DOI: https://doi.org/10.36910/6775-2524-0560-2024-56-13
УДК 004.652
**Бондарчук Олег Іванович[1],** магістр
http://orcid.org/0009-0003-9626-1124
**Козуб Владислав Юрійович[2]**, д.ф. з комп.н., асистент
http://orcid.org/0000-0003-2710-7206
**Козуб Юрій Гордійович[2],** д.т.н., професор
http://orcid.org/0000-0002-3366-6031
[1] Akademia WSB/Університет WSB, м. Домброва Гурнича, Польща
[2]Луганський національний університет імені Тараса Шевченка, м. Полтава, Україна

# АНАЛІЗ ЕФЕКТИВНОСТІ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ В ОБРОБЦІ ВЕЛИКИХ ДАНИХ

**Бондарчук О.І., Козуб В.Ю., Козуб Ю.Г. Аналіз ефективності алгоритмів машинного навчання в обробці великих даних.** Метою статті є аналіз ефективності алгоритмів машинного навчання та пошук оптимальних підходів для їх використання в умовах високого навантаження і великих обсягів інформації. Особливу увагу приділено завданням, які потребують високої точності та швидкості, зокрема фінансові прогнози, медична діагностика та аналіз поведінкових даних. Методологія дослідження включала порівняльний аналіз різних алгоритмів машинного навчання, таких як лінійна регресія, дерева рішень, метод опорних векторів та глибокі нейронні мережі. Оцінювалися ключові фактори, що впливають на швидкість та точність обробки даних, зокрема розмір даних, складність моделей, обчислювальні ресурси та якість вхідних даних. Було проведено серію експериментів на реальних наборах даних, щоб оцінити результативність кожного алгоритму з погляду точності, часу навчання та потреби в обчислювальних ресурсах. У результатах дослідження встановлено, що глибокі нейронні мережі забезпечують високу точність на неструктурованих даних, але мають значні вимоги до обчислювальних ресурсів і часу навчання. Алгоритми, такі як лінійна регресія та дерева рішень, показали високу швидкість обробки на простіших наборах даних, проте їх точність знижується зі збільшенням складності завдань. Метод опорних векторів продемонстрував ефективність у завданнях класифікації та прогнозування, зокрема у фінансових та медичних застосуваннях. Випадкові ліси виявилися ефективними для текстової класифікації та забезпечили баланс між швидкістю і точністю. Висновки свідчать, що вибір алгоритму залежить від специфіки завдання, розміру даних та вимог до точності. Оптимізація моделей для роботи в умовах розподілених обчислень є ключовим напрямом для підвищення продуктивності, оскільки дозволяє паралелізувати процеси й зменшити час навчання.
**Ключові слова**: штучний інтелект, предиктивна аналітика, обробка даних, алгоритми класифікації, оптимізація моделі

**Bondarchuk O., Kozub V., Kozub Yu. Analysis of the effectiveness of machine learning algorithms in big data processing.** The aim of this article is to analyze the effectiveness of machine learning algorithms and identify optimal approaches for their use under heavy loads and large volumes of information. Special attention is paid to tasks requiring high accuracy and speed, such as financial forecasting, medical diagnostics, and behavioral data analysis. The research methodology included a comparative analysis of various machine learning algorithms, such as linear regression, decision trees, support vector machines, and deep neural networks. Key factors influencing the speed and accuracy of data processing were evaluated, including data size, model complexity, computational resources, and data quality. A series of experiments on real-world datasets was conducted to assess each algorithm's performance in terms of accuracy, training time, and computational resource demands. The results showed that deep neural networks provide high accuracy on unstructured data, but require significant computational resources and training time. Algorithms such as linear regression and decision trees demonstrated high processing speed on simpler datasets, though their accuracy decreases as task complexity increases. The support vector machine method proved effective for classification and prediction tasks, particularly in financial and medical applications. Random forests were found to be efficient for text classification, providing a balance between speed and accuracy. The conclusions indicate that the choice of algorithm depends on the task's specific characteristics, data size, and accuracy requirements. Optimizing models for distributed computing environments is a key direction for improving productivity, as it allows for parallelization and reduces training time.
**Keywords:** artificial intelligence, predictive analytics, data processing, classification algorithms, model optimization

**Постановка наукової проблеми.** Обробка великих даних стала однією з ключових проблем сучасної науки й техніки через стрімке збільшення обсягу інформації, що генерується в різних галузях, таких як медицина, економіка, соціальні мережі та наукові дослідження. Алгоритми машинного навчання забезпечують ефективні інструменти для аналізу та інтерпретації цих даних, що відкриває нові можливості для прийняття рішень та прогнозування. Проте ефективність цих алгоритмів залишається під питанням через складність їх налаштування для великих і неоднорідних наборів даних. Основним викликом є необхідність розробки таких методів обробки, які дозволять оптимізувати час обчислень, підвищити точність прогнозів та адаптувати алгоритми до умов розподілених обчислень. Це завдання має важливе практичне значення для підвищення продуктивності великих систем аналізу даних, зокрема у сфері бізнес-аналітики, фінансових ринків, охорони здоров'я та наукових досліджень.

