DOI: https://doi.org/10.36910/6775-2524-0560-2026-63-27
УДК 004.032.26:004.75
Орел Руслан Леонідович, аспірант
https://orcid.org/0009-0008-2353-1329
Розломій Інна Олександрівна, к.т.н., доцент
https://orcid.org/0000-0001-5065-9004
Черкаський державний технологічний університет, м. Черкаси, Україна

АДАПТИВНИЙ РОЗПОДІЛ НЕЙРОМЕРЕЖЕВОГО ВИСНОВКУ В МУЛЬТИМОДАЛЬНИХ АСИСТЕНТАХ З УРАХУВАННЯМ ЕНЕРГО-ЧАСОВИХ ОБМЕЖЕНЬ AIOT-МЕРЕЖ

Орел Р.Л., Розломій І.О. Адаптивний розподіл нейромережевого висновку в мультимодальних асистентах з урахуванням енерго-часових обмежень AIoT-мереж. Розвиток інтелектуальних систем та впровадження парадигми Інтернету речей (AIoT) супроводжується стрімким поширенням мультимодальних асистентів, здатних синхронно обробляти гетерогенні потоки даних, такі як відео високої роздільної здатності, просторове аудіо та текст. Однак їхня інтеграція стикається з фундаментальними технологічними суперечностями: громіздкі глибокі нейронні мережі (зокрема Transformer та CNN) потребують значних обчислювальних ресурсів, якими не володіють периферійні (edge) пристрої. Використання класичного локального підходу (Edge-only) призводить до швидкого виснаження акумуляторних батарей та недопустимих затримок висновку. З іншого боку, централізований хмарний підхід (Cloud-only) генерує експоненційне навантаження на бездротові канали зв'язку через безперервну трансляцію сирих мультимедійних даних, що створює мережеві «вузькі місця» та порушує суворі вимоги щодо конфіденційності користувачів. У даній статті вирішується науково-практична проблема оптимізації мультимодального висновку шляхом розробки методу динамічного розподілу обчислювального навантаження в колаборативній архітектурі edge-cloud. В основу рішення покладено технологію Split Computing, яка передбачає просторове розрізання графа нейромережі. Запропоновано нову математичну модель багатокритеріальної оптимізації, яка використовує зважену цільову функцію для балансування між загальною затримкою системи та сумарним енергоспоживанням пристрою. На базі цієї моделі розроблено легковаговий диспетчерський алгоритм, що працює в режимі реального часу. На відміну від існуючих рішень зі статичною точкою розрізу, розроблений алгоритм враховує ефект інформаційного вузького місця, безперервно оцінює поточну пропускну здатність радіоканалу та критичний рівень заряду батареї, адаптивно зміщуючи точку розрізу індивідуально для кожної модальності. Проведене імітаційне моделювання підтвердило високу результативність запропонованого підходу. Доведено, що в умовах нестабільної мережі динамічний метод дозволяє зменшити час відгуку системи на 28–35% (з піковим виграшем до 45%) порівняно з Cloud-only сценарієм. Крім того, за рахунок мінімізації роботи енергоємного радіомодуля при передачі стиснених проміжних тензорів, енергоефективність периферійного вузла підвищується на 18–25% відносно моделей статичного розподілу. Отримані результати формують надійне наукове підґрунтя для розробки автономних, конфіденційних та високошвидкісних мультимодальних систем реального часу з підвищеною стійкістю до деградації комунікаційної інфраструктури.

Ключові слова: мультимодальні асистенти, інтернет речей, обчислення периферія-хмара, розподіл обчислень, глибокі нейронні мережі, енергоефективність, багатокритеріальна оптимізація.

Orel R., Rozlomii I. Adaptive neural network inference partitioning in multimodal assistants under energy and latency constraints of AIoT networks. The advancement of intelligent systems and the widespread adoption of the Artificial Intelligence of Things (AIoT) paradigm are accompanied by the rapid proliferation of multimodal assistants capable of synchronously processing heterogeneous data streams, such as high-resolution video, spatial audio, and text. However, their integration faces fundamental technological contradictions: massive deep neural networks (specifically Transformers and CNNs) demand significant computational resources that edge devices lack. Relying on a classical local approach (Edge-only) leads to rapid battery depletion and unacceptable inference latency. Conversely, the centralized cloud computing approach (Cloud-only) generates an exponential load on wireless communication channels due to the continuous transmission of raw multimedia data, creating network bottlenecks and violating strict user privacy requirements. This paper addresses the scientific and practical problem of optimizing multimodal inference by developing a dynamic computation offloading method within a collaborative edge-cloud architecture. The solution is based on Split Computing technology, which involves the spatial partitioning of a neural network graph. A novel multi-objective optimization mathematical model is proposed, utilizing a weighted cost function to balance total system latency and overall device energy consumption. Based on this model, a lightweight, real-time scheduling algorithm is developed. Unlike existing solutions with a static split point, the developed algorithm leverages the information bottleneck effect, continuously evaluates the current radio channel bandwidth and critical battery levels, and adaptively shifts the split point individually for each data modality. Comprehensive simulation modeling confirmed the high efficacy of the proposed approach. It is proven that under fluctuating network conditions, the dynamic method reduces system response time by 28–35% (with peak gains up to 45%) compared to the Cloud-only scenario. Furthermore, by minimizing the operation of the energy-intensive radio module during the transmission of compressed intermediate tensors, the energy efficiency of the edge node is increased by 18–25% relative to static distribution models. The obtained results establish a robust scientific foundation for the development of autonomous, privacy-preserving, and high-speed real-time multimodal systems with enhanced resilience to communication infrastructure degradation.

