

УДК 004.89:004.41

Інженерія MLOps: метасинтез інструментів, практик та архітектур для автоматизації машинного навчання

Данило Олегович**Ганчук**

ORCID: 0009-0004-6474-3521

danilhanchuk@gmail.com

Криворізький державний педагогічний університет

Сергій Олексійович**Семеріков**професор, старший дослідник,
д-р пед. наук

ORCID: 0000-0003-0789-0272

semerikov@gmail.com

Криворізький державний педагогічний університет

Інститут цифровізації освіти НАПН України

Державний університет "Житомирська політехніка"

Криворізький національний університет

Академія когнітивних та природничих наук

Ключові слова:

MLOps;

автоматизація;

інструменти;

фреймворки;

архітектура;

розгортання моделей;

конвеєри машинного

навчання;

метасинтез.

Автоматизація повного життєвого циклу моделей машинного навчання критично важлива для їх ефективного впровадження в робочому середовищі. За останні роки з'явилися різноманітні інструменти, фреймворки та архітектури для підтримки практик Machine Learning Operations (MLOps). У цій статті представлено метасинтез оглядів для комплексного аналізу технологій, що забезпечують реалізацію MLOps. Проведено порівняння можливостей та функцій, що пропонуються популярними комерційними та відкритими MLOps-платформами. Ідентифіковано шаблони в архітектурі MLOps та філософія проектування. Розглянуто роль контейнеризації, оркестрації, управління конфігурацією та автоматизації інфраструктури в ML-конвеєрах. Також обговорюються підходи до розгортання моделей у хмарі та на периферійних пристроях. Основні отримані результати: 1) проведено метасинтез систематичних оглядів для узагальнення знань щодо практик MLOps та визначено, що MLOps є перспективним підходом для ефективного розгортання моделей машинного навчання, який вимагає подальших досліджень; 2) проаналізовано зв'язки між принципами, процесами та практиками MLOps та запропоновано схему взаємозв'язків між ключовими принципами, етапами розробки і впровадження моделей та основними практиками MLOps; 3) визначено найбільш ефективні практики MLOps для розгортання моделей – безперервна інтеграція / доставка, версіонування моделей і даних, автоматизація конвеєрів машинного навчання, моніторинг продуктивності, управління експериментами, життєвим циклом, безпека та конфіденційність даних, пояснюваність моделей, управління якістю даних, конфігурацією, стратегії розгортання, автоматизація інфраструктури, співпраця, управління ризиками. Отримані результати мають теоретичну значущість в узагальненні та систематизації знань про практики MLOps і практичну значущість для впровадження та вдосконалення процесів MLOps в організаціях.

DOI: <https://doi.org/10.31558/2786-9482.2024.2.3>

Вступ

У сучасному світі машинне навчання стає все більш важливою технологією, яка знаходить застосування в різноманітних галузях, як-от фінанси, охорона здоров'я, промисловість, роздрібна торгівля тощо [1]. Незважаючи на значний прогрес у розробці алгоритмів та моделей машинного навчання, їх ефективне розгортання у виробничих середовищах залишається складним [2, 3]. Це обумовлено низкою факторів, як-от необхідність забезпечення масштабованості, відтворюваності, безпеки та надійності моделей, а також складністю інтеграції процесів розробки й експлуатації. Для вирішення цих проблем виникла методологія MLOps (Machine Learning Operations), яка має на меті застосування принципів та практик DevOps (Development & Operations) до процесів розробки й розгортання моделей машинного навчання [4, 5]. MLOps охоплює широкий спектр практик, як-от автоматизація конвеєрів машинного навчання, версіонування даних та моделей, моніторинг продуктивності моделей, управління експериментами тощо [6, 7]. Дослідження показують, що застосування практик MLOps дає змогу суттєво підвищити ефективність та надійність розгортання моделей машинного навчання у виробничих середовищах [8, 9].

Водночас, незважаючи на значний інтерес до теми MLOps з боку як науковців, так і практиків, все ще існують певні прогалини та невирішені проблеми. Зокрема, відсутні загальноприйняті стандарти за кращими практиками впровадження MLOps, недостатньо досліджені питання інтеграції MLOps з DataOps (Data Operations), ModelOps (Model Operations), AIOps (Artificial intelligence for IT operations) та з іншими підходами), а також існує потреба в розробці нових інструментів та платформ для автоматизації процесів MLOps [5, 10, 11].

Ця робота спрямована на вирішення актуальної проблеми визначення та аналізу практик MLOps, необхідних для ефективного розгортання моделей машинного навчання. Підставою для виконання роботи є необхідність систематизації та узагальнення знань щодо практик MLOps, а також потреба в розробці рекомендацій щодо їх впровадження в організаціях для підвищення ефективності та надійності розгортання моделей машинного навчання у виробничих середовищах.

Відповідно до мети визначено такі основні завдання дослідження:

- 1) виконати метасинтез систематичних оглядів для узагальнення знань щодо практик MLOps, необхідних для ефективного розгортання моделей машинного навчання;
- 2) проаналізувати зв'язки між принципами, процесами та практиками MLOps;
- 3) виявити найбільш ефективні практики MLOps для розгортання моделей машинного навчання.

Метасинтез практик MLOps

Основні поняття дослідження

DevOps набуває все більшого поширення, і компанії застосовують його методи в різних галузях [6]. У цьому контексті MLOps автоматизують робочі процеси машинного навчання,

як-от конвеєри, застосовуючи практики DevOps, а саме CI / CD (Continuous Integration / Continuous Deployment) для проєктів машинного навчання [6, 8, 12].

Згідно з [12], ключові практики MLOps можуть бути реалізовані за допомогою GitHub Actions і CML (Continuous Machine Learning). Хоча деякі робочі процеси автоматизують завдання ML за допомогою GitHub Actions і CML, наскрізні конвеєри MLOps виробничого рівня у проаналізованих проєктах із відкритим кодом на GitHub зустрічаються рідко. Практики більше зосереджені на звітності та метриках, ніж на перенавчанні чи розгортанні.

Методика дослідження

“Систематичні огляди ... можуть забезпечити узагальнення стану знань у певній галузі, на основі яких можна визначити пріоритети майбутніх досліджень; вони можуть відповісти на питання, на які в іншому випадку не можуть дати відповіді окремі дослідження; вони можуть виявити проблеми в первинних дослідженнях, які повинні бути виправлені в майбутніх дослідженнях; і вони можуть породжувати або оцінювати теорії про те, як або чому відбуваються ті чи інші явища” [13]. Основною метою систематичного огляду є сприяння прийняттю обґрунтованих рішень – рішень, які ґрунтуються на доведених фактах [13]. Основна відмінність між систематичним оглядом і оглядом літератури полягає в тому, яку роль відіграє ідея. Огляд літератури може бути керованим ідеєю: у цьому випадку всі джерела можуть бути відібрані для підтвердження певної ідеї. У систематичному огляді, попри наявність певної провідної ідеї, джерела відбираються за процедурою, метою якої є не підтвердження ідеї, а перевірка гіпотез та надання відповіді на дослідницькі питання. Результатом використання систематичного огляду як наукового методу дослідження є отримання нового знання.

23 лютого 2024 р. ми здійснили такий пошуковий запит до бази даних Scopus:

TITLE ((systematic OR review OR survey) AND mlops)

Виявлено 5 документів (табл. 1), з яких 3 [5, 6, 8] належать до систематичних оглядів. Роботи [14] та [15] не є систематичними оглядами, проте відповідно до [16] їх зміст взято до уваги під час метасинтезу для об'єднання та тематичного аналізу результатів, отриманих у систематичних оглядах.

