операційній системі [15].

TensorFlow — це комплексна платформа з відкритим вихідним кодом для машинного навчання. Вона має всеосяжну гнучку екосистему інструментів, бібліотек і ресурсів спільноти, які дозволяють дослідникам просувати новітні досягнення в області машинного навчання, а розробникам легко створювати і розгортати додатки на основі машинного навчання.

NumPy — open-source модуль для python, який надає загальні математичні і числові операції у вигляді пре-скомпільовані, швидких функцій. Вони об'єднуються в високорівневі пакети. Вони забезпечують функціонал, який можна порівняти з функціоналом MatLab. NumPy (Numeric Python) надає базові методи для маніпуляції з великими масивами і матрицями. SciPy (Scientific Python) розширює функціонал numpy величезною колекцією корисних алгоритмів, таких як мінімізація, перетворення Фур'є, регресія, і інші прикладні математичні техніки [16].

Pandas — фреймворк, створений для мови програмування Python для роботи з даними та їх аналізу. Він дозволяє працювати з файлами як з структурами. За допомогою pandas можна легко та зручно обробляти великі файли. Також перевагою серед інших інструментів є багато можливостей для аналізу файлових баз даних та часових рядів [16].

В ході тестування програмного продукту були визначені наступні вимоги щодо ПК:

- Операційна система Windows x64, Ubuntu та MacOS;
- 8 гб оперативної пам'яті;
- Intel Core i7-5960X Processor або вище;
- CUDA (Nvidia Library for NN);
- Tensorflow, numpy, pandas;
- Встановлене середовище для мови програмування Python, бажано PyCharm (для редагування проекту)[не обов'язково];
- Google Colab [не обов'язково].

Приведемо опис частин програми.

- Модуль завантаження даних - даний модуль являє собою набір функцій, призначених для завантаження вхідних даних та містить класи і функції, що полегшують доступ до конкретних значень та параметрів вибірки. Модуль pandas використовується для зберігання даних у датасеті з .csv файла отриманої інформації.
- Модуль “Deeplearning_main” - в даному модулі реалізовано реалізація алгоритмів методів машинного навчання, а саме - глибоке навчання.
- Модулі “CNN_main” - в даному модулі реалізовано реалізація алгоритмів методів машинного навчання, а саме – згорткова нейронна мережа.

3.4 Порівняння результатів

Використовуючи Tensorflow та keras була побудовано та натреновано CNN модель (Рис. 3.8).

```
model = tf.keras.models.Sequential([
```

```

tf.keras.layers.Conv2D(64, (3,3), activation='relu',
input_shape=(sample.shape[0],sample.shape[1],3)),
tf.keras.layers.MaxPooling2D(2, 2),

tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Flatten(),
tf.keras.layers.Dropout(0.5),

tf.keras.layers.Dense(512, activation='relu'),
tf.keras.layers.Dense(22, activation='softmax')
])

model.summary()

model.compile(loss = 'categorical_crossentropy', optimizer='rmsprop', metrics=['accuracy'])

```

```

Train for 32 steps, validate for 32 steps
Epoch 1/50
32/32 [=====] - 38s 1s/step - loss: 3.2316 - accuracy: 0.3213 - val
_loss: 1.7998 - val_accuracy: 0.5674
Epoch 2/50
32/32 [=====] - 33s 1s/step - loss: 1.2745 - accuracy: 0.6816 - val
_loss: 0.8806 - val_accuracy: 0.7705
Epoch 3/50
32/32 [=====] - 34s 1s/step - loss: 0.6786 - accuracy: 0.8232 - val
_loss: 0.8063 - val_accuracy: 0.7920
Epoch 4/50
32/32 [=====] - 34s 1s/step - loss: 0.5944 - accuracy: 0.8379 - val
_loss: 0.4790 - val_accuracy: 0.8486
Epoch 5/50
32/32 [=====] - 33s 1s/step - loss: 0.5085 - accuracy: 0.8623 - val
_loss: 0.5619 - val_accuracy: 0.8555
Epoch 6/50
32/32 [=====] - 42s 1s/step - loss: 0.4930 - accuracy: 0.8711 - val
_loss: 0.4568 - val_accuracy: 0.8555
Epoch 7/50
32/32 [=====] - 33s 1s/step - loss: 0.4552 - accuracy: 0.8740 - val
_loss: 0.4118 - val_accuracy: 0.8613
Epoch 8/50
32/32 [=====] - 34s 1s/step - loss: 0.3630 - accuracy: 0.9023 - val
_loss: 0.3204 - val_accuracy: 0.9053
Epoch 9/50
32/32 [=====] - 33s 1s/step - loss: 0.4049 - accuracy: 0.8887 - val
_loss: 0.3561 - val_accuracy: 0.9004
Epoch 10/50
32/32 [=====] - 33s 1s/step - loss: 0.4089 - accuracy: 0.8867 - val
_loss: 0.2495 - val_accuracy: 0.9365
Epoch 11/50
32/32 [=====] - 48s 1s/step - loss: 0.3751 - accuracy: 0.9072 - val
_loss: 0.2879 - val_accuracy: 0.9277
Epoch 12/50
32/32 [=====] - 34s 1s/step - loss: 0.2990 - accuracy: 0.9160 - val
_loss: 0.3510 - val_accuracy: 0.9111

```

Рисунок 3.8 – Тренування CNN моделі

Як було очікувано, після довгого тренування Accuracy для набору даних train та validation перевищила 0,9, що є позитивним результатом (Рис. 3.9). Loss для обидвох також показав гарні результати (Рис. 3.10).