**Аналіз останніх досліджень і публікацій.** Аналіз наукових досліджень, присвячених ефективності алгоритмів машинного навчання в обробці великих даних, демонструє різноманітність підходів і технологій, що застосовуються для покращення продуктивності й точності в різних галузях.

О. Данчак і М. Войтюк [1] акцентують на ефективності сховищ даних, спеціально налаштованих для алгоритмів машинного навчання, які забезпечують швидкий доступ і обробку великих масивів інформації. Вони пропонують рішення для підвищення продуктивності таких алгоритмів у практичних сценаріях.

В. Нестеров [2] досліджує вплив аналітики великих даних на бізнес у цифрову епоху, наголошуючи на важливості машинного навчання для покращення бізнес-процесів і прийняття рішень.

К. Лозовська [3] аналізує застосування методів машинного навчання для оцінки показників інтернет-ресурсів. На думку дослідниці, алгоритми машинного навчання є ключовими інструментами для аналізу великих масивів даних, що дозволяє автоматизувати процеси й забезпечити точні прогнози.

K. Ngiam та I. Khor [4] зосереджуються на використанні великих даних та алгоритмів машинного навчання в охороні здоров'я. Їхнє дослідження демонструє значення алгоритмів для покращення якості надання медичних послуг і прогнозування ризиків.

V. Koliesetty і D. Rajput [5] та M. Jan і співавтори [6] здійснюють огляди застосування машинного навчання для аналізу великих даних, підкреслюючи роль глибокого навчання в підвищенні ефективності обробки даних у різних галузях. Їхні дослідження підтверджують важливість розробки спеціалізованих моделей для адаптації під конкретні потреби.

Колектив авторів на чолі з G. Saranya [7] та G. Saranya і S. Asha [8] аналізують продуктивність алгоритмів машинного навчання для систем виявлення вторгнень у систему охорони здоров'я. Вони наголошують на важливості вибору оптимальних моделей для забезпечення безпеки та надання точних прогнозів.

J. Shang і Z. You [9] вивчають вплив машинного навчання на розвиток розумного виробництва, підкреслюючи, що успішна інтеграція алгоритмів машинного навчання в процеси виробництва дозволяє оптимізувати управління та підвищити ефективність роботи.

A. Adadi [10] робить огляд алгоритмів, які забезпечують ефективну обробку даних у великих масштабах, наголошуючи на важливості економічності моделей для роботи з обмеженими ресурсами. Це узгоджується з висновками колективу авторів під керівництвом M. Amanullah [11], який досліджує глибоке навчання та технології великих даних у сфері безпеки IoT.

O. Faker та E. Dogdu [12] продовжують тему безпеки, зосереджуючись на виявленні вторгнень за допомогою глибокого навчання, підкреслюючи значення адаптації моделей під конкретні потреби індустрій, де безпека є пріоритетом.

Дослідження в цій галузі демонструють важливість удосконалення алгоритмів машинного навчання та оптимізації сховищ даних для забезпечення ефективної обробки великих обсягів інформації.

**Виділення раніше невирішених частин проблеми.** Попри різноманітність досліджень, залишаються нерозв'язаними питання дослідження наявних алгоритмів машинного навчання з погляду їхньої ефективності при роботі з великими даними; оцінки ключових факторів, що впливають на швидкість і точність обробки даних; порівняння результативності різних алгоритмів на основі практичних експериментів і наукових досліджень.

**Метою статті** є аналіз ефективності алгоритмів машинного навчання для обробки великих даних та виявлення оптимальних підходів до їх використання в умовах високого навантаження та великих обсягів інформації.

Завдання статті:

1)дослідити наявні алгоритми машинного навчання з погляду їхньої ефективності при роботі з великими даними;

2) оцінити ключові фактори, що впливають на швидкість і точність обробки даних;

3) порівняти результативність різних алгоритмів на основі практичних експериментів і наукових досліджень;

4) вивчити проблеми та обмеження, пов'язані з масштабуванням алгоритмів для великих обсягів даних;

5) розробити рекомендації щодо оптимізації алгоритмів для роботи в умовах розподілених обчислень.

**Виклад основного матеріалу.** Дослідження наявних алгоритмів машинного навчання з погляду їхньої ефективності при роботі з великими даними є важливим аспектом аналізу продуктивності сучасних систем обробки інформації. Алгоритми, такі як лінійна регресія, дерева рішень, метод опорних векторів та глибокі нейронні мережі, мають різні підходи до обробки великих обсягів даних, що впливає на їх точність та швидкість виконання. Наприклад, алгоритми глибокого навчання, хоча й забезпечують високу точність, вимагають значних обчислювальних ресурсів та часу на тренування, особливо при збільшенні розмірів даних. З іншого боку, простіші алгоритми, такі як лінійна регресія, можуть бути швидкими та менш ресурсомісткими, але їх ефективність може знижуватися при складних і великомасштабних наборах даних. У таблиці 1 наведено порівняння основних алгоритмів за кількома критеріями, такими як швидкість навчання, масштабованість та точність прогнозів.

