

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

факультет інформаційних технологій

(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня магістра

(бакалавра, спеціаліста, магістра)

Студента Шепегуха Антона Сергійовича

(ПІБ)

академічної групи 126м-20-1

(шифр)

спеціальності 126 «Інформаційні системи та технології»

(код і назва спеціальності)

за освітньо-професійною програмою

«Інформаційні системи та технології»

(офіційна назва)

на тему Обґрунтування використання методів штучного інтелекту

для обробки результатів психологічних досліджень

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	проф. Коротенко Г.М.			
розділів:				
Рецензент	доц. Галушко О.М.			
Нормоконтролер	проф. Коротенко Г.М.			

Дніпро
2022

ЗАТВЕРДЖЕНО:

завідувач кафедри

інформаційних технологійта комп'ютерної інженерії

(повна назва)

Гнатушенко В.В.

(підпис)

(прізвище, ініціали)

« » 2022 року**ЗАВДАННЯ****на кваліфікаційну роботу****ступеня магістр**

(бакалавра, спеціаліста, магістра)

студенту Шепетука А.С. академічної групи 126М-20-1
(прізвище та ініціали) (шифр)спеціальності 126 «Інформаційні системи та технології»

за освітньою-професійною програмою _____

«Інформаційні системи та технології»на тему Обґрунтування використання методів штучного інтелекту
для обробки результатів психологічних досліджень

затверджену наказом ректора НТУ «Дніпровська політехніка» від 10.12.2021 р. № 1036-с

Розділ	Зміст	Термін виконання
Розділ 1	Аналіз стану області рішення задачі	1.10.2021 – 24.10.2021
Розділ 2	Математичні моделі використовувані для досліджень даних x психологічних тестів	25.10.2021 – 30.11.2021
Розділ 3	Аналіз результатів тесту на стійкість до стресу	1.12.2021 – 26.12.2021

Завдання видано _____ Коротенко Г.М.
(підпис керівника) (прізвище, ініціали)Дата видачі 1.10.2021 р.Дата подання до екзаменаційної комісії 17.01.2022 р.Прийнято до виконання _____ Шепетука А.С.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 74 стор., 11 рис., 4 додатки, 40 джерел.

Об'єкт дослідження: методи штучного інтелекту для аналізу результатів психологічних досліджень.

Предмет дослідження: способи підвищення якості застосування методів штучного інтелекту для обробки результатів психологічних досліджень.

Мета кваліфікаційної роботи: обґрунтування підбору метода штучного інтелекту для підвищення швидкодії обробки результатів психологічних експериментів без суттєвого погіршення точності обробки.

Наукова новизна виконаної роботи полягає у використанні методу класифікації за допомогою випадкового лісу для дослідження та поліпшення результатів виконання тесту на стійкість до стресу.

Пояснювальна записка містить аналітичний огляд математичних методів використовуваних під час аналізу результатів експериментальних даних, а також порівняльний аналіз відповідних швидкісних характеристик з обґрунтуванням добору методів випадкового лісу.

Список ключових слів: ШТУЧНИЙ ІНТЕЛЕКТ, ПСИХОЛОГІЧНИЙ ТЕСТ, ЕМОЦІЙНИЙ СТАН, СТРЕС, УВАГА, КОНЦЕНТРАЦІЯ, ВИПАДКОВИЙ ЛІС, НЕЙРОННА МЕРЕЖА, АНАЛІЗ ДАНИХ.

ABSTRACT

Explanatory note: 74 pages, 11 figures, 4 applications, 40 sources.

Object of research: Methods of the artificial intelligence for analysis of results psychological research.

Subject of research: Improving the quality of application of methods of artificial intelligence in handling the results of psychological research.

The purpose of the qualification work: to explain the selection of the method of artificial intelligence to increase the performance of the results of psychological experiments without a significant deterioration of the accuracy of processing.

The scientific novelty of the performed work is to use the classification method with a random forest to study the results of the stress resistance test.

Explanatory note contains an analytical review of mathematical methods used when analyzing the results of experimental data. Comparative analysis of high-speed characteristics and substantiation of the selection of random forest methods.

List of Keywords: ARTIFICIAL INTELLIGENCE, PSYCHOLOGICAL TEST, EMOTIONAL STATUS, STRESS, ATTENTION, CONCENTRATION, RANDOM FOREST, NEURAL NETWORK, DATA ANALYSIS.

ЗМІСТ

Перелік умовних позначень.....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ.....	10
1.1. Виявлення та оцінка емоцій.....	10
1.2. Електроенцефалографія (ЕЕГ).....	11
1.3. Електрокардіографія (ЕКГ).....	13
1.4. Емоційний інтелект.....	15
1.5. Виявлення поточних емоцій людини.....	16
1.6. Розпізнавання емоцій за допомогою штучного інтелекту.....	18
1.7. Висновок до розділу.....	21
РОЗДІЛ 2 МАТЕМАТИЧНІ МОДЕЛІ ВИКОРИСТОВУВАНІ ДЛЯ	
ДОСЛІДЖЕНЬ ДАНИХ Х ПСИХОЛОГІЧНИХ ТЕСТІВ.....	22
2.1. Древа рішень.....	22
2.1.1. Древа класифікації.....	26
2.1.2. Древа регресії.....	27
2.1.3. Методи роботи з деревами.....	27
2.1.3.1. ID3.....	27
2.1.3.2. CART.....	28
2.1.3.3. C4.5.....	30
2.1.4. Висновок по деревах.....	33
2.2. Випадковий ліс.....	34
2.3. Нейромережі.....	37
2.3.1. Архітектура нейронних мереж.....	38
2.3.2. Згорткові нейронні мережі.....	41
2.3.2.1. Згортковий шар.....	42
2.3.2.2. Об'єднуючий шар.....	45
2.3.2.3. Повністю пов'язаний шар.....	46
2.3.3. Типи згорткових нейронних мереж.....	46

2.4. Вибір найкращого алгоритму.....	47
2.4.1. Критерії вибору алгоритму.....	47
2.5. Висновок до розділу.....	51
РОЗДІЛ 3 АНАЛІЗ РЕЗУЛЬТАТІВ ТЕСТУ НА СТІЙКІСТЬ ДО	
СТРЕСУ.....	52
3.1. Аналіз вхідних даних.....	52
3.2. Попередня обробка вхідних даних.....	53
3.3. Реалізація класифікаторів.....	54
3.4. Висновок до розділу.....	55
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57
Додаток А. Відомість матеріалів кваліфікаційної роботи	62
Додаток Б. Програмний код	63
Додаток В Відгук керівника кваліфікаційної роботи.....	73
Додаток Г. Рецензія	74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CNN - Convolutional Neural Network – Згортка нейронної мережі.

LSTM - Long Short-Term Memory – Довга короткочасна пам'ять.

SREIT - Self-Report Emotional Intelligence Test – Тест на емоційний інтелект.

STAI - The State-Trait Anxiety Inventory – Інвентаризація тривожності станів.

STAS - State-Trait Anger Scale – Оцінка проявів гніву.

STAXI - State Trait Anger Expression Inventory Definition – Визначення інвентаризації ознак гніву.

STPI - The State-Trait Personality Inventory – Інвентаризація стану особистості.

TEIQue - Trait Emotional Intelligence Questionnaire – Опитувальник емоційного інтелекту.

ЕЕГ – електроенцефалографія

ЕІ – емоційний інтелект

ЕКГ – електрокардіографія

ІІ – штучний інтелект

ВСТУП

Актуальність роботи. Штучний інтелект у сучасному світі набуває все більш і більш вагомому значення. Він використовується у різних галузях науки, техніки, інформаційних технологій та мистецтва. Його відчутний вплив помітний, у тому числі, у сфері комунікацій та освіти.

Методи штучного інтелекту використовуються у найрізноманітніших галузях наук, в тому числі і в, так званих, когнітивних науках; ці науки досліджують різні аспекти діяльності людського мозку та свідомості з медичними, психологічними, освітніми та іншими цілями.

Когнітивні науки поділяються на два великі класи що до джерел вихідних даних про діяльність мозку та свідомості.

По-перше, це науки, які досліджують нейрофізіологічні зміни під час когнітивних процесів за допомогою спеціальної апаратури, та науки, які опосередковано оцінюють когнітивні процеси за допомогою фіксації змін поведінки, сприйняття, уваги та інших когнітивних проявів.

Друга група наук як правило у якості джерел вихідних даних використовує або реєстраційні дані з медичних установ або результати тестів, які люди виконують добровільно, надаючи таким чином вченим можливість досліджувати когнітивні процеси людської особистості.

Дослідження результатів тестування методів штучного інтелекту дозволяє вчасно помітити відхилення та проблеми у розумовій та ментальній діяльності людини і запобігти небажаним проблемам. Тому тема даної роботи безумовно є актуальною.

Об'єктом досліджень є методи штучного інтелекту, що застосовуються для аналізу результатів психологічних досліджень.

Предметом досліджень є підвищення якості застосування методів штучного інтелекту при обробці результатів психологічних досліджень.

Мета роботи – обґрунтувати добір методу штучного інтелекту для підвищення швидкодії обробки результатів психологічних експериментів без суттєвого погіршення точності обробки.

З огляду на поставлену мету виконується аналіз методів штучного інтелекту, що найчастіше використовуються під час аналізу багатofакторних даних, добираються алгоритми, які потенційно матимуть більшу швидкість виконання та меншу складність за умови несуттєвого погіршення якості прогнозування. Виконується процес випробування обраних моделей на наборах даних з реальних експериментів та на основі проведених досліджень робиться висновок щодо ефективності застосування обраного методу на даних експерименту обраного типу.

Методи дослідження. Під час виконання роботи були досліджені такі математичні моделі як дерева класифікації, випадковий ліс, нейронні мережі прямого розповсюдження, та згорткові нейронні мережі.

Постановка задачі

- Дослідити методи штучного інтелекту що використовуються під час багатofакторного аналізу.
- Реалізувати два обрані класифікатори різного ступеня сили.
- Порівняти швидкодію та якість аналізу з використанням експериментальних даних та обґрунтувати добір певного класифікатора.

Висновок. У цій роботі передбачається проаналізувати використання набору даних результатів тесту на стійкість до стресу за допомогою випадкового лісу з ID3 дерев та згорткової нейронної мережі та показати, що випадковий ліс виконує класифікацію не набагато гірше ніж нейронна мережа, але має порівняно малий час на навчання, що дозволяє гнучко змінювати склад набору даних.

РОЗДІЛ 1

АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ

Вираз обличчя є незамінним джерелом інформації у спілкуванні. Вирішальним компонентом успішної особистісної взаємодії є швидке сприйняття виразів обличчя людини та правильний висновок про внутрішні стани інших, які вони передають. Хоча між теоретиками емоцій тривають дебати щодо функцій та значень виразів обличчя, більшість сучасних підходів припускають, що міміка емоцій визначається результатами оцінки та представляє еферентний вплив останніх на рухову поведінку. Конкретні вирази обличчя — це емоції особи, і вони відіграють вирішальну роль у емоційному спілкуванні. Сприйняття та ідентифікація емоцій на обличчі передбачає ефективність соціально-емоційних показників та оцінки соціально-емоційних навичок.

1.1. Виявлення та оцінка емоцій

Методи оцінки емоцій, представлені в літературі, можна розділити на дві основні групи за основними прийомами, що використовуються для розпізнавання емоцій: методики саморефлексії (самоостійної оцінки), засновані на самооцінці емоцій шляхом заповнення різноманітних анкет [1-3]; методи машинної оцінки, засновані на вимірюванні різних параметрів тіла людини [4-6]. Крім того, часто зустрічаються випадки одночасного використання кількох методів з метою підвищення достовірності отриманих результатів. Відповідно до [7-8], кожен емоційний стан можна оцінити шляхом аналізу п'яти основних компонентів емоції (поведінкові тенденції, фізіологічні реакції, рухові вирази, когнітивні оцінки та суб'єктивні відчуття), але лише перші чотири можна оцінити автоматично та дати вказівки на емоційний стан користувача під час взаємодії, не перериваючи її. Суб'єктивні відчуття зазвичай оцінюються тільки за допомогою прийомів самооцінки.

Автоматичне розпізнавання емоцій зазвичай виконується шляхом вимірювання різних параметрів людського тіла або електричних імпульсів у нервовій системі та аналізу їх змін. Найпопулярнішими методами є електроенцефалографія, вимірювання опору шкіри, артеріального тиску, частоти серцевих скорочень, зорової активності, аналізу рухів тощо.

1.2. Електроенцефалографія (ЕЕГ)

ЕЕГ — це електрофізіологічна неінвазивна методика для реєстрації електричної активності, що виникає в мозку людини [9]. Перша доповідь про застосування цієї методики, представлена німецьким психіатром Гансом Бергером, першовідкривачем ЕЕГ у 1924 році [9]. Сигнали ЕЕГ зазвичай збираються за допомогою спеціального приладу, який називається електроенцефалографом. Основними частинами цього пристрою є спеціальні металеві пластинчасті електроди, які слід розташовувати на шкірі голови людини, а в особливих випадках альтернативні електроди можуть бути введені безпосередньо в шкіру голови [10].

Сигнал ЕЕГ – це коливання напруги між двома спареними електродами відносно часу [11] (рис. 1.1a), а амплітуда сигналу зазвичай оцінюється за допомогою техніки від піку до піку (рис. 1.1).

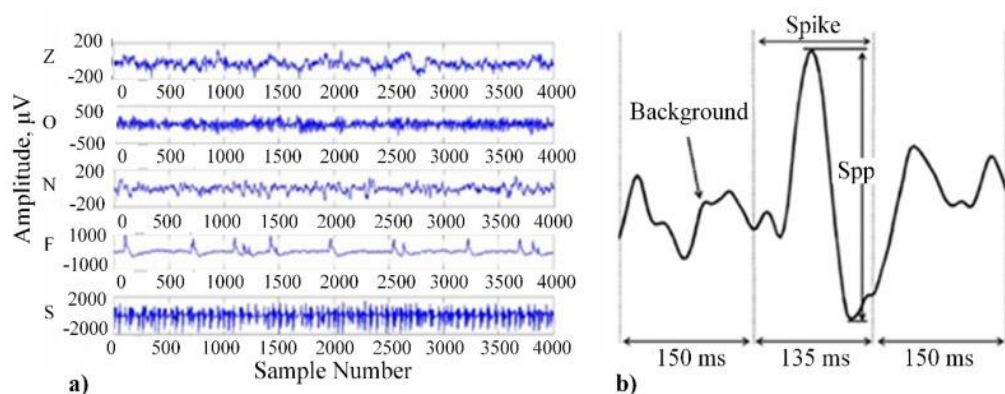


Рис. 1.1. Приклад сигналу ЕЕГ

Для оцінки людських емоцій реакція мозку на різні подразники зазвичай вимірюється та аналізується п'ятьма діапазонами частот від сигналів ЕЕГ, а саме: дельта, тета, альфа, бета і гамма. Ці смугові хвилі всюди присутні в різних частинах мозку [12,13] і пов'язані з різними емоційними станами (Табл. 1.1).