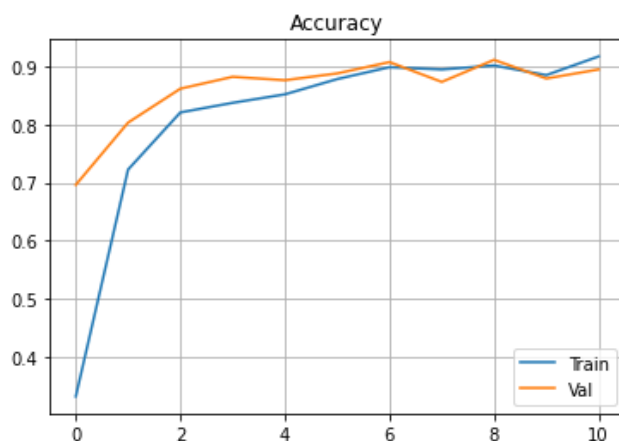


Рисунок 3.9 – Accuracy для train та validation

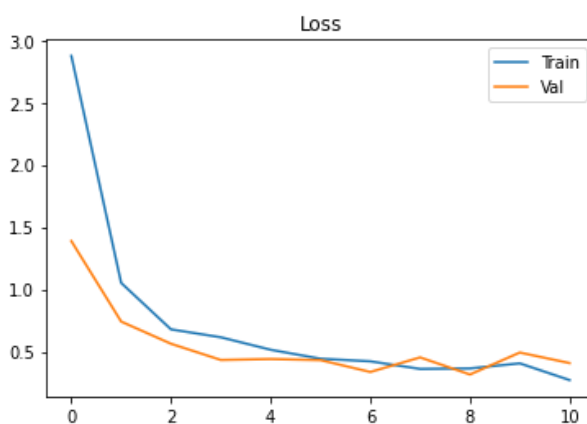


Рисунок 3.10 – Loss для train та validation

Після була проведена оцінка роботи на test даних (Рис. 3.11).

```
TEST_DIR = '/kaggle/input/gravity-spy-gravitational-waves/test/test/'
test_datagen = ImageDataGenerator(rescale = 1./255)

test_generator = test_datagen.flow_from_directory(
    TEST_DIR,
    target_size=(sample.shape[0],sample.shape[1]),
    batch_size = 1,
    class_mode='categorical',
    shuffle=False
)

Found 4720 images belonging to 22 classes.

prediction = model.predict_generator(test_generator, steps = 4720 , verbose = 1)

4720/4720 [=====] - 68s 14ms/step
```

Рисунок 3.11 – Test data evaluation

Була побудована матриця помилок, яка показала, що модель правильно визначає майже усі класи. У вигляді графіку (Рис. 3.12).

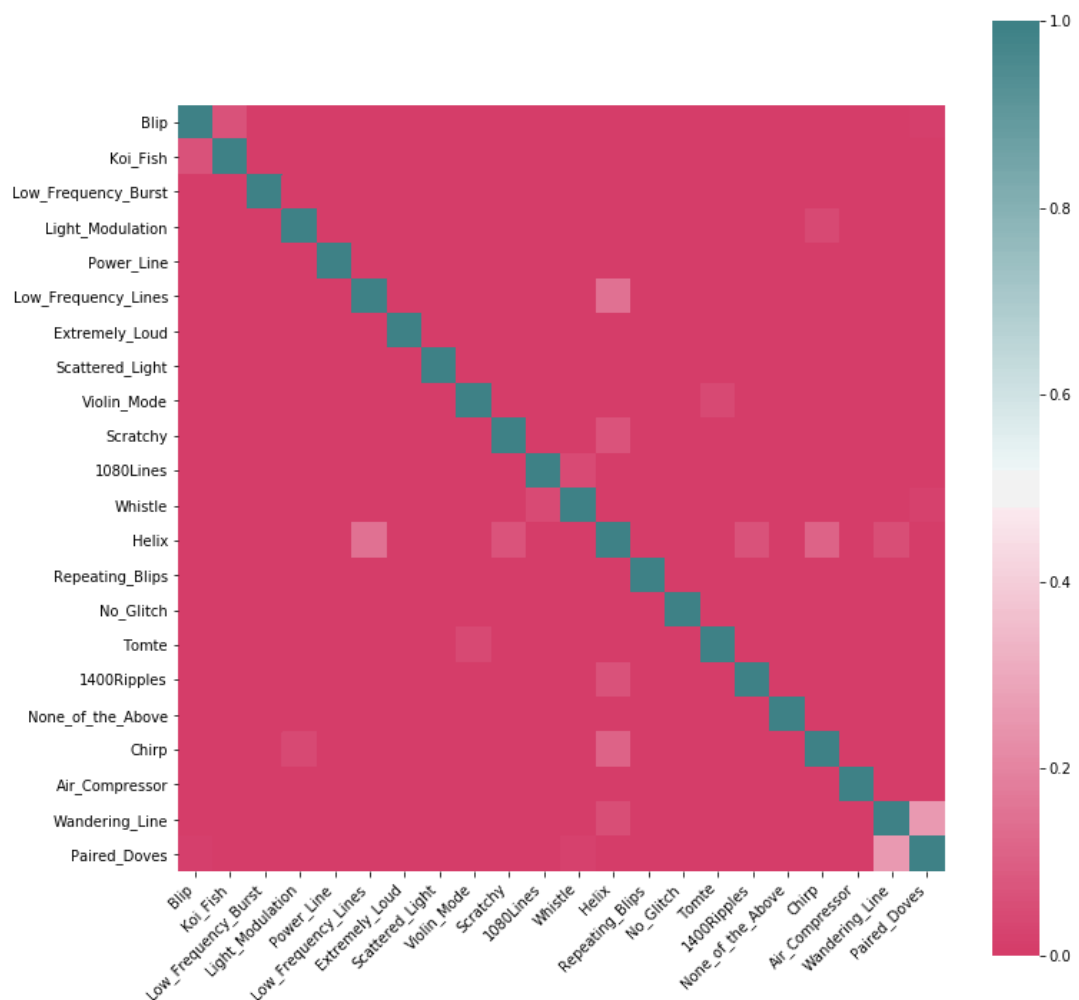


Рисунок 3.12 – Матриця помилок моделі

Використовуючи Keras deep learning model, та той самий датасет було побудовано та натреновано нову модель глибокого навчання (Рис. 3.13).

```
input_shape = (img_dim, img_dim, 3)

model = Sequential()
model.add(tf.keras.layers.Conv2D(32, (10, 10), input_shape=input_shape))
model.add(tf.keras.layers.Activation('relu'))
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))
model.add(tf.keras.layers.Dropout(0.25))

model.add(tf.keras.layers.Conv2D(32, (5, 5)))
model.add(tf.keras.layers.Activation('relu'))
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))
model.add(tf.keras.layers.Dropout(0.25))

model.add(tf.keras.layers.Conv2D(64, (3, 3)))
model.add(tf.keras.layers.Activation('relu'))
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))
model.add(tf.keras.layers.Dropout(0.25))

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(img_dim, activation="relu"))
model.add(tf.keras.layers.Dropout(0.25))
model.add(tf.keras.layers.Dense(22, activation="softmax"))
model.compile(loss = "categorical_crossentropy",
              optimizer = 'adam',
              metrics=["accuracy"])

training_step_size = 32
validation_step_size = 32

history = model.fit_generator(
    train_generator,
    steps_per_epoch = training_step_size,
    epochs = 20,
    validation_data = validation_generator,
    validation_steps = validation_step_size,
    verbose = 1)
```

```

02/02 [====...] - 35s 1s/step - loss: 0.1874 - accuracy: 0.8393 - val
_loss: 0.3727 - val_accuracy: 0.8848
Epoch 12/20
32/32 [====...] - 36s 1s/step - loss: 0.3121 - accuracy: 0.9097 - val
_loss: 0.2862 - val_accuracy: 0.9189
Epoch 13/20
32/32 [====...] - 35s 1s/step - loss: 0.2768 - accuracy: 0.9248 - val
_loss: 0.2793 - val_accuracy: 0.9219
Epoch 14/20
32/32 [====...] - 36s 1s/step - loss: 0.2860 - accuracy: 0.9146 - val
_loss: 0.2357 - val_accuracy: 0.9355
Epoch 15/20
32/32 [====...] - 36s 1s/step - loss: 0.2282 - accuracy: 0.9316 - val
_loss: 0.2736 - val_accuracy: 0.9219
Epoch 16/20
32/32 [====...] - 35s 1s/step - loss: 0.2470 - accuracy: 0.9238 - val
_loss: 0.2357 - val_accuracy: 0.9346
Epoch 17/20
32/32 [====...] - 36s 1s/step - loss: 0.2605 - accuracy: 0.9229 - val
_loss: 0.2550 - val_accuracy: 0.9326
Epoch 18/20
32/32 [====...] - 36s 1s/step - loss: 0.2465 - accuracy: 0.9199 - val
_loss: 0.2618 - val_accuracy: 0.9238
Epoch 19/20
32/32 [====...] - 35s 1s/step - loss: 0.2728 - accuracy: 0.9185 - val
_loss: 0.2715 - val_accuracy: 0.9238
Epoch 20/20
32/32 [====...] - 36s 1s/step - loss: 0.2437 - accuracy: 0.9312 - val
_loss: 0.2514 - val_accuracy: 0.9336

