УДК 004.4'24 https://doi.org/10.15407/pp2025.04.023

Я.О. Гайдукевич, А.Ю. Дорошенко

МОДЕЛЬ АДАПТИВНОГО ІНФЕРЕНСУ В МОБІЛЬНИХ СИСТЕМАХ

У статті запропоновано та досліджено нову модель адаптивного розподілу процесу інференсу (застосування ML-моделі для отримання прогнозу) між локальними та серверними обчисленнями для мобільних інтелектуальних систем прогнозування. Метою розробки моделі є подолання фундаментального протиріччя між вимогами до високої точності прогнозів (що досягається за рахунок потужних серверних ML-моделей) та необхідністю забезпечити короткий час відгуку, автономність роботи та енергоефективність на пристроях з обмеженими ресурсами. Запропонована модель формалізує динамічний механізм вибору шляху виконання інференсу (локальний TFLite, серверний мікросервіс або гібридний режим) на основі аналізу контексту виконання: якості мережевого з'єднання, рівня заряду батареї, обчислювальної складності запиту та терміновості результату. Модель реалізована в архітектурі, що поєднує Flutter-клієнт із контейнеризованими мікросервісами, та валідована на завданні короткострокового метеорологічного прогнозу. Експериментальні результати демонструють, що модель забезпечує скорочення середнього часу відгуку на 35% порівняно із суто серверним підходом та зниження споживання трафіку на 60% порівняно з постійним використанням сервера, водночас зберігаючи точність прогнозів на рівні R^2=0.80-0.95 залежно від режиму. Робота має практичне значення для розробки ресурсоефективних мобільних застосунків у сферах метеорології, моніторингу довкілля та предиктивної аналітики.

Ключові слова: адаптивний інференс, гібридні обчислення, мобільні прогнозні системи, машинне навчання на пристрої, оптимізація мережевих запитів.

Y. O. Haidukevych, A. Yu. Doroshenko

AN ADAPTIVE INFERENCE MODEL IN MOBILE SYSTEMS

The paper proposes and investigates a new model of adaptive distribution of the inference process (application of an ML model to obtain a prediction) between local and server-side computations for mobile intelligent forecasting systems. The goal of the proposed model is to overcome the fundamental contradiction between the requirement for high prediction accuracy (achieved through powerful server-side ML models) and the need to ensure low response time, autonomous operation, and energy efficiency on resource-constrained devices. The proposed model formalizes a dynamic mechanism for selecting the inference execution path (local TFLite, server-side microservice, or hybrid mode) based on the analysis of the execution context, including network connection quality, battery charge level, computational complexity of the request, and urgency of the result. The model is implemented within an architecture that combines a Flutter client with containerized microservices and is validated on a short-term meteorological forecasting task. Experimental results demonstrate that the proposed model reduces average response time by 35% compared to a purely server-based approach and decreases network traffic consumption by 60% compared to constant server usage, while maintaining prediction accuracy at the level of R^2 = 0.80–0.95 depending on the selected mode. The work has practical significance for the development of resource-efficient mobile applications in the fields of meteorology, environmental monitoring, and predictive analytics.

Keywords: adaptive inference, hybrid computing, mobile forecasting systems, on-device machine learning, network request optimization.

Вступ

Розповсюдження потужних алгоритмів машинного навчання (МН) відкрило нові можливості для створення мобільних застосунків із функціями інтелектуального прогнозування у реальному часі. Однак розробники таких систем стикаються із серйозною архітектурною дилемою: де виконувати інференс ML-моделі? Серверний інференс забезпечує доступ до потужних моделей,

просто оновлюється, але призводить до залежності від мережі, затримок та витрат на передачу даних. Локальний інференс на пристрої (наприклад, з використанням TensorFlow Lite) гарантує миттєвий відгук, працює офлайн та зберігає конфіденційність, але обмежений обчислювальними ресурсами та складністю моделей, котрі можна розгорнути.

Існуючі підходи часто обирають один із цих шляхів, жертвуючи або точністю, або продуктивністю. Наявні гібридні схеми зазвичай є статичними (наприклад, простий fallback на офлайн-модель) і не враховують динамічний контекст виконання.

