УДК 004.415.2
DOI https://doi.org/10.32782/2663-5941/2025.3.2/37

Олещенко Л.М.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ДОСЛІДЖЕННЯ АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПАРАЛЕЛЬНОЇ ОБРОБКИ ВЕЛИКИХ ДАНИХ І ЗАДАЧ МАШИННОГО НАВЧАННЯ У ХМАРНОМУ СЕРЕДОВИЩІ

У сучасних умовах стрімкого зростання обсягів інформації та складності обчислювальних задач виникає потреба у високопродуктивних підходах до обробки великих даних та реалізації алгоритмів машинного навчання. Одним з ключових напрямків у цій сфері є застосування паралельних та розподілених обчислень із використанням спеціалізованих апаратних і програмних засобів у хмарному та периферійному середовищах. Стаття присвячена аналізу розвитку апаратних архітектур (GPU, TPU, FPGA), технологій обробки великих даних у хмарному середовищі, а також дослідженню сучасних інструментів машинного навчання (TensorFlow, PyTorch, Apache Spark), які забезпечують ефективну обробку та аналіз даних у хмарній інфраструктурі. У статті розкрито переваги та недоліки різних архітектур з точки зору продуктивності, масштабованості та гнучкості. З'ясовано, що GPU забезпечують найвищу продуктивність, однак поступаються FPGA за гнучкістю, TPU демонструють високу ефективність для вузькоспеціалізованих задач, але обмежені в адаптивності до гетерогенних середовищ. Визначено, що серед програмних фреймворків TensorFlow вирізняється високим рівнем продуктивності та масштабованості, PyTorch має перевагу у зручності розробки й адаптації моделей, Apache Spark, незважаючи на загальне призначення, демонструє високу ефективність у розподіленій обробці даних. Висвітлено низку відкритих проблем, зокрема, складність досягнення балансу між точністю, швидкістю та конфіденційністю у розподілених ML-системах; нестача уніфікованих інструментів для інтеграції різнорідних обчислювальних платформ; обмежені можливості edge-пристроїв щодо підтримки складних ML-моделей; а також відсутність систематичних порівнянь методів у єдиних експериментальних умовах. Метою даного дослідження є узагальнення існуючих підходів до організації паралельної обробки великих даних у хмарному середовищі, виявлення ключових тенденцій, порівняння їх ефективності та визначення актуальних проблем, які потребують подальших досліджень. Проведено аналіз та сформульовано стратегічні напрями для оптимізації апаратно-програмної архітектури систем обробки великих даних, враховуючи вимоги до адаптивності, ефективності та масштабованості у динамічному середовищі.

Ключові слова: *паралельні обчислення, хмарні технології, великі дані, машинне навчання, GPU, TPU, FPGA, TensorFlow, PyTorch, Apache Spark, розподілені системи, продуктивність, масштабованість, гнучкість.*

Постановка проблеми. Зі зростанням обсягів даних та складності моделей машинного навчання (ML) виникає потреба в ефективних обчислювальних ресурсах. Хмарні обчислення, поєднані з паралельними обчислювальними методами, надають можливість масштабувати обробку великих даних та навчання ML-моделей. Виникає потреба в систематизації сучасних підходів до паралельних обчислень у хмарному середовищі, що зумовлює необхідність класифікації методів, апаратних і програмних засобів та порівняння їх ефективності для вибору оптимальних рішень у реальних умовах. Основні проблеми при обробці великих даних та навчанні ML-моделей у хмарі включають масштабованість – необхідність ефективного розподілу обчислень на великі обсяги даних; продуктивність – забезпечення високої швидкості обробки та навчання моделей; вартість – оптимізація витрат на обчислювальні ресурси та сумісність – інтеграція з існуючими інструментами та фреймворками.

Аналіз останніх досліджень і публікацій. У статті [1] розглядається проблема обробки великих даних, які розподілені між різними обчислювальними вузлами через обмеження інфраструктури або вимоги до конфіденційності. Традиційні централізовані методи обробки даних стають непридатними в таких умовах, що зумовлює необхідність розробки розподілених мето-

дів навчання, які забезпечують високу точність інференції та прогнозування, дотримуючись при цьому політик конфіденційності. Автори пропонують класифікацію існуючих методів розподіленого навчання за чотирма критеріями: статистичною точністю (здатністю методу досягати результатів, порівнянних з централізованими підходами), обчислювальною ефективністю (швидкістю та ресурсоємністю обчислень), гетерогенністю (здатністю працювати з неоднорідними даними та системами) і конфіденційністю (ступенем захисту приватності даних). Такий підхід дозволяє комплексно оцінювати нові методи з різних аспектів. У статті надається огляд теоретичних результатів, що стосуються статистичної еквівалентності та обчислювальної ефективності різних методів розподіленого навчання. Особливу увагу приділено умовам, за яких ці методи можуть досягати точності, порівнянної з централізованими підходами, а також аналізується їхня обчислювальна складність. Автори також розглядають наявні програмні реалізації методів розподіленого навчання та еталонні набори даних, що використовуються для їх тестування та порівняння. Це включає популярні фреймворки та бібліотеки, а також набори даних з різних галузей застосування. У статті окреслено кілька ключових напрямків для подальших досліджень, серед яких розробка методів, стійких до гетерогенності даних та систем, покращення обчислювальної ефективності при збереженні високої точності, а також інтеграція механізмів забезпечення конфіденційності, таких як диференційна приватність.