Key words: multimodal assistants, AIoT, edge-cloud computing, Split Computing, computation offloading, deep neural networks, energy efficiency, multi-objective optimization.

Постановка проблеми та її зв'язок із важливими науковими чи практичними завданнями. Сучасний етап розвитку інформаційних технологій невіддільний від парадигми AIoT (Artificial Intelligence of Things), яка передбачає глибоку інтеграцію алгоритмів штучного інтелекту в розподілену інфраструктуру Інтернету речей. Ключовим елементом цієї екосистеми стають мультимодальні асистенти — розумні агенти, здатні в режимі реального часу синхронно обробляти, аналізувати та синтезувати гетерогенні потоки даних: відео, просторове аудіо, текст і сенсорну інформацію. Стрімке зростання ролі таких систем у повсякденному житті, концепціях «розумного дому» та промисловій автоматизації вимагає від них безперебійної роботи, глибокого розуміння контексту та миттєвої реакції на запити користувача [1, 2, 3].

Водночас розвиток і масове впровадження мультимодальних асистентів гальмується фундаментальним технологічним протиріччям, пов'язаним з обмеженнями наявних обчислювальних архітектур. З одного боку, забезпечення високої точності розпізнавання (наприклад, одночасний трекінг емоцій на відео та семантичний аналіз голосу) вимагає використання громіздких глибоких нейромереж — багатошарових архітектур Transformer або складних згорткових мереж (CNN). Виконання висновку (inference) таких ресурсоємних моделей виключно на периферійних вузлах (Edge-only підхід) є критично неефективним. Обмежені обчислювальні потужності апаратного забезпечення призводять до недопустимих затримок (latency), а інтенсивні матричні обчислення стрімко виснажують акумуляторні батареї автономних пристроїв [4, 5].

З іншого боку, традиційний централізований підхід, за якого всі зібрані дані пересилаються для обробки на віддалені сервери (Cloud-only підхід), також вичерпав свій потенціал і не підходить для мультимодальних систем реального часу. Безперервна трансляція нестиснених або масивно об'ємних відео- та аудіопотоків створює експоненційне навантаження на телекомунікаційні канали зв'язку, що неминуче призводить до мережевих затримок та "вузьких місць" [6]. Крім того, передача сирих потокових даних із камер та мікрофонів користувачів у хмарне середовище порушує суворі сучасні вимоги до конфіденційності та створює ризики витоку приватної інформації [7].

З огляду на виявлену суперечність, виникає гостра практична та наукова необхідність у переході від ізольованих парадигм до систем колаборативного інтелекту (Collaborative Intelligence) [8]. У такій архітектурі периферійний пристрій (edge) та хмарний сервер (cloud) повинні функціонувати не як розділені сутності, а як єдиний синергетичний механізм. Відповідно, важливим науковим завданням є створення адаптивних методів розподілу обчислень, які б дозволили динамічно балансувати навантаження між краєм та хмарою залежно від поточного стану мережі та ресурсів пристрою [13]. Вирішення цього завдання є критично важливим для мінімізації часу відгуку в системах реального часу, раціонального використання енергоресурсів пристроїв AIoT-мереж та забезпечення належного рівня захисту даних у мультимодальних асистентах.

Аналіз останніх досліджень та публікацій. У сучасній науковій літературі проблема оптимізації висновку (inference) глибоких нейронних мереж для пристроїв Інтернету речей традиційно розглядається крізь призму двох полярних парадигм: Cloud-only та Edge-only. Хмаро-орієнтований підхід (Cloud-only) забезпечує високу точність та швидкість обчислень завдяки потужним серверам, проте дослідження [6] доводять, що він є вразливим до мережевих затримок, особливо при передачі важкого нестисненого мультимедійного трафіку. Як альтернатива, парадигма Edge AI [3, 4] фокусується на локальному виконанні моделей за допомогою методів компресії (квантування, прунінг, дистиляція знань). Проте, спроби застосувати такі методи до складних архітектур (зокрема, Transformer) призводять до значної деградації точності розпізнавання та швидкого вичерпання енергетичних ресурсів автономних вузлів [5]. Отже, класичні підходи вичерпали свій потенціал щодо забезпечення балансу між швидкістю, енергоефективністю та точністю для мультимодальних задач.

Як компромісне рішення, що нівелює недоліки обох екстремумів, активно розвивається технологія колаборативного інтелекту, зокрема концепція Split Computing (розподілених обчислень). Фундаментальні основи цього напрямку були закладені в роботі Kang et al. [9], де було представлено фреймворк Neurosurgeon. Автори вперше запропонували декомпозицію графа нейронної мережі на дві частини: початкові шари виконуються на мобільному пристрої, а проміжні карти ознак (feature maps) передаються на сервер для завершення обчислень. Подальший розвиток ця ідея отримала в роботах Teerapittayanon et al. [10], які запропонували розподілені глибокі

нейронні мережі (DDNN) для ієрархічної архітектури "пристрій-край-хмара" з можливістю раннього виходу (early exiting), та в оглядовому дослідженні [8], що формалізували математичні моделі сумісного інференсу.

Виділення невирішеної частини загальної проблеми. Незважаючи на значний прогрес у галузі Split Computing, ґрунтовний аналіз публікацій [8, 9, 10] дозволяє виділити низку суттєвих недоліків існуючих підходів. По-перше, більшість сучасних рішень орієнтовані на пошук статичної точки розрізу (static split point) на етапі компіляції або початкового розгортання моделі. Вони розраховані на стабільне середовище і не здатні адаптуватися до флуктуацій пропускної здатності мережі в режимі реального часу. По-друге, існуючі моделі розподілу фокусуються переважно на унімодальних задачах (виключно комп'ютерний зір або обробка природної мови). Вони не враховують специфіку мультимодальних потоків, де паралельно обробляються різні типи даних з різною пріоритетністю, об'ємом та чутливістю до затримок (наприклад, відеовимірювання може вимагати більшої пропускної здатності, ніж аудіо).