Метасинтез виконаємо за схемою з [17]:

- 1) визначення предмета дослідження – практики MLOps для ефективного розгортання моделей;
- 2) визначення релевантних джерел – систематичні літературні огляди [5, 6, 8] та огляд продуктів і постачальників MLOps [15];
- 3) ретельне вивчення з метою визначення спільного для аналізованих робіт часового проміжку, спільного та відмінного у меті, дослідницьких питаннях, джерелах, критеріях включення, виключення та якості, визначеннях MLOps та етапів MLOps;
- 4) визначення зв'язку між роботами через виявлення та групування ключових тем;
- 5) взаємна трансляція результатів різних робіт через визначення спільної термінології, пояснення суперечностей у результатах та узагальнення результатів із різних робіт;
- 6) синтез результатів;
- 7) оприлюднення метасинтезу.

Таблиця 1. Результати пошуку систематичних оглядів у Scopus

Джерело	Зміст огляду
[6]	Здійснено “багатоголосий” огляд літератури (multivocal literature review) – різновид систематичного огляду, який використовує як “білі” (статті, розділи книг тощо), так і “сірі” джерела (дописи в блогах, технічні документи, відео тощо). Метою авторів було <i>визначити інструменти</i> створення конвеєрів MLOps та проаналізувати їх основні характеристики та особливості. Автори дослідили функціональність 13 інструментів MLOps та показали, що більшість із них підтримують однакові функції, але застосовують різні підходи, які можуть надавати різні переваги залежно від вимог користувача
[8]	Здійснено систематичний огляд 30 статей з метою визначення практик, стандартів, ролей, моделей зрілості, викликів та інструментів MLOps. Результати дослідження дали змогу зробити висновок, що MLOps все ще перебуває на початковій стадії
[14]	Здійснено огляд застосувань MLOps у проєктах Data Science. За сучасних підходів MLOps наголос робиться на розробці та розгортанні моделі, тоді як організаційним аспектам (бізнес-розуміння, оцінювання) приділяється недостатня увага. Оскільки успіх проєкту з Data Science залежить не лише від технічних питань, автори пропонують у майбутніх дослідженнях продовжувати розвивати сферу MLOps шляхом подолання розриву між бізнес-цілями організації та способом, яким ці цілі представлені та моделюються за допомогою відповідних концепцій
[15]	Представлено огляд продуктів і постачальників MLOps, щоб надати аналітикам даних можливість проактивно налаштувати відповідну інфраструктуру машинного навчання. Встановлено, що як частина цифрової стратегії, машинне навчання стало поширеним інструментарієм і можливістю для багатьох компаній. Однак MLOps-аспекти часто залишаються в проєктах поза увагою доти, поки вони не будуть встановлені і не почнуть виконуватися в бізнес-середовищі
[5]	Проаналізовано відкриті питання, можливості й тенденції, з якими стикаються організації під час впровадження MLOps та AIOps, відповідні фреймворки й архітектури, а також сфери їх використання. Систематичний огляд 93 досліджень надав можливість виявити: 1) для успішної реалізації проєктів зі штучного інтелекту потрібні спільна культура та комбінація навичок програмної інженерії, науки про дані та DevOps; 2) для підтримки життєвого циклу MLOps / AIOps корисні контейнеризація, версіювання даних і моделей, FaaS (Function-as-a-Service) та безсерверні архітектури; 3) для перенавчання і повторного впровадження компонентів важливий моніторинг середовища; 4) AIOps використовуються переважно у складних середовищах, як-от технології 5G і 6G, тоді як MLOps більш поширені у традиційних промислових середовищах

Ретельне вивчення та визначення зв'язку між роботами

Розподіл оглядів за роками

Дві роботи [6, 8] належать до 2022 р., одна [5] – до 2023 р. Джерела, що аналізуються у [6], обмежені 2020 р., у [8] – 2021 р., а у [5] – 2023 р. До того ж робота [5] згадує роботу [8] як попередню.

Цілі оглядів

Спільні аспекти цілей розглянутих оглядів:

- 1) усі огляди [5, 6, 8] спрямовані на дослідження та узагальнення знань щодо методології MLOps, її практик, інструментів та викликів;
- 2) огляди [6, 8] мають на меті виявлення та аналіз інструментів MLOps, які використовуються для автоматизації процесів розробки та розгортання моделей машинного навчання;

- 3) огляди [5, 8] прагнуть надати розуміння щодо загального стану впровадження практик MLOps в індустрії та в академічній сфері.

Відмінні аспекти цілей оглядів:

- 1) огляд [6] більше фокусується на виявленні та аналізі функціональних можливостей інструментів MLOps для створення конвеєрів машинного навчання;
- 2) огляд [8] приділяє увагу ширшому колу аспектів MLOps, як-от практики, ролі, моделі зрілості, виклики, на додачу до інструментів;
- 3) огляд [5], окрім методології MLOps, також розглядає споріднену концепцію AIOps; більше уваги приділяється висвітленню можливостей, викликів та майбутніх трендів в обох областях.

Незважаючи на певні відмінності у фокусі та широті охоплення, всі розглянуті огляди об'єднує спільна мета – дослідити та узагальнити знання щодо методології MLOps, її практичного застосування, інструментів, викликів та стану впровадження для сприяння подальшому розвитку цієї області.

Дослідницькі питання оглядів

Спільні аспекти дослідницьких питань:

- 1) усі огляди [5, 6, 8] містять питання щодо інструментів та платформ, які використовуються для впровадження практик MLOps, автоматизації процесів розробки, розгортання та моніторингу моделей машинного навчання;
- 2) огляди [5, 8] включають питання щодо викликів та відкритих проблем, із якими стикаються організації під час впровадження MLOps;
- 3) огляди [5, 8] розглядають питання щодо можливостей та майбутніх трендів в області MLOps.

Відмінні аспекти дослідницьких питань:

- 1) огляд [6] містить більш специфічні питання щодо функціональних можливостей та особливостей інструментів MLOps для створення конвеєрів машинного навчання;
- 2) огляд [8] включає питання щодо ролей та обов'язків спеціалістів, залучених до впровадження MLOps, а також моделей зрілості для оцінки рівня автоматизації процесів розгортання моделей;
- 3) огляд [5], окрім MLOps, також розглядає питання, специфічні для методології AIOps, та приділяє увагу поточним і майбутнім сферам застосування цих підходів.

Дослідницькі питання розглянутих оглядів охоплюють широкий спектр аспектів MLOps, від інструментів та платформ до викликів, можливостей та сфер застосування. Незважаючи на деякі відмінності у фокусі питань, всі огляди прагнуть дослідити ключові компоненти й фактори, які впливають на впровадження та розвиток MLOps-практик в організаціях.

Інформаційні джерела оглядів

Спільні аспекти інформаційних джерел:

- 1) усі огляди [5, 6, 8] використовували електронні бази даних наукових публікацій для пошуку релевантних досліджень;
- 2) огляди [5, 6] включали в пошук як академічні (рецензовані), так і неакадемічні ("сірі") джерела літератури, як-от блоги, вебсайти, відео, репозитарії коду тощо.

Відмінні аспекти інформаційних джерел:

- 1) огляд [6] використовував Google Scholar для пошуку наукових публікацій та звичайний пошук Google для “сірої” літератури;
- 2) огляд [8] обмежився пошуком лише в академічних базах даних ACM Digital Library, IEEE Xplore, ScienceDirect та SpringerLink;
- 3) огляд [5] використовував декілька баз даних (Scopus, arXiv, Springer, IEEE), але основним джерелом була база даних Scopus.