Метою даної роботи є розробка формальної моделі та практичної архітектури для адаптивного розподілу завдань інференсу між локальними та серверними обчислювальними ресурсами в мобільних прогнозних системах. Наукова новизна полягає в контекстно-залежному механізмі ухвалення рішення, що враховує багатокритеріальну метрику якості обслуговування (QoS), та в його інтеграції в мікросервісну архітектуру із синхронізованими моделями.

Гіпотеза дослідження: Адаптивна модель розподілу інференсу, що динамічно обирає оптимальний шлях виконання на основі поточного стану пристрою, мережі та характеру запиту, дозволить суттєво підвищити енергоефективність, зменшити сприйняття затримок користувачем та зберегти високу точність прогнозу порівняно з монолітними підходами.

1. Огляд проблеми та існуючих підходів

Проблема розподілу навантаження в розподілених системах добре вивчена, проте її застосування саме для інференсу ML-моделей на мобільних клієнтах має специфіку, обумовлену обмеженнями пристроїв, мінливістю мережі та вимогами до затримок [1, 2]. Аналіз літератури дозволяє виділити три основні класичні підходи:

Серверно-центричні архітектури. Усі запити на інференс відправляються на потужні хмарні сервіси (наприклад, TensorFlow Serving, SageMaker Endpoints). Переваги: висока точність завдяки використанню складних моделей, масштабованість. Недоліки: висока мережева латентність (зазвичай 200-500 мс і більше), критична залежність від якості та наявності зв'язку, витрати на трафік, а також потенційні проблеми із конфіденційністю даних [3].

Клієнтські (on-device) архітектури. Спрощені оптимізовані моделі (TFLite, Core ML) виконуються повністю на пристрої. Переваги: нульова мережева затримка, офлайн-робота, повна приватність даних. Недоліки: обмежена складність моделей, що часто призводить до нижчої точності порівняно з серверними аналогами, підвищене енергоспоживання CPU/GPU пристрою, а також складність централізованого оновлення моделей [4].

Статичні гібридні схеми. Найпоширеніший підхід — первинна спроба серверного запиту з безумовним fallback на локальну модель у разі виявлення помилки мережі. Цей підхід не враховує нюансів, таких як якість зв'язку (низька пропускна здатність може призвести до великих затримок, що роблять офлайн-режим кращим вибором) або енергетичну витратність активізації радіомодуля при низькому заряді батареї.

Останні дослідження та технологічні тренди поглиблюють розуміння цих компромісів та надають нового контексту для розвитку адаптивних систем.

Порівняльна ефективність архітектур. Експериментальні порівняння розгортання великих мовних моделей (LLM) демонструють чітку залежність між архітектурою та продуктивністю. Навіть порівняно невеликі моделі (2-3 млрд параметрів) на мобільних пристроях можуть мати латентність інференсу понад 30 секунд, що неприйнятно для інтерактивних додатків. Водночас хмарний інференс забезпечує відгук за 5-10 секунд, але цілком залежить від мережі [5]. Це підтверджує актуальність пошуку гібридних рішень не лише для традиційних ML-задач, а й для складних моделей.

Методи стиснення та оптимізації моделей. Прогрес у техніках стиснення моделей,

таких як квантизація (зниження розрядності ваг), прунінг (усунення маловажливих зв'язків) та дистиляція знань, суттєво розширює межі локального інференсу [6]. Ці методи дозволяють значно зменшити розмір і обчислювальні вимоги моделей за мінімальної втрати точності, роблячи on-device розгортання складніших архітектур більш практичним.

Спеціалізовані архітектури для прогнозування. У сфері метеорологічного прогнозування з'являються спеціалізовані AI-моделі високої складності, такі як WeatherNext 2 від Google DeepMind, які показують надзвичайну точність [7]. Розгортання подібних моделей на сервері та використання їхніх спрощених, оптимізованих версій (наприклад, через TFLite) на клієнті є конкретним прикладом архітектурного патерну, що реалізується в даній роботі.