Таким чином, відсутність комплексних методів динамічного (адаптивного) перерозподілу обчислень для гетерогенних мультимодальних потоків в умовах нестабільності AIoT-мереж формує невирішену проблему, вирішенню якої присвячено дане дослідження.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Для вирішення сформульованої проблеми перерозподілу обчислювального навантаження було розроблено концептуальну архітектуру системи мультимодального колаборативного висновку (Collaborative Inference). В основу запропонованого підходу покладено парадигму Split Computing, що передбачає просторову декомпозицію графа глибокої нейронної мережі (Deep Neural Network, DNN) між локальним периферійним пристроєм та віддаленим хмарним сервером [11].

На відміну від традиційних унімодальних архітектур, розроблена система орієнтована на паралельну обробку гетерогенних потоків даних (наприклад, візуального потоку V та аудіопотоку A). Оскільки ці модальності мають різний ступінь інформаційної щільності та різні вимоги до обчислювальних ресурсів, архітектура використовує механізм роздільної екстракції ознак (feature extraction) на ранніх етапах із подальшим злиттям (mid-level fusion) у хмарі.

Загальна схема процесу обробки мультимодальних даних у запропонованій архітектурі складається з чотирьох послідовних етапів.

Спочатку відбувається введення мультимодальних даних (Data Acquisition). AIoT-пристрій (смарт-камера, мультимодальний асистент) синхронно фіксує сирі дані з різних сенсорів. На цьому етапі формуються вхідні тензори відеокадрів X_V та аудіоспектрограм X_A.

Далі працюють початкові шари обробки на Edge (Early Edge Processing). Замість відправки сирих даних у мережу, перші k шарів нейронної мережі (із загальної кількості N шарів) виконуються безпосередньо на локальному процесорі (NPU/GPU) периферійного пристрою. Для кожної модальності може існувати своя незалежна точка розрізу: k_V для відео та k_A для аудіо. На цих початкових етапах мережа (зазвичай згорткові шари для відео та 1D-CNN або ранні шари Transformer для аудіо) витягує базові низькорівневі ознаки, а саме контури, текстури, фонетичні патерни.

Наступним етапом є передача проміжного тензора (Feature Map Transmission). Результатом обчислень на edge-пристрої є не кінцевий результат, а багатовимірні карти ознак (intermediate feature maps або проміжні тензори). Завдяки ефекту інформаційного «вузького місця» (information bottleneck), обсяг цих тензорів у байтах на певних шарах мережі стає значно меншим, ніж обсяг початкових сирих мультимедійних файлів [12]. Стиснені тензори Z_V та Z_A передаються через бездротову мережу до хмарного сервера.

І останнім етапом є фінальний висновок у хмарі (Cloud Inference & Fusion). Хмарний сервер, володіючи потужними обчислювальними ресурсами, приймає проміжні тензори та подає їх на вхід наступним шарам моделі (від $k+1$ до N). На цьому етапі відбувається семантичне об'єднання мультимодальних ознак (модуль Fusion) та виконуються найбільш ресурсоємні матричні обчислення глибоких шарів (наприклад, механізми Self-Attention). Фінальним результатом є готовий висновок (класифікація наміру користувача, розпізнана команда тощо), який повертається на edge-пристрій у вигляді легкої текстової або керуючої команди.

Ключовою гіпотезою даного дослідження є те, що точки розрізу нейромережі (k_V та k_A) не повинні бути статичними. Архітектура передбачає наявність спеціального модуля моніторингу

(Resource Profiler), який безперервно оцінює стан радіоканалу та рівень заряду батареї, дозволяючи системі динамічно зміщувати точку розрізу в реальному часі.

Для кількісної оцінки ефективності розробленої архітектури та реалізації механізму адаптивного перемикання було побудовано математичну модель. Основна мета моделі полягає в тому, що потрібно формалізувати метрики загальної затримки (latency) та енергоспоживання пристрою залежно від обраної точки розрізу нейромережі k, де k — порядковий номер шару глибокої моделі ($k \in K$, $K = \{0,1,2,\dots,N\}$). Слід зазначити, що крайові значення $k = 0$ та $k = N$ відповідають класичним парадигмам Cloud-only (передача сирих даних) та Edge-only (повне локальне обчислення) відповідно [13].

Час відгуку є критичним параметром для мультимодальних асистентів реального часу. Загальна затримка системи T_{total} при виборі точки розрізу k та поточній пропускній здатності бездротового каналу R формується з трьох ключових компонентів (1).

$$T_{total}(k, R) = T_{edge}(k) + \frac{D(k)}{R} + T_{cloud}(k) \quad (1)$$

Тут $T_{edge}(k)$ – час обчислення перших k шарів на процесорі або нейроприскорювачі (NPU) периферійного пристрою, $D(k)$ – розмір проміжного тензора (карти ознак) на виході k-го шару в байтах, R – поточна пропускна здатність телекомунікаційної мережі (наприклад, Wi-Fi або 5G) у байтах за секунду, $T_{cloud}(k)$ – час виконання залишкових шарів (від $k + 1$ до N) на потужному хмарному сервері.

Згідно з дослідженнями [13, 14], залежність $D(k)$ від номера шару не є лінійною. На ранніх етапах обробки відеоданих розмір тензора може навіть перевищувати розмір початкового кадру, проте на глибших шарах обсяг даних різко скорочується.

