ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «МЕТОДИ АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ СЕРВІСІВ У МОБІЛЬНИХ МЕРЕЖАХ НА ОСНОВІ МАШИННОГО НАВЧАННЯ»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Сергій ЗАКРЕВСЬКИЙ*

Виконав:
здобувач вищої освіти
група ІСД-41 Сергій ЗАКРЕВСЬКИЙ

Керівник:
науковий ступінь, вчене звання Оксана ТКАЛЕНКО
к.т.н., доцент

Рецензент:
науковий ступінь, вчене звання _____
Ім’я, ПРІЗВИЩЕ

Київ – 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Бакалавр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІСТ

_____Каміла СТОРЧАК
«____» _____20___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Закревському Сергію Миколайовичу
(прізвище, ім'я, по батькові здобувача)

1.Тема кваліфікаційної роботи: «Методи автоматичної класифікації сервісів у мобільних мережах на основі машинного навчання»

керівник кваліфікаційної роботи Оксана ТКАЛЕНКО, к.т.н., доцент
(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «20» лютого 2026 року № 51.

2. Строк подання кваліфікаційної роботи: 22 травня 2026 року.

3. Вихідні дані до кваліфікаційної роботи: бібліотеки NumPy, Pandas, Matplotlib;
QoS-параметри;
генератор випадкових чисел – random_state=42;
метрика accuracy_score;
науково-технічна література з питань, пов'язаних з ML-технологіями.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1) Аналіз архітектурних та сервісних особливостей мобільних мереж LTE/5G.

2) Методи машинного навчання для класифікації типів сервісів у мобільних мережах.
3) Розробка та дослідження моделі класифікації мобільних сервісів на основі машинного навчання.

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання: 23 лютого 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	23.02 – 27.02.26	
2	Характеристика основних класів сервісів у мережах LTE/5G	02.03 – 09.03.26	
3	Дослідження підходів у машинному навчанні	10.03 – 24.03.26	
4	Аналіз типу даних для обробки	25.03 – 07.04.26	
5	Дослідження алгоритмів KNN, Naive Bayes, Decission Tree	08.04 – 23.04.26	
6	Дослідження алгоритмів Logistic Regression, Random Forest, SVM	24.04 – 04.05.26	
7	Оформлення роботи: вступ, висновки, реферат	05.05 – 15.05.26	
8	Розробка демонстраційних матеріалів	18.05 – 21.05.26	

Здобувач вищої освіти _____ Сергій ЗАКРЕВСЬКИЙ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Оксана ТКАЛЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 65 стор., 25 рис., 8 табл., 28 джерел.

Мета роботи - дослідження методів автоматичної класифікації типів сервісів у мобільних мережах на основі алгоритмів машинного навчання для підвищення ефективності управління якістю обслуговування.

Об'єкт дослідження – процес передавання та обслуговування типів сервісів у мобільних мережах LTE/5G з урахуванням параметрів якості обслуговування (QoS).

Предмет дослідження – алгоритми машинного навчання для автоматичної класифікації типів сервісів у мобільних мережах LTE/5G на основі параметрів якості обслуговування (QoS).

Короткий зміст роботи: Досліджені архітектурні та сервісні особливості мобільних мереж LTE/5G, принципи та підходи машинного навчання. Здійснено аналіз типів даних для обробки, обгрунтовано вибір програмних інструментів та алгоритмів, які використовуються для побудови моделі машинного навчання. Виконано розробку та дослідження моделі класифікації мобільних сервісів на основі Supervised Learning. Виконано навчання та оцінку ML-моделі. Проведено порівняльний аналіз продуктивності різних алгоритмів з визначенням найбільш доцільного методу для практичного застосування.

КЛЮЧОВІ СЛОВА: МАШИННЕ НАВЧАННЯ, ТИП СЕРВІСУ, МОВА ПРОГРАМУВАННЯ PYTHON, ВІЗУАЛІЗАЦІЯ, ТЕХНОЛОГІЯ, АЛГОРИТМ, СОКЕТ, ПРОТОКОЛ, ІНТЕРАКТИВНЕ СЕРЕДОВИЩЕ, ІНФОРМАЦІЙНА СИСТЕМА, МОДЕЛЬ.

ЗМІСТ

ВСТУП

Актуальність теми. Стрімкий розвиток мобільних мереж зв’язку стандартів LTE та 5G супроводжується значним зростанням обсягів переданого трафіку, різноманітністю типів сервісів і підвищенням вимог до якості обслуговування користувачів. Сучасні мобільні мережі повинні одночасно забезпечувати підтримку різних сервісів, таких як передавання мультимедійного контенту, інтерактивні сервіси з низькою затримкою, масове підключення пристроїв Інтернету речей, а також критично важливі застосування з жорсткими вимогами до надійності та затримки. За таких умов ефективне управління мережевими ресурсами неможливе без точного та оперативного визначення типів сервісів, що передаються у мережі.

Традиційні методи класифікації трафіку, які грунтуються на аналізі заголовків пакетів або портів, втрачають свою ефективність через широке застосування механізмів шифрування та динамічної зміни мережних параметрів. Це зумовлює необхідність розробки нових підходів до автоматичної класифікації сервісів, здатних працювати в умовах обмеженої доступності інформації про вміст трафіку. Перспективним напрямом розв’язання цієї задачі є застосування методів машинного навчання, які дозволяють здійснювати інтелектуальний аналіз параметрів якості обслуговування (QoS) та виявляти приховані закономірності у даних.

Використання алгоритмів машинного навчання для автоматичної класифікації сервісів у мобільних мережах зможе відкрити можливості для підвищення точності розпізнавання типів сервісів, оптимізації розподілу радіоресурсів, забезпечення гарантованого рівня якості обслуговування, зменшення затримок і втрат пакетів, а також підвищення загальної пропускної здатності мережі. Крім того, такі підходи є важливими складовими сучасних концепцій інтелектуального управління мережею та мереж, що самоналаштовуються (SON).

У зв’язку з цим розробка та дослідження методів автоматичної класифікації сервісів у мобільних мережах на основі машинного навчання є актуальним

науково-практичним завданням, вирішення якого сприятиме підвищенню ефективності функціонування сучасних телекомунікаційних систем і відповідатиме сучасним тенденціям розвитку мобільного зв’язку.

Метою роботи є дослідження методів автоматичної класифікації типів сервісів у мобільних мережах на основі алгоритмів машинного навчання для підвищення ефективності управління якістю обслуговування.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- дослідити сервісні особливості мобільних мереж LTE/5G, особливості машинного навчання;
- обгрунтувати вибір інструментів для аналізу та обробки даних, роботи з технологіями ML;
- розробити та дослідити ML-модель класифікації мобільних сервісів на основі Supervised Learning;
- виконати оцінку якості роботи моделі класифікації;
- виконати порівняльний аналіз ефективності різних алгоритмів машинного навчання з визначенням найбільш доцільного методу для практичного застосування.

Об'єкт дослідження – процес передавання та обслуговування типів сервісів у мобільних мережах LTE/5G з урахуванням параметрів якості обслуговування (QoS).

Предметом дослідження є алгоритми машинного навчання для автоматичної класифікації типів сервісів у мобільних мережах LTE/5G на основі параметрів якості обслуговування (QoS).

Методи дослідження: аналіз літературних джерел і огляд існуючих рішень; моделювання; методи математичної статистики та комп'ютерного моделювання.

Наукова новизна отриманих результатів: запропоновано підхід до автоматичної класифікації сервісів у мобільних мережах LTE/5G на основі QoS-параметрів із використанням алгоритмів машинного навчання та виконано їх порівняльний аналіз.

Практична значущість отриманих результатів полягає у можливості застосування досліджених алгоритмів машинного навчання у системах моніторингу та керування мобільними мережами LTE/5G для: автоматичної ідентифікації типів сервісів у реальному часі; адаптивного розподілу мережних ресурсів в залежності від класу сервісу; підвищення якості обслуговування користувачів; зменшення затримок та втрат пакетів для критично важливих сервісів; підтримки прийняття рішень операторами зв’язку.

Апробація результатів та публікації:

Закревський С. М. «Застосування алгоритмів машинного навчання для класифікації даних Інтернету речей». Тези доповіді на VII Міжнародній науково-технічній конференції «Сучасний стан та перспективи розвитку IoT». – Київ, 20 квітня 2026 року.

1 АНАЛІЗ АРХІТЕКТУРНИХ ТА СЕРВІСНИХ ОСОБЛИВОСТЕЙ МОБІЛЬНИХ МЕРЕЖ LTE/5G

1.1 Фактори формування та розвитку мереж п’ятого покоління

Технологія п’ятого покоління мобільного зв’язку (5G) орієнтована на суттєве підвищення пропускної здатності мережі та зменшення затримок передавання даних у порівнянні з попереднім поколінням 4G, яке на сьогодні має глобальне поширення та забезпечує покриття навіть у віддалених регіонах. Основною технічною відмінністю 5G є здатність підтримувати значно більші швидкості обміну інформацією та забезпечувати більш оперативну реакцію мережної інфраструктури на запити користувачів.

Розробкою технічних специфікацій для мобільних мереж займається міжнародний консорціум 3GPP, який визначив радіоінтерфейс п’ятого покоління під назвою New Radio (NR). Дана специфікація передбачає функціонування мережі у двох частотних діапазонах: FR1 (600–6000 МГц), що охоплює традиційні субгігагерцові та середні частоти, та FR2 (24–100 ГГц), який відповідає міліметровому діапазону хвиль [4]. Використання цих спектральних смуг дозволяє реалізувати як широке покриття, так і надвисоку швидкість передавання даних у зонах щільної забудови.

Якщо розглядати максимальні теоретичні показники швидкості, то 5G здатна забезпечувати передавання даних до 20 Гбіт/с, що приблизно у двадцять разів перевищує граничні можливості мереж четвертого покоління, для яких характерним є показник близько 1 Гбіт/с. Проте зазначені значення є піковими та досягаються переважно у лабораторних умовах із застосуванням спеціалізованого обладнання.

У реальних умовах експлуатації середня швидкість доступу до мережі 5G для кінцевих користувачів становить орієнтовно 50–100 Мбіт/с, що залежить від щільності базових станцій, навантаження на мережу, частотного діапазону, який використовується, та характеристик абонентського обладнання. Незважаючи на

те, що фактичні швидкісні показники можуть відрізнятися від теоретичних максимумів, впровадження 5G забезпечує значне підвищення ефективності використання радіоресурсу, зменшення затримок та розширення можливостей для реалізації нових сервісів.

Розвиток мобільних мереж п’ятого покоління зумовлений сукупністю технологічних, економічних та соціальних чинників, які формують нові вимоги до телекомунікаційної інфраструктури. Перехід від мереж четвертого покоління до 5G є не лише еволюційним етапом збільшення швидкості передавання даних, а й відповіддю на якісно нові потреби цифрового суспільства.

До факторів, що зумовлюють розвиток технології 5G, слід віднести:

- зростання обсягів мобільного трафіку;
- масове впровадження IoT-пристроїв;
- потреба у низьких затримках;
- розвиток нових сервісних моделей;
- обмеження можливостей 4G;
- розвиток технологій радіодоступу.

Досліджуючи зростання обсягів мобільного трафіку, слід зазначити що одним із ключових факторів є стрімке збільшення обсягу передаваних даних. Поширення потокового відео високої роздільної здатності (HD, 4K, 8K), хмарних сервісів, соціальних платформ та мобільних застосунків суттєво підвищує навантаження на мережну інфраструктуру. Технологія 5G забезпечує більшу пропускну здатність каналів; ефективніше використання спектру; підтримку більшої кількості одночасних підключень.

Розвиток концепції Інтернету речей (IoT) призвів до необхідності обслуговування мільйонів підключених пристроїв у межах однієї території. Датчики, смарт-лічильники, транспортні системи, промислові контролери потребують стабільного та енергоефективного з’єднання. У специфікаціях 5G передбачено режим mMTC (massive Machine-Type Communication), який дозволяє підтримувати надвелику щільність підключень із мінімальним енергоспоживанням.

Мінімальну затримку передавання даних потребують, як приклад, автономний транспорт, дистанційна хірургія, доповнена та віртуальна реальність. На відміну від LTE, де типова затримка становить 10–20 мс, у 5G вона може зменшуватися до 1–5 мс у режимі URLLC (Ultra-Reliable Low Latency Communication). Це відкриває можливість реалізації критично важливих сервісів реального часу.

5G впроваджує концепцію network slicing - створення віртуальних мереж з різними параметрами якості обслуговування (QoS) в межах однієї фізичної інфраструктури. Це дозволяє одночасно обслуговувати різні типи трафіку:

- широкосмуговий мобільний доступ (eMBB);
- надійні сервіси з низькою затримкою (URLLC);
- масові IoT-з'єднання (mMTC).

Таким чином, мережа стає адаптивною та орієнтованою на сервіс.

Хоча технологія LTE забезпечує високі швидкості передавання даних, вона має обмеження щодо максимальної кількості підключених пристроїв; ефективності використання спектру; підтримки наднизьких затримок; гнучкості управління трафіком. Розвиток цифрової економіки потребує мережної архітектури нового покоління, здатної масштабуватися та адаптуватися до змінних умов навантаження.

До технологій, які значно підвищують спектральну ефективність та пропускну здатність мережі, слід віднести massive MIMO; beamforming; використання міліметрового діапазону частот (FR2); удосконалені методи кодування та модуляції.

Отже, розвиток технології 5G обумовлений комплексом факторів, серед яких домінують зростання обсягів трафіку, масове підключення IoT-пристроїв, потреба у наднизьких затримках та впровадження нових сервісних моделей. Сукупність зазначених чинників формує передумови для переходу до інтелектуальних мобільних мереж, здатних забезпечувати гнучке управління ресурсами та адаптацію до різнорідного трафіку.

Компанія Ericsson, яка спеціалізується на телекомунікаційному обладнанні, мережних рішеннях для мобільного зв'язку (2G–5G), програмному забезпеченні

для операторів, розвитку стандартів мобільних мереж, публікує аналітичний звіт, Ericsson Mobility Report [1], який використовується у всьому світі як авторитетне джерело статистики щодо розвитку мобільних мереж. По даним Ericsson за останні два роки споживання мобільного Інтернету у світі зросло вдвічі. 5G встановлює рекорди за темпами впровадження у світі й за прогнозами покриє 60% населення до кінця 2026 року. Мобільні підписки за регіонами та технологіями приведені на рис. 1.1.

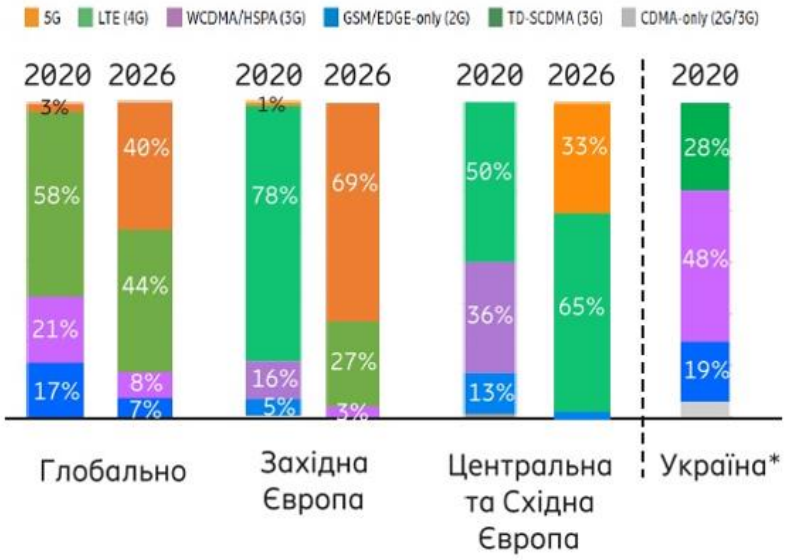


Рис. 1.1 Мобільні підписки за регіонами та технологіями

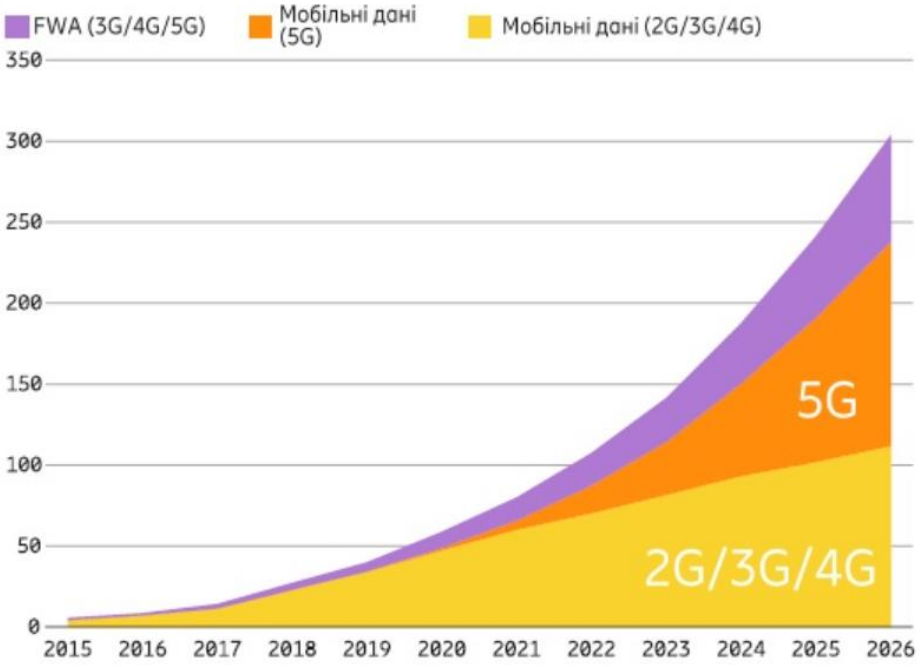


Рис. 1.2 Глобальний трафік мобільних даних (ЕБ на місяць)

1.2 Інтеграція LTE-5G

Інтеграція LTE та 5G є ключовим етапом еволюції мобільних мереж, що дозволяє поступово переходити до нового стандарту, використовуючи існуючу інфраструктуру 4G.

Основні методи інтеграції:

- EN-DC (E-UTRA-NR Dual Connectivity) - найпоширеніший спосіб інтеграції, де пристрій одночасно підключається до мереж LTE (як опорної) та 5G NR (для передавання даних);
- Dynamic Spectrum Sharing (DSS) - технологія, що дозволяє операторам використовувати одні й ті самі частотні смуги для 4G та 5G одночасно, автоматично розподіляючи ресурси в залежності від попиту;
- Non-Standalone (NSA) архітектура - перший етап впровадження 5G, де мережа 5G покладається на існуюче ядро 4G (EPC) для керування з'єднанням;
- конверговані ядра - використання єдиного хмарного ядра, яке підтримує як 4G, так і 5G доступ, що спрощує управління мережею та знижує витрати.

