ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ, ПРАВА ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Пояснювальна записка

до кваліфікаційної роботи на здобуття ступеня вищої освіти магістр

на тему: **«Прогнозування завантаженості станцій мобільного зв'язку за допомогою нейронної мережі»**

Виконала: здобувачка вищої освіти
за освітньо-професійною програмою
Інформаційні управляючі системи та технології спеціальності
126 Інформаційні системи та технології
ступеня вищої освіти магістр
групи 126ІСТ_мз_2022[1](л.н.)
Христенко А. В.
Керівник: Слюсар В. І.
Рецензент: Муравльов В. В.

Полтава – 2023 року

ВСТУП

Актуальність теми кваліфікаційної роботи підтверджується необхідністю прогнозування мобільного трафіку для ефективного планування інфраструктури та управління мережею мобільного зв'язку. Традиційні підходи вимагають збору вимірювань з різних базових станцій і надсилання їх до центрального об'єкта з подальшим виконанням машинного навчання з використанням отриманих даних. Однак вони обмежені інформацією про реєстрацію базової станції та забезпеченням гарантій конфіденційності. Все це свідчить про актуальність теми роботи.

Зв'язок роботи з науковими програмами, темами. Робота відповідає дослідженням в рамках науково-дослідної роботи «Управління стратегією інноваційного розвитку підприємств в контексті підвищення їх конкурентоспроможності на аграрному ринку, сталого розвитку та забезпечення продовольчої безпеки держави» (2021 р.), що фінансувалась господарськими договорами із замовниками, Концепції розвитку штучного інтелекту в Україні (розпорядження Кабінету Міністрів України № 1787-р від 29.12.2021), тематиці досліджень навчально-дослідної лабораторії інтелектуальних систем, комп'ютерних мереж та інтернет речей кафедри інформаційних систем та технологій Полтавського державного аграрного університету.

Метою кваліфікаційної роботи є підвищення ефективності систем мобільного зв'язку.

Завданнями кваліфікаційної роботи є:

– визначення особливостей трафіку станцій мобільного зв'язку;

– вибір архітектури нейронної мережі для обробки часових рядів;

– оцінка розробленої моделі глибокого навчання прогнозування завантаженості станцій мобільного зв'язку.

Об'єктом дослідження є процес прогнозування завантаженості станцій мобільного зв'язку.

Предметом дослідження є точність нейронної мережі, що застосовуються для прогнозування завантаженості станцій мобільного зв'язку.

Методами дослідження є аналітичний, інформаційно-пошуковий, методи синтезу та навчання нейронних мереж обробки часових рядів, робота з фреймворком PyTorch.

Інформаційна база кваліфікаційної роботи базується на ресурсах з даними про згорткові та рекурентні нейронні мережі, інструментарій для виконання глибокого машинного навчання.

Елементи наукової новизни роботи полягають у створенні моделі глибокого навчання нейронної мережі для прогнозування завантаженості станцій мобільного зв'язку.

Практична значущість роботи полягає у розробці рекомендацій щодо використання моделі глибокого навчання нейронної мережі для прогнозування завантаженості станцій мобільного зв'язку, які можуть бути використані для подальших досліджень за даною тематикою та при проектуванні систем мобільного зв'язку.

Апробація результатів відбувалася в рамках XXII Міжнародної науково-технічної конференції «Приладобудування: стан і перспективи», (травень 2023 р., м. Київ), V Міжнародної студентської конференції «Теоретичне та практичне застосування результатів сучасної науки» (жовтень 2023 р., м. Рівне).

За результатами досліджень здійснено 2 публікації тез доповідей.

Структура кваліфікаційної роботи логічно пов'язана з завданнями досліджень і містить вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг пояснювальної записки кваліфікаційної роботи складає 67 сторінок формату А4. Вона містить 17 рисунків.

РОЗДІЛ 1
АНАЛІЗ ОСОБЛИВОСТЕЙ ПРОГНОЗУВАННЯ ТРАФІКУ МОБІЛЬНОГО ЗВ'ЯЗКУ

1.1 Ключові аспекти розподіленого навчання

Розподілене навчання (Federated Learning, FL) [1] для нейронних мереж [2] – це відносно новий підхід у галузі машинного навчання, який почав розвиватися останніми роками. FL відкриває нові можливості для застосування штучного інтелекту (Artificial Intelligence, AI) у чутливих до конфіденційності областях, забезпечуючи при цьому ефективне та безпечне використання даних.

Вперше концепція FL [3] була представлена у наукових колах у 2016 р. Макмеханом та ін. [4] як проект Google AI. Самі перші застосування FL реалізувала Google у кількох своїх продуктах, у тому числі для покращення прогнозів тексту в клавіатурах на мобільних пристроях, таких як Gboard. Це дозволило покращити якість прогнозів, не розкриваючи особисті дані користувачів. Також команда запропоновала підхід до розподіленого навчання моделей машинного навчання, який дозволяє спільно навчати дані на кількох пристроях. Він відрізняється від традиційних централізованих методів машинного навчання, де всі локальні дані завантажуються на центральний сервер, а також від більш звичайних децентралізованих підходів, які часто припускають, що локальні зразки даних повинні бути однаково розподілені [5]. Тобто під час такого навчання зберігається конфіденційності даних (рис. 1.1). Після початкового впровадження концепція FL привернула увагу академічної спільноти та промисловості. У наступні роки з'явилося багато досліджень, спрямованих на покращення ефективності, безпеки та масштабованості FL. В останні роки FL отримав подальший розвиток, особливо в областях, де конфіденційність даних має вирішальне значення, таких як охорона здоров'я та фінанси. Дослідники та

інженери продовжують працювати над покращенням цієї технології, роблячи її більш ефективною та доступною для широкого спектру додатків.

Рисунок 1.1 – Концепція Federated Learning [6]

На даний час, до основних аспектів Federated Learning слід віднести наступні положення. На відміну від традиційного навчання, де всі дані збираються в одному центральному місці [7], FL дозволяє моделі навчатися на даних, які залишаються на пристроях. Кожен пристрій навчає модель локально на своїх даних, а потім надсилає лише оновлення моделі (наприклад, ваги або градієнти) на центральний сервер. Ці оновлення агрегуються для покращення загальної моделі. Оскільки вихідні дані не залишають локальний пристрій, FL забезпечує велику захист конфіденційності, що особливо важливо при роботі з чутливою інформацією, як-от медичні або фінансові дані. Ефективність використання даних ґрунтується на тому, що FL використовує дані з багатьох джерел, моделі потенційно можуть бути різноманітнішими і стійкішими до перенавчання.

Таким чином, Federated Learning є одним з найбільш перспективних напрямків у сфері AI та машинного навчання, що забезпечує баланс між використанням потужних алгоритмів та захистом конфіденційності даних. Незважаючи потенціал, FL стикається з низкою технічних викликів, таких як управління неоднорідними даними, зменшення витрат на передачу даних та забезпечення безпеки[8].

1.2 Використання розподіленого навчання для прогнозування часових рядів

З розвитком і збільшенням розгортання мобільних мереж 5-го покоління (5G) мережева інфраструктура стикається з проблемами обробки трафіку від різнорідних пристроїв, балансування навантаження та, загалом, надійності управління трафіком [9, 10]. Це спонукає використовувати методи прогнозування трафіку з низькою похибкою передбачення, щоб покращити якість послуг і забезпечити інтелектуальне управління ресурсами та оркестрування без погіршення продуктивності. Оскільки прогнозування базується на історичних даних, мета полягає в тому, щоб визначити в них просторово-часові шаблони даних для створення високоякісних прогнозів. Прогнозування часових рядів має широкий спектр застосувань у різних галузях промисловості та багато практичних застосувань. Прогнозування часових рядів є важливим завданням у широкому діапазоні проблем реального життя, таких як погода [11], споживання енергії [12], продажі [13] і прогноз трафіку [14]. В останні роки відбулося значне збільшення методів машинного навчання, які застосовуються до даних часових рядів. На відміну від класичних методів, таких як моделі на основі ARIMA, підходи глибокого навчання показали великий успіх у моделюванні часових рядів [15]. Більшість останніх підходів до прогнозування часових рядів стосуються моделей на основі нейронних архітектур, які можуть моделювати дані часових рядів, класифікуються за такими 3-ма категоріями [16]:

- повнозв'язані нейронні мережі (MLP);
- рекурентні (RNN, LSTM і GRU);
- згорткові мережі (CNN).

Як відомо, 3D CNN [17] є важливим інструментом у галузях, де необхідно розуміння 3-вимірних структур та відносин, пропонуючи глибші аналітичні можливості порівняно з традиційними методами. 3D CNN – це тип згорткової нейронної мережі, який розширює концепцію звичайних

згорткових нейронних мереж (CNN) для аналізу 3-вимірних даних. Традиційні CNN ефективні для аналізу 2-вимірних даних, таких як зображення . Натомість, 3D CNN призначені для обробки 3-вимірних даних, наприклад, медичних зображень (МРТ, КТ), відео (де 3-ій вимір – це час) або будь-яких інших об'ємних даних. У 3D CNN використовуються 3-имірні згорткові фільтри. Це означає, що фільтр переміщається не тільки по висоті та ширині вхідного об'єму, але і в глибину. Це дозволяє мережі вловлювати просторові відносини між об'єктами у 3-вимірному просторі. 3D CNN широко використовуються у різних областях, зокрема в медицині для аналізу тривимірних медичних зображень, в автоматизованих системах відеонагляду для розпізнавання дій та поведінки людей, у віртуальній та доповненій реальності для обробки тривимірних сцен. Завдяки здатності аналізувати 3-вимірні відносини, 3D CNN можуть бути ефективнішими у виявленні складних шаблонів та структур у даних, ніж традиційні 2-вимірні CNN. Одним з недоліків 3D CNN є те, що вони вимагають значно більших обчислювальних ресурсів та пам'яті, порівняно з 2D CNN, оскільки обсяг даних, який потрібно обробити, значно більший.

Стосовно FL, у [18] використовувалась модель LSTM, що пристосована застосовану до об’єднаних налаштувань для завдання прогнозування навантаження домогосподарств. Після створення глобальної моделі учасники виконують етап тонкого налаштування, використовуючи свої локальні дані для досягнення персоналізації. Об’єднаний LSTM оцінюється на наборі даних, що складається з 200 домогосподарств за допомогою алгоритму FedAvg [4]. В одній із найперших робіт із з FL [19], запропоновано FedGRU з алгоритмом FedAvg [4] для прогнозування кількості транспортних засобів у зоні. Перед запуском об’єднаного навчання суб’єкти-учасники об’єднуються в кластери за допомогою алгоритму K-Means, а після групування учасники навчають глобальну модель для кожного кластера. Експериментальна частина розглядає набір даних із 39000 транспортних датчиків у великих мегаполісах Каліфорнії, а FedGRU порівнюється з централізованими моделями, такими як

GRU та LSTM. Виходячи з повідомлених результатів, FedGRU перевершує централізовані моделі, коли увімкнений механізм кластеризації. Подальшим розвитком стала робота [20] з об'єднаною моделлю CNN-LSTM на основі механізму уваги для виявлення аномалій з використанням даних часового ряду з алгоритмом FedAvg. Крім того, вартість зв'язку під час FL мінімізується за допомогою алгоритму вибору на основі градієнтного стиснення. У [21] розглянкуто FedLSTM для прогнозування навантаження, подібно до [18], використовуючи набір даних, що складається зі 100 домогосподарств за допомогою алгоритму FedAvg [4]. FedLSTM інтегрує екзогенні дані, такі як інформація про погоду, і застосовує етап кластеризації серед об'єктів-учасників, враховуючи подібність ваги локальних моделей. Згідно з експериментальними результатами FedLSTM, розподілене навчання з локальним тонким налаштуванням дає найкращі результати, тоді як точність прогнозування майже еквівалентна індивідуальному, централізованому та федеративному навчанню. У [22] автори адаптували LSTM до об'єднаних налаштувань і використовували FedAvg і FedSGD [5] для завдання прогнозування навантаження, подібно до [21]. Перед навчанням власники даних виконали локальну попередню обробку, тобто учасники виконують локальне масштабування функцій. Таке масштабування призводить до неузгодженості під час навчання, що свідчить на користь глобального масштабування, яке призводить до найменшої помилки прогнозування. У [23] запропоновано декомпозицію часових рядів на сезонні, трендові та циклічні компоненти та ввели отримані вектори в модель GRU. Усі попередні роботи з об'єднаного прогнозування часових рядів не піднімають питання про дані, що не є IID (Independent And Identically Distributed, IID), і оцінюють створені глобальні моделі лише за допомогою алгоритму FedAvg. Незважаючи на останні досягнення в моделюванні даних часових рядів, більшість робіт стосується централізованого навчання, яке суттєво відрізняється від FL. Крім того, подібні роботи адаптують лише одну архітектуру нейронної мережі до об'єднаних налаштувань, наприклад, LSTM

[18, 21, 22] або GRU [19], і, отже, уявлення про різні об'єднані моделі, які застосовані до прогнозування часових рядів, не розглянуто. У роботі [24] запропонований алгоритм DeepTP і мережу на основі LSTM, яка моделює просторову залежність часових рядів і часових змін. Такий підхід значно перевершує інші моделі, наприклад, LSTM без удосконалень і традиційні методи, такі як ARIMA. Ще одним варіантом запропонованого підходу є кластеризована модель на основі LSTM для багатовимірного моделювання часових рядів [25], а у [26] розглянуто вдосконалення мережі LSTM з оцінкою стаціонарності (SLSTM), спеціально розробленої для вимірювань мережі мобільного зв'язку 5G.

1.3 Особливості прогнозування трафіку мобільного зв'язку

Незважаючи на нещодавні досягнення нейронних мереж, які застосовуються для прогнозування часових рядів, точність прогнозування обмежена кількістю спостережень. З іншого боку, точність прогнозування зосереджена на спостереженнях однієї сторони. Наприклад, у мережах мобільного зв'язку кожна базова станція (одна «сторона» в цьому сенсі) має свої власні унікальні характеристики, і, отже, немає гарантій, що модель, навчена на одній базовій станції, буде узагальнена для інших базових станцій, навіть у той самий регіон/територія. Прямим рішенням було б дозволити базовим станціям передавати свої дані в єдиний центр обробки даних і навчати модель машинного навчання за допомогою комбінованих спостережень. Однак передача даних і навчання моделі машинного навчання фрагментовані правилами та законами в усьому світі, такими як GDPR і HIPAA [27].

Крім того, конфіденційність бізнесу та проблеми конкуренції не дозволяють організаціям ділитися своїми даними з третьою стороною. Тобто, в даному випадку, кількість об'єктів-учасників обмежена 3-ма (в роботі під

ними варто розуміти базові станції мобільного зв'язку – рис. 1.2) і, отже, кластерне об'єднане навчання [19, 21] не може бути застосоване.