У роботі [2] автори розглядають актуальність розробки апаратних прискорювачів для глибокого навчання на периферійних пристроях. Зі зростанням застосування машинного навчання в різних сферах, таких як комп›ютерний зір, розпізнавання мови та Інтернет речей (IoT), виникає потреба в ефективних апаратних рішеннях для обробки даних безпосередньо на пристроях з обмеженими ресурсами. Автори аналізують основні типи апаратних прискорювачів, включаючи ASIC, FPGA та GPU. Кожна з цих платформ має свої переваги та обмеження щодо продуктивності, енергоспоживання та гнучкості. Наприклад, ASIC забезпечують високу продуктивність та енергоефективність, але менш гнучкі в порівнянні з FPGA, які дозволяють перепрограмування для різних задач. У статті розглядаються різні архітектурні рішення для прискорення виконання нейронних мереж, включаючи оптимізацію потоків даних, топології мереж та використання новітніх технологій. Особлива увага приділяється методам стиснення моделей, таким як квантування та обрізання, які зменшують обсяг обчислень та енергоспоживання без значної втрати точності. Автори наголошують про відсутність стандартів для порівняння ефективності різних рішень та необхідність балансування між продуктивністю, енергоефективністю та точністю моделей. Також обговорюються перспективні напрямки досліджень, включаючи розвиток реконфігурованих архітектур та інтеграцію нових матеріалів для створення більш ефективних прискорювачів.

У статті [3] розглянуто рішення, пов›язані з масштабуванням алгоритмів машинного навчання для обробки великих обсягів даних у паралельних обчислювальних середовищах. Розглянуто необхідність ефективного використання обчислювальних ресурсів для обробки великих наборів даних, що вимагає розподілу обчислень між кількома процесорами або вузлами.

Одним із ключових методів, які розглядає автор, є використання техніки хешування для зменшення розмірності вхідних даних. Цей підхід дозволяє ефективно обробляти великі обсяги даних, зменшуючи кількість необхідних обчислень та пам›яті. Автор також обговорює використання онлайн-навчання та алгоритмів, які можуть адаптуватися до нових даних у режимі реального часу, що є важливим для обробки потокових даних у великих системах. Для забезпечення ефективної паралельної обробки даних автор пропонує використання операції AllReduce, яка дозволяє об’єднувати результати обчислень з різних вузлів у єдине значення. Цей підхід зменшує обсяг переданих даних між вузлами та покращує загальну продуктивність системи. Автор також підкреслює важливість балансування навантаження між вузлами та оптимізації комунікацій для досягнення високої ефективності паралельних обчислень. У статті також розглядаються практичні аспекти реалізації паралельного машинного навчання, включаючи використання фреймворків, таких як Hadoop, для розподіленої обробки даних. Автор ділиться досвідом впровадження цих методів у реальних системах та обговорює переваги та обмеження різних підходів до паралельного машинного навчання.

У статті [4] розглядається проблема забезпечення пояснюваності (explainability) аналітики великих даних у хмарному середовищі шляхом

використання самоструктуровного штучного інтелекту (Self-Structuring AI). Автори наголошують, що зростання обсягів і складності даних створює нові труднощі щодо зрозумілості прийнятих моделей та рішень, що особливо актуально для критичних галузей, таких як охорона здоров'я, фінанси чи безпека. У таких випадках важливо не лише отримати точний прогноз, а й пояснити, чому він був зроблений саме так. Запропонована авторами архітектура базується на поєднанні хмарних обчислювальних ресурсів з підходом до самоструктуризації моделей, який дозволяє штучному інтелекту адаптивно формувати власну структуру в залежності від характеру даних. Це забезпечує як високу продуктивність, так і покращену прозорість моделей завдяки вбудованим механізмам інтерпретації. Основними компонентами архітектури є модулі попередньої обробки даних, самоструктурованого моделювання, генерації пояснень, а також розподіленої обробки у хмарі для масштабування. Автори детально описують механізм самоструктуризації, який базується на принципах еволюційного навчання та динамічного формування структури моделі під час навчання. Це дозволяє адаптуватися до нових типів даних без ручного втручання. Однією з ключових переваг підходу є інтеграція модулів пояснюваності (наприклад, SHAP або LIME) без значної втрати продуктивності, що дозволяє зберігати баланс між точністю, прозорістю та обчислювальною ефективністю. У роботі також наведено експериментальні результати на основі кількох публічних наборів даних, зокрема, з галузей медицини, фінансів та кібербезпеки. Показано, що запропонована архітектура перевершує традиційні підходи як за швидкістю обробки, так і за метриками пояснюваності. Наприклад, самоструктурні моделі продемонстрували на 15–25% вищий рівень інтерпретаційності (згідно з метриками user trust та feature importance clarity) порівняно з класичними нейромережевими структурами, без суттєвого зниження точності. У роботі підкреслено, що поєднання хмарних обчислень, розподіленої обробки та самоструктуровної штучної інтелектуальної аналітики є перспективним напрямом для розв'язання проблем масштабованої, ефективної та пояснюваної обробки великих даних. Автори також вказують на потенціал розширення запропонованого підходу шляхом інтеграції з існуючими фреймворками машинного навчання у хмарних середовищах, такими як Amazon SageMaker, Google Vertex AI або Microsoft Azure ML.