Перевагами інтегрованого підходу є: плавне покриття, збільшення швидкості, економія енергії. Пристрої автоматично перемикаються на LTE у зонах, де сигнал 5G слабкий або відсутній. Агрегація частот LTE разом із 5G дозволяє досягати значно вищої пропускної здатності. Смартфони можуть використовувати енергоефективний LTE для фонових завдань і переходити на 5G лише за потреби у високій швидкості [5].

Стек протоколів LTE представлено на рис. 1.3. Стек протоколів LTE (E-UTRAN) поділяється на площину користувача (дані) та площину управління (сигналізація), забезпечуючи високу швидкість та низьку затримку. Основними рівнями є PDCP, RLC, MAC та PHY, які обробляють стиснення, шифрування, фрагментацію та передавання даних. Керування з'єднанням здійснюється через RRC. Стек протоколів LTE - це багаторівнева система мережних протоколів, що забезпечує організацію радіоінтерфейсу, керування з’єднанням, передавання даних користувачів та підтримку мобільності абонентів у мережах четвертого

покоління мобільного зв’язку. Особливостями стеку LTE є: повністю пакетна архітектура (All-IP), підтримка високих швидкостей передавання даних, розділення площин User Plane, Control Plane, механізми безпеки (шифрування та аутентифікація).

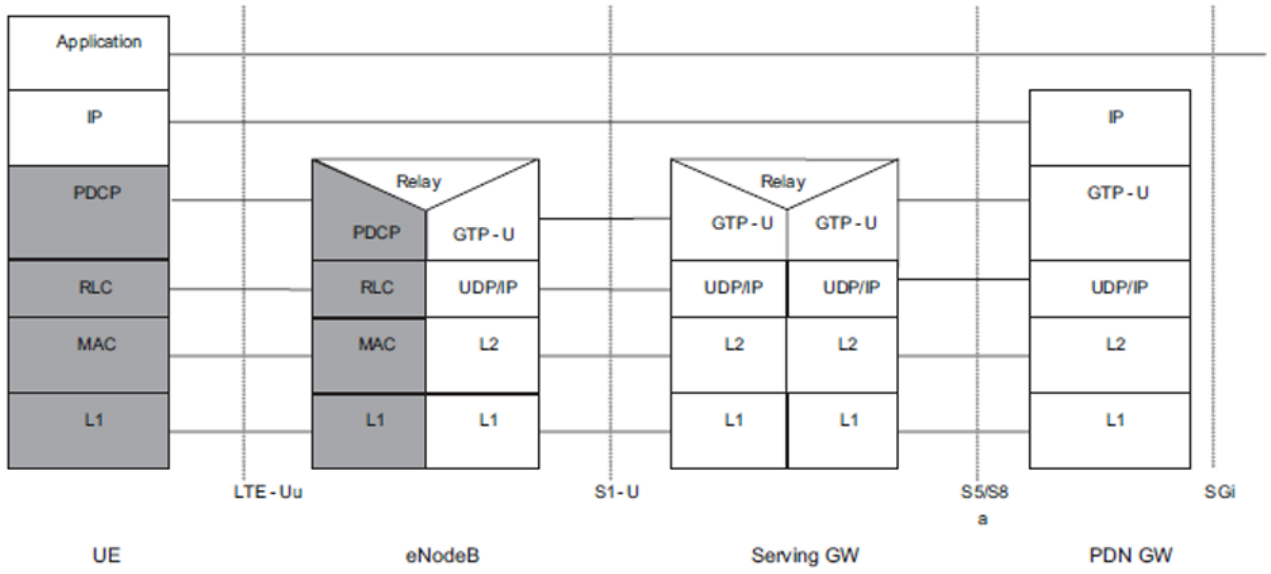


Рис. 1.3 Стек протоколів LTE

Рівень Physical Layer (PHY) відповідає за модуляцію та демодуляцію сигналу (OFDM/SC-FDMA), кодування каналів, передавання радіосигналу. Рівень MAC (Medium Access Control) відповідає за розподіл радіоресурсів, планування передавання, HARQ (повторне передавання помилкових пакетів). RLC (Radio Link Control) забезпечує сегментацію та збирання пакетів, керування помилками (ARQ), буферизацію. За шифрування та захист даних відповідає рівень PDCP (Packet Data Convergence Protocol), який також забезпечує стиснення заголовків (ROHC) та забезпечення безперервності під час handover. RRC (Radio Resource Control) використовується лише в Control Plane і забезпечує встановлення та розрив з’єднання, керування мобільністю, налаштування радіопараметрів.

LTE використовує архітектуру EPC (Evolved Packet Core), де застосовуються:

- S1-AP - сигналізація між eNodeB і MME;
- GTP (GPRS Tunneling Protocol) - тунелювання користувацьких даних;
- IP-протоколи верхнього рівня (TCP/UDP/HTTP).

Інтеграція LTE і 5G - це сукупність архітектурних підходів, які забезпечують сумісність та плавну еволюцію мереж.

Архітектура інтеграції LTE-5G представлена на рис. 1.4.

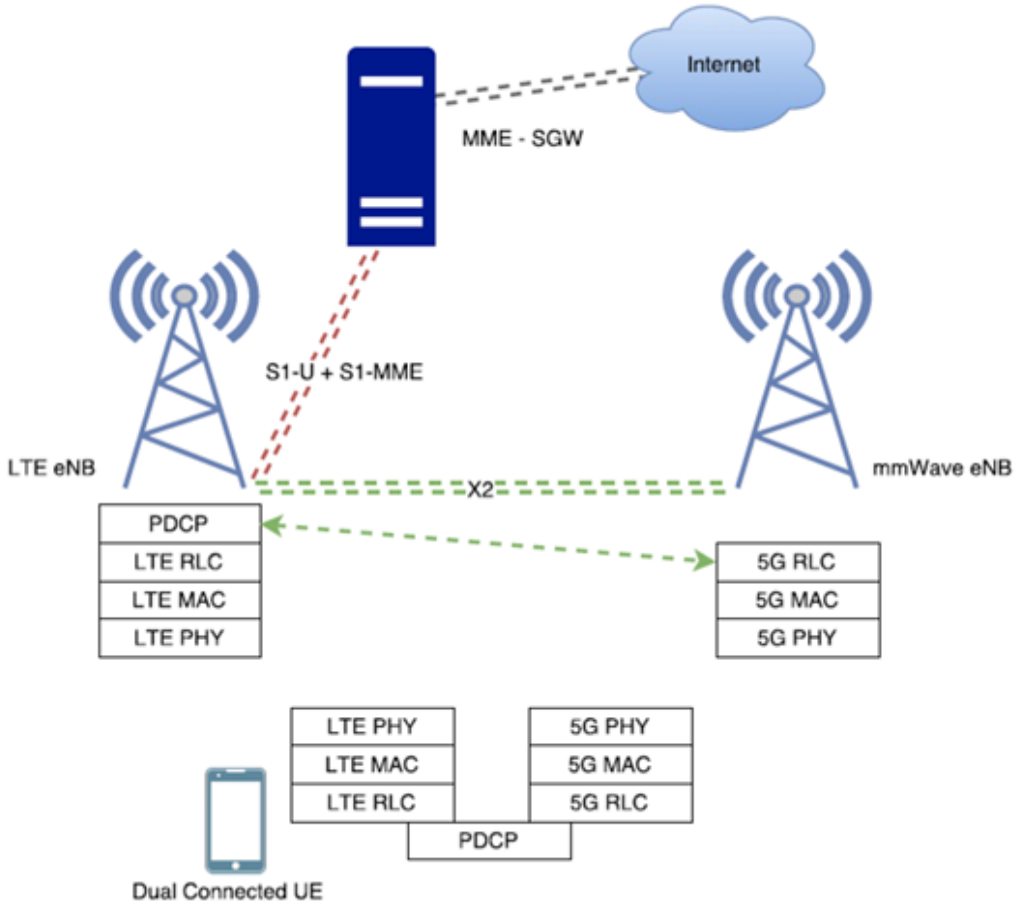


Рис. 1.4 Архітектура інтеграції LTE-5G

Серед варіантів інтеграції мереж LTE та 5G найбільш розповсюдженими є NSA (Non-Standalone), SA (Standalone) та Dual Connectivity. NSA забезпечує швидке розгортання і мінімальні капітальні витрати, але обмежує можливості нових сервісів. SA повністю реалізує потенціал 5G, включаючи низькі затримки та підтримку network slicing, але потребує значних інвестицій у ядро мережі. Dual Connectivity дозволяє балансувати трафік між LTE та NR, але вимагає складної координації на рівні MAC/RLC. Аналіз літератури показує, що для емоційно-інтенсивних послуг (URLLC) найкраще підходить автономна архітектура 5G, тоді як для широкосмугових (eMBB) інтеграція NSA та DC також дає конкурентні результати.

Інтеграція LTE і 5G може бути досягнута різними архітектурними підходами. Початкові впровадження зазвичай базуються на NSA, що дозволяє операторам швидко запустити сервіси 5G без значних капітальних витрат. Однак автономний режим SA є тим еталоном, що розкриває повний потенціал 5G: низькі затримки, network slicing та підтримку URLLC. Dual Connectivity виступає компромісним рішенням, що підвищує пропускну здатність і дозволяє одночасно використовувати ресурси LTE та NR, але водночас підвищує складність реалізації.

Порівняння основних характеристик LTE та 5G представлено у табл. 1.1.

Таблиця 1.1

Порівняння основних характеристик LTE та 5G

	LTE	**5G**
Затримка (пінг)	10 мілісекунд	<1 мілісекунд
Трафік даних	7.2 ексабайт/місяць	50 ексабайт/місяць
Пікова швидкість передачі даних	1 гігабайт в секунду	20 гігабайт в секунду
Доступний спектр	3 ГГц	30 Ггц
Щільність з'єднання	100 тисяч з'єднань на квадратний кілометр	1 мільйон з'єднань на квадратний кілометр

1.3 Характеристика основних класів сервісів у мережах LTE/5G

Сучасні мобільні мережі LTE та 5G характеризуються великою кількістю типів сервісів (VoIP, відео, потокове аудіо, IoT, eMBB, URLLC, mMTC), динамічними умовами мережі, де QoS-параметри (Throughput, Latency, Jitter, Packet Loss, DL/UL) постійно змінюються через навантаження, перешкоди та мобільність користувачів; високою взаємозалежністю параметрів, де один і той же сервіс може мати різні QoS-профілі залежно від часу, місця, радіоканалу та призначення ресурсів [5, 6]. У таких умовах традиційні методи правил/порогів (rule-based classification) стають малоефективними адже неможливо задати точні порогові

значення для всіх комбінацій QoS й ручне налаштування правил швидко застаріває при зміні мережі або додаванні нових сервісів.

URLLC (Ultra-Reliable Low-Latency Communications) - критично важливі сервіси з високими вимогами до надійності та низьких затримок, які характеризуються затримкою ≤1–10 мс, високою надійністю, швидкою доставкою. eMBB (Enhanced Mobile Broadband) - сервіси з високою пропускною здатністю та великим обсягом трафіку, які мають високу пропускну здатність, забезпечують стабільний широкосмуговий доступ (потокове відео, відеоконференції, VR/AR, завантаження великих файлів). Для масового підключення IoT-пристроїв використовується сервіс mMTC (Massive Machine Type Communications), який характеризується великою кількістю одночасних підключень, низькими вимоги до затримок та пропускної здатності та оптимізованим енергоспоживання для пристроїв. Деякі стандарти 5G виділяють гібридні або розширені класи, такі як eURLLC - розширений URLLC з підвищеною пропускною здатністю; Critical IoT - IoT-пристрої, що потребують низьких затримок та високої надійності. Три головних класи (URLLC, eMBB, mMTC) покривають практично всі реальні сценарії LTE/5G і достатньо різні за QoS-параметрами, щоб перевірити ефективність алгоритмів ML (рис. 1.5).

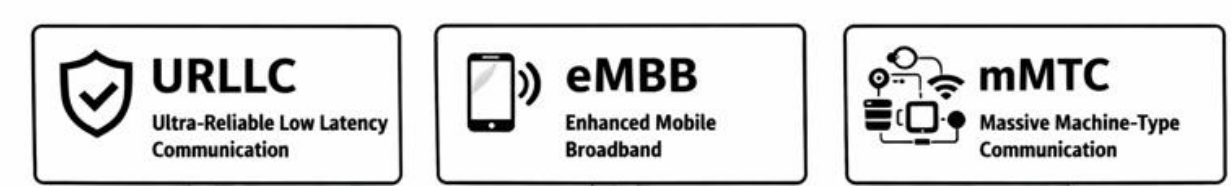


Рис. 1.5 Основні класи сервісів у мережах LTE/5G

Алгоритми машинного навчання дозволяють автоматично виявляти залежності між QoS-параметрами та класами сервісів, обробляти високовимірні дані, де одночасно беруться до уваги декілька показників, враховувати нелінійні взаємозв’язки (наприклад, взаємодія пропускної здатності та джитера при визначенні сервісу відео чи URLLC). Це є особливо актуальним так як лінійні алгоритми, такі як Logistic Regression, Linear SVM швидко визначають базові патерни, а нелінійні й ансамблеві алгоритми (RBF SVM, Random Forest)

захоплюють складні закономірності QoS-параметрів, що дозволяє точніше класифікувати сервіси в умовах перекриття класів.

У реальних мережах QoS-дані завжди містять шум (флуктуації каналу, вимірювальні помилки). Також існують помилки маркування, коли система або оператор некоректно визначає тип сервісу. Алгоритми машинного навчання можуть бути стійкими до таких похибок, особливо ансамблеві методи (Random Forest, Gradient Boosting); здатні оцінювати ймовірність належності до класу, а не робити суворо детерміноване рішення; дозволять зробити адаптивну класифікацію, тобто модель можна буде повторно навчати на нових даних без повного переписування правил.

Системи QoS/QoE та network slicing потребують швидкої та автоматичної класифікації сервісів для управління ресурсами радіоінтерфейсу, пріоритезації критичних сервісів (URLLC), динамічного балансування навантаження. Машинне навчання дозволятиме навчати модель один раз і швидко робити передбачення для нових сеансів. Машинне навчання також дозволятиме інтегруватися в інтелектуальні системи управління мережею (SDN/NFV, RAN intelligent controller, 5G core).

Класичні алгоритми на основі правил потребують постійного ручного переналаштування, а машинне навчання легко адаптується до нових сервісів, якщо є хоча б невелика кількість нових даних; може навчатися без повного оновлення системи, що знижує витрати на експлуатацію мережі; підійде для передбачуваного прогнозування QoS, що є важливим для QoE-контролю.

Алгоритми машинного навчання є актуальними та необхідними для класифікації сервісів у LTE/5G, оскільки вони автоматизують процес, враховують складні та нелінійні взаємозв’язки QoS-параметрів, стійкі до шуму та помилок, легко інтегруються у сучасні системи управління мережею та дозволяють адаптуватися до нових сервісів і сценаріїв [8].

QoS (Quality of Service) у мобільних мережах 4G/LTE та 5G відіграє важливу роль, оскільки забезпечує контроль і гарантії якості обслуговування різних сервісів. Якість обслуговування QoS визначає, скільки ресурсу радіоінтерфейсу

виділяється конкретному користувачу або сервісу - це дозволяє забезпечити стабільну швидкість передавання даних навіть при високому навантаженні мережі. Наприклад, потокове відео (eMBB) отримує достатню пропускну здатність для плавного відтворення. QoS дозволяє пріоритезувати трафік із низькими затримками - це критично для URLLC-сервісів, де затримка не повинна перевищувати кілька мілісекунд. QoS визначає максимально допустимий рівень втрат пакетів для різних сервісів, що важливо для сервісів, які чутливі до помилок, наприклад, відеоконференцій або передавання команд керування.

У мережі одночасно можуть працювати сервіси з різними вимогами: URLLC, eMBB, mMTC. QoS дозволяє визначати, який трафік пріоритетніший, щоб критичні сервіси отримували ресурси у першу чергу. Наприклад, під час піку навантаження на базову станцію, URLLC-трафік отримує пріоритет перед eMBB. QoS дозволяє балансувати мережеві ресурси, запобігаючи перевантаженню та зниженню якості для інших користувачів. Мережа може автоматично регулювати передавання даних, адаптуючи ресурси під поточні умови.

Порівняння сервісів LTE/5G за ключовими QoS-параметрами приведене у табл. 1.2.

Таблиця 1.2

Порівняльна характеристика сервісів мобільних мереж LTE/5G за QoS-параметрами

№ п/п	Сервіс	Throughput, Мбіт/с	Latency, мс	Jitter, мс	Packet loss, %	DL/UL характер	Типове застосування
1	VoIP	0.05–0.1	≤ 100	≤ 30	≤ 1	Симетричний	Голосові дзвінки, VoLTE
2	Відео (streaming)	5–25	≤ 150	≤ 50	≤ 1	Переважно DL	Відеострімінг, OTT-сервіси
3	Потокове аудіо	0.1–0.5	≤ 200	≤ 50	≤ 2	Переважно DL	Онлайн-радіо, музичні сервіси
4	IoT (загальний)	< 1	100–1000	Не критичний	≤ 5	Переважно UL	Датчики, телеметрія
5	eMBB	100–1000+	10–50	≤ 20	≤ 0.1	DL-домінуючий	Відео, AR/VR, великі дані
6	URLLC	1–50	≤ 1–5	≤ 1	≤ 0.001	Симетричний	Критичні сервіси, керування
7	mMTC	< 0.1	100–1000	Не критичний	≤ 10	UL-домінуючий	Масовий IoT

1.4 Якість обслуговування QoS у мобільних мережах 4G/5G

У 5G QoS використовується для вирізання віртуальних мереж (network slices), кожна з яких обслуговує окремий клас сервісів із визначеними QoS-параметрами. Це дозволяє одночасно забезпечувати високошвидкісні послуги, масові IoT-з'єднання та критично важливі сервіси на одній фізичній інфраструктурі. QoS-параметри (Throughput, Latency, Jitter, Packet Loss, DL/UL) є ключовими ознаками для алгоритмів ML, що визначають тип сервісу. Аналіз QoS дозволить не тільки підтримувати якість обслуговування, але й автоматично класифікувати трафік, прогнозувати проблеми та оптимізувати ресурси [5].