Табл. 1. Порівняння основних алгоритмів машинного навчання

| Алгоритм | Опис алгоритму | Швидкість навчання | Точність прогнозів | Масштабованість | Обчислювальні вимоги |
|---|---|---|---|---|---|
| Лінійна регресія | Статистичний метод для моделювання залежності між змінними | Висока | Низька в складних завданнях | Висока | Низькі |
| Дерева рішень | Деревоподібна структура для прийняття рішень на основі правил | Середня | Середня | Обмежена у великих наборах | Середні |
| Метод опорних векторів | Алгоритм для класифікації та регресії, що знаходить оптимальну гіперплощину | Низька | Висока | Середня | Високі |
| Глибокі нейронні мережі | Алгоритм із великою кількістю шарів нейронів для виявлення складних закономірностей. | Низька | Висока | Висока | Дуже високі |

Лінійна регресія є одним із найстаріших алгоритмів і використовується переважно для простих моделей передбачення в таких сферах, як економіка, аналітика продажів або прогнозування клімату. Вона дозволяє швидко отримати результат, однак її точність може бути низькою в умовах складних нелінійних даних. Дерева рішень широко застосовуються в галузях, де необхідно пояснити процес прийняття рішень, наприклад, в медицині для діагностичних завдань або у фінансах для оцінки ризиків. Вони надають візуально зрозумілу структуру, що легко інтерпретується, але їх продуктивність знижується при великій кількості ознак. Метод опорних векторів, завдяки своїй здатності працювати з високовимірними даними, застосовується в біоінформатиці, обробці зображень і машинному перекладі, але має високі вимоги до обчислювальних потужностей. Глибокі нейронні мережі, які здатні вивчати надзвичайно складні зв'язки в даних, сьогодні є основою таких технологій, як розпізнавання мови, зображень та прогнозування поведінки в реальному часі [2], [5]. Вони широко використовуються в таких галузях, як штучний інтелект, автономні транспортні засоби та персоналізована медицина.

У сучасних умовах штучний інтелект є важливим інструментом для оптимізації процесів обробки великих обсягів даних, особливо в складних середовищах, де швидкість та точність рішень є критично важливими. Інтеграція методів машинного навчання та штучного інтелекту дає змогу використовувати алгоритми класифікації для прогнозування й автоматизації процесів прийняття рішень у реальному часі.

Наприклад, штучний інтелект може використовувати алгоритми класифікації, такі як підтримуючі векторні машини (SVM), або глибокі нейронні мережі (DNN), для аналізу поведінкових патернів користувачів на онлайн-платформах. Такі моделі здатні ефективно опрацьовувати великі обсяги даних, виявляти ключові закономірності та на основі цих даних

проводити прогнозування, як-от передбачення потенційних ризиків або визначення ключових трендів (табл. 2).

Табл. 2. Порівняння алгоритмів штучного інтелекту для класифікації та прогнозування великих даних

| Алгоритм ШІ | Призначення | Переваги | Недоліки |
|---|---|---|---|
| Глибокі нейронні мережі (DNN) | Обробка складних і неструктурованих даних | Висока точність, можливість глибокого навчання | Потребують великих обчислювальних ресурсів |
| Підтримуючі векторні машини (SVM) | Класифікація та прогнозування на основі великих даних | Висока точність, добре працює з лінійно роздільними даними | Менша ефективність для великих неструктурованих даних |
| Багатошарова персептронна модель (MLP) | Прогнозування та класифікація в реальному часі | Здатність обробляти дані в реальному часі | Чутливість до перенавчання моделі |

Глибокі нейронні мережі застосовуються для обробки великих наборів даних, таких як зображення або текст, де необхідно знайти складні закономірності. Вони забезпечують високу точність, але потребують значних обчислювальних потужностей для навчання й прогнозування. Алгоритми підтримуючих векторних машин (SVM) підходять для розв'язання завдань класифікації, де дані можуть бути лінійно розділеними, що дозволяє досягти високої точності прогнозів. Мультишарова персептронна модель (MLP) використовується для обробки великих потоків даних у реальному часі, що дозволяє виконувати прогнозування на основі даних, що надходять в режимі реального часу.

На практиці такі алгоритми можуть використовуватися для підвищення ефективності систем штучного інтелекту у сфері фінансів, медицини та промисловості.

Процес вибору алгоритму залежно від обсягу даних та вимог до точності прогнозування представлено на рис. 1


Рис. 1. Процес вибору алгоритму залежно від обсягу даних та вимог до точності прогнозування

У сучасних умовах алгоритми машинного навчання відіграють ключову роль в ефективному обробленні величезних масштабів даних. Прості алгоритми, такі як лінійна регресія, використовуються при невеликому обсязі даних для швидких прогнозів з обмеженими вимогами до ресурсів. Для середніх обсягів даних та потреби в інтерпретації результатів краще підходять дерева рішень, особливо в ситуаціях, коли важлива наочність процесу прийняття рішень. Якщо ж дані

складні та їх обсяг значний, метод опорних векторів забезпечить високу точність, але за умови наявності достатніх обчислювальних ресурсів. Нарешті, для роботи з величезними масивами даних найкраще підходять глибокі нейронні мережі, здатні виявляти складні взаємозв'язки й використовувати сучасні обчислювальні системи для масштабованого навчання й прогнозування.