У роботі [5] розглядається проблема ефективної обробки електроенцефалографічних (EEG) сигналів із застосуванням згорткових нейронних мереж (CNN) та апаратного прискорення. Автори наголошують, що EEG-дані мають високий об'єм, складну структуру та потребують обробки в реальному часі, що створює значні обчислювальні навантаження. Це особливо актуально для медичних додатків, таких як виявлення епілепсії, контролю нейроінтерфейсів, або діагностики психоневрологічних станів. У зв'язку з цим зростає потреба у використанні апаратно-прискорених рішень, що забезпечують високу продуктивність, енергоефективність та низьку затримку. У роботі представлено огляд сучасних апаратних технологій прискорення обробки сигналів EEG, зокрема FPGA (Field-Programmable Gate Arrays), ASIC (Application-Specific Integrated Circuits), GPU (Graphics Processing Units) та edge-архітектур. Наведено класифікацію архітектур за їх здатністю виконувати згорткові обчислення, ступенем паралелізації, споживанням енергії, затримкою та продуктивністю. Особлива увага приділена системам на кристалі (SoC), які поєднують високий рівень інтеграції з можливістю обробки даних поблизу джерела (near-sensor computing), що критично для мобільних і вбудованих додатків. Окремо розглядаються особливості застосування CNN для аналізу EEG, включаючи специфіку попередньої обробки сигналів, вибору архітектури мережі та налаштування гіперпараметрів. Автори підкреслюють, що ефективність CNN у цій сфері залежить від здатності адаптуватися до просторово-часових шаблонів у даних, що вимагає гнучких та масштабованих архітектур. Стаття містить порівняльний аналіз ефективності різних апаратних реалізацій. Наприклад, FPGA-рішення показали найменше енергоспоживання (до 50% менше порівняно з GPU) при збереженні високої продуктивності, тоді як GPU залишаються кращими у задачах з високою паралельністю, забезпечуючи до 30% вищу швидкість обробки. ASIC-платформи демонструють найвищу ефективність при виконанні фіксованих алгоритмів, однак обмежені в гнучкості. Також наводяться експериментальні результати для систем, які обробляють EEG-сигнали в режимі реального часу – затримки в таких системах при використанні FPGA залишались нижчими за 10 мс, що критично для застосувань у медичній діагностиці. У підсумку автори роблять висновок, що апаратне прискорення є ключовим фактором для

масштабованої, швидкої та енергоефективної обробки EEG-сигналів з використанням CNN. Вони підкреслюють важливість подальших досліджень у напрямку оптимізації обчислювальних архітектур для глибокого навчання у сфері біомедичної інженерії, включаючи гібридні системи з підтримкою інтелектуального керування обчислювальними ресурсами в реальному часі.

У статті [6] автори досліджують, як методи паралельних обчислень можуть бути ефективно застосовані для видобування знань з великих обсягів даних (big data mining). Зважаючи на експоненційне зростання даних у різних доменах (соціальні мережі, сенсорні системи, біоінформатика, фінансові сервіси тощо), традиційні алгоритми машинного навчання та аналізу даних втрачають ефективність через обмеження обчислювальних ресурсів. У зв'язку з цим автори зосереджуються на дослідженні архітектур та алгоритмів паралельних обчислень, які дозволяють масштабувати процеси обробки та аналізу даних. У статті проведено систематичний огляд існуючих підходів до паралельного видобування даних з використанням трьох основних обчислювальних моделей: Multicore-based parallel computing, Cluster-based parallel computing та Cloud-based computing. Автори аналізують типові архітектури кожного з підходів, їх переваги та недоліки, а також ефективність застосування в різних сценаріях обробки даних. Для кожного класу паралельних архітектур описуються конкретні реалізації. Наприклад, для кластерних систем розглядаються системи, побудовані на основі Hadoop/MapReduce та Spark, які демонструють високу масштабованість та ефективність при обробці великих розподілених наборів даних. У випадку з багатоядерними системами (multicore), перевага полягає у зниженні затримок і високій швидкодії, однак спостерігаються обмеження щодо обсягу даних, які можуть оброблятись локально. Хмарні рішення, зокрема, з використанням інфраструктури Amazon EC2 або Google Cloud, пропонують гнучке масштабування обчислень і спрощене керування, але можуть мати вищі затрати. Автори також здійснюють огляд алгоритмів паралельної кластеризації, класифікації та асоціативного правила утворення, таких як паралельний k-means, паралельний SVM та алгоритми FP-growth на Hadoop. Наводяться дані про приріст продуктивності при паралельному виконанні: наприклад, для паралельного k-means у хмарному середовищі з використанням Spark спостерігається прискорення до 70% у порівнянні з послідовною реалізацією, а для класифікації на базі Random Forest прискорення становить до 60% при збереженні аналогічного рівня точності. Автори відзначають, що паралельні обчислення є ключем до успішного масштабованого аналізу великих даних. Проте проблеми залишаються в галузі балансування навантаження між вузлами, оптимізації витрат на обчислювальні ресурси та забезпечення конфіденційності. Автори наголошують на перспективності подальших досліджень в автоматичному масштабуванні ресурсів, гібридних моделях обчислень (локальні та хмарні) та інтеграції паралельних підходів з новими парадигмами, такими як глибоке навчання та потокова обробка даних у реальному часі.

У дослідженні [7] автори розглядають впровадження високопродуктивної хмарної платформи для виконання інференції моделей машинного навчання з апаратним прискоренням за допомогою графічних процесорів GPU у контексті обробки великих наукових даних. Через масштаб і складність даних, які генеруються у фізичних дослідженнях, традиційні підходи обробки даних не забезпечують необхідної швидкодії й масштабованості, тому автори пропонують сервісну архітектуру з GPU-прискореною інференцією як рішення. Основна проблема, яку піднімає дослідження, – це складність розгортання та масштабування моделей глибокого навчання на традиційних CPU-серверах у великих наукових експериментах, де існує потреба у швидкій обробці даних для аналізу подій. Автори пропонують підхід, заснований на парадигмі Inference-as-a-Service (IaaS), у якому GPU-ресурси централізовано керуються через хмарну інфраструктуру з контейнеризованими сервісами. Це дозволяє багаторазово використовувати потужні GPU для численних запитів одночасно, оптимізуючи продуктивність і витрати (рис. 1).