Розглянуті огляди демонструють різні підходи до вибору інформаційних джерел. Деякі дослідження [5, 6] включають як академічні, так і неакадемічні джерела для отримання більш повної картини щодо практичного застосування MLOps. Інші [8] фокусуються виключно на рецензованих наукових публікаціях. Вибір джерел може впливати на охоплення й тип знайдених досліджень, а отже, – і на результати та висновки оглядів.

Критерії включення інформаційних джерел до оглядів

Спільні аспекти критеріїв включення:

- 1) усі огляди [5, 6, 8] включали дослідження, які безпосередньо стосуються теми MLOps, її практик, інструментів та застосування;
- 2) огляди [6, 8] розглядали дослідження, які описують досвід, практики, архітектуру або реалізацію інструментів та процесів MLOps.

Відмінні аспекти критеріїв включення:

- 1) огляд [6] включав дослідження, які описують компоненти мінімального життєвого циклу MLOps або представляють досвід та думки експертів щодо MLOps;
- 2) огляд [8] додатково включав дослідження, які оцінюють зрілість процесів MLOps, розглядають ролі та обов'язки в життєвому циклі розробки моделей машинного навчання, а також визначають виклики в розробці та впровадженні рішень MLOps;
- 3) огляд [5] мав більш загальні критерії включення, розглядаючи дослідження, опубліковані з 2018 р. до 2023 р., які містять нові ідеї і тісно пов'язані з темою MLOps та AIOps.

Незважаючи на деякі відмінності, критерії включення у розглянутих оглядах переважно зосереджені на дослідженнях, які безпосередньо стосуються MLOps, описують практичний досвід, інструменти та процеси, а також розглядають різні аспекти життєвого циклу розробки моделей машинного навчання. Огляди [5, 8] мають дещо ширші критерії, включно з дослідженнями, пов'язаними з оцінюванням зрілості, ролями та викликами MLOps.

Критерії виключення інформаційних джерел з оглядів

Спільні аспекти критеріїв виключення інформаційних джерел:

- 1) огляди [6, 8] виключали дослідження, які не надають достатньо деталей щодо архітектури, реалізації або застосування інструментів та процесів MLOps;
- 2) огляди [5, 8] виключали дослідження, опубліковані не англійською мовою.

Відмінні аспекти критеріїв виключення інформаційних джерел:

- 1) огляд [6] виключав дослідження, які просувають комерційні платформи MLOps без надання деталей щодо їх реалізації або використання;

- 2) огляд [8] виключав дослідження, які стосуються лише застосування моделей ML без розгляду аспектів MLOps, а також короткі статті, пости та дослідження без доступу до повного тексту;
- 3) огляд [5] виключав дослідження з недостатньою кількістю цитувань (залежно від року публікації), а також матеріали з обмеженим доступом (за передплатою) та статті, опубліковані в недостатньо надійних джерелах.

Розглянуті огляди застосовують різні критерії виключення для відсіювання досліджень, які не відповідають вимогам. Спільним є виключення досліджень із недостатнім описом MLOps процесів та інструментів, а також неангломовних публікацій. Відмінності полягають у додаткових критеріях, як-от виключення комерційних платформ без технічних деталей [6], коротких статей та постерів [8], а також матеріалів з обмеженим доступом та низькою кількістю цитувань [5].

Критерії якості інформаційних джерел в оглядах

Спільні аспекти критеріїв якості:

- 1) огляди [5, 8] оцінювали якість досліджень на основі повноти опису методології, контексту та результатів;
- 2) огляди [5, 8] враховували наявність обґрунтованих доказів та аргументів на підтримку висновків дослідження.

Відмінні аспекти критеріїв якості:

- 1) огляд [6] використовував кількісні показники популярності (кількість зірок на Github, кількість переглядів на YouTube) для оцінювання якості та релевантності “сірої” літератури;
- 2) огляд [8] оцінював, чи представляє дослідження емпіричні результати, а не лише думки експертів, а також чи валідовані результати у належний спосіб;
- 3) огляд [5] використовував розширений набір критеріїв, включно з наявністю всебічного огляду літератури, перевіркою результатів на прикладах використання, кількістю досліджуваних питань, наявністю відкритого доступу, опублікуванням у журналах із високим імпакт-фактором та кількістю цитувань.

Розглянуті огляди застосовують різні підходи до оцінювання якості досліджень. Загальним є прагнення включати дослідження з повним описом методології та обґрунтованими результатами. Однак конкретні критерії якості різняться – від використання показників популярності для “сірої” літератури [6] до оцінки емпіричності результатів [8] та врахування бібліометричних показників, як-от імпакт-фактор журналу та кількість цитувань [5].

Взаємна трансляція результатів різних робіт та синтез результатів

Трансляція результатів – це процес інтерпретації та перенесення результатів з одного дослідження або контексту в інший. У межах метасинтезу трансляція результатів означає пошук спільних тем і висновків у різних дослідженнях та їх узагальнення.

Для взаємної трансляції результатів різних робіт, окрім систематичних оглядів [5, 6, 8], залучимо огляд продуктів та постачальників MLOps [15].

Визначення MLOps

Спільні аспекти визначень MLOps:

- 1) усі огляди [5, 6, 8, 15] розглядають MLOps як набір практик, принципів та процесів для автоматизації та управління життєвим циклом моделей машинного навчання;
- 2) огляди [5, 6] наголошують на використанні в MLOps підходів та практик з DevOps, як-от безперервна інтеграція, доставка та моніторинг;
- 3) огляди [8, 15] підкреслюють роль MLOps в операціоналізації рішень машинного навчання та передачі їх в промислову експлуатацію.

Відмінні аспекти визначень MLOps:

- 1) огляд [6] більше фокусується на технічних аспектах MLOps, як-от управління життєвим циклом моделей, автоматизація конвеєрів та моніторинг продуктивності;
- 2) огляд [8] розглядає MLOps як набір практик конкретно для операціоналізації рішень науки про дані (Data Science);
- 3) огляд [5] наголошує на використанні в MLOps принципів інженерії програмного забезпечення та машинного навчання для створення продуктів на основі моделей;
- 4) огляд [15] розглядає MLOps як окрему сферу, що фокусується на автоматизації життєвого циклу моделей машинного навчання як частини цифрової стратегії компаній.

Незважаючи на деякі відмінності в акцентах та формулюваннях, всі розглянуті огляди визначають *MLOps* як *підхід для управління, автоматизації та операціоналізації процесів розробки, розгортання та підтримки моделей машинного навчання на основі практик з інженерії програмного забезпечення та DevOps*. MLOps є ключовим компонентом для успішного впровадження рішень машинного навчання в промисловому середовищі.

Етапи робочого процесу MLOps

Спільні етапи робочого процесу MLOps:

- 1) усі огляди [5, 6, 8, 15] включають етапи збору та обробки даних, розробки та навчання моделей, а також розгортання моделей в робочому середовищі;
- 2) огляди [5, 6, 15] виділяють етапи моніторингу продуктивності та деградації моделей після розгортання як важливу частину робочого процесу MLOps;
- 3) огляди [6, 15] включають етап повторного навчання моделей на основі нових даних або за розкладом як частину життєвого циклу MLOps.

Відмінні аспекти етапів робочого процесу MLOps:

- 1) огляд [6] пропонує детальний розподіл етапів робочого процесу MLOps, включно з кроками вилучення, аналізу, очищення та трансформації даних, а також валідації моделей;
- 2) огляд [8] менше фокусується на детальних етапах, а більше на загальних функціях MLOps, як-от збір даних, трансформація, навчання та впровадження моделей;
- 3) огляд [5] групує етапи в ширші категорії, як-от управління даними, розподілене навчання, розгортання та моніторинг;
- 4) огляд [15] додатково виділяє етапи генерування прогнозів та управління моделями й даними як частину робочого процесу MLOps.