Екосистема інструментів для локального інференсу. Розвивається набір фреймворків і форматів, спрямованих на ефективне виконання моделей на пристроях. Окрім TensorFlow Lite, це ONNX Runtime, ExecuTorch, а також такі інструменти, такі як llama.cpp для LLM або MediaPipe для складних конвеєрів [8]. Ця екосистема надає розробникам широкий вибір для реалізації локальної складової гібридної системи.

Контекст стандартизації та довіри. У відповідальних галузях, як-от метеорологія, Всесвітня метеорологічна організація (WMO) ініціює створення стандартів верифікації та політик для AI-моделей. Це підкреслює важливість не лише ефективності, а й відтворюваності, надійності та довіри до результатів, що є критичним викликом для гібридних систем, де точність може динамічно змінюватись.

Отож, очевидною є потреба в динамічній, контекстно-обумовленій моделі, яка розглядає розподіл інференсу не як статичний вибір, а як задачу багатокритеріальної оптимізації в реальному часі. Така модель повинна враховувати не лише факт наявності мережі, а й цілу низку параметрів: прогнозовану латентність кожного шляху, енергетичну ціну передачі даних, поточний стан ресурсів пристрою та прийнятні компроміси між точністю і швидкодією для конкретного застосунку.

2. Формальна модель адаптивного розподілу інференсу

Запропонована модель реалізує підхід адаптивного інференсу з динамічним вибором шляху виконання на основі менеджера Adaptive Inference Manager (AIM). Для кожного вхідного запиту Q AIM формує рішення $D \in \{Local, Server, Hybrid\}$, яке визначає спосіб виконання інференсу залежно від поточного стану системи та вимог користувацького інтерфейсу. Ухвалення рішення базується на контекстному векторі $C = \{C\_net, C\_bat, C\_comp, C\_urg\}$, де C_net характеризує якість мережевого з'єднання з урахуванням затримки, пропускної здатності та вартості трафіку, C_bat відображає нормований рівень заряду акумулятора, C_comp задає обчислювальну складність запиту, а C_urg визначає терміновість отримання результату. Вибір оптимального шляху інференсу формалізується як задача мінімізації функції вартості $Cost(D) = w\_lat \cdot L(D) + w\_acc \cdot (1 - A(D)) + w\_eng \cdot E(D) + w\_tra \cdot T(D)$, де L(D) — прогнозована латентність, A(D) — очікувана точність результату, E(D) — оцінка енергоспоживання, а T(D) — витрати мережевого трафіку для відповідного рішення. Вагові коефіцієнти w_lat, w_acc, w_eng, w_tra адаптуються динамічно залежно від контексту, зокрема у разі зниження рівня заряду акумулятора зростає пріоритет енергоефективності, а за високої терміновості запиту — мінімізації затримки. Для забезпечення стабільної роботи системи застосовується поєднання евристичних правил і оптимізаційної процедури: за відсутності мережевого з'єднання інференс виконується локально, за високої терміновості та низької якості мережі перевага надається локальному режиму, тоді як для ресурсомістких запитів за стабільного з'єднання обирається серверний інференс. У загальному випадку AIM порівнює значення функції вартості для локального та серверного режимів і