```

Рисунок 3.13 – Тренування Deep learning моделі

Результати роботи моделі, accuracy (Рис. 3.14), loss (Рис. 3.15)

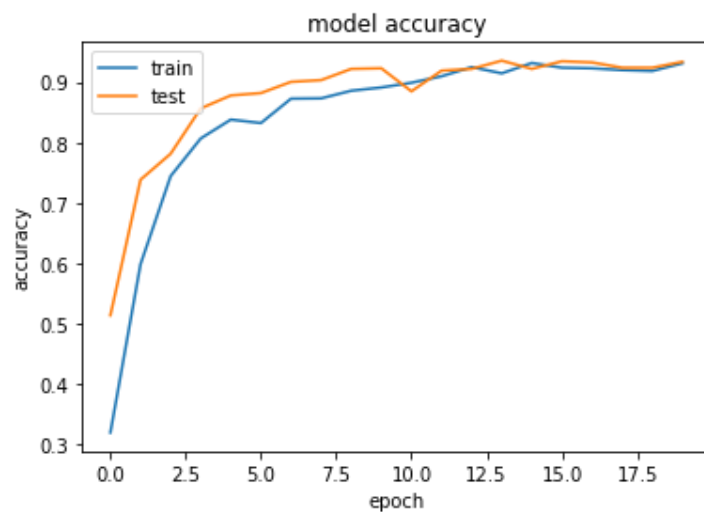


Рисунок 3.14 – Accuracy для train та test

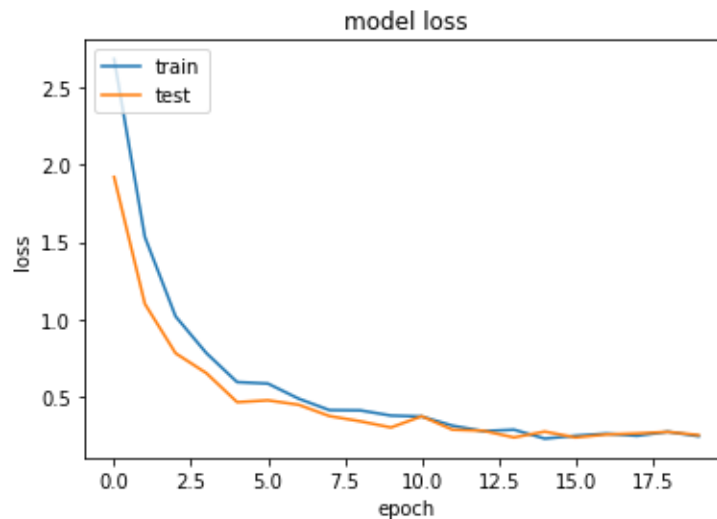


Рисунок 3.15 – Loss для train та test

Зроблений прогнозу на тестових даних (Рис. 3.16). Матриця помилок (Рис. 3.17).

```
In [14]: predictions = model.predict_generator(test_generator, steps = test_size, verbose = 1)
```

4720/4720 [=====] - 97s 20ms/step

Рисунок 3.16 – Результат прогнозу на тестових даних

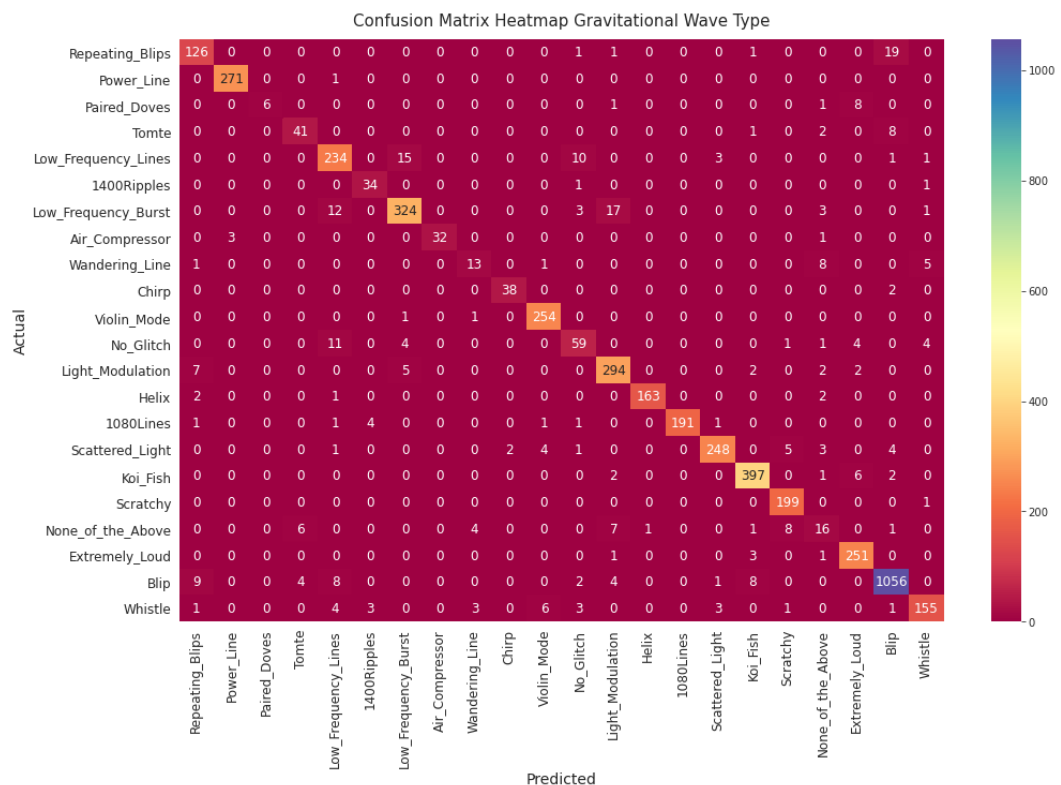


Рисунок 3.17 – Матриця помилок для Deep learning

Архітектура згорткових нейронних мереж: Model 1(Рис. 3.18), Model 2 (Рис. 3.19).