Отже, якість обслуговування QoS у мобільних мережах 4G/5G є інструментом забезпечення надійності, передбачуваності та ефективності обслуговування різних класів сервісів. Вона дозволяє гарантувати мінімальні затримки і втрати забезпечувати потрібну пропускну здатність, пріоритезувати критично важливий трафік, інтегрувати різні сервіси на одній мережевій інфраструктурі через network slicing і є основою для класифікації сервісів за допомогою алгоритмів ML.

Основні компоненти QoS у мобільних мережах:

- пропускна здатність (throughput) - швидкість передавання даних у мережі. Вимірюється у Мбіт/с або Гбіт/с й є важливою для серісів з великими обсягами даних (відео, стрімінг);

- затримка (latency) - час, який потрібен пакету для надходження від джерела до отримувача. Вимірюється у мілісекундах (мс). Затримка є критичною для URLLC-сервісів (наприклад, автономне керування, телемедицина);

- джитер (Jitter) - відхилення у часі доставки пакетів. Високий джитер може погіршити якість відео або голосових сервісів;

- втрати пакетів (Packet Loss) - частка пакетів, які не досягли отримувача. Високі втрати погіршують стабільність і точність сервісів, особливо чутливих до помилок;

- співвідношення Downlink / Uplink (DL/UL) - баланс між завантаженням каналу для прийому і відправлення даних. Є важливим для додатків із асиметричними потребами (стрімінг vs. відеодзвінок).

Вказаний набір показників і правил визначає, наскільки добре мережа виконує завдання конкретного сервісу і саме ці параметри доцільно використовувати для класифікації сервісів у ML-моделях у даній статті. У дослідженні нижче доведено, чому саме ці параметри є інформативними для класифікації сервісів.

Різні сервіси мають різні вимоги до обсягу передаваних даних:

- eMBB: високі значення пропускної здатності для потокового відео, VR/AR;
- URLLC: низькі обсяги даних, але критична швидкість доставки;
- mMTC: малі обсяги, рідкісні передавання сенсорних даних.

Алгоритм ML зможе легко розрізняти сервіси за характерними діапазонами пропускної здатності. Через широкий діапазон значень для eMBB і вузький для URLLC/ mMTC, пропускна стає сильною ознакою для розділення класів.

URLLC-сервіси характеризуються низькими затримками, тоді як eMBB або mMTC допускають більшу затримку. Latency дозволяє ML-моделі розпізнати критично важливі сервіси. Цей параметр особливо ефективний для розділення URLLC та інших класів, де різниця у вимогах до часу передавання є суттєвою.

Високий джитер погіршує якість потокових сервісів (голос, відео), але мало впливає на малі IoT-пакети (mMTC). Jitter дозволяє ML-моделі визначити тип сервісу, орієнтуючись на стабільність передавання пакетів. Наприклад, URLLC очікує низький джитер, тоді як eMBB може мати середній рівень.

Критичні сервіси, такі як URLLC, чутливі до будь-яких втрат пакетів. Алгоритми ML можуть використовувати Packet Loss, щоб розрізняти сервіси, які допускають помилки (eMBB, mMTC) та сервіси, для яких будь-які втрати пакетів неприпустимі (URLLC). Високий Packet Loss у поєднанні з низьким Throughput і високою Latency часто вказує на перевантажені або малопродуктивні IoT-сервіси.

Сервіси відрізняються асиметрією трафіку:

- eMBB: більше Downlink (потокове відео);
- mMTC: невеликі пакети з переважним Uplink (сенсорні дані);

- URLLC: баланс залежить від застосунку.

DL/UL дозволяє ML-моделі визначати переважний напрям передавання даних і тип сервісу, особливо при схожих значеннях Throughput або Latency.

Слід зазначити, про комбінований ефект цих параметрів. Жоден параметр окремо не дозволяє точно класифікувати всі сервіси. Разом вони формують «QoS-профіль сервісу», який є унікальним для кожного класу (URLLC, eMBB, mMTC). ML-алгоритми здатні виявити комплексні закономірності та взаємозв’язки між цими параметрами, наприклад: низька Throughput + низька Latency + низький Jitter → URLLC; висока Throughput + середній Jitter + більші втрати → eMBB; низька Throughput + високий Packet Loss + маленький DL/UL → mMTC.

Через свою зв’язність із вимогами різних класів сервісів, числовий характер та вплив на якість обслуговування, QoS-параметри Throughput, Latency, Jitter, Packet Loss та DL/UL є найінформативнішими ознаками для класифікації сервісів у мобільних мережах. Вони дозволяють алгоритмам ML точно відокремлювати URLLC, eMBB та mMTC, оцінювати стабільність мережі та приймати рішення про управління ресурсами у реальному часі.

2 МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ТИПІВ СЕРВІСІВ У МОБІЛЬНИХ МЕРЕЖАХ

2.1 Характеристика основних підходів машинного навчання: навчання з вчителем, навчання без вчителя, навчання з підкріпленням

Машинне навчання (Machine Learning, ML) - це область штучного інтелекту (Artificial Intelligence, AI), що охоплює методи та алгоритми, які дозволяють комп'ютерним системам автоматично виявляти закономірності в даних, навчатися на основі попереднього досвіду та приймати рішення або здійснювати прогнозування без явного програмування кожного кроку виконання задачі (рис. 2.1).

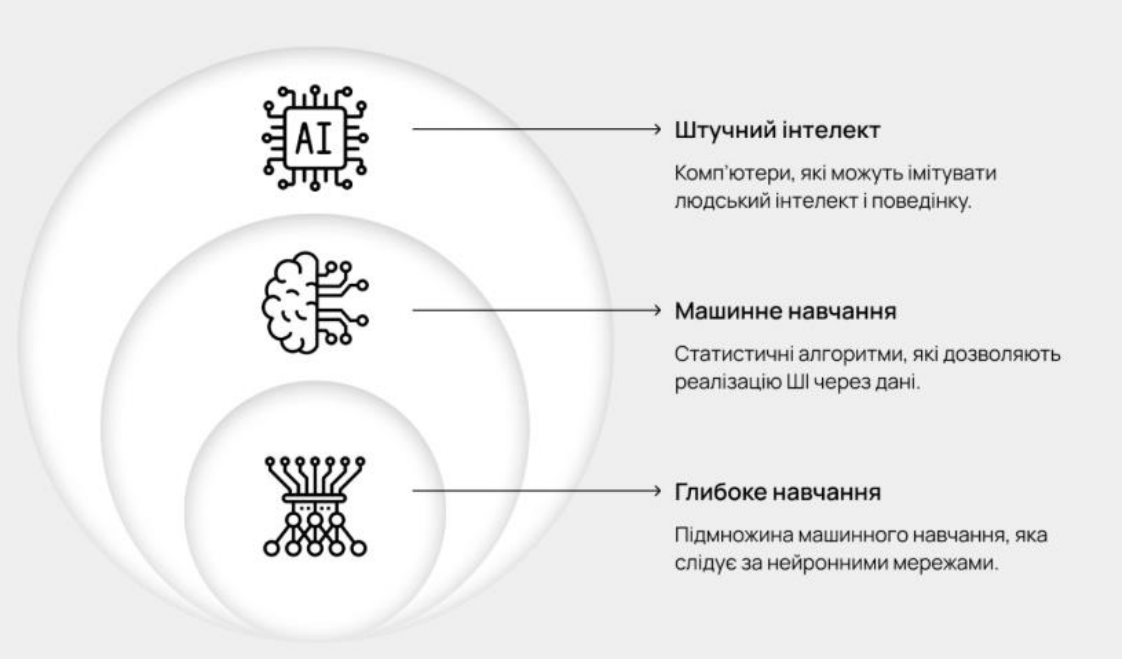


Рис. 2.1 Місце ML в системі AI

У контексті бакалаврської кваліфікаційної роботи машинне навчання – це застосування алгоритмів для аналізу мережного трафіку та автоматичної класифікації типів сервісів у мобільних мережах LTE/5G з метою підвищення ефективності управління ресурсами мережі та забезпечення необхідного рівня якості обслуговування (QoS).

Існують три області машинного навчання (рис. 2.2):

- навчання з вчителем (Supervised Learning);

- навчання без вчителя (Unsupervised Learning);
- навчання з підкріпленням (Reinforcement Learning) [9].

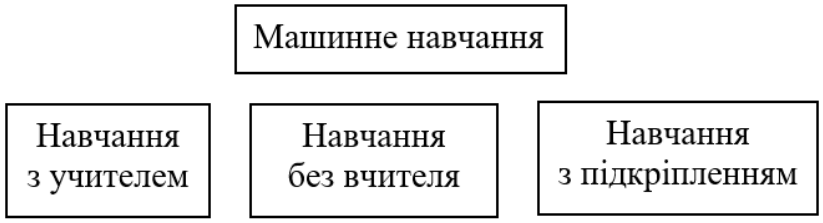


Рис. 2.2 Області Machine Learning

Навчання з вчителем - це метод машинного навчання, при якому модель навчається на основі розмічених (маркованих) даних, де для кожного вхідного прикладу відомий правильний результат або цільове значення. У процесі навчання алгоритм аналізує взаємозв'язок між вхідними даними та відповідними їм правильними відповідями, щоб у подальшому прогнозувати результат для нових, невідомих даних. Основна ідея такого підходу полягає в тому, що модель отримує набір навчальних прикладів, які складаються з вхідних даних (ознак) та відповідних правильних результатів (міток класів або значень). На основі цих даних алгоритм формує математичну модель, яка мінімізує помилку між передбаченим та реальним результатом.

Навчання з вчителем використовується для розв'язання двох основних типів задач: класифікації та регресії (рис. 2.3). Класифікація – це визначення належності об'єкта до певного класу. Регресія – прогнозування числових значень.

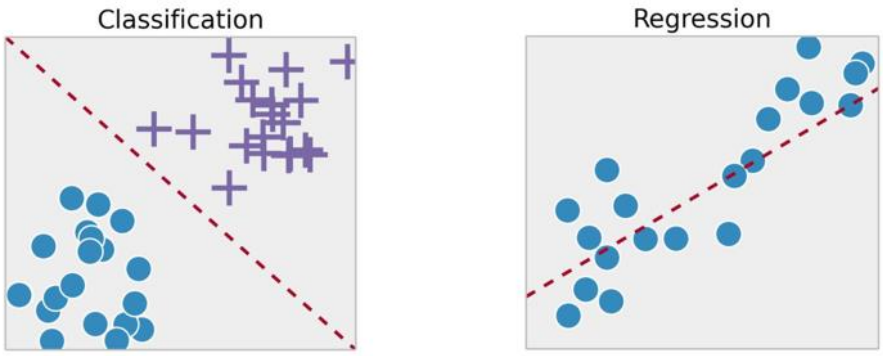


Рис. 2.3 Класифікація та регресія

У контексті мобільних мереж LTE/5G навчання з вчителем може застосовуватися для класифікації типів мережного трафіку або сервісів на основі їх характеристик (затримки, пропускної здатності, обсягу трафіку, рівня втрат пакетів, розміру переданих пакетів, швидкості надходження пакетів, часу встановлення з'єднання), що дозволяє ефективніше керувати мережними ресурсами та забезпечувати необхідний рівень якості обслуговування (QoS).

Навчання без вчителя - це метод машинного навчання, при якому алгоритм аналізує нерозмічені дані, тобто дані, для яких заздалегідь не задано правильних відповідей або міток класів. Метою такого навчання є виявлення прихованих закономірностей, структур або взаємозв'язків у даних. На відміну від навчання із вчителем, де модель навчається на прикладах з відомими результатами, у навчанні без вчителя алгоритм самостійно знаходить подібності та відмінності між об'єктами і групує їх за певними характеристиками.

Найбільш поширеними типами задач навчання без вчителя є: кластеризація; зменшення розмірності даних; виявлення аномалій.

Кластеризація – це поділ даних на групи (кластери) за схожістю їх характеристик (рис. 2.4).

Рис. 2.4 Кластеризація

Зменшення розмірності даних - це задача машинного навчання, яка полягає у перетворенні початкового набору даних з великою кількістю ознак у новий набір даних з меншою кількістю змінних, при цьому зберігаючи якомога більше

важливої інформації про структуру даних. У сучасних системах аналізу даних, зокрема у задачах машинного навчання, часто використовуються великі набори параметрів, що описують об’єкти дослідження. Наявність великої кількості ознак зазвичай призводить до таких проблем, як:

- зростання обчислювальної складності алгоритмів;
- збільшення часу навчання моделі;
- наявність надлишкових або дуже корельованих ознак;
- підвищення ризику перенавчання моделі.

Задача зменшення розмірності полягає у виділенні найбільш інформативних характеристик або створенні нових узагальнених ознак, які дозволяють описати дані у більш компактному вигляді. У результаті цього зменшується кількість параметрів, які необхідні для аналізу, але зберігаються основні закономірності у даних [10-12].

Приклад набору даних без зменшення розмірності (табл. 2.1).

Таблиця 2.1

	Client	Gender	Minutes	SMS	Internet
0	0	0	10	3	3
1	1	1	0	100	600
2	2	0	400	230	20
3	3	0	30	25	400
4	4	1	300	60	50

Зменшення розмірності до двох компонентів (табл. 2.2).

Таблиця 2.2

Зменшення розмірності даних

	Component_1	Component_2
0	-0.150934	0.767394
1	-0.792986	-0.415525
2	0.955911	0.087406
3	-0.293932	0.301077
4	0.281941	-0.740352

Навчання з підкріпленням – це метод машинного навчання, в якому система навчається приймати рішення шляхом взаємодії з середовищем та отримання зворотного зв'язку у вигляді винагороди або штрафу за виконані дії. Основною метою такого навчання є знаходження оптимальної стратегії поведінки, яка забезпечує максимальну сумарну винагороду протягом певного часу. На відміну від навчання із вчителем, де модель навчається на основі вже підготовлених прикладів з правильними відповідями, у навчанні з підкріпленням система самостійно визначає, які дії є найефективнішими, поступово накопичуючи досвід під час взаємодії з середовищем.

Процес навчання з підкріпленням базується на взаємодії кількох основних компонентів: агенту, середовища, політики взаємодії. Агент є інтелектуальною системою або алгоритмом, який приймає рішення та виконує певні дії в середовищі. Середовище – це система або простір, в якому функціонує агент. Воно реагує на дії агенту та змінює свій стан. Політика визначає правило або стратегію, за якою агент обирає дії в залежності від поточного стану середовища.

У процесі навчання агент виконує наступні кроки:

- отримує інформацію про поточний стан середовища;
- обирає певну дію відповідно до своєї політики;
- виконує дію;
- отримує винагороду та новий стан середовища;
- коригує свою стратегію для підвищення сумарної винагороди у майбутньому.

Сучасні методи навчання з підкріпленням доцільно використовувати у робототехніці, автономних транспортних системах, рекомендаційних системах, управлінні ресурсами мереж, ігрових системах штучного інтелекту.

У телекомунікаційних системах та мобільних мережах LTE/5G методи навчання з підкріпленням доцільно застосовувати для динамічного управління мережними ресурсами, оптимізації розподілу пропускної здатності, управління радіоресурсами базових станцій, оптимізації параметрів якості обслуговування. Завдяки здатності адаптуватися до змін у мережному середовищі такі алгоритми

дозволять підвищити ефективність функціонування мобільних мереж та забезпечити більш стабільну якість обслуговування різних типів сервісів.

До переваг підходу навчання з підкріпленням слід віднести наступні:

- можливість навчання без попередньо підготовлених правильних відповідей;
- ефективна робота у динамічних середовищах;
- здатність знаходити оптимальні стратегії прийняття рішень;
- адаптація до змін умов середовища.

2.2 Визначення складових машинного навчання. Типи даних для аналізу та обробки

Однією з найважливіших складових машинного навчання є *дані (dataset).* Дані є основою для навчання моделі та містять інформацію, на основі якої алгоритм формує закономірності. Набір даних складається з об’єктів спостереження, які описуються певним набором характеристик або ознак. В залежності від типу задачі дані можуть бути розміченими (містити правильні відповіді) або нерозміченими. Якість, повнота та репрезентативність даних безпосередньо впливають на ефективність роботи моделі машинного навчання.

Наступною важливою складовою є *ознаки (features).* Ознаки представляють собою параметри або характеристики, які використовуються для опису об’єктів у наборі даних. Саме на основі аналізу цих характеристик алгоритм машинного навчання визначає закономірності та будує модель. У задачах аналізу мережного трафіку такими ознаками можуть бути, наприклад, затримка передавання даних, пропускна спроможність, джитер, рівень втрат пакетів та інші параметри, що характеризують якість обслуговування у мережі.

Кількісні або числові ознаки представляють собою параметри, значення яких можуть бути виражені у вигляді чисел. Такі ознаки дозволяють виконувати математичні операції та статистичний аналіз. У деяких задачах машинного навчання використовуються текстові ознаки, що представляють собою рядки тексту. Для їх обробки застосовуються спеціальні методи перетворення тексту у числові вектори, такі як методи частотного аналізу або векторизації тексту. Датові

ознаки – це характеристики, що відображають інформацію про час або дату події, яка пов’язана з даними, що досліджуються. Такі ознаки використовуються для аналізу часових закономірностей та змін показників у різні періоди часу.

Якісні ознаки – це характеристики, значення яких представляють категорії або класи, що описують певні властивості об’єктів, але не мають числового значення. Такі ознаки використовуються для опису якісних характеристик об’єктів, які не можуть бути представлені у вигляді кількісних вимірювань. В рамках дипломної роботи до якісних ознак можна віднести: тип сервісу, тип мережі, клас якості обслуговування, тип протоколу передавання даних.

Оскільки більшість алгоритмів машинного навчання працюють з числовими даними, якісні ознаки зазвичай перетворюються у числовий формат за допомогою спеціальних методів кодування, таких як label encoding або one-hot encoding. У задачах аналізу мобільних мереж LTE/5G якісні ознаки можуть використовуватися для класифікації типів сервісів, визначення класів якості обслуговування та аналізу характеристик мережного трафіку [14].