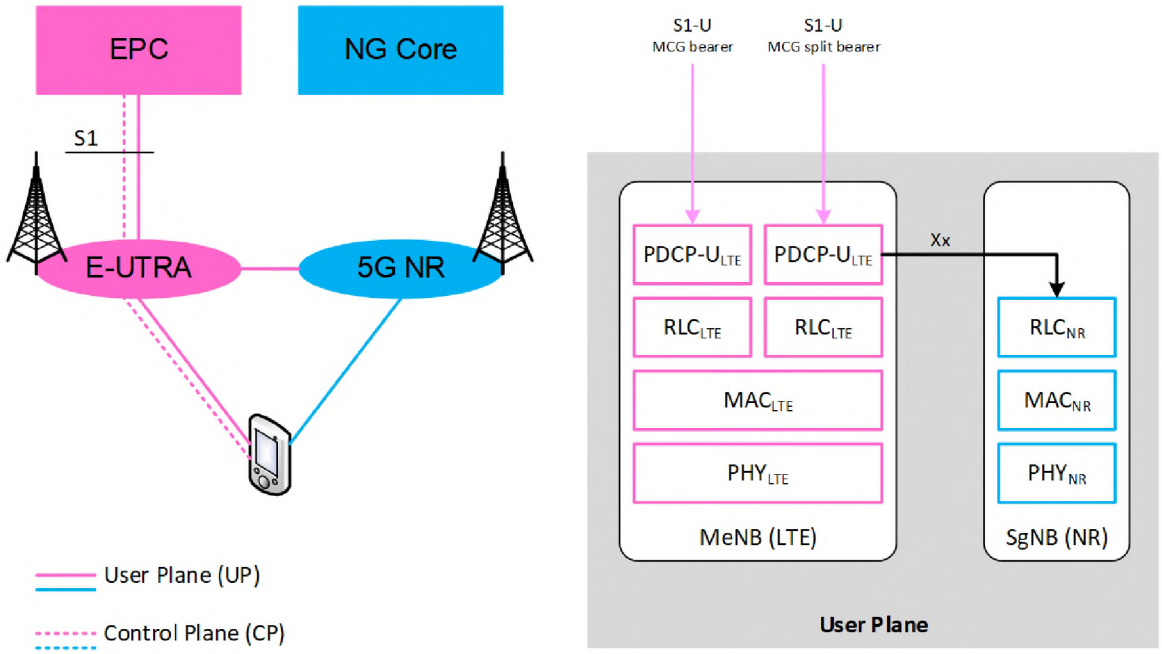


Рисунок 1.2 – Базова станція LTE

Щоб усунути зазначені обмеження та відповідати нормам і законам, а також зберегти конфіденційність користувачів, FL досліджувалось у контексті прогнозування часових рядів. Воно дозволяє кожній стороні співпрацювати в процесі навчання моделі без обміну або розкриття власних даних [4]. Після спільного навчання кінцева глобальна модель має потенціал для узагальнення, принаймні, для сторін, які брали участь у розподіленому навчанні. FL суттєво відрізняється від традиційного або централізованого навчання і має власні обмеження та особливості. За винятком відмінностей у процесі навчання між двома підходами, у централізованому навчанні передбачається, що оброблені дані є незалежними та однаково розподіленими IID (Independent And Identically Distributed, IID). Слід зауважити, що об'єднане налаштування було оцінено в багатьох завданнях машинного навчання, включаючи класифікацію зображень [4], передбачення наступного слова [28], аналіз клієнтів [29], спільну фільтрацію [30] і прогнозування часових рядів [18]. З іншого боку, припущення щодо даних

IID навряд чи витримає об'єднані налаштування через те, що суб'єкти-учасники можуть суттєво відрізнятися не лише розподілом даних, але й кількістю спостережень [31, 32]. При цьому, варто застосувати об'єднаний аналіз часових рядів необроблених даних LTE для моделювання просторово-часової залежності попиту на трафік. Прогнозування трафіку є ключовим завданням в епоху 5G через швидке зростання трафіку та зміну моделей з часом [33]. У контексті прогнозування трафіку [14] доцільно реалізовувати навчання об'єднаних моделей за наявності 4-ох категорій даних, що не є IID, наприклад, різні функції агрегації. В цілому, локальна попередня обробка може призвести до підвищення точності прогнозування. Під час процесу об'єднаного навчання кожен учасник виконує локальне навчання на своїх особистих даних, використовуючи вагові коефіцієнти моделі, які спільно використовують центральний сервер. Тоді, на відміну від традиційного машинного навчання, учасники діляться лише розрахованими параметрами моделі, таким чином уникаючи прямого витоку конфіденційності своїх локальних даних. Центральний сервер (часто також званий сервером агрегації) потім збирає ці оновлення параметрів і об'єднає їх за допомогою функції агрегації.

Результат агрегації відповідає новій глобальній моделі, яка потім буде передана учасникам для наступного раунду. Огляд одного раунду об'єднаного процесу наведено на рис. 1.3. Незважаючи на те, що розподілене навчання обіцяє високоякісну генерацію моделі зі збереженням конфіденційності, існують дослідження щодо впливу даних, які не є IID, на точність моделі [31, 34, 5]. Зокрема, спостереження з базових станцій містять більшість категорій даних, не пов'язаних з IID, тобто: атрибути, мітки, кількість і часові перекоси [34, 5]. Таким чином, доцільно використовувати історичні дані для надання однокрокових прогнозів, значення атрибутів і міток представлені як у навчальних функціях, так і в цільових значеннях. Якщо припустити локальний розподіл $P(x_i, y_i) = P(x_i | y_i) \cdot P(y_i)$, нерівність розподілу ознак і міток полягає в тому, що $P(x_i)$ і $P(y_i)$ відрізняються між об'єктами-

учасниками. Кількісний дисбаланс означає, що кількість спостережень різна для користувачів.

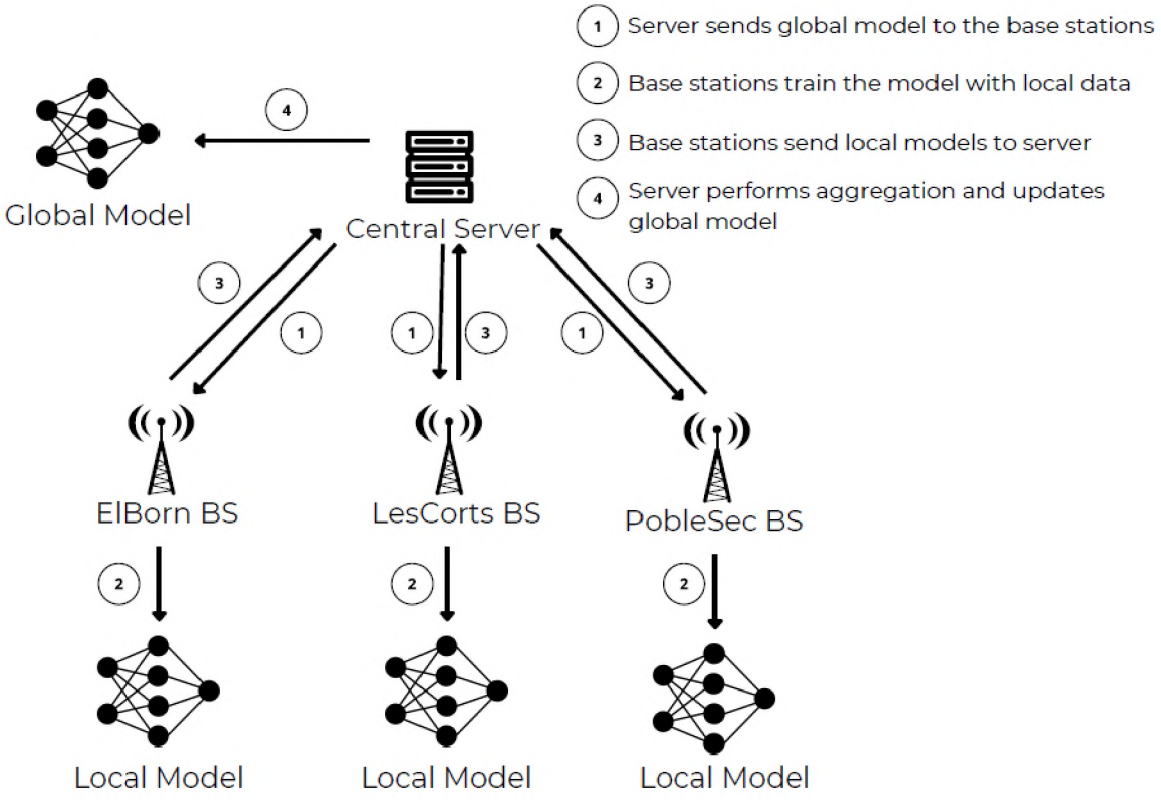
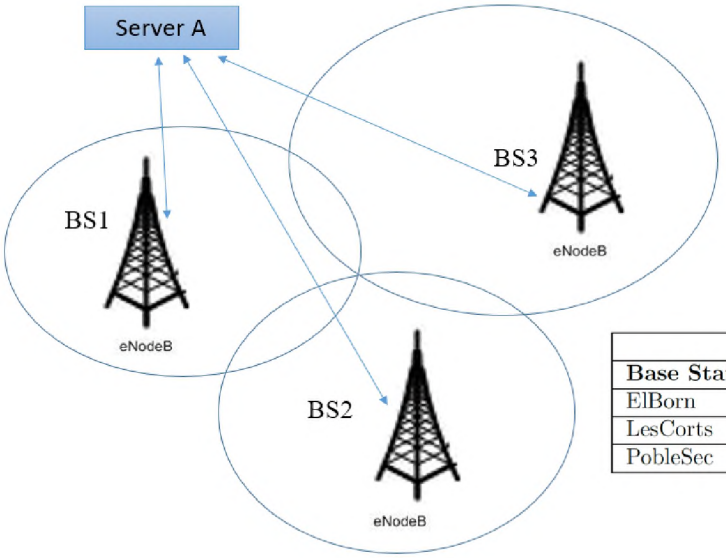


Рисунок 1.3 – Модель реалізації FL

Нарешті, часовий дисбаланс означає, що часові кроки спостережень можуть змінюватися або розподіл ознак може змінюватися з часом (додаток А, рис. А.1). Таким чином, для формування датасету доцільно використовувати [G1], що являє собою 3 набора даних від 3-ох окремих базових станцій, кожен з яких містить багатовимірний часовий ряд, тобто кілька залежних від часу змінних, наприклад, кількість часових ідентифікаторів радіомережі (RNTI), кількість виділених блоків ресурсів (RB) та ін. Як наслідок, доцільно реалізувати, модель глибокого навчання нейронної мережі для прогнозування часових рядів за допомогою FL шляхом об'єднаного навчання та передбачати значення 5-ти окремих вимірювань для наступного часового кроку, враховуючи вимірювання безпосередньо попередніх 10 часових кроків. Серед 5-и вимірювань завдання

зосереджується на розмірах транспортних блоків висхідної (UpLink) та низхідної (DownLink) ліній зв'язку, які мають вирішальне значення для визначення обсягу та напрямку потоків. Незважаючи на зріз на вказаному максимумі (див. додаток А, рис. А.1), легко помітити, що розподіл між базовими станціями різний. Наприклад, розподіл у першій базовій станції (BS1) є позитивно спотвореним. У другій базовій станції (BS2) розподіл наближається до нормального, а в третій базовій станції (BS3) – бімодальний. На рис 1.4 наведено асиметрію та ексцес для вимірювань висхідної та низхідної лінії зв'язку на базову станцію без максимального скорочення.



	Skewness		Kurtosis	
Base Station	Up	Down	Up	Down
ElBorn	8.3262	2.3513	132.0776	5.8289
LesCorts	12.2926	0.4804	321.1801	-0.5705
PobleSec	8.0330	3.5667	101.8521	23.8463

Рисунок 1.4 – Статистична оцінка трафіку

В даному контексті термін Skewness (асиметрія) визначає ступінь асиметрії розподілу щодо його середнього значення. Вона може бути позитивною або негативною, залежно від того, чи зміщений розподіл праворуч або ліворуч. Термін Kurtosis (Ексцес) відноситься до гостроти піку розподілу даних. Високий ексцес вказує на гостріший пік і більш «важкі» хвости розподілу, тоді як низький ексцес свідчить про більш плоский розподіл. Асиметрія та ексцес для вимірювань висхідної лінії зв'язку перевищують 1 і 3 на базову станцію відповідно. Таким чином, розподіли для

вимірювань вгору позитивно спотворені, а наявність викидів є високоймовірною. Щодо значень низхідної лінії зв'язку, у BS2 асиметрія становить [-0,5, 0,5], а ексцес становить < 3, що вказує на те, що розподіл є майже симетричним, а ймовірність викидів низька. Решта базових станцій дотримуються спостережень із вимірювань висхідної лінії зв'язку, тобто позитивної асиметрії та наявності викидів. Крім даних і нерівності розподілу цілей, відповідні атрибути містять часові характеристики. Іншими словами, оскільки ми маємо справу з уявленнями часових рядів, розподілене навчання створює ще одну проблему, а саме часову диспропорцію [34]. По-перше, спостереження збираються в різні періоди часу, тобто березень/квітень 2018 р. – січень 2019 р. та лютий/березень 2018 р. на базову станцію відповідно. По-друге, розподіл даних змінюється з часом, наприклад, зміна розподілу на базовій станції BS1 для вимірювань вгору та вниз – див. додаток А, рис. А.1. Спостерігається, що (принаймні) три з семи днів дотримуються різних моделей щодо вимірювань вгору та вниз, що вказує на локальні скроневий дисбаланс. Наприклад, шаблон попередніх днів може не відповідати шаблону наступних днів. Зверніть увагу, що лінійні графіки відповідають середнім значенням за годину з 95 % довірчим інтервалом, оскільки ми виконали операцію групування за годинами. Таким чином, доцільно дослідити ефективність FL, розглядаючи комбінацію категорій даних, не пов'язаних з IID, які ще належить дослідити. При цьому можна використовувати [31], у якій категорії, що не належать до IID, розглядаються незалежно від наборів зображень і табличних даних для завдання класифікації. Одним із найважливіших кроків у FL є функція агрегації, особливо при роботі з даними, не пов'язаними з IID.

В цілому, загального підхід до FL на необроблених даних часових рядів базової станції передбачає, що третя сторона несе відповідальність за збір оновлень ваги за об'єднаний раунд, узагальнює результати за допомогою алгоритму агрегації та розподіляє агреговані дані між учасниками для локальних тренувань і оновлення ваги.

Висновки до розділу 1

На даний час, нейронні мережі застосовуються для централізованого налаштування з метою прогнозування часових рядів. Методи попередньої обробки можуть призвести до значного погіршення помилок прогнозування та мати більший вплив на остаточний прогноз, ніж сучасніші алгоритми агрегації для даних, які не є IID.

Як наслідок, для оцінки об'єднаних часових рядів доцільно використовувати FL. Для цього можна застосувати різні архітектури нейронних мереж. Найбільш перспективними виглядають RNN і CNN. Більшість із запропонованих агрегаторів було оцінено на різних предметних завданнях, таких як класифікація зображень, аналіз настроїв і передбачення наступного символу/слова.

Навчання потрібно проводити за трьома різними налаштуваннями: індивідуальне навчання (кожна базова станція виконує навчання та оцінку, використовуючи лише локальні спостереження), централізоване навчання та об'єднане навчання.

РОЗДІЛ 2
СИНТЕЗ МОДЕЛІ ГЛИБОКОГО НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ ПРОГНОЗУВАННЯ ЗАВАНТАЖЕНОСТІ СТАНЦІЙ МОБІЛЬНОГО ЗВ'ЯЗКУ

2.1 Попередня обробка даних

Традиційні підходи ML [35] не передбачають участі кількох клієнтів на етапі навчання. Наприклад, учасник може виконувати навчання, використовуючи лише локальні дані (індивідуальне навчання), або традиційні підходи з кількома користувачами, що вимагають передачі локальних даних (централізоване навчання). Навпаки, об'єднане навчання вимагає активної участі, оскільки глобальна модель створюється шляхом збору та агрегування локальних моделей за об'єднаний раунд. Загалом, дані часових рядів збираються автоматично з датчиків або механізмів реєстрації. Згідно [6], існує диспропорції розподілу між 3-ма базовими станціями (додаток А, рис. А.2) відносно розміру транспортного блоку DownLink (рис. 2.1), який відображає атрибут і цільову асиметрію.

Наприклад, базові станції зберігають інформацію про мережеві вимірювання або будинки обладнані розумними лічильниками для вимірювання споживання електроенергії. Цілком природно, що дані містять надлишкову інформацію, помилки або незвичні сплески. Отже, методи попередньої обробки є вирішальними для подальшого навчального завдання. Попередня обробка відбувається перед етапом навчання, і, з точки зору ML є незамінною, оскільки застосовані перетворення можуть призвести до підвищення точності прогнозування. У контексті часових рядів можна виділити кілька кроків.

1. Очищення даних. Метою цього кроку є обробка відсутніх або пошкоджених даних, виявлення викидів і керування ними. Відсутні дані можна обробити, видаливши їх, використовуючи техніку перетворення,

наприклад, перетворення в константу, заповнивши їх репрезентативним значенням, наприклад середнім, або оцінивши їх за допомогою алгоритмічного підходу.