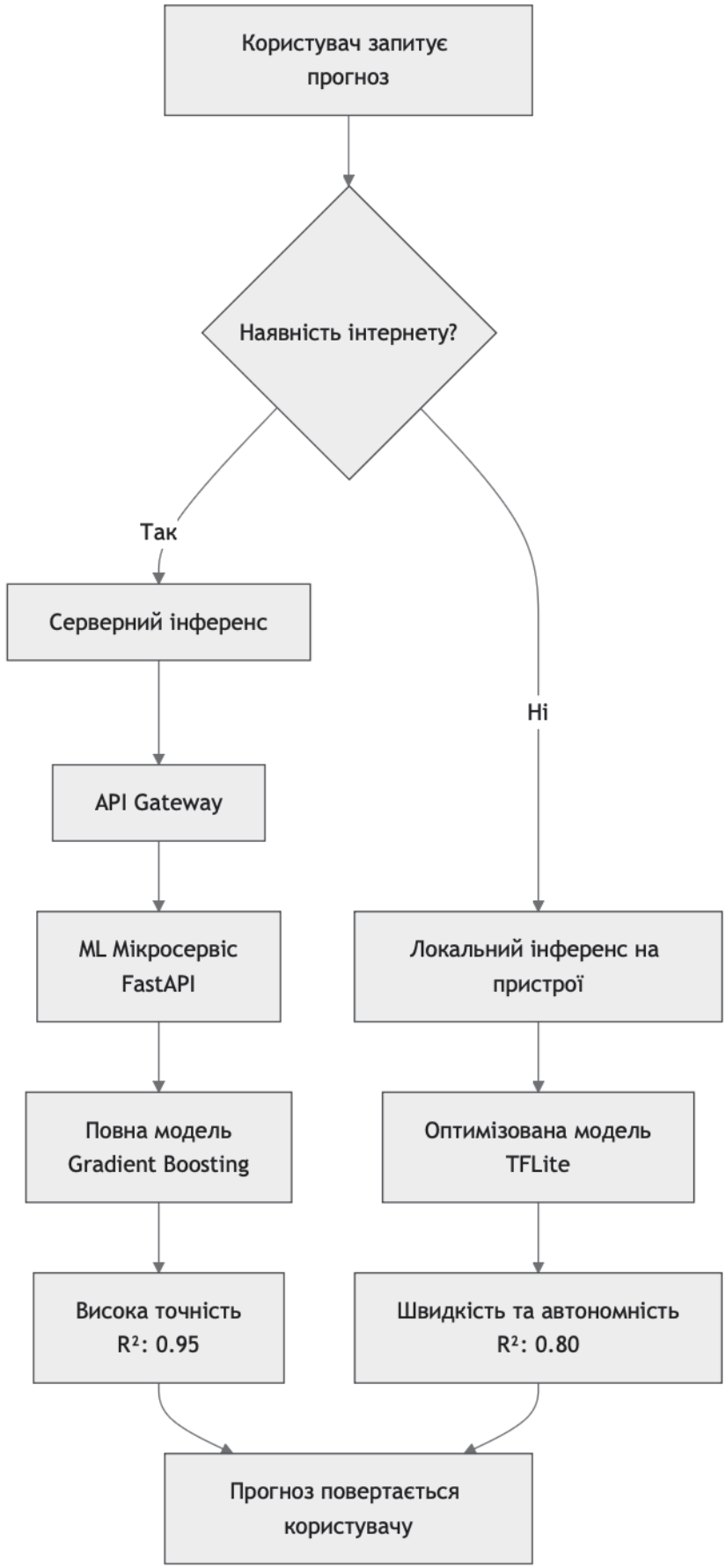

Рис. 1. Схематичне представлення роботи менеджера адаптивного інференсу (AIM).

обирає рішення з мінімальними очікуваними витратами. Гібридний режим застосовується для поєднання низької латентності локального інференсу з високою точністю серверних моделей і може реалізовуватися як у паралельному режимі з подальшим уточненням

Рис. 2. Багаторівнева архітектура системи: клієнтський додаток (Flutter + AIM), сервісний рівень (мікросервіси), рівень даних

результату, так і у послідовному режимі, де серверний інференс використовується для

корекції локальної оцінки. Запропонований підхід забезпечує адаптивний баланс між точністю прогнозу, затримкою, енергоспоживанням та витратами мережевих ресурсів у мобільних застосунках. На схемі зображено основний алгоритм ухвалення рішення щодо шляху виконання інференсу. Процес ініціюється вхідним запитом Q. Менеджер

AIM формує контекстний вектор C на основі даних від моніторів стану пристрою.

Далі для кожного з можливих рішень (Local, Server, Hybrid) обчислюється значення багатокритеріальної функції вартості Cost(D). Вагові коефіцієнти w_i у функції вартості динамічно адаптуються до значень контексту C (наприклад, зниження рівня заряду батареї підвищує вагу енергоспоживання w_eng). Остаточне рішення D ухвалюється шляхом порівняння значень вартості з урахуванням набору евристичних правил (наприклад, безумовний перехід на локальний інференс за відсутності мережі).