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 477, 567, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 238, 283, 64)	0
conv2d_1 (Conv2D)	(None, 236, 281, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 118, 140, 64)	0
conv2d_2 (Conv2D)	(None, 116, 138, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 58, 69, 128)	0
conv2d_3 (Conv2D)	(None, 56, 67, 128)	147584
max_pooling2d_3 (MaxPooling2D)	(None, 28, 33, 128)	0
flatten (Flatten)	(None, 118272)	0
dropout (Dropout)	(None, 118272)	0
dense (Dense)	(None, 512)	6055776
dense_1 (Dense)	(None, 22)	11286

```

Total params: 60,827,222
Trainable params: 60,827,222
Non-trainable params: 0

```

Рисунок 3.18 – Model 1

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 149, 149, 4)	112
dropout (Dropout)	(None, 149, 149, 4)	0
max_pooling2d (MaxPooling2D)	(None, 74, 74, 4)	0
batch_normalization (Batch Normalization)	(None, 74, 74, 4)	16
conv2d_1 (Conv2D)	(None, 36, 36, 16)	592
dropout_1 (Dropout)	(None, 36, 36, 16)	0
max_pooling2d_1 (MaxPooling2D)	(None, 18, 18, 16)	0
batch_normalization_1 (Batch Normalization)	(None, 18, 18, 16)	64
conv2d_2 (Conv2D)	(None, 18, 18, 32)	4640
conv2d_3 (Conv2D)	(None, 16, 16, 32)	9248
dropout_2 (Dropout)	(None, 16, 16, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 32)	0
batch_normalization_2 (Batch Normalization)	(None, 5, 5, 32)	128
flatten (Flatten)	(None, 800)	0
dense (Dense)	(None, 200)	160200
dense_1 (Dense)	(None, 100)	20100
dropout_3 (Dropout)	(None, 100)	0
batch_normalization_3 (Batch Normalization)	(None, 100)	400
dense_2 (Dense)	(None, 22)	2222

```

Total params: 197,722
Trainable params: 197,418
Non-trainable params: 304

```

Рисунок 3.19 – Model 2

З таб. 3.1 можна зробити висновок, що запропоновані моделі показали високий рівень точності, та для подальшого практичного використання потрібно обрати CNN (Model 1).

Таблиця 3.1 – Порівняння робіт моделей

Модель	Accuracy	Accuracy by model	Accuracy by validation	Precision	Recall	F1-score
Model 1	0.9326	0.9934	0.9242	0.8832	0.9764	0.9353
Model 2	0.9235	0.9931	0.9229	0.8542	0.9673	0.9182
Model 3	0.9214	0.9893	0.9212	0.8567	0.9654	0.9145

Існуючі моделі, як GravySpy Model, мають точність не перевищуючу 0.91. Model 1, Model 2, Model 3 показали кращий результат ніж всі відомі на сьогоднішній день моделі.

3.5 Висновки до розділу

В цьому розділі було проведено практичне моделювання.

Було проаналізовано базу даних обсерваторії LIGO. Проведено обробку даних, масштабування датасету та зображень. Потім було навчено 2 моделі згорткової нейронної мережі для задачі класифікації гравітаційних хвиль. Результати моделювання оцінювались метриками Accuracy, Precision, Recall та F1-score. Графіки для результатів кожної моделі у вигляді кривої помилок.

В результаті обидва методи показали гарний результат. Точність була більша за 0.9. Алгоритми успішно класифікували різні види гравітаційних хвиль у правильні класи. Навчання відбувалося на даних train, та перевірена і протестована на validation, test. Було використано Python 3.9 та бібліотеки TensorFlow, numpy та Pandas, оскільки ці інструментарії найкраще підходять для виконання поставленої задачі, а також для проектів машинного навчання та штучного інтелекту.

Отримана точність є достатньо високою. Збільшити її можливо в ході подальших досліджень шляхом об'єднання різних класифікаторів, додатковою обробкою початкових даних тощо.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП ПРОЕКТУ

На сьогодні великої популярності набуває такий вид підприємництва як стартап. Стартап-проект — є комерційним проектом, який знаходиться в стані розробки, або нещодавно вийшов на ринок. Характерною особливістю стартапу, що відрізняє його від малого бізнесу, є оригінальність та інновації, він не може бути копією вже реалізованих ідей. При цьому проект не обов'язково повинен бути масштабного характеру, головне, щоби він був креативним, а його завдання — спрощувати людям будь-які дії в їх повсякденному житті.

Наразі, з появою Інтернету та сучасних технологій, стало простіше заходити на ринок, знаходити інвесторів та споживачів. З'явилося набагато більше можливостей для розвитку свого проекту за кордоном, ніж раніше.

Проте розробка стартапу є досить ризикованим завданням. Не всім вдається довести свій стартап-проект до ринкового впровадження. За статистикою успіху досягає лише 10–20 % від усіх стартап-проектів.

Запуск стартапу передбачає цілий ряд обов'язкових дій, у межах яких визначають ринкові перспективи стартапу, графік розробки, принципи організації виробництва, заходи з залучення інвесторів та аналіз ризиків.

4.1 Опис ідеї проекту

У таблиці 4.1 описані основні ідеї проекту, можливі напрямки застосування та основні вигоди, що може отримати користувач товару. У таблиці 4.2 визначені сильні, слабкі та нейтральні сторони проекту.

Таблиця 4.1 – Опис ідеї стартапу.

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Програмний продукт для швидкого та точного розпізнання гравітаційних хвиль	Астрономія, Астрофізика, Фізика	Дозволяє проводити швидку обробку даних для подальшої побудови прогнозуючих моделей.

Таблиця 4.2 – Порівняльна характеристика ідеї

Техніко-економічні характеристики ідеї	Мій проект	GraviSpy Identifier	NASA Interferometer API
Ціна	Низька	Висока	Висока
Функціонал	Вузький	Широкий	Широкий

Отже з таблиці видно що сильною стороною є ціна, але функціонал не дозволяє замінити більш відомі конкурентні програмні продукти.