Таблиця 1.1.

Класифікація мозкових хвиль

Тип хвиль	Пов'язаний емоційний стан	Короткий опис
Дельта (δ)(0,5-4Гц)	Сильне почуття емпатії та інтуїції	Найповільніші мозкові хвилі часто асоціюються зі сном. Кілька частот у цьому діапазоні супроводжуються виділенням гормону росту людини, який корисний для лікування. Ці хвилі, що виникають у стані неспання, показують можливість доступу до підсвідомої діяльності.
Тета (θ)(4-8Гц)	Глибоке розслаблення медитація	В основному дорослі виробляють тета-хвилі мозку, коли людина перебуває в легкому сні. Ці хвилі зазвичай з'являються при закритті очей і зникають при відкритті очей. Частота цих хвиль в основному пов'язана зі зняттям стресу і запам'ятовуванням. Сутінкові умови можна використовувати для досягнення більш глибокої медитації, що призведе до покращення здоров'я, а також збільшення творчих здібностей і здібностей до навчання

Продовження табл. 1.1.

Тип хвиль	Пов'язаний емоційний стан	Короткий опис
Альфа (α)(8-16Гц)	Креативність, розслаблення	Ці хвилі здебільшого виникають у стані розслаблення неспання із закритими очима. Альфа - це стан спокою для мозку. Активність альфа-хвиль знижується у відповідь на всі види рухової активності. Альфа-хвилі сприяють загальній розумовій координації, спокою, пильності, інтеграції розуму та тіла та навчанню
Бета (β)(16-32Гц)	Обережність, концентрація.	Бета-хвилі виникають, коли людина перебуває в стані тривоги. Зазвичай вони утворюються в лобовій і центральній частині мозку. У цьому стані мозок може легко виконувати: аналіз, підготовку інформації, генерувати рішення та нові ідеї.
Гамма (γ)(32+Гц)	Навчання, пам'ять і мовна обробка та уявлення.	Ці хвилі випромінюються, коли людина перебуває в аномальному стані або має якийсь психічний розлад. Мозкові гамма-хвилі є найшвидшими з мозкових хвиль і стосуються одночасної обробки інформації з різних областей мозку.

1.3. Електрокардіографія (ЕКГ)

Серце є одним з найважливіших органів в організмі людини, а електрокардіографія (ЕКГ) вважається одним з найпотужніших діагностичних інструментів у медицині, який зазвичай використовується для оцінки функціональності серця. ЕКГ як фізіологічний сигнал використовується як традиційний метод неінвазивної інтерпретації електричної активності серця в реальному часі [14]. Оскільки серцева діяльність пов'язана з центральною нервовою системою людини, ЕКГ корисна не тільки для аналізу діяльності

серця, її також можна використовувати для розпізнавання емоцій [15].

Найважливішими точками на ЕКГ-сигналі є піки: P, Q, R, S, T і U [16] (рис. 1.2). Кожен із цих піків пов'язаний із серцевою діяльністю [16] і має свої особливості. Розпізнавання емоцій за допомогою фізіологічного сигналу є більш складним процесом порівняно з ЕЕГ через його чутливість до артефактів руху та нездатність візуально сприймати емоції з даних [17]. Щоб усунути шуми, викликані зовнішніми факторами, такими як рух об'єкта під час процедури вимірювання [18], ЕКГ зазвичай виконується в захищених від впливу навколишнього середовища просторах, коли людина знаходиться в спокійному стані (рис. 1.2).

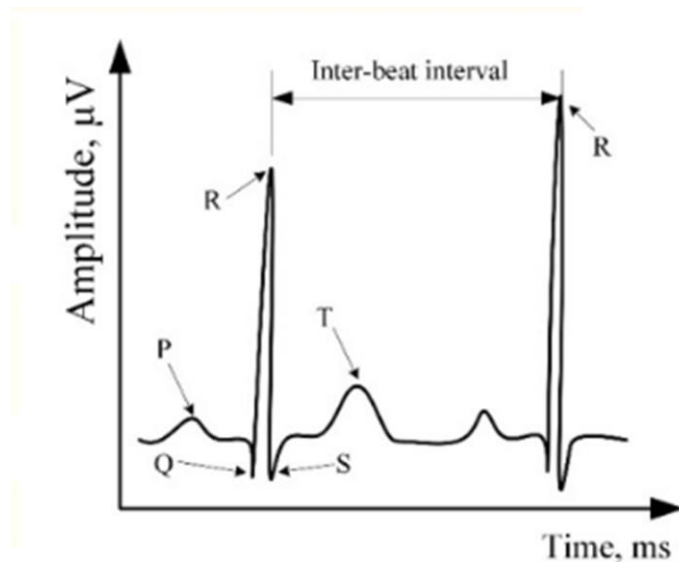


Рис. 1.2. Приклад ЕКГ

Існує п'ять основних параметрів, які часто використовуються для оцінки сигналів ЕКГ. Зазвичай всі п'ять параметрів аналізують лише в медичних цілях. Для розпізнавання емоцій у більшості випадків використовується комплекс QRS, який визначає активацію серця, пов'язану з емоційним станом людини, і є відповідним індикатором для розпізнавання основних емоцій, але також виникають труднощі з розпізнаванням емоцій через те, що цей показник має варіантну чутливість до конкретних емоцій. Результати досліджень Цай,

Лю та Хао [19] показують, що смуток можна розпізнати легше і точніше, ніж емоцію радості. Більшість досліджень, пов'язаних з розпізнаванням емоцій на основі ЕКГ, зосереджені на визначенні та оцінці амплітуд QRS і тривалості між цими хвилями. Далі існує низка досліджень, зосереджених на аналізі дисперсії QT/QTc [20], що дає докази того, що цей інтервал пов'язаний з рівнем тривоги і може використовуватися як маркер для розпізнавання сильного гніву.

Через складність аналізу ЕКГ-сигналу в практичному застосуванні, досить часто ЕКГ використовується в поєднанні з іншими методами розпізнавання емоцій.

Незважаючи на вищеописаний недолік, ЕКГ залишається потужним і перспективним методом розпізнавання емоцій, оскільки дозволяє вимірювати сигнали в організмі людини, які безпосередньо пов'язані з емоційними станами. Те, що багато досліджень зосереджені на створенні нових методів вилучення корисної інформації, дозволяє стверджувати, що розпізнавання емоцій на основі ЕКГ є чудовим засобом для впровадження різноманітних технік машинного навчання. Машинне навчання дозволяє автоматично аналізувати величезну кількість даних і визначати співвідношення між вимірюваннями, що виконуються за різних обставин: стани, коли людина розслаблена або піддається впливу якогось подразника. Крім того, завдяки високій точності ЕКГ, доповненій методами аналізу та обробки сигналів на основі машинного навчання, її можна використовувати для дослідження механізмів сприйняття емоцій та для створення прогностичних моделей на основі довготривалого моніторингу поведінки людини та емоційної реакції.

1.4. Емоційний інтелект

Емоційний інтелект (EI) визначили в 1990-х роках як суму навичок і здібностей людини розпізнавати емоції, розуміти наміри, мотивацію і бажання інших людей і свої власні, а також здатність керувати своїми емоціями та емоціями інших людей з метою вирішення практичних задач. В даний час

розроблено понад 30 різних широко використовуваних заходів для вимірювання EI.

У статті [21] автори поставили за мету є розглянути основні, широко використовувані заходи для вимірювання емоційного інтелекту (EI) та рекомендації щодо їх відповідного використання.

Одним з методів, описаних у статті були психологічні тести та анкети, засновані на самостійній оцінці людей своєї поведінки.

Психологічні тести - це інструмент, який використовують психологи, щоб діагностувати клієнтів. Психологічні тести забезпечують спосіб формально і точно вимірювати різні фактори, які можуть бути причинами проблем людей.

Серед наведених у статті [21] варіантів тестування одним з найцікавіших є дуже хороший, комплексний показник риси EI - це анкета емоційного інтелекту (Trait Emotional Intelligence Questionnaire, або TEIQue). TEIQue - широко використовувана анкета, яка вимірює 4 фактори та 15 граней ознак EI. Анкета була використана у більш ніж 2000 дослідженнях.

Також широко використовуваним, вільно доступним варіантом є Self-Report Emotional Intelligence Test (SREIT). Хоча він був розроблений для вимірювання загального EI, подальші дослідження показують, що він працює краще як багатовимірна шкала, що вимірює 4 окремих чинника, включаючи: оптимізм / регулювання настрою, оцінка емоцій, соціальні навички та утилізація емоцій.

1.5. Виявлення поточних емоцій людини

Тривога, гнів, депресія та цікавість є основними показниками психологічного стану, що вимагає ретельної оцінки.

Вимірювання цих психологічних життєво важливих ознак має вирішальне значення в діагностиці, і може сприяти лікуванню, безпосередньо зв'язуючи інтенсивні емоції з подіями, які породжують їх.

За даними Всесвітньої організації охорони здоров'я [22], емоції, такі як тривога, гнів, біль або радість можуть мотивувати людину до цілеспрямованої дії. Однак, коли певні емоції переважають у психіці людей за їх цілеспрямовану поведінку, емоції стають хворобливими або патологічними. Тому необхідно оцінювати та контролювати емоційні стани в діагностиці так само, як лікарі у медичних оглядах, регулярно вимірюють частоту пульсу, артеріальний тиск, температуру тощо.

Прояви тривожності, гніву та депресії - критичні психологічні життєво важливі ознаки, які сильно пов'язані з добробутом людини. Варіації інтенсивності та тривалості цих емоцій забезпечують основну інформацію про психічне здоров'я людини, яка допомагає визначити останні події та довготривалі конфлікти, які мають значний вплив на життя людини. Оцінка емоцій та забезпечення своєчасної реакції на них підвищить усвідомлення та розуміння почуттів людини, а також допоможе особам розпізнавати та ефективно визнавати свої емоції.

У статті [23] описується шкала STAI (The State-Trait Anxiety Inventory) для забезпечення надійної, відносно швидкої оцінки стану тривожності людини. STAI може бути використаний у клінічних умовах для діагностування тривоги та відокремлення її від депресивних синдромів.

Для шкали був розроблений відповідний тест, що складається з єдиного набору питань, які можна вводити з різними інструкціями для оцінки як постійної, так і тимчасової тривоги.

Форма тривоги Y, найпопулярніша версія тесту, має 20 пунктів для оцінки постійної тривоги та 20 для тимчасової тривоги. Пункти для оцінки тимчасової тривоги включають: «Я напружений; Я переживаю» та «Я відчуваю себе спокійним; Я відчуваю себе в безпеці». Пункти для оцінки постійної тривоги включають: «Я занадто багато хвилююсь над тим, що насправді не має значення» і «Я задоволений; Я стійка людина». Всі пункти оцінюються за 4-бальною шкалою (наприклад, від "майже ніколи" до "майже завжди"). Вищі результати вказують на більшу тривогу.

Також широко відомий такий тест як STAXI [24] (State Trait Anger Expression Inventory Definition). STAXI був розроблений для оцінки тимчасового гніву, постійного гніву та вираження гніву. Щоб дослідити ефекти гніву на психічні та фізичні розлади, досвід гніву повинен бути чітко відділений від вираження гніву та контролю гніву.

Одним з варіантів тесту для оцінки гніву є State-Trait Anger Scale (STAS). Форма для оцінки за цією шкалою складається з 15 питань на визначення постійного гніву та 15 питань для визначення тимчасового гніву.

Також у статті [25] задується тест для розпізнавання емоцій більш широкого спектру – The State-Trait Personality Inventory (STPI). STPI - це анкета самостійної оцінки, призначена для вимірювання тимчасового та постійного гніву, тривоги, цікавості та депресії у дорослих. STPI складається з 8-10 типових підрозділів: тимчасова та постійна тривога, тимчасовий та постійний гнів, тимчасова та постійна цікавість, і тимчасова депресія. Тимчасові ознаки спрямовані на оцінку поточного емоційного стану суб'єкта, і оцінюються за шкалою чотирьох точок інтенсивності. Елементи постійні спрямовані на оцінку емоційного характеру суб'єкта, і оцінюються за чотирикутною частотною шкалою.

1.6. Розпізнавання емоцій за допомогою штучного інтелекту

Автоматичне розпізнавання емоцій є важливою проблемою в різних сферах діяльності, які використовують емоційні реакції людини як сигнал для маркетингу, технічного обладнання або взаємодії людини з роботом.

У роботі [26] автори вивчають спосіб визначення емоцій на основі тексту з використанням штучного інтелекту у великих даних соціальних мереж. Онлайніві соціальні медіа, такі як Facebook і Twitter, стали потужним інструментом для висловлення своєї точки зору за допомогою тексту думок і почуттів людей щодо поточних подій. Розуміння емоцій на тонкому рівні цих висловлених думок є важливим для покращення системи. Таку важливу інформацію неможливо повністю отримати, виконуючи аналіз настроїв на

основі ШІ; отже, визначення емоцій на основі тексту з використанням штучного інтелекту у великих даних соціальних мереж стало майбутнім напрямком досліджень обробки природної мови. Його можна використовувати в різних областях, таких як розуміння виражених емоцій, взаємодія людини та комп'ютера, аналіз даних, онлайн-освіта, рекомендаційні системи та психологія.

Згідно з новим дослідженням Лондонського університету королеви Марії [26], новий підхід штучного інтелекту (ШІ), заснований на бездротових сигналах, може допомогти розкрити наші внутрішні емоції.

Дослідження, опубліковане в журналі PLOS ONE, демонструє використання радіохвиль для вимірювання частоти серцевих скорочень і дихальних сигналів, а також передбачення того, що людина відчуває навіть за відсутності будь-яких інших візуальних сигналів, таких як вираз обличчя.

Спочатку учасникам було запропоновано переглянути відео, відібране дослідниками за його здатність викликати один з чотирьох основних типів емоцій; гнів, смуток, радість і задоволення. Поки людина дивилася відео, дослідники випромінювали нешкідливі радіосигнали, такі як ті, що передаються з будь-якої бездротової системи, включаючи радар або Wi-Fi, на людину та вимірювали сигнали, які відбивалися від неї. Аналізуючи зміни цих сигналів, спричинені незначними рухами тіла, дослідники змогли виявити «приховану» інформацію про серце та частоту дихання людини.

Традиційно виявлення емоцій покладалося на оцінку видимих сигналів, таких як вираз обличчя, мова, жести тіла або рухи очей. Однак ці методи можуть бути ненадійними, оскільки вони не ефективно вловлюють внутрішні емоції людини, і дослідники все частіше звертаються до «невидимих» сигналів, таких як ЕКГ, щоб зрозуміти емоції.