Обчислювальні ресурси є ще одним визначальним чинником, оскільки алгоритми потребують значної кількості ресурсів для роботи з великими наборами даних. Алгоритми з високими вимогами до ресурсів, такі як метод опорних векторів або нейронні мережі, можуть уповільнювати процес навчання та прогнозування в умовах обмеженого доступу до обчислювальних ресурсів. Окрім цього, якість вхідних даних також впливає на точність моделі. Алгоритми можуть забезпечити кращі результати, якщо дані попередньо оброблені та очищені від шуму або пропущених значень.

У таблиці 3 наведено порівняння основних факторів, що впливають на швидкість і точність обробки даних різними алгоритмами машинного навчання.

Табл. 3. Порівняння основних факторів, що впливають на швидкість і точність обробки даних різними алгоритмами машинного навчання.

| Фактор | Вплив на швидкість обробки | Вплив на точність обробки | Пояснення |
|---|---|---|---|
| Розмір даних | Знижує швидкість | Залежить від алгоритму | Великий обсяг даних уповільнює роботу складніших алгоритмів, зокрема нейронних мереж |
| Складність моделі | Знижує швидкість | Підвищує точність | Більша кількість параметрів або шарів вимагає більше часу для навчання та налаштувань. |
| Обчислювальні ресурси | Підвищує швидкість | Не впливає безпосередньо | Доступ до потужних серверів або розподілених систем прискорює обробку даних. |
| Якість вхідних даних | Не впливає значно | Підвищує точність | Попередня обробка та очищення даних знижують ймовірність помилок у прогнозах. |

У сучасних умовах ключові фактори, що впливають на швидкість і точність обробки даних, значно еволюціонували завдяки розвитку обчислювальних технологій та новим підходам до управління даними [15]. Обсяг даних постійно збільшується через широке застосування Інтернету речей (IoT), соціальних медіа та великих комерційних і наукових систем збору даних. Це вимагає використання більш складних алгоритмів і потужніших обчислювальних ресурсів. У зв'язку з цим масштабованість обчислень і можливості хмарних інфраструктур стали критичними для збереження швидкості обробки навіть при значних обсягах даних.

Складність моделей зросла разом із розвитком алгоритмів глибокого навчання, які тепер можуть виконувати завдання, що раніше були недоступні через обмеження в обчислювальних потужностях. Однак збільшення кількості параметрів і шарів у нейронних мережах призвело до підвищення вимог до ресурсів, що робить необхідною інтеграцію розподілених систем обробки. Водночас якість даних залишається важливим аспектом, оскільки навіть найкращі алгоритми не можуть дати точних результатів за відсутності належної попередньої обробки даних. Використання сучасних методів очищення та нормалізації даних дозволяє значно підвищити точність моделей, мінімізуючи вплив шумів або пропущених значень.

Порівняння результативності алгоритмів машинного навчання на основі практичних експериментів є одним із ключових способів визначення найбільш ефективних підходів до обробки великих даних. У ході експериментів було проведено тестування декількох популярних алгоритмів на різних наборах даних із реальних застосувань, таких як аналіз медичних зображень, прогнозування фінансових ринків і класифікація текстових документів. Метою експериментів було оцінити ефективність кожного алгоритму з погляду точності, швидкості навчання та вимог до обчислювальних ресурсів, а також визначити їхню придатність для різних типів завдань.

У таблиці 4 наведено результати експериментів, що включають основні показники роботи алгоритмів при обробці наборів даних різного розміру та складності.

Табл. 4. Основні показники роботи алгоритмів при обробці наборів даних різного розміру та складності

| Алгоритм | Набір даних | Точність (%) | Час навчання (с) | Обчислювальні ресурси (CPU) | Обсяг даних (млн записів) | Використання на практиці |
|---|---|---|---|---|---|---|
| Нейронні мережі | Медичні зображення | 94 | 600 | Високі | 2.5 | Діагностика захворювань |
| Метод опорних векторів | Фінансові ринки | 89 | 450 | Середні | 1.2 | Прогнозування трендів |
| Випадкові ліси | Текстова класифікація | 87 | 120 | Середні | 3.1 | Аналіз документів |
| Лінійна регресія | Економічні показники | 72 | 30 | Низькі | 0.5 | Просте моделювання |
| K-ближчих сусідів | Маркетингові дані | 78 | 300 | Середні | 0.7 | Сегментація клієнтів |

Експерименти проводилися з різними наборами даних, щоб оцінити ефективність алгоритмів в умовах, наближених до реальних застосувань. Наприклад, для медичних зображень використовувалися нейронні мережі, які продемонстрували найвищу точність (94%), але час навчання був значно тривалішим порівняно з іншими алгоритмами, що пояснюється великою кількістю параметрів, необхідних для обробки зображень. Метод опорних векторів був успішно застосований для прогнозування на фінансових ринках, досягаючи точності 89% за середнього часу навчання (450 секунд) та помірного використання ресурсів.