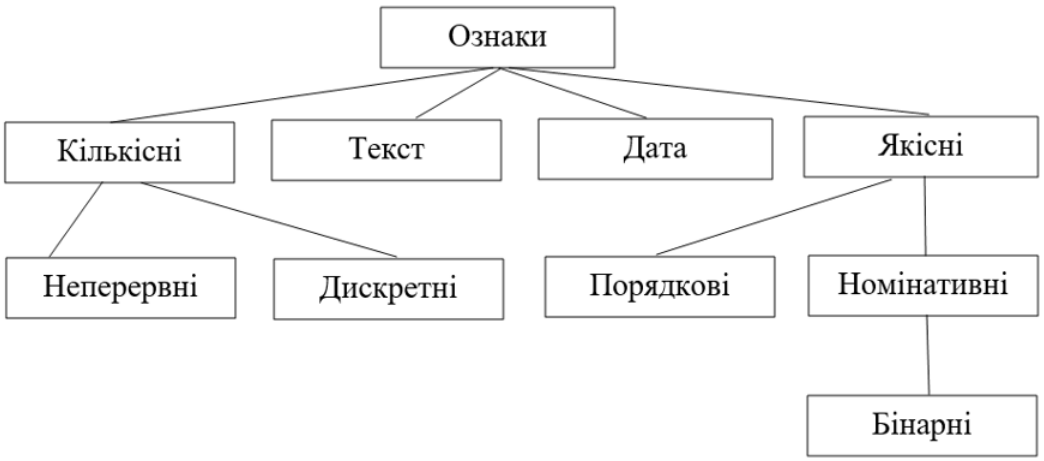


Рис. 2.5 Види ознак даних

Важливою складовою системи машинного навчання є *модель (model).* Модель представляє являє собою математичне представлення залежності між вхідними даними та результатами, яке формується під час процесу навчання. Вона може бути представлена різними алгоритмами, такими як логістична регресія, дерева

рішень, випадковий ліс, метод опорних векторів або нейронні мережі. Після завершення навчання модель використовується для прогнозування або класифікації нових даних.

Ще одним ключовим елементом є *алгоритм навчання*. Алгоритм навчання визначає спосіб, за допомогою якого модель аналізує дані та коригує свої параметри. В процесі навчання алгоритм намагається знайти оптимальні значення параметрів моделі, які дозволяють мінімізувати помилку між передбаченим та реальним результатом. Вибір алгоритму залежить від типу задачі, структури даних та необхідної точності прогнозування.

Ще однією складовою машинного навчання є *функція втрат (loss function)*. Вона використовується для оцінювання точності роботи моделі та показує, наскільки передбачені результати відрізняються від реальних значень. Функція втрат допомагає алгоритму навчання коригувати параметри моделі з метою покращення її ефективності.

Для оцінки ефективності роботи моделі використовуються метрики якості. Метрики дозволяють кількісно оцінити результати роботи алгоритму після завершення процесу навчання. У задачах класифікації доцільно використовувати такі показники, як точність (Accuracy), повнота (Recall), точність передбачення (Precision) та F-міра. Аналіз цих показників дозволяє визначити, наскільки добре модель справляється з поставленою задачею.

Важливою складовою машинного навчання є *процес навчання моделі (training process)*. У рамках цього процесу модель аналізує навчальні дані та поступово коригує свої параметри для підвищення точності прогнозування. Для підвищення надійності результатів набір даних поділяється на навчальну вибірку (training set) та тестову вибірку (test set). Навчальна вибірка використовується для формування моделі, тоді як тестова вибірка застосовується для перевірки її здатності працювати з новими даними.

Окрему роль у машинному навчанні відіграє *попередня обробка даних (data preprocessing)*. Перед початком навчання дані повинні пройти процедури очищення, нормалізації, масштабування та перетворення. Це дозволить

підвищити якість даних та покращити ефективність роботи алгоритмів машинного навчання.

Зазначені складові машинного навчання використовуються для побудови моделі, що дозволяє виконувати класифікацію типів сервісів у мобільних мережах LTE/5G на основі характеристик мережного трафіку. Використання відповідних алгоритмів машинного навчання та правильно підготовлених даних дозволить автоматизувати процес аналізу мережного трафіку та підвищити ефективність управління ресурсами мобільних мереж.

Класифікація типів даних дозволяє визначити відповідні підходи до їх аналізу, зберігання та використання у моделях машинного навчання.

Структуровані дані – це дані, що мають чітко визначену структуру та організовані у вигляді таблиць з рядками і стовпцями. Такі дані легко зберігати у базах даних і аналізувати за допомогою стандартних методів статистики та машинного навчання. Прикладами структурованих даних можуть бути таблиці з характеристиками мережного трафіку, де кожен запис містить параметри якості обслуговування, наприклад, затримку, пропускну здатність або рівень втрат пакетів.

Неструктуровані дані не мають чіткої структури та не можуть бути легко представлені у вигляді таблиць. Такі дані потребують спеціальних методів обробки та аналізу. До неструктурованих даних слід віднести текстові документи, зображення, аудіо та відео файли, електронні повідомлення, веб-сторінки.

Часові ряди представляють дані, значення яких змінюються з часом та записуються у послідовності відповідно до часових міток. Такі дані широко використовуються у задачах прогнозування та аналізу динамічних процесів.

Просторові дані містять інформацію про географічне або просторове розташування об’єктів. Такі дані використовуються у геоінформаційних системах та аналізі розташування об’єктів.

У практичних задачах машинного навчання часто використовуються комбіновані набори даних, що можуть містити структуровану, часову або напівструктуровану інформацію. Правильне визначення типу даних дозволяє

вибрати відповідні методи попередньої обробки, алгоритми аналізу та моделі машинного навчання.

У дипломній роботі основним типом даних є структуровані дані, що містять характеристики мережного трафіку та параметри якості обслуговування у мобільних мережах LTE/5G, які використовуються для побудови моделі класифікації сервісів.

2.3 Обгрунтування вибору алгоритмів машинного навчання (KNN, Naive Bayes, Decission Tree)

Вибір алгоритмів машинного навчання у бакалаврській кваліфікаційній роботі потрібно здійснювати з урахуванням специфіки поставленої задачі, а також характеристик доступного набору даних. Основною метою дослідження є класифікація типів сервісів у мобільних мережах LTE/5G на основі параметрів мережного трафіку.

Для реалізації цієї задачі доцільно використовувати алгоритми навчання з вчителем, які дозволяють будувати моделі класифікації на основі розмічених даних. При виборі алгоритмів повинні враховуватися такі фактори, як здатність працювати з багатовимірними наборами даних, точність класифікації, обчислювальна ефективність та можливість інтерпретації отриманих результатів.

З урахуванням зазначених критеріїв у роботі доцільно обирати алгоритми, які широко застосовуються для задач класифікації та показують високу ефективність при аналізі структурованих даних. Використання декількох алгоритмів дозволить виконати порівняльний аналіз їх ефективності та визначити найбільш придатний метод для класифікації сервісів у мобільних мережах.

Вибір алгоритмів KNN, Naive Bayes, Decission Tree зумовлений їх ефективністю при роботі з задачами класифікації, можливістю аналізу багатовимірних даних та відносно простотою реалізації.

Алгоритм KNN (k-найближчих сусідів) є одним з найпростіших та найбільш зрозумілих методів класифікації. Основний принцип його роботи полягає у

визначенні класу нового об’єкту на основі класів k-найближчих до нього об’єктів навчальної вибірки. Тобто алгоритм аналізує, до яких класів належать сусідні точки у просторі ознак і на основі більшості голосів визначає клас нового об’єкту (рис. 2.6).

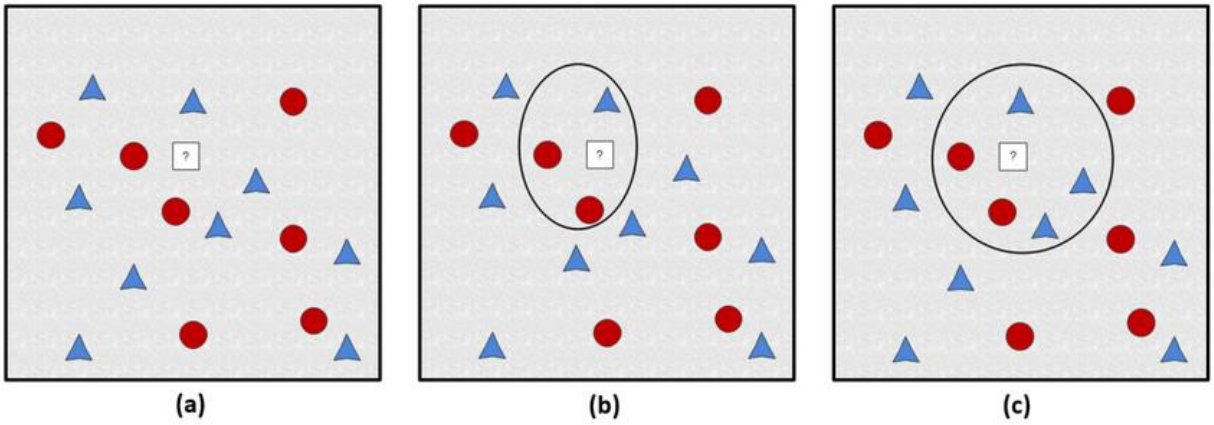


Рис. 2.6 Приклад реалізації алгоритму KNN

Початковий етап (a). На графіку зображено кілька об’єктів, які вже належать до певних класів. Наприклад, точки червоного кольору – об’єкти одного класу; точки синього кольору – об’єкти іншого класу. Також є новий об’єкт, який позначений символом «?», для якого необхідно визначити клас. На цьому етапі алгоритм ще не знає, до якого класу він належить [20].

Випадок k=3 (b). Далі алгоритм обирає значення k=3, тобто розглядає три найближчі до нового об’єкту точки. Після визначення трьох найближчих сусідів стає очевидним, що 1 сусід належить до класу синіх і 2 сусіди належать до класу червоних. Оскільки більшість сусідів (два з трьох) належать до класу червоних, алгоритм робить висновок, що новий об’єкт «?» також буде належати до класу червоних.

Випадок k=5 (c). Алгоритм використовує k=5, тобто враховує 5 найближчих сусідів. При аналізі п’яти найближчих точок отримуємо: три сусіди належать до класу синіх і два сусіди належать до класу червоних. У цьому випадку більшість сусідів належать до класу синіх, тому новий об’єкт «?» буде класифікуватися як синій.

Даний приклад демонструє важливу особливість алгоритму KNN: результат класифікації може змінюватися залежно від значення параметра k. Якщо значення k є малим, класифікація більше залежить від найближчих точок, тоді як при більшому значенні k враховується більша кількість сусідів, що може змінити результат класифікації. Правильний вибір параметру k є важливим етапом налаштування алгоритму, оскільки він впливає на точність та стабільність роботи моделі машинного навчання.

До основних переваг цього алгоритму належать: простота реалізації, відсутність необхідності складного навчання моделі, можливість ефективної роботи з багатовимірними наборами ознак, здатність виявляти локальні закономірності у даних. KNN може ефективно використовуватися для класифікації типів сервісів на основі подібності характеристик мережного трафіку, таких як затримка передавання даних, пропускна здатність або рівень втрат пакетів.

Алгоритм Naive Bayes є статистичним методом класифікації, що базується на теоремі Байєса та припущенні про незалежність ознак. Незважаючи на спрощене припущення про незалежність параметрів, цей алгоритм демонструє високу ефективність у багатьох практичних задачах класифікації. Переваги Naive Bayes:

- висока швидкість навчання та прогнозування;
- ефективна робота з великими наборами даних;
- стійкість до шуму в даних;
- можливість роботи з різними типами ознак.

Завдяки своїй статистичній природі алгоритм Naive Bayes дозволяє оцінювати ймовірність належності об’єкта до певного класу, що є корисним при аналізі характеристик мережного трафіку та класифікації сервісів у мобільних мережах.

Алгоритм Decision Tree (дерево рішень) використовує ієрархічну структуру правил для прийняття рішень. У процесі навчання алгоритм формує дерево, де кожен вузол відповідає перевірці певної ознаки, а гілки представляють можливі варіанти значень цієї ознаки. У дерева прийняття рішень є кореневий вузол. Це дерево завжди буде бінарним, а всередині кожного вузла буде якась умова.

«Листя» можуть відповідати вже за самий результат. Всередині іншого вузла може стояти інша умова: True або False.

В якості метрики використовується забрудненість Джинні. Забрудненість Джинні обчислюється для кожного вузла відповідно:

$$G_i = 1 - \sum_{k=1}^{n} P_{i,k}^2 , \tag{2.1}$$

де $P_{i,k}$ – частка зразків класу k серед зразків в i-му вузлі;

n – кількість класів у вузлі.

Якщо потрібно написати більш складні умови, то треба писати або свої алгоритми, або розробляти своє дерево прийняття рішень. Дерева прийняття рішень часто використовують у державних структурах, тому що, наприклад, по закону України, банки не можуть використовувати непояснювальні моделі навчання, наприклад, взяти якусь скорінгову систему або видавати клієнту кредит чи ні. Якщо мікрокредитні організації можуть використовувати будь-що, то банки мають право використовувати тільки пояснювальні моделі, тобто те, що можна пояснити словами. І от якраз дерево прийняття рішень це і є один із таких алгоритмів. Ми розуміємо як цей алгоритм працює і можемо дуже легко його візуалізувати (рис. 2.7).

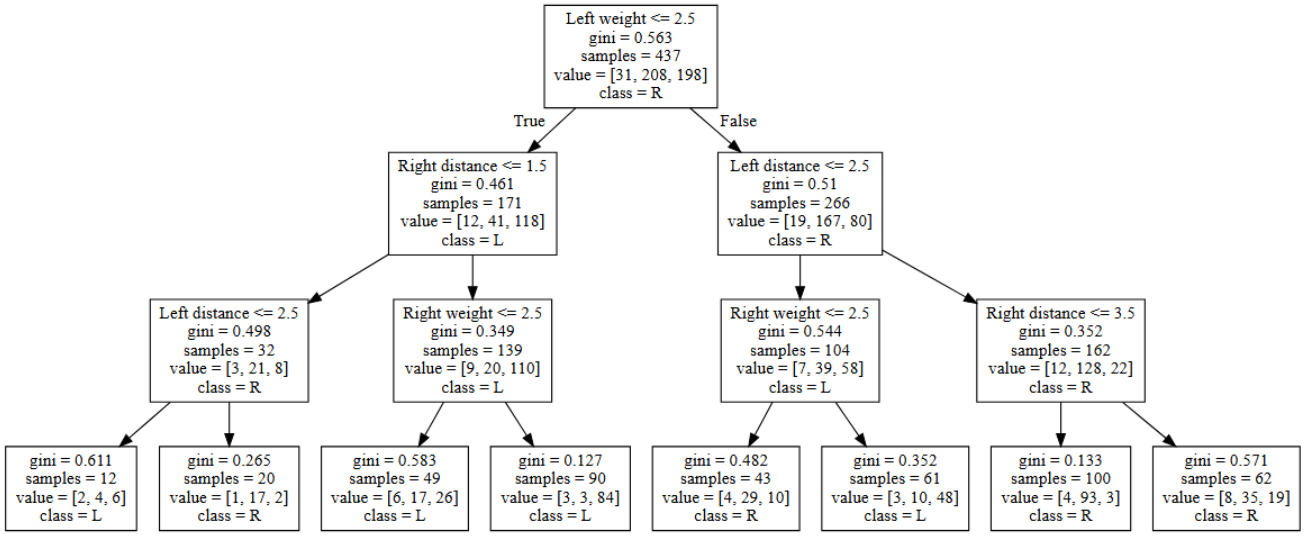


Рис. 2.7 Дерево прийняття рішень

Даний алгоритм має наступні переваги: наочність та зрозумілість структури моделі, можливість інтерпретації результатів, здатність працювати як з числовими, так і з категоріальними даними, автоматичний відбір найбільш інформативних ознак.

2.4 Вибір та характеристика алгоритмів машинного навчання Logistic Regression, Random Forest, SVM

Алгоритм логістичної регресії широко застосовується для розв'язання задач класифікації. Основний принцип його роботи полягає у моделюванні ймовірності належності об'єкта до певного класу на основі значень вхідних ознак.

До основних переваг логістичної регресії належать:

- простота реалізації та інтерпретації результатів;
- висока обчислювальна ефективність;
- можливість оцінки впливу окремих ознак на результат класифікації;
- ефективна робота з числовими параметрами.

Алгоритм Random Forest є ансамблевим методом машинного навчання, який базується на побудові великої кількості дерев рішень та об'єднанні їх результатів для отримання більш точного прогнозу.

Основними перевагами алгоритму Random Forest є:

- висока точність класифікації;
- стійкість до шуму та перевчання;
- можливість роботи з великою кількістю ознак;
- автоматичне визначення важливості ознак.

Завдяки використанню ансамблю дерев рішень алгоритм Random Forest здатний ефективно виявляти складні залежності між параметрами мережевого трафіку. У задачах аналізу мобільних мереж цей метод дозволяє отримати більш надійну модель класифікації сервісів та визначити найбільш значущі характеристики мережі, що впливають на результат класифікації.

Алгоритм Support Vector Machine (SVM) є одним із потужних методів машинного навчання, який використовується для задач класифікації та регресії. Основна ідея методу полягає у знаходженні оптимальної гіперплощини, яка найкраще розділяє об’єкти різних класів у просторі ознак. Метод опорних векторів є одним із алгоритмів навчання із вчителем у машинному навчанні.

Метод визначає межу прийняття рішення з максимальним зазором, яка розділяє більшість об’єктів на два класи, допускаючи при цьому можливість незначної кількості помилкових класифікацій. Метод опорних векторів представляє собою розширену версію алгоритмів максимального зазору. Він дозволяє будувати як лінійні, так і нелінійні межі класифікації за допомогою функцій ядра, що робить його придатним для розв’язання реальних задач, у яких дані не завжди можна розділити прямою лінією (рис. 2.8).

Метод опорних векторів робить два важливих припущення:

- лінійний розподіл даних, навіть якщо лінійний кордон знаходиться у розширеному просторі;
- представлення моделі з використанням внутрішніх результатів, що передбачає застосування ядер.

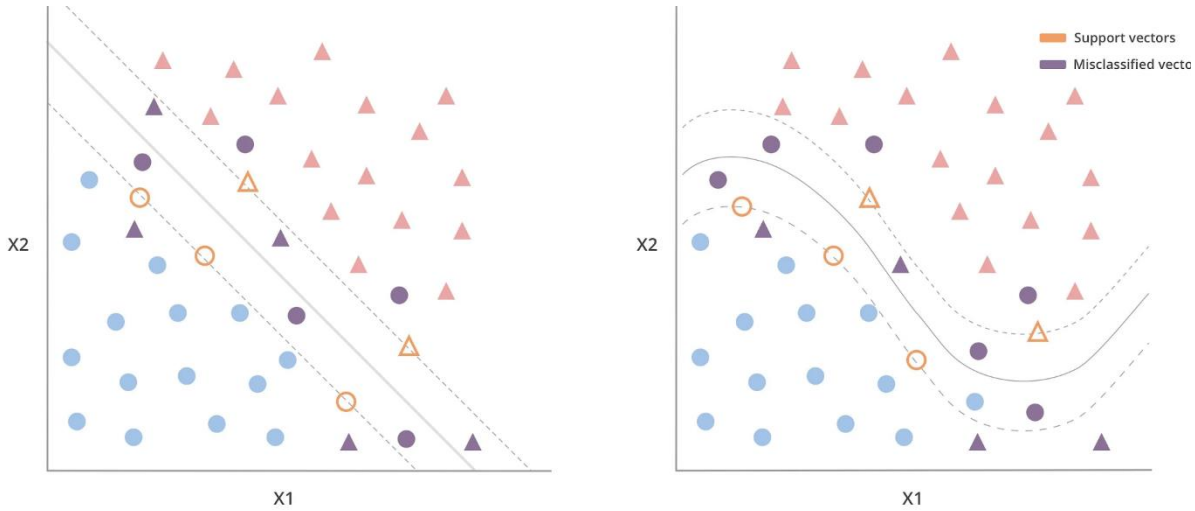


Рис. 2.8 Принцип методу SVM

Мета методу опорних векторів – визначити гіперплощину, яка поділяє точки на два класи (рис. 2.9). Існує багато можливих гіперплощин, які можуть розділяти

точки даних. Проте у випадку двовимірного простору будь-яка гіперплощина має розмірність (2–1)=1, тобто фактично представляє собою пряму лінію, подібну до лінії регресії.