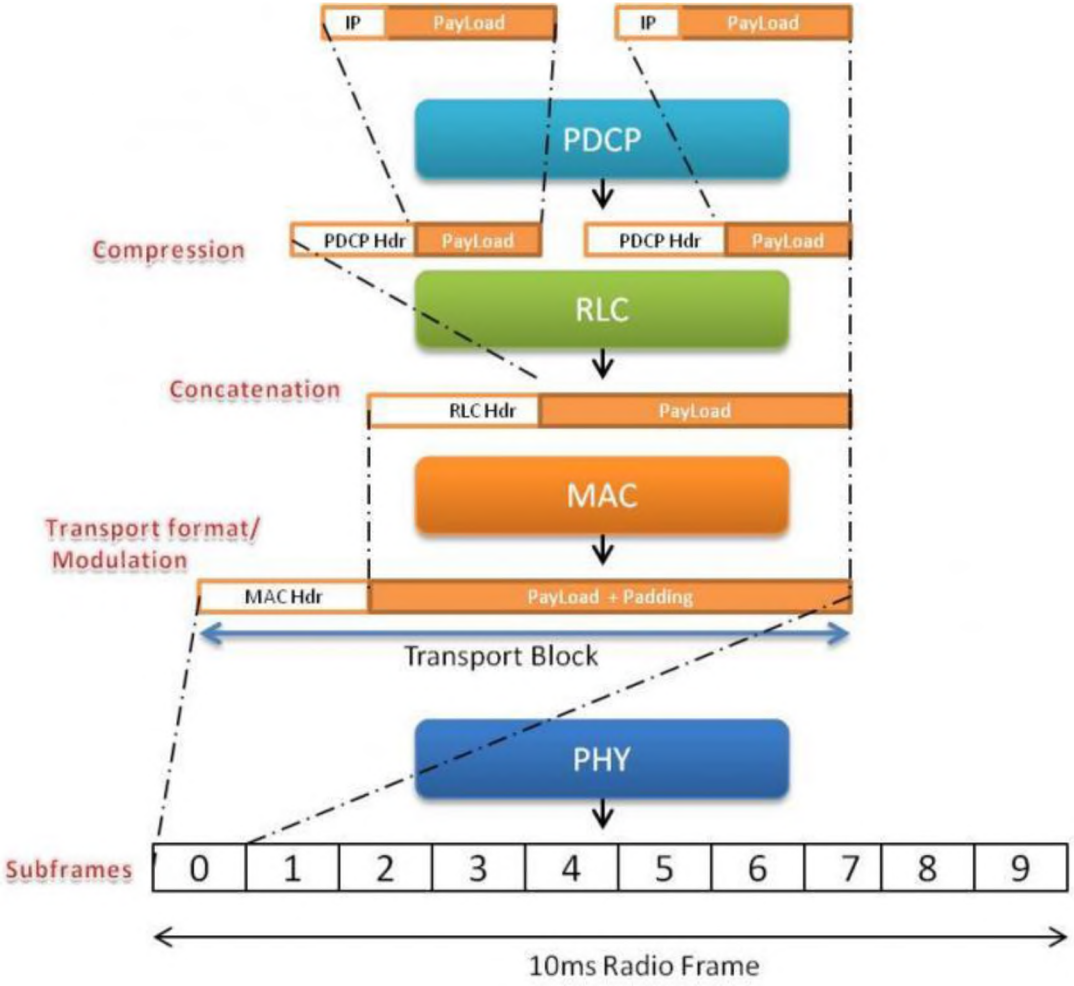


Рисунок 2.1 – Транспортний блоку DownLink базової станції

Ще один підхід – використання простої техніки та перетворення відсутніх значень на нулі, тобто, нульове перетворення, оскільки просте видалення порушує концепцію безперервних даних. Крім того, часові ряди змінюються з часом, тобто приписування їм константи, такої як середнє, може не бути репрезентативним під час логічного висновку, а їх оцінка може зайняти багато часу та енергії. Оцінка впливу додаткових підходів до очищення даних потребує додаткових досліджень.

Викиди стосуються значень, які значно відрізняються від більшості спостережень у наборі даних, і можуть негативно вплинути на ефективність

навчання. Всі базові станції на вимірюваннях UpLink містять екстремальні викиди (див. додаток А, рис. А1). Подібно до обробки відсутніх значень, існує кілька підходів для виявлення викидів, наприклад, статистичні моделі або моделі на основі навчання. Однак, ці методи вимагають кількох випробувань, що, знову ж таки, потребує часу та енергії. Що стосується викидів, то можна також використовувати загальновживаний і простий підхід це техніка заміни на порогові значення. Тобто, у цій техніці немає етапу виявлення, але значення, які падають нижче та вище вказаних порогів, перетворюються на значення заданих нижнього та верхнього порогів.

2. Формування датасету [36]. Цей крок застосовує розподіл набору даних на три складові. Навчальна вибірка використовується для навчання моделі. Перевірочна вибірка використовується для обчислення корисних статистичних даних щодо ефективності навчання та налаштування гіперпараметрів моделі. Тестова вибірка використовується після навчання для оцінки точності моделі на даних, які раніше модель не бачила. При цьому доцільно використовувати найбільш поширений варіант розподілу навчальної та тестової вибірок для кожної базової станції. Навчальний набір даних розділено на 80 % для навчання та 20 % для перевірочної вибірки, тоді як набір для тестування залишається недоторканим.

3. Масштабування функцій. Завдання стосується багатовимірних часових рядів, що означає, що на кожному кроці часу є кілька різних атрибутів з різними масштабами (рис. 2.1). На відміну від середнього значення медіана практично не чутлива до викидів та асиметрії розподілу. Тому її оптимально використовувати як «нульове» значення під час центрування. Щоб усунути вплив діапазонів значень а також викидів (рис. 2.2), доцільно використовувати нормалізацію Min-Max і перетворюємо всі змінні на значення між [0, 1].

Це дозволяє отримати низку переваг. Приведення всіх вхідних змінних до одного масштабу може значно прискорити збіжність алгоритму навчання. Це означає, що нейронна мережа зможе навчитися швидше. Менші значення

ваг можуть допомогти уникнути проблем, пов'язаних із затуханням або вибухом градієнтів під час зворотного поширення помилки. Нормалізація може зменшити вплив викидів, оскільки всі дані будуть знаходитися в обмеженому діапазоні.

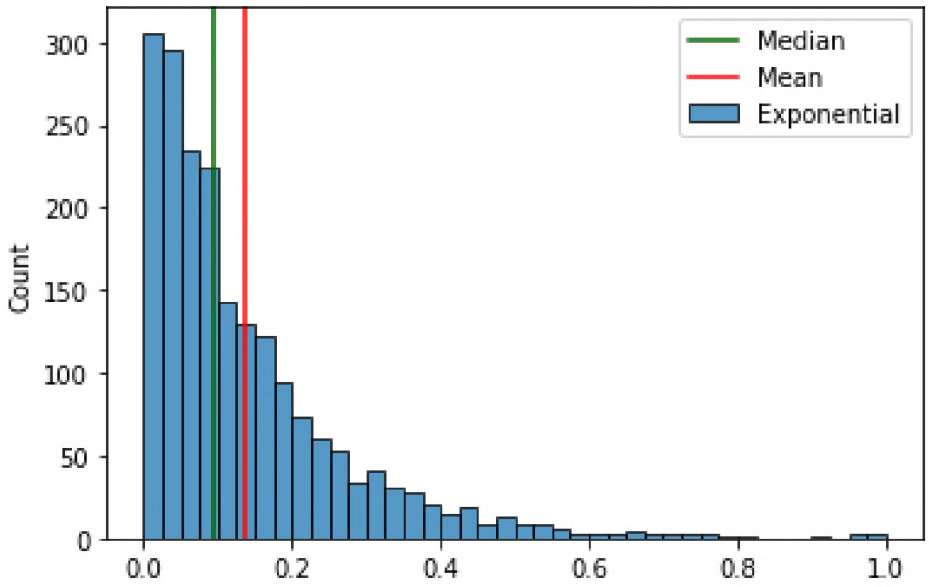


Рисунок 2.1 – Відмінності між медіаною та середнім арифметичним значенням для експоненційного розподілу

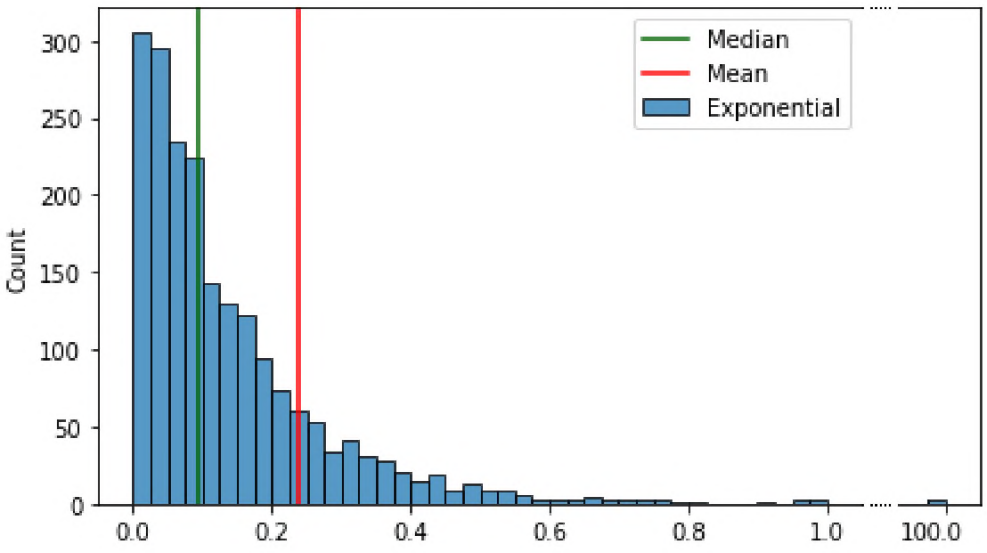


Рисунок 2.2 – Відмінності між медіаною та середнім арифметичним значенням для експоненційного розподілу при додаванні викиду

Всі змінні будуть мати однаковий вплив на навчання, уникаючи ситуацій, коли нейронна мережа надмірно «фокусується» на змінних із

великими або малими величинами. Для деяких типів нейронних мереж, таких як ті, що використовуються в комп'ютерному зорі, нормалізація даних може покращити точність прогнозування. Багато активаційні функції (наприклад, сигмоїд) працюють найкраще, коли вхідні дані знаходяться у невеликому діапазоні. Метод Min-Max легко реалізувати і не вимагає складних підрахунків або додаткових параметрів. Таким чином, перетворення для кожного вимірювання в кожній змінній має вигляд:

$$x' = \frac{x - \min}{\max - \min}, \tag{2.1}$$

де x – спостережуване вимірювання,

min і max – мінімальне та максимальне значення, що спостерігаються у змінній відповідно.

4. Представлення часових рядів. Вихідні дані, які відносяться до варіацій за часовий крок, повинні бути перетворені на представлення часових рядів із заданим вікном T, а також для введення ознак X і цільових змінних Y. Іншими словами, надані вимірювання об’єднуються для створення набору даних ковзних вікон, що керуються від T. В якості допущення приймаємо $T = 10$ і перетворюємо набір даних таким чином, щоб цілі Y відповідали наступному часовому кроку. Технічно, припускаючи 100 спостережень за 11 змінними та використовуючи $T = 10$, отримуємо набір даних, що складається з 90 екземплярів, де кожен екземпляр відповідає масиву, що містить 11 × 10 (варіанти × T) вимірювань для вхідних ознак X.

У сфері аналізу часових рядів однією з ключових перешкод є явище автокореляції (рис. 2.3). Часто буває, що помилка, розрахована мережею, здається незначною, однак реальність показує невдоволені прогнози. Для ефективнішої роботи з часовими рядами застосовують такі метрики, як середньоквадратична помилка (MSE) та середня абсолютна помилка (MAE), які вимірюють відхилення між прогнозованими та фактичними значеннями. Ключова мета полягає у тому, щоб добитися, аби кореляційний графік максимально відповідав еталонному (рис. 2.4).

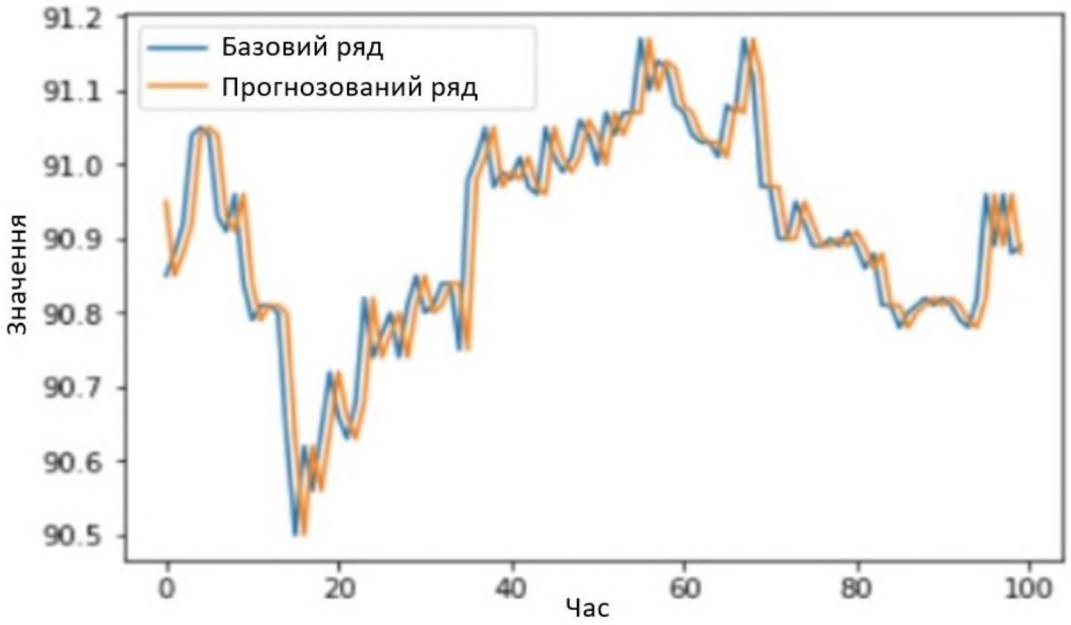


Рисунок 2.3 – Автокореляція

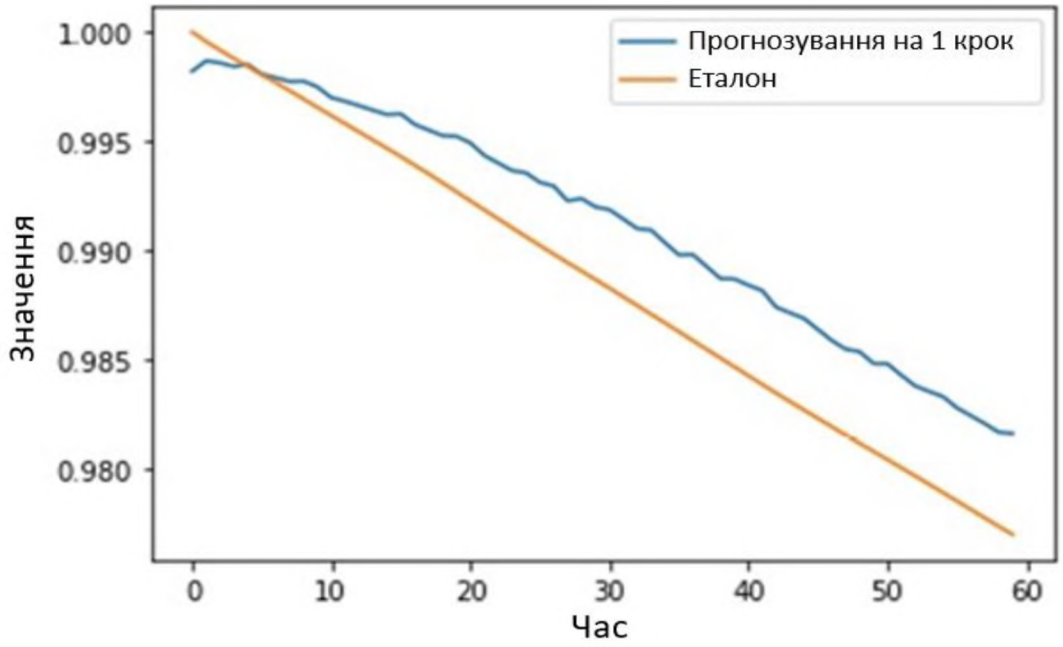


Рисунок 2.4 – Кореляція

Використання нейронних мереж може значно покращити точність прогнозування, наприклад, в такій галузі, як прогнозування ринкових курсів акцій, демонструючи вищу точність порівняно з іншими методами, такими як аналіз трендів або ARIMA. Остання модель часто використовується для прогнозування майбутніх значень у часовому ряді на основі історичних

даних. Це особливо корисно в фінансовому моделюванні, економетриці, погодних прогнозах тощо. Підбір моделі ARIMA вимагає визначення оптимальних значень для параметрів AR, I та MA, які часто визначаються за допомогою статистичних методів, таких як критерій Акаіке (AIC) або Байєсівський інформаційний критерій (BIC).