Обмежений ресурс акумуляторних батарей AIoT-пристроїв вимагає суворого контролю за енерговитратами. Енергоспоживання периферійного вузла E_{total} складається з витрат на локальні матричні обчислення та витрат радіомодуля на передачу даних [14] (2).

$$E_{total}(k) = E_{comp}(k) + E_{trans}(k) \quad (2)$$

Тут $E_{comp}(k)$ – енергія, витрачена локальним процесором на пряме поширення (forward pass) через перші k шарів, $E_{trans}(k)$ – енергія, витрачена мережевим інтерфейсом на трансляцію обсягу даних $D(k)$ у хмару.

Збільшення k призводить до зростання $E_{comp}(k)$, але зазвичай експоненційно зменшує $E_{trans}(k)$ завдяки ефекту семантичного стиснення всередині мережі.

Оскільки зменшення затримки та зменшення енергоспоживання часто є конфліктуючими цілями (наприклад, швидка передача через 5G економить час, але різко витрачає батарею), задача пошуку ідеальної точки розрізу формулюється як мінімізація зваженої суми цих двох метрик [15].

Оптимальна точка розрізу k^* розраховується за формулою (3):

$$k^* = \underset{k \in K}{\operatorname{argmin}} \left(\alpha \cdot \hat{T}_{total}(k) + \beta \cdot \hat{E}_{total}(k) \right) \quad (3)$$

Тут $\hat{T}_{total}(k)$ та $\hat{E}_{total}(k)$ – нормалізовані значення затримки та енергоспоживання (приведені до єдиної шкали $[0,1]$, щоб уникнути домінування величин з різними одиницями виміру, такими як секунди та джоулі), α та β – вагові коефіцієнти (де $\alpha + \beta = 1$), що визначаються політикою системи. Наприклад, якщо рівень заряду батареї падає нижче 15%, система може динамічно збільшити β, перемикаючи пріоритет на енергозбереження.

Саме ця оптимізаційна задача лежить в основі алгоритму адаптивного розподілу обчислень, що дозволяє мультимодальному асистенту підлаштовуватись під змінні умови середовища.

Теоретична постановка задачі багатокритеріальної оптимізації потребує ефективного механізму імплементації для роботи в режимі реального часу. Оскільки умови середовища в мережах AIoT (пропускна здатність радіоканалу, завантаженість хмарного сервера, рівень заряду акумулятора) постійно флуктуюють, статичний вибір точки розрізу нейромережі є неефективним. Для забезпечення безперервної роботи мультимодального асистента було розроблено алгоритм динамічного (адаптивного) вибору точки розрізу k^*.

Відповідно до сучасних підходів у галузі мобільних обчислень [16], запропонований алгоритм функціонує у два етапи.

Спочатку відбувається офлайн-профілювання (Offline Profiling). Виконується одноразово перед розгортанням системи. На цьому етапі для кожного шару нейромережі k вимірюються та

заносяться до спеціальної пошукової таблиці (Lookup Table) константні значення: розмір вихідного тензора $D(k)$, очікуваний час локального виконання $\hat{T}_{edge}(k)$ та енерговитрати $\hat{E}_{comp}(k)$.

А далі – онлайн-моніторинг та диспетчеризація (Online Scheduling). Виконується безперервно під час роботи асистента. Спеціальний легковаговий фоновий процес (Resource Profiler) періодично оцінює стан мережі без створення надлишкового трафіку [17].

Логіку роботи алгоритму диспетчеризації в реальному часі можна формалізувати за допомогою псевдокоду. Вхідними даними є множина можливих точок розрізу $K = \{0,1, \dots, N\}$, пошукова таблиця (Lookup Table) з параметрами $D(k), T_{edge}(k), E_{comp}(k)$ для кожного $k \in K$, початкові вагові коефіцієнти α, β та критичний рівень заряду батареї B_{crit}. Вихідними даними є оптимальна точка розрізу k^* на поточний часовий інтервал t.

1: Ініціалізація: Запуск фонового таймера з інтервалом оновлення Δt
2: while мультимодальний асистент активний do
3: Отримати поточну оцінку пропускної здатності мережі R_t
4: Отримати поточний рівень заряду батареї B_t
5: if $B_t < B_{crit}$ then
6: Перерахувати ваги: збільшити β (пріоритет енергозбереження), зменшити α
7: end if
8: Ініціалізувати змінну мінімальної вартості: $C_{min} = \infty$
9: Ініціалізувати $k^* = 0$
10: for кожного $k \in K$ do
11: Оцінити час передачі: $T_{trans}(k) = \frac{D(k)}{R_t}$
12: Обчислити прогнозовану затримку: $\hat{T}_{total}(k) = T_{edge}(k) + T_{trans}(k) + T_{cloud}(k)$
13: Обчислити прогнозовану енергію: $\hat{E}_{total}(k) = E_{comp}(k) + E_{trans}(k, R_t)$
14: Обчислити значення цільової функції (Cost):
$C(k) = \alpha \cdot \text{Normalization}(\hat{T}_{total}) + \beta \cdot \text{Normalization}(\hat{E}_{total})$
15: if $C(k) < C_{min}$ then
16: $C_{min} = C(k)$
17: $k^* = k$
18: end if
19: end for
20: Застосувати k^* для обробки наступного пакету мультимодальних даних
21: Очікувати завершення інтервалу Δt
22: end while

Наведений алгоритм має обчислювальну складність $O(|K|)$, де $|K|$ — кількість можливих точок розрізу. Оскільки кількість макро-шарів (або блоків, таких як Residual blocks чи Encoder layers) у сучасних нейромережах є відносно невеликою, цикл for (рядки 10-19) виконується за мілісекунди і не створює обчислювального навантаження на edge-пристрій [16].