Напевно найпоширенішим останнім часом методом розпізнавання емоцій є використання машинного зору. Емоційне розпізнавання за допомогою використання машинного зору - це дуже активна поточна сфера дослідження комп'ютерного зору, яке передбачає виявлення емоцій обличчя та автоматичну

оцінку настроїв від візуальних даних. Взаємодія людини з машиною є важливою сферою досліджень, де штучно інтелектуальні системи з візуальним сприйняттям прагнуть отримати розуміння поведінки людини.

Розпізнавання емоції - це завдання машин, які намагаються проаналізувати, інтерпретувати та класифікувати людську емоцію шляхом аналізу рис обличчя.

Серед усіх завдань високого рівня, аналіз візуальної емоції є одним з найбільш складних завдань. Розуміння людських емоцій є важливим кроком до сильного штучного інтелекту. З швидким розвитком згорткових нейронних мереж (CNN), глибоке навчання стало новим методом вибору для завдань емоційного аналізу.

У статті [28] описуються які саме нейронні мережі застосовуються для рішення даної задачі. Найпопулярнішими є CNN, CNN-LSTM.

Методи глибокого навчання для розпізнавання емоційного стану обличчя включають CNN (згортає нейронна мережа) та гібридний метод CNN-LSTM (CNN-довга короткочасна пам'ять). CNN може зменшити залежність фізичних моделей та/або інших методів попередньої обробки. CNN здатна на це через навчання безпосередньо з вхідних зображень. Для цього, CNN оснащений трьома шарами: згортковий шар, шар максимального об'єднання та повністю пов'язаний шар. Однак CNN не може відображати тимчасові варіації компонентів обличчя. Цей недолік подолано гібридом.

Метод, CNN-LSTM. LSTM - це особливий тип RNN (рекурентна нейронна мережа), який має ланцюгову структуру, незважаючи на повторювані модулі, забезпечує різну структуру. Сама модель LSTM є простою у налаштуванні з іншими моделями, і підтримує як входи і виходи з фіксованим розміром, так і зі змінним. Важливо відзначити, що багато досліджень розпізнавання емоцій обличчя побудовані на основі методів CNN та CNN-LSTM [29].

Глибоке навчання, що застосовується у визначенні емоцій обличчя має багато переваг, включаючи легку у використанні структуру та високу

продуктивність. Однак великий об'єм набору вхідних даних для навчання та масивні обчислювальні пристрої необхідні для того, щоб метод міг функціонувати.

1.7. Висновок до розділу

З викладеного вище видно, що дослідження психоемоційних станів людини є актуальним напрямком сучасної медицини, психології та антропології. Також, як будь-яка складна сучасна наука, заснована на результатах численних експериментальних досліджень, дослідження психоемоційних станів потребує використання різних методів штучного інтелекту. В тому числі методів аналізу великих даних шляхом класифікації або регресійного аналізу, які будуть досліджені у наступному розділі.

РОЗДІЛ 2

МАТЕМАТИЧНІ МОДЕЛІ ВИКОРИСТОВУВАНІ ДЛЯ ДОСЛІДЖЕНЬ ДАНИХ Х ПСИХОЛОГІЧНИХ ТЕСТІВ

2.1. Деревя рішень

Деревя регресії та класифікації, відомі також під загальною назвою як деревя рішень (Decision Tree), представляють собою структури даних, що дозволяють інтерпретувати шаблони даних з метою їх розпізнавання. Деревя рішень організовані у вигляді ієрархічної структури, що складається з вузлів прийняття рішень щодо оцінки значень певних змінних для прогнозування кінцевого значення [30].

Для навчання дерев рішень використовуються вибірки даних, які мають бути зібрані заздалегідь. З одного боку, такі дані можуть бути згенеровані експертом або накопичені під час реальних експериментів. На основі виключно тільки інтерпретації подібних даних може бути створено багато програм, у тому числі і моделі ШІ. На практиці кожна розглянута при цьому задача може бути представлена за допомогою множини атрибутів, на основі якої дерево рішень прогнозує знайдений атрибут.

Деревя рішень можуть використовуватися як компонент, що відповідає за прийняття рішень під час реалізації персонажів гри. Кожна ситуація моделюється у вигляді множини атрибутів, тому в кожній конкретній ситуації дерево рішень може запропонувати найкращий спосіб дії. Крім того, деревя регресії можуть використовуватися для оцінки сильних сторін будь-якого об'єкта або для прогнозування результату, як позитивного, так і негативного.

Будь-яке дерево рішень виводить прогнозоване значення, отримане в результаті оцінки деяких вхідних атрибутів. Деревя рішень діляться на два різних типи: деревя класифікації та деревя регресії [30]. Цей поділ не залежить від вхідних даних, оскільки деревя обох типів здатні працювати як з безперервними так із дискретними даними. Визначальним фактором, від якого

залежить тип дерева, є вихідне значення. Дерево рішень з безперервними вихідними значеннями називається деревом регресії, а дерева класифікації замість цього виводять конкретні значення. Підсумкові відомості, що визначають відмінності між типами дерев рішень, наведені в табл. 2.1.

Таблиця 2.1

Відмінності між двома типами дерев рішень, що визначаються тим, результат якого типу вони повертають

Тип дерева	Прогнозування	Тип даних
Дерево класифікації	Дискретне	Символи
Дерево регресії	Безперервне	Дійсні числа

Кожне дерево рішень є різновидом деревоподібного графу в прямому сенсі слова. Ця структура даних складається з вузлів, з'єднаних один з одним ребрами (рис. 2.1).

При цьому не допускається, щоб ребра утворювали цикл, оскільки в іншому випадку дерево перетворюється на граф, відмінний від деревоподібного.

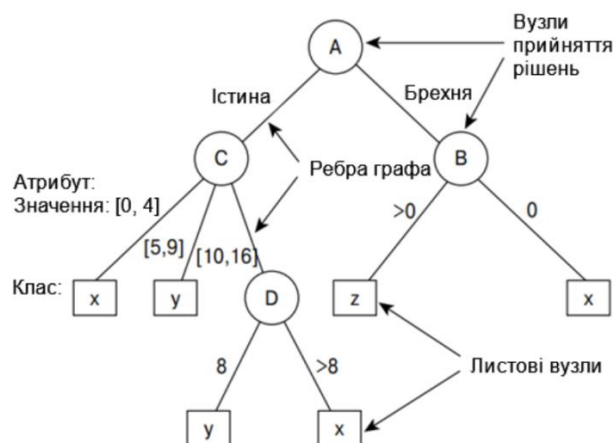


Рис. 2.1. Просте дерево рішень, в якому кореневий вузол показаний зверху, а кожен вузол прийняття рішень позначений колом; листові вузли позначаються прямокутниками

У дереві є один особливий вузол, відомий як кореневий. По суті, цей

вузол є основою дерева, оскільки від кореня можна перейти по дереву до будь-якого іншого вузла. Ще до одного особливого різновиду вузлів належать вузли, що знаходяться в кінці ланцюжка які називаються листовими або термінальними вузлами. Наведений вище опис деревоподібної структури даних є дуже розповсюдженим. Якщо йдеться про використання дерев рішень у штучному інтелекті, то кожне з викладених вище понять набуває особливого сенсу.

Кожен рівень у дереві може розглядатися як одне з рішень; вузол прийняття рішень забезпечує перевірку умови, а кожне ребро позначає один з можливих варіантів. Більш формально можна відзначити, що вузли прийняття рішень містять критерії вибору, а ребра виражають взаємовиключні результати перевірки відповідності цим критеріям.

По суті, при кожній перевірці умови відбувається сортування вибірок даних таким чином, що кожен елемент даних визначається як відповідний тільки одному ребру. Якщо всі вибірки розглядаються як одна множина даних, то критерії прийняття рішень розбивають цю множину на безліч підмножин, що не перетинаються, і постійно зменшуються за розміром як показано на рис. 2.2. Листовий вузол містить невелику унікальну частину вихідної множини даних.

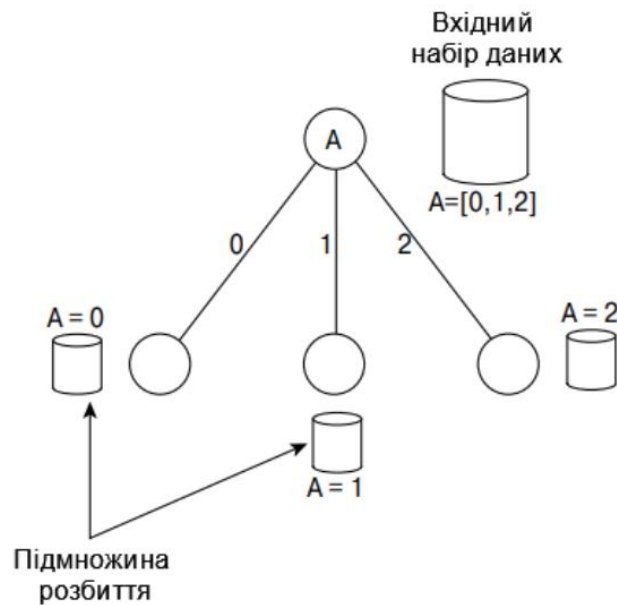


Рис. 2.2. Розбиття вибірки даних на підмножини, що взаємно виключають одна одну, з використання вузла прийняття рішень

Дерева класифікації та регресії є одним з найбільш популярних методів вирішення багатьох практичних завдань, що обумовлено наступними причинами:

- Дерева рішень дозволяють отримувати дуже легко інтерпретовані моделі, що виглядають як набір причинно-наслідкових правил. Інтерпретація полегшується через можливість представити ці правила у вигляді наочної деревоподібної структури.
- Завдяки своїй будові дерева рішень дозволяють працювати зі змінними будь-якого типу без необхідності будь-якої попередньої підготовки цих змінних для введення в модель.
- Досліднику немає необхідності в явному вигляді задавати форму взаємозв'язку між відгуком і предикторами, як це, наприклад, відбувається у випадку зі звичайними регресійними моделями. Це виявляється особливо корисним при роботі з даними великого обсягу, про властивості яких мало що

відомо.

– Дерева рішень автоматично виконують відбір інформативних предикторів і враховують можливі взаємодії між ними. Це, зокрема, робить дерева рішень корисним інструментом розвідувального аналізу даних.

– Дерева рішень можна ефективно застосовувати до даних з пропущеними значеннями, що дуже корисно при вирішенні практичних завдань, де наявність пропущених значень - це, швидше, правило, ніж виняток.

– Дерева рішень однаково добре застосовуються як до кількісних, так і до якісних залежних змінних.

До недоліків цього класу моделей іноді відносять нестабільність і невисоку точність передбачень, що, не завжди підтверджується. Дерева використовують «наївний підхід» в тому сенсі, що вони виходять з припущення про взаємну незалежність ознак. Тому моделі регресійних дерев статистично найбільш працездатні, коли комплекс аналізованих змінних є не надто мультиколінеарним або є регулярна внутрішня множинна альтернатива у вихідній комбінації ознак.

2.1.1. Дерева класифікації

Дерева класифікації - це метод, що дозволяє передбачати приналежність спостережень або об'єктів до того чи іншого класу категорійної залежної змінної залежно від відповідних значень однієї або декількох предикторних змінних [31].

Класифікаційні моделі дерев рекурсивно ділять набір даних на підмножини, що є все більш і більш гомогенними щодо певних ознак. Створюється вирішальне правило класифікації ієрархічного типу та формується асоціативний логічний ключ, що дає можливість виконувати розпізнавання об'єктів з нових вибірок.

Найпростішим випадком дерев рішень є дерева бінарної класифікації, у вузлах яких розгалуження може відбуватись тільки в двох напрямках, тобто здійснюється вибір з двох альтернатив «так» або «ні». Наприклад, створення

такої дихотомічної класифікаційної моделі здійснюється з використанням алгоритму ID3 (Interactive Dichotomizer).

2.1.2. Деревя регресії

Дерево регресії - це тип дерева рішень. Воно використовує обчислення суми квадратів та регресійного аналізу для передбачення значень поля призначення. Прогнози базуються на комбінаціях значень у полях введення [31].

У дереві регресії обчислюється передбачене середнє значення для кожного вузла в дереві. Цей тип дерева створюється, коли поле призначення є безперервним.

Дерева регресії є одним з важливих класів регресійних моделей, що дозволяють здійснити поділ вхідного простору на сегменти з подальшою побудовою для кожного з них власної (локальної) моделі та представити частково задану функцію регресії в інтуїтивно зрозумілій та наочній формі. У такому дереві внутрішні вузли містять правила поділу простору пояснювальних змінних.

Більшість сучасних алгоритмів побудови дерев регресії є жадібними. Такі алгоритми здійснюють побудову дерев зверху вниз шляхом рекурсивного поділу навчальних даних і коротко можуть бути описані наступним чином:

1. Вибір кращого варіанту поділу.
2. Розділення вихідних даних на підмножини.
3. Рекурсивне застосувати даної процедури для кожної з виділених підмножин.

2.1.3. Методи роботи з деревами

2.1.3.1. ID3

В основі алгоритму ID3 [32] лежить циклічне розбиття навчальної вибірки на класи відповідно до змінної, що має найбільшу інформативність. Кожна підмножина спостережень, що виділяється такою змінною, знову

розбивається на класи з використанням наступної змінної з найбільшою класифікуючою здатністю тощо. Розбиття закінчується, коли в підсумковій підмножині опиняються об'єкти лише одного класу. Шляхи руху по отриманому дереву рішень з верхнього рівня на найнижчі визначаються логічними правилами у вигляді ланцюжків кон'юнкцій.

При виборі критерію, що оцінює класифікуючу силу, зазвичай використовують теоретико-інформаційний підхід. Міра інформаційного виграшу IG (Information Gain) для розбиття A визначається як різниця ентропії Шеннона $H(S)$ для вихідного набору даних і суми ентропій всіх фрагментів даних після розбирання $H(S_v)$ з відповідною вагою залежно від розміру кожної частини :

$$IG(S, A) = H(S) - \sum_{v \in V} \frac{S_v}{S} H(S_v) \quad 2.1$$

де позначене опорне значення змінної V , використане для розбиття A . Якщо поділ на класи виконано оптимально, то кожна підмножина S_v буде мати невелику ентропію отже, значення IG буде максимальним.

2.1.3.2. CART

Алгоритм CART (Classification and Regression Tree) [31] рекурсивно ділить вихідний набір даних на підмножини, які стають все більш і більш гомогенними щодо певних ознак, в результаті чого формується деревовидна ієрархічна структура. Поділ здійснюється на основі традиційних логічних правил у вигляді *ЯКЩО (A) ТО (B)*, де A - деяка логічна умова, а B - процедура ділення підмножини на дві частини, для однієї з яких умова A істинна, а для іншої - хибна. Приклади умов: $X_i == F, X_i \leq V, X_i \geq V$ та ін., де X_i - один з предикторів вихідної таблиці, F - вибране значення категорійної змінної, V - спеціально підібране опорне значення (поріг).