Для роботи з нелінійними та сезонними даними, ARIMA може бути розширена до SARIMA (Сезонна ARIMA) або ARIMAX, де «X» відноситься до екзогенних змінних. Авторегресія (AR) – складова моделі ARIMA передбачає, що поточне значення часового ряду може бути описане як лінійна комбінація попередніх значень часового ряду. Це основа «авторегресійної» частини моделі. Інтегрованість (I) – частина моделі відноситься до різниць між послідовними спостереженнями, які використовуються для досягнення стаціонарності часового ряду. Стаціонарний часовий ряд має постійні статистичні характеристики, такі як середнє значення та дисперсія, незалежно від часу. Ковзаюча середня (MA) – частина моделі описує поточне значення часового ряду як лінійну комбінацію термінів помилок (або шуму) попередніх прогнозів. Це дозволяє моделі враховувати випадкові зміни в часовому ряді.

Конвеєр попередньої обробки є невід'ємним і незамінним етапом машинного навчання. Очищення даних і масштабування функцій у об'єднаних налаштуваннях можуть бути більш складними, ніж індивідуальні та централізовані підходи. Наприклад, центральний сервер збирає дані від різних об'єктів і, враховуючи високу доступність, виконує обчислення, застосовані до об'єднаних наборів даних.

Однак, інтегроване навчання створює низку ускладнень. По-перше, реалізацію випробувань перетворення даних не можна виконати в реальних умовах, оскільки FL вимагає активної участі. По-друге, локальне приписування відсутніх значень або викидів із константою, такою як середнє, призводить до наступної неузгодженості: кожен учасник має різні спостереження, і, отже, обчислена статистика природним чином

відрізняється між об'єктами, що, у свою чергу, може призвести до введення шуму в машинне навчання алгоритм. По-третє, локальна оцінка відсутніх значень призводить не тільки до додаткових витрат, але і до такої ж неузгодженості з локальним умовним обчисленням. Спрощені підходи, такі як нульове перетворення, настил і укриття, які вибрано в цій роботі, не передбачають додаткового зв'язку, а перетворення призводять до узгоджених зразків. Подібним чином масштабування ознак вимагає обчислення мінімуму та максимуму (Min-Max) або інші методи, такі як стандартизація, потребують середнього значення та стандартного відхилення. Одним з підходів є локальне обчислення необхідної статистики та масштабування відповідних значень. Однак, такий підхід призводить до неузгодженості. Наприклад, розглянемо двох клієнтів, які мають максимум, відповідно, 500 і 1000. Виконання перетворення Min-Max локально призводить до того, що перший клієнт перетворює значення 500 на 1, а другий клієнт перетворює значення 1000 на 1. Тепер алгоритм машинного навчання під час навчання спостерігає значення 1 двічі, можливо, з різними/суперечливими цілями. На відміну від очищення даних, на етапі масштабування функції майже неминуче обчислення глобальної змінної над локальними значеннями. На практиці учасники повинні лише передавати свої мінімальні та максимальні значення третій стороні, яка оголошує глобальні мінімум і максимум. Якщо конфіденційність має велике значення, слід інтегрувати протокол збереження конфіденційності для розрахунку відповідних значень. Таким чином, дана ситуація вимагає додаткових досліджень.

В цілому, конвеєр попередньої обробки розроблений таким чином, щоб обробляти три різні налаштування навчання. У централізованому навчанні конвеєр належить центральному серверу, тоді як в індивідуальних і об'єднаних налаштуваннях попередня обробка передається на сторону клієнта. Точніше, після збору деяких даних власник передає спостереження у конвеєр попередньої обробки. Потім виконується обробка відсутніх значень, і набір даних розбивається на набори для

навчання та перевірки. Після розподілу до навчального набору застосовується обробка викидів. Викиди не слід перетворювати в наборі перевірки, оскільки це стосується зразків, які не надаються під час навчання. Після обробки викидів функції масштабуються, і, нарешті, дані представляються як часові ряди за допомогою ковзного вікна *T*. Після попередньої обробки дані готові до передачі в алгоритм машинного навчання для навчання. Зауважимо, що для обробки відсутніх значень і викидів вибрано нульове перетворення, нижнє та верхнє значення відповідно. На практиці, ці методи не вимагають спілкування між учасниками. Для масштабування функції вибирається техніка Min-Max, яка вимагає пошуку глобального мінімуму та максимуму. В даному випадку, припускаємо, що учасники передають свій локальний мінімум і максимум на змінну третій стороні, яка оголошує глобальні значення. Варто пам'ятати, що методи попередньої обробки сильно впливають на продуктивність моделі. Однак, наскільки нам відомо, для цього етапу було проведено обмежені дослідження щодо FL, тобто того, як різні об'єкти мають обробляти та перетворювати свої дані. Подібне спостереження зроблено в [22], де автори використовували локальну попередню обробку та масштабували дані окремо для кожного клієнта. Індивідуальна попередня обробка призводить до непослідовних перетворень серед учасників.

2.2 Формування моделі прогнозування завантаженості станцій мобільного зв'язку за допомогою розподіленого навчання

Розглянемо більш детально особливості індивідуального навчання. Тобто умови, коли одна сторона проводить спостереження та проводить локальне навчання. Позначаємо через $X_t = \{x_0, \ldots, x_d\}$ дані, що отримані шляхом вимірювання на кроці часу *t*, де *d* є кількістю вимірювань (змінних). Під час індивідуального навчання весь процес виконується внутрішньо без

передачі даних. Враховуючи часовий ряд вікна T на кроці часу t, $X_t^* = \{X_{t-T+1}, \ldots, X_t\}$, мета полягає в тому, щоб передбачити вимірювання для наступного кроку часу, $\hat{y}_{t+1}$ на основі T минулого спостереження. Для цього потрібно побудувати модель, яка може узагальнювати невидимі майбутні ряди, використовуючи весь набір даних вимірювань:

$$D_{ind} = \sum_{i=1}^{m} X_t^* , \tag{2.2}$$

де m – кількість часових рядів, що містять вікно T.

У централізованому навчанні зібрані вимірювання передаються третій стороні або центру обробки даних, і, отже, сторона, яка тренує модель машинного навчання, використовує комбінацію спостережених вимірювань. Як наслідок, потрібно побудувати модель, яка може узагальнити майбутні ряди, принаймні, для n учасників, враховуючи об'єднані дані від них:

$$D_{cen} = \sum_{i=1}^{n} D_{ind} . \tag{2.3}$$

Інтегроване навчання можна класифікувати як проміжне налаштування індивідуального та централізованого навчання. Точніше, є n учасників, які бажають спільно побудувати модель прогнозування, яка може узагальнювати їхні майбутні спостереження. Кожен учасник $p \in n$ зберігає власний часовий ряд і тренує модель навчання локально протягом обмеженої кількості епох.

Потім учасники передають свої вагові коефіцієнти, отримані на основі місцевих спостережень, агрегатору, і генерується усереднена модель. Процес триває до тих пір, поки усереднена глобальна модель не зможе узагальнити спостереження n учасників. Отже, у федеративному навчанні кожна сутність-учасник оптимізує модель на основі локальних даних, що еквівалентно формулюванню проблеми індивідуального навчання, тоді як створена глобальна модель має можливість узагальнення на n сутностей, що є кінцевою метою машинного навчання.

На відміну від централізованого та індивідуального навчання, FL може

потенційно використовувати гарантії конфіденційності індивідуального навчання, оскільки учасники не діляться своїми необробленими даними та використовують прогнози *n* локальних моделей для створення глобальної моделі, яка може призвести до узагальнення. При цьомуFL, саме по собі, не забезпечує суворих гарантій конфіденційності [37, 38]. Наприклад, зловмисник може повернути вагові коефіцієнти моделі та ідентифікувати спостереження з високою ймовірністю, наприклад, використовуючи градієнтну інверсію [39] або атаку на приналежність [40]. Оцінка гарантій конфіденційності FL, відповідно, виходить за межі цієї роботи та залишається для майбутніх досліджень. Щоб змоделювати реальний сценарій для завдання прогнозування трафіку, пропонується структура, яку можна налаштувати та розширити до узагальненого прогнозування часових рядів. Основна перевага запропонованого підходу полягає у підтримці переходу від інтегрованого навчання до індивідуального або централізованого. Рис. 2.5 ілюструє основні компоненти моделі.

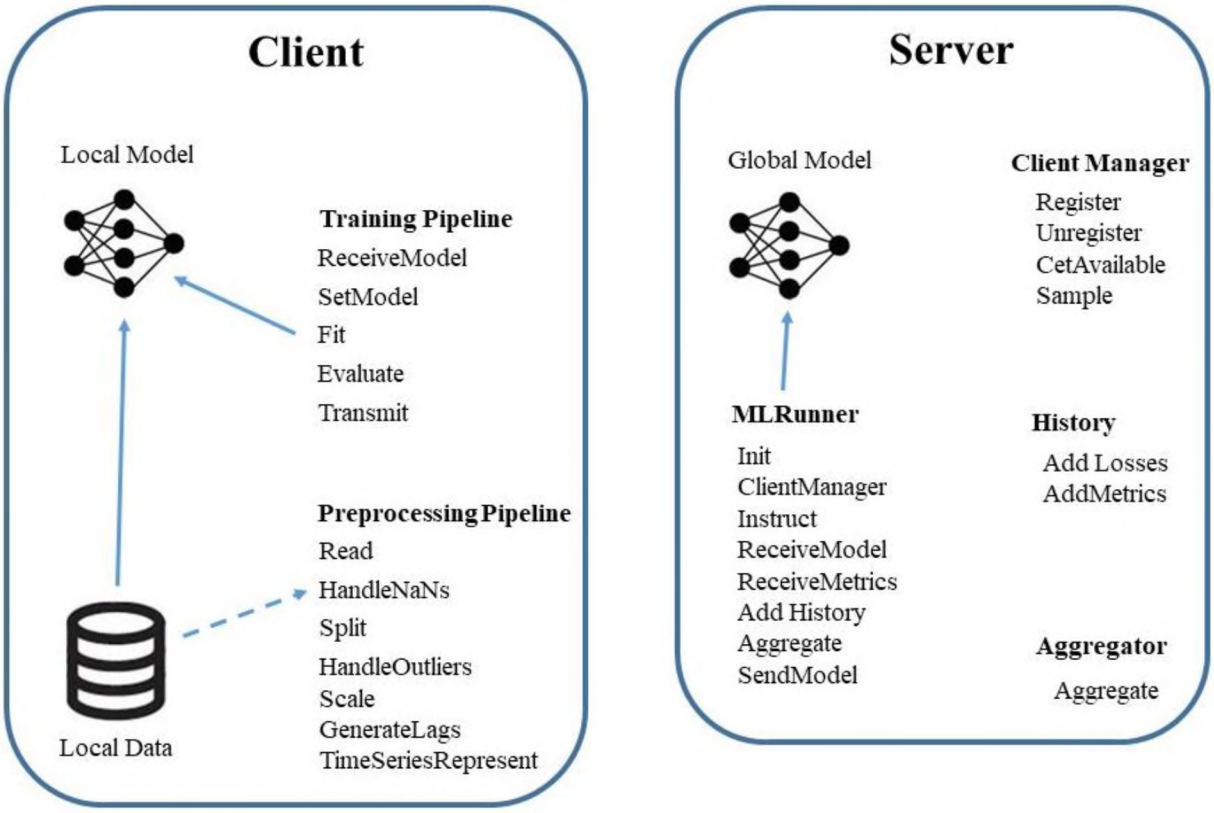


Рисунок 2.5 – Структура пропонованої моделі прогнозування часових рядів

Вона складається з двох основних сутностей, Сервера та Клієнта. Сервер відповідає за вибір доступних клієнтів для об'єднаного раунду, реєстрацію історичної інформації, такої як показники втрат і оцінки для кожного клієнта, а також агрегування та генерування параметрів глобальної моделі. Клієнт представляє окремі організації, які беруть участь у об'єднаних обчисленнях. На вищому рівні кожному учаснику надаються методи, які застосовують дві основні операції для машинного навчання:

– попередня обробка, яка виконується перед інтегрованим навчанням;

– конвеєр навчання, який дозволяє учасникам виконувати локальне навчання та оцінювати локальні (глобальна) модель.

Синтезована архітектура базується на системі об'єднаного навчання Flower [41] і використовує PyTorch [42] як базову бібліотеку глибокого навчання. Як відомо, PyTorch надає гнучку та інтуїтивну робочу платформу, яка дозволяє легко експериментувати з архітектурами нейронних мереж. Її динамічне створення графіків (dynamic computation graphs) дозволяє змінювати архітектуру нейронної мережі «на льоту» та спрощує процес дебагінгу. PyTorch має велику та активну спільноту користувачів і розробників, що забезпечує хорошу підтримку, багато навчальних матеріалів і готових до використання модулів. PyTorch глибоко інтегрований з Python, що робить його легко використовувати для тих, хто вже знайомий з цією мовою програмування. Це також дозволяє легко використовувати бібліотеки Python, такі як NumPy. PyTorch ефективно використовує апаратні ресурси, що дозволяє швидко тренувати складні моделі, особливо коли використовуються GPU. Вони мають сотні або навіть тисячі ядер, що дозволяє обробляти велику кількість операцій одночасно. Це гарно підходить для нейронних мереж, де потрібно одночасно проводити велику кількість дрібних, але нескладних математичних обчислень. Завдяки своїм паралельним обчислювальним можливостям, GPU може значно прискорити навчання і висновки нейронних мереж, особливо у великих масштабах. Нейронні мережі часто потребують обробки великих обсягів даних. GPU,

завдяки своїй високій пропускній здатності пам'яті та швидкості обчислень, забезпечують ефективну обробку цих датасетів. Більшість сучасних фреймворків для глибинного навчання, таких як TensorFlow, PyTorch, і CUDA, оптимізовані для використання з GPU, що забезпечує покращену інтеграцію та легкість використання. На великих масштабах GPU може бути більш енергоефективним порівняно з CPU, оскільки вони можуть виконувати більше обчислень на ват енергії. Оскільки GPU були спочатку розроблені для обробки графіки, вони є особливо ефективними у задачах, пов'язаних з обробкою зображень та відео, які є важливими для багатьох застосувань нейронних мереж.

PyTorch легко інтегрується з багатьма іншими бібліотеками машинного навчання та аналізу даних, такими як Scikit-Learn, Pandas, та ін. Scikit-Learn є бібліотекою з відкритим кодом, що дозволяє йому використовуватися і модифікуватися вільно. Вона надає багатий набір інструментів для різних задач машинного навчання, включаючи класифікацію, регресію, кластеризацію, відбір змінних, обробку даних та багато іншого. Scikit-Learn славиться своїм чистим, зрозумілим та зручним у використанні API. Це робить її доступною навіть для тих, хто лише починає свій шлях у машинному навчанні. Має велику і активну спільноту користувачів і розробників, а також відмінну документацію, яка полегшує вивчення та застосування бібліотеки. Scikit-Learn легко інтегрується з іншими науковими та аналітичними бібліотеками Python, такими як NumPy, SciPy та Pandas. Хоча Scikit-Learn не є найшвидшою бібліотекою для машинного навчання, вона оптимізована для високої продуктивності у багатьох загальноприйнятих сценаріях.