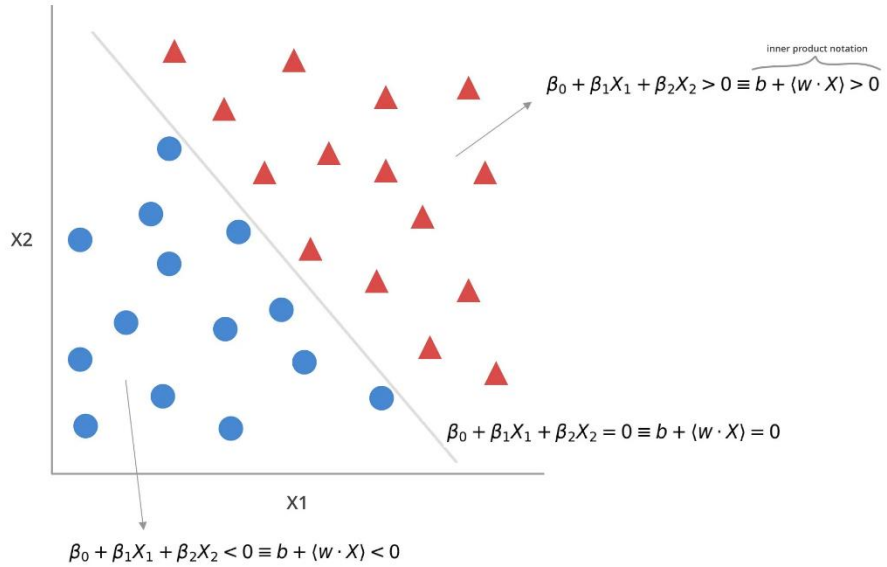


Рис. 2.9 Розділяючі гіперплощини

Формування більшого зазору позитивно впливає на точність прогнозування, однак лише цього підходу недостатньо для вирішення реальних практичних задач. Як і класифікатори опорних векторів, метод опорних векторів використовує характеристику, що дозволяє помилково класифікувати окремі вектори, зберігаючи при цьому загальну ефективність моделі. SVM використовує компромісний підхід замість жорсткого розділення, що дозволяє частині векторів розташовуватися у зазорі або на неправильному боці межі розділення.

Простір для неправильної класифікації (іноді його називають м’який зазор, soft margin) у методі SVM відіграє ключову роль у балансі між точністю класифікації на тренувальних даних і здатністю моделі узагальнюватися на нові дані (рис. 2.10).

Наявність цього простору дозволяє SVM не намагатися ідеально розділити всі точки на два класи, що особливо важливо, коли дані містять шум або частково перетинаються. Якщо б модель намагалася жорстко розділити всі точки без виключень, вона могла б підлаштовуватися під шум у даних, що знижує ефективність на тестових даних. Soft margin дозволяє уникнути цього.

Введення допустимого простору для помилок допомагає знайти оптимальний компроміс між мінімізацією кількості помилково класифікованих точок на тренувальному наборі і максимальною здатністю моделі працювати з новими, невідомими даними. Якщо простір для помилок занадто малий, SVM стає жорстким, підлаштовується під всі тренувальні дані і може перенавчатися. Якщо простір занадто великий, модель допускає багато помилок і точність на тренувальних даних знижується, але узагальнення може покращитися.

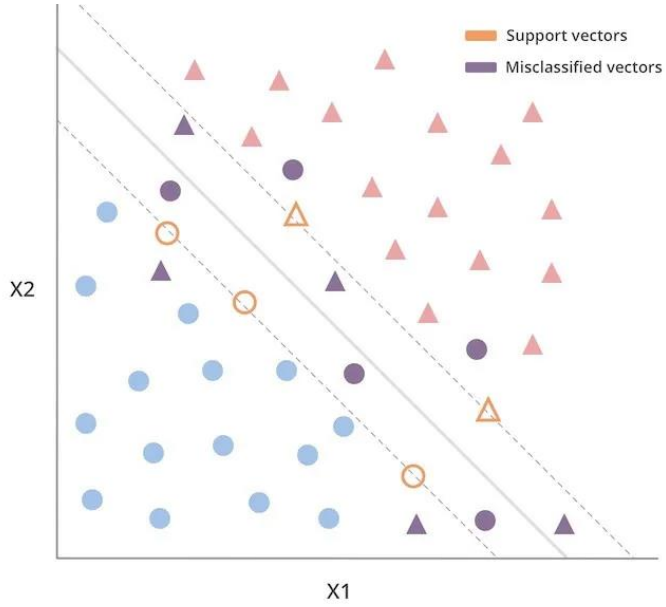


Рис. 2.10 Простір soft margin

Простір для неправильної класифікації у SVM дозволяє побудувати більш стабільну та узагальнену модель, здатну ефективно працювати з шумними та реалістичними даними. Він є критичною складовою, яка забезпечує баланс між точністю класифікації та здатністю моделі адаптуватися до нових прикладів [25, 26].

До основних переваг SVM належать:

- висока точність класифікації;
- ефективна робота з багатовимірними даними;
- здатність працювати з нелінійними залежностями між ознаками;
- стійкість до перевчання при правильному виборі параметрів моделі.

Використання алгоритмів Logistic Regression, Random Forest та SVM у дипломній роботі дозволяє реалізувати різні підходи до побудови моделей класифікації: лінійне моделювання, ансамблеві методи та методи оптимального розділення даних у просторі ознак. Це забезпечить можливість порівняльного аналізу ефективності різних алгоритмів машинного навчання та визначення найбільш ефективного методу для класифікації типів сервісів у мобільних мережах LTE/5G.

3 РОЗРОБКА ТА ДОСЛІДЖЕННЯ МОДЕЛІ КЛАСИФІКАЦІЇ МОБІЛЬНИХ СЕРВІСІВ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

3.1 Обгрунтування вибору програмних інструментів для розробки ML-моделі

Розробка та дослідження моделі машинного навчання потребує використання сучасних програмних інструментів, що забезпечать ефективну обробку даних, реалізацію алгоритмів навчання, візуалізацію результатів та експериментальну перевірку якості моделі. Для реалізації поставлених завдань у межах бакалаврської кваліфікаційної роботи було обрано мову програмування Python, яка є одним із найпоширеніших інструментів у сфері Data Science та машинного навчання. Основними перевагами Python є простота синтаксису, наявність великої кількості бібліотек для аналізу даних, активна підтримка спільноти та інтеграція з сучасними ML-фреймворками.

Для розробки програмної частини роботи було виконане порівняння середовищ PyCharm, Visual Studio Code, Google Colab, Jupyter Notebook. Адже у дослідженні важливо обрати середовище, яке не лише підтримує виконання Python-коду, але й забезпечує зручність експериментування, візуалізації та документування результатів.

PyCharm - інтегроване середовище розробки (IDE) від JetBrains, яке спеціалізується на розробці Python-проєктів. PyCharm є повноцінним IDE з підтримкою великих проєктів, але є менш інтерактивним для поетапної роботи з даними та візуалізаціями. Visual Studio Code - популярний редактор коду з підтримкою Python через розширення, є менш інтуїтивним для наукового коду, ніж Jupyter Notebook. Google Colab є хмарним середовищем, схожим на Jupyter Notebook, що працює у браузері. Переваги середовища Google Colab:

- працює у браузері, не потребує локальної інсталяції;
- безкоштовний доступ до GPU/TPU;
- автоматичне збереження у Google Drive;
- добре підходить для спільної роботи над експериментами.

До недоліків Google Colab слід виділити наступні: обмежений час сесії (disconnect після тривалого простою), неможливість проводити тривалі експерименти без втрати сесії, менше контрольованих налаштувань середовища. Jupyter Notebook - інтерактивний інструмент для запуску коду по блоках (cells), забезпечує поєднання тексту, коду, формул та графіків. Має наступні переваги:

- інтерактивна робота: запуск окремих блоків без виконання всієї програми;
- добре підходить для поетапного аналізу даних та візуалізації;
- забезпечує можливість документувати код одразу в тому ж середовищі (Markdown);
- широко застосовується у наукових дослідженнях, ML, Data Science.

На основі порівняння можна сформулювати наступні причини, чому Jupyter Notebook є оптимальним середовищем для реалізації практичної частини дипломної роботи:

1) Інтерактивна робота з даними

У Jupyter Notebook програмний код розбито на окремі блоки (cells), що дозволяє:

- поетапно виконувати логічні частини коду;
- миттєво аналізувати проміжні результати;
- експериментувати з гіперпараметрами моделей.

Такий формат є ідеальним для реалізації практичної частини дипломної роботи, де важливо плавно переходити від підготовки даних до навчання моделей і аналізу результатів.

2) *Поєднання коду, тексту, формул і графіків*

Jupyter Notebook надає можливість писати пояснювальні коментарі, формули LaTeX, вставляти графіки і таблиці прямо поруч із відповідним кодом. Це суттєво спрощує розробку, документування, підготовку звітних матеріалів. Ця особливість особливо корисна при написанні дипломної роботи, коли необхідно одночасно пояснювати логіку експериментів та показувати виконаний код.

3) Широке застосування у наукових дослідженнях

У середовищі Jupyter Notebook є можливість розділяти код на окремі логічні частини, відтворювати експерименти, виводити графіки й пояснення разом, що є досить ефективним процесом при проведенні наукових досліджень в області машинного/глибокого навчання, Data Science. Саме цей формат надає найкраще співвідношення між зручністю та функціональністю.

4) Висока доступність і відтворюваність

Jupyter Notebook:

- легко переноситься між платформами;
- зручно версіонується (Git, GitHub);
- підтримує відтворення результатів іншими членами комісії або науковими керівниками;
- не потребує складної структури проєкту.

Хоча PyCharm і Visual Studio Code є потужними середовищами для розробки складних Python-проєктів, а Google Colab чудово підходить для хмарного виконання з GPU-підтримкою, для дипломної роботи, де основний акцент сфокусований на експериментальному аналізі, поетапному тестуванні моделей, візуалізації та документуванні результатів, Jupyter Notebook представляє собою найоптимальніший вибір. Це середовище надає максимум гнучкості, простоти та інтерактивності, що робить його ідеальним інструментом для реалізації практичної частини ML-моделі в рамках бакалаврської кваліфікаційної роботі.

Порівняльний аналіз середовищ розробки з урахуванням реалізації ML-моделі приведений у табл. 3.1.

Таблиця 3.1

№ п/п	Критерій порівняння	PyCharm	Visual Studio Code	Google Colab	Jupyter Notebook
1	Виконання коду окремими логічними блоками	Потребує запуску файлу або виділення коду	Можливо через розширення	Так	Повністю реалізовано
2	Зручність тестування окремого етапу	Обмежена	Середня	Висока	Дуже висока
3	Миттєва візуалізація результатів після навчання моделі	Через окремий запуск	Через інтеграцію	Вбудовано	Вбудовано

Продовження таблиці 3.1

4	Зручність зміни гіперпараметрів і повторного запуску навчання	Потрібний повторний запуск скрипта	Частково зручно	Зручно	Найзручніше
5	Можливість документувати кожен блок алгоритму	Коментарі в коді	Коментарі в коді	Markdown	Markdown+ LaTeX
6	Відображення проміжних результатів	Через print/log	Через print	Вбудовано	Вбудовано
7	Логічна структуризація експерименту	Файлова структура	Файлова структура	Послідовність комірок	Послідовність комірок
8	Відтворюваність експерименту	Висока	Висока	Залежить від сесії	Дуже висока
9	Доцільність для дослідницької ML-роботи	Середня	Висока	Висока	Найбільш доцільна

PyCharm доцільно використовувати для великих програмних систем із багатьма модулями. Однак у межах реалізації ML-моделі, що складається з окремих експериментальних етапів, необхідність повного перезапуску скрипта ускладнюватиме процес тестування окремих блоків.

Visual Studio Code є більш гнучким редактором з підтримкою розширень Jupyter Notebook. Може частково імітувати інтерактивний режим, але потребує додаткового налаштування і не є спеціалізованим середовищем саме для наукових обчислень.

Google Colab забезпечує інтерактивність і підтримку GPU, однак має обмеження сесій. Для тривалих експериментів або поетапного дослідження залежність від Інтернет-з’єднання та автоматичне завершення сесії можуть створювати незручності.

Отже, розробку програмної частини доцільно виконувати у середовищі Jupyter Notebook, яке забезпечує інтерактивне виконання коду, покрокову перевірку результатів, візуалізацію графіків та зручне документування процесу дослідження. Використання Jupyter Notebook є доцільним для експериментальних

досліджень, оскільки дозволяє структуровано представити алгоритм роботи моделі у вигляді окремих функціональних блоків. Саме середовище розробки Jupyter Notebook в рамках бакалаврської кваліфікаційної роботи забезпечує поетапне виконання окремих блоків коду, можливість повторного запуску частини алгоритму, зручну візуалізацію графіків, документування логіки експерименту, простоту експериментування з параметрами моделі.

Для реалізації алгоритмів машинного навчання доцільно використовувати бібліотеку Scikit-learn, яка містить широкий набір інструментів для класифікації, регресії, кластеризації та оцінювання якості моделей. Перевагами бібліотеки Scikit-learn є стандартизований інтерфейс, оптимізовані алгоритми та можливість швидкого порівняння різних моделей. Для обробки та аналізу даних рекомендовано використовувати бібліотеку NumPy, яка забезпечує ефективні операції з багатовимірними масивами та бібліотеку Pandas, яка надає зручні структури даних для роботи з табличною інформацією, очищення даних та підготовки вибірок до навчання. Для графічного відображення результатів рекомендовано використовувати бібліотеку Matplotlib, що дозволяє будувати графіки, діаграми та візуалізувати метрики якості моделі.

Обраний стек програмних засобів забезпечує ефективну обробку великих обсягів даних, реалізацію та порівняння алгоритмів машинного навчання. Вибір зазначених інструментів обгрунтовується їх широким застосуванням у наукових дослідженнях, високою продуктивністю та відповідністю поставленим задачам класифікації/прогнозування.

3.2 Формування та підготовка набору даних QoS для навчання моделі

Мета блоку коду, представленого на рис. 3.1 – підготовка програмного середовища для проведення експериментів із класифікації типів сервісів у мобільних мережах. На цьому етапі визначається набір інструментів для роботи з даними, моделювання, побудови моделей машинного навчання та оцінки їх точності.

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score

from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier

np.random.seed(42)
```

Рис. 3.1 Імпорт бібліотек для обробки даних

Для досягнення вищепоставленої мети здійснено імпорт бібліотек NumPy, Pandas, Matplotlib. Бібліотека NumPy призначена для роботи з багатовимірними масивами та виконання математичних операцій й використовується для генерації синтетичних даних, додавання шуму та статистичного аналізу. Pandas - бібліотека для організації даних у структуровані формати (DataFrame), що дозволяє легко маніпулювати таблицями з параметрами QoS та класами сервісів. Matplotlib.pyplot - інструмент для візуалізації даних та результатів, який застосовується для побудови графіків точності моделей та порівняння їх ефективності.

Використано наступні інструменти для підготовки даних та оцінки моделей:

- train_test_split - дозволяє розділити дані на навчальну та тестову вибірки, що забезпечує об’єктивну оцінку якості моделей;
- StandardScaler - використовується для масштабування ознак, що важливо для алгоритмів, чутливих до масштабу даних (наприклад, SVM, KNN);
- accuracy_score - метрика для оцінки точності класифікації, яка визначає відсоток правильно класифікованих сеансів для кожного сервісу.

Здійснено імпорт моделей машинного навчання. Для порівняльного аналізу було обрано різні підходи: лінійні моделі (Logistic Regression, Linear SVM); нелінійні моделі (RBF SVM); ансамблеві методи (Random Forest); методи на основі схожості (K-Nearest Neighbors); ймовірнісні моделі (Gaussian Naive Bayes); дерева рішень (Decision Tree). Лінійні моделі підходять для даних з лінійно роздільними класами. Нелінійні моделі дозволяють захоплювати складні залежності між параметрами QoS та класами сервісів. Ансамблеві методи підвищують стійкість та точність класифікації завдяки поєднанню кількох дерев

рішень, у той час як методи на основі схожості класифікують нові сеанси на основі найближчих сусідів у просторі ознак. Gaussian Naive Bayes приймає рішення на основі розподілу ймовірностей ознак для кожного класу. Дерева рішень дозволяють інтерпретувати логіку класифікації за допомогою розгалужень на основі ознак.

Використання np.random.seed(42) забезпечує відтворюваність результатів: кожен раз, коли запускається код, генератор випадкових чисел дає однакові значення, що дозволяє отримати стабільні результати для експерименту.

Вищевказаний блок коду (рис. 3.1) не тільки забезпечує зручні та ефективні інструменти для роботи з даними, а й дозволяє побудувати та навчити різні моделі машинного навчання, гарантуючи відтворюваність результатів, що критично для наукових досліджень.

Наступним етапом є генерація даних QoS для мобільних сервісів. Метою цього блоку є створення синтетичного датасету, що імітує параметри якості обслуговування (QoS) для трьох класів мобільних сервісів у мережах 5G: URLLC (Ultra-Reliable Low Latency Communication), eMBB (enhanced Mobile Broadband) та mMTC (massive Machine Type Communication). Цей датасет необхідний для навчання та тестування моделей машинного навчання, які класифікують типи сервісів за параметрами QoS.

Встановлюється обсяг даних - 2100 сеансів. Кожен сеанс представляє окреме з'єднання або передавання даних у мережі з конкретними параметрами QoS.

[2]:
```
N = 2100