Випадкові ліси виявилися ефективними для класифікації текстів, забезпечуючи точність 87% з відносно швидким навчанням (120 секунд), що робить їх оптимальними для завдань із великими наборами текстових даних. Лінійна регресія використовувалася для економічних показників, де точність 72% виявилася прийнятною для простих завдань прогнозування, але її низькі вимоги до обчислювальних ресурсів і час навчання всього в 30 секунд роблять цей алгоритм ефективним у простих моделях. Алгоритм K-ближчих сусідів був використаний для сегментації маркетингових даних, де показав середню точність (78%) та відносно високий час навчання (300 секунд), що обмежує його використання на великих наборах даних.

Оцінка результатів показує, що вибір алгоритму залежить від конкретного завдання, його складності, обсягу даних та вимог до точності. Для складних завдань, таких як медична діагностика, нейронні мережі забезпечують високу результативність, тоді як для швидкого аналізу економічних показників краще підходять простіші моделі, такі як лінійна регресія. У таблиці 5 проаналізовано взаємозв'язок між обсягом даних та часом навчання алгоритмів.

Табл. 5. Взаємозв'язок між обсягом даних та часом навчання алгоритмів

| Алгоритм | Обсяг даних (млн записів) | Час навчання (с) | Залежність часу від обсягу даних |
|---|---|---|---|
| Нейронні мережі | 1.0 | 300 | Лінійне збільшення |
| Нейронні мережі | 2.5 | 600 | Лінійне збільшення |
| Метод опорних векторів | 0.6 | 200 | Лінійне збільшення |
| Метод опорних векторів | 1.2 | 450 | Лінійне збільшення |
| Випадкові ліси | 1.5 | 70 | Нелінійне збільшення |
| Випадкові ліси | 3.1 | 120 | Нелінійне збільшення |

Результати експериментів показують, що для нейронних мереж та методу опорних векторів час навчання збільшується лінійно зі збільшенням обсягу даних. Це пояснюється складністю цих моделей, які потребують більше ресурсів із кожним новим записом у наборі даних. Водночас випадкові ліси показують нелінійне збільшення часу навчання, що свідчить про їхню здатність ефективніше обробляти великі обсяги даних завдяки розподіленню обчислень між різними деревами прийняття рішень.

Масштабування алгоритмів машинного навчання для обробки великих обсягів даних є складним завданням, яке супроводжується низкою проблем і обмежень. Однією з ключових проблем є потреба в значних обчислювальних ресурсах, які стають усе важливішими зі збільшенням обсягу даних. Зростання складності даних, що обробляються, призводить до того, що навіть найбільш оптимізовані алгоритми можуть не справлятися з вимогами щодо швидкості навчання та точності результатів. Це створює проблеми не лише для наукових досліджень, але й для практичних застосувань у сфері фінансів, медицини та інших галузях, де рішення необхідно приймати оперативно.

Ще однією важливою проблемою є низька масштабованість деяких алгоритмів. Наприклад, традиційні моделі, такі як лінійна регресія чи K-ближчих сусідів, мають обмежену здатність ефективно працювати з великими наборами даних через зростання обчислювальних витрат або зниження точності результатів. Крім того, багато сучасних алгоритмів, таких як нейронні мережі, демонструють високу точність на малих обсягах даних, але їх масштабування на великі набори потребує спеціалізованого обладнання, зокрема графічних процесорів (GPU) або розподілених систем, що збільшує вартість і складність обчислень.

У таблиці 6 представлено основні проблеми та обмеження, пов'язані з масштабуванням різних алгоритмів для великих обсягів даних.

Табл. 6. Основні проблеми та обмеження, пов'язані з масштабуванням різних алгоритмів для великих обсягів даних

| Алгоритм | Проблеми масштабування | Обмеження | Приклад впливу на практику |
|---|---|---|---|
| Нейронні мережі | Значні вимоги до обчислювальних ресурсів | Низька швидкість навчання | Потреба в розподілених обчислювальних системах |
| Метод опорних векторів | Збільшення часу навчання | Високі вимоги до пам'яті | Труднощі при обробці великих наборів даних |
| Випадкові ліси | Ускладнене паралельне обчислення | Лінійне зростання часу навчання | Обмежена масштабованість для масивних даних |
| Лінійна регресія | Низька точність на великих обсягах | Обмежена складність моделі | Використовується тільки для простих завдань |
| K-ближчих сусідів | Зростання обчислювальних витрат | Зниження швидкості зі збільшенням обсягу даних | Низька ефективність для великих наборів |

Отже, нейронні мережі, хоча й забезпечують високу точність на малих наборах даних, мають значні вимоги до обчислювальних ресурсів при масштабуванні. Для їхнього ефективного використання потрібні потужні апаратні засоби, такі як GPU або розподілені системи, що робить процес навчання дорогим і повільним. Метод опорних векторів також характеризується збільшенням часу навчання і вимогами до пам'яті при обробці великих наборів даних, що обмежує його ефективність у масштабованих задачах.