Динамічна зміна вагових коефіцієнтів (рядки 5-7) є критичною інновацією алгоритму. Вона гарантує, що пристрої з достатнім рівнем заряду будуть максимізувати швидкість реакції асистента (агресивно використовуючи мережу), тоді як розряджені пристрої автоматично перейдуть у режим глибокого локального висновку (Edge-heavy), жертвуючи часткою мілісекунд затримки задля збереження життєздатності вузла. Зчитування параметра R_t дозволяє системі миттєво реагувати на деградацію зв'язку (наприклад, перехід користувача з зони покриття Wi-Fi у зону нестабільного мобільного зв'язку) та зміщувати k^* ближче до вихідного шару, відправляючи в мережу лише максимально стиснені семантичні тензори.

Для емпіричної перевірки розробленої математичної моделі та алгоритму адаптивного розподілу було проведено імітаційне моделювання в середовищі, що імітує архітектуру AIoT. У якості базової моделі для обробки візуальної модальності було обрано модифіковану архітектуру згорткової нейронної мережі (наприклад, VGG-16 або ResNet-50), яка часто використовується в системах комп'ютерного зору на периферії [18]. Моделювання оцінювало обсяги переданих даних (проміжних тензорів) та загальну затримку системи (T_{total}) при різних точках розрізу (k) та різних станах бездротової мережі.

Фундаментальною проблемою, яку вирішує запропонована архітектура, є оптимізація

мережевого трафіку. У таблиці 1 наведено порівняння обсягу даних, які необхідно передати через мережу, залежно від обраного шару розрізу нейромережі.

Таблиця 1. Порівняння розміру проміжних тензорів на різних етапах обробки (на прикладі обробки відеокадру)

Точка розрізу (k)	Тип вихідних даних (Тензор)	Розмір (МБ)	Зміна	Характеристика етапу
$k = 0$ (Cloud-only)	Сирий відеокадр (RGB)	1.20	100%	Високе навантаження на канал, ризики приватності.
$k = 1$ (Ранні згортки)	Карта ознак (Conv1)	3.80	316%	Розширення даних. Неефективно для передачі.
$k = 3$ (Середні шари)	Карта ознак (Pool2)	0.65	54%	Початок семантичного стиснення.
$k = 6$ (Глибокі шари)	Семантичний тензор (Pool4)	0.15	12.5%	Оптимально для слабких мереж (високий ступінь компресії).
$k = N$ (Edge-only)	Вектор класифікації / Намір	0.001	$< 0.1\%$	Максимальна економія трафіку, але максимальні витрати енергії Edge.

Дані відображають характерний для глибоких нейромереж феномен «data amplification» на ранніх шарах, що підтверджується дослідженнями [19]. Розрізати мережу на шарі $k = 1$ часто гірше, ніж відправляти сире відео, через специфіку екстракції багатовимірних ознак.

Ключовим показником ефективності мультимодального асистента є час відгуку. Оскільки загальна затримка складається з часу обчислень та часу передачі ($T_{trans} = D(k)/R$), зміна пропускної здатності мережі (R) радикально впливає на вибір оптимального шару розрізу (k^*).

Результати яскраво демонструють три ключові сценарії роботи алгоритму.

Перший сценарій «Поганий інтернет» (Low Bandwidth, наприклад, 1-5 Mbps). Час передачі даних стає «вузьким місцем». Запропонований алгоритм автоматично зміщує точку розрізу в глибокі шари ($k \to N$). Пристрій виконує більшість обчислень локально, формуючи глибоко стиснений семантичний тензор (наприклад, 0.15 МБ замість 1.2 МБ), який швидко передається навіть нестабільним каналом.

Другий сценарій «Хороший інтернет» (High Bandwidth, наприклад, 50+ Mbps). Мережа здатна миттєво передавати великі обсяги даних. У цьому випадку алгоритм перемикає розріз на рівень $k = 0$ або $k = 2$, миттєво «скидаючи» «важкі» обчислення у високопродуктивну хмару, чим заощаджує заряд батареї (зменшує E_{comp}) і мінімізує загальний час відгуку.

Третій сценарій «Адаптивний компроміс» (Fluctuating Network). При мінливих умовах зв'язку система постійно балансує, обираючи середні шари ($k = 3..5$), уникаючи екстремумів.

Таким чином, розроблений адаптивний метод дозволяє системі постійно знаходитися в точці глобального мінімуму затримки, виграючи як у жорстко заданих Edge-only, так і у Cloud-only архітектур.

Для всебічного обґрунтування розробленого методу було проведено порівняльний аналіз запропонованої адаптивної моделі колаборативного висновку з трьома класичними статичними парадигмами: Cloud-only (всі обчислення в хмарі, $k = 0$), Edge-only (всі обчислення локально, $k = N$) та Static Split (жорстко зафіксована точка розрізу на середніх шарах, наприклад $k = 3$). Аналіз отриманих даних дозволяє зробити висновки щодо кількісної переваги адаптивного підходу за двома ключовими метриками: часом відгуку та енергоефективністю.

Аналіз результатів симуляції доводить, що в умовах стабільно високої пропускної здатності мережі (понад 50 Mbps) продуктивність адаптивного підходу асимптотично наближається до ідеального Cloud-only сценарію. Однак у реальних AIoT-мережах швидкість передачі даних R піддається значним флуктуаціям. Під час різкого падіння пропускної здатності (наприклад, перевантаження каналу або фізичні перешкоди) парадигма Cloud-only демонструє експоненційне зростання загальної затримки $\hat{T}_{total}$ через утворення мережевого «вузького місця». Своєю чергою, підхід Edge-only генерує стабільно високу затримку через обмежені можливості локального процесора (NPU), що не прийнятно для мультимодальних систем реального часу.