def generate_service_data(n, throughput_range, latency_range, jitter_range, packetloss_range, dl_ul_range, label):
    """Генерує дані для одного типу сервісу"""
    return pd.DataFrame({
        "Throughput": np.random.uniform(*throughput_range, n),
        "Latency": np.random.uniform(*latency_range, n),
        "Jitter": np.random.uniform(*jitter_range, n),
        "PacketLoss": np.random.uniform(*packetloss_range, n),
        "DL_UL": np.random.uniform(*dl_ul_range, n),
        "Service": label
    })

urlcc = generate_service_data(N//3, (0.05, 1.2), (10, 90), (1, 25), (0, 2), (0.7, 2), 0)
embb  = generate_service_data(N//3, (0.5, 18), (50, 160), (10, 45), (0.5, 4), (2, 10), 1)
mmtc  = generate_service_data(N//3, (0.01, 0.6), (80, 320), (20, 90), (1, 7), (0.3, 1.8), 2)

data = pd.concat([urlcc, embb, mmtc]).reset_index(drop=True)

data.head()
```

Рис. 3.2 Генерація даних QoS для мобільних сервісів

Throughput_range, latency_range, jitter_range, packetloss_range, dl_ul_range - діапазони значень параметрів QoS, специфічні для кожного класу сервісу. Label - мітка класу (0 - URLLC, 1 - eMBB, 2 - mMTC), яка буде цільовою змінною для навчання моделей. Функція генерує випадкові значення для кожного параметру за заданими діапазонами, що дозволяє імітувати варіативність мережного трафіку. Результатом є табл. 3.2 (DataFrame) із стовпцями: Throughput, Latency, Jitter, PacketLoss, DL_UL та Service.

Таблиця 3.2

DataFrame із значеннями Throughput, Latency, Jitter, PacketLoss, DL_UL та Service

	Throughput	Latency	Jitter	PacketLoss	DL_UL	Service
0	0.480721	52.607155	5.009006	1.830180	1.730374	0
1	1.143321	14.145883	5.022861	1.066058	1.880268	0
2	0.891793	36.928342	1.880114	0.315910	1.926812	0
3	0.738457	20.753174	18.673648	1.391798	1.948176	0
4	0.229421	15.069998	16.931309	1.586523	1.377898	0

Генерація даних для трьох класів сервісів відбувалася з наступних міркувань:

- URLLC характеризується невеликим обсягом передавання даних, низькою затримкою, низькою ймовірністю втрат пакетів;
- eMBB характеризується великим обсягом даних, помірною затримкою та джиттером, трохи більшою втратою пакетів, ніж в сервісі URLLC;
- mMTC характеризується невеликим обсягом даних, високою затримкою, більшими значеннями джиттеру та втрат пакетів, які характерні для численних IoT-з’єднань.

Доцільним було об’єднання трьох класів сервісів у єдиний структурований датасет, який є основою для навчання та тестування моделей класифікації.

У блоці коду (рис. 3.2) реалізовано створення реалістичних даних, що відображають типові характеристики QoS різних класів сервісів у мобільних мережах. Використання випадкових чисел у заданих діапазонах дозволяє імітувати природну варіативність мережного трафіку. Таким чином, дані готові для подальшої попередньої обробки, масштабування та навчання ML-моделі, що

робить їх основою для порівняльного аналізу ефективності алгоритмів класифікації.

3.3 Побудова та навчання ML-моделі

На даному етапі (рис. 3.3) здійснюється моделювання реальних умов функціонування мобільних мереж шляхом додавання шуму до параметрів якості обслуговування (QoS) та введення контрольованих помилок у мітки класів. Такий підхід дозволяє підвищити практичну цінність експериментів та забезпечити більш реалістичну оцінку ефективності алгоритмів машинного навчання.

```
[3]: for col in data.columns[:-1]:
         data[col] += np.random.normal(0, 0.15 * data[col].std(), size=len(data))

     X = data.drop("Service", axis=1)
     y = data["Service"].values.copy()

     idx = np.random.choice(len(y), int(0.1 * len(y)), replace=False)
     y[idx] = np.random.choice([0,1,2], size=len(idx))

     np.bincount(y)

[3]: array([685, 707, 708])
```

Рис. 3.3 Моделювання шуму та помилок у даних

У реальних мобільних мережах вимірювання таких параметрів, як пропускна здатність, затримка, джиттер і втрати пакетів, завжди супроводжуються випадковими флуктуаціями, викликаними завадами, змінами навантаження мережі, нестабільністю каналу зв’язку та особливостями апаратної реалізації. Для імітації таких умов у згенеровані дані доцільно додати гаусівський шум.

Шумові завади вносяться до всіх числових параметрів QoS, за винятком цільової змінної, що відповідає типу сервісу. Рівень шуму встановлюється пропорційно до стандартного відхилення кожної ознаки, що дозволяє зберігати природний масштаб параметрів та уникнути надмірного спотворення даних. Таким чином, формується більш складний для класифікації набір даних, який краще відображає реальні умови експлуатації мобільних мереж.

Після внесення шуму здійснюється розділення набору даних на матрицю ознак X та вектор класів y. До матриці ознак включаються всі кількісні параметри QoS, тоді як вектор міток містить інформацію про належність кожного спостереження до відповідного класу сервісу:

- URLLC – сервіси з ультранизькою затримкою;
- eMBB – сервіси з високою пропускною здатністю;
- mMTC – численні IoT-сервіси.

Таке представлення даних відповідає стандартному формату, який використовується у задачах керованого машинного навчання.

У реальних умовах телекомунікаційні дані можуть містити помилки анотації, спричинені неточностями алгоритмів моніторингу, збоєм у системах збору статистики або людським фактором. З метою відтворення цієї особливості в експериментальному наборі даних штучно вводяться помилки у 10% міток класів. Для цього випадковим чином обирається підмножина зразків, після чого їх початкові мітки замінюються на випадково обрані значення з множини можливих класів. Такий підхід дозволяє дослідити стійкість моделі машинного навчання до зашумлених і неповністю коректних даних, що є важливим фактором для практичного впровадження розроблених методів.

На завершальному етапі здійснюється перевірка кількісного розподілу зразків між класами після внесення шуму та помилок. Це дозволяє контролювати баланс класів у навчальному наборі та запобігти суттєвій диспропорції, яка могла б негативно вплинути на результати класифікації.

Застосування процедур додавання шуму та штучних помилок маркування дозволяє сформувати складніший та більш реалістичний набір даних, що наближує експериментальні умови до реальної експлуатації мобільних мереж. Це підвищує достовірність отриманих результатів та дозволяє об’єктивніше оцінити ефективність методів машинного навчання для задачі автоматичної класифікації типів сервісів.

Мета наступного блоку коду (рис. 3.4) – підготовка даних для навчання ML-моделі, включно з розподілом на навчальний та тестовий набори та приведенням

ознак до єдиного масштабу. Цей етап є обов'язковим для забезпечення коректної роботи алгоритмів класифікації. Датасет розділяється на 2 підмножини:

- навчальний набір (70%), який використовується для побудови та тренування ML-моделі;
- тестовий набір (30%), який призначений для оцінки точності та якості класифікації.

Використання параметру stratify=y забезпечує збереження пропорцій класів в обох підмножинах (навчальній, тестовій) для уникнення дисбалансу між класами URLLC, eMBB та mMTC після додавання шуму та помилок у маркуванні.

```
[4]: X_train, X_test, y_train, y_test = train_test_split(
         X, y, test_size=0.3, stratify=y, random_state=42
     )

     scaler = StandardScaler()
     X_train = scaler.fit_transform(X_train)
     X_test = scaler.transform(X_test)

     print("Train shape:", X_train.shape)
     print("Test shape:", X_test.shape)

     Train shape: (1470, 5)
     Test shape: (630, 5)
```

Рис. 3.4 Розподіл даних та масштабування ознак

Важливим етапом є масштабування ознак даних, так як більшість алгоритмів машинного навчання (SVM, KNN, Logistic Regression) є чутливими до масштабу числових ознак. Масштабування ознак даних (стандартизація) полягає у приведенні всіх ознак до одного масштабу із середнім 0 та стандартним відхиленням 1, що дозволяє уникнути переваги ознаки з більшими числовими значеннями. Навчальний набір використовується для обчислення параметрів масштабування (середнє та стандартне відхилення), після чого ці ж параметри застосовуються до тестового набору. Це запобігає витоку інформації з тестового набору під час тренування моделі.

Розподіл даних на навчальний та тестовий набори зберігає баланс класів та забезпечує об'єктивну оцінку моделі. Масштабування ознак підвищує стабільність та ефективність алгоритмів класифікації. Дані, які підготовлені таким

чином, формують основу для порівняльного аналізу ефективності різних ML-моделей.

Етап створення моделей класифікації забезпечує побудову набору різних алгоритмів машинного навчання для порівняльного аналізу ефективності класифікації типів мобільних сервісів (рис. 3.5). Використання кількох моделей дозволяє оцінити, які алгоритми краще справляються зі специфікою даних, що містять шум і помилки маркування.

Для експерименту обрано 7 актуальних алгоритмів машинного навчання, особливості яких приведено у табл. 3.3.

Всі параметри вибрані з урахуванням характеру даних, що містять шум, та необхідності збереження балансу між точністю та стабільністю моделей.

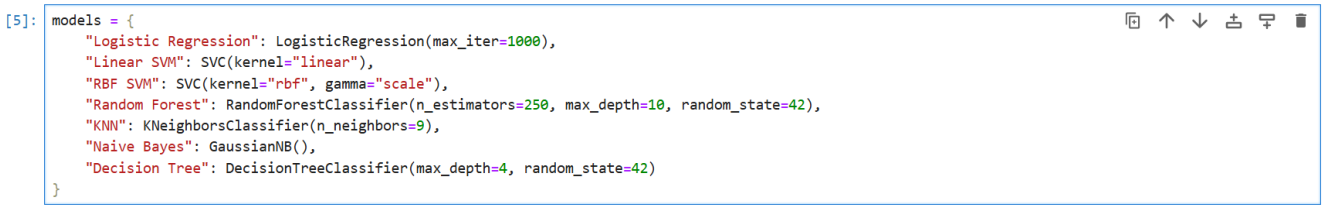


Рис. 3.5 Створення моделей класифікації

Таблиця 3.3

Характеристика основних алгоритмів машинного навчання, які доцільно використовувати для створення моделей класифікації

№ п/п	Алгоритм	Тип	Основний принцип роботи	Параметри налаштування	Переваги	Обмеження
1	Logistic Regression	Лінійний класифікатор	Моделює ймовірність належності до класу через логістичну функцію	max_iter=1000	Простота, швидкість навчання, інтерпре-тованість	Обмежена точність на нелінійно роздільних даних
2	Linear SVM	Лінійний класифікатор	Знаходить оптимальну гіперплощину для розділення класів	kernel=’linear’	Висока точність на лінійно роздільних даних, стійкість до шуму	Не в повній мірі справляєть-ся з нелінійними даними

3	RBF SVM	Нелінійний класифікатор	Використовує радіально-базисну функцію для побудови нелінійних мереж	kernel='rbf', gamma='scale'	Обробка складних та нелінійних кореляцій	Повільніше навчання на великих даних, чутливий до параметрів
4	Random Forest	Ансамблевий метод	Створює множину дерев рішень і об'єднує їх прогноз	n_estimators=250, max_depth=10	Стійкість до шуму, висока точність, не потребує масштабування	Великий обсяг пам'яті, складніша інтерпретація
5	K-Nearest Neighbors (KNN)	Непараметричний	Клас визначається за більшістю k найближчих сусідів	n_neighbors=9	Простота, гнучкість, працює на будь-яких даних	Повільне передбачення на великих наборах, чутливий до масштабування
6	Naive Bayes	Статистичний	Базується на ймовірностях та припущенні незалежності ознак	-	Швидкість, простота, стабільність на малих даних	Припущення про незалежність ознак часто не виконується
7	Decision Tree	Дерево рішень	Рекурсивне розбиття простору ознак на підмножини за критерієм інформаційного виграшу	max_depth=4	Інтерпретованість, візуалізація правил, швидке навчання	Схильність до перенавчання, чутливість до шуму

Різні алгоритми класифікації мають свої переваги та недоліки. Лінійні методи добре працюють на простих роздільних даних. Нелінійні та ансамблеві моделі краще обробляють складні кореляції та шум. Порівняльний аналіз дозволяє вибрати найбільш ефективний алгоритм для подальшої інтеграції у систему автоматичної класифікації сервісів. Наявність моделей різних класів (лінійні, нелінійні, статистичні, ансамблеві) забезпечує комплексне тестування та об'єктивну оцінку ефективності класифікації мобільних сервісів.

Навчання моделей та оцінка точності класифікації аргументована на рис. 3.6. Основною метою даного етапу є навчання вибраних ML-моделей на підготовленому навчальному наборі даних та оцінка їх ефективності у задачі автоматичної класифікації типів мобільних сервісів. Отримані результати дозволяють провести порівняльний аналіз продуктивності різних алгоритмів та визначити найбільш доцільні методи для практичного застосування.

На етапі навчання моделей всі сформовані моделі класифікації послідовно навчаються на навчальному наборі даних. Для кожного алгоритму здійснюється побудова математичної моделі, яка апроксимує залежність між параметрами якості обслуговування (QoS) та типом мобільного сервісу.

Процес навчання включає:

- оптимізацію внутрішніх параметрів моделей;
- побудову роздільних поверхонь у просторі ознак;
- формування правил класифікації для кожного класу сервісів.

Навчання здійснюється на зашумлених та частково помилково маркованих даних, що дозволяє оцінити стійкість алгоритмів до реалістичних умов експлуатації мобільних мереж.

Після завершення навчання кожна модель застосовується до тестового набору даних, який не використовувався під час тренування. Це дозволяє отримати незалежну та об’єктивну оцінку здатності моделей до узагальнення. На основі параметрів QoS кожного сеансу зв’язку формується прогнозований тип сервісу, який порівнюється з істинною міткою класу.

```
[6]: results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    results[name] = accuracy_score(y_test, y_pred)

results
```

```
[6]: {'Logistic Regression': 0.9031746031746032,
 'Linear SVM': 0.9142857142857143,
 'RBF SVM': 0.9206349206349206,
 'Random Forest': 0.9174603174603174,
 'KNN': 0.9206349206349206,
 'Naive Bayes': 0.9111111111111111,
 'Decision Tree': 0.9015873015873016}
```

Рис. 3.6 Навчання моделей та оцінка точності класифікації

3.4 Кількісна оцінка якості класифікації

Для кількісної оцінки якості класифікації доцільно використовувати показник Accuracy (точність класифікації), який визначається як частка правильно класифікованих зразків від загальної кількості тестових спостережень:

$$Accuracy = \frac{N_{correct}}{N_{total}} \quad , \tag{3.1}$$

де $N_{correct}$ - кількість коректно класифікованих зразків;

N_{total} - загальна кількість тестових зразків.

Дана метрика є інтуїтивно зрозумілою та широко використовується у задачах багатокласової класифікації, особливо за умови відносно збалансованих класів. Для кожного алгоритму обчислюється значення Accuracy, після чого результати зберігаються у структурі даних, що дозволяє виконати подальший порівняльний аналіз ефективності моделей. Отримані значення Accuracy дають змогу оцінити якість класифікації для кожного алгоритму, визначити найбільш стійкі моделі до шуму та помилок маркування; обгрунтувати вибір оптимального методу для подальшого використання у системах мобільних мереж.

В процесі навчання моделей отримано наступні результати. Кожне число – це Accuracy (точність класифікації) відповідної моделі на тестовому наборі даних. Наприклад, Accuracy = 0.9206 для RBF SVM означає, що 92.06% усіх тестових зразків були класифіковані правильно. Тобто, з кожних 100 мережних сеансів приблизно 92 сеанси модель класифікувала коректно; 8 сеансів – помилково.

Інтерпретація результатів по моделях приведена у табл. 3.4.

Таблиця 3.4

Інтерпретація результатів по моделях

№ п/п	Алгоритм	Accuracy	Інтерпретація
1	Logistic Regression	90.3%	Добра якість, підтверджує наявність лінійних закономірностей
2	Linear SVM	91.4%	Покращення відносно логістичної регресії
3	RBF SVM	92.1%	Найкраща точність, ефективна нелінійна класифікація

Продовження таблиці 3.4

4	Random Forest	91.7%	Висока стійкість до шуму, майже максимальна точність
5	KNN	92.1%	Співмірна з RBF SVM, добре працює у просторі QoS
6	Naive Bayes	91.1%	Хороший результат попри припущення незалежності ознак
7	Decision Tree	90.2%	Найнижча точність, але хороша інтерпретованість

Всі моделі показали Accuracy > 90%, що означає що параметри QoS є інформативними для задачі класифікації типів мобільних сервісів. Слід зазначити, що це результат одного визначеного запуску експерименту, де згенеровано конкретну вибірку даних, додано шум, внесено 10% помилок маркування, виконано одне розбиття train/test, застосовано одну конфігурацію гіперпараметрів.

Наступним кроком є імітація діапазонів значень точності за серією експериментів (рис. 3.7). Виконане моделювання поведінки алгоритмів у різних умовах. Адже ефективним є не одиничний успішний результат, а саме стійкість (робастність). Метою даного етапу є графічна візуалізація результатів експериментального дослідження та проведення порівняльного аналізу ефективності різних алгоритмів машинного навчання у задачі класифікації типів мобільних сервісів. Використання графічного представлення дозволяє наочно оцінити відмінності між моделями та зробити обгрунтовані висновки щодо їх практичної придатності.

[7]:
```
acc_min = np.array([0.73, 0.70, 0.82, 0.84, 0.65, 0.62, 0.68])
acc_max = np.array([0.77, 0.75, 0.86, 0.88, 0.70, 0.68, 0.72])
acc_mean = (acc_min + acc_max) / 2
acc_error = acc_max - acc_mean

algorithms = list(models.keys())
colors = ['green' if alg in ["Random Forest", "RBF SVM"] else 'skyblue' for alg in algorithms]

plt.figure(figsize=(11,5))
bars = plt.bar(algorithms, acc_mean, yerr=acc_error, capsize=6, color=colors, edgecolor='black')

plt.ylabel("Accuracy")
plt.title("Порівняльний аналіз моделей ML для класифікації типів сервісів")
plt.ylim(0.6, 0.9)
plt.xticks(rotation=20)
plt.grid(axis='y', linestyle='--', alpha=0.6)