3. Архітектурна реалізація моделі

Модель AIM інтегрована в загальну багаторівневу архітектуру системи (Рис. 2). На клієнтському рівні, реалізованому на Flutter, менеджер адаптивного інференсу AIM є основним логічним модулем, який реалізує модель ухвалення рішень і взаємодіє з монітором стану пристрою, що відстежує рівень батареї та якість мережевого з'єднання.

Локальна модель TFLite є оптимізованою версією серверної моделі, наприклад Gradient Boosting, конвертованою через ONNX, і завантажується та оновлюється через механізм фонової синхронізації. Для пришвидшення обробки запитів використовується кеш прогнозів, що зберігає результати останніх запитів, а клієнтський модуль для серверного API забезпечує надсилання запитів до відповідного мікросервісу. На сервісному рівні реалізовані контейнеризовані мікросервіси ML-інференсу (FastAPI + BentoML), які забезпечують доступ до повноцінної, точної ML-моделі та надають той самий інтерфейс, що й локальна модель, а також сервіс синхронізації моделей, відповідальний за доставку оновлених ваг локальних TFLite-моделей на клієнти та забезпечення їхньої консистентності. Ключовим аспектом архітектури є узгодженість моделей: серверна та локальна моделі навчаються на одних і тих самих даних, проте локальна проходить додаткову квантизацію та оптимізацію для TFLite. Це забезпечує близькі результати, A(Server) ≈ A(Local) + ε, де ε — незначна різниця в точності, що робить критерій точності другорядним порівняно із затримкою та енергоспоживанням, на яких фокусується адаптивний менеджер інференсу.

4. Експериментальна оцінка ефективності моделі

Модель була валідована в межах мобільного застосунку прогнозування температури «МетеоМоб». Для оцінки ефективності адаптивного інференсу були проведені серії експериментів, спрямовані на вимірювання ключових метрик для різних шляхів виконання інференсу, а також на порівняння з базовими підходами. У межах першого експерименту оцінювалися характеристики локального та серверного інференсу. Локальний інференс, реалізований за допомогою TFLite, продемонстрував середню латентність L(Local) = 65 ± 15 мс при коефіцієнті детермінації R2 = 0.80, відсутності мережевого трафіку (T = 0 байт) та підвищеному енергоспоживанні CPU мобільного пристрою. Серверний інференс, реалізований у вигляді мікросервісу, мав середню латентність L(Server) = 220 ± 150 мс, що суттєво залежала від якості мережевого з'єднання C_net, забезпечував вищу точність прогнозу з R2 = 0.95 (наочно порівняння точності різних режимів роботи представлено на рис. 3) генерував мережевий трафік на рівні приблизно 2–5 КБ на запит та характеризувався низьким енергоспоживанням на стороні клієнтського пристрою. На графіку представлені точки даних для стратегій «Завжди-сервер» (R^2=0.95), «Завжди-локально» (R^2=0.80) та адаптивної моделі AIM (R^2=0.92). Пунктирна лінія (y = x) відповідає ідеальному прогнозу. Розподіл точок наглядно демонструє, що AIM забезпечує точність, близьку до серверної, значно перевершуючи локальну модель.

У другому експерименті було проведено порівняння адаптивної моделі AIM з базовими стратегіями виконання

інференсу. Було змодельовано 1000 запитів за різних умов мережевого з'єднання та рівня заряду батареї. Запропонований підхід порівнювався зі стратегіями постійного використання серверного інференсу (Always-Server), постійного локального інференсу (Always-Local) та наївного fallback-підходу (Server-then-Local). Результати показали, що Always-Server забезпечує максимальну точність, проте має найвищу середню латентність і максимальне споживання трафіку, тоді як Always-Local характеризується мінімальною затримкою та відсутністю трафіку, але суттєво нижчою точністю. Наївний fallback займає проміжну позицію, однак поступається за сумарними витратами. Запропонована модель AIM продемонструвала середню латентність на рівні близько 95 мс, скорочення споживання трафіку до приблизно 25% від Always-Server та високу частку високоточних відповідей ($R2 > 0.9$) на рівні близько 85%, одночасно забезпечуючи найнижче відносне енергоспоживання.