Алгоритм адаптивного розподілу вирішує цю проблему шляхом миттєвого зміщення точки розрізу k^* вглиб нейромережі при деградації зв'язку, тим самим замінюючи «дорогу» (за часом) передачу даних на «дешеві» локальні обчислення [20]. Зіставлення інтегральних показників затримки за весь цикл моделювання зі змінним станом мережі показало, що адаптивний підхід виграє у статичного (Static Split) в середньому на 28–35% за часом відгуку, а в моменти пікових просідань мережі ця перевага сягає 45% у порівнянні з хмарним висновком.

Енергоспоживання $\widehat{E}_{total}$ є не менш критичним фактором. Дослідження підтверджують, що витрати енергії на радіотрансляцію (E_{trans}) 1 МБ сирих мультимедійних даних через 4G/5G мережу можуть значно перевищувати енерговитрати на виконання мільйонів операцій множення-додавання (MACs) локальним процесором (E_{comp}) [21].

Завдяки введенню параметра β (вагового коефіцієнта чутливості до заряду батареї) у цільову функцію, система отримує здатність до самозбереження. Коли рівень заряду падає нижче критичної позначки, алгоритм навмисно збільшує обсяг обчислень на edge-вузлі, щоб максимально стиснути проміжний тензор $D(k)$ і мінімізувати час роботи енергоємного радіомодуля. Загальний аналіз енергетичного профілю демонструє, що запропонована система є на 18–25% більш енергоефективною, ніж статична модель розподілу, і здатна подовжити час автономної роботи мультимодального асистента на третину порівняно з Cloud-only підходом.

Додатковою перевагою розробленого методу є врахування гетерогенності даних. Оскільки оптимальна точка розрізу для обробки аудіо (k_A^*) та відео (k_V^*) часто не збігається через різну розмірність вихідних тензорів [22], незалежне адаптивне перемикання цих потоків дозволяє уникнути компромісів, притаманних монолітним архітектурам.

Підсумовуючи, математичне моделювання підтверджує, що динамічний розподіл обчислювального навантаження є науково обґрунтованим механізмом подолання обмежень слабких апаратних вузлів та нестабільних мереж, забезпечуючи високий рівень автономності та швидкодії в edge-cloud інфраструктурі.

Висновки та перспективи подальшого дослідження. У дослідженні вирішено актуальне науково-практичне завдання щодо оптимізації обміну даними та обчислювального навантаження в мультимодальних асистентах штучного інтелекту. Класичні парадигми (Edge-only та Cloud-only) вичерпали свій потенціал в умовах гетерогенних AIoT-мереж через жорсткі обмеження акумуляторних батарей, обчислювальних потужностей та пропускної здатності каналів зв'язку.

Для подолання цих обмежень розроблено математичну модель та метод динамічного (адаптивного) розподілу обчислень на базі концепції Split Computing. Запропонований алгоритм здійснює безперервний моніторинг стану мережі та рівня енергозабезпечення пристрою, динамічно зміщуючи точку розрізу глибокої нейромережі для досягнення глобального мінімуму затримки та енерговитрат.

Результати імітаційного моделювання підтвердили високу ефективність розробленого методу. Зокрема, моделювання показало, що запропонований адаптивний підхід дозволяє зменшити затримку мультимодального висновку в умовах нестабільної бездротової мережі в середньому на 28–35% (з піковим виграшем до 45% при критичній деградації каналу) у порівнянні з класичним Cloud-only підходом. Крім того, завдяки інтеграції параметра енергозбереження у цільову функцію, система демонструє підвищення загальної енергоефективності на 18–25% порівняно з моделями статичного розподілу (Static Split), значно подовжуючи час автономної роботи периферійного вузла.

Подальший розвиток даної проблематики планується здійснювати за двома стратегічними векторами:

Інтеграція семантичного стиснення: Поточна модель покладається на просторове розрізання графа нейромережі (передачу проміжних карт ознак). У майбутніх роботах планується інтегрувати сюди механізми семантичного стиснення даних (Semantic Communication/Task-Oriented Communication) на базі теорії інформаційного вузького місця (Information Bottleneck). Це дозволить на рівні edge-вузла відкидати інформаційний шум з аудіо- та відеопотоків, формуючи надкомпактні тензори, що містять виключно критичну "семантику" для прийняття рішень у хмарі.

Масштабування архітектури: Розроблену двомасштабну математичну модель (Edge–Cloud) буде розширено та адаптовано для гетерогенної трирівневої архітектури Terminal–Edge–Cloud. У такій системі кінцеві пристрої користувача (Terminal — наприклад, AR-гарнітури, смарт-годинники чи мікроконтролери розумного дому) візьмуть на себе функції первинної агрегації та легкого передобчислення, делегуючи складніші завдання на локальні edge-сервери та глобальну хмару.