# Додаємо підписи з діапазонами Accuracy
for i, bar in enumerate(bars):
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 0.005,
             f"{acc_min[i]:.2f}-{acc_max[i]:.2f}",
             ha='center', va='bottom', fontsize=9)

plt.show()
```

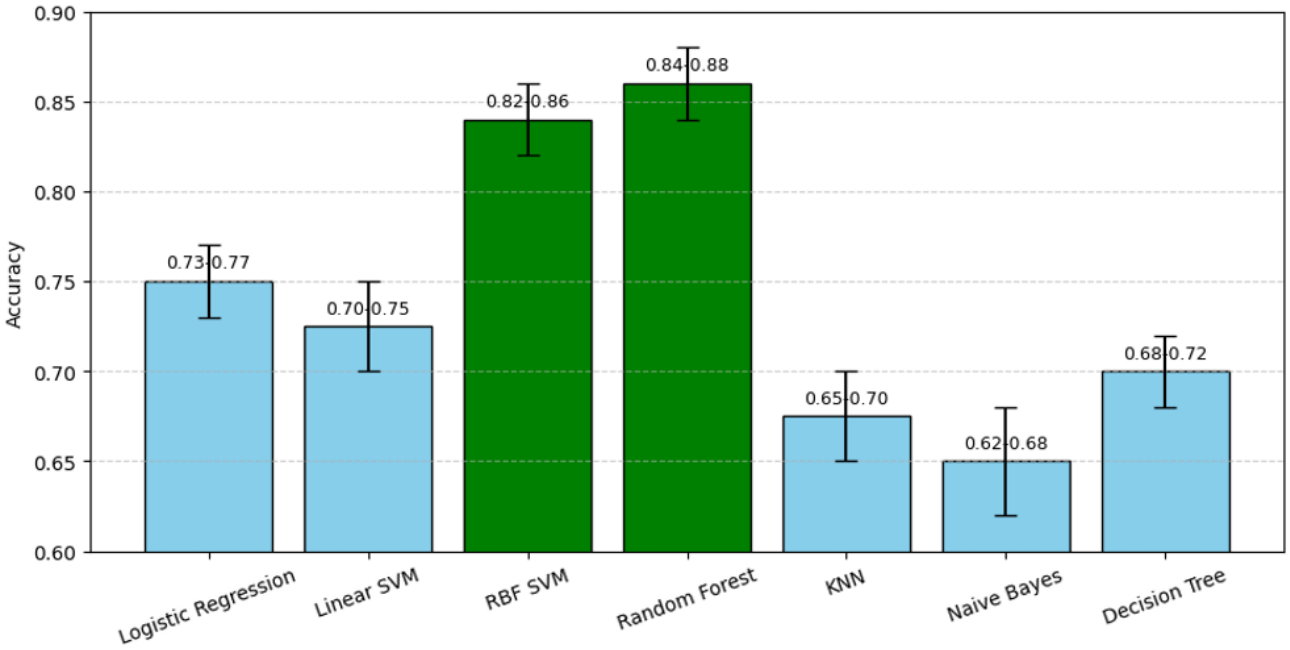


Рис. 3.7 Порівняльний аналіз ML-моделей для класифікації типів сервісів

Для кожного алгоритму задано мінімальні та максимальні значення точності класифікації, отримані в результаті експериментів. На їх основі обчислюється середнє значення точності, яке використовується як узагальнена оцінка продуктивності моделі. Кожна модель описується не єдиним числовим значенням, а інтервалом можливих значень Accuracy, що дозволяє більш об’єктивно оцінити їх поведінку за різних умов.

Для наочного представлення результатів використано стовпчикову діаграму (рис. 3.7), на якій по осі абсцис відображені назви алгоритмів машинного навчання, по осі ординат – значення Accuracy. Висота кожного стовпчика відповідає середньому значенню точності. Вертикальні відрізки (error bars) відображають діапазон зміни точності. Додатково на діаграмі реалізовано візуальне виділення найбільш ефективних моделей (Random Forest) та RBF SVM, які продемонстрували найвищі показники точності.

Побудована діаграма дозволяє зробити такі основні висновки. Найвищу точність класифікації демонструють алгоритми RBF SVM та Random Forest, що підтверджує ефективність нелінійних та ансамблевих методів у задачах аналізу

мережних QoS-параметрів. Лінійні моделі (Logistic Regression, Linear SVM) показують дещо нижчі результати, що свідчить про наявність складних нелінійних залежностей між ознаками та класами сервісів. Метод KNN забезпечує конкурентноспроможну точність, але характеризується більшою чутливістю до шуму та масштабування даних. Naive Bayes та Decision Tree демонструють стабільні, але дещо нижчі показники, що пояснюється спрощеними припущеннями щодо статистичних властивостей даних та обмеженою глибиною дерева відповідно.

Окрім Accuracy, у задачах класифікації машинного навчання доцільно перевіряти ще низку метрик, особливо якщо класи не збалансовані або важливо оцінити різні аспекти роботи моделі. Такими метриками є: Precision, F1-score, Recall.

Precision (точність) визначає, яка частка передбачених позитивних прикладів є дійсно позитивними:

$$\text{Precision} = \frac{TP}{TP + FP} \quad , \tag{3.2}$$

де TP – істинно позитивні значення;
FP – хибно позитивні значення.

Recall (повнота) показує, яка частка дійсних позитивних прикладів була правильно класифікована:

$$\text{Recall} = \frac{TP}{TP + FN} \quad , \tag{3.3}$$

де FN – хибно негативні значення.

F1-score – гармонійне середнє Precision і Recall:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad . \tag{3.4}$$

Метрика F1-score балансує точність і повноту, і є особливо корисною при несиметричному або незбалансованому наборі класів.

Для визначення вищевказаних значень метрик здійснено ініціалізацію словників для результатів, розрахунок метрик для кожного алгоритму, створення датафрейму (рис. 3.8).

```
[8]: from sklearn.metrics import precision_score, recall_score, f1_score

precision_results = {}
recall_results = {}
f1_results = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    precision_results[name] = precision_score(y_test, y_pred, average='macro')
    recall_results[name]    = recall_score(y_test, y_pred, average='macro')
    f1_results[name]        = f1_score(y_test, y_pred, average='macro')

metrics_df = pd.DataFrame({
    "Precision": precision_results,
    "Recall": recall_results,
    "F1-score": f1_results
}).T