На першій ітерації кореневий вузол дерева зв'язується з найбільш

оптимальним умовним твердженням, і вся множина об'єктів ділиться на дві групи. Кожний батьківський вузол може мати до двох дочірніх вузлів (бінарне дерево). Кінцевими вузлами дерева є «листки», що відповідають знайденим рішенням і об'єднують всі розділені на групи об'єкти навчальної вибірки. Загальне правило вибору опорного значення для кожного вузла побудованого дерева можна сформулювати наступним чином: «обрана ознака повинна розбити множину X^* так, щоб отримувані в результаті підмножини X_k^* , $k = 1, 2, \dots, p$, склалися з об'єктів, що належать до одного класу, або були максимально наближені до цього».

Описаний процес відноситься до так званих «жадібних» алгоритмів, які прагнуть, не зважаючи ні на що, побудувати максимально «розгалужене» дерево. Звісно, чим більш розгалужене дерево, тим краще будуть результати його тестування на навчальній вибірці, але вони будуть не настільки успішними на перевірочній вибірці. Тому побудована модель повинна бути ще й оптимальною за розмірами, тобто містити інформацію, що покращує якість розпізнавання, і ігнорувати ту інформацію, яка його не покращує. Для цього зазвичай проводять «обрізання» дерева - відсікання гілок там, де ця процедура не призводить до серйозного зростання помилки.

Неможливо підібрати об'єктивний внутрішній критерій, що призводить до хорошого компромісу між безпомилковістю і компактністю, тому стандартний механізм оптимізації дерев заснований на перехресній перевірці. Для цього навчальна вибірка поділяється, наприклад, на 10 рівних частин: 9 частин використовується для побудови дерева, а решта відіграє роль перевірочної сукупності. Після багаторазового повторення цієї процедури з деякого набору дерев-претендентів, у яких є допустиме відхилення критеріїв якості моделі, обирається дерево, що показало найкращий результат при перехресній перевірці.

2.1.3.3. C4.5

Алгоритм був запропонований Росом Куїнленом як вдосконалена версія алгоритму ID3, в яку додана можливість роботи з пропущеними даними [33].

Нехай задана навчальна множина S , що містить m атрибутів і n прикладів. Для багатьох S визначено k класів C_1, C_2, \dots, C_k . Завдання полягає в побудові ієрархічної класифікаційної моделі у вигляді дерева рішень на основі навчальної множини S .

Побудова дерева рішень здійснюється зверху вниз - від кореневого вузла до листових.

На першому кроці навчання формується «порожнє» дерево, яке складається тільки з кореневого вузла, що містить всю навчальну множину даних. Потрібно розбити кореневий вузол на підмножини з яких будуть сформовані вузли-нащадки. Для цього вибирається один з атрибутів і формуються правила, які розбивають навчальну множину на підмножини, число яких дорівнює кількості p унікальних значень атрибуту.

У результаті розбиття виникає p підмножин і, відповідно, формуються p нащадків кореневого вузла, кожному з яких поставлено у відповідність свою підмножину. Потім ця процедура рекурсивно застосовується до всіх підмножин до тих пір, поки не буде виконана умова зупинки навчання.

Основною проблемою при навчанні дерев рішень є вибір атрибуту, який забезпечить найкраще розбиття в поточному вузлі. Хоча деякі алгоритми навчання дерев рішень допускають використання кожного атрибуту тільки один раз, в цьому випадку це обмеження застосовуватися не буде - кожен атрибут може застосовуватися для розбиття довільне число разів. Нехай до навчальної множини застосовується правило розбиття, в якому використовується атрибут A , що приймає p значень a_1, a_2, \dots, a_p . У результаті буде створено p підмножин S_1, S_2, \dots, S_p , куди будуть розподілені приклади, в яких атрибут A набуває відповідного значення. При цьому виникає питання: чи є розбиття за обраним атрибутом кращим, або вибравши інший атрибут, можливо було б отримати краще розбиття? Для відповіді на це питання

використовуємо інформацію про кількість прикладів усіх класів у навчальній множині і в кожній отриманій підмножині.

Позначимо $N(C_j S)$ число прикладів класу C_j у множині S . Тоді ймовірність класу C_j у цій множині буде:

$$P = \frac{N(C_j S)}{N(S)} \quad (2.2)$$

де $N(S)$ - загальна кількість прикладів у множині S .

Величину

$$Info(S) = - \sum_{i=1}^m \frac{N(S_i)}{N(S)} \log\left(\frac{N(C_j S)}{N(S)}\right) \quad (2.3)$$

в теорії інформації називають ентропією множини S . Вона показує середню кількість інформації, необхідну для визначення класу прикладу з множини S .

Цю ж оцінку, отриману після розбиття множини S по атрибуту A , можна записати у вигляді:

$$Info_A(S) = \sum_{i=1}^k \frac{N(C_j S)}{N(S)} Info(S_i) \quad (2.4)$$

де S_i - i -й вузол, отриманий при розбитті по атрибуту A . Тоді для вибору кращого атрибута гілки можна використовувати наступний критерій:

$$Gain(A) = Info(S) - Info_A(S), \quad (2.5)$$

Потім значення критерію обчислюється для всіх потенційних атрибутів розбиття, і вибирається той атрибут, який максимізує його.

Описана процедура застосовується до підмножин S_i і далі, до тих пір, поки значення критерію не перестануть значимо збільшуватися при нових розбиттях або не буде виконано іншу умову зупинки.

Якщо в процесі побудови дерева буде сформовано «порожній» вузол, куди не потрапило жодного прикладу, то він перетворюється на термінальний елемент, який асоціюється з класом, що найбільш часто зустрічається біля безпосереднього предка вузла.

Критерій приросту інформації заснований на властивості ентропії, що полягає в тому, що вона найбільша, коли всі класи рівноймовірні, тобто вибір класу максимально невизначений, і дорівнює 0, коли всі приклади у вузлі належать одному класу. Таким чином, приріст інформації відображає збільшення класової однорідності результуючих вузлів.

Описана процедура застосовна до дискретних атрибутів. У разі безперервних атрибутів алгоритм працює дещо інакше. Вибирається поріг, з яким порівнюватимуться всі значення. Нехай числовий атрибут X приймає кінцеву кількість значень x_1, x_2, \dots, x_p . Впорядкувавши приклади за зростанням значень атрибута, отримаємо, що будь-яке значення, що лежить між x_i та x_{i+1} ділить всі приклади на дві підмножини. Перша підмножина міститиме значення атрибута x_1, x_2, \dots, x_p , а друга - $x_{i+1}, x_{i+2}, \dots, x_p$.

Тоді як поріг можна вибрати середнє:

$$T_i = \frac{x_i + x_{i+1}}{2} \quad (2.6)$$

Таким чином, завдання знаходження порогу зводиться до розгляду $n-1$ потенційних порогових значень T_1, T_2, \dots, T_{n-1} .

Послідовно застосовуючи формули (2.2), (2.3) і (2.4) до всіх потенційних порогів, вибираємо ту, що дає максимальне значення за критерієм (2.4). Потім це значення порівнюється зі значенням критерію (2.4), розрахованим для

інших атрибутів. Якщо це значення виявиться найбільшим з усіх атрибутів, воно вибирається як поріг для перевірки.

Слід зазначити, що всі числові тести є бінарними, тобто ділять вузол дерева на дві гілки.

2.1.4. Висновок по деревах

Дерева рішень застосовуються як основа надзвичайно ефективного методу обчислення оцінок для невідомих вибірок. Структура даних, що лежить в їх основі, є простою і невеликою, а для проходження по дереву рішень потрібно мало зусиль. Дерева рішень особливо добре підходять для завдань аналізу прихованих закономірностей в даних, тому з їх допомогою усувається проблема масової обробки інформації в реальному часі.

Крім того, для використання дерев рішень потрібен малий обсяг інформації, тому вони займають у пам'яті дуже мало місця. Після того як необхідні знання представлені у вигляді дерева, найчастіше відпадає необхідність зберігати вибірки даних, що були застосовані для навчання дерева.

Нарешті, дерева рішень є відносно гнучкими. Вони можуть використовуватися для обробки безперервних та дискретних змінних прогнозування. Для вирішення завдань регресії або класифікації можуть також використовуватися змінні відгуку, завдяки чому з'являється можливість без особливих складнощів розв'язувати найрізноманітніші завдання.

Однак слід зазначити деякі недоліки. Дерева рішень добре підходять для пакетної обробки наборів даних. Але після того як виникає необхідність проводити їх навчання в оперативному режимі, використовувані алгоритми можуть виявитися досить громіздкими. З урахуванням наведеного вище можна сказати що найбільш простий підхід полягає в тому, щоб інкрементно накопичувати вибірки, а потім використовувати їх під час навчання, що зазвичай призводить до створення дерев рішень, які виявляють таку ж поведінку, як і після навчання в оперативному режимі.

Застосований для навчання дерев рішень алгоритм рекурсивного секціонування за своїм характером є жадібним. Цей алгоритм показує відносно високу продуктивність, але зазвичай не дозволяє досягти оптимальних результатів з точки зору якості та розміру дерева. Відносна кількість сформованих дерев, в яких проявляється цей недолік напрочуд велика, хоча досягнута при цьому якість цілком прийнятна.

Останнім, але не найменш важливим недоліком є те, що в процесі навчання може відбутися перенавчання. Для усунення цього недоліку може використовуватися ще один етап обробки дерева, на якому здійснюється його обрізка, але для цього потрібні додаткові обчислення і додаткові дані, що дозволяють провести перевірку результатів. Але, на жаль, інтегровані алгоритми втрачають таку перевагу, як простота.

2.2. Випадковий ліс

Випадковий ліс - це група дерев рішень, в якій кінцевий/листовий вузол буде або класом більшості для задач класифікації, або середнім для задач регресії (рис.2.3) [34].

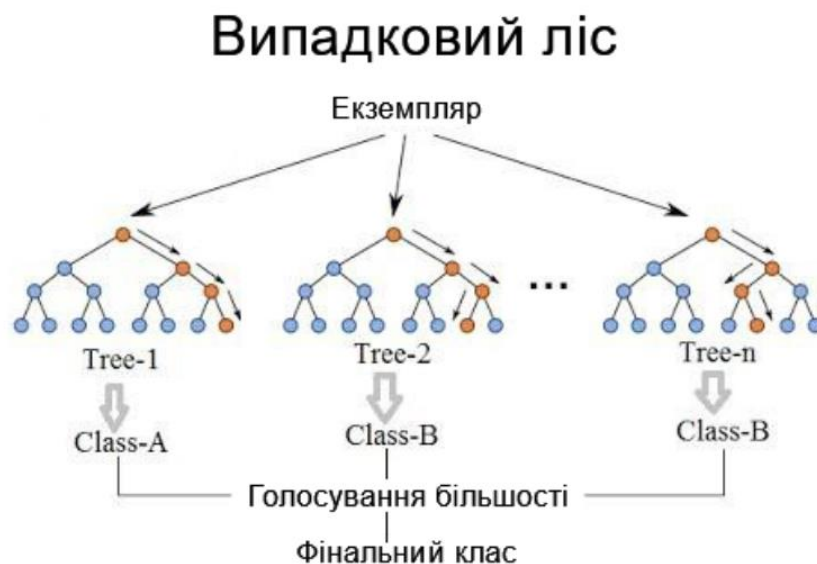


Рис. 2.3. Схема випадкового лісу

В лісі може рости багато дерев за допомогою яких можна знайти різні

класи рішень.

Дерево вирощують за допомогою наступних кроків:

1. Випадкова вибірка рядків з тренувальних даних буде взята для кожного дерева.
2. Із зразка, взятого в п.1, буде використано підмножину ознак для розбиття на кожному дереві.
3. Кожне дерево вирощується в найбільшій мірі, визначеній параметрами, поки не досягне права голосу за клас.

Випадкові ліси належать до сімейства алгоритмів дерев рішень. Дерево рішень являє собою модель класифікації або регресії в ієрархічній структурі. Кожен вузол в дереві являє собою функцію з вхідного простору, кожна гілка рішення і кожен листок в кінці гілки відповідні вихідному значенню (див. рис. 2.4).

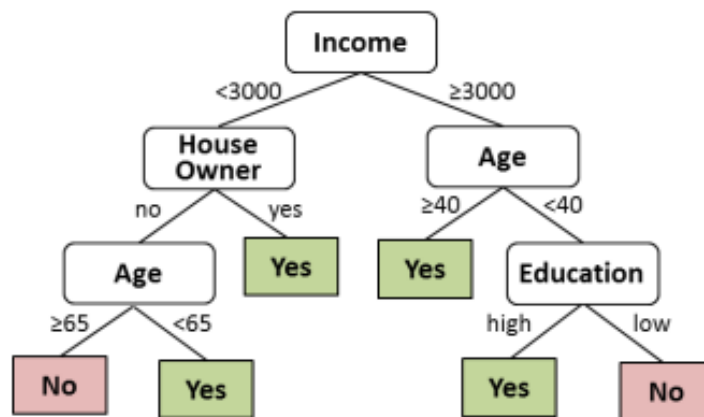


Рис. 2.4. Архітектура дерева рішень

Щоб отримати результат для конкретного вхідного об'єкта, наприклад, людини, яка подає заявку на кредит, процес прийняття рішення починається з кореневого вузла і проходить по дереву, поки не буде досягнуто листа, який містить результат. На кожному вузлі шлях, до конкретного листа, залежить від значення функції для конкретного вхідного об'єкта. Наприклад на рис. 2.4 якщо у людини дохід буде нижче 3000 процес прийняття рішення бути рухатися ліворуч.

Подібно до нейронних мереж, дерево будується за допомогою процесу навчання з використанням тренувальних даних. Процес навчання створює дерево крок за кроком відповідно до важливості вхідних функцій в контексті конкретної програми. Використовуючи всі об'єкти тренувальних даних, спочатку найважливіша властивість визначається шляхом порівняння всіх ознак за допомогою статистичної міри. Відповідно до отриманої величини поділу, дані навчання поділяються. Для кожної отриманої підмножини ідентифікується друга найважливіша ознака і створюється новий поділ. Вибрані властивість можуть бути різними для кожної підмножини. Тепер процес повторюється на кожній отриманій підмножині, поки не будуть знайдені листові вузли у всіх гілках дерева.

Дерево рішень легко створити, воно здатне обробляти різні типи вхідних даних і легко інтерпретується завдяки своєму виду представлення. Однак дерева рішень часто демонструють недостатню надійність в їх застосуванні до нових даних. Однією з причин цього є їх тенденція ідеально підходити до всіх зразків в тренувальних даних. Це призводить до поганої якості програми, якщо дані шумні або містять сторонні дані (рис. 2.5).