Клієнтські CPU надсилають запити через мережу до балансувальника навантаження, який перенаправляє їх до одного з серверів інференції (CPU або GPU), розгорнутих у контейнерах (Docker, Kubernetes, Podman). Сервери використовують попередньо збережені моделі з репозиторію моделей AI.

У дослідженні представлена технічна реалізація системи: використовуються Docker-контейнери, Kubernetes для оркестрації, а також NVIDIA Triton Inference Server для ефективного керування інференційними запитами на GPU. Підтримується масштабована й модульна

обробка вхідних даних, що надходять із розподілених систем збору даних (рис. 2).

Значну увагу приділено інтеграції з існуючими науковими робочими процесами без потреби кардинальної модифікації моделей. Важливим елементом аналізу є оцінка продуктивності запропонованої системи.

В експериментах автори демонструють, що GPU-інференція забезпечує прискорення в 16–24 рази в порівнянні з CPU для моделей CNN, які використовуються у фізичних задачах класифікації подій. У середньому затримка обробки запиту (latency) зменшується до <20 мілісекунд, що дозволяє використовувати систему в режимі майже реального часу.

Також відзначено масштабованість рішення: система підтримує горизонтальне масштабування, дозволяючи підключати додаткові GPU-інстанси в хмарному середовищі залежно від навантаження. Це особливо важливо для експериментів, у яких обробка може бути критичною під час пікових періодів збору даних. Автори наголошують, що поєднання хмарної інфраструктури, контейнеризації та GPU-прискорення створює ефективне, гнучке і продуктивне середовище для машинного навчання у великих наукових проєктах.

У статті [8] автори здійснюють систематичний аналіз трьох провідних хмарних платформ – Amazon Web Services (AWS), Microsoft Azure та Google Cloud Platform (GCP) – з точки зору їх можливостей для реалізації проєктів машинного навчання (ML) і обробки великих даних. Проблематика, що розглядається в дослідженні, стосується вибору оптимальної хмарної інфраструктури, яка забезпечить баланс між обчислювальною продуктивністю, масштабованістю, простотою інтеграції та вартістю виконання

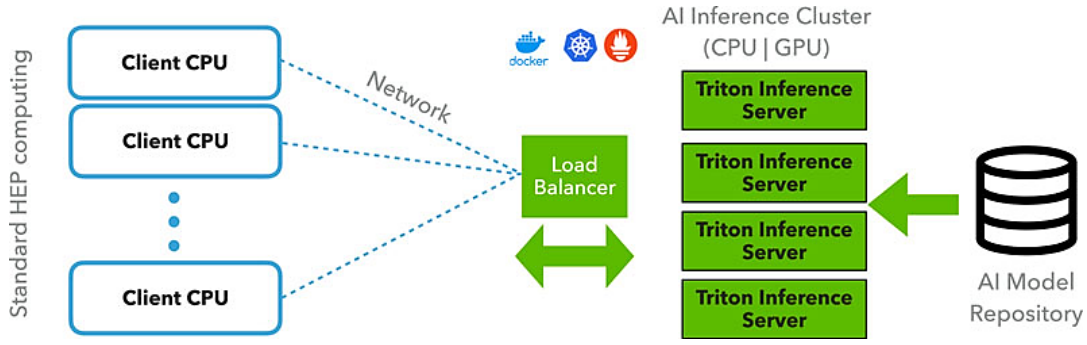

Рис. 1. Архітектура віддаленої інференції за допомогою кластерів Triton Inference Server у середовищі з розподіленими клієнтськими обчисленнями [7]

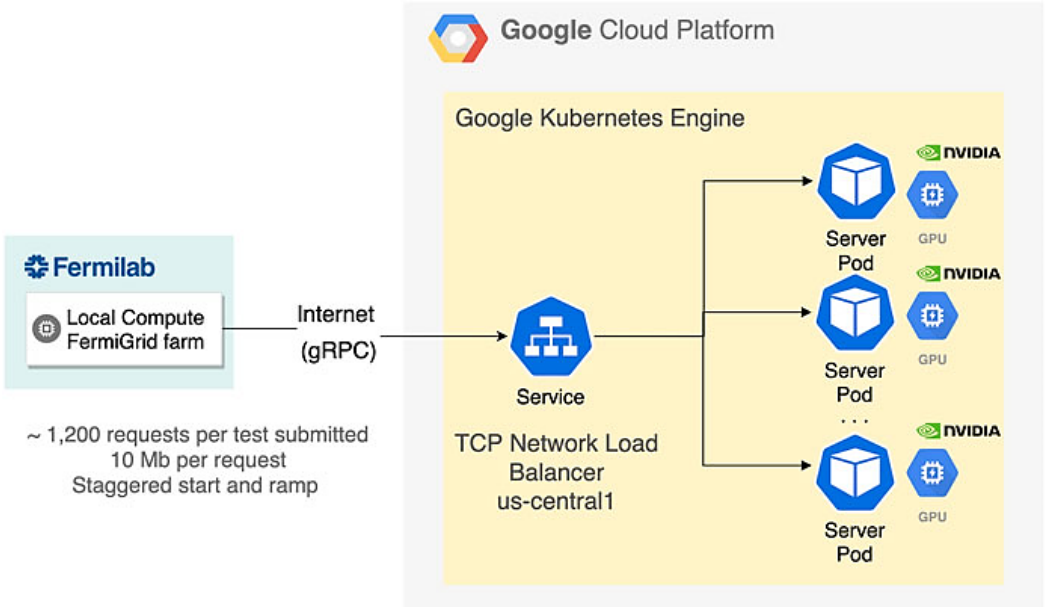

Рис. 2. Схема інференції моделей на хмарній платформі Google Cloud Platform з використанням Google Kubernetes Engine [7]

задач штучного інтелекту в умовах зростаючих обсягів даних і складності моделей. Автори порівнюють платформи за рядом ключових критеріїв: типи доступних обчислювальних ресурсів (CPU, GPU, TPU), підтримка фреймворків глибокого навчання (TensorFlow, PyTorch, MXNet), сервіси для автоматизованого машинного навчання (AutoML), інструменти для розгортання моделей (MLOps), зручність інтеграції з DevOps, а також політики безпеки, конфіденційності та ціноутворення.