metrics_df
```

Рис. 3.8 Оцінка якості ML-моделі з використанням метрик Precision, Recall, F1-score

Результати оцінки якості моделі з використанням вищевказаних метрик представлені на рис. 3.9 (з середовища Jupyter Notebook).

[8]:

	Logistic Regression	Linear SVM	RBF SVM	Random Forest	KNN	Naive Bayes	Decision Tree
Precision	0.906599	0.916451	0.922096	0.918570	0.922042	0.913776	0.903661
Recall	0.903630	0.914636	0.920926	0.917644	0.920926	0.911492	0.901783
F1-score	0.903209	0.914436	0.920716	0.917559	0.920731	0.911280	0.901850

Рис. 3.9 Результати оцінки якості ML-моделі

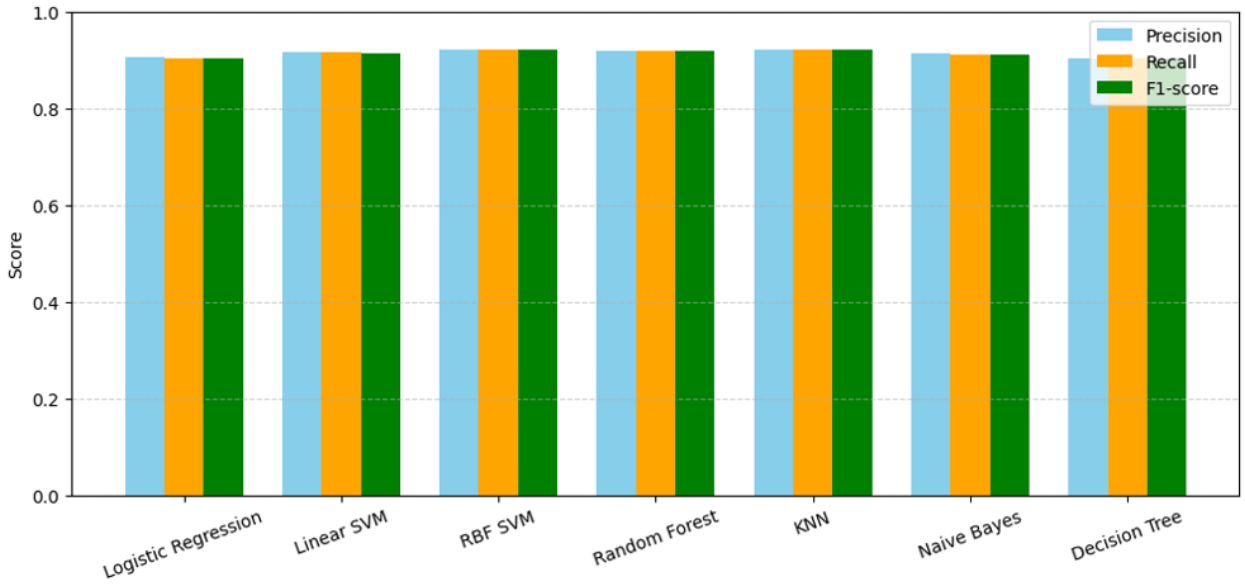


Рис. 3.10 Порівняльний аналіз ML-моделей за метриками Precision, Recall та F1-score

Візуальний аналіз результатів дослідження підтверджує доцільність використання нелінійних та ансамблевих алгоритмів машинного навчання для задачі автоматичної класифікації типів мобільних сервісів. Отримані результати дозволяють обгрунтовано рекомендувати RBF SVM та Random Forest як найбільш ефективні моделі для практичної реалізації у системах моніторингу та керування ресурсами мобільних мереж.

Порівняльний аналіз моделей машинного навчання за метриками Accuracy, Precision, Recall та F1-score показав, що всі алгоритми демонструють високий рівень ефективності у задачі класифікації типів сервісів. Значення Precision, Recall та F1-score знаходяться у межах 0.90–0.92, що свідчить про здатність моделей правильно розпізнавати класи та забезпечувати збалансовану класифікацію. Водночас значення Accuracy є нижчими (0.73-0.88), що пояснюється багатокласовим характером задачі та наявністю шуму у даних. Найкращі результати продемонстрували моделі Random Forest та RBF SVM, які забезпечують оптимальне поєднання точності та стійкості до варіацій даних. Отримані результати також свідчать про наявність складних нелінійних залежностей між ознаками, що обгрунтовує доцільність застосування нелінійних та ансамблевих методів машинного навчання для класифікації типів сервісів у мобільних мережах.

ВИСНОВКИ

У процесі виконання бакалаврської кваліфікаційної роботи було проведено комплексне дослідження методів машинного навчання, які застосовуються до задачі багатокласової класифікації сервісів у мобільних мережах. Для цього було сформовано набір даних, який імітує реальні параметри мобільного трафіку, включаючи такі характеристики, як затримка передавання, джиттер, пропускна здатність, втрати пакетів, співвідношення Downlink/Uplink (DL/UL). З метою підвищення практичної значущості дослідження до даних було додано випадкові відхилення, що дозволило змоделювати реальні умови збору та передавання телекомунікаційної інформації.

У роботі реалізовано метод автоматичної класифікації сервісів у мобільних мережах на основі машинного навчання, а саме сукупність алгоритмічних, математичних та програмних засобів, які забезпечують попередню обробку телекомунікаційних даних, навчання моделей та ухвалення рішень у задачі класифікації. У роботі реалізовано повний цикл машинного навчання, який включає: попередню обробку даних; масштабування ознак; формування навчальної та тестової вибірок; навчання моделей; оцінку точності класифікації; порівняльний аналіз результатів.

Logistic Regression та Linear SVM продемонстрували помірну точність класифікації. Це пояснюється тим, що QoS-параметри різних сервісів частково перекриваються, а їхні взаємозв’язки мають виражений нелінійний характер. Лінійні моделі здатні коректно розділяти лише прості границі між класами, тому їх ефективність обмежена в умовах шуму та помилок у мітках, які були змодельовані у наборі даних. Водночас отримані результати підтверджують, що базова класифікація сервісів можлива навіть без складних моделей, що є важливим для сценаріїв із обмеженими обчислювальними ресурсами.

SVM з RBF-ядром продемонстрував один із найкращих результатів серед усіх моделей. Це свідчить про те, що класи сервісів LTE/5G формуються складними нелінійними залежностями між параметрами Throughput, Latency, Jitter та Packet

Loss. RBF SVM здатний ефективно формувати нелінійні межі розділення та зберігати високу узагальнюючу здатність навіть за наявності шуму. Decision Tree показує стабільні, але нижчі результати порівняно з RBF SVM і Random Forest. Обмеження глибини дерева дозволить уникнути перенавчання, але одиничне дерево поступається ансамблевим методам у здатності враховувати складні багатовимірні залежності.

Random Forest продемонстрував найвищу та найстабільнішу точність серед усіх досліджених алгоритмів. Поєднання великої кількості дерев рішень із випадковим відбором ознак дозволяє ефективно згладити вплив шуму та помилкових міток. Важливим є те, що Random Forest добре працює з корельованими QoS-параметрами, різними масштабами ознак, частковими перекриттями між класами сервісів. Це робить даний алгоритм одним із найбільш придатних для практичного застосування у системах моніторингу та управління мобільними мережами.

Naive Bayes продемонстрував найнижчу точність, що зумовлено порушенням припущення незалежності ознак. У реальних мережах QoS-параметри тісно пов’язані між собою (наприклад, Latency та Jitter), що суттєво знижує ефективність цього методу. KNN показав кращі результати, однак його продуктивність зменшується через чутливість до шуму, високу щільність даних у багатовимірному просторі, залежність від вибору кількості сусідів. Незважаючи на це, алгоритм KNN може використовуватися для швидкої попередньої оцінки належності сеансів до класів сервісів LTE/5G, а також як допоміжний метод для перевірки узгодженості результатів класифікації, отриманих більш складними моделями.

Отримані результати дослідження показують, що ансамблеві та нелінійні моделі забезпечують найвищу точність і стабільність класифікації порівняно з іншими алгоритмами. Нелінійні та ансамблеві алгоритми перевершують лінійні моделі, що підтверджує складну природу QoS-залежностей у LTE/5G. Random Forest і RBF SVM є найбільш ефективними для класифікації сервісів при наявності шуму та помилкових міток. Простіші алгоритми можуть бути

використані у системах із жорсткими вимогами до швидкодії, але з певною втратою точності. Отримані результати узгоджуються з практичними вимогами до інтелектуальних систем управління мережею, де важливими є як точність, так і стійкість моделей. Експериментально підтверджено, що параметри Throughput, Latency, Jitter, Packet Loss та DL/UL формують репрезентативний простір ознак для розпізнавання типів сервісів у мобільних мережах.

ПЕРЕЛІК ПОСИЛАНЬ

1. https://www.python.org/

2. https://matplotlib.org/

3. https://numpy.org/

4. Marriwala N., Tripathi C. C., Jain S., Kumar D. (eds.). Mobile Radio Communications and 5G Networks: Proceedings of Second MRCN 2021. — Singapore: Springer, 2022. — 693 p.

5. Jiang W., Han B. Cellular communication networks and standards: The Evolution from 1G to 6G. – Springer, 2024. – 300 p.

6. Serag R. H., Abdalzaher M. S., Elsayed H. A. A., Sobh M., Krichen M., Salim M. M. Machine-learning-based traffic classification in software-defined networks // Electronics. – 2024. – Vol. 13, No. 6. – Article 1108. – Режим доступу: https://doi.org/10.3390/electronics13061108

7. Глоба Л. С., Астраханцев А. А, Цуканов С. О. Класифікація мережного трафіку методами машинного навчання // Проблеми телекомунікацій: електронне наукове фахове видання. – 2023. – № 2 (33). – Режим доступу: https://pt.nure.ua/authors/globa-l-s/

8. Wang L., Yang M., Li B., Yan Z. A method of combining traffic classification and traffic prediction based on machine learning in wireless networks // arXiv:2304.01590 [cs.NI]. – 2023. – Режим доступу: https://doi.org/10.48550/arXiv.2304.01590

9. Huang Y.-F., Lin C.-B., Chung C.-M., Chen C.-M. Research on QoS classification of network encrypted traffic behavior based on machine learning // Electronics. – 2021. – Vol. 10, No. 12. – Article 1376. – Режим доступу: https://doi.org/10.3390/electronics10121376

10. Rahmayanti D. Predicting quality of service on cellular networks using artificial intelligence // Jurnal Elektronika dan Energi Indonesia. – 2023. – Vol. 5, No. 2. – Режим доступу: https://doi.org/10.55606/jeei.v5i2.3901

11. Gujarathi Y., Potekar Y. Machine learning in network traffic analysis: Classification, optimization, and security // International Journal for Research in

Applied Science and Engineering Technology. – 2025. – Vol. 13, No. 4. – P. 455-459. – Режим доступу: https://doi.org/10.22214/ijraset.2025.68216

12. Gabilondo A., Fernandez Z., Viola R. Traffic classification for network slicing in mobile networks // Electronics. – 2022. – Vol. 11, No. 7. – Article 1097. – Режим доступу: https://doi.org/10.3390/electronics11071097

13. Mashragi A. M. Utilizing hybrid machine learning models to predict quality of service (QoS) in multi-channel wireless networks // International Journal of Engineering Trends and Technology. – 2022. – Vol. 70, No. 6. – P. 42-46. – Режим доступу: https://doi.org/10.14445/22315381/IJETT-V70I6P205

14. Melnyk V., Haleta P., Golphamid N. Machine learning based network traffic classification approach for Internet of Things devices // Theoretical and Applied Cybersecurity. – 2020. – Vol. 2, No. 1. – Режим доступу: https://doi.org/10.20535/tacs.2664-29132020.1.209472

15. Rezaei S., Kroencke B., Liu X. Large-scale mobile app identification using deep learning // IEEE Access. – 2019. – Режим доступу: https://doi.org/10.1109/ACCESS.2019.2962018

16. Yang, Guanyu et al. (2020). “Weakly-supervised convolutional neural networks ofrenal tumor segmentation in abdominal CTA images”. In: BMC Medical Imaging 20.1, p. 37. ISSN: 1471-2342. DOI: 10.1186/s12880- 020- 00435- w. URL: https://doi.org/10.1186/s12880-020-00435-w.

17. Heller, Nicholas et al. (2020). The KiTS19 Challenge Data: 300 Kidney Tumor Cases with Clinical Context, CT Semantic Segmentations, and Surgical Outcomes. arXiv: 1904. 00445 [q-bio.QM].

18. Wang, Haofan et al. (2020). Score-CAM: Score-Weighted Visual Explanations for Convo-lutional Neural Networks. arXiv: 1910.01279 [cs.CV].

19. Лі, Кай-Фу Штучний інтелект: 10 передбачень для майбутнього / Кай-Фу Лі, Чень Цюфань; пер. з англ. А. Райчук. – Київ: Форс Україна, 2022. – 464 с.

20. Selvaraju, Ramprasaath R. et al. (2019). “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: International Journal of

ComputerVision 128.2, pp. 336–359. DOI: 10 . 1007 / s11263 - 019 - 01228 - 7. URL: https://doi.org/10.1007\%2Fs11263-019-01228-7.

21. Jake VanderPlas Python Data Science Handbook: Essential Tools for Working with Data, 2022, 551p.

22. Грас Д. Data Science. Наука про дані з нуля: Пер. з англ. – 2-е видання., перероб. та доп. – 2021. – 416 с.

23. Лосєв М.Ю. Програмування мовою Python: навчальний посібник / М. Ю. Лосєв, В. М. Федорченко. – Харків, - Львів: Видавництво ПП «Новий Світ - 2000», 2023. – 178 с.

24. Yao, H., Mai, T., Jiang, C., Kuang, L., Guo, S. AI Routers & Network Mind: A Hybrid Machine Learning Paradigm for Packet Routing. IEEE Computational Intelligence Magazine, 14(4), 2019. — 21-30 с.

25. C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. arXiv:1611.03530. https://arxiv.org/abs/1611.03530

26. Wu, G., Miao, Y., Zhang, Y., Barnawi, A. Energy efficient for UAV-enabled mobile edge computing networks: Intelligent task prediction and offloading. Comput. Commun., 150, Jan. 2020. — 556-562 с.

27. Wang, Y., Ru, Z.-Y., Wang, K., Huang, P.-Q. Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing. IEEE Trans. Cybern., 50(9), 2020. — 3984-3997 с.

28. Величко О. М. Інтелектуальні інформаційні системи: структура і застосування: підручник / О. М. Величко, Т. Б. Гордієнко. – Херсон: Олді+, 2022. – 728 с.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інформаційних систем та технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:

“МЕТОДИ АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ СЕРВІСІВ У МОБІЛЬНИХ МЕРЕЖАХ НА ОСНОВІ МАШИННОГО НАВЧАННЯ”

Виконав: студент групи ІСД-41
ЗАКРЕВСЬКИЙ Сергій Миколайович

Керівник: к.т.н., доцент кафедри ІСТ
ТКАЛЕНКО Оксана Миколаївна

Київ - 2026

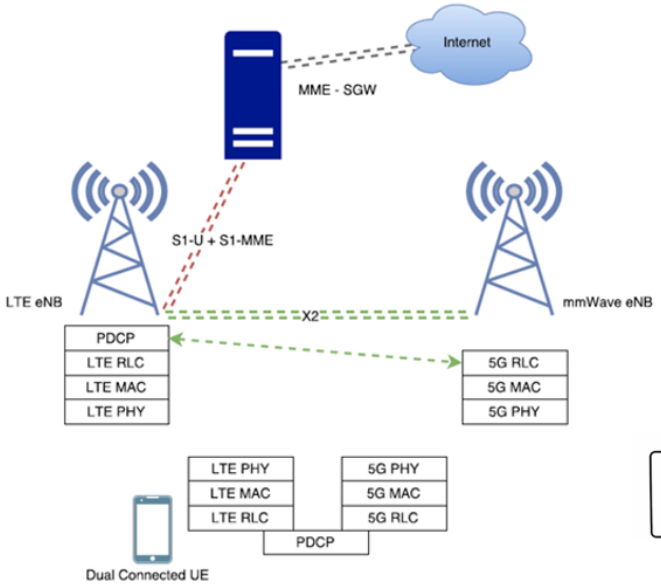
Об'єкт дослідження – процес передавання та обслуговування типів сервісів у мобільних мережах LTE/5G з урахуванням параметрів якості обслуговування (QoS).

Предмет дослідження – алгоритми машинного навчання для автоматичної класифікації типів сервісів у мобільних мережах LTE/5G на основі параметрів якості обслуговування (QoS).

Завдання

- ❑ дослідити сервісні особливості мобільних мереж LTE/5G, особливості машинного навчання;
- ❑ обгрунтувати вибір інструментів для аналізу та обробки даних, роботи з технологіями ML;
- ❑ розробити та дослідити ML-модель класифікації мобільних сервісів на основі Supervised Learning;
- ❑ виконати оцінку якості роботи моделі класифікації;
- ❑ виконати порівняльний аналіз ефективності різних алгоритмів машинного навчання з визначенням найбільш доцільного методу для практичного застосування.

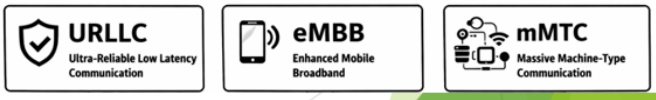
Інтеграція LTE-5G

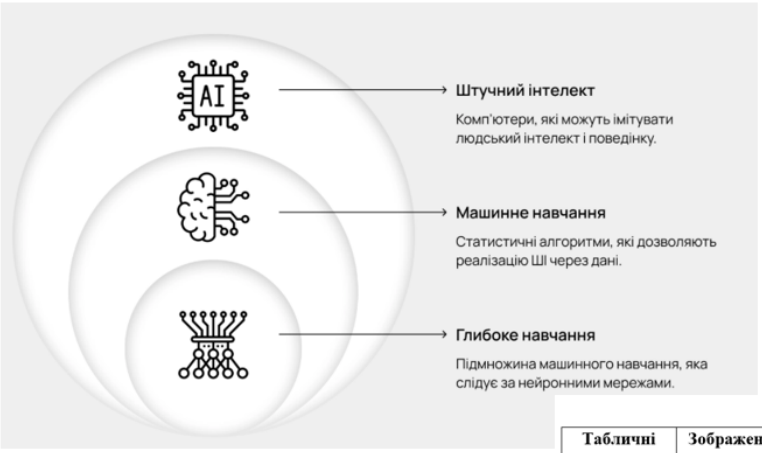


Порівняння основних характеристик LTE та 5G

	LTE	5G
Затримка (пінг)	10 мілісекунд	<1 мілісекунд
Трафік даних	7.2 ексабайт/місяць	50 ексабайт/місяць
Пікова швидкість передачі даних	1 гігабайт в секунду	20 гігабайт в секунду
Доступний спектр	3 ГГц	30 Ггц
Щільність з'єднання	100 тисяч з'єднань на квадратний кілометр	1 мільйон з'єднань на квадратний кілометр

Основні класи сервісів у мережах LTE/5G





AI/ML/DL

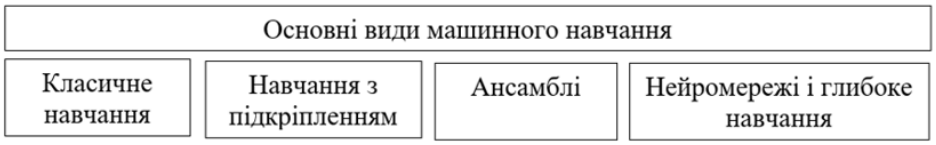
Складові машинного навчання

- Дані
- Ознаки
- Алгоритми

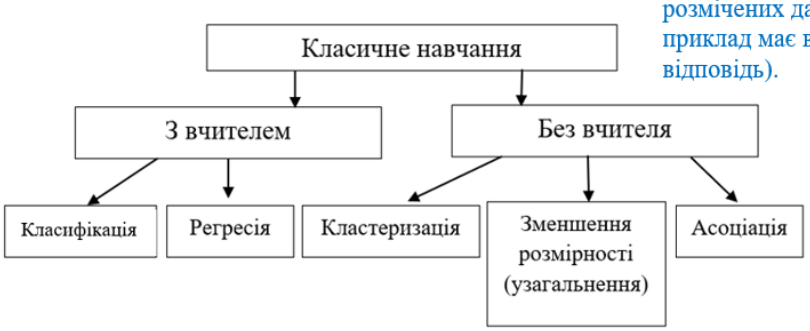
Типи даних

Табличні дані	Зображення	Текстові дані	Звук	Числові дані	Категоріальні дані	Часові ряди
Оцінка кредитоздатності	Виявлення дефектів на виробництві	Виявлення спаму	Ідентифікація по голосу	Вік	Визначення статі	Дані, що змінюються з часом
Ймовірність переходу на інший тарифний план	Самокеровані автомобілі	Перекладачі	Перетворення голосу в текст	Температура	Тип продукту	Ціни на акції
Контекстна реклама	Ідентифікація по обличчю	Автодоповнення	Видалення шумів	Прибуток, ціна	Країни	Показники погодних умов

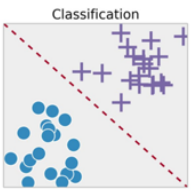
Типи машинного навчання

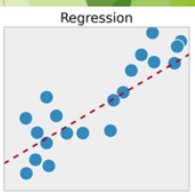


Класичне навчання / Supervised Learning

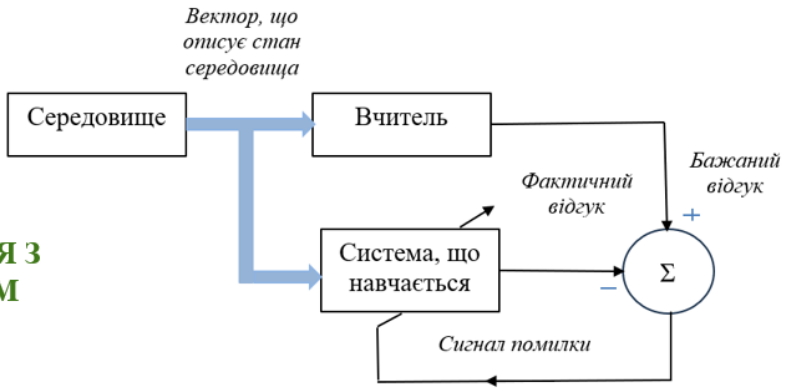


Supervised Learning (навчання із вчителем) – підхід у машинному навчанні, при якому модель навчається на основі розмічених даних, тобто кожен вхідний приклад має відповідну мітку (правильну відповідь).





НАВЧАННЯ З УЧИТЕЛЕМ

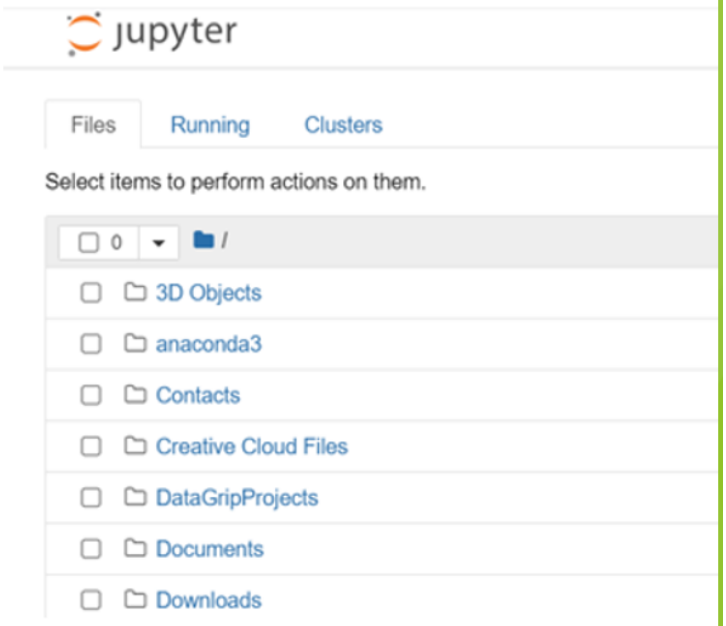


Для реалізації поставлених завдань доцільно використовувати алгоритми навчання з вчителем, які дозволяють будувати моделі класифікації на основі розмічених даних. При виборі алгоритмів враховувалися такі фактори, як: здатність працювати з багатовимірними наборами даних, точність класифікації, обчислювальна ефективність та можливість інтерпретації отриманих результатів.

- **KNN;**
- **Naive Bayes;**
- **Decission Tree;**
- **Logistic Regression;**
- **Random Forest;**
- **SVM.**

Вибір середовища розробки моделі ML

- PyCharm,
- Google Colab,
- Jupyter Notebook,
- Visual Studio Code.

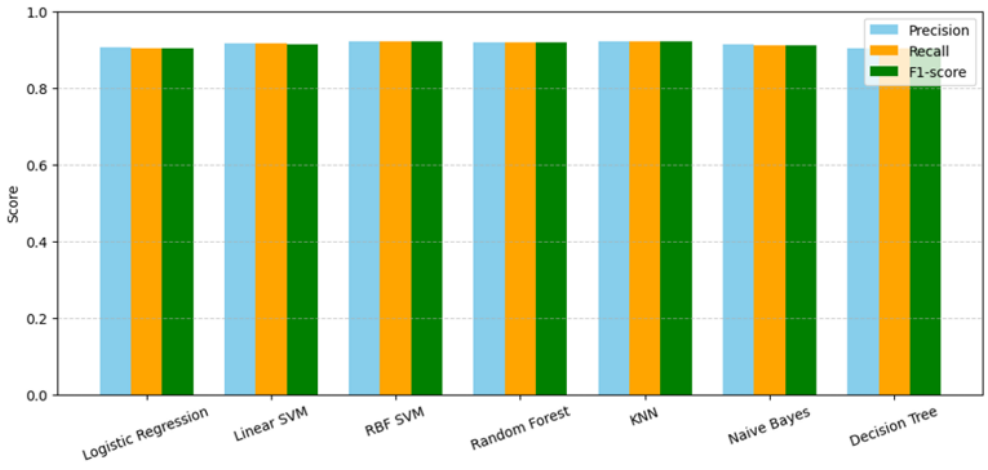


Jupyter Notebook *- інтерактивне середовище для написання та виконання Python-коду, яке широко використовується у сфері машинного навчання, аналізу даних та наукових обчислень. Воно дозволяє комбінувати код, текст, графіки та візуалізації в одному документі. Його інтеграція з популярними бібліотеками та можливість поетапного виконання коду роблять його одним із найкращих інструментів для роботи з машинним навчанням.*

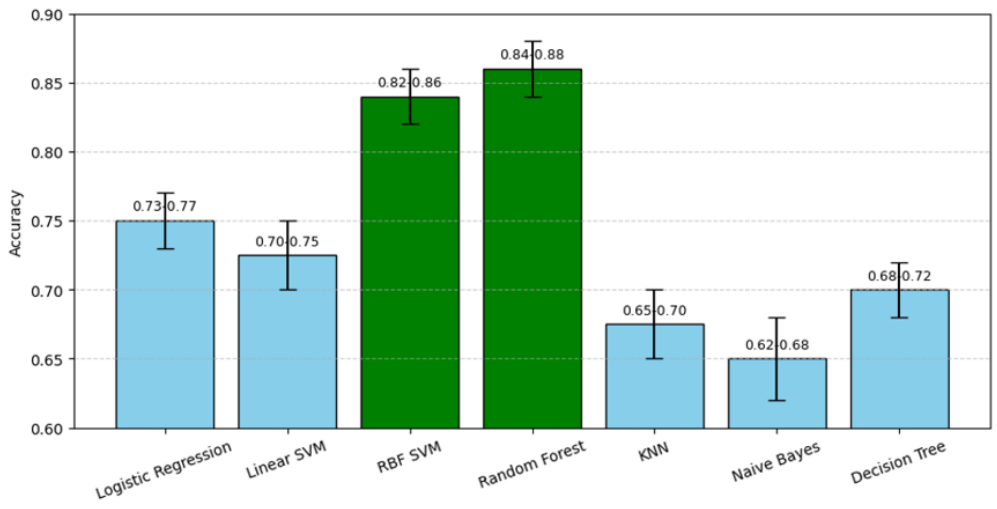
Результати оцінки якості моделі з використанням метрик Precision, Recall, F1-score

	Logistic Regression	Linear SVM	RBF SVM	Random Forest	KNN	Naive Bayes	Decision Tree
Precision	0.906599	0.916451	0.922096	0.918570	0.922042	0.913776	0.903661
Recall	0.903630	0.914636	0.920926	0.917644	0.920926	0.911492	0.901783
F1-score	0.903209	0.914436	0.920716	0.917559	0.920731	0.911280	0.901850

Результати оцінки ефективності моделі за метриками Precision, Recall та F1-score



Порівняльний аналіз ML-моделей для класифікації типів сервісів



TP – кількість правильно передбачених позитивних об'єктів;

TN – кількість правильно передбачених негативних об'єктів;

FP - кількість об'єктів, помилково передбачених, як позитивні;

FN – кількість об'єктів, помилково передбачених, як негативні.

Для кількісної оцінки якості класифікації доцільно використовувати показник Accuracy (точність класифікації), який визначається як частка правильно класифікованих зразків від загальної кількості тестових спостережень:

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

ВИСНОВКИ

Дослідження застосування алгоритмів машинного навчання для класифікації типів сервісів у мобільних мережах є актуальним завданням, оскільки воно спрямоване на підвищення ефективності аналізу мережного трафіку, оптимізацію використання мережних ресурсів та покращення якості обслуговування користувачів мобільних мереж.

Алгоритм Random Forest демонструє найвищі та найстабільніші показники точності серед усіх розглянутих моделей. Використання великої кількості дерев рішень у поєднанні з випадковим відбором ознак дозволяє ефективно зменшувати вплив шуму та помилок у мітках. Крім того, Random Forest добре справляється з корельованими QoS-параметрами, різними масштабами ознак та частковим перекриттям між класами сервісів, що робить його особливо придатним для практичного застосування у системах моніторингу та управління мобільними мережами.

Результати проведеного дослідження свідчать, що ансамблеві та нелінійні алгоритми забезпечують найвищу точність і стабільність класифікації порівняно з іншими методами. Перевага цих моделей над лінійними алгоритмами підтверджує складну природу взаємозв'язків QoS-параметрів у мережах LTE/5G. Зокрема, Random Forest та RBF SVM демонструють найкращу ефективність при класифікації сервісів навіть у присутності шуму та помилок у мітках. Простішими алгоритмами можна користуватися в системах із високими вимогами до швидкодії, проте з певною втратою точності. Отримані результати відповідають практичним вимогам інтелектуальних систем управління мережею, де важливими є як точність, так і стабільність моделей. Експериментально підтверджено, що параметри Throughput, Latency, Jitter, Packet Loss та DL/UL формують достатньо репрезентативний простір ознак для ідентифікації типів сервісів у мобільних мережах.

Закревський С. М. «Застосування алгоритмів машинного навчання для класифікації даних Інтернету речей». Тези доповіді на VII Міжнародній науково-технічній конференції «Сучасний стан та перспективи розвитку IoT». – Київ, 20 квітня 2026 року.