Незважаючи на різні рівні деталізації та групування, розглянуті огляди демонструють загальну узгодженість щодо основних етапів робочого процесу MLOps. Ці етапи охоплюють весь життєвий цикл моделей машинного навчання – від збору й обробки даних до

розгортання, моніторингу та повторного навчання моделей. Відмінності в представленні етапів відображають різні підходи до структурування та опису робочого процесу MLOps.

Фреймворки та архітектури, що сприяють впровадженню MLOps

Спільні фреймворки та архітектури, що сприяють впровадженню MLOps:

- 1) огляди [5, 6, 8] виділяють платформи та фреймворки з відкритим кодом, як-от MLflow, Kubeflow та TensorFlow Extended, як ключові компоненти екосистеми MLOps;
- 2) огляди [5, 6, 15] підкреслюють важливість використання хмарних обчислювальних платформ та сервісів, як-от AWS, Google Cloud та Azure, для розгортання та масштабування рішень MLOps;
- 3) огляди [5, 8] зазначають, що архітектури, які засновано на контейнеризації (наприклад, з використанням Docker) та оркестрації контейнерів (наприклад, за допомогою Kubernetes), є ключовими для забезпечення переносимості та масштабованості рішень MLOps.

Відмінні аспекти розглянутих фреймворків та архітектур:

- 1) огляд [6] додатково виділяє платформи для оркестрації конвеєрів MLOps, як-от Apache Airflow, Jenkins та Polyaxon;
- 2) огляд [8] згадує специфічні фреймворки та платформи, як-от Kafka-ML та MLModelCI, які використовуються для управління життєвим циклом моделей ML;
- 3) огляд [5] розглядає ширший спектр архітектурних підходів, включно з використанням периферійних обчислень (edge computing), безсерверних обчислень (serverless) та архітектур, керованих подіями (event-driven architectures);
- 4) огляд [15] фокусується переважно на пропрієтарних платформах та рішеннях від комерційних постачальників, як-от Iguazio, Domino Data Lab, Comet та Valohai.

Отже, існує багато фреймворків і архітектурних підходів, що сприяють впровадженню MLOps – від відкритих платформ та бібліотек до комерційних рішень і хмарних сервісів. Ключовими факторами є підтримка автоматизації, масштабованості, переносимості та інтеграції з наявними системами та інструментами. Вибір фреймворків і архітектур залежить від конкретних вимог та обмежень організації, а також від рівня зрілості її процесів MLOps.

Інструменти MLOps для створення конвеєрів машинного навчання та операціоналізації моделей

Інструменти MLOps – це програмні засоби, які автоматизують та спрощують різні етапи життєвого циклу моделі машинного навчання, від підготовки даних до розгортання та моніторингу. Вони включають платформи для організації експериментів, версіонування моделей, автоматизації конвеєрів, розгортання та моніторингу.

Операціоналізація моделей – це переведення моделі машинного навчання з етапу розробки та експериментування у стан, готовий до використання у виробничому середовищі. Це включає розгортання моделі, інтеграцію в бізнес-процеси, моніторинг продуктивності та підтримку. Важливим складником цього процесу є версіонування – практика відстеження та управління змінами в артефактах (як-от код, дані, моделі) з часом. Це дає змогу зберігати історію змін, повертатися до попередніх версій та співпрацювати над артефактами. У MLOps версіонування застосовується як до коду, так і до моделей та даних.

Спільними інструментами MLOps, згаданими в розглянутих оглядах, є такі:

- 1) у [6, 8] виділяють MLflow як популярну платформу з відкритим кодом для управління життєвим циклом моделей машинного навчання, експериментами та розгортанням;
- 2) у [6, 15] як інструменти для операціоналізації моделей згадують хмарні платформи AWS SageMaker, Google Cloud AI Platform та Azure Machine Learning;
- 3) у [5, 8] зазначають, що для розгортання моделей часто використовуються інструменти контейнеризації, як-от Docker, та оркестрації, як-от Kubernetes.

Відмінними аспектами розглянутих інструментів MLOps є такі:

- 1) огляд [6] надає детальний перелік інструментів для різних етапів конвеєра MLOps, включно з платформами оркестрації (Apache Airflow, Jenkins, Kubeflow, Polyaxon, Seldon Core та ін.) та розгортання (TensorFlow Extended);
- 2) огляд [8] додатково згадує інструменти Kubeflow, Polyaxon, Comet.ml, Kafka-ML, MLModelCI для управління конвеєрами та розгортання моделей;
- 3) огляд [5] більше фокусується на загальних категоріях інструментів, як-от системи управління експериментами, версіонування даних і моделей, автоматизація інфраструктури;
- 4) огляд [15] деталізує функціональність популярних комерційних платформ MLOps, як-от Iguazio, Domino Data Lab, Comet, Valohai та ін. (табл. 2).

Таблиця 2. Популярні платформи та продукти MLOps (на основі [15])

Платформа / продукт	Постачальник	Посилання
MLflow	MLflow	https://mlflow.org/
Google Cloud AI	Google	https://cloud.google.com/products/ai
Kaggle	Kaggle	https://www.kaggle.com/
SageMaker	Amazon	https://aws.amazon.com/sagemaker/
Cloud-Native Toolkit	IBM	https://develop.cloudnativetoolkit.dev/resources/workshop/ai/
Iguazio MLOps Platform	Iguazio	https://www.iguazio.com/
Azure Machine Learning	Microsoft	https://azure.microsoft.com/en-us/products/machine-learning
Huawei Cloud ModelArts	Huawei	https://www.huaweicloud.com/intl/en-us/product/modelarts.html
SparkCognition Generative AI Suite	SparkCognition	https://www.sparkcognition.com/products/sparkcognition-generative-ai-suite
Comet	Comet	https://www.comet.com/site/
Grid.AI	Grid.AI	https://www.grid.ai/
Modzy ModelOps Platform	Modzy	https://github.com/modzy
Valohai MLOps Platform	Valohai	https://valohai.com/
HPE Ezmeral ML Ops	Hewlett Packard Enterprise	https://www.hpe.com/us/en/software/ezmeral-ml-ops.html
Domino Enterprise MLOps Platform	Domino	https://domino.ai/

Отже, існує широкий спектр інструментів MLOps для створення конвеєрів машинного навчання та операціоналізації моделей – від відкритих платформ, як-от MLflow, до

комерційних рішень від хмарних провайдерів і спеціалізованих компаній. Вибір конкретних інструментів залежить від потреб і масштабу організації, а також сумісності з існуючим стеком технологій.

Основні функції, що надаються інструментами MLOps

Спільні функції інструментів MLOps:

- 1) у [5, 6, 8] зазначають, що інструменти MLOps зазвичай надають можливості для відстеження експериментів, версіонування моделей та даних;
- 2) у [6, 8, 15] підкреслюють важливість функцій автоматизації та оркестрації робочих процесів MLOps, як-от конвеєри навчання та розгортання моделей;
- 3) у [5, 6, 15] вказують на наявність в інструментах MLOps компонентів для моніторингу продуктивності та деградації розгорнутих моделей.