PyTorch є популярним вибором як у наукових дослідженнях, так і в промисловому застосуванні, що свідчить про його гнучкість і масштабованість. PyTorch має добре структуровану документацію та багато навчальних ресурсів, що робить його доступним для новачків у галузі глибокого навчання. Крім цього в модель інтегрується кілька сучасних

алгоритмів агрегації, які дозволяють оцінювати їх застосовність і точність прогнозування на даних, не пов'язаних з IID, використовуючи пакет NumPy [43]. Сформований код можна буде легко налаштувати та розширити за допомогою додаткових моделей, реалізованих за допомогою популярної бібліотеки PyTorch та/або додаткових алгоритмів агрегації за допомогою NumPy. Основна особливість NumPy – підтримка багатовимірних масивів (ndarray), які є ефективнішими та швидшими за стандартні списки Python для обробки великих даних. NumPy дозволяє виконувати математичні операції на цілих масивах без необхідності використання циклів. Бібліотека містить широкий спектр функцій для операцій з лінійною алгеброю, таких як векторні та матричні операції, розклад матриць, обертання та інші. NumPy може обробляти складні числа, що є необхідним у деяких наукових та інженерних обчисленнях. NumPy легко інтегрується з багатьма іншими бібліотеками Python, включаючи Scikit-Learn, SciPy, Pandas, Matplotlib, і багато інших. Включає велику колекцію математичних функцій, включаючи статистичні операції, тригонометричні функції, логарифмічні/експоненційні функції тощо. NumPy дозволяє ефективно обробляти великі набори даних, здійснюючи операції сортування, фільтрації, перетворення та інші. Масиви NumPy зберігають дані у спеціалізованому форматі, що є більш ефективним за звичайні списки Python, особливо при роботі з великими обсягами даних. NumPy може використовуватися для розпаралелених обчислень, що покращує продуктивність при роботі з великими даними та складними обчислювальними задачами.

2.3 Застосування розподіленого навчання

Навчання на основі FL (рис. 2.6) [44] потребує деяких модифікацій, принаймні для створення глобальної моделі, порівняно з традиційними підходами.

Розглянемо більш детально введені сутності, які беруть участь у обчисленні: Сервер і Клієнт. На вищому рівні сервер спочатку робить вибірку доступних клієнтів і передає поточну глобальну модель. Вибіркові клієнти отримують глобальну модель і проводять локальне навчання, використовуючи свої дані. Після локального навчання оновлена модель передається назад на сервер разом із історичними показниками, такими як показники втрат і оцінки. Після отримання всіх локальних моделей сервер здійснює їх агрегацію і оновлює стан глобальної моделі. При цьому, всі клієнти представлені у введеній структурі Клієнт.

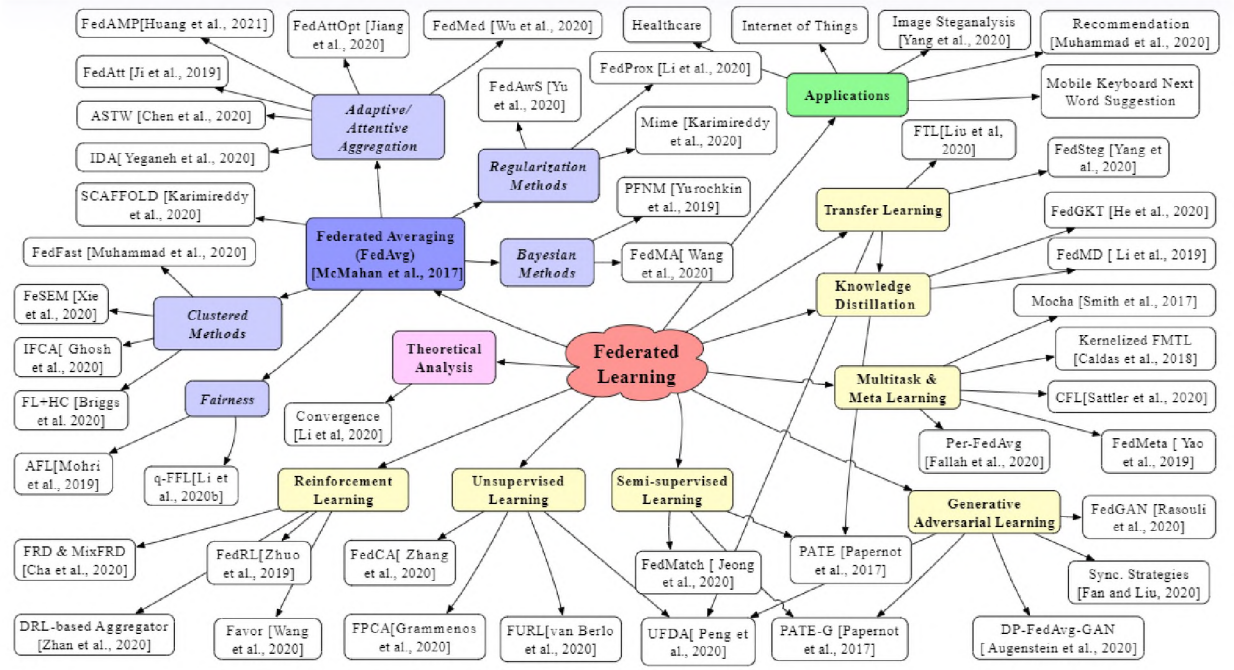


Рисунок 2.6 – Прикладні аспекти впровадження FL

Клієнтські операції. Після попередньої обробки локальні дані готові до передачі до моделі глибокого навчання. Навчальні операції в рамках інтегрованого навчання еквівалентні централізованим та індивідуальним налаштуванням. Точніше, кожен обраний учасник отримує глобальну модель, яка була передана з Сервера, і оцінює її, використовуючи свої локальні дані. Потім він встановлює ваги для локальної моделі шляхом копіювання параметрів глобальної моделі. Після ініціалізації модель адаптується до локальних даних протягом обмеженої кількості епох,

указаних сервером. У кожну епоху клієнт оцінює локальну модель за допомогою відповідного перевірочної вибірки. Остаточна локальна модель – це модель, яка досягла найменшої помилки з набором перевірки, який передається назад на сервер.

Серверні операції. Ключова відмінність між інтегрованим навчанням і традиційним підходом до навчання полягає в тому, що центральний суб'єкт відповідає за координацію навчання та створення глобальної моделі для кожного об'єднаного раунду. Рис. 2.7 ілюструє операції та зв'язок між сервером і клієнтами за раунд.

1. Оголошення. Сервер повідомляє про обчислення, і клієнти реєструються як учасники. На цьому етапі сервер також інформує учасників про попередню обробку, яку необхідно виконати для перетворення даних.

2. Вибірка клієнтів. Після реєстрації сервер ініціалізує модель, готову для локального навчання. Під час кожного об'єднаного раунду сервер отримує доступних клієнтів і відбирає частину з них. Наш підхід забезпечує гнучкість щодо цих операцій через інтерфейс Client Manager: реєстрацію/видалення клієнта, а також методи вибірки можна розширити та налаштувати для різних сценаріїв.

Наприклад, симуляція вибуття клієнтів і різні критерії відбору клієнтів і вибірки підтримуються через невеликі зміни. В даному випадку, цікавить маломасштабне FL з 3 учасниками, які є завжди доступні, а сервер вибирає всіх учасників для об'єднаного раунду.

3. Інструктаж. Після відбору учасники отримують інструкції щодо виконання локального навчання з деякими заданими параметрами. Точніше, сервер передає параметри глобальної моделі та гіперпараметри, які будуть використовуватися для локального навчання, такі як кількість локальних епох і швидкість навчання через інтерфейс MLRunner. Знову ж таки, ці параметри повністю налаштовуються, і структура дозволяє динамічно змінюватися. Тобто інструкції можуть відрізнятися для кожного клієнта в кожному раунді. У цій роботі використовуємо простий сценарій для

інструкцій клієнта та вибираємо попередньо визначені гіперпараметри без динамічної зміни.

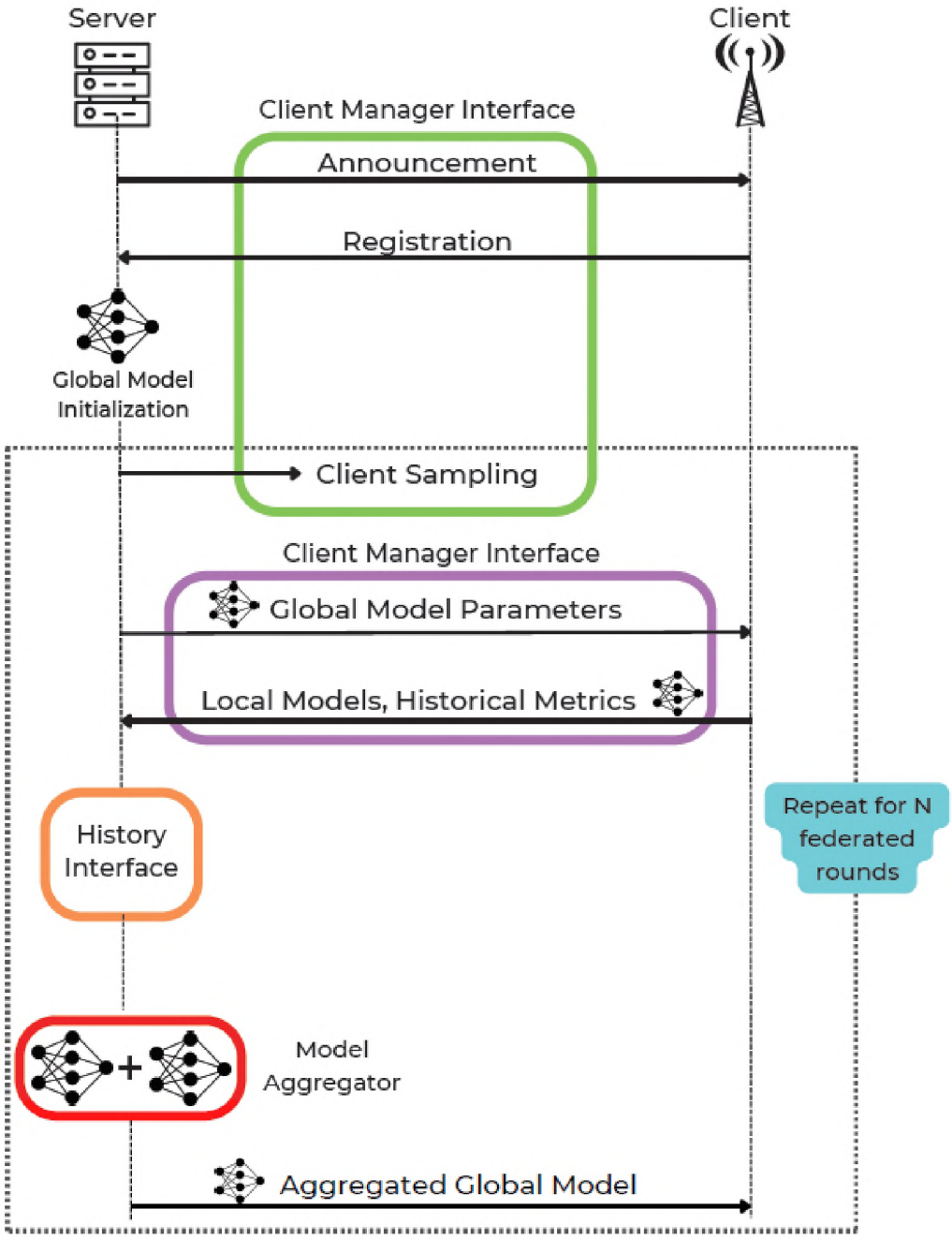


Рисунок 2.7 – Операції клієнта сервера під час інтегрованого навчання

4. Модель і метрична колекція. Сервер очікує, що клієнти виконають локальне навчання та виконають передачу своїх локальних моделей разом із історичними показниками. Після збору сервер додає історичну навчальну інформацію за допомогою інтерфейсу історії. Історичні показники

використовуються для створення остаточної глобальної моделі після FL і для інтерпретації моделі. Зокрема, сервер зберігає в пам'яті глобальну модель, яка досягла найменших середніх втрат. Усереднені втрати обчислюються з використанням вагових коефіцієнтів кількості локальних тренувальних зразків на учасника. Подібним чином сервер зберігає історичні показники за допомогою зваженого агрегування. Методи агрегування історичної інформації можна налаштувати відповідно до потреб використання.

5. Агрегація. Після додавання історичних даних і збору локальних моделей сервер виконує функцію агрегації для створення нової глобальної моделі. Існує кілька алгоритмів агрегації, щоб оцінити їх застосовність і стійкість до налаштувань із комбінованими категоріями даних, не пов’язаних з IID. Після агрегування кроки 2…5 повторюються до заданої кількості раундів.

Після об’єднаного навчання сервер передає остаточну глобальну модель учасникам. Згідно з дизайном, FL може виконуватися динамічно в різні періоди часу, щоб отримувати нові спостереження та підвищувати точність прогнозування. Крім того, учасники також можуть виконати крок локального тонкого налаштування, щоб наблизити модель до своїх локальних даних і забезпечити більш якісні прогнозні прогнози. Нарешті, для виконання прогнозів на тестових даних кожен клієнт повинен виконати операції попередньої обробки, визначені на етапі оголошення, і передати перетворені дані в навчену модель. Щоб отримати остаточний прогноз, його слід повернути до вихідного діапазону ознак.

2.4 Агрегування в рамках розподіленого навчання

Одним із найважливіших кроків інтегрованого навчання є агрегування моделей. Під час цієї фази центральний сервер збирає та агрегує моделі від учасників для оновлення стану глобальної моделі. Цей процес викликає

багато труднощів, особливо коли дані не є IID і неоднорідні серед клієнтів. З цієї причини існує багато досліджень, спрямованих на алгоритми агрегації. Досить поширеним алгоритмом, який використовується для операцій FL, є FedAvg, запропонований у [4]. Агрегація виконується шляхом обчислення середньозваженого клієнтського моделі на основі кількості даних, що, отже, призводить до того, що клієнти з більшою кількістю вибірок мають більший вплив на нову агреговану модель. Однак цей процес не завжди призводить до належних результатів. У деяких випадках, FedAvg може спричинити об'єктивну неузгодженість, тобто глобальна модель збігається до стаціонарної точки неузгодженої цільової функції, яка може будь-яким чином відрізнятися від істинної (глобальної) цілі. Ця невідповідність є результатом наявності неоднорідних і неоднорідних даних. З цієї причини з'явилися альтернативи FedAvg для фіксації неоднорідності даних. Щоб забезпечити теоретичні гарантії щодо даних, не пов'язаних з IID, у [45] запропоновано повторну параметризацію FedAvg (рис. 2.8).

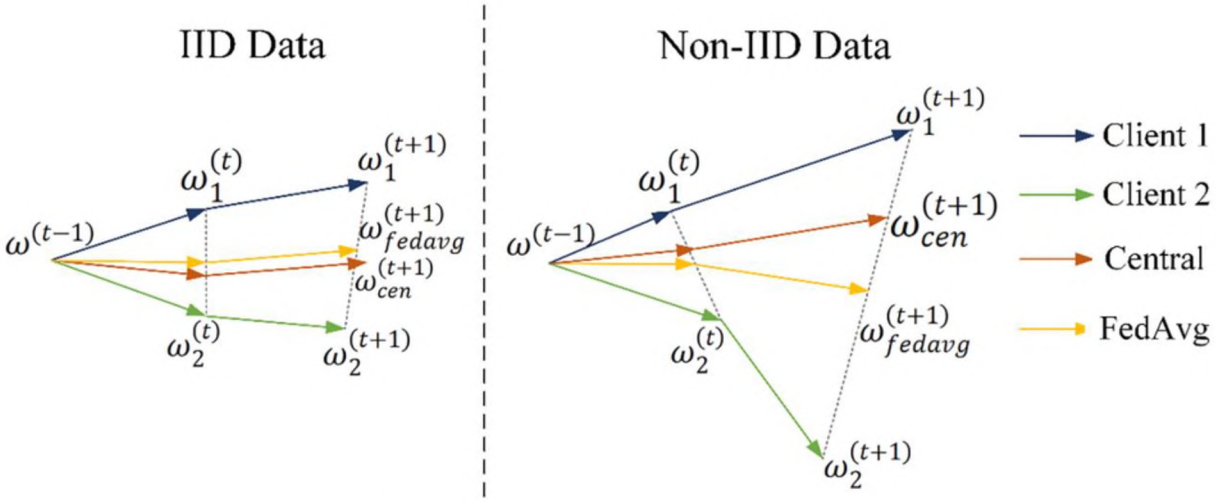


Рисунок 2.8 – Повторна параметризація FedAvg

FedProx є узагальненням алгоритму FedAvg і забезпечує надійну конвергенцію, особливо у дуже неоднорідних налаштуваннях. Точніше, налаштований член g вводиться для керування локальною ціллю, яка обмежує відстань між попередньою та поточною вагами моделі. Іншим багатообіцяючим алгоритмом є FedNova [46], метод, який правильно

нормалізує локальні оновлення моделі під час усереднення. Основний принцип FedNova полягає в тому, що вона усереднює нормалізовані локальні градієнти шляхом ділення їх на кількість локальних кроків, які кожен клієнт виконав окремо, на відміну від усереднення сукупного локального градієнта без будь-якого кроку нормалізації. FedNova забезпечує об'єктивну узгодженість, зберігаючи швидку конвергенцію помилок. FedAvgM [12] – це варіант FedAvg, який отримує перевагу шляхом накопичення оновлень моделі з використанням імпульсу сервера. Імпульс обчислюється шляхом ітеративного множення попередніх оновлень моделі на гіперпараметр 3 у кожній епосі з одночасним додаванням до результату нових. Нарешті, спроба створити об'єднані версії відомих адаптивних оптимізаторів, а саме FedAdagrad, FedYogi та FedAdam, представлена в [47]. Ці методи обіцяють вирішити проблему неоднорідності клієнта, підвищити продуктивність і знизити вартість зв'язку. Однак їх ефективність значною мірою залежить від оптимізації гіперпараметрів $(\alpha, \beta_1, \beta_2)$. Алгоритм 1 узагальнює сучасні методи агрегації, розглянуті в даній роботі.