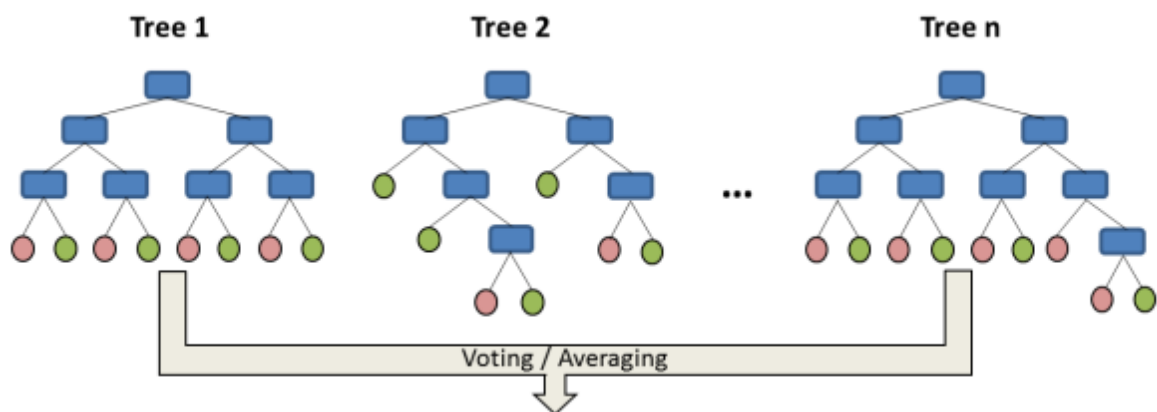


Рис. 2.5. Випадковий ліс як ансамбль дерев

Так званий випадковий ліс є відповіддю на цю проблему. За визначенням, ліс - це група поодиноких дерев. Кожне дерево різне; з їх

сукупності створюється ліс. За цією аналогією, алгоритм випадковий ліс буде різні дерева на основі одного джерела даних. Кожне дерево рішень створюється з іншою, випадково обраною підмножиною тренувальних даних і з випадково обраною підмножиною ознак на кожному вузлі. Таким чином, дерева завжди будуються тільки на основі підмножини даних.

У порівнянні з класичним деревом рішень, описаним вище, кожне дерево в випадковому лісі є слабким деревом, тому що вони не мають доступу до повних даних під час процесу збирання. Але всі разом вони будують групу, яка приймає рішення, де колективний результат може бути створений більшістю голосів у випадку класифікації або усереднення у випадку регресії. Це компенсує потенційні помилки одиничних дерев у лісі, так що модель має меншу ймовірність отримати результати відмінні від реальних значень. Часто на практиці при використанні випадкових лісів використовують розміри дерев від 100 до 500. У своєму практичному застосуванні вони виявляються дуже якісними.

Важлива особливість випадкових лісів полягає в тому, що вони здатні працювати продуктивно в разі відсутності деяких даних. За своїм будівельним принципом не кожне дерево використовує одні і ті ж властивості. Якщо є якесь відсутнє значення для функції під час застосування, зазвичай залишається достатньо дерев, які не використовують цю функцію для отримання точних прогнозів.

З іншого боку, при застосуванні до проблем регресії, випадкові ліси мають обмеження, що вони не можуть перевищувати в діапазоні значень цільової змінної, використовуваної в тренуванні. Таким чином, випадкові ліси погано працюють з даними, що виходять з діапазону тренувальних даних.

2.3. Нейромережі

Нейронна мережа - це обчислювальна модель, яка вільно базується на функціонуванні кори головного мозку людини для відтворення того ж стилю мислення та сприйняття. Нейронні мережі організовані в шари, що

складаються з взаємопов'язаних вузлів, які містять функцію активації, яка у свою чергу обчислює вихід мережі [35].

Нейронні мережі є ще одним засобом машинного навчання, в якому комп'ютер вчиться виконувати завдання, аналізуючи приклади тренувань. Оскільки нейронна мережа базується на принципах функціонування людського мозку, вона буде складатися з тисяч або мільйонів вузлів, які взаємопов'язані між собою. Вузол може бути з'єднаний з декількома вузлами в шарі під ним, з якого він отримує дані, і декількома вузлами над ним, які отримують дані. Кожна вхідна точка даних отримує вагу, потім множиться та додається. Зсув додається, якщо зважена сума дорівнює нулю, а потім передається функції активації.

2.3.1. Архітектура нейронних мереж

Нейронна мережа має 3 основні архітектури:

1. Одношарові мережі.

- Це найпростіша мережа, яка є розширеною версією перцептрона. Вона має додаткові приховані вузли між вхідним шаром і вихідним шаром.

2. Багатошарові мережі.

- Даний тип мережі має один або кілька прихованих шарів, крім вхідних і вихідних. Його роль полягає в втручанні в передачу даних між вхідним і вихідним шаром.

3. Рекурентні мережі.

- Рекурентні нейронні мережі схожі на вищезгадані, але широко прийняті для прогнозування послідовних даних, таких як текст і часові ряди. Найвідомішою рекурентною нейронною мережею є модель «Long - Short Term Memory» (LSTM) [36].

Нейронні мережі являють собою універсальний механізм обчислення на основі розпізнавання образів. Ідея полягає в тому, щоб об'єднати прості одиниці для вирішення складних завдань. Ці одиниці, які також називають нейронами, зазвичай організовані в кілька шарів, які мають певні ролі. Базова

архітектура так званого багатошарового перцептрона показана на рис. 2.6.

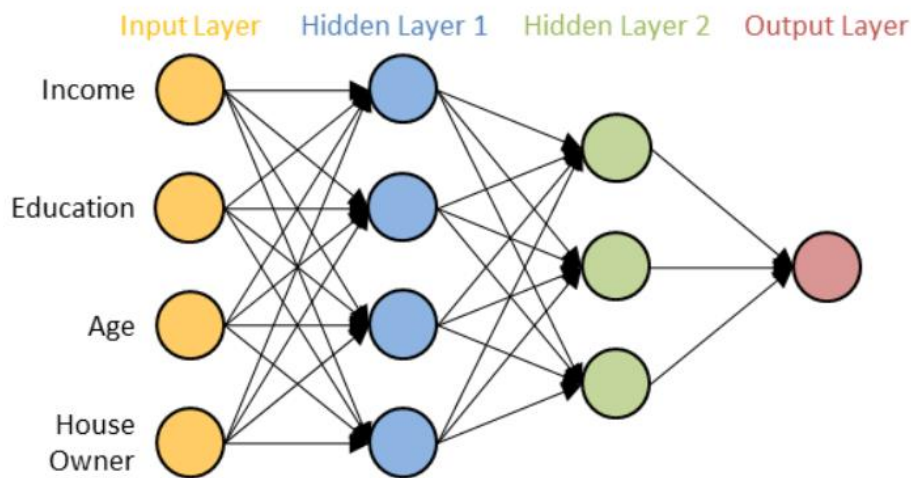


Рис. 2.6. Архітектура нейронної мережі

Нейронна мережа складається з вхідного і вихідного шару, а також, в більшості випадків, одного або кількох прихованих шарів, які мають завдання перетворити входи в те, що може використовувати вихідний шар.

Нейронні мережі можуть обробляти всі види даних, які кодуються в числовій формі. Дані вставляються в мережу через вхідний шар, трансформуються через прихований шар і, нарешті, масштабуються до бажаного результату у вихідному шарі. У випадку оцінки кредитоспроможності користувача банком, вхідні нейрони візьмуть значення для доходів, освіти, віку та володіння будинком. Тоді вихідний нейрон забезпечить ймовірність кредитоспроможності (рис. 2.7).

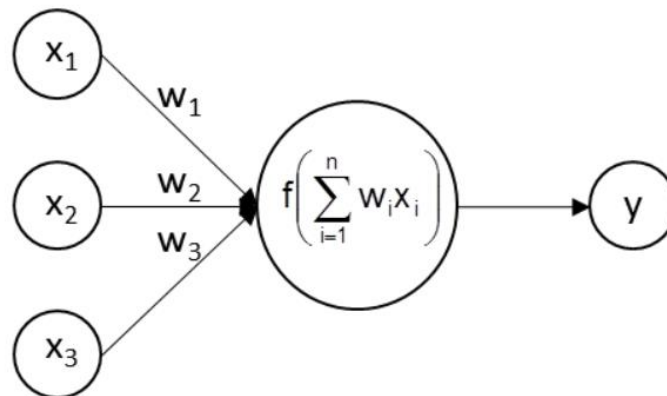


Рис. 2.7. Функціонування нейрона

Якщо буде збільшено прихований або вихідний вузол, ми побачимо те, що показано на рис 2.7. Вузол отримує виходи x_i попередніх вузлів, до яких він підключений. Вони множаться на вагові коефіцієнти W_i і підсумовуються. Потім сума перетворюється з функцією $f(x)$ і передається вузлам наступного шару або виходу в результаті. Після обрання архітектури нейронної мережі, мережа тренується шляхом пошуку вагових коефіцієнтів, які виробляють бажаний вихід. Алгоритми навчання регулюють вагові коефіцієнти зв'язку між нейронами відповідно до мінімізації різниці між дійсними значеннями цільових змінних та обчисленими мережею.

Таким чином, нейронна мережа - це ланцюжок тренованих, чисельних перетворень, які застосовуються до набору вхідних даних і дають певні вихідні дані. За допомогою цієї дуже загальної парадигми ми можемо побудувати майже все: системи класифікації зображень, систему розпізнавання мови, торгівельні системи, системи виявлення шахрайства тощо.

Нейронну мережу можна зробити глибшою, збільшивши кількість прихованих шарів. Чим більше шарів, тим складнішим може бути представлення області застосування. Глибокі мережі мають тисячі або мільйони нейронів і мільйони зв'язків.

З часом були розроблені різні варіанти нейронних мереж для конкретних

областей застосування. Спеціально для обробки зображень були розроблені згорткові нейронні мережі. Рекурентні нейронні мережі призначені для розпізнавання закономірностей в послідовностях даних і довгих короткострокових мереж пам'яті як подальшого розвитку для вивчення довгострокових залежностей.

2.3.2. Згорткові нейронні мережі

Нейронні мережі є підмножиною машинного навчання, і вони є основою алгоритмів глибокого навчання. Вони складаються з шарів вузлів, які містять вхідний шар, один або кілька прихованих шарів і вихідний шар. Кожен вузол з'єднується з іншим і має відповідну вагу та поріг. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі.

Існують різні типи нейронних мереж, які використовуються для різних випадків використання та типів даних. Наприклад, рекурентні нейронні мережі зазвичай використовуються для обробки природної мови та розпізнавання мовлення, тоді як згорткові нейронні мережі (ConvNets або CNN) частіше використовуються для завдань класифікації та комп'ютерного зору. До появи CNN [37] для ідентифікації об'єктів на зображеннях використовувалися ручні, трудомісткі методи вилучення ознак. Однак згорткові нейронні мережі тепер забезпечують більш масштабований підхід до завдань класифікації зображень і розпізнавання об'єктів, використовуючи принципи лінійної алгебри, зокрема, множення матриці, для виявлення шаблонів у зображенні. Тим не менш, вони можуть бути вимогливими до обчислень.

Згорткові нейронні мережі відрізняються від інших нейронних мереж своєю високою продуктивністю при введенні зображення, мови або аудіосигналу. Вони мають три основних типи шарів, а саме:

- Згортковий шар.

- Об'єднуючий шар.
- Повністю пов'язаний шар (fully connected).

Згортковий шар є першим шаром згорткової мережі. У той час як за згортковими шарами можуть слідувати додаткові згорткові шари або шари об'єднання, повністю пов'язаний шар є останнім шаром. З кожним шаром CNN збільшується у своїй складності, ідентифікуючи більші частини зображення. Більш ранні шари зосереджені на простих елементах, таких як кольори та краї. Коли дані зображення просуваються через шари CNN, вона починає розпізнавати більші елементи або форми об'єкта, поки нарешті не ідентифікує передбачуваний об'єкт.

2.3.2.1. Згортковий шар

Згортковий шар є основним будівельним блоком CNN, і саме там відбувається більшість обчислень. Для цього потрібно кілька компонентів, які є вхідними даними, фільтром і картою ознак. Припустимо, що вхідним буде кольорове зображення, яке складається з матриці пікселів у 3D. Це означає, що вхідні дані будуть мати три виміри — висоту, ширину та глибину — які відповідають RGB зображення. У нас також є детектор функцій, також відомий як ядро або фільтр, який буде переміщатися по сприйнятливих полях зображення, перевіряючи, чи є функція. Цей процес відомий як згортка.

Детектор ознак — це двовимірний (2D) масив ваг, який представляє частину зображення. Хоча вони можуть відрізнятися за розміром, розмір фільтра зазвичай є матрицею 3x3; це також визначає розмір рецептивного поля. Потім фільтр застосовується до області зображення, і між вхідними пікселями та фільтром обчислюється точковий добуток. Цей точковий добуток потім подається у вихідний масив. Після цього фільтр зміщується на один крок, повторюючи процес, поки ядро не охопить усе зображення. Остаточний вихід із ряду точкових добутків із входу та фільтра відомий як карта ознак, карта активації або згорткова ознака (рис. 2.8).

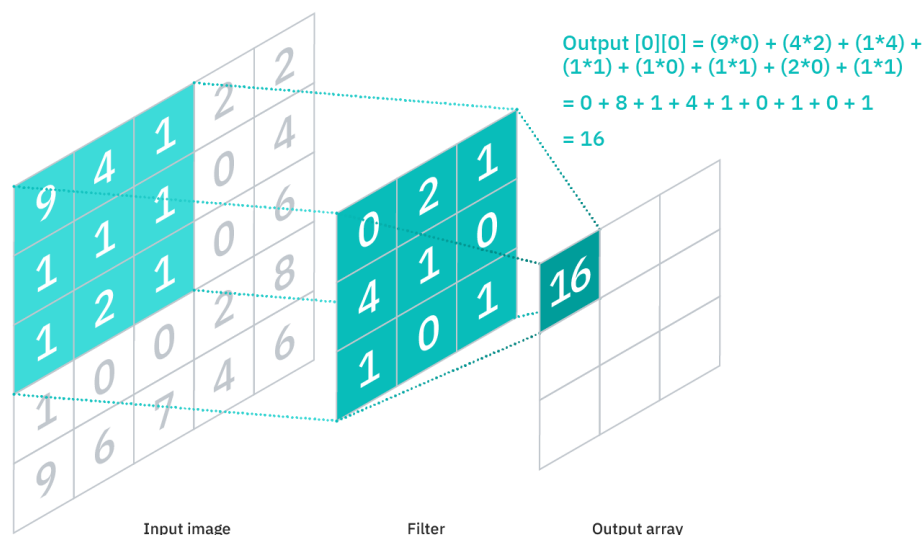


Рис. 2.8. Схема роботи CNN

Як можна побачити на рис. 2.8 вище, кожне вихідне значення на карті ознак не обов'язково має з'єднуватися з кожним значенням пікселя у вхідному зображенні. Йому потрібно лише підключитися до сприймаючого поля, де застосовується фільтр. Оскільки вихідний масив не потребує безпосереднього відображення кожного вхідного значення, згорткові (і об'єднуючі) шари зазвичай називають «частково зв'язаними» шарами. Однак цю характеристику можна також описати як локальну зв'язність.