Відмінні аспекти функцій інструментів MLOps:

- 1) у [6] запропонована детальна класифікація функцій у межах таких трьох категорій:
 - загальні функції, що стосуються всіх етапів конвеєра MLOps (підтримка відкритого вихідного коду, масштабованість та еластичність, розширюваність, безхмарність або підтримка хмарних середовищ, управління метаданими, безперервна інтеграція та доставка та GUI-, CLI- та API-інтерфейси користувача);
 - функції управління даними (передача даних у реальному часі, зберігання даних, аналіз, очищення і трансформація даних, моніторинг даних, управління метаданими та забезпечення доступу до даних через API);
 - функції управління моделями (підтримка різних бібліотек та фреймворків машинного навчання, відстеження експериментів та версіонування моделей, реєстр моделей, автоматична оптимізація гіперпараметрів, тестування моделей, виявлення аномалій і дрейфу моделей, моніторинг продуктивності моделей, управління метаданими моделей та розгортання моделей через API);
- 2) у [8] додатково виділено функції автоматичної оптимізації гіперпараметрів моделей та підтримки мобільності для розгортання в різних середовищах;
- 3) у [5] зазначена важливість інтеграції інструментів MLOps із наявними системами та підтримки колаборативної роботи команд;
- 4) у [15] детально описано функції комерційних платформ MLOps, як-от:
 - розробка моделей – середовище для аналізу даних, розробки функцій, навчання та експериментів з моделями;
 - операціоналізація навчання моделей – створення повторюваних конвеєрів навчання та тестування моделей;
 - безперервне навчання моделей – автоматична підтримка частоти перенавчання моделей на основі розкладу, подій або ad-hoc запитів;
 - розгортання моделей – упаковка, тестування та розгортання навчених моделей у виробничому середовищі;
 - генерування прогнозів – надання прогнозів або класифікацій у режимі реального часу або пакетної обробки;

- моніторинг продуктивності моделей – відстеження ефективності та деградації моделей, попередження про необхідність перенавчання;
- управління даними та функціями – підтримка зберігання, обробки та доступу до даних і згенерованих функцій.

Отже, інструменти MLOps надають широкий спектр функцій для підтримки життєвого циклу моделей машинного навчання з акцентом на автоматизацію, відстеження експериментів, версіонування, моніторинг та розгортання моделей. Деякі інструменти пропонують більш спеціалізовані функції, як-от оптимізація гіперпараметрів або управління даними. Вибір інструменту з відповідним набором функцій залежить від конкретних потреб і цілей організації в області MLOps.

Способи розгортання моделей машинного навчання у виробничих середовищах

Способи розгортання моделей – це різні підходи та стратегії для інтеграції навчених моделей машинного навчання в робочі середовища, де вони можуть використовуватись для отримання прогнозів чи рішень на основі реальних даних. Це може включати розгортання на локальних серверах, у хмарі, на периферійних пристроях тощо.

Спільні способи розгортання моделей машинного навчання у виробничих середовищах:

- 1) у [5, 6, 15] зазначають, що моделі часто розгортаються з використанням контейнерних технологій, як-от Docker, що забезпечує мобільність та ізоляцію моделей;
- 2) у [6, 8, 15] вказують на поширеність розгортання моделей у хмарних середовищах з використанням платформ і сервісів AWS, Google Cloud та Azure;
- 3) у [5, 6] зазначають, що моделі часто розгортаються як вебсервіси з використанням REST API або інших протоколів для забезпечення доступу до прогнозів у режимі реального часу.

Відмінні аспекти способів розгортання моделей:

- 1) у [6] додатково описано розгортання моделей з використанням платформ оркестрації Apache Airflow, Jenkins, Kubeflow, MLflow, Polyaxon та Seldon Core, Valohai для автоматичного масштабування та управління контейнерами;
- 2) у [8] зазначено, що деякі інструменти MLOps, як-от MLflow, Kubeflow та Kafka-ML, мають вбудовані можливості для полегшення розгортання моделей у різних середовищах;
- 3) у [5] розглянуто розгортання моделей не лише в хмарі, але й на периферійних пристроях з використанням спеціальних фреймворків, як-от TensorFlow Lite та Core ML;
- 4) [15] надає такий детальний опис основних етапів та особливостей розгортання моделей машинного навчання з використанням конвеєрів CI/CD та підтримкою різних середовищ на прикладі комерційних платформ MLOps:
 - створення конвеєра CI/CD для моделей – платформи MLOps, як-от SageMaker, Azure ML та Databricks, дають змогу створювати конвеєри CI/CD для автоматизації процесу розгортання моделей, які включають етапи збирання, тестування і розгортання моделей, а також відстеження артефактів та управління версіями;
 - підтримка різних середовищ розгортання – платформи MLOps зазвичай підтримують кілька середовищ розгортання, як-от середовища розробки,

тестування та виробничі середовища; моделі можуть бути розгорнуті в різних середовищах із використанням відповідних конфігурацій та політик доступу;

- процес розгортання моделей – навчені моделі упаковуються в стандартизований формат (наприклад, Docker-контейнер) разом з необхідними залежностями; модель проходить через етапи тестування та валідації, щоб переконатися у її коректності та відповідності вимогам; після успішного проходження тестів модель розгортається в цільовому середовищі, зокрема для виробничого середовища можуть застосовуватись додаткові заходи безпеки та моніторингу;
- автоматизація та оркестрація розгортання – платформи MLOps використовують інструменти автоматизації, як-от Jenkins або GitLab CI/CD, для забезпечення безперервної інтеграції та розгортання моделей, яке може бути налаштоване на автоматичний запуск за певних подій, як-от оновлення коду моделі або поява нових даних;
- моніторинг та управління розгорнутими моделями – платформи MLOps надають інструменти моніторингу продуктивності та метрик розгорнутих моделей у режимі реального часу і у разі виявлення проблем або деградації моделі платформа може автоматично ініціювати перенавчання або відкат до попередньої версії.

Отже, найбільш поширеними способами розгортання моделей машинного навчання у виробничих середовищах є використання контейнерних технологій, хмарних платформ і сервісів, а також розгортання моделей як вебсервісів. Вибір конкретного підходу залежить від вимог до латентності, масштабованості та доступності моделей, а також від наявної інфраструктури та системи інструментів в організації.

Моделі зрілості для оцінювання рівня автоматизації розгортання моделей машинного навчання

Моделі зрілості MLOps – це концептуальні рамки, які описують різні рівні або стадії зрілості процесів MLOps в організації. Вони допомагають оцінити поточний стан практик MLOps, визначити області для вдосконалення та надати орієнтири для досягнення вищих рівнів зрілості.

У [8] розглянуто такі моделі зрілості для оцінювання рівня покращення процесу розробки рішень машинного навчання:

- 1) заснована на Capability Maturity Model та методиці Six Sigma модель зрілості [18] перевіряє, чи діяльність (а) має визначені цілі, (б) послідовно реалізована, (в) задокументована, (г) автоматизована, (д) вимірюється і відстежується, і (е) постійно вдосконалюється;
- 2) за моделлю [19] організації класифікуються за трьома рівнями зрілості розробки рішень машинного навчання, а саме такі, що орієнтовані на дані, що орієнтовані на модель, що орієнтовані на конвеєр;
- 3) за моделлю [20] є 5 етапів вдосконалення практик розробки – (а) ручний процес на базі науки про дані, (б) стандартизований процес експериментально-операційної симетрії, (в) автоматизований процес робочого процесу машинного навчання, (г) інтегровані процеси розробки програмного забезпечення та робочого процесу машинного

навчання, і (д) автоматизований і повністю інтегрований процес робочого процесу CD і машинного навчання.

- 4) у [21] запропоновано адаптацію моделі Capability Maturity Model з визначенням п'яти рівнів зрілості для кожної оціненої здатності – (а) початковий, (б) повторюваний, (в) визначений, (г) керований та (д) оптимізований.

Усі систематичні огляди [5, 6, 8] вказують, що рівень автоматизації процесів MLOps є одним із ключових факторів для оцінювання зрілості організації в цій сфері. Хоча розглянуті огляди не надають вичерпного опису моделей зрілості MLOps, вони підкреслюють важливість оцінювання рівнів автоматизації процесів розробки, тестування та розгортання моделей як ключового фактора зрілості організації. Адаптація моделей зрілості розробки програмного забезпечення до специфіки MLOps може бути ефективним підходом до оцінювання та вдосконалення процесів машинного навчання в організації.