2.5 Формування базової архітектури нейронної мережі

Згідно п. 2.1, необхідно синтезувати кілька нейронних мереж на основі архітектур, що найбільш вдало підходять для вирішення завдань прогнозування часових рядів. Для цього завантажуємо необхідні бібліотеки та компоненти. За їх допомогою реалізується варіанти моделей, відображення графіків, оптимізатор, стандартні шари, нормувальники, складові для генерації вибірки часових рядів, підтримка роботи з файлами. Надалі завантажується набір даних і формується управління попередженнями, і, нарешті сама архітектура. Приклад завантажених даних (файли у форматі .csv) за 4-ма параметрами наведено на рис. 2.9, а варіант синтезованої моделі згорткової мережі на основі одномірної згортки – рис. 2.10.

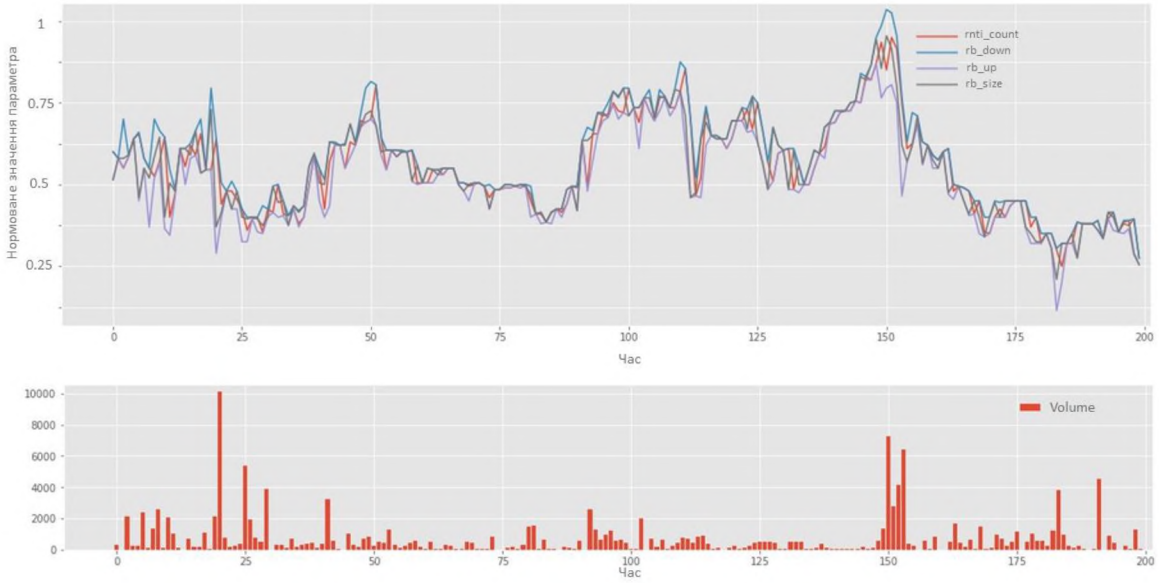


Рисунок 2.9 – Завантажені дані

```
[ ] from sklearn import metrics
    def get_model_Conv1D():
      modelC = Sequential()

      modelC.add(Conv1D(64, 5, input_shape = (xLen, 5), activation="relu"))
      modelC.add(Conv1D(128, 5, activation="relu"))
      modelC.add(MaxPooling1D())

      modelC.add(Flatten())
      modelC.add(Dense(100, activation="relu"))
      modelC.add(Dropout(0.2))

      modelC.add(Dense(1, activation="linear"))

      modelC.compile(loss="mse", optimizer=Adam(lr=1e-4))
      return modelC
[ ] model = get_model_Conv1D()
    model.summary()
    print()
    plot_model(model, dpi=70, show_shapes=True )
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 296, 64)           1664

 conv1d_1 (Conv1D)           (None, 292, 128)          41088

 max_pooling1d (MaxPooling1D  (None, 146, 128)         0
 )

 flatten_1 (Flatten)         (None, 18688)             0

 dense_2 (Dense)             (None, 100)               1868900

 dropout (Dropout)           (None, 100)               0

 dense_3 (Dense)             (None, 1)                 101

=================================================================
Total params: 1,911,753
Trainable params: 1,911,753
Non-trainable params: 0
```

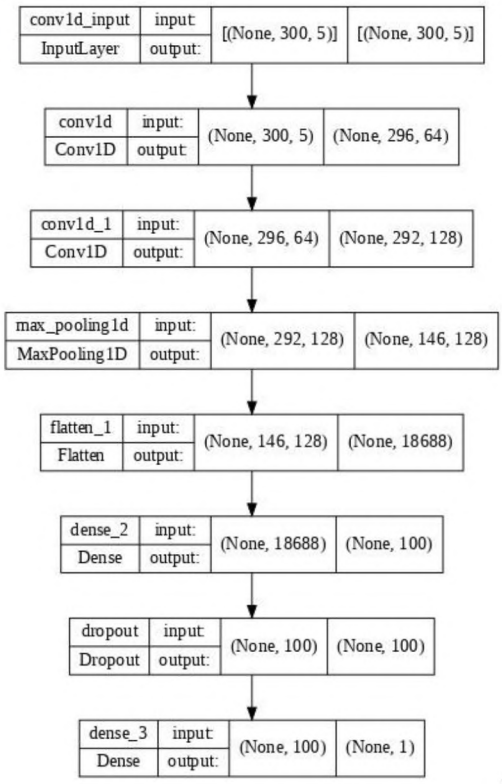


Рисунок 2.10 – Архітектура CNN на основі Conv1D
для прогнозування часових рядів на 1 крок

Після застосування методу «.fit» і реалізації навчання мережі протягом 30-ти епох для зазначеної архітектури отримаємо відповідні залежності, що

характеризуються графіками: помилки (рис. 2.11), кореляції (рис. 2.12) і прогнозу, наприклад, параметру rb_size (рис. 2.13).

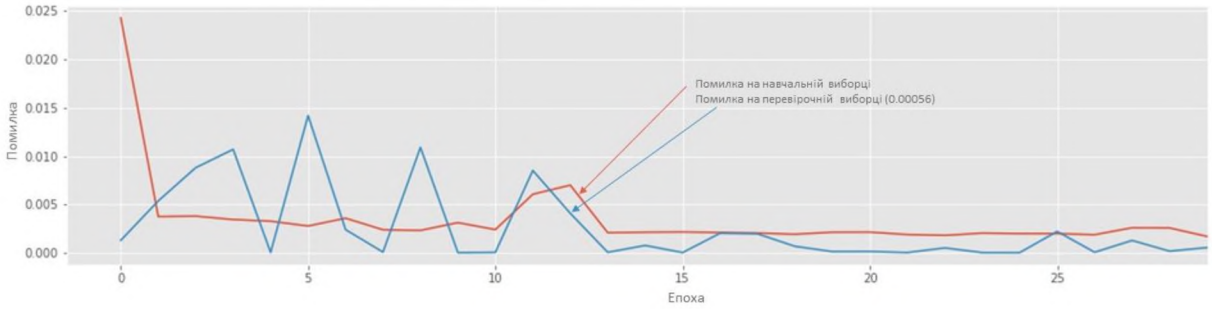


Рисунок 2.11 – Помилка для CNN на основі Conv1D

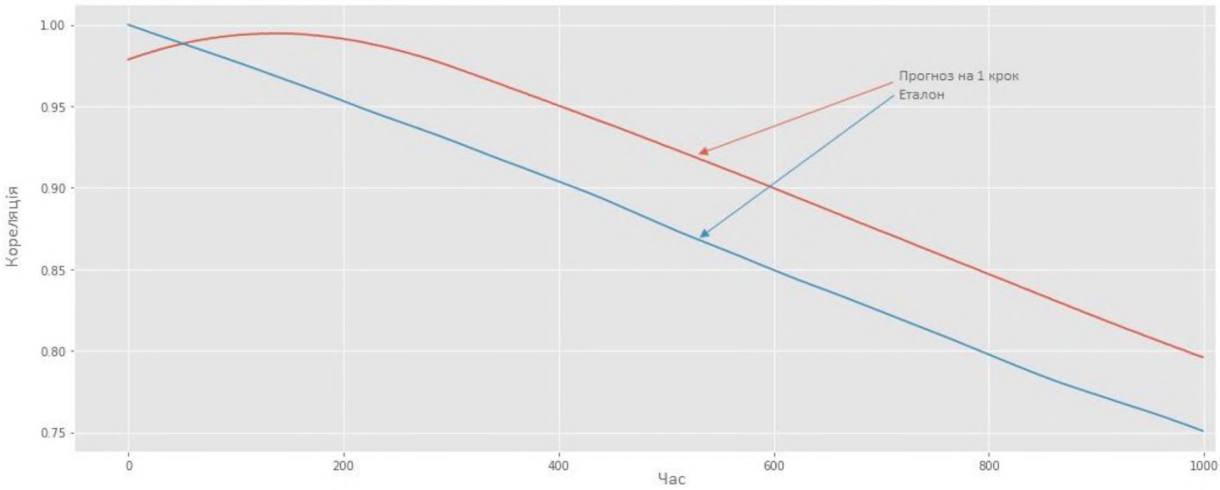


Рисунок 2.12 – Кореляція для CNN на основі Conv1D

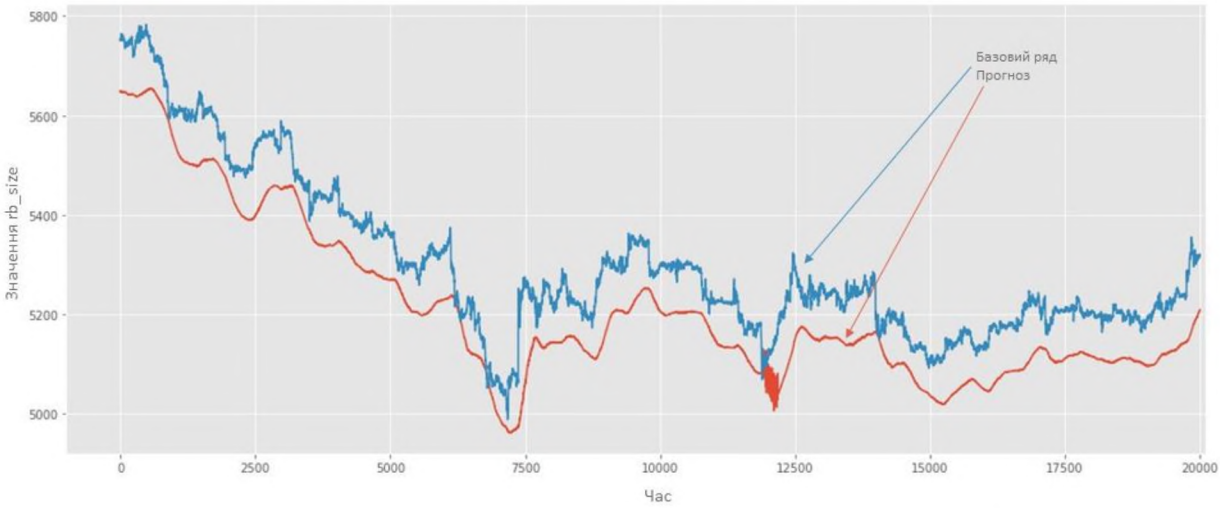


Рисунок 2.13 – Прогноз на 1 крок для CNN на основі Conv1D

Надалі залишається синтезувати найбільш оптимальні архітектури нейронних мереж та виконати підбір їх гіперпараметрів.

Висновки до розділу 2

Для підвищення точності прогнозування важлива попередня обробка даних. З цією метою проводиться очищення даних, виявлення викидів і керування ними, а також методики заповнення відсутніх даних. Щоб забезпечити однорідність даних виконується масштабування функцій та нормування. Це сприяє швидшому навчанню нейронної мережі. Надалі формується датасет, що містить навчальну, перевірочну та тестову вибірки за типовим розподілом.

Нейронні мережі можуть значно підвищити точність прогнозування в порівнянні з традиційними методами. Для оцінки можуть використовуватись метрики MSE або MAE.

Враховуючі особливості складових трафіку станцій мобільного зв'язку, для забезпечення гарантій конфіденційності пропонується використання FL. Навчання на основі FL, включає вибірку та реєстрацію клієнтів, інструктаж, локальне навчання, агрегацію локальних моделей та створення глобальної моделі. В свою чергу, формування моделі з використанням індивідуального та централізованого навчання, а також розглядається інтегроване навчання, де кожен учасник оптимізує модель на основі локальних даних, але створена глобальна модель має можливість узагальнення на більшу кількість сутностей. Глобальна модель може базуватись на одній з кількох придатних для прогнозування часових рядів архітектур, наприклад, RNN, CNN, LSTM (додаток Б), GRU. Розглянута технологія створення нейронної мережі CNN на основі Conv1D.

РОЗДІЛ 3
ОЦІНКА ВЛАСТИВОСТЕЙ СИНТЕЗОВАНОЇ МОДЕЛІ

3.1 Опис набору даних та вибір метрики

Згідно [6], використовується набір даних реальних вимірювань LTE Physical Downlink Control Channel (PDCCH), отриманих із 3-ох різних базових станцій (BS1-BS3), які вважаються окремими клієнтами FL. Набір даних надається в рамках розподіленого прогнозування трафіку для 5G. Забезпечується як анонімність, так і точність, оскільки унікальні ідентифікатори користувачів недоступні, тоді як індивідуальне спілкування може бути розділене в межах набору даних для отримання слідів високої чіткості. Для кожної базової станції передбачено 11 функцій щодо UpLink і DownLink. Завдання полягає в тому, щоб передбачити rnti_count, кількість виділених блоків ресурсів для завантаження (rb_up, rb_down), а також розміри транспортних блоків у UpLink, DownLink, використовуючи як вхідні дані спостереження з вікно $T = 10$ на базову станцію.

Як відомо, параметр rnti_count – це унікальний ідентифікатор, який присвоюється кожному пристрою (наприклад, мобільному телефону) у мережі мобільного зв'язку для керування та координації передачі даних. Його можна використовувати для моніторингу кількості активних користувачів або пристроїв, підключених до базової станції в даний момент часу. Це може бути корисним для управління мережею, аналізу продуктивності та планування потужності мережі. Коли RNTI наближається до максимальної межі, це може вказувати на те, що базова станція працює на межі своїх можливостей, що може вимагати уваги оператора мережі. RNTI виконує кілька ключових функцій, включаючи:

– ідентифікація пристроїв – кожен пристрій отримує унікальний RNTI, що дозволяє базовій станції правильно направляти вхідні та вихідні дані до правильного пристрою;

– управління ресурсами – RNTI використовується для керування розподілом радіочастотних ресурсів між пристроями;

– безпека та конфіденційність – RNTI допомагає забезпечити, що дані передаються та приймаються конфіденційно, і лише призначеним пристроям.

В свою чергу, параметри RB_up і RB_down – кількість ресурсних блоків і відносяться до виділених радіоресурсів для UpLink) і низхідного (DownLink) каналів зв'язку. Це кількість ресурсних блоків, виділених для висхідного каналу. Східний канал використовується для передачі даних від мобільного пристрою (наприклад, смартфона) до базової станції. Вихідний канал важливий для таких функцій, як передача голосових даних, надсилання повідомлень та завантаження даних на сервер. Низхідний канал використовується передачі даних від базової станції до мобільного пристрою. Це включає такі завдання, як прийом веб-сторінок, потокове відео і завантаження файлів.