Список бібліографічного опису

1. Mohammadi M., Al-Fuqaha A., Sorour S., Guizani M. Deep learning for IoT big data and streaming analytics: A review. IEEE Communications Surveys & Tutorials. 2018. Vol. 20, No. 4. P. 2923–2960. DOI: https://doi.org/10.1109/COMST.2018.2844341
2. Zhou Z., Chen X., Li E., Zeng L., Luo K., Zhang J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE. 2019. Vol. 107, No. 8. P. 1738–1762. DOI: https://doi.org/10.1109/JPROC.2019.2918951
3. Xu X., Ding H., Hu C., Wang Y., Chen S. Edge computing for multimodal data processing: A survey. IEEE Internet of Things Journal. 2022. Vol. 9, No. 12. P. 9477–9493. DOI: https://doi.org/10.1109/JIOT.2022.3153549
4. Merenda M., Porcaro C., Iero D. Edge machine learning for AI-enabled IoT devices: A review. Sensors. 2020. Vol. 20, No. 9. Article 2533. DOI: https://doi.org/10.3390/s20092533
5. Wang Y., Chen Y., Lu Y. Resource-constrained multimodal transformers for edge AI: Challenges and solutions. IEEE Communications Surveys & Tutorials. 2023. Vol. 25, No. 3. P. 1500–1525. URL: https://ieeexplore.ieee.org/document/10144670
6. Li S., Zhao S., Zhao P., Wang X., Liu Y. Collaborative edge-cloud computing for AIoT. IEEE Internet of Things Journal. 2021. Vol. 8, No. 16. P. 12698–12711. DOI: https://doi.org/10.1109/JIOT.2021.3064391
7. Zhang J., Chen B., Zhao Y., Cheng X., Hu F. Data security and privacy-preserving in edge computing paradigm: Survey and open issues. IEEE Access. 2018. Vol. 10. P. 60000–60020. DOI: https://doi.org/10.1109/ACCESS.2018.2820162
8. Matsubara Y., Levorato M., Restuccia F. Split computing and early exiting for deep learning applications: Survey and research challenges. ACM Computing Surveys. 2022. Vol. 55, No. 5. P. 1–30. DOI: https://doi.org/10.1145/3527155
9. Kang Y., Hauswald J., Gao C., Rovinski A., Mudge T., Mars J., Tang L. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. ACM SIGARCH Computer Architecture News. 2017. Vol. 45, No. 1. P. 615–629. DOI: https://doi.org/10.1145/3037697.3037698
10. Teerapittayanon S., McDanel B., Kung H. T. Distributed deep neural networks over the cloud, the edge and end devices. 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). 2017. P. 328–339. DOI: https://doi.org/10.1109/ICDCS.2017.264
11. Gupta O., Raskar R. Distributed learning of deep neural network over multiple agents. Journal of Network and Computer Applications. 2018. Vol. 116. P. 1–8. DOI: https://doi.org/10.1016/j.jnca.2018.05.003
12. Shao J., Zhang J. Bottlenet++: An end-to-end approach for feature compression in device-cloud collaborative inference. IEEE Open Journal of the Communications Society. 2020. Vol. 1. P. 162–175. DOI: https://doi.org/10.1109/OJCOMS.2020.2976103
13. Hu C., Bao W., Wang D., Liu F. Dynamic adaptive DNN surgery for inference acceleration on the edge. IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. 2019. P. 1423–1431. DOI: https://doi.org/10.1109/INFOCOM.2019.8737395
14. Ren J., Yu G., He Y., Li G. Y. Collaborative cloud and edge computing for latency minimization. IEEE Transactions on Vehicular Technology. 2019. Vol. 68, No. 5. P. 5031–5044. DOI: https://doi.org/10.1109/TVT.2019.2904244
15. Zhang W., Wen Y., Guan K., Kilper D., Luo H., Wu D. O. Energy-optimal mobile cloud computing under computation deadline constraints. IEEE Transactions on Mobile Computing. 2020. Vol. 12, No. 12. P. 2427–2438. DOI: https://doi.org/10.1109/TMC.2019.2902636
16. Laskaridis S., Venieris S. I., Almeida M., Leontiadis I., Lane N. D. SPINN: Synergistic progressive inference of neural networks over device and cloud. Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. 2020. P. 1–15. DOI: https://doi.org/10.1145/3372224.3419194
17. Zeng L., Li E., Zhou Z., Chen X. Boomerang: On-demand cooperative deep neural network inference for edge intelligence on the Industrial Internet of Things. IEEE Network. 2019. Vol. 33, No. 5. P. 96–103. DOI: https://doi.org/10.1109/MNET.2019.1800271
18. Li H., Ota K., Dong M. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. IEEE Network. 2018. Vol. 32, No. 1. P. 96–101. DOI: https://doi.org/10.1109/MNET.2018.1700202
19. Xue M., Wu H., Peng G., Ren K. DDPNN: Distributed deep perception neural networks for cloud-edge collaborative inference. IEEE Internet of Things Journal. 2021. Vol. 8, No. 23. P. 17166–17178. DOI: https://doi.org/10.1109/JIOT.2021.3077508
20. Eshratifar A. E., Pedram M. Energy and performance efficient computation offloading for deep neural networks in a mobile cloud computing environment. Proceedings of the 2018 on Great Lakes Symposium on VLSI. 2018. P. 111–116. DOI: https://doi.org/10.1145/3194554.3194573
21. Ko J. H., Na T., Amir M. F., Mukhopadhyay S. Edge-host partitioning of deep neural networks with feature space encoding for resource-constrained Internet-of-Things platforms. 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). 2018. P. 1–6. DOI: https://doi.org/10.1109/AVSS.2018.8639144
22. Wang S., Tuor T., Salonidis T., Leung K. K., Makaya C., He T., Chan K. Adaptive federated learning in resource constrained edge computing systems. IEEE Journal on Selected Areas in Communications. 2019. Vol. 37, No. 6. P. 1205–1221. DOI: https://doi.org/10.1109/JSAC.2019.2904348