Варто зауважити, що ваги в детекторі функцій залишаються фіксованими, коли він рухається по зображенню, що також відоме як спільне використання параметрів. Деякі параметри, як-от значення ваги, коригуються під час навчання за допомогою процесу зворотного поширення та градієнтного спуску. Однак є три гіперпараметри, які впливають на розмір виводу, який необхідно встановити перед початком навчання нейронної мережі. До них належать:

1. Кількість фільтрів - впливає на глибину виводу. Наприклад, три різні фільтри дадуть три різні карти об'єктів, створюючи потрібну глибину.

2. Крок – це відстань або кількість пікселів, на які ядро переміщається по вхідній матриці. Хоча значення кроку два або більше зустрічається рідко, більший крок дає менший результат.

3. Нульове заповнення - зазвичай використовується, коли фільтри не підходять до вхідного зображення. Воно обнуляє всі елементи, які виходять за межі вхідної матриці, створюючи більший або однаковий за розміром вихід. Існує три типи заповнення:

- Дійсне заповнення: також відоме як відсутність заповнення. У цьому випадку остання згортка не враховується, якщо розміри не вирівнюються.
- Незмінне заповнення: це заповнення гарантує, що вихідний шар має той самий розмір, що і вхідний шар.
- Повне заповнення: цей тип заповнення збільшує розмір виводу, додаючи нулі до межі вхідних даних.

Після кожної операції згортки CNN застосовує перетворення випрямленого лінійного блоку (Rectified Linear Unit, ReLU) до карти ознак, вносячи нелінійність у модель.

Як згадувалось раніше, наступний шар згортки може слідувати за початковим шаром згортки. Коли це станеться, структура CNN може стати ієрархічною, оскільки пізні рівні можуть бачити пікселі в полях сприйняття попередніх шарів. Як приклад, припустимо, що ми намагаємося визначити, чи містить зображення велосипед. Можна думати про велосипед як про суму частин. Він складається з рами, керма, коліс, педалей тощо. Кожна окрема частина велосипеда складає шаблон нижнього рівня в нейронній мережі, а комбінація його частин представляє шаблон вищого рівня, створюючи ієрархію функцій у CNN (рис. 2.9).

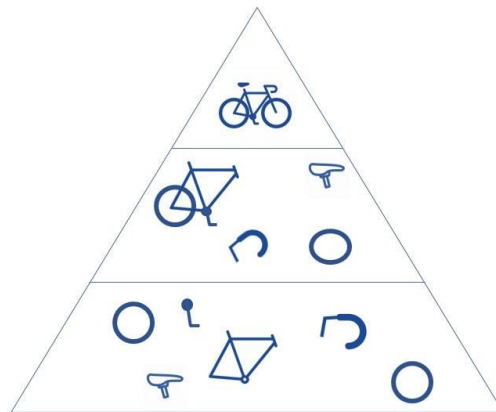


Рис. 2.9. Ієрархія розпізнавання конструкції велосипеда за допомогою CNN

Зрештою, згортковий шар перетворює зображення в числові значення, дозволяючи нейронній мережі інтерпретувати та вилучати відповідні шаблони.

2.3.2.2. Об'єднуючий шар

Об'єднуючі шари, також відомі як «даунсемплінг» (зниження дискретизації), зменшують розмірність, зменшуючи кількість параметрів у вхідних даних. Подібно до згорткового шару, операція об'єднання об'єднує фільтр по всьому входу, але різниця в тому, що цей фільтр не має жодних ваг. Замість цього ядро застосовує функцію агрегації до значень у сприйнятливому полі, заповнюючи вихідний масив. Існує два основних типи об'єднання:

- Максимальне об'єднання: під час переміщення фільтра по входу воно вибирає піксель з максимальним значенням для відправки на вихідний масив. Крім того, цей підхід, як правило, використовується частіше в порівнянні зі середнім об'єднанням.

- Середнє об'єднання: коли фільтр переміщується по вхідному сигналу, він обчислює середнє значення в полі сприйняття для відправлення до вихідного масиву.

Хоча багато інформації втрачається на рівні об'єднання, це також має ряд переваг для CNN, таких як: зменшення складності, підвищення ефективності та обмеження ризику перенавчання.

2.3.2.3. Повністю пов'язаний шар

Назва повністю пов'язаного шару влучно описує себе. Як згадувалося раніше, значення пікселів вхідного зображення не пов'язані безпосередньо з вихідним шаром у частково з'єднаних шарах. Однак у повністю пов'язаному шарі кожен вузол вихідного шару підключається безпосередньо до вузла попереднього шару.

Цей шар виконує завдання класифікації на основі ознак, витягнутих через попередні шари та їх різні фільтри. У той час як рівні згортки та об'єднання, як правило, використовують функції ReLu, рівні повністю пов'язаних шарів зазвичай використовують функцію активації softmax для належної класифікації вхідних даних, створюючи ймовірність від 0 до 1.

2.3.3. Типи згорткових нейронних мереж

Куніхіко Фукусіма та Ян Лекун у своїй роботі заклали основу дослідження згорткових нейронних мереж у 1980 [38] і 1989 [39] відповідно. Більш відомо, Янн Лекун успішно застосував зворотне поширення для навчання нейронних мереж для визначення та розпізнавання шаблонів у серії рукописних поштових індексів. Він продовжив дослідження зі своєю командою протягом 1990-х років, завершившись «LeNet-5» [40], яка застосовувала ті самі принципи попередніх досліджень для розпізнавання документів. Відтоді з'явилася низка варіантів архітектур CNN із введенням нових наборів даних, таких як MNIST і CIFAR-10, а також датасетів, таких як

ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Деякі з цих інших архітектур включають:

- [AlexNet](#)
- [VGGNet](#)
- [GoogLeNet](#)
- [ResNet](#)
- ZFNet

Однак LeNet-5 відомий як класична архітектура CNN.

2.4. Вибір найкращого алгоритму

З нейронними мережами та випадковими лісами є два підходи, які мають потенціал для створення класифікаційних та регресійних моделей високої якості. На практиці обидва підходи широко і успішно використовуються в різних областях застосування. Це ставить питання, яким з методів слід віддати перевагу при запуску конкретного проекту.

Важливим висновком в машинному навчанні є те, що жоден алгоритм не працює найкраще за всіма можливими сценаріями. Таким чином, жоден алгоритм строго не домінує у всіх додатках; продуктивність алгоритмів машинного навчання швидко змінюється в залежності від застосування і розмірності набору даних. Відповідно, гарна практика полягає в тому, щоб порівняти продуктивність різних алгоритмів навчання, щоб знайти найкращий варіант для конкретної проблеми. У деяких випадках також вигідно будувати групи з декількох моделей, створених різними методами поєднання міцності і усунення недоліків.

Однак часто не вистачає часу або грошей для тестування та оптимізації кожного алгоритму. З іншого боку, певні недоліки підходу можуть призвести до уникнення певного алгоритму в конкретному контексті. У цих випадках перед початком проекту необхідно прийняти рішення про те який алгоритм використовувати.

2.4.1. Критерії вибору алгоритму

Продуктивність: Продуктивність в сенсі якості класифікації або регресії є основним показником вибору алгоритму. Алгоритм повинен вміти захоплювати основну функціональність і представляти її в кінцевій моделі. Алгоритми, які можуть захоплювати тільки лінійні зв'язки, як зазвичай завжди непридатні за критерієм продуктивності, якщо в них існують переважно нелінійні зв'язки в області застосування. Як нейронні мережі, так і випадкові ліси мають здатність моделювати лінійні, а також складні нелінійні зв'язки. Завдяки своїй побудові нейронні мережі мають переваги за критерієм продуктивності.

Надійність: При оцінці продуктивності якості і надійності застосування повинні відігравати вирішальну роль. Модель, яка дуже добре працює з даними, які використовуються для її створення не обов'язково однаково добре працюватиме з новими даними. Така ситуація називається «перенавчання». У цьому випадку відповідна модель не забезпечує прийнятну надійність, що відображає її узагальненість.

У сучасному світі доводиться мати справу з великими обсягами даних високої розмірності, що надходять з різних джерел. На відміну від минулого, в якому в основному працювали з попередньо обробленими і агрегованими даними. Як зазвичай людина не в змозі повністю контролювати якість даних. У наслідку чого, майбутні користувачі повинні знати, що дані не є повністю репрезентативними і можуть містити деякі помилки та відхилення.

Дуже схожа ситуація виникає для додатків в динамічному середовищі, в якому в будь-який час і часто випадково можуть відбуватися зміни у відношеннях, такі як кореляції між прибутковістю акцій. Це дуже характерно для економічних додатків. З технічної точки зору, такі зміни призводять до дещо різних виходів з однаковими входами. Модель повинна реагувати на це стабільно і не надто чутливо. Як наслідок, алгоритм, який використовується для оцінки моделі, повинен бути в змозі знайти основні зв'язки і закономірності в даних і не намагатися максимально відповідати даним.

Одним зі способів є зменшення набору можливих моделей, які можуть бути побудовані алгоритмом. У літературі термін іменують «складністю». Простіше кажучи, ми можемо описати це як можливі виходи, які модель може генерувати по відношенню до можливих входів. Лінійна модель має меншу складність, тому що вона не може представляти однаковий вихідний простір порівняно з нелінійною моделлю. Таким чином, лінійна модель, застосована до нелінійного випадку, призводить до низької придатності, тоді як нелінійна модель, застосована до лінійного випадку, містить ризик перенавчання, оскільки вона здатна виробляти вихід, що відхиляється від лінійного світу.

Перспективною стратегією в машинному навчанні є пошук моделі, що має складність, відповідну складності області застосування.

В цьому випадку ми знижуємо ризик перенавчання. Це можна зробити шляхом зменшення набору можливих моделей, які можуть бути створені відповідним алгоритмом. Для цього обидва підходи, нейронні мережі та випадкові ліси, пропонують різні можливості. У нейронних мережах, наприклад, кількість прихованих нейронів і кількість їх шарів впливають на складність моделей або методи регуляризації можуть бути використані при оптимізації ваг під час процесу навчання. У випадкових лісах можна відрегулювати кількість дерев або максимальний розмір або глибину окремих дерев. Таким чином, обидва підходи надають можливості для вирішення проблем складності і перенавчання. Для порівняння, нейронні мережі мають вищу чутливість до входів, що, як правило, призводить до більш високого ризику відхилень у випадку «нетипових» входів.

Зрозумілість: Моделі та їх результати повинні бути «інтерпретованими». Люди повинні розуміти, що роблять моделі, особливо коли вони несуть відповідальність за наслідки їх застосування. У таких галузях, як фінансова сфера, додаткове регулювання змушує компанії використовувати прозорі алгоритми, щоб зрозуміти їх основні ризики. Однією з головних проблем нейронних мереж є те, що навчені моделі важко інтерпретувати. Вони складаються зі ста або мільйонів різних параметрів в залежності від розміру

мережі, притому що вони всі взаємодіють комплексно. Ця проблема чорної скриньки ускладнює використання нейронних мереж в вузьких областях, де надійність передбачень мають велике значення. Випадкові ліси також важко інтерпретувати, тому що вони складаються з багатьох окремих дерев. Навіть якщо одне дерево легко зрозуміти, велика кількість дерев ускладнює розуміння. Однак зовсім нещодавно були розроблені підходи до визначення найбільш репрезентативних дерев. За допомогою їх аналізу випадкові ліси нарешті можуть бути інтерпретованими.

Вартість і витрати часу: З економічної точки зору, витрати і час, відіграють важливу роль та необхідні при створення моделі. У цьому контексті навчання нейронних мереж є дуже трудомістким і ресурсомістким. Крім самого навчального процесу, великий обсяг підготовчих робіт також необхідно виконати щоб привести вхідні дані в необхідну форму. Вони повинні бути в цифровому форматі і нормалізовані. Наприклад щоб знайти кращу модель, потрібно розрахувати і протестувати кілька варіантів. Гіперпараметри можуть бути різними, наприклад, кількість шарів або нейронів на шар або швидкість їх навчання. Різні комбінації цих гіперпараметрів тестуються перед тим як їх використовувати. Для кожної комбінації потрібно обчислювати і оцінювати повну модель. Чим більше гіперпараметрів існує, тим більше комбінацій потрібно перевірити. У поєднанні з необхідним часом навчання моделі, це призводить до значних витрат часу та ресурсів. Випадкові ліси вимагають набагато меншої підготовки. Вони можуть обробляти бінарні функції, категоріальні особливості, а також числові особливості, і немає необхідності в нормалізації ознак. Випадкові ліси швидко тренуються і оптимізуються відповідно до їх гіперпараметрів. Таким чином, ресурсозатрати і час навчання випадкового лісу порівняно низькі.

Крім того, випадковий ліс можна тренувати з відносно невеликою кількістю даних. Нейронні мережі зазвичай потребують більше даних для досягнення того ж рівня точності. З іншого боку, випадкові ліси часто мають невелику продуктивність, коли досягається певна кількість даних, в той час як

нейронні мережі зазвичай отримують вигоду від великих обсягів даних і постійно покращують свою точність.

2.5. Висновок до розділу

Метою цієї роботи було показати, що нейронні мережі, незважаючи на їх нинішню високу популярність, не завжди є кращим вибором методології машинного навчання.

Випадкові ліси також досягають аналогічно гарних результатів продуктивності в практичному застосуванні в багатьох областях, зокрема під час дослідження результатів психологічних експериментів.

РОЗДІЛ 3

АНАЛІЗ РЕЗУЛЬТАТІВ ТЕСТУ НА СТІЙКІСТЬ ДО СТРЕСУ

3.1. Аналіз вхідних даних

Результати тесту на стійкість до стресу були зібрані за допомогою спеціалізованого веб-сервісу, який протягом певного часу опитує піддослідного надаючи йому різні аудіо, візуальні подразники, такі як зображення та записи команд. Реакції піддослідного на здійснений вплив були виражені у вигляді натискань на контрольні зображення, причому був зафіксований час, який був витрачений піддослідним на прийняття того, чи іншого рішення щодо наступного зображення для натискання. Метою дослідження було визначити які аудіовізуальні подразники мають більший вплив на перехід людини в стан стресу.

Відповідно до теорії статистик, стани, які більше впливають на перехід людини в стан стресу, мають більшу інформативність. Перехід людини у стан стресу означав характерні зміни між часом сусідніх натискань. Тобто, набір даних попередньо перетворювався таким чином, щоб проаналізувати та замінити час між сусідніми натисканнями на опис стану піддослідного: спокійний, знервований, пік стресу, післястресовий стан. Після цього перетворення необхідно було класифікувати стани, для чого була збудована згорткова нейронна мережа за допомогою модуля PyTorch та був збудований випадковий ліс дерев класифікації ID3. Було визначено що побудова нейронних мереж згорткового класу з використанням звичайних центральних процесорів персональних комп'ютерів є нераціональною через відносно невеликий обсяг експериментальних даних та вимогливість нейромережі щодо обчислювальних потужностей апаратури.