Ролі та обов'язки, визначені в діяльності з операціоналізації моделей машинного навчання

Ролі та обов'язки в MLOps – це набір функцій та сфер відповідальності різних членів команди, залучених до процесу розробки, розгортання та підтримки моделей машинного навчання. Вони можуть включати дата-сайєнтистів, інженерів даних, інженерів MLOps, менеджерів проєкту тощо.

Метою метасинтезу ролей та обов'язків, визначених у діяльності з операціоналізації моделей машинного навчання, в оглядах [5, 6, 8, 15] було визначити ключових учасників процесу MLOps та їх функції.

Спільні ролі та обов'язки:

- 1) усі огляди [5, 6, 8, 15] згадують про залучення фахівців з даних та наукових співробітників з даних, які відповідають за розробку, навчання та експерименти з моделями машинного навчання;
- 2) у [5, 6, 8, 5] виділяють роль інженерів з даних та провайдерів даних, які займаються вилученням, обробкою, перетворенням та забезпеченням якості даних для навчання моделей;
- 3) у [5, 6, 15] зазначають важливість інженерів DevOps, інженерів ML/MLOps та інженерів програмного забезпечення в операціоналізації моделей, автоматизації процесів розгортання, створенні конвеєрів та управлінні середовищами;
- 4) у [5, 6] підкреслюють роль менеджерів, керівництва та бізнес-зацікавлених сторін у визначенні вимог до моделей, прийнятті рішень та підтримці стратегії MLOps.

Відмінні аспекти ролей та обов'язків:

- 1) у [8] додатково виділено такі ролі:
 - *фахівець з предметної галузі (domain specialist)*, який має глибокі знання предметної галузі та відіграє важливу роль в отриманні даних та валідації результатів;
 - *науковець або інженер з обчислювальних наук (computational scientist / engineer)*, який має високі технічні навички для підготовки середовища для роботи моделей машинного навчання;

- *науковець або інженер з машинного навчання* (machine learning scientist / engineer), який має поглиблені знання статистики та алгоритмів машинного навчання та відповідає за проєктування нових моделей машинного навчання;
 - *провайдер* (provenance specialist), який має знання як предметної галузі, так і машинного навчання, та керує постачанням даних у життєвому циклі розробки рішень машинного навчання;
 - *менеджер* (manager), який оцінює моделі перед їх оприлюдненням;
 - *розробник додатків* (application developer), який розробляє додатки, в яких будуть реалізовано створені моделі;
 - *керівник з розгортання* (deployment lead), який оцінює компоненти інфраструктури для розгортання моделей машинного навчання у виробництво;
- 2) у [5] згадується роль експертів предметної галузі в розміченні даних у специфічних доменах.

Незважаючи на деякі відмінності в деталізації ролей, розглянуті огляди визнають необхідність залучення фахівців з різних сфер – з розробки програмного забезпечення, інженерії даних, машинного навчання, експертів предметної області та менеджменту, для успішної операціоналізації моделей машинного навчання. Тісна співпраця та комунікація між цими ролями є критичною для реалізації MLOps практик в організаціях (рис. 1).

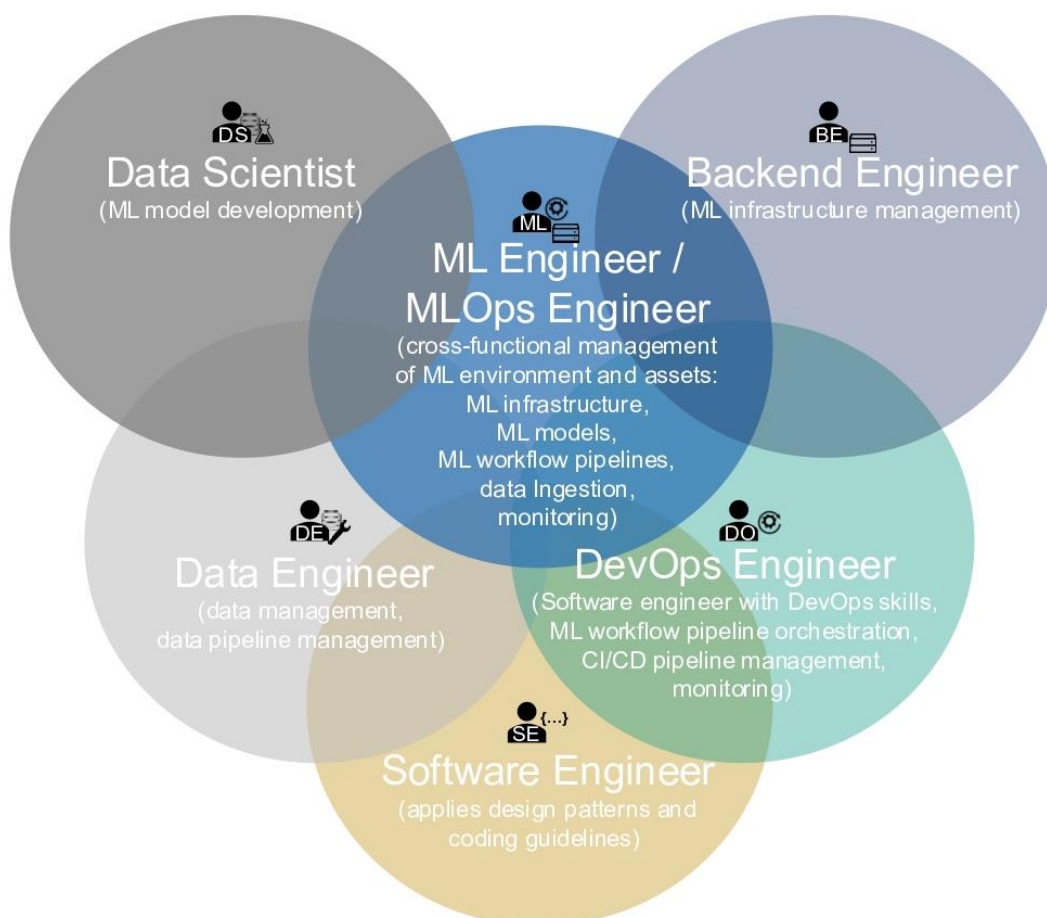


Рисунок 1. Перетин ролей та обов'язків [2]

Виклики розгортання моделей машинного навчання у виробничих середовищах

Виклики в MLOps – це труднощі та перешкоди, які виникають під час впровадження моделей машинного навчання в робочі середовища. Вони можуть стосуватися технічних аспектів (наприклад, забезпечення масштабованості та продуктивності), організаційних питань (наприклад, співпраці між командами), а також дотримання регуляторних вимог та етичних норм.

У [6] та [15] явно не наводяться конкретні виклики, однак їх можна визначити, базуючись на обговоренні MLOps та автоматизації конвеєрів машинного навчання у [6] та описі різних етапів MLOps і необхідності у відповідних інструментах у [15].

Спільні виклики:

- 1) у [5, 6, 8, 15] зазначено складності управління життєвим циклом моделей машинного навчання, зокрема версіонування, відстеження та відтворюваність моделей і даних, а також проблему забезпечення масштабованості та продуктивності моделей в реальних умовах використання з великими обсягами даних та запитів;
- 2) у [5, 8, 15] вказано на виклики моніторингу та підтримки моделей у виробничому середовищі, включно з виявленням дрейфу даних та деградації продуктивності моделей;
- 3) у [6, 8] розглянуто виклики інтеграції розробки програмного забезпечення із конвеєром машинного навчання;
- 4) у [5, 6] розглянуто виклики забезпечення безпеки та конфіденційності даних під час розгортання моделей машинного навчання;
- 5) у [5, 15] вказано, що якість, доступність, підготовка, маркування та інтеграція даних з різних джерел є значним викликом, що потребує багато часу та ресурсів; також виділено проблему інтерпретації й пояснення результатів роботи моделей для кінцевих користувачів та бізнес-стейкхолдерів.