У контексті мереж мобільного зв'язку, управління цими ресурсами критично важливе для забезпечення якісного зв'язку, оскільки воно впливає на пропускну здатність, швидкість і якість зв'язку між пристроями користувача і мережею.

Модель глибокого навчання для частини «клієнт» у випадку застосування регулярізації під час навчання дозволяє отримати залежності, що наведені на Рис. А.3. Він ілюструє навчання, перевірку та тестування вимірювань UpLink, DownLink на базову станцію. Легко помітити, що дані трафіку відрізняються для кожної базової станції, порушуючи припущення щодо даних IID традиційного ML.

Метрики, які ми використовували для обчислення нашої похибки передбачення, це середня абсолютна помилка (MAE), середньоквадратична помилка (RMSE) і нормалізована середньоквадратична помилка (NRMSE). MAE є дуже поширеним показником, який використовується для задач регресії, оскільки його одиниці значення відповідають передбачуваним цілям. MAE розраховується як середнє значення абсолютних похибок, тобто

абсолютне (завжди позитивне) значення різниці між очікуваним і прогнозованим результатом. MAE для вектору y з n прогнозованих значень і вектору y з n очікуваних значень обчислюється за такою формулою:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|. \quad (3.1)$$

RMSE є дуже популярним показником для оцінки якості прогнозів. По суті, він показує, наскільки далеко прогнози відхиляються від очікуваних значень, використовуючи евклідову відстань:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\overline{y}_i - y_i)^2}{n}}, \quad (3.2)$$

де n – кількість точок даних;

y_i – і-те вимірювання;

$\overline{y}_i$ – відповідне передбачення.

Наявність квадратного кореню призводить до того, що більші помилки мають непропорційно великий вплив на RMSE, тому RMSE чутливий до викидів. Крім того, треба брати до уваги, що точки даних, які є вимірюваннями UpLink і DownLink в байтах, дуже великі за масштабом, тому вони призведуть до великих помилок.

Щоб полегшити порівняння між нашими експериментами, ми використовуємо NRMSE для вимірювань висхідної та низхідної лінії зв'язку, що є нормалізованою формою RMSE:

$$NRMSE = \frac{1}{\overline{y}} \cdot \sqrt{\frac{\sum_{i=1}^{n}(\overline{y}_i - y_i)^2}{n}}. \quad (3.3)$$

Метрики MAE і RMSE використовуються для обчислення похибки передбачення з використанням 5-ти прогнозованих значень. NRMSE розглядає лише вимірювання UpLink і DownLink, оскільки вони є основною метою дослідження.

3.2 Налаштування навчання

Щоб дослідити ефективність сучасних моделей глибокого навчання, застосованих до FL, а також внесок об'єднаних алгоритмів агрегації в кінцеву якість глобальної моделі, треба порівняти модельні архітектури за допомогою індивідуального навчання, у якому кожна базова станція використовує лише локальні дані та навчає персоналізовану модель і централізоване навчання (тобто базові станції передають свої дані третій стороні), а навчання виконується за допомогою комбінованого набору даних та об'єднаного навчання (тобто базові станції спільно навчають модель без обміну необробленими даними). Отримані параметри треба порівняти з такими архітектурами моделі.

1. MLP – проста штучна нейронна мережа прямого зв'язку. MLP не можуть працювати з матричними представленнями, тому часові ряди зводяться до одновимірних масивів. Вибрана архітектура складається з 3 прихованих шарів, $h = \{256, 128, 64\}$.

2. RNN – простий тип рекурентної нейронної мережі, який може моделювати часові залежності від послідовних даних. Вибрана архітектура включає рівень RNN із 128 одиниць. Потім вихідні дані подаються в MLP з одним прихованим шаром і 128 одиницями.

3. LSTM – вдосконалення RNN, що обмежує проблему розривних градієнтів і може моделювати довші послідовні дані. Подібно до RNN, вибираємо рівень LSTM з 128 одиницями, а вихідні дані передаються на MLP з одним прихованим шаром і 128 одиницями для створення остаточного прогнозу.

4. GRU вирішує проблему вибухових градієнтів на RNN. На більш високому рівні це спрощена версія LSTM, що забезпечує подібні результати до LSTM з меншими обчислювальними параметрами. Для GRU реалізований рівень і 128 одиниць і MLP з одним рівнем.

5. CNN [48, 49] – безпосередньо працює з необробленими даними за допомогою шарів згортки та успішно виконує завдання CV. У багатовимірних часових рядах операції згортки можуть ефективно використовувати структурну коваріацію. Вибрана CNN приймає як вхідні дані 3-вимірну матрицю розміру $(1, T, \#variates)$ і подає її на 4-ри 2-вимірні згорткові шари з розмірами фільтрів $\{16, 16, 32, 32\}$. Після роботи зі згортковими шарами вихідні дані подаються на 2-вимірний середній шар об'єднання і, нарешті, на повністю пов'язаний рівень.

Щоб зберегти узгодженість і забезпечити справедливе порівняння, зберігаємо ту саму архітектуру моделі для трьох параметрів навчання. Для кожної моделі та налаштування навчання вибираємо оптимізатор Адама [40] для швидшої конвергенції з кроком швидкістю навчання 0.001. Функцією активації серед шарів моделі є ReLU. Під час навчання оптимізуємо середню квадратичну помилку та використовуємо розмір батча 128. Для індивідуального та централізованого навчання використовуємо 270 епох, а для об'єднаних налаштувань вибираємо 30 раундів із 3 локальними епохами на учасника. Кількість епох для індивідуального та централізованого налаштування вибрано таким чином, щоб забезпечити справедливе порівняння з інтегрованим навчанням, тобто в інтегрованому навчанні повна кількість епох становить 270 (раунди × локальні епохи × #учасники).

У всіх налаштуваннях вибираємо модель, яка досягла найменшої середньоквадратичної помилки на перевірочній виборці під час навчання. Для індивідуальних і централізованих підходів інтегруємо компонент ранньої зупинки з порогом 50 епох. Приклад синтезованої LSTM наведено в додатку Б.

3.3 Вплив етапу попередньої обробки

Надалі виконуємо FC на базову станцію. Використовуючи FC, значення, які падають нижче та вище вказаного мінімального та максимального порогів, обмежуються та зменшуються до вказаного мінімуму та максимуму відповідно. Оптимальні пороги на базову станцію після застосування пошуку за малою сіткою наведені в табл. А.2.

Експериментальне дослідження виконується на Python. Якщо не зазначено інше, кожен експеримент повторюється з нуля 150 разів з різними даними, щоб забезпечити точні та узагальнені результати. Для кожного експерименту повідомляємо усереднені результати та стабільність із використанням стандартного відхилення. У рамках цієї експериментальної оцінки вивчаємо вплив етапу попередньої обробки на остаточні прогнози з використанням об'єднаних налаштувань. На цьому етапі виконуємо FL з використанням алгоритму FedAvg [5]. Наша мета тут полягає в тому, щоб оцінити трансформацію викидів з використанням підміни та обмеження, а також використання глобального масштабування, що призводить до підвищення якості прогнозів. в більшості випадків метод FC забезпечує майже еквівалентні результати. На рис. А.4 представлено усереднений NRMSE з і без техніки FC на наборах валідації та тестування відповідно. Загалом FC призводить до значного зменшення помилки прогнозування, а також є більш стійким щодо випадковості, внесеної ініціалізацією. Наприклад, розглянемо результати на моделях LSTM. Стандартне відхилення незначне за методики FC. З іншого боку, навчання без перетворень не тільки призводить до зниження точності прогнозування, але також призводить до того, що моделі сильно залежать від випадкової ініціалізації, оскільки існує величезне відхилення між циклами. Щоб отримати більш вичерпні результати, представляємо NRMSE для кожної базової станції окремо (рис. А.5) спостерігаємо, що тенденція навчання моделей за допомогою техніки FC для досягнення меншої похибки також представлена окремо,

принаймні для базових станцій BS1 і BS2. Наприклад, щодо валідаційного зниження NRMSE на BS1 і BS2, використання моделі LSTM становить приблизно 1/3 від відповідної моделі, навченої без FC. Що стосується помилки тестування, спостерігаємо, що в BS1 спостерігається зменшення приблизно на 1/5 у всіх випадках, тоді як у BS2 зниження є помітним. З іншого боку, метод FC не покращує NRMSE у BS3, що пояснюється випадковими стрибками та значною наявністю викидів.

Окрім обробки викидів, невід'ємним етапом попередньої обробки є техніка масштабування ознак. У FL можемо дозволити кожному учаснику локально обчислювати перетворення або використовувати глобальну змінну для масштабування функцій. Рис. А.6 ілюструє усереднену перевірку та тестування NRMSE з локальним [22] і глобальним масштабуванням. Усереднене NRMSE на перевірочному наборі з глобальним масштабуванням призводить до меншої помилки та стандартного відхилення порівняно з локальним масштабуванням. На наборі для тестування вплив глобального масштабування менший, але в усіх випадках він перевершує відповідний метод локального масштабування.

3.4 Порівняння налаштувань навчання

Після вибору застосованих перетворень під час попередньої обробки ми продовжуємо порівняння розглянутих параметрів навчання. У табл. А.3 наведено усереднені значення MAE і RMSE з використанням усіх 5-ти прогнозованих виходів, а також усереднене значення NRMSE з використанням усереднених результатів передбачень висхідної та низхідної лінії зв'язку. Модель найвищої якості для налаштування навчання позначена жирним шрифтом, а друга найкраща підкреслена. Зауважте, що ми використовуємо однакові параметри навчання для кожного параметра, щоб зберегти послідовність і забезпечити справедливе порівняння. Крім того, ми

зберігаємо FedAvg [5] як об'єднаний алгоритм агрегації. Всі моделі забезпечують відносно однакову похибку прогнозування. В окремих налаштуваннях двома найкращими моделями є LSTM і GRU, оскільки вони представляють найменшу похибку в усіх показниках. У централізованому навчанні немає очевидного способу визначити модель(и) найвищої якості. LSTM і GRU працюють найкраще щодо MAE, GRU і CNN щодо RMSE та MLP і CNN щодо NRMSE. Тим не менш, усі моделі призводять до подібних помилок прогнозування. З огляду на стабільність моделі, яка дає індикатор якості надійності моделі, LSTM і GRU перевершують інші моделі. З іншого боку, CNN, яка є найкращою моделлю щодо RMSE і другою найкращою щодо NRMSE, має найвище стандартне відхилення. Це вказує на те, що хоча CNN може надавати високоякісні прогнози, її здатність до узагальнення схильна до випадковості. Нарешті, LSTM і GRU перевершують інші моделі в об'єднаних налаштуваннях, що також призводить до нижчого стандартного відхилення. Виходячи з наведених вище спостережень, LSTM і GRU є моделями найвищої якості в кожному з трьох навчальних закладів.

Безпосередньо порівнюючи ідентифіковані найкращі моделі серед налаштувань навчання щодо точності передбачення, можна стверджувати, що індивідуальне налаштування призводить до найвищої якості, за яким слідує централізоване та об'єднане навчання, що узгоджується з [21]. На рис. А.7 показано усереднене NRMSE для перевірки та тестування. У всіх налаштуваннях MLP і CNN мають більш високу нестабільність як у перевірці, так і в тестових наборах. З іншого боку, LSTM і GRU забезпечують високу стійкість на наборах перевірки, що відображається у вищій якості прогнозів на тестових наборах. Окрім усереднених результатів, також цікавить точність прогнозування на кожній базовій станції окремо. Рис. А.8 ілюструє отриманий тестовий NRMSE для сайту та налаштування навчання. Загалом, помилка прогнозування майже еквівалентна для моделі та налаштування навчання. Точність прогнозування вища на BS3, що пояснюється тим фактом, що спостереження на цьому сайті не містять

екстремальних викидів, принаймні для вимірювань DownLink. З іншого боку, помилка прогнозу на BS1 і BS2 приблизно на 75 % вища. Така поведінка пояснюється не лише екстремальними викидами, але і вищими значеннями для вимірювань UpLink і DownLink. Спеціально для об'єднаних налаштувань спостерігаємо, що модель LSTM забезпечує не лише високу точність прогнозування, але й високу стабільність як окремо, так і разом. На рис. А.9 показано прогноз об'єднаної моделі LSTM на базову станцію щодо вимірювань низхідної та висхідної лінії зв'язку порівняно зі значеннями правдивості на землі. З візуалізації прогнозів видно, що прогнози на BS2 відповідають реальній дійсності. На моделях BS1 і BS2 модель не може вловити стрибки та надзвичайно високі значення. В BS3 відповідні вимірювання набагато вищі, ніж на BS2, тобто приблизно на 90 % і понад 97 % вище для вимірювання UpLink і DownLink відповідно. Тим не менш, шаблони вимірювань добре представлені, а прогнозна точність інтегрованого навчання на одному рівні з індивідуальними та централізованими налаштуваннями. Рис. А.10 перевіряє збіжність FL щодо централізованого налаштування щодо MAE на наборі для навчання та перевірки (з використанням масштабованих даних і врахуванням усередненого MAE). Тобто, що FL може забезпечити високу якість прогнозування часових рядів із можливістю узагальнення (що неможливо забезпечити при індивідуальному навчанні), поважаючи конфіденційність учасників (на відміну від централізованого навчання). Хоча індивідуальне навчання призводить до меншої помилки прогнозування на базову станцію, це стосується моделей, які працюють безпосередньо на локальних даних, і, отже, можливість узагальнення обмежена. Іншими словами, навчання моделі для кожної базової станції фіксує просторово-часову динаміку спостережень цього сайту і не може забезпечити високоякісні прогнози для інших базових станцій. Крім того, централізоване навчання вимагає передачі отриманих вимірювань, таким чином обмежуючи використання особистих даних. У цьому підрозділі ми виконуємо локальне тонке налаштування, подібне до [18,

21], щоб спостерігати, чи досягають додаткові локальні епохи персоналізації. Кожна модель навчається з нуля 10 разів з використанням різних початкових значень ініціалізації. У табл. А.4 наведено отримане NRMSE на базову станцію, а також усереднене NRMSE на налаштування навчання на тестовому наборі. В усіх випадках локальні точно налаштовані моделі здатні глибше охоплювати місцеві характеристики та призводити до меншої похибки прогнозування. Розглядаючи усереднений NRMSE, спостерігаємо, що як централізоване, так і FL перевищують індивідуальні налаштування. Інтуїтивно це означає, що глобально навчена модель фіксує подібну динаміку екзогенних спостережень до цільової базової станції, і, виконуючи локальне тонке налаштування, отримуємо вищу точність прогнозування. Індивідуальне навчання не може забезпечити узагальнення, і, отже, моделі, навчені в централізованих і об'єднаних налаштуваннях, перевершують відповідні локально навчені моделі. Точно налаштовані CNN і GRU як з централізованого, так і з FL перемагають найкращу модель (GRU) в індивідуальних налаштуваннях. Розглядаючи результуючу помилку на базову станцію, спостерігаємо подібну поведінку: у більшості випадків точно налаштовані моделі перемагають відповідні глобальні моделі. Деякі винятки трапляються в BS3, наприклад, модель CNN як у централізованому, так і в інтегрованому навчанні. У BS1 модель CNN має найкращу продуктивність, тоді як об'єднана тонко налаштована CNN декілька перевершує інші налаштування. У BS2 тонко налаштована централізована CNN перевершує відповідну федеративну модель. Нарешті, у BS3 поведінка відрізняється для кожного параметра навчання, оскільки найкраща архітектура моделі в одному параметрі не відображає найкращу модель в решті параметрів. Така поведінка, знову ж таки, пояснюється довільними стрибками та наявністю екстремальних викидів. Загалом етап локального тонкого налаштування забезпечує більш якісні прогнози, а також забезпечує узагальнення для централізованих і об'єднаних налаштувань. Наприклад, розглянемо згенеровану модель у рамках централізованого або об'єднаного навчання.