Випадкові ліси, хоча і є відносно ефективними для невеликих наборів даних, демонструють лінійне зростання часу навчання зі збільшенням обсягу даних, що обмежує їхню масштабованість у великих системах. Лінійна регресія як проста модель має обмежену точність і складність, тому її застосування обмежене тільки простими завданнями, що не потребують глибокого аналізу великих обсягів даних. Алгоритм K-ближчих сусідів демонструє значне зростання обчислювальних витрат зі збільшенням обсягу даних, що негативно впливає на його ефективність при масштабуванні.

Для наукових досліджень і реальних застосувань стає важливим пошук компромісу між точністю, швидкістю та ресурсами, щоб забезпечити оптимальне масштабування алгоритмів на

великих обсягах даних. Наприклад, гібридні підходи або використання оптимізованих обчислювальних платформ можуть дозволити розв'язати деякі з цих проблем. Таблиця 7 демонструє вплив об'єму даних на продуктивність алгоритмів.

Табл. 7. Вплив обсягу даних на продуктивність алгоритмів

| Алгоритм | Обсяг даних (млн записів) | Точність (%) | Час навчання (с) | Обчислювальні витрати (CPU/GPU) |
|---|---|---|---|---|
| Нейронні мережі | 1.0 | 92 | 300 | Високі (GPU) |
| Нейронні мережі | 5.0 | 90 | 1000 | Дуже високі (GPU) |
| Метод опорних векторів | 0.5 | 85 | 200 | Середні (CPU) |
| Метод опорних векторів | 2.0 | 83 | 700 | Високі (CPU) |
| Випадкові ліси | 1.5 | 80 | 150 | Середні (CPU) |
| Випадкові ліси | 4.0 | 78 | 400 | Високі (CPU) |

Так, нейронні мережі демонструють високу точність на мільйоні записів, але їхня продуктивність значно погіршується зі збільшенням обсягу даних до п'яти мільйонів записів, що проявляється в значному збільшенні часу навчання та витрат на GPU. Метод опорних векторів демонструє схожу динаміку, де збільшення обсягу даних зменшує точність і збільшує час навчання, роблячи його менш ефективним для великих наборів.

Ці експерименти вказують на необхідність пошуку балансів між точністю й ефективністю алгоритмів для обробки великих обсягів даних, а також на можливість оптимізації цих алгоритмів шляхом використання розподілених обчислювальних систем або гібридних моделей, які можуть комбінувати переваги різних підходів.

Оптимізація алгоритмів для роботи в умовах розподілених обчислень є актуальним завданням в епоху постійного збільшення обсягів даних і ускладнення процесів їхньої обробки. Сучасні алгоритми штучного інтелекту, що використовуються для аналізу великих даних, стикаються з проблемами ефективного використання ресурсів при розподілі обчислень між різними вузлами обчислювальної системи. Розподілені обчислення дозволяють прискорити обробку даних через паралелізацію процесів, проте алгоритми потребують спеціальних підходів до оптимізації для забезпечення ефективної роботи в такому середовищі.

Основними аспектами, що потребують оптимізації, є зменшення затримок при передачі даних між вузлами, забезпечення узгодженості результатів паралельних обчислень, а також максимізація використання доступних обчислювальних потужностей. Для цього необхідно впроваджувати інтелектуальні стратегії розподілу завдань та обчислювальних ресурсів між вузлами системи.

Наприклад, глибокі нейронні мережі можуть бути оптимізовані через розподіл обчислень між кількома графічними процесорами (GPU) або обчислювальними кластерами. Алгоритми, що використовують ансамблеве навчання, також можуть бути адаптовані для паралельного виконання, що дає змогу обробляти великі обсяги даних у різних частинах системи одночасно, зменшуючи загальний час обробки (табл. 8).

Табл. 8 Оптимізація алгоритмів штучного інтелекту для розподілених обчислень

| Алгоритм | Підхід до оптимізації в умовах розподілених обчислень | Переваги | Недоліки |
|---|---|---|---|
| Глибокі нейронні мережі | Паралельне навчання на декількох GPU або CPU | Швидкість обробки, масштабованість | Складність синхронізації між вузлами |
| Випадкові ліси | Розподілена обробка дерев рішень | Можливість ефективної паралелізації | Ресурсомісткість при роботі з великими даними |
| Градієнтний бустинг | Паралелізація обчислень між вузлами | Покращення швидкості навчання | Високі вимоги до пам'яті та передачі даних |

У випадку глибоких нейронних мереж паралельне навчання на декількох GPU дозволяє значно скоротити час, необхідний для навчання моделі на великих обсягах даних. Водночас виникає потреба в синхронізації результатів між різними вузлами, що може бути складним завданням. Алгоритми ансамблевого навчання, такі як випадкові ліси або градієнтний бустинг, дозволяють

розподілити процеси між різними частинами системи, що дає можливість паралельно виконувати незалежні обчислення для кожного дерева рішень або послідовної моделі. Однак це вимагає значних ресурсів, особливо для великих наборів даних.