Випадковий ліс з дерев класифікацій ID3 показав набагато менший час навчання відсутність спеціальних вимог до апаратного забезпечення та відносно непогану точність вимірювань.

3.2. Попередня обробка вхідних даних

У якості вхідних даних спеціалізований веб-сервіс надав набір з 30 000 записів тестів, які містили такі відомості:

- тип зображення;
- яскравість зображення;
- наявність шуму на зображенні;
- тип звукового сигналу;
- гучність;
- наявність сторонніх шумів у звуку;
- колір обраного рішення;
- швидкість відбору (час від початку чергового тесту до натискання на відповідь);
- пульс піддослідного.

Фактично для класифікації спочатку були перетворені стовпчики з часом реакції та пульсом на стовпчик оцінки стресового стану, при чому класифікація відбувалась за параметрами запропонованими фахівцями з психології.

В результаті цього усі стани були розподілені на чотири класи:

- спокійний,
- знервований,
- пік стресу
- післястресовий стан

3.3. Реалізація класифікаторів

Для порівняння результатів класифікації були збудовані та випробувані два типи класифікаторів. За допомогою пакету PyTorch була збудована згорткова нейронна мережа та збудовані випадкові ліси з ID3 дерев розмірами 100, 300 та 500 дерев.

Результати навчання та прогнозування наведені у таблиці 3.1.

Таблиця. 3.1.

Порівняння класифікаторів

Тип класифікатора	Розмір	Час навчання	Точність
Нейронна мережа	1	> 3000 с	98%
Випадковий ліс	100	< 300 с	78%
Випадковий ліс	300	< 1200 с	90%
Випадковий ліс	500	< 2000 с	95%

З наведеної таблиці можна зробити висновок, що класифікатор випадкового лісу є достатньо точним при порівняно меншому часі роботи. Також з наведених даних можна зробити висновок, що кількість дерев у випадковому лісі має вплив на точність результатів аналізу. Помітно, що збільшення кількості дерев від 300 до 500 не суттєво впливає на зростання точності класифікації з чого можна зробити висновок що для цих вхідних даних кількість дерев у лісі має бути близькою до 500 задля досягнення максимального результату.

Задля визначення характеру впливу окремих факторів з набору даних на результати прогнозування були застосовані як класичний метод обчислення інформаційної ентропії, так и метод Бреймана. Результати виміру наведені у наступній таблиці.

Таблиця. 3.1.

Чутливість результатів відносно параметрів класифікації

Фактор	Метод інформаційної ентропії	Метод Бреймана
тип зображення	0,16	0,18
яскравість зображення	0,9	0,8
наявність шуму на зображенні	0,23	0,24
тип звукового сигналу	0,13	0,11
гучність	0,14	0,12
наявність сторонніх шумів у звуку	0,21	0,22
колір обраного рішення	0,04	0,05

З наведених вище результатів можна зробити висновок, що для цих умов експерименту наявність шумів на зображенні та на аудіо доріжці найбільше впливали на стресовий стан піддослідного. Відповідно найменший вплив мав колір обраного рішення.

3.4. Висновок до розділу

В результаті виконання аналізу психологічних досліджень методами штучного інтелекту було створено два типи класифікаторів та експериментальним шляхом встановлено, що для цих умов психологічного експерименту найбільш доцільним є використання випадкового лісу з числом дерев близьким до 500. При цьому можна очікувати 95% точності прогнозування. Також було встановлено, що найбільший вплив на перехід

піддослідного на стан стресу мають фактори шуму на пропонованих зображеннях та аудіо доріжці.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи були досліджені методи штучного інтелекту, які використовуються під час виконання досліджень психоемоційних психоемоційних станів людини.

Було визначено можливість застосування методів математичного моделювання для аналізу результатів непрямих досліджень з метою визначення наявності стресових станів.

Був обґрунтований вибір класифікаторів різної сили для аналізу результатів тесту на стійкість до стресу.

В результаті виконання експерименту було показано, що випадковий ліс з дерев класифікації ID3 кількістю 500 одиниць показує точність 95% за умови відносно невеликого часу на навчання (менше 2000 секунд). Цей результат виявився прийнятнішим за результати лісів меншого розміру та нейронної мережі.

Також було з'ясовано, що з наданих факторів для класифікації найбільший вплив на результати мають фактори шумів на зображеннях та на аудіо доріжці. Однак слід зауважити, що такі результати, як і застосування будь-яких методів штучного інтелекту сильно залежить від характеру вхідних даних та умов експерименту. Тому кількісний та якісний вплив різних факторів на стійкість до стресу у людей з різними типами нервової діяльності потребує додаткових досліджень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wallbott H.G., Scherer K.R. The Measurement of Emotions. Academic Press; Cambridge, MA, USA. 1989. Assessing emotion by questionnaire. PP. 55–82.
2. Becker A., Hagenberg N., Roessner V., Woerner W., Rothenberger A. Evaluation of the self-reported SDQ in a clinical setting: Do self-reports tell us more than ratings by adult informants? *Eur. Child. Adolesc. Psychiatry*. 2004.
3. Isomursu M., Tähti M., Väinämö S., Kuutti K. Experimental evaluation of five methods for collecting emotions in field settings with mobile applications. *Int. J. Hum. Comput. Stud.* 2007.
4. Mahlke S., Minge M., Thüring M. CHI '06 Extended Abstracts on Human Factors in Computing Systems-CHI EA '06. ACM Press; New York, NY, USA. 2006. Measuring multiple components of emotions in interactive contexts. P. 1061.
5. Liapis A., Katsanos C., Sotiropoulos D., Xenos M., Karousos N. Recognizing Emotions in Human Computer Interaction: Studying Stress Using Skin Conductance. Springer; Cham, Switzerland. 2015. PP. 255–262.
6. Camurri A., Lagerlöf I., Volpe G. Recognizing emotion from dance movement: Comparison of spectator recognition and automated techniques. *Int. J. Hum. Comput. Stud.* 2003.
7. Scherer K.R. What are emotions? And how can they be measured? *Soc. Sci. Inf.* 2005.
8. Gonçalves V.P., Giancristofaro G.T., Filho G.P.R., Johnson T., Carvalho V., Pessin G., de Almeida Neris V.P., Ueyama J. Assessing users' emotion at interaction time: a multimodal approach with multiple sensors. *Soft Comput.* 2017.
9. St. Louis E.K., Frey L.C., Britton J.W., Frey L.C., Hopp J.L., Korb P., Koubeissi M.Z., Lievens W.E., Pestana-Knight E.M., St. Louis E.K.

- Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants. American Epilepsy Society; Chicago, IL, USA. 2016.
10. Aminoff M.J. Electroencephalography: General principles and clinical applications. In: Saunders W.B., editor. Aminoff's Electrodiagnosis in Clinical Neurology. Elsevier B.V.; Amsterdam, The Netherlands: 2012. PP. 37–84.
 11. Electroencephalography | Definition, Procedure, & Uses | Britannica.com. URL: <https://www.britannica.com/science/electroencephalography> (дата звернення: 09.10.2021).
 12. Kaur B., Singh D., Roy P.P. EEG Based Emotion Classification Mechanism in BCI; Proceedings of the Procedia Computer Science; Sanur, Bali, Indonesia. 17–19 April 2018.
 13. Pagani C. Violence and Complexity. Open Psychol. J. 2015.
 14. International Neural Network Society. Verband der Elektrotechnik. Institute of Electrical and Electronics Engineers . ANNA '18: Advances in Neural Networks and Applications 2018 September 15–17, 2018, St. St. Konstantin and Elena Resort, Bulgaria. Vde Verlag GmbH; Berlin, Germany. 2018.
 15. Goshvarpour A., Abbasi A., Goshvarpour A. An Emotion Recognition Approach Based on Wavelet Transform and Second-Order Difference Plot of ECG. J. AI Data Min. 2017.
 16. Al Khatib I., Bertozzi D., Poletti F., Benini L., Jantsch A., Bechara M., Khalifeh H., Hajjar M., Nabiev R., Jonsson S. Transactions on High-Performance Embedded Architectures and Compilers I. Springer; Berlin/Heidelberg, Germany: 2007. Hardware/software architecture for real-time ECG monitoring and analysis leveraging MPSoC technology; PP. 239–258. Lecture Notes in Computer Science.
 17. Paithane A.N., Bormane D.S., Dinde S. Human Emotion Recognition using Electrocardiogram Signals. Int. J. Recent Innov. Trends Comput. Commun. 2014.

18. Amri M.F., Rizqyawan M.I., Turnip A. ECG signal processing using offline-wavelet transform method based on ECG-IoT device; Proceedings of the 2016 3rd International Conference on Information Technology, Computer and Electrical Engineering; Semarang, Indonesia. 18–20 October 2016; PP. 25–30.
19. Cai J., Liu G., Hao M. The Research on Emotion Recognition from ECG Signal; Proceedings of the 2009 International Conference on Information Technology and Computer Science; Kiev, Ukraine. 25–26 July 2009; PP. 497–500.
20. Uyarel H., Okmen E., Cobanoğlu N., Karabulut A., Cam N. Effects of anxiety on QT dispersion in healthy young men. *Acta Cardiol.* 2006.
21. O'Connor Peter J., Hill Andrew, Kaya Maria, Martin Brett. The Measurement of Emotional Intelligence: A Critical Review of the Literature and Recommendations for Researchers and Practitioners. *Frontiers in Psychology*, vol. 10. 2019.
22. World Health Organization: URL: <http://www.who.int/en/Section1174/Section1199/Section1567/Section18268103.htm> (дата звернення: 29.10.2021).
23. The State-Trait Anxiety Inventory (STAI). American psychological association: website. URL: <https://www.apa.org/pi/about/publications/caregivers/practice-settings/assessment/tools/trait-state> (дата звернення: 29.10.2021).
24. State-Trait Anger Expression Inventory–2. PAR : website. URL: <https://www.parinc.com/Products/Pkey/429> (дата звернення: 03.11.2021).
25. Spielberger, C.D. and Reheiser, E.C. (2009), Assessment of Emotions: Anxiety, Anger, Depression, and Curiosity. *Applied Psychology: Health and Well-Being*.
26. Kusal, S.; Patil, S.; Kotecha, K.; Aluvalu, R.; Varadarajan, V. AI Based Emotion Detection for Textual Big Data: Techniques and Contribution. *Big Data Cogn. Comput.* 2021.

27. Scientists propose new way to detect emotions using wireless signals. ScienceDaily : website. URL: <https://www.sciencedaily.com/releases/2021/02/210203144447.htm> (дата звернення: 05.11.2021).
28. Jonathan, Jonathan & Lim, Andreas & Paoline, & Kusuma Negara, I Gede Putra & Zahra, Amalia. Facial Emotion Recognition Using Computer Vision. 2018. PP. 46-50.
29. В. С. Ко, "A brief review of facial emotion recognition based on visual information," Sensors, vol. 18, no. 401, 30 January 2018.
30. Decision Trees. ScikitLearn : website. URL: <https://scikit-learn.org/stable/modules/tree.html> (дата звернення: 06.11.2021).
31. Classification And Regression Trees for Machine Learning. Machine Learning Mastery : website. URL: <https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/> (дата звернення: 07.11.2021).
32. Bahety, A. Extension and Evaluation of ID3 – Decision Tree Algorithm. Department of Computer Science University of Maryland, College Park. 2009.
33. Древа прийняття рішень, алгоритми ID3 та C4.5. Andriy Buday: website. URL: <https://andriybuday.com/2010/03/id3-45.html> (дата звернення: 15.11.2021).
34. Understanding Random Forest. Analytics Vidhya: website. URL: <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/> (дата звернення: 15.11.2021).
35. Neural Network. IBM : website. <https://www.ibm.com/cloud/learn/neural-networks> (дата звернення: 17.11.2021).
36. A Beginner's Guide to LSTMs and Recurrent Neural Networks. PathMind : website. URL: <https://wiki.pathmind.com/lstm> (дата звернення: 21.11.2021).

37. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. Towards Data Science : website. URL: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (дата звернення: 25.11.2021).
38. Fukushima, K. Neocognitron: A Self-organizing Neural Network Model. for a Mechanism of Pattern Recognition Unaffected by Shift in Position. Biological Cybernetics 9 by Springer-Verlag. 1980.
39. LeCun, Y. Backpropagation Applied to Handwritten Zip Code Recognition. AT&T Bell Laboratories, Holmdell, USA. 1989.
40. Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11. PP. 2278-2324, Nov. 1998.

Додаток А
Відомість матеріалів кваліфікаційної роботи

		Позначення			Найменування	Кількість	Примітка		
	1								
	2				Документація	76			
	3								
	4	ІТКІ. ДР 15.01.ДА.ПЗ			Пояснювальна записка	1			
	5								
	6				Диск CD-R з презентацією	1			
					ІТКІ.ДП 15.01.ДА.ПЗ				
Зм	Лист	№ докум.	Підпис	Дата					
Розроб.	Шепегуха				Матеріали кваліфікаційної роботи	Літ.	Аркуш	Аркушів	
Керівник	Коротенко						1	1	
Рецензент	Галушко					НТУ «ДП» 8; 126м-20-1			
Н.контр.	Коротенко								
Зав. каф.	Гнатушенко								

Додаток Б

Вихідний код згорткової нейромережі

```
import torch
import torch.nn as nn
from torch.utils.data import DataLoader
import torchvision.transforms as transforms
import torchvision.datasets
from bokeh.plotting import figure
from bokeh.io import show
from bokeh.models import LinearAxis, Range1d
import numpy as np

# Hyperparameters
epochs = 6
class_count = 10
batch_size = 100
learning_rate = 0.001

DATA_PATH = 'C:\\Users\\User\\PycharmProjects\\MNISTData'
MODEL_PATH = 'C:\\Users\\User\\PycharmProjects\\pytorch_models\\'

# transforms to apply to the data
trans = transforms.Compose([transforms.ToTensor(),
transforms.Normalize((0.1307,), (0.3081,))])

# MNIST dataset
training_dataset = torchvision.datasets.MNIST(root=DATA_PATH, train=True,
transform=trans, download=True)

test_dataset = torchvision.datasets.MNIST(root=DATA_PATH, train=False,
transform=trans)

# Data loader
training_loader = DataLoader(dataset=training_dataset, batch_size=batch_size,
shuffle=True)
```



```

test_loader = DataLoader(dataset=test_dataset, batch_size=batch_size,
shuffle=False)

# Convolutional neural network (two convolutional layers)
class ConvolutionalNetwork(nn.Module):
    def __init__(self):
        super(ConvolutionalNetwork, self).__init__()
        self.layer1 = nn.Sequential(
            nn.Conv2d(1, 32, kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.layer2 = nn.Sequential(
            nn.Conv2d(32, 64, kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.drop_out = nn.Dropout()
        self.fc1 = nn.Linear(7 * 7 * 64, 1000)
        self.fc2 = nn.Linear(1000, 10)

    def forward(self, x):
        out = self.layer1(x)
        out = self.layer2(out)
        out = out.reshape(out.size(0), -1)
        out = self.drop_out(out)
        out = self.fc1(out)
        out = self.fc2(out)
        return out

cnn = ConvolutionalNetwork()

# Loss and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(cnn.parameters(), lr=learning_rate)

# Train the model