Зібрані метрики базуються на практичному тестуванні моделей класифікації, регресії та обробки природної мови в межах кожної платформи, що дозволяє дати об'єктивну оцінку переваг і недоліків кожної з них. Згідно з аналізом, AWS показує найкращі результати у продуктивності GPU-обчислень та гнучкості конфігурації інфраструктури, особливо при використанні Amazon SageMaker. Azure відзначається сильною інтеграцією з корпоративними інструментами Microsoft, високим рівнем безпеки та аналітичними можливостями через Azure Machine Learning Studio.

Google Cloud демонструє лідерство в інноваціях завдяки нативній підтримці TensorFlow та доступу до TPU, що робить його привабливим для проєктів із глибоким навчанням і масштабною інференцією (табл. 1).

Таблиця 1

Порівняльні показники часу тренування моделей (в секундах), продуктивності інференції (кількість запитів на секунду) та вартості (у $/год) [8]

Платформа	Час тренування (GPU), с	Інференція (запитів/с)	Середня вартість ($/год)
AWS	85	1200	2.40
Azure	97	1100	2.60
Google Cloud	78	1250	2.30

Згідно з отриманими результатами, Google Cloud забезпечує найкращу продуктивність при найнижчій вартості, тоді як AWS надає найбільше можливостей для тонкого налаштування і масштабування, а Azure використовується для організацій, інтегрованих у Microsoft-екосистему. Автори підкреслюють, що вибір хмарної платформи для задач машинного навчання має базуватися не лише на продуктивності, але й на специфічних вимогах проєкту: типах моделей, потребах у безпеці, бюджеті та можливостях для автоматизації робочих процесів. Водночас вони відзначають, що розвиток хмарних ML-сервісів і надалі буде тісно пов'язаний із підтримкою паралельних обчислень, GPU/TPU-прискорення та безсерверних архітектур, що відкриває нові горизонти для масштабованої обробки великих даних.

У статті [9] автори розглядають актуальні проблеми інференції (застосування вже навчених моделей) глибокого навчання в умовах зростання складності моделей і потреби в реальному часі обробки великих обсягів даних. Особливу увагу приділено апаратним прискорювачам високої продуктивності, таким як GPU, TPU, FPGA та спеціалізовані ASIC, які дедалі частіше використовуються у хмарних середовищах та на периферії (edge computing).

Автори наголошують, що інференція великих нейронних мереж у промислових і наукових застосуваннях стикається з проблемами, пов'язаними з затримками, обмеженнями енергоспоживання та вимогами до обчислювальної потужності. У зв'язку з цим вони аналізують архітектурні особливості сучасних апаратних рішень, що дозволяють оптимізувати інференцію, зокрема: паралелізм обчислень, кешування, оптимізацію пам'яті, підтримку тензорних операцій та спеціалізованих інструкцій.

У статті представлено порівняльний аналіз апаратних прискорювачів у контексті типових задач глибокого навчання (класифікація зображень, розпізнавання мовлення, аналіз відео тощо), який відображено у табл. 2.

Таблиця 2

Експериментальні результати дослідження продуктивності (кількості інференцій на секунду), затримки (в мс), споживання енергії (Вт) та ефективності (інференції/Вт) на прикладі ResNet50 [9]

Прискорювач	Продуктивність (inf/s)	Затримка (мс)	Споживання енергії (Вт)	Ефективність (inf/Вт)
NVIDIA V100	18,000	1.5	250	72
Google TPUv3	22,000	1.2	280	78.6
Intel FPGA	6,500	3.8	90	72.2
ASIC (Edge)	4,800	2.0	40	120

Згідно з наведеним аналізом, TPUv3 демонструє найвищу продуктивність, тоді як спеціалізовані ASIC для edge-застосунків мають найкращу енергоефективність, що робить їх привабливими для застосування в автономних пристроях. FPGA забезпечують гнучкість та адаптивність до нестандартних моделей, хоча поступаються в продуктивності та затримці. У статті також розглянуто оптимізаційні підходи, що дозволяють адаптувати нейронні мережі для ефективної інференції на різних типах прискорювачів, зокрема, квантизація, яка призначена для апаратної сумісності, обрізання (pruning), компіляція графів виконання (XLA, TensorRT) та повторне використання попередніх обчислень (caching inference results). Автори підкреслюють значення вибору апаратного прискорювача, що найкраще відповідає конкретним завданням і обмеженням, а також важливість міжплатформенних фреймворків для забезпечення сумісності, продуктивності та масштабованості. Вони зазначають, що майбутнє інференції глибокого навчання значною мірою пов'язане з розвитком спеціалізованих систем на чипі (SoC), хмарних сервісів із підтримкою ML-as-a-Service та зростанням використання edge AI.