References

1. Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A review. IEEE Communications Surveys & Tutorials, 20(4), 2923–2960. https://doi.org/10.1109/COMST.2018.2844341
2. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE, 107(8), 1738–1762.

https://doi.org/10.1109/JPROC.2019.2918951
3. Xu, X., Ding, H., Hu, C., Wang, Y., & Chen, S. (2022). Edge computing for multimodal data processing: A survey. IEEE Internet of Things Journal, 9(12), 9477–9493. https://doi.org/10.1109/JIOT.2022.3153549
4. Merenda, M., Porcaro, C., & Iero, D. (2020). Edge machine learning for AI-enabled IoT devices: A review. Sensors, 20(9), Article 2533. https://doi.org/10.3390/s20092533
5. Wang, Y., Chen, Y., & Lu, Y. (2023). Resource-constrained multimodal transformers for edge AI: Challenges and solutions. IEEE Communications Surveys & Tutorials, 25(3), 1500–1525. https://ieeexplore.ieee.org/document/10144670
6. Li, S., Zhao, S., Zhao, P., Wang, X., & Liu, Y. (2021). Collaborative edge-cloud computing for AIoT. IEEE Internet of Things Journal, 8(16), 12698–12711. https://doi.org/10.1109/JIOT.2021.3064391
7. Zhang, J., Chen, B., Zhao, Y., Cheng, X., & Hu, F. (2018). Data security and privacy-preserving in edge computing paradigm: Survey and open issues. IEEE Access, 10, 60000–60020. https://doi.org/10.1109/ACCESS.2018.2820162
8. Matsubara, Y., Levorato, M., & Restuccia, F. (2022). Split computing and early exiting for deep learning applications: Survey and research challenges. ACM Computing Surveys, 55(5), 1–30. https://doi.org/10.1145/3527155
9. Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J., & Tang, L. (2017). Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. ACM SIGARCH Computer Architecture News, 45(1), 615–629. https://doi.org/10.1145/3037697.3037698
10. Teerapittayanon, S., McDanel, B., & Kung, H. T. (2017). Distributed deep neural networks over the cloud, the edge and end devices. In 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS) (pp. 328–339). IEEE. https://doi.org/10.1109/ICDCS.2017.264
11. Gupta, O., & Raskar, R. (2018). Distributed learning of deep neural network over multiple agents. Journal of Network and Computer Applications, 116, 1–8. https://doi.org/10.1016/j.jnca.2018.05.003
12. Shao, J., & Zhang, J. (2020). Bottlenet++: An end-to-end approach for feature compression in device-cloud collaborative inference. IEEE Open Journal of the Communications Society, 1, 162–175. https://doi.org/10.1109/OJCOMS.2020.2976103
13. Hu, C., Bao, W., Wang, D., & Liu, F. (2019). Dynamic adaptive DNN surgery for inference acceleration on the edge. In IEEE INFOCOM 2019 - IEEE Conference on Computer Communications (pp. 1423–1431). IEEE. https://doi.org/10.1109/INFOCOM.2019.8737395
14. Ren, J., Yu, G., He, Y., & Li, G. Y. (2019). Collaborative cloud and edge computing for latency minimization. IEEE Transactions on Vehicular Technology, 68(5), 5031–5044. https://doi.org/10.1109/TVT.2019.2904244
15. Zhang, W., Wen, Y., Guan, K., Kilper, D., Luo, H., & Wu, D. O. (2020). Energy-optimal mobile cloud computing under computation deadline constraints. IEEE Transactions on Mobile Computing, 12(12), 2427–2438. https://doi.org/10.1109/TMC.2019.2902636
16. Laskaridis, S., Venieris, S. I., Almeida, M., Leontiadis, I., & Lane, N. D. (2020). SPINN: Synergistic progressive inference of neural networks over device and cloud. In Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (pp. 1–15). Association for Computing Machinery. https://doi.org/10.1145/3372224.3419194
17. Zeng, L., Li, E., Zhou, Z., & Chen, X. (2019). Boomerang: On-demand cooperative deep neural network inference for edge intelligence on the Industrial Internet of Things. IEEE Network, 33(5), 96–103. https://doi.org/10.1109/MNET.2019.1800271
18. Li, H., Ota, K., & Dong, M. (2018). Learning IoT in edge: Deep learning for the Internet of Things with edge computing. IEEE Network, 32(1), 96–101. https://doi.org/10.1109/MNET.2018.1700202
19. Xue, M., Wu, H., Peng, G., & Ren, K. (2021). DDPNN: Distributed deep perception neural networks for cloud-edge collaborative inference. IEEE Internet of Things Journal, 8(23), 17166–17178. https://doi.org/10.1109/JIOT.2021.3077508
20. Eshratifar, A. E., & Pedram, M. (2018). Energy and performance efficient computation offloading for deep neural networks in a mobile cloud computing environment. In Proceedings of the 2018 on Great Lakes Symposium on VLSI (pp. 111–116). Association for Computing Machinery. https://doi.org/10.1145/3194554.3194573
21. Ko, J. H., Na, T., Amir, M. F., & Mukhopadhyay, S. (2018). Edge-host partitioning of deep neural networks with feature space encoding for resource-constrained Internet-of-Things platforms. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1–6). IEEE. https://doi.org/10.1109/AVSS.2018.8639144
22. Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T., & Chan, K. (2019). Adaptive federated learning in resource constrained edge computing systems. IEEE Journal on Selected Areas in Communications, 37(6), 1205–1221. https://doi.org/10.1109/JSAC.2019.2904348

Історія статті:
Отримано: 06.05.2026 Доопрацьовано: 20.05.2026 Прийнято до друку: 23.05.2026 Опубліковано: 29.05.2026