```

```

total_step = len(training_loader)
list_of_loss = []
list_of_acc = []
for epoch in range(epochs):
    for i, (imgs, labels) in enumerate(training_loader):
        # Run the forward pass
        outputs = cnn(imgs)
        loss = criterion(outputs, labels)
        list_of_loss.append(loss.item())

        # Backprop and perform Adam optimisation
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        # Track the accuracy
        total = labels.size(0)
        _, predicted = torch.max(outputs.data, 1)
        correct = (predicted == labels).sum().item()
        list_of_acc.append(correct / total)

    if (i + 1) % 100 == 0:
        print('Epoch [{} / {}], Step [{} / {}], Loss: {:.4f}, Accuracy:
{: .2f}%'.format(epoch + 1, epochs, i + 1, total_step, loss.item(),
                    (correct / total) * 100))

# Test the model
cnn.eval()
with torch.no_grad():
    correct = 0
    total = 0
    for imgs, labels in test_loader:
        outputs = cnn(imgs)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)

```

```

correct += (predicted == labels).sum().item()

print('Test Accuracy of the model (on the 10000 test images): {}
%.format((correct / total) * 100))

# Save the model and plot
torch.save(cnn.state_dict(), MODEL_PATH + 'conv_net_model.ckpt')

p = figure(y_axis_label='Loss', width=850, y_range=(0, 1), title='PyTorch
ConvNet results')

p.extra_y_ranges = {'Accuracy': Range1d(start=0, end=100)}

p.add_layout(LinearAxis(y_range_name='Accuracy', axis_label='Accuracy (%)'),
'right')

p.line(np.arange(len(list_of_loss)), list_of_loss)

p.line(np.arange(len(list_of_loss)), np.array(list_of_acc) * 100,
y_range_name='Accuracy', color='red')

show(p)

```

Вихідний код випадкового лісу

Classifiers.py

```

from math import log
from pprint import pprint
import json
import copy

DEPTH = 4

class ID3DecisionTree:

    def __init__(self, dataset, features):
        self.root = self.build(dataset, features)

    def entropy(self, dataset):
        classes = [d[-1] for d in dataset]
        classes_set = set(classes)
        h = 0
        for i in classes_set:

```

```

        h -=
        (classes.count(i)/float(len(dataset)))*log((classes.count(i)/float(len(datase
t))), 2)

        return h

def best_gain(self, dataset):
    max_feature = -1
    max_gain = None
    base_entropy = self.entropy(dataset)
    for i in range(len(dataset[0])-1):
        feature_list = [row[i] for row in dataset]
        feature_set = set(feature_list)
        new_entropy = 0
        for value in feature_set:
            sub_dataset = [row[:i]+row[i+1:] for row in dataset]
            new_entropy += self.entropy(sub_dataset)
        gain = base_entropy - new_entropy
        if max_gain is None or gain > max_gain:
            max_gain = gain
            max_feature = i
    return max_feature

def predict(self, dataset):
    classes = [row[-1] for row in dataset]
    classes_set = set(classes)
    max = 0
    prediction = None
    for _class in classes_set:
        if classes.count(_class) > max:
            max = classes.count(_class)
            prediction = _class
    return prediction

def build(self, dataset, features, depth=1):
    classes = [row[-1] for row in dataset]
    classes_set = set(classes)
    if len(classes_set) == 1 or len(dataset[0]) == 1:
        return self.predict(dataset)

    lengths = [len(row) for row in dataset]
    feature_index = self.best_gain(dataset)
    sub_tree = {features[feature_index]: {}}
```

```

        unused_features = copy.deepcopy(features)
        feature = unused_features.pop(feature_index)
        values_list = [row[feature_index] for row in dataset]
        values_set = set(values_list)
        for value in values_set:
            sub_dataset = [row[:feature_index]+row[feature_index+1:] for
row in copy.deepcopy(dataset) if row[feature_index] == value]
            sub_tree[feature][value] = self.build(sub_dataset,
copy.copy(unused_features), depth+1)
        return sub_tree

    def classify(self, data, features, root=None):
        if root is None:
            root = self.root
        if root is None:
            return
        if type(root).__name__ == 'str':
            return root
        key = list(root.keys())[0]
        value = root[key]
        # if type(value).__name__ != 'dict':
        #     return value
        # else:
        index = features.index(key)
        feature = features.pop(index)
        feature_value = data.pop(index)
        if feature_value not in root[feature].keys():
            return None
        return self.classify(data, features,
root[feature][feature_value])

    def __str__(self):
        return json.dumps(self.root, indent=2)

```

test.py

```

import os
import csv
import copy
from classifiers import *

```

```

from pprint import pprint
from random import randint, shuffle
import matplotlib.pyplot as plt

features = ["parents", "has_nurs", "form", "children", "housing", "finance",
"social", "health"]

def read_data(filename, waiting_for=12960):
    dataset = []
    with open(filename, newline='') as f:
        for row in csv.reader(f):
            if len(row) > 0:
                dataset.append(row)
    if len(dataset) < waiting_for:
        print("Dataset reading error")
        print("Found ", len(dataset))
        exit()
    return dataset

def learn_and_test(dataset, k, percent, shuffle=False):
    if shuffle:
        random.shuffle(dataset)
    result = {"accuracy": 0, "sensitivity": {}}
    accuracy_list = []
    checks_list = []
    trees = []
    test_length = randint(percent*5, percent*15)
    learning_length = len(dataset) - test_length
    ds = copy.deepcopy(dataset)[:learning_length]
    test_ds = copy.deepcopy(dataset)[learning_length:]
    fs = copy.copy(features)
    test_fs = copy.copy(features)
    for i in range(k-1):
        if i < (k-2):
            x = randint(int(0.95*learning_length/(k-1)),
int(1.05*learning_length/(k-1)))

```

```

else:
    x = learning_length
    learning_length -= x
    learning_set = ds[:x]
    ds = ds[x:]
    trees.append(ID3DecisionTree(learning_set, fs))
for tree in trees:
    classes = [row[-1] for row in dataset]
    classes_set = set(classes)
    check = {i: {"true": 0, "false": 0, "unknown": 0} for i in
classes_set}
    accuracy = 0
    for data in test_ds:
        unknown = False
        c = tree.classify(copy.copy(data), copy.copy(test_fs))
        if c is None:
            check[data[-1]]["unknown"] += 1
            unknown = True
            max = 0
            for _class in classes_set:
                if max < classes.count(_class):
                    max = classes.count(_class)
                    c = _class
            if data[-1] == c:
                if not unknown:
                    check[data[-1]]["true"] += 1
                accuracy += 1/len(test_ds)
            else:
                if not unknown:
                    check[data[-1]]["false"] += 1
    accuracy_list.append(accuracy)
    checks_list.append(check)
result["accuracy"] = sum(accuracy_list)/len(accuracy_list)
classes = [data[-1] for data in dataset]
classes_set = set(classes)
for _class in classes_set:

```

```

        sensitivity = 0
        n = 0
        for check in checks_list:
            if _class in check and
(check[_class]["true"]+check[_class]["false"]) > 0:
                sensitivity +=
check[_class]["true"]/(check[_class]["true"]+check[_class]["false"])
                n += 1
        if n > 0:
            result["sensitivity"][_class] = sensitivity/n
    return result

def run(dataset, shuffle=False):
    K = [2, 3, 5, 7, 10, 20]
    classes = [data[-1] for data in dataset]
    classes_set = set(classes)
    results = []
    percent = int(0.01*len(dataset))
    for k in K:
        results.append(learn_and_test(dataset, k, percent))
    print(results)
    x = copy.copy(K)
    y = [r["accuracy"] for r in results]
    plt.plot(x, y)
    plt.title("accuracy")
    plt.xlabel("K")
    plt.ylabel("Accuracy")
    if shuffle:
        plt.savefig("accuracy_with_shuffle.png")
    else:
        plt.savefig("accuracy_without_shuffle.png")
    for _class in classes:
        y = [0]*len(x)
        for i in range(len(results)):
            if _class in results[i]["sensitivity"]:
                y[i] = results[i]["sensitivity"][_class]
    plt.plot(x, y)

```



```
plt.title("ID3 sensitivity")
plt.xlabel("K")
plt.ylabel("Sensitivity")
if shuffle:
    plt.savefig("sensitivity_with_shuffle.png")
else:
    plt.savefig("sensitivity_without_shuffle.png")

if __name__ == '__main__':
    dataset = read_data(os.path.join(os.getcwd(), "nursery.data"))
    print("Classify without shuffle")
    run(copy.deepcopy(dataset))
    print("Classify with shuffle")
    run(copy.deepcopy(dataset), True)
```

ВІДГУК

на кваліфікаційну роботу рівня магістра

**«Обґрунтування використання методів штучного інтелекту для обробки
результатів психологічних досліджень»**

студента групи 126м-20-1 Шепетуки Антона Сергійовича

1 Мета даної кваліфікаційної роботи – на основі існуючих наборів (виборках) даних дослідити доцільність і можливість використання технологій штучного інтелекту (ШІ).

2 Обрана тема актуальна тому, що об'єм світового ринку програмного забезпечення, що використовує алгоритми штучного інтелекту, у 2021 році досягне 51,5 млрд доларів, збільшившись на 21,3% у порівнянні з 2020-м. Такі дані в листопаді 2021 року опублікували в дослідницькій компанії Gartner. До розглянутого ринку в Gartner відносять програмні додатки з вбудованими ШІ можливостями, наприклад, програми машинного зору, обробки великих даних, а також програмне забезпечення для ШІ-систем іншого спрямування.

3 Тема кваліфікаційної роботи відповідного рівня безпосередньо пов'язана з об'єктом діяльності магістра спеціальності 126 «Інформаційні системи та технології» галузі знань 12 «Інформаційні технології» – створення інформаційних технологій різного призначення і застосування.

4 Явища і процеси, що досліджуються в даній кваліфікаційній роботі і обрані для моделювання, оцінювання та дослідження – віднесені в освітньо-кваліфікаційній характеристиці магістрів до класу дослідних та евристичних, рішення яких заснована на знаково-понятійних уміннях.

5 Робота складається з трьох розділів. Перший розділ присвячений аналізу теми дослідження та постановці задачі. У другому розділі наведено проектну складову вирішення завдання. Третій розділ присвячено детальному опрацюванню відповідних тренувальних даних з отриманням потрібних результатів. Оригінальність отриманих в роботі наукових результатів і їх наукова новизна полягають у наступному:

- дослідженні відповідних технологій штучного інтелекту;

- виконанні необхідних заходів щодо розробки та налагодження відповідних складових елементів виконання роботи;

- дослідженні результатів виконаних чисельних експериментів.

6 Практичне значення результатів роботи полягає в розробці моделі застосування технологій штучного інтелекту до виконання необхідних маніпулювань з вибірками рейтингових та соціологічних даних.

7 Практичні результати кваліфікаційної роботи отримані із застосуванням відповідних технологій, а також програмних продуктів MS Word і MS PowerPoint на інформаційно-технологічній платформі Windows.

8 Оформлення графічних матеріалів до кваліфікаційної роботи рівня магістр виконано на сучасному рівні і відповідає вимогам, що пред'являються до рівня виконання робіт даної кваліфікації.

9 Ступінь самостійності виконання кваліфікаційної роботи достатньо висока.

10 Деякі дискусійні положення та недоліки, які мають місце в роботі:

а) недостатньо чітко описано особливості побудови моделей щодо вирішення поставленої задачі;

б) досить стисло описано компоненти процесу отримання кінцевих результатів.

Незважаючи на вищевказані зауваження, кваліфікаційна робота в цілому заслуговує оцінки «добре» та присвоєння здобувачу відповідної кваліфікації.

Керівник кваліфікаційної роботи,
проф. кафедри ІТКІ, д.т.н.

Г.М. Коротенко

РЕЦЕНЗІЯ

на кваліфікаційну роботу рівня магістра
«Обґрунтування використання методів штучного інтелекту для
обробки результатів психологічних досліджень»
студента групи 126м-20-1 Шепетухи Антона Сергійовича

Розглянута робота присвячена застосуванню елементів штучного інтелекту до виконання практичних завдань обробки стандартних датасетів. Слід мати на увазі, що згідно Концепції розвитку штучного інтелекту в Україні, пріоритетними напрямками у цієї галузі є створення сприятливих умов щодо поширення та підвищення якості наукових досліджень у галузі застосувань широкого кола методів штучного інтелекту з метою виведення України на провідні позиції у світовому науковому середовищі у галузі штучного інтелекту.

Завдання і зміст кваліфікаційної роботи відповідає головній цілі - перевірці знань і ступеня підготовленості студента за фахом 126 «Інформаційні системи та технології» галузі знань 12 «Інформаційні технології».

Зміст пояснювальної записки кваліфікаційної роботи відповідає необхідним критеріям та затвердженій темі.

Актуальність обраної теми обумовлена тим, що дослідження ефективності застосування інформаційних технологій і відповідних комп'ютерних засобів продовжують розвиватися на базі розширення застосування всього їхнього спектру.

Повнота і глибина вирішення задач, поставлених в завданні на кваліфікаційну роботу, є достатньою.

Оформлення пояснювальної записки кваліфікаційної роботи виконано в повній відповідності з діючими стандартами і нормативними вимогами.

Список літератури, наведений в роботі, налічує біля 40-ка джерел, що свідчить про вміння автора працювати з літературою та іншими інформаційними джерелами та матеріалами.

Наукова новизна результатів дипломної роботи визначається тим, що вперше проведено дослідження елементів застосування технологій штучного інтелекту до обробки тренувальних датасетів у галузі використання набору даних результатів тесту на стійкість до стресу за допомогою випадкового лісу з ID3 дерев та згорткової нейронної мережі.

Практичне значення результатів роботи полягає в обґрунтуванні добіру методу штучного інтелекту для підвищення швидкодії обробки результатів психологічних експериментів без суттєвого погіршення точності обробки.

До числа загальних зауважень і недоліків роботи слід віднести:

1) відсутність порівняння результатів дослідження з іншими відомими роботами;

2) недостатнє розкриття компоненти роботи, пов'язаної з обґрунтуванням добіру методів штучного інтелекту для підвищення швидкодії обробки результатів психологічних експериментів без суттєвого погіршення точності обробки.

Однак, зазначені зауваження не здійснюють істотного впливу на підсумкові результати кваліфікаційної роботи і не знижують її безумовну практичну та наукову цінність.

Таким чином, слід зробити висновок, що кваліфікаційна робота в цілому заслуговує оцінки «_____», а її виконавець присвоєння відповідної кваліфікації.

Рецензент, **доцент** кафедри безпеки
інформації та телекомунікацій, к.т.н.

О.М. Галушко