У статті [10] автори проаналізували проблеми, що постають при реалізації розподіленого машинного навчання (ML) у середовищі edge computing – тобто в обчисленнях, наближених до джерела даних. Такий підхід є ключовим для зниження затримок, економії пропускної здатності мережі та забезпечення конфіденційності даних, особливо в додатках Інтернету речей (IoT), розумних міст, автономного транспорту та охорони здоров'я. Автори виділили велику різноманітність апаратного забезпечення, обчислювальних можливостей та енергоспоживання на периферії; обмежені ресурси: пристрої edge мають обмежену обчислювальну потужність, пам'ять та енергію, що ускладнює виконання складних ML-моделей; нестабільні мережі: периферійні вузли часто мають нестійке або переривчасте з'єднання та конфіденційність даних: необхідність уникати централізації чутливих даних, що вимагає приватного або федеративного навчання. Автори запропонували класифікацію підходів до розподіленого ML на edge за типом навчання: централізоване, федеративне, кооперативне, самоорганізоване; архітектура: client-server, peer-to-peer, багаторівнева (multi-tier); стратегія обміну: передача ваг моделей, градієнтів або оновлень параметрів; також обговорюється роль стискання та асинхронного навчання; механізми конфіденційності: диференційна приватність, захищене багатостороннє обчислення (SMC), гомоморфне шифрування.

Особливу увагу приділено федеративному навчанню FL (Federated Learning), яке є однією з провідних технологій для edge ML. Проаналізовано такі алгоритми, як FedAvg, FedProx, Scaffold, та їхню здатність справлятися з гетерогенними даними, затримками та втратою зв'язку. Автори окреслили перспективи подальших досліджень, які включають розробку більш стійких до збоїв та динамічних топологій моделей, інтеграцію edge ML із мережами 5G/6G, автоматизацію оптимального розподілу ML-моделей з урахуванням ресурсів пристроїв та балансування обчислень між хмарою, fog і edge-рівнями.

Постановка завдання. Метою даного дослідження є аналіз та систематизація сучасних наукових підходів до реалізації паралельних, розподілених та апаратно-прискорених обчислень у хмарному й edge-середовищах для обробки великих даних і реалізації методів машинного навчання, з акцентом на їх ефективність, масштабованість, безпеку та практичну застосовність. Стаття також має на меті виявити поточні обмеження та проблеми, а також сформувати підґрунтя для подальших досліджень у цьому міждисциплінарному напрямі.

Виклад основного матеріалу. Станом на сьогодні паралельна обробка великих даних і задач машинного навчання активно досліджується в контексті програмного, апаратного забезпечення та хмарних платформ.

Серед ключових програмних рішень варто виділити *Apache Spark*, що є розподіленою обчислювальною платформою з вбудованим модулем MLlib для машинного навчання. Завдяки обробці даних у пам'яті Spark забезпечує високу продуктивність, особливо при роботі з великими обсягами інформації.

Іншим важливим інструментом є *Dask* – бібліотека для паралельних обчислень у Python, яка легко інтегрується з популярними науковими бібліотеками, такими як Pandas, NumPy та Scikit-learn. Dask підтримує масштабування від окремих машин до розподілених кластерів, що робить її зручною для адаптивного використання.

TensorFlow – це потужна open-source платформа від Google для розробки та розгортання моделей машинного навчання, яка особливо ефективна в масштабованих хмарних середовищах.

PyTorch – це гнучка бібліотека для глибокого навчання, розроблена Facebook, яка відома своєю зручністю для досліджень та підтримкою динамічних обчислювальних графів.

GraphLab – ще один приклад ефективної розподіленої платформи, яка спеціалізується на обробці графових даних і використовується в завданнях, що потребують високої взаємозв'язаної структури, таких як рекомендаційні системи або соціальні мережі.

З апаратної точки зору паралельні обчислення найчастіше реалізуються через спеціалізовані процесори.

GPU (Graphics Processing Unit) вже давно стали стандартом у прискоренні машинного навчання завдяки своїй архітектурі, що дозволяє одночасно обробляти тисячі потоків. У деяких випадках використання GPU в хмарному середовищі дозволяє досягти прискорення обчислень до 215 разів у порівнянні з традиційними CPU.

TPU (Tensor Processing Unit) – це спеціалізовані чіпи, розроблені Google для оптимізації роботи з TensorFlow і глибоким навчанням.

FPGA (Field-Programmable Gate Array) – це програмований апаратний пристрій, який дозволяє створювати спеціалізовані обчислювальні архітектури для прискорення задач машинного навчання та обробки великих даних з високою енергоефективністю.

Новим типом процесора є *IPU (Intelligence Processing Unit)* від компанії Graphcore, який орієнтований на обробку графових структур, характерних для багатьох задач штучного інтелекту. Ще більш інноваційним рішенням є *WSE-2 (Wafer-Scale Engine 2)* – найбільший у світі комп'ютерний чіп, розроблений компанією Cerebras Systems для прискорення AI-завдань, особливо в умовах високої складності моделей.

У сфері хмарних обчислень провідні провайдери надають інфраструктуру, орієнтовану на потреби машинного навчання та обробки великих даних. *Amazon Web Services (AWS)* пропонує спеціалізовані екземпляри з GPU та TPU, а також власні AI-чіпи Trainium, розроблені для підвищення продуктивності та зменшення вартості тренування моделей.

Google Cloud Platform (GCP) інтегрує TPU та GPU у свої сервіси, зокрема для роботи з TensorFlow.

Microsoft Azure надає широкий спектр обчислювальних ресурсів для машинного навчання, включаючи GPU-інстанси, і підтримує різноманітні фреймворки та середовища для аналізу великих даних.

Методи паралельних обчислень, що використовуються в цих контекстах, можна класифікувати за трьома основними підходами (табл. 3).

Data parallelism передбачає розподіл даних між кількома обчислювальними вузлами, кожен з яких виконує однакову операцію на окремому фрагменті даних.