На практиці такі підходи використовуються для оптимізації роботи систем штучного інтелекту в хмарних обчислювальних середовищах, де обробка великих даних потребує ефективного використання розподілених обчислень для досягнення максимальної швидкості та точності результатів.

**Висновки та перспективи подальших досліджень.** У результаті здійсненого дослідження встановлено, що ефективність алгоритмів машинного навчання для обробки великих даних залежить від кількох ключових факторів, таких як розмір і складність даних, обчислювальні ресурси та якість вхідних даних. Основними проблемами, які виникають під час роботи з великими наборами даних, є складність масштабування алгоритмів, що потребує значних обчислювальних ресурсів, а також час навчання моделей, який збільшується пропорційно обсягам даних. Зокрема, алгоритми глибокого навчання, хоча і забезпечують високу точність, потребують значних ресурсів, що робить їх менш ефективними у випадках обмежених обчислювальних потужностей.

Рекомендується застосовувати оптимізаційні підходи, такі як розподілені обчислення та паралелізація процесів, для прискорення навчання алгоритмів. Це дозволить зменшити затримки при обробці великих обсягів даних і покращити узгодженість результатів у різних обчислювальних вузлах. Крім того, для підвищення ефективності алгоритмів необхідно впроваджувати методи очищення та нормалізації даних, що дозволить підвищити точність моделей і знизити їхню чутливість до пропущених значень або шуму в даних.

Перспективи подальших досліджень полягають у розвитку нових методів оптимізації алгоритмів для роботи з великими даними в умовах хмарних та розподілених обчислень. Також необхідним є дослідження гібридних моделей, які поєднують різні алгоритми для забезпечення більш високої ефективності й точності прогнозів у складних умовах обробки даних.

**Список бібліографічного опису**

1. Данчак О., Войтюк М. Ефективні сховища даних для рішень машинного навчання. *Herald of Khmelnytskyi National University. Technical sciences.* 2024. Вип. 337. № 3(2). С. 57–63. DOI: https://doi.org/10.31891/2307-5732-2024-337-3-8
2. Нестеров В. Дослідження впливу аналітики великих даних на ефективність бізнесу в цифрову епоху. *Інформаційні технології та суспільство*. 2024. Вип. 1 (12). С. 70–76. DOI: https://doi.org/10.32689/maup.it.2024.1.1
3. Лозовська К. Аналіз використання методів машинного навчання в аналітиці показників інтернет-ресурсів. *Сталий розвиток економіки.* 2023. Вип. 2 (47). С. 65–69. DOI: https://doi.org/10.32782/2308-1988/2023-47-9
4. Ngiam K. Y., Khor W. Big data and machine learning algorithms for health-care delivery. *The Lancet Oncology.* 2019. Vol. 20. № 5. P. e262–e273. URL: https://www.thelancet.com/journals/lanonc/article/PIIS1470-2045(19)30149-4/abstract (дата звернення: 02.09.2024)
5. Koliesetty V. V., Rajput D. S. A review on the significance of machine learning for data analysis in big data. *Jordanian Journal of Computers and Information Technology*. 2020. Vol. 6. № 1. P. 155–171. URL: https://research.vit.ac.in/publication/a-review-on-the-significance-of-machine-learning-for-data/pdf/publisher-pdf-fulltext-a-review-on-the-significance-of-machine-learning-for-data.pdf (дата звернення: 02.09.2024)
6. Bilal J., Haleem F., Murad K., Muhammad I., Ihtesham U., Awais A., Shaukat A., Gwanggil J. Deep learning in big data analytics: a comparative study. *Computers & Electrical Engineering.* 2019. Vol. 75. P. 275–287. DOI: https://doi.org/10.1016/j.compeleceng.2017.12.009
7. Saranya T., Sridevi S., Deisy C., Tran Duc Chung, Ahamed Khan M. Performance analysis of machine learning algorithms in intrusion detection system: A review. *Procedia Computer Science*. 2020. Vol. 171. P. 1251–1260. DOI: https://doi.org/10.1016/j.procs.2020.04.133
8. Saranya P., Asha P. Survey on big data analytics in health care. *In: 2019 International conference on smart systems and inventive technology (ICSSIT).* (27–29 November 2019) (pp. 46–51). DOI: https://doi.org/10.1109/ICSSIT46314.2019.8987882
9. Shang J., You Z. Data analytics and machine learning for smart process manufacturing: Recent advances and perspectives in the big data era. *Engineering.* 2019. Vol. 5. № 6. P. 1010–1016. DOI: https://doi.org/10.1016/j.eng.2019.01.019
10. Adadi A. A survey on data-efficient algorithms in big data era. *Journal of Big Data.* 2021. Vol. 8. № 1. URL: https://link.springer.com/article/10.1186/S40537-021-00419-9
11. Amanullah M., Ariyaluran Habeeb R, Fariza R., Gani A, Ahmed E., Abdul Nainar A., Akim N., Imran M. Deep learning and big data technologies for IoT security. *Computer Communications*. 2020. Vol. 151. P. 495–517. DOI: https://doi.org/10.1016/j.comcom.2020.01.016
12. Faker O., Dogdu E. Intrusion detection using big data and deep learning techniques. *In: Proceedings of the 2019 ACM Southeast conference* (pp. 86–93). DOI: https://doi.org/10.1145/3299815.3314439