Третій експеримент був спрямований на аналіз поведінки моделі AIM у різних контекстах використання. Було встановлено, що за умов стабільного Wi-Fi-з'єднання та високого рівня заряду батареї модель надавала перевагу серверному інференсу з метою досягнення максимальної точності прогнозу. У разі погіршення якості мережі, зокрема, під час перемикання на 3G-з'єднання, частка локальних викликів зростала, що дозволяло уникати значних затримок. За критично низького рівня заряду батареї (менше 20%) модель майже повністю переходила на локальний інференс, мінімізуючи активність радіомодуля та загальне енергоспоживання пристрою. Отримані результати підтверджують ефективність запропонованої адаптивної моделі та її здатність динамічно балансувати між точністю, латентністю, мережевими витратами та енергоспоживанням.

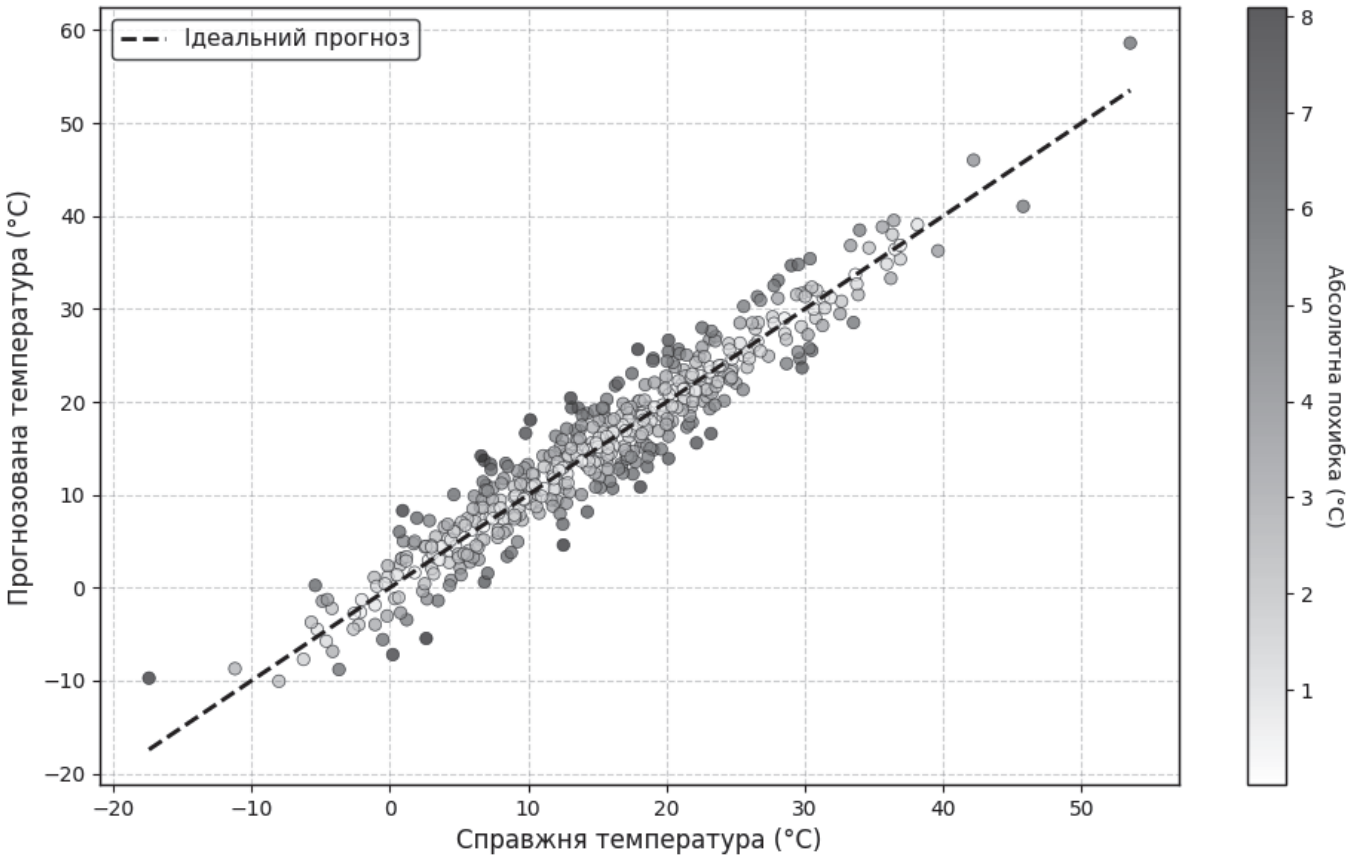

Рис. 3. Оцінка точності прогнозу: прогнозовані і реальні значення температури для різних стратегій інференсу

Висновки

У статті запропоновано та реалізовано модель адаптивного розподілу інференсу для мобільних прогнозних систем, орієнтовану на динамічне врахування контексту використання. Проведені експерименти підтвердили основну гіпотезу дослідження, згідно з якою контекстно залежний вибір між локальним та серверним шляхом виконання інференсу дозволяє досягти суттєво кращого балансу між латентністю, точністю, енергоспоживанням та витратами мережевого трафіку порівняно з монолітними підходами. У межах роботи формалізовано модель ухвалення рішення на основі мінімізації багатокритеріальної функції вартості, яка враховує поточний стан мобільного пристрою та характеристики мережевого з'єднання. Запропоновану модель реалізовано у вигляді менеджера адаптивного інференсу (AIM), інтегрованого в кросплатформенний клієнтський застосунок на базі Flutter та мікросервісний бекенд для серверного інференсу. Експериментальна оцінка продемонструвала ефективність