4.2 Технологічний аудит проекту

Наступним етапом стартап-аналізу є технологічний аудит. Його метою є порівняння продукту що розробляється з іншими доступними на ринку продуктами.

В межах даного підрозділу проведено аудит технології, за допомогою якої реалізовано ідею проекту.

Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 4.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту?
- чи існують такі технології, чи їх потрібно розробити/доробити?
- чи доступні такі технології авторам проекту?

Таблиця 4.3 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

<i>Техніко-економічні характеристики ідеї</i>	<i>(потенційні) товари/концепції конкурентів</i>		
	<i>Мій проект</i>	<i>LIGO API</i>	<i>NASA Interferometr API</i>
Точність прогнозування	Кілька моделей з різними результатами	Власний алгоритм	Власний алгоритм
Складання ймовірності класифікації	Присутня, але точність не найкраща.	Дана функція повністю відсутня	Дана функція повністю відсутня
Ризики невірного прогнозу	Існують, через велику кількість факторів.	Невідомо	Відсутні через відсутність прогнозу
Доступність, зручність	Десктопний застосунок	Власний інтерфейс	Власний інтерфейс

4.2.1 Технологічна здійсненність ідеї проекту

В таблиці 4.4 показана технологічна здійсненність ідеї проекту. Ця таблиця показує чому серед аналогічних мов програмування, для дослідження була обрана саме мова програмування Python.

Таблиця 4.4 – Технологічна здійсненність ідеї проекту

<i>Ідея проекту</i>	<i>Технології реалізації</i>	<i>Наявність технологій</i>	<i>Доступність технологій</i>
Створення системи оцінки ймовірності повернення кредиту	Використання мови C++	Відсутні	Недоступні
	Використання мови програмування C#	Відсутні	Недоступні
	Використання мови програмування Python	Наявна	Доступні
Обрана технологія реалізації ідеї проекту: мова програмування Python			

4.2.2 Попередня характеристика потенційного ринку

Значної уваги треба приділити характеристиці ринку (Таблиця 4.5), оскільки подальший ефект від впровадження залежить від зацікавленості основних клієнтів та зацікавлених сторін.

Таблиця 4.5 – Попередня характеристика потенційного ринку автоматизованих прогнозів по поверненню кредитів.

<i>Показники стану ринку (найменування)</i>	<i>Характеристика</i>
Кількість головних гравців, од	50
Загальний обсяг продаж, грн/ум.од	100000
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу (вказати характер обмежень)	Немає
Специфічні вимоги до стандартизації та сертифікації	Немає
Середня норма рентабельності в галузі (або по ринку),%	17 %

4.2.3 Характеристика потенційних клієнтів

Окремо треба розглядати особливості клієнтів (Таблиця 4.6) з метою визначення цільової аудиторії (цільові сегменти ринку) та відмінності у поведінці різних потенційних цільових груп клієнтів.

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності в поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Статистичні параметри роботи моделей, місячний прогноз	Аналітики, аналітичні відділи	Низька ціна, велика кількість статистичних даних.	Простота використання.

Продовження таблиці 4.6

Створення якісного прогнозу	Малі фінансові установи	Цікавить простота у використанні, низька ціна клієнська підтримка	Низька ціна, репутація
Створення точного прогнозу, та постійна довгострокова підтримка	Великі компанії, Банки	Цікавить передусім точність довгострокових прогнозів, клієнська підтримка	Висока якість, бренд, ім'я на ринку, успішний досвід

4.2.4 Фактори загроз

Подальше впровадження проекту тісно пов'язано з факторами загроз, оскільки саме вони дозволяють виявити доцільність реалізації проекту (Таблиця 4.7).

Після визначення потенційних груп клієнтів проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають. Фактори в таблиці подані в порядку зменшення значущості.

Таблиця 4.7 – Фактори загроз

<i>Фактор</i>	<i>Зміст загрози</i>	<i>Можлива реакція компанії</i>
Конкуренція	Цього року очікується вихід на ринок крупної іноземної компанії-конкурента	Пришвидшити вихід програмного продукту
Збут	Ускладнення збуту через цінову політику конкурентів	Розміщення додаткових рекламних банерів в Інтернеті

Також треба визначити фактори можливостей (Таблиця 4.8) та провести аналіз конкуренції (Таблиця 4.9, Таблиця 4.10).

Таблиця 4.8 – Фактори можливостей

<i>Фактор</i>	<i>Зміст можливості</i>	<i>Можлива реакція компанії</i>
Гнучкі ціни	Зменшення ціни товару задля збільшення попиту	Введення власних гнучких цін
Диференціація витрат	Зменшення витрат за рахунок їх перерозподілу	Зменшення витрат на додаткові, непрофільні задачі.

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>У чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства (можливі дії компанії, щоби бути конкурентоспроможною)</i>
1. Вказати тип конкуренції— Досконала конкуренція	Багато систем/команд аналітиків	Розробити впізнаваний продукт, якість, що вирізнятиме нас від конкурентів
2. За рівнем конкурентної боротьби: міжнародний	На ринку присутні системи, розроблені за кордоном.	Розширення аудиторії, розширення списку мов, які підтримуються системою
3. За галузевою ознакою— міжгалузева	Робота із НБУ в різних галузях.	Розширення списку спеціалізованих моделей, де система може бути застосована
4. Конкуренція за видами товарів: товарно-родова	Конкуренція між прогнозами інформаційних систем та команд аналітиків.	Збільшення функціоналу системи
5. За характером конкурентних переваг: Нецінова	Різні способи прогнозування дають різну точність	Розробка кращих(точніших) алгоритмів
6. За інтенсивністю: Марочна	Впізнаваний бренд надає великих переваг	Велику увагу приділити розвитку бренду

4.2.5 Обґрунтування факторів конкурентоспроможності

Проведений аналіз дозволяє виявити фактори конкурентоспроможності (Таблиця 4.10).

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

<i>Фактор конкурентоспроможності</i>	<i>Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)</i>
Багатофункціональність	Жоден конкурент не є настільки багатофункціональним, не здатен на прогноз та порівняння результату моделей.
Якість	Висока якість прогнозу, велика кількість допоміжних статистичних даних
Обслуговування	Робота з клієнтами — передусім великим та середнім бізнесом

4.2.6 SWOT- аналіз стартап-проекту

Для проведення аналізу можуть залучатись різноманітні методи, зокрема SWOT – аналіз (Таблиця 4.11).

Таблиця 4.11 – SWOT- аналіз стартап-проекту

Сильні сторони: Висока якість прогнозу, якість, багатофункціональність	Слабкі сторони: відсутній, інтерфейс користувача, немає налагодженої клієнтської бази
Можливості: Попит, зміна рівня доходів населення, вдосконалення системи	Загрози: конкуренція, збут

Дані з альтернативи ринкового впровадження наведено у таблиці 4.12.