Model parallelism полягає в розподілі частин самої моделі між обчислювальними вузлами, що дозволяє одночасну обробку різних її компонентів.

Pipeline parallelism реалізується через поділ обчислювального процесу на етапи (як на конвеєрі), які виконуються послідовно на різних вузлах. Такий підхід дозволяє ефективно використовувати ресурси та зменшувати затримки при обробці великих обсягів даних.

Проведений порівняльний аналіз дозволяє оцінити ключові характеристики різних методів паралельного та розподіленого обчислення, а також апаратних і програмних засобів, які використовуються для реалізації систем машинного навчання у хмарних і периферійних середовищах.

Таблиця 3

Порівняльний аналіз використання різних методів паралельних обчислень

Засіб / Платформа	Тип обчислень	Продуктивність	Масштабованість	Вартість	Сумісність
Apache Spark	Data Parallelism	Висока	Висока	Середня	Висока
Dask	Data/Model Parallelism	Висока	Висока	Низька	Висока
GraphLab	Graph Parallelism	Середня	Середня	Середня	Середня
GPU (NVIDIA)	Data Parallelism	Дуже висока	Висока	Висока	Висока
TPU (Google)	Data Parallelism	Дуже висока	Висока	Висока	Середня
IPU (Graphcore)	Graph Parallelism	Висока	Середня	Висока	Низька
WSE-2 (Cerebras)	Model Parallelism	Дуже висока	Висока	Дуже висока	Низька

Data Parallelism демонструє високу продуктивність (85%) і масштабованість (90%), однак дещо поступається в гнучкості (70%), що пояснюється складністю адаптації до гетерогенних даних.

Model Parallelism має порівняні характеристики, але вищу гнучкість (75%), оскільки дозволяє ефективно розділяти обчислювальні навантаження між різними компонентами моделей.

Pipeline parallelism демонструє збалансовані показники з перевагою у гнучкості (80%), що обумовлено можливістю обробки даних поетапно, хоча він поступається іншим підходам у продуктивності (75%).

Найкращі результати в усіх трьох категоріях демонструє гібридний підхід – 90% продуктивності, 95% масштабованості та 85% гнучкості – завдяки комбінації кількох форм паралелізму, що дозволяє адаптувати систему під конкретні вимоги.

Серед апаратних засобів GPU показує найвищу продуктивність (95%) та хорошу масштабованість (85%), але поступається у гнучкості (70%) через високу енергоспоживаність і складність оптимізації під різні типи задач.

TPU має нижчу продуктивність (90%) і масштабованість (90%), але ще меншу гнучкість (65%), оскільки розроблений для вузького класу завдань.

FPGA, навпаки, забезпечує найвищу гнучкість (90%), хоча дещо відстає у продуктивності (85%) і масштабованості (80%), але залишається цінним варіантом у специфічних умовах.

Щодо програмних засобів, TensorFlow демонструє високу продуктивність (90%) і хорошу масштабованість (85%), при цьому підтримуючи належну гнучкість (80%), що робить TensorFlow популярним вибором для систем з розподіленим навчанням.

PyTorch поступається в продуктивності (85%) і масштабованості (80%), але забезпечує вищу гнучкість (85%) завдяки інтуїтивно зрозумілому синтаксису та активній спільноті. Apache Spark, хоча й не є спеціалізованим інструментом для глибокого навчання, проте показує високі результати в масштабованості (90%) та прийнятну продуктивність (80%), але має обмежену гнучкість (75%) у контексті виконання ML-задач.

Таким чином, результати порівняння свідчать про відсутність універсального рішення – вибір методів та засобів слід здійснювати з урахуванням специфіки задачі, обчислювального середовища та вимог до продуктивності, масштабованості та гнучкості.

Висновки. Усі розглянуті наукові статті охоплюють широкий спектр підходів до реалізації паралельних, розподілених та апаратно-прискорених обчислень у хмарному й периферійному середовищах для обробки великих даних і виконання алгоритмів машинного навчання. Попри значний прогрес у цій галузі, автори робіт констатують наявність низки проблем, що стримують ефективне впровадження таких технологій на практиці.

Однією з ключових проблем залишається гетерогенність систем і нестабільність інфраструктури. Величезна різноманітність апаратних засобів – від потужних GPU у хмарі до обмежених edge-пристроїв – у поєднанні з нестабільними або асиметричними мережевими умовами ускладнює побудову уніфікованих, масштабованих рішень. При цьому забезпечення балансу між точністю, швидкістю обробки та дотриманням політик конфіденційності залишається ще одним складним завданням, яке не має універсального вирішення. Розподілені системи машинного навчання часто втрачають точність або обчислювальну ефективність, а реалізація механізмів приватності ще більше ускладнює архітектуру. Особливої уваги потребує ситуація з периферійними обчисленнями, де ресурси пристроїв є істотно обмеженими. Це стримує повноцінне використання сучасних моделей глибокого навчання навіть за наявності апаратних прискорювачів. Окремою проблемою є нестача стандартизації між численними фреймворками й платформами, що перешкоджає створенню узгоджених розподілених екосистем і підвищує поріг входження для розробників. Більшість існуючих рішень мають високу складність реалізації, потребують спеціалізованих знань в галузі паралельного програмування, а також часто не адаптовані до умов прикладного використання в динамічному середовищі. Також системи не демонструють стабільної масштабованості – їх продуктивність помітно погіршується при зміні розміру вхідних даних, кількості обчислювальних вузлів або доступних обчислювальних потужностей. Суттєвим недоліком поточного стану досліджень є також відсутність систематизованих порівняльних оцінок – методи часто тестуються в різних умовах, на різних наборах даних та апаратних конфігураціях, що унеможливлює їх об'єктивне порівняння. Низка принципових питань досі залишається відкритою. Наприклад, як знайти баланс між обчислювальною ефективністю, точністю прогнозування і конфіденційністю даних у розподілених ML-системах? Які архітек-

турні підходи найкраще масштабуються у гетерогенному середовищі? Які апаратні прискорювачі забезпечують найкращі результати для конкретних типів задач? Як уніфікувати розробку в умовах мультихмарної або cloud-edge архітектури?