Відмінні виклики:

- 1) у [6] підкреслена необхідність автоматизації всіх етапів конвеєра MLOps та інтеграції з наявними системами і процесами розробки програмного забезпечення;
- 2) у [8] зазначено проблему вибору та управління інфраструктурою для розгортання моделей, зокрема вибір між хмарними та локальними середовищами.
- 3) у [5] зазначено такі проблеми: а) розрив між інженерією програмного забезпечення та навичками машинного навчання – науковці з даних часто не розуміють вимоги виробничих середовищ, а розробники програмного забезпечення не мають достатніх навичок машинного навчання; б) неефективний розподіл, паралелізація та оркестрація даних і завдань машинного навчання; в) різноманіття обчислювальної інфраструктури.

Розглянуті огляди демонструють, що розгортання моделей машинного навчання у виробничих середовищах пов'язане з низкою викликів, як-от управління життєвим циклом моделей, забезпечення масштабованості та продуктивності, моніторинг і підтримка моделей у реальних умовах використання. Вирішення цих викликів вимагає комплексного підходу, що включає автоматизацію процесів MLOps, вибір відповідної інфраструктури, забезпечення безпеки та конфіденційності даних, а також ефективну комунікацію з бізнес-стейкхолдерами.

Відкриті питання, виклики та особливості MLOps

В оглядах [5, 6, 8, 15] визначено найбільш актуальні та перспективні напрями досліджень та розробок у цій галузі. У [6] та [15] безпосередньо не обговорюються відкриті проблеми, виклики та особливості MLOps, однак їх можна виокремити на основі аналізу інструментів MLOps і їх можливостей у [6] та опису компонентів та функцій платформ MLOps [15].

Спільні відкриті питання та виклики MLOps:

- 1) у [5, 6, 8, 15] вказано на потребу в розробці методів та інструментів для забезпечення інтерпретовності, відтворюваності й відповідального використання моделей машинного навчання в контексті MLOps;
- 2) у [5, 6, 15] підкреслена важливість розробки підходів до управління даними в MLOps, включно із забезпеченням якості, конфіденційності та безпеки даних;
- 3) у [6, 8] відзначена необхідність розробки та впровадження стандартів і кращих практик MLOps для забезпечення узгодженості та сумісності між різними інструментами та платформами;
- 4) у [5, 8] підкреслена важливість людського фактора в MLOps, включно із необхідністю забезпечення ефективної комунікації та співпраці між різними ролями й командами та підготовку кваліфікованих кадрів із кросфункціональними навичками програмування, обробки даних та операційної діяльності.

Особливості MLOps:

- 1) у [6] зазначено, що MLOps має враховувати специфіку процесу розробки моделей машинного навчання, який відрізняється від традиційної розробки програмного забезпечення;
- 2) у [15] розглянуто MLOps у контексті загальної цифрової стратегії організації та підкреслено необхідність узгодження практик MLOps із бізнес-цілями й потребами.

Отже, розглянуті огляди визначають низку відкритих питань та викликів в MLOps, як-от необхідність розробки стандартів і кращих практик, забезпечення інтерпретовності та відповідального використання моделей, а також ефективне управління даними. Особливості MLOps, як-от відмінність від традиційної розробки програмного забезпечення, важливість людського фактора та необхідність інтеграції знань з різних галузей, вимагають врахування під час впровадження практик MLOps в організаціях.

Можливості, майбутні тенденції та сфери застосування MLOps

У [6] перспективні напрями розвитку та потенційні області, в яких практики MLOps можуть принести значну користь, безпосередньо не обговорюються, однак їх можна визначити, базуючись на представлених інструментах MLOps та їх можливостях.

Можливості та майбутні тенденції MLOps:

- 1) у [5, 6, 8] відзначено потенціал розвитку стандартизованих платформ та інструментів MLOps, які дають змогу спростити та прискорити впровадження моделей машинного навчання в виробництво;
- 2) у [5, 8] відзначено перспективи інтеграції MLOps з іншими підходами, як-от DataOps, ModelOps та DevSecOps, для забезпечення комплексного управління життєвим циклом моделей машинного навчання;

- 3) у [8] вказано на: а) значні можливості для подальших академічних досліджень та розробок через те, що MLOps все ще знаходиться на початковій стадії; б) очікування зростання попиту на інструменти та платформи MLOps із поширенням застосування рішень штучного інтелекту; в) появу нових ролей та компетенцій, пов'язаних з MLOps;
- 4) у [5] вказано на: а) можливості застосування практик MLOps у контексті розподіленого та федеративного навчання моделей, що дасть змогу ефективно використовувати децентралізовані дані; б) залучення бізнес-підрозділів та навчання керівництва принципів MLOps; в) використання апаратних платформ FPGA та IoT для покращення продуктивності та конфіденційності.

Поточні та майбутні сфери застосування MLOps:

- 1) у [5, 6, 8] відзначено, що MLOps уже активно застосовується у фінансах, охороні здоров'я, торгівлі, маркетингу та виробництві, де моделі машинного навчання використовуються для вирішення реальних бізнес-задач;
- 2) у [5] вказано на потенціал застосування MLOps у сфері IoT та периферійних обчислень, де моделі машинного навчання можуть бути розгорнуті на пристроях з обмеженими ресурсами, а також на застосуванні MLOps для технологій 5G й 6G, та в навчальній і науковій діяльності;
- 3) у [8] відзначено перспективи використання MLOps у транспорті та логістиці.

Розглянуті огляди окреслюють низку можливостей та тенденцій розвитку MLOps, як-от створення стандартизованих платформ, застосування в контексті розподіленого навчання та інтеграція з іншими підходами управління життєвим циклом даних та моделей. Поточні та майбутні сфери застосування MLOps включають широкий спектр галузей, від фінансів та охорони здоров'я до IoT та обробки природної мови, що свідчить про значний потенціал впливу цього підходу.

Аналіз практик MLOps

Співвідношення принципів, процесів і практик MLOps

MLOps базується на наборі принципів [2] та процесів, які забезпечують ефективні практики MLOps – розроблення, розгортання та підтримку моделей машинного навчання.

Принципи MLOps визначають фундаментальні основи проєктування конвеєрів машинного навчання. Принципи є такими:

- *автоматизація* – максимальна автоматизація всіх етапів життєвого циклу моделей машинного навчання для зменшення ручних втручань та покращення ефективності;
- *відтворюваність* – забезпечення можливості відтворення результатів експериментів та процесів розгортання моделей;
- *співпраця* – налагодження ефективної співпраці та комунікації між різними командами, залученими до розробки та впровадження моделей;
- *безперервне навчання та покращення* – регулярне оновлення моделей на основі нових даних та зворотного зв'язку, постійне вдосконалення процесів MLOps;
- *керуваність даними* – забезпечення якості, безпеки та конфіденційності даних протягом усього життєвого циклу моделей.