Таблиця 4.12 – Альтернативи ринкового впровадження стартап-проекту

<i>Альтернатива (орієнтовний комплекс заходів) ринкової поведінки</i>	<i>Ймовірність отримання ресурсів</i>	<i>Строки реалізації</i>
Швидкий вихід на ринок із «сирим» продуктом, можливі проблеми із точністю прогнозу та універсальністю	30 %	3 місяці
Поступовий вихід із готовим, відлагодженим продуктом. Висока якість та конкурентоспроможна ціна.	70 %	6 місяців

4.3 Розроблення ринкової стратегії проекту

4.3.1 Вибір цільових груп потенційних споживачів

За результатами аналізу потенційних груп споживачів було обрано цільову групу (Таблиця 4.13), для якої буде запропоновано даний товар, та визначено стратегію охоплення ринку (Таблиця 4.14) — стратегію концентрованого маркетингу (компанія зосереджується на одному сегменті). Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиці 4.15 – 4.17).

Таблиця 4.13 – Вибір цільових груп потенційних споживачів

<i>n/n</i>	<i>Опис профілю цільової групи потенційних клієнтів</i>	<i>Готовність споживачів сприйняти продукт</i>	<i>Орієнтовний попит у межах цільової групи (сегменту)</i>	<i>Інтенсивність конкуренції в сегменті</i>	<i>Простота входу в сегмент</i>
1	Окремі аналітики та аналітичні відділи	Низька готовність	25 %	Висока	Середня
2	Малі та середні аутсорсингові контакт центри	Висока	40 %	Середня	Висока
3	Великі компанії із власними кол-центрами	Висока	20 %	Низька	Середня

Таблиця 4.14 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
2 та 3	Стратегія диференційованого маркетингу	Висока універсальність, багатогалузевість, висока якість, ціна.	Стратегія лідерства по витратах

4.3.2 Визначення базової стратегії конкурентної поведінки

В таблиці 4.15 наведена інформація про базові стратегії конкурентної поведінки.

Таблиця 4.15 – Визначення базової стратегії конкурентної поведінки

<i>Чи є проект «першопрохідцем» на ринку?</i>	<i>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</i>	<i>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</i>	<i>Стратегія конкурентної поведінки*</i>
Ні	Так	Ні	Стратегія виклику лідера

4.3.3 Визначення стратегії позиціонування

Важливим етапом стартап-аналізу проекту є визначення стратегії позиціонування. Вона дозволяє зрозуміти яким чином потрібно позиціонувати товар на ринку. Наведена стратегія у таблиці 4.16

Таблиця 4.16 – Визначення стратегії позиціонування

<i>Вимоги до товару цільової аудиторії</i>	<i>Базова стратегія розвитку</i>	<i>Ключові конкурентоспроможні позиції власного стартап-проекту</i>	<i>Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту (три ключових)</i>
Якість, точність, простота у використанні	Стратегія лідерства по витратах	Якість прогнозу, універсальність, велика кількість статистичної інформації	По іміджу. Позиціонування на низькій ціні. Позиціонування за сферою застосування

4.4 Розроблення маркетингової програми

4.4.1 Ключові переваги

Особливу увагу треба приділити визначенню ключових переваг концепції потенційного товару в таблиці 4.17.

Таблиця 4.17 – Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)		
Статистичні параметри роботи поведінкової моделі, місячний прогноз	Розгорнута статистика, яка охоплює всі особливості роботи поведінкової моделі.	Дана статистика має бути максимально розгорнутою та зрозумілою.		
Створення якісного та точного прогнозу	Точний прогноз — основа функціонування будь-якого Банку.	Розробка коротко- та довгострокових прогнозів, використання різних методів для покращення точності прогнозу		
II. Товар у реальному виконанні	Властивості/характеристики		М/Нм	Вр/Тх /Тл/Е/Ор
	Програмний продукт — складна, комплексна система			
	Якість — тестування сторонніми фірмами			
	Пакування — зручний інсталятор			
	Марка: назва організації-розробника + назва товару			
III. Товар із підкріпленням	До продажу: доставка, гнучкі умови оплати			
	Після продажу: гарантія якості			
За рахунок чого потенційний товар буде захищено від копіювання: Система авторського права				

4.4.2 Формування систем збуту

Для визначення меж цін (Таблиця 4.18) та системи збуту (Таблиця 4.19) можуть залучатись додаткові дані фінансового ринку.

Таблиця 4.18 – Визначення меж встановлення ціни

<i>Рівень цін на товари-замінники</i>	<i>Рівень цін на товари-аналоги</i>	<i>Рівень доходів цільової групи споживачів</i>	<i>Верхня та нижня межі встановлення ціни на товар/послугу</i>
10000	15000	3,00 млн	8000–9000

Таблиця 4.19 – Формування системи збуту

<i>Специфіка закупівельної поведінки цільових клієнтів</i>	<i>Функції збуту, які має виконувати постачальник товару</i>	<i>Глибина каналу збуту</i>	<i>Оптимальна система збуту</i>
Канал нульового рівня	Доставка товару	0	+

4.5 Висновки до розділу

В даному розділі розглянуто стартап-ідею проекту.

Основні цілі, що були досягнуті в цьому розділі:

- 1) з'ясовано, що є можливість ринкової комерціалізації проекту (наявний попит, динаміка ринку, рентабельність роботи на ринку);
- 2) з'ясовано, що є перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження, стан конкуренції;
- 3) конкурентоспроможність проекту є високою;
- 4) подальша імплементація проекту є можливою.

З огляду на потенційні групи клієнтів та бар'єри, які стоять на шляху, можна сказати, що у даного проекту є досить непогані перспективи впровадження. При цьому рівень конкуренції на даний момент є не дуже високим, а конкурентоспроможність проекту є достатньою.

Для ринкової реалізації проекту, на даний момент, краще обрати варіант розробки продукту, при якому буде використовуватися мова програмування Python.

ВИСНОВКИ

Дана робота присвячена дослідженню гравітаційних хвиль за допомогою методів машинного навчання. Було проведено аналіз роботи лазерної інтерферометричної гравітаційно-хвильової обсерваторії (Laser Interferometer Gravitational-Wave Observatory, LIGO) та обсерваторії Virgo. Було ретельно вивчено загальну теорію Ейнштейна.

Дослідження полягало у тому, щоб навчити моделі, методами машинного навчання, класифікувати та розпізнавати види гравітаційних хвиль які було отримано обсерваторією LIGO. Детальний опис методів та засобів для моделювання, обробки вхідних даних та способів оцінювання якості роботи моделей було проведено у другому розділі. Було детально досліджено види машинного навчання, та проведено аналіз, чому вибрані для роботи методи є найкращими інструментами для розпізнання гравітаційних хвиль.