Усе це підкреслює необхідність подальшого комплексного аналізу та дослідження ефективності існуючих рішень, їх обмежень, сфери застосування та потенціалу розвитку, що дозволить сформувати стратегічне бачення майбутнього розвитку паралельних і розподілених обчислень у хмарних середовищах для обробки великих даних і реалізації машинного навчання.

Список літератури:

1. Ling Zhou, Ziyang Gong, and Pengcheng Xiang. Distributed Computing and Inference for Big Data. *Annual Review of Statistics and Its Application*. Vol. 11. 2024. P. 533-551. DOI: https://doi.org/10.1146/annurev-statistics-040522-021241.

2. Samanta A., Hatai I., Mal A.K. A Survey on Hardware Accelerator Design of Deep Learning for Edge Devices. *Wireless Personal Communications*. Vol. 137. 2024. P. 1715–1760. DOI: https://doi.org/10.1007/s11277-024-11443-2.

3. John Langford. Parallel machine learning on big data. *XRDS: Crossroads*. Vol. 19. Issue 1. 2012, P. 60–62. DOI: https://doi.org/10.1145/2331042.2331060.

4. Mills N., Issadeen Z., Matharaarachchi A. et al. A cloud-based architecture for explainable Big Data analytics using self-structuring Artificial Intelligence. *Discover Artificial Intelligence*. Vol. 4 (33). 2024. DOI: https://doi.org/10.1007/s44163-024-00123-6.

5. Xie Y, Oniga S. A Comprehensive Review of Hardware Acceleration Techniques and Convolutional Neural Networks for EEG Signals. *Sensors*. 2024. 24(17): 5813. DOI: https://doi.org/10.3390/s24175813.

6. Chih-Fong Tsai, Wei-Chao Lin, and Shih-Wen Ke. Big data mining with parallel computing. *Journal of Systems and Software*. Vol. 122. 2016. P. 83–92. DOI: https://doi.org/10.1016/j.jss.2016.09.007.

7. Wang M., Yang T., Flechas M.A., Harris P., Hawks B., Holzman B., Knoepfel K., Krupa J., Pedro K., Tran N. GPU-Accelerated Machine Learning Inference as a Service for Computing in Neutrino Experiments. *Frontiers in Big Data*. 2021. 3:604083. DOI: 10.3389/fdata.2020.604083.

8. Katherine Evans, Ethan Stewart, Grace Foster. Comprehensive Review of Cloud-Driven Machine Learning: A Comparative Analysis of AWS, Azure, and Google Cloud. *TechRxiv*. January 05. 2025. DOI: 10.36227/techrxiv.173611042.27406103/v1.

9. Luke Kljucaric and Alan D. George. Deep Learning Inferencing with High-performance Hardware Accelerators. *ACM Transactions on Intelligent Systems and Technology*. 14 (4). Article 68. 2023. P. 1–25. DOI: https://doi.org/10.1145/3594221.

10. Jingke Tu, Lei Yang, and Jiannong Cao. Distributed Machine Learning in Edge Computing: Challenges, Solutions and Future Directions. *ACM Computing Surveys*. 57 (5). Article 132. 2025. P. 1–37. DOI: https://doi.org/10.1145/3708495.

Oleshchenko L.M. INVESTIGATION OF HARDWARE AND SOFTWARE TOOLS FOR PARALLEL PROCESSING OF BIG DATA AND MACHINE LEARNING TASKS IN CLOUD ENVIRONMENT

In the context of rapidly growing data volumes and increasing computational complexity, there is a pressing need for high-performance approaches to Big Data processing and machine learning implementation. One of the key directions in this field is the use of parallel and distributed computing, supported by specialized hardware and software tools within cloud and edge environments. This article is dedicated to the analysis of the development of hardware architectures (GPU, TPU, FPGA), big data processing technologies in cloud environments, and the study of modern machine learning tools (TensorFlow, PyTorch, Apache Spark), which enable efficient data processing and analysis in cloud infrastructures. The article outlines the advantages and disadvantages of different architectures in terms of performance, scalability, and flexibility. It is found that GPUs deliver the highest performance but are less flexible compared to FPGAs, TPUs show high efficiency for narrow, specialized tasks but are limited in adaptability to heterogeneous environments. It has been determined that among software frameworks TensorFlow stands out for its high performance and scalability; PyTorch offers better convenience for model development and adaptation; Apache Spark, despite its general-purpose nature, demonstrates strong efficiency in distributed data processing. Several open challenges are highlighted, including the difficulty of balancing accuracy, speed, and privacy in distributed ML systems; the lack of unified tools for integrating heterogeneous computing platforms; limited capabilities of edge devices to support complex ML models; and the absence of systematic comparisons of methods under unified

experimental conditions. The aim of this research is to generalize existing approaches to organizing parallel big data processing in cloud environments, identify key trends, compare their effectiveness, and define current challenges requiring further research. An analysis was conducted and strategic directions were formulated for optimizing the hardware and software architecture of big data processing systems, taking into account the requirements for adaptability, efficiency, and scalability in a dynamic environment.

Key words: *parallel computing, cloud technologies, big data, machine learning, GPU, TPU, FPGA, TensorFlow, PyTorch, Apache Spark, distributed systems, performance, scalability, flexibility.*