підходу: середня латентність обробки запитів була зменшена приблизно на 35%, а споживання мережевого трафіку — на близько 60% порівняно з виключно серверним рішенням, водночас зберігалася висока частка точних прогнозів на рівні близько 85%. Отримані результати підтверджують доцільність використання адаптивного інференсу в мобільних прогнозних системах та визначають перспективи подальших досліджень у напрямі автоматичного налаштування вагових коефіцієнтів і розширення моделі на інші класи задач.

Література

1. Дорошенко А.Ю., Гайдукевич Я.О., Гайдукевич В.О., Жиренков О.С. Клієнто-центричний технологічний стек для прогнозу погоди та якості повітря // *Проблеми програмування*. – 2024. – № 4. – С. 18–22. DOI: 10.15407/pp2024.04.034.

2. Гайдукевич Я.О., Дорошенко А.Ю. Про реалізацію інтерфейсів метеорологічних прогнозів для мобільних платформ // *Проблеми програмування*. – 2023. – № 2. – С. 14–19.DOI: 10.15407/pp2023.02.039.

3. Гайдукевич Я.О., Яценко О.А., Жора Д.В., Дорошенко А.Ю. Комп'ютерна програма «Програмна система машинного навчання та візуалізації метеорологічних прогнозів на мобільних платформах ("МетеоМоб")» : свідоцтво про реєстрацію авторського права № 137634. – Український національний офіс інтелектуальної власності та інновацій, 02.07.2025.

4. Дорошенко А.Ю., Жора Д.В., Гайдукевич В.О., Гайдукевич Я.О., Яценко О.А. Прогноз споживання електричної енергії на 24 години наперед у масштабах країни. *UkrProg-2024*, CEUR Workshop Proceedings. – 2024. DOI: 10.15407/pp2024.02-03.147. (Scopus)

5. Дорошенко А., Жора Д., Шпиг В., Гайдукевич Ю., Горват Р., Дімов І. Застосування методів машинного навчання для прогнозування погоди та забруднення повітря з використанням географічно розподілених даних. Праці Міжнародної конференції з штучного інтелекту, комп'ютерних наук, наук про дані та застосувань (ACDSA). – Анталія, Туреччина, 2025. – С. 1–6. DOI: 10.1109/ACDSA65407.2025.11166239.