Третій розділ присвячений практичній частині цього дослідження. Була описана практична та програмна складова дослідження. Показані основні бібліотеки та модулі, які використовувались при програмуванні на мові Python. Було проведено обробку вхідних даних, результатами якої стало формування вибірки для навчання і тестування моделей. Результати показали майже 100% точність роботи написаної програми для кожного методу.

В четвертому розділі було проведено стартап-аналіз проекту за результатами якого було визначено, що проект є конкурентноздатним і можлива подальша інтеграція його на ринок.

Проведене дослідження відповідає всім поставленим завданням. Подальший можливий розвиток даного дослідження передбачає підвищення точності моделей класифікації, написання відповідного програмного забезпечення для NASA, ESA, JAXA тощо. Також можливе покращення дослідження шляхом збору більш актуальних даних, або збільшення вибірки за допомогою обсерваторії LIGO.

ПЕРЕЛІК ПОСИЛАНЬ

1. Крячко І. Ю. Гравітаційні хвилі відкрили в 2017 році нову еру в астрономії URL: <http://www.nashenebo.in.ua/nauka/hravitatsiini-khvyli-vidkryly-v-2017-rotsi-novu-eru-v-astronomii>.
2. Gao H. Observation of Gravitational Waves from a Binary Black Hole Merger. *Phys. Rev. Lett.* 2018. Vol.3. P. 198–256
3. Чорногор Л. І. Сенсация XXI сторіччя: Гравітаційні хвилі знайдено, *Світосгляд*. 2016. Том. 1. № 23. С. 2–23
4. Castelveccchi D., Witze A. Einstein's gravitational waves found at last URL: <https://www.nature.com/news/einstein-s-gravitational-waves-found-at-last-1.19361#:~:text=LIGO%20hears'%20space%2Dtime,produced%20by%20black%2Dhole%20collision>.
5. Kostas D. Kokkotas Gravitational Wave Physics. Thessaloniki, Greece: Department of Physics, Aristotle University of Thessaloniki Thessaloniki Press. 2015. 349 p.
6. Jorge L. Cervantes-Cota, Salvador Galindo-Uribarri, George F. Smoot. A Brief History of Gravitational Waves. URL: <https://arxiv.org/ftp/arxiv/papers/1609/1609.09400.pdf>.
7. Alan Weinstein Gravitational Waves and LIGO. URL: https://labcit.ligo.caltech.edu/~ajw/ligo_public.pdf.
8. Ana Caramete, Andrei-Ieronim Constantinescu, Laurent,iu-Ioan Caramete, Traian Popescu et. al., Characterization of Gravitational Waves Signals Using Neural Networks. URL: <https://arxiv.org/pdf/2009.06109.pdf>.
9. Timothy D. Gebhard, Niki Kilbertus, Ian Harry, Bernhard Schölkopf Convolutional neural networks: A magic bullet for gravitational-wave detection? *Journal of Intelligent Information Systems (JIIS)*. 2018. Vol.9. P. 347–360

10. Kyungmin Kim, Ian W Harry, Kari A Hodge, et. al., Application of artificial neural network to search for gravitational-wave signals associated with short gamma-ray bursts. *Computational Intelligence*. 2013. Vol.2. P. 20–149
11. M. Carrillo, J. A Gonz'alez, M. Gracia-Linares, F. S. Guzm'an Time series analysis of Gravitational Wave signals using neural networks. *Systems, Man, and Cybernetics*. 2015. P. 23–153
12. Yuting L., Kentaro S. Detection and Localization of Gravitational Waves Using Convolutional Neural Networks. *Artificial Intelligence*. 2018. Vol. 4. P. 73–152
13. Kari A Hodg, John J Ohm, LIGO Detection Results. *MDPI Applied Sciences*. 2018. Vol. 1. P. 2–102
14. Snapp R. Pattern Recognition, Course Notes. *Journal of Computer and System Sciences*. 2017. P. 3 – 7.
15. Plath B. Neural Networks. URL: <http://cs231n.github.io/neural-networks-1>
16. Convolutional Neural Networks (CNNs / ConvNets). URL: <http://cs231n.github.io/convolutional-networks/>.
17. Martín Abadi, Paul Barham. TensorFlow: A System for Large-Scale Machine Learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*. 2016. P. 24–63
18. Machine Learning Frameworks Comparison URL: <https://blog.paperspace.com/which-ml-framework-should-i-use/>.
19. Deep Learning Part 1: Comparison of Symbolic Deep Learning Frameworks URL: <https://www.r-bloggers.com/deeplearning-part-1-comparison-of-symbolic-deep-learning-frameworks/>.
20. Офіційний сайт TensorFlow. URL: <https://www.tensorflow.org/>.
21. Моделирование процессов обучения в нейронных сетях. URL: <http://old.exponenta.ru/soft/others/mvs/stud3/3.asp/>.

22. Fukushima K. Neural network model for a mechanism of pattern recognition unaffected by shift in position-neocognitron. *IEICE Technical Report*. 1979. Vol.10. P. 658–665.
23. Demush R. A Brief History of Computer Vision (and Convolutional Neural Networks). URL: <https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3>.
24. Freund, Y., Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to Boosting. *Journal of Computer and System Sciences*. 1997. Vol. 55. P.119–139
25. Breiman L. Bagging predictors. *Machine Learning*. 1996. Vol. 24. P.123–140.
26. Wolpert, D.H. Stacked generalization. Neural Networks. Complex Systems Group, Theoretical Division, and Center for Non-linear Studies. *Journal of Computer and System Sciences*. 1992. No. 5 P.241–260.
27. Weiss GM, Provost F The effect of class distribution on classifier learning: An empirical study. *Complex Systems Group, Theoretical Division, and Center for Non-linear Studies*. 2004. No. 8. P 23-27.

ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ

Deeplearning_main.py

```

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator

import os
import numpy as np
import matplotlib.pyplot as plt

import pandas as pd
from sklearn.metrics import confusion_matrix, roc_curve, auc, accuracy_score
from keras.preprocessing import image

import matplotlib.image as mpimg
plt.rcParams.update({'figure.max_open_warning': 0})

train_dir = '../input/gravity-spy-gravitational-waves/train/train/'
validation_dir = '../input/gravity-spy-gravitational-waves/validation/validation/'
test_dir = '../input/gravity-spy-gravitational-waves/test/test/'

train_datagen = ImageDataGenerator(rescale=1. / 255)
validation_datagen = ImageDataGenerator(rescale=1. / 255)
test_datagen = ImageDataGenerator(rescale=1. / 255)

df = pd.read_csv("../input/gravity-spy-gravitational-waves/trainingset_v1d1_metadata.csv")
df.label.value_counts()

fullpath = train_dir + 'Blip/' + os.listdir(train_dir + 'Blip')[0]
img=mpimg.imread(fullpath)
plt.imshow(img)

classes_list = df.label.value_counts().index
classes_list = list(classes_list)

plt.figure(figsize=(25, 25))

for i in range(0, 22):
    plt.subplot(8, 3, i+1)
    fullpath = train_dir + classes_list[i] + '/' + os.listdir(train_dir + classes_list[i])[0]
    img=mpimg.imread(fullpath)
    plt.imshow(img)

```

```

training_batch_size = 64
validation_batch_size = 32
img_dim = 250

train_generator = train_datagen.flow_from_directory(
    train_dir,
    classes = classes_list,
    target_size = (img_dim, img_dim),
    batch_size = training_batch_size,
    class_mode = "categorical",
    shuffle = True,
    seed = 123)

validation_generator = validation_datagen.flow_from_directory(
    validation_dir,
    classes = classes_list,
    target_size = (img_dim, img_dim),
    batch_size = validation_batch_size,
    class_mode = "categorical",
    shuffle = True,
    seed = 123)

test_size = !find './input/gravity-spy-gravitational-waves/test/test/' -type f | wc -l
test_size = int(test_size[0])
test_batch_size = 1

test_generator = test_datagen.flow_from_directory(
    test_dir,
    classes = classes_list,
    target_size = (img_dim, img_dim),
    batch_size = test_batch_size,
    class_mode = "categorical",
    shuffle = False)

input_shape = (img_dim, img_dim, 3)

model = Sequential()
model.add(tf.keras.layers.Conv2D(32, (10, 10), input_shape=input_shape))
model.add(tf.keras.layers.Activation('relu'))
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))
model.add(tf.keras.layers.Dropout(0.25))

model.add(tf.keras.layers.Conv2D(32, (5, 5)))
model.add(tf.keras.layers.Activation('relu'))
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))
model.add(tf.keras.layers.Dropout(0.25))

model.add(tf.keras.layers.Conv2D(64, (3, 3)))
model.add(tf.keras.layers.Activation('relu'))
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))

```

```

model.add(tf.keras.layers.Dropout(0.25))

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(img_dim, activation="relu"))
model.add(tf.keras.layers.Dropout(0.25))
model.add(tf.keras.layers.Dense(22, activation="softmax"))

model.compile(loss = "categorical_crossentropy",
              optimizer = 'adam',
              metrics=["accuracy"])

training_step_size = 32
validation_step_size = 32

history = model.fit_generator(
    train_generator,
    steps_per_epoch = training_step_size,
    epochs = 20,
    validation_data = validation_generator,
    validation_steps = validation_step_size,
    verbose = 1)

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

predictions = model.predict_generator(test_generator, steps = test_size, verbose = 1)

df = pd.DataFrame(predictions)
df['filename'] = test_generator.filenames
df['truth'] = ""
df['truth'] = df['filename'].str.split('/', 1, expand = True)
df['prediction_index'] = df[[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]].idxmax(axis=1)

df['prediction'] = ""

for i in range(0,22):
    df['prediction'][df['prediction_index'] == i] = classes_list[i]

```

```

cm = confusion_matrix(df['truth'], df['prediction'])

cm_df = pd.DataFrame(cm)

cm_df.columns = classes_list
cm_df['signal'] = classes_list

import seaborn as sns

plt.figure(figsize=(12, 12))

corr = cm_df.corr()
ax = sns.heatmap(
    corr,
    vmin=0, vmax=1, center=0.5,
    cmap=sns.diverging_palette(0, 200, n=200),
    square=True
)
ax.set_xticklabels(
    ax.get_xticklabels(),
    rotation=45,
    horizontalalignment='right'
);

accuracy = accuracy_score(df['truth'], df['prediction'])

```

CNN_main.py

```

import numpy as np
import pandas as pd

import os
from google.colab import drive
drive.mount("/content/gdrive")

sample = img.imread('/content/gdrive/Shared drives/Vladyslav Butovskyi/Serhii Stets/Diploma/grav waves data/train/train/Power_Line/H1_9wlnDo4Vtg_spectrogram_4.0.png')

plt.imshow(sample)
print(sample.shape)

from keras_preprocessing.image import ImageDataGenerator

TRAINING_DIR = '/content/gdrive/Shared drives/Vladyslav Butovskyi/Serhii Stets/Diploma/grav waves data/train/train/'
training_datagen = ImageDataGenerator(rescale = 1./255)

```

```

VALIDATION_DIR = '/content/gdrive/Shareddrives/VladyslavButovskyi/Serhii Stets/Diploma/grav waves d
ata/validation/validation/'
validation_datagen = ImageDataGenerator(rescale = 1./255)

sample.shape[0]

train_generator = training_datagen.flow_from_directory(
    TRAINING_DIR,
    target_size=(sample.shape[0],sample.shape[1]),
    batch_size = 32,
    class_mode='categorical',
    shuffle=True
)

validation_generator = validation_datagen.flow_from_directory(
    VALIDATION_DIR,
    target_size=(sample.shape[0],sample.shape[1]),
    batch_size = 32,
    class_mode='categorical',
    shuffle=True
)

train_generator.class_indices

import tensorflow as tf
import keras_preprocessing
from keras_preprocessing import image

model = tf.keras.models.Sequential([

    tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(sample.shape[0],sample.shape[1],3)),
    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.5),

    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(22, activation='softmax')
])

model.summary()

```

```

model.compile(loss = 'categorical_crossentropy', optimizer='rmsprop', metrics=['accuracy'])

callbacks = tf.keras.callbacks.EarlyStopping(patience=2)

history = model.fit_generator(train_generator, epochs=50,
                             validation_data = validation_generator, validation_steps = 32,
                             verbose = 1, steps_per_epoch = 32, callbacks=[callbacks])

plt.grid()

plt.plot(history.history['accuracy'], label = 'Train')
plt.plot(history.history['val_accuracy'], label = 'Val')
plt.title('Accuracy')
plt.legend()
plt.show()

plt.grid()
plt.figure()
plt.plot(history.history['loss'], label = 'Train')
plt.plot(history.history['val_loss'], label = 'Val')
plt.title('Loss')
plt.legend()

TEST_DIR = '/content/gdrive/Shareddrives/VladyslavButovskyi/Serhii Stets/Diploma/grav waves data/test/test'
test_datagen = ImageDataGenerator(rescale = 1./255)

test_generator = test_datagen.flow_from_directory(
    TEST_DIR,
    target_size=(sample.shape[0],sample.shape[1]),
    batch_size = 1,
    class_mode='categorical',
    shuffle=False
)

prediction = model.predict_generator(test_generator, steps = 4720 , verbose = 1)

```