13. Habeeb R, Nasaruddin R., Gani A., Abaker Targio Hashem I., Ahmed E., Imran M. Real-time big data processing for anomaly detection: A survey. *International Journal of Information Management.* 2019. Vol. 45. P. 289–307. DOI: https://doi.org/10.1016/j.ijinfomgt.2018.08.006
14. Sarker I. H. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science.* 2021. Vol. 2. № 3. DOI: https://doi.org/10.1007/s42979-021-00592-x
15. Deng A. Database task processing optimization based on performance evaluation and machine learning algorithm. *Soft Computing – A Fusion of Foundations, Methodologies & Applications*. 2023. Vol. 27. № 10. DOI: https://doi.org/10.1007/s00500-023-08111-1

**References**

1. Danchak, O., & Voitiuk, M. (2024). Efficient data warehouses for machine learning solutions. *Herald of Khmelnytskyi National University. Technical sciences*, 337(3–2), 57–63. DOI: https://doi.org/10.31891/2307-5732-2024-337-3-8
2. Nesterov, V. (2024). Study of the impact of big data analytics on business efficiency in the digital era. *Informatsiyni technolohiyi i suspilstvo – Information Technology and Society*, 1(12), 70–76. DOI: https://doi.org/10.32689/maup.it.2024.1.10
3. Lozovska, K. (2023). Analysis of the use of machine learning methods in the analytics of internet resource indicators. *Stalyi rozvytok ekonomiky – Sustainable Economic Developme*nt, 2(47), 65–69. DOI: https://doi.org/10.32782/2308-1988/2023-47-9
4. Ngiam, K., & Khor, W. (2019). Big data and machine learning algorithms for health-care delivery. *The Lancet Oncology*, 20(5), e262–e273. Retrieved from https://www.thelancet.com/journals/lanonc/article/PIIS1470-2045(19)30149-4/abstract
5. Koliesetty, V., & Rajput, D. (2020). A review on the significance of machine learning for data analysis in big data. *Jordanian Journal of Computers and Information Technology*, 6(1), 155–171. Retrieved from https://research.vit.ac.in/publication/a-review-on-the-significance-of-machine-learning-for-data/pdf/publisher-pdf-fulltext-a-review-on-the-significance-of-machine-learning-for-data.pdf
6. Bilal, J., Haleem, F., Murad, K., Muhammad, I., Ihtesham, U., Awais, A., Shaukat, A., & Gwanggil, J. (2019). Deep learning in big data analytics: A comparative study. Computers & *Electrical Engineering*, 75, 275–287. DOI: https://doi.org/10.1016/j.compeleceng.2017.12.009.
7. Saranya, T., Sridevi, S., Deisy, C., Tran, D. C., & Ahamed, K. M. (2020). Performance analysis of machine learning algorithms in intrusion detection system: A review. *Procedia Computer Science*, 171, 1251–1260. DOI: https://doi.org/10.1016/j.procs.2020.04.133
8. Saranya, P., & Asha, P. (2019). Survey on big data analytics in health care. *In 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)* (27–29 November 2019) (pp. 46–51). DOI: https://doi.org/10.1109/ICSSIT46314.2019.8987882
9. Shang, J., & You, F. (2019). Data analytics and machine learning for smart process manufacturing: Recent advances and perspectives in the big data era. *Engineering*, 5(6), 1010–1016. DOI: https://doi.org/10.1016/j.eng.2019.01.019
10. Adadi, A. (2021). Survey on data-efficient algorithms in big data era. *Journal of Big Data*, 8(1). Retrieved from https://link.springer.com/article/10.1186/S40537-021-00419-9
11. Amanullah, M., Ariyaluran Habeeb, R., Fariza, R., Gani, A., Ahmed, E., Abdul Nainar, A., Akim, N., & Imran, M. (2020). Deep learning and big data technologies for IoT security. *Computer Communications*, 151, 495–517. DOI: https://doi.org/10.1016/j.comcom.2020.01.016
12. Faker, O., & Dogdu, E. (2019). Intrusion detection using big data and deep learning techniques. *In Proceedings of the 2019 ACM Southeast Conference*, 86–93. DOI: https://doi.org/10.1145/3299815.3314439.
13. Habeeb, R., Nasaruddin, R., Gani, A., Abaker Targio Hashem, I., Ahmed, E., & Imran, M. (2019). Real-time big data processing for anomaly detection: A survey. *International Journal of Information Management*, 45, 289–307. DOI: https://doi.org/10.1016/j.ijinfomgt.2018.08.006
14. Sarker, I. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3). DOI: https://doi.org/10.1007/s42979-021-00592-x
15. Deng, A. (2023). Database task processing optimization based on performance evaluation and machine learning algorithm. *Soft Computing – A Fusion of Foundations, Methodologies & Applications*, 27(10). DOI: https://doi.org/10.1007/s00500-023-08111-1