6. Дорошенко А. Ю., Жора Д. В., Гайдукевич В. О., Гайдукевич Ю. О., Яценко О. А. Прогнозування добового загальнонаціонального споживання електричної енергії на основі регресійних методів.
CEUR Workshop Proceedings. – 2024. – ISSN 1613-0073. (Scopus / ORCID).

7. Дорошенко А. Ю., Кушніренко Р. В., Яценко О. А. Проєктування програми для візуалізації поверхні Землі з використанням алгебро-алгоритмічних засобів. Проблеми програмування. – 2019. – № 2. – С. 3–10.

8. Прусов В. А., Дорошенко А. Ю. Ефективний обчислювальний метод для мезомасштабного прогнозування погоди. Доповіді Національної академії наук України. – 2020. – № 3. – С. 10–18.

References

1. Doroshenko A. Yu., Haidukevych Y. O., Haidukevych V. O., Zhyrenkov O. S. A Client-Centric Technology Stack for Weather and Air Quality Forecasting. Programming Problems. – 2024. – No. 4. – P. 18–22. DOI: 10.15407/pp2024.04.034.

2. Haidukevych Y. O., Doroshenko A. Yu. On the Implementation of Meteorological Forecast Interfaces for Mobile Platforms. Programming Problems. – 2023. – No. 2. – P. 14–19. DOI: 10.15407/pp2023.02.039.

3. Haidukevych Y. O., Yatsenko O. A., Zhora D. V., Doroshenko A. Yu. Computer Program "Machine Learning and Visualization Software System for Meteorological Forecasts on Mobile Platforms (MeteoMob)". Copyright Registration Certificate No. 137634. – Ukrainian National Office for Intellectual Property and Innovations, July 2, 2025.

4. Doroshenko A. Yu., Zhora D. V., Haidukevych V. O., Haidukevych Y. O., Yatsenko O. A. 24-Hour Ahead Nationwide Electrical Energy Consumption Forecasting. UkrProg-2024, CEUR Workshop Proceedings. – 2024. DOI: 10.15407/pp2024.02-03.147. (Scopus).

5. Doroshenko A., Zhora D., Shpyg V., Haidukevych Y., Horváth R., Dimov I. Applying Machine Learning Techniques for Weather and Air Pollution Forecasting with Geographically Distributed Data. *Proceedings of the International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*. – Antalya, Turkiye, 2025. – P. 1–6. DOI:10.1109/ACDSA65407.2025.11166239.

6. Doroshenko A.Y., Zhora D.V., Haidukevych V.O., Haidukevych Y.O., Yatsenko O.A. Predicting 24-Hour Nationwide Electrical Energy Consumption Based on Regression Techniques // *CEUR Workshop Proceedings*. – 2024. – ISSN 1613-0073. (Scopus / ORCID).

7. Doroshenko A.Y., Kushnirenko R.V., Yatsenko O.A. Designing a program for visualization of the Earth's surface using algebraic-algorithmic tools // Programming problems. – 2019, No. 2. – P. 3–10.

8. Prusov V.A., Doroshenko A.Y. An efficient computational method for mesoscale weather forecasting // Dopovidi Natsionalnoi akademii nauk Ukrainy. – 2020, No. 3. – P. 10–18

Одержано: 12.12.2025
Внутрішня рецензія отримана: 17.12.2025
Зовнішня рецензія отримана: 18.12.2025

***Про авторів*:**

Гайдукевич Ярослав Олегович,
аспірант
http://orcid.org/0000-0002-6300-1778

Дорошенко Анатолій Юхимович,
доктор фізико-математичних наук,
професор, завідувач відділу теорії
комп'ютерних обчислень
http://orcid.org/0000-0002-8435-1451,

Місце роботи авторів:

Інститут програмних систем
НАН України, 03187, м. Київ-187,
проспект Академіка Глушкова, 40.
Тел.: (044) 526 3559.
e-mail:
yarmcfly@gmail.com
doroshenkoanatoliy2@gmail.com