Глобальна модель забезпечує узагальнення серед базових станцій, оскільки вона фіксує глобальну динаміку, тоді як високоякісне прогнозування також може надаватися зовнішній, не залученій організації. Крім того, зовнішня базова станція може отримати навчену модель і виконати локальне тонке налаштування, яке неможливо легко забезпечити через індивідуальні налаштування через проблеми конфіденційності або коли розподіл між двома сторонами сильно спотворений. Крім того, FL створює динамічне середовище, де під час навчання можуть з'являтися додаткові учасники. Це унікальна властивість FL, і її не можна моделювати на основі решти параметрів. Нарешті, об'єднане навчання може відбуватися на вимогу, наприклад, раз на день або тиждень, щоб своєчасно фіксувати динаміку нових спостережень і потенційно сприяти кращим прогнозам.

Проведені порівняння стосуються FL за алгоритмом агрегації FedAvg. Крім того розглядаємо просте усереднення, що позначається як SimpleAvg, і агрегування на основі медіани отриманих вагів, що позначається як MedianAvg. Решта розглянутих агрегаторів намагаються вирішити проблему, не пов'язану з IID, і запровадити налаштування параметри. Наприклад, FedProx [45] контролює розбіжність ваги за допомогою параметра регуляризації p.

Ми обираємо об'єднану модель LSTM, оскільки вона забезпечує високі показники прогнозування та ефективності. Навчаємо модель з нуля 20 разів, використовуючи різні початкові числа для ініціалізації, щоб отримати вичерпні результати для кожного агрегатора. Отримане усереднене NRMSE на мережу в тесті показано на рис. 3.1. Штрихова лінія відповідає базовій лінії FedAvg, а точки показують помилку прогнозування на мережу. Для FedProx спостерігаємо, що чим нижчі значення μ, тобто $\mu \in \{10^{-2}, 10^{-3}\}$, тим краща точність прогнозування. Коли $\mu = 0$, FedProx еквівалентний FedAvg. Оскільки μ є параметром регуляризації, вищі значення призводять до повільнішої збіжності, і, отже, вибрані 30 об'єднаних раундів не призводять до низької помилки передбачення. Коли μ становить лише 10^3, FedProx

декілька перевищує базовий рівень FedAvg. Для агрегатора FedAvgM спостерігаємо, що чим вище значення β, тим більша помилка. Зауважте, що коли $\beta = 0$, FedAvgM еквівалентний FedAvg.

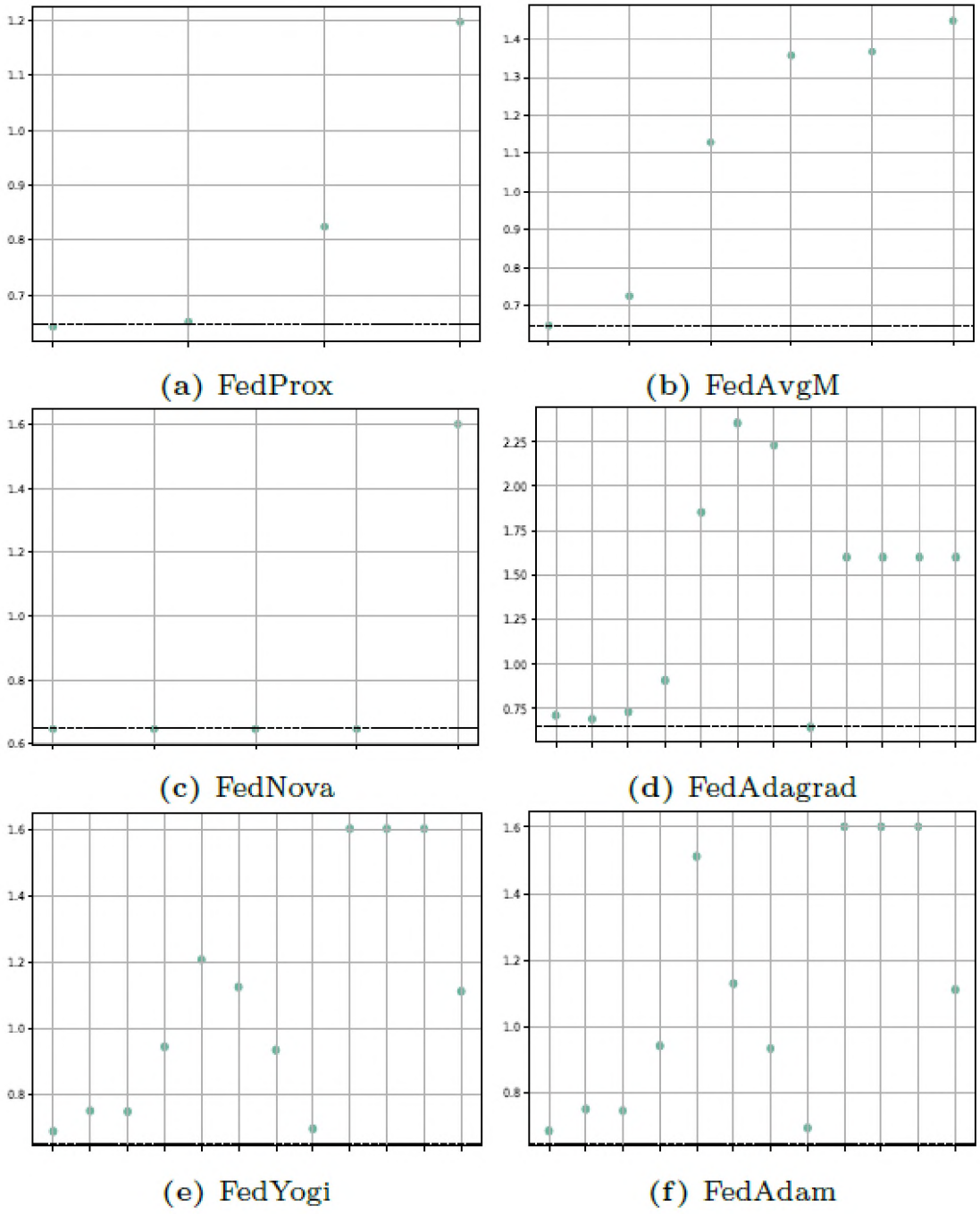


Рисунок 3.1 – Усереднений тестовий NRMSE для агрегатора з використанням об’єднаної моделі LSTM

Згідно з експериментальними результатами, FedAvgM не призводить до меншої похибки прогнозування та не може подолати FedAvg. Що

стосується FedNova, спостерігаємо, що він відповідає прогнозній точності FedAvg, коли $\rho \leq 10^{-1}$, і трохи перевищує FedAvg. Для FedAdagrad, FedYogi та FedAdam потрібен пул із 4 гіперпараметрів. На помилку прогнозування цих агрегаторів значною мірою впливають базові параметри, і в багатьох випадках вони призводять до високих значень помилок. FedAdagrad перемагає FedAvg лише в одному випадку, тоді як FedYogi та FedAdam не дають прогнозів вищої якості, ніж FedAvg. Загалом очевидно, що FedAvg відповідає FedProx і FedNova, а сучасні алгоритми агрегації не перевершують простий підхід FedAvg. Насправді алгоритми FedOpt [14] призводять до нестабільної продуктивності прогнозування, а кінцева глобальна модель є дуже чутливою до введених гіперпараметрів. Параметр імпульсу FedAvgM не покращує відповідну помилку, тоді як FedProx і Fed-Nova в деяких випадках дещо перевершують FedAvg. На рис. А.11 показано зведені результати NRMSE для алгоритму агрегації для наборів перевірки та тестування. Зауважте, що ми вибираємо найкращі гіперпараметри для кожного агрегатора на основі результатів набору перевірки. FedProx забезпечує найвищу надійність проти випадкової ініціалізації та нижчу середню NRMSE як для перевірки, так і для тестування. Далі, незважаючи на свою простоту, SimpleAvg, здається, перевершує решту агрегаторів як за NRMSE, так і за стабільністю. FedNova декілька перевершує FedAvg і демонструє відносно високу стабільність. Решта агрегаторів демонструють високий поріг, тобто чутливість до випадковості, тоді як FedAdagrad вимагає ретельного налаштування своїх параметрів. Існує чотири категорії даних, не пов’язаних з IID: характеристика, ціль, кількість і часові перекоси, змішаний випадок, який є унікальним і безпосередньо стосується реального світу. Ключова відмінність результатів порівняно з [31] полягає в тому, що FedNova демонструє високу стабільність. Пояснюємо таку поведінку змішаним типом даних, які не є IID, у даному наборі даних, а також нашим варіантом використання. Тим не менш, все ще є місце для вдосконалення, і додаткові експериментальні дослідження, особливо в області досліджень

об'єднаного прогнозування часових рядів, яким приділялася обмежена увага, повинні бути ретельно досліджені. Створення високоякісних моделей прогнозування трафіку з можливістю узагальнення є складним завданням, враховуючи різні моделі дат на різних базових станціях. У цій роботі ми застосували підхід інтегрованого навчання для вирішення кількох виявлених проблем через природу даних, що не є IID. На відміну від централізованих підходів, запропоновані методи зводять до мінімуму проблеми конфіденційності та конфіденційності бізнесу, а розподілене навчання закладає основу для широкомасштабної участі, яка може призвести до створення інтелектуальних прогнозів. Результати показують, що розподілене навчання має потенціал узагальнення, не вимагає передачі приватних даних третім особам і забезпечує динамічне середовище виконання. Нарешті, локальне тонке налаштування призводить до вищої точності прогнозування, і, отже, розподілене навчання може призвести до більш розумних моделей, ніж індивідуальні та централізовані підходи. Важливим перспективним напрямком є зосередження на етапі попередньої обробки, який, згідно з нашими результатами, сильно впливає на ефективність навчання. Незважаючи на те, що попередня обробка є невід'ємним етапом машинного навчання, існує обмежена кількість проведених досліджень щодо впливу попередньої обробки, застосованої до об'єднаних налаштувань. Всі архітектури навчання забезпечують однакову точність прогнозування. LSTM і GRU призводять як до менших похибок, так і до високої стійкості. Оцінка додаткових нових архітектур може призвести до моделей вищої якості. В більшості випадків агрегатори відповідають базовому сценарію FedAvg. Дослідження додаткових алгоритмів агрегації, які можуть обробляти часову динаміку, може призвести до вищої точності прогнозування.

3.5 Техніко-економічне обґрунтування прийнятих рішень

При оцінці вартості створення моделі глибокого навчання доцільно враховувати витрати на обміні даними між локальними та глобальною моделями. Його обсяг буде залежати від архітектури нейронної мережі, що використовується. Щоб виміряти його, вибираємо об'єднаний раунд, який досяг найменшої помилки щодо набору перевірки, і вимірюємо вартість, використовуючи розмір моделі, який потрібно передати від сервера до учасників і назад. Кожен учасник (клієнт) отримує та передає вагові коефіцієнти моделі за об'єднаний раунд. Сервер передає вагові коефіцієнти моделі вибраним учасникам, що означає передачу та прийом вагових коефіцієнтів моделі 3 рази за об'єднаний раунд на стороні сервера. Для кожної моделі визначаємо відповідний розмір у кілобайтах, раунд, який досяг найменшої помилки підтвердження, а також загальне вимірювання висхідного та низхідного каналів на стороні клієнта та сервера в мегабайтах. Враховуючі, що найбільш вагомим є показник NRMSE, GRU забезпечує ефективніші обчислення. Однак розмір передачі LSTM відносно близький до GRU і забезпечує вищу надійність. На користь LSTM у порівнянні з GRU свідчить те, що така мережа працює найкраще в наборі перевірки, стійка до шуму, ефективність LSTM близька до GRU. Модель на основі архітектури LSTM містить ≈ 3000 рядків коду Python. Якщо замовляти розглянутий в роботі програмний продукт з подібним функціоналом у фрілансера, визначальним чинником вартості розробки нейронної мережі є трудомісткість. Середня продуктивність програміста Python – це 25 рядків коду на 1 год. Тобто необхідний середній час складає:

$$T = \frac{3000}{20} = 150 \text{ [год]}. \tag{3.4}$$

Згідно [50], в середньому, 1 година роботи програміста Python з навичками реалізації моделей глибокого навчання нейронних мереж для обробки даних регресії та прогнозування часових рядів складає

≈ 950-1550 грн. При цьому, вважаємо, що він працює з готовим датасетом. Відповідно, на оплату програміста-фрілансера з відповідною класифікацією витрати складатимуть ≈ 142500 грн.

Висновки до розділу 3

Розглядаються різні аспекти застосування та оцінки моделей глибокого навчання, включаючи вибір метрик, налаштування навчання, попередню обробку даних та порівняння різних архітектур нейронних мереж. Використання метрик, таких як MAE, RMSE та NRMSE виявилося ефективним для оцінки точності прогнозування моделей. Вивчення впливу різних параметрів навчання (як індивідуального, так і централізованого) показало, що вибір правильного підходу до навчання є критичним для досягнення високої точності прогнозування.

Етап попередньої обробки значно впливає на якість прогнозів. Техніки, такі як фільтрація викидів та масштабування ознак, суттєво покращують результати.

Різні архітектури нейронних мереж, включаючи RNN, LSTM, GRU та CNN, були порівняні для визначення їх ефективності у задачах прогнозування. LSTM та GRU показали найкращі результати у більшості налаштувань навчання, з яких перевага віддається LSTM. Аналіз витрат на створення та використання моделей глибокого навчання також враховувався, при цьому LSTM виявилася найбільш ефективною з точки зору співвідношення вартості та ефективності.

Для досягнення високої точності та ефективності відповідних архітектур моделей необхідний комплексний підхід до вибору та налаштування моделей глибокого навчання для завдань прогнозування в мережах мобільного зв’язку.

ВИСНОВКИ

Підготовка даних є ключовим фактором для підвищення ефективності прогнозувань. Цей процес включає очистку даних, управління аномаліями та заповнення пропущених значень. Масштабування та нормалізація даних також важливі для забезпечення консистенції і сприяють ефективнішому навчанню моделей. Важливо також правильно формувати датасети, включаючи тренувальні, валідаційні та тестові набори даних.

Використання нейронних мереж демонструє значне підвищення точності у прогнозуванні порівняно з класичними методами. Для оцінки точності моделей можна застосовувати такі метрики, як MSE (середня квадратична помилка) або MAE (середня абсолютна помилка).

З огляду на особливості мобільних мереж, для забезпечення конфіденційності даних рекомендується використання FL. Воно передбачає вибір клієнтів, їх інструктаж, локальне навчання, агрегацію локальних моделей і створення загальної моделі. Цей підхід дозволяє створювати моделі, які можуть бути загальними для різних сутностей, із збереженням можливості індивідуального та централізованого навчання.

Розглянуто вплив різних підходів до навчання, обробки даних та вибору архітектури нейронних мереж на ефективність моделей глибокого навчання. Виявлено, що правильний вибір параметрів навчання та архітектури є ключовим для забезпечення високої точності прогнозування.

Порівняльний аналіз різних архітектур нейронних мереж показав, що LSTM і GRU є найбільш ефективними у більшості налаштувань. При цьому LSTM виокремилась як найбільш вигідна з точки зору балансу вартості та ефективності. Розробка та налаштування моделі глибокого навчання - це ітеративний процес. Архітектура мережі повинна бути адаптована до специфіки даних мобільних мереж. Це означає, що необхідно постійно експериментувати з різними архітектурами, гіперпараметрами та методами обробки даних, щоб знайти найкраще рішення для конкретного застосування.

Неправильно налаштовані параметри можуть призвести до перенавчання або недонавчання моделі. Оптимізація гіперпараметрів, таких як швидкість навчання, розмір пакету (batch size), кількість епох, та інших, відіграє ключову роль у забезпеченні ефективності моделі. Перед тренуванням моделі, дані мають бути ретельно оброблені. Це включає видалення шуму, нормалізацію, стандартизацію, а також обробку втрачених даних. Ефективна обробка даних забезпечує краще «розуміння» даних моделлю. Під час тренування моделі важливо регулярно перевіряти її продуктивність на валідаційних та тестових наборах даних. Це допомагає ідентифікувати та виправити проблеми, такі як перенавчання, та забезпечити, що модель буде ефективно працювати на реальних даних.

Таким чином, результатами роботи є модель глибокого навчання нейронної мережі для прогнозування завантаженості станцій мобільного зв'язку; рекомендацій щодо використання моделі глибокого навчання нейронної мережі для прогнозування завантаженості станцій мобільного зв'язку можуть бути використані для подальших досліджень за даною тематикою та при проектуванні систем мобільного зв'язку.