Процеси MLOps визначають порядок дій із проєктування та реалізації конвеєрів машинного навчання. Вони є такими:

- *визначення бізнес-цілей та вимог* – узгодження цілей розробки моделей машинного навчання з бізнес-стратегією організації;
- *збір та підготовка даних* – збір, очищення, трансформація та збагачення даних для навчання моделей;
- *розробка та навчання моделей* – вибір алгоритмів, розробка архітектури моделей, навчання та валідація моделей;
- *оцінювання та тестування моделей* – оцінювання продуктивності моделей на тестових даних, проведення тестів на надійність, безпеку та відповідність вимогам;
- *розгортання моделей* – пакування моделей з необхідними залежностями, розгортання в цільових середовищах;
- *моніторинг та обслуговування моделей* – відстеження продуктивності моделей, виявлення й вирішення проблем, оновлення моделей за потреби;
- *управління життєвим циклом моделей* – координація всіх етапів розробки, розгортання та підтримки моделей, забезпечення відповідності регуляторним вимогам.

Практики MLOps визначають найбільш ефективні методи та технології реалізації конвеєрів машинного навчання. Основні практики MLOps є такими:

- *безперервна інтеграція та доставка (CI / CD)* – автоматизація процесів збирання, тестування та розгортання моделей машинного навчання;
- *версіонування моделей та даних* – відстеження змін у моделях та датасетах, забезпечення відтворюваності результатів;
- *автоматизація конвеєрів машинного навчання* – створення автоматизованих робочих процесів для збору, обробки даних, навчання та оцінки моделей;
- *моніторинг продуктивності моделей* – відстеження метрик якості моделей у виробничому середовищі, виявлення деградації продуктивності;
- *управління експериментами* – організація, відстеження та порівняння різних експериментів з моделями та гіперпараметрами;
- *розгортання моделей* – упакування моделей з необхідними залежностями, розгортання в різних середовищах (хмара, периферія тощо);
- *управління життєвим циклом моделей* – координація процесів розробки, тестування, розгортання та моніторингу моделей для найкращого врахування вимог.

Додаткові практики MLOps є такими:

- *безпека та конфіденційність даних* – забезпечення захисту даних, що використовуються для навчання моделей, відповідність регуляторним вимогам;
- *пояснюваність та інтерпретовність моделей* – використання методів та інструментів для розуміння і пояснення поведінки моделей, особливо в регульованих галузях;
- *управління якістю даних* – моніторинг та забезпечення якості даних, які використовуються для навчання й оцінювання моделей, виявлення та обробки аномалій;
- *управління конфігурацією* – версіонування та управління конфігураціями середовищ, у яких розгортаються моделі, забезпечення узгодженості між різними середовищами;

- *стратегії розгортання моделей* – вибір та реалізація стратегій розгортання;
- *автоматизація інфраструктури* – використання Infrastructure as Code для автоматизації управління інфраструктурою навчання та розгортання моделей;
- *співпраця та комунікація* – налагодження ефективної співпраці між командами науки про дані, розробки, операційної діяльності та бізнес-підрозділами;
- *управління ризиками та комплаєнс* – ідентифікація ризиків використання моделей машинного навчання, запобігання їм та задоволення регуляторних вимог.

Рис. 2 ілюструє зв'язки між ключовими принципами MLOps (блакитні прямокутники), основними процесами (зелені прямокутники) та поширеними практиками (помаранчеві прямокутники). Стрілки показують, як принципи впливають на процеси, а процеси реалізуються через конкретні практики. Наприклад, принцип автоматизації впливає на всі процеси MLOps, від визначення цілей до управління життєвим циклом моделей. Процес розробки моделей пов'язаний із практиками версіонування, автоматизації конвеєрів, управління експериментами та інтерпретовності моделей.

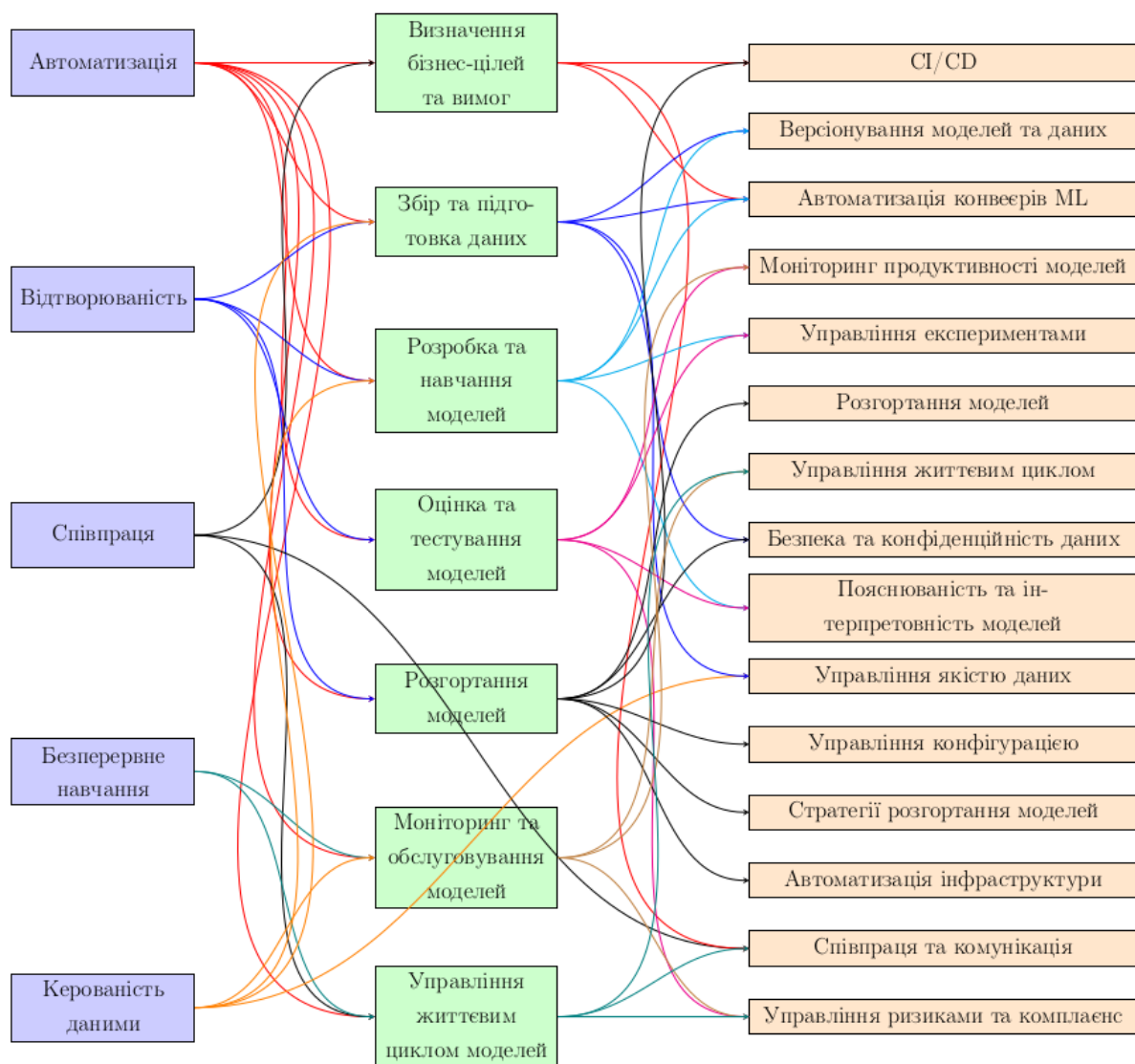


Рисунок 2. Взаємозв'язок принципів, процесів та практик MLOps

CI / CD

CI / CD – це ключовий елемент DevOps для автоматичного тестування та розгортання коду, даних і моделей у виробничому середовищі (рис. 3). У MLOps CI / CD розширюється для автоматизації процесу розробки та розгортання моделей машинного навчання, включно з етапами побудови, тестування, доставки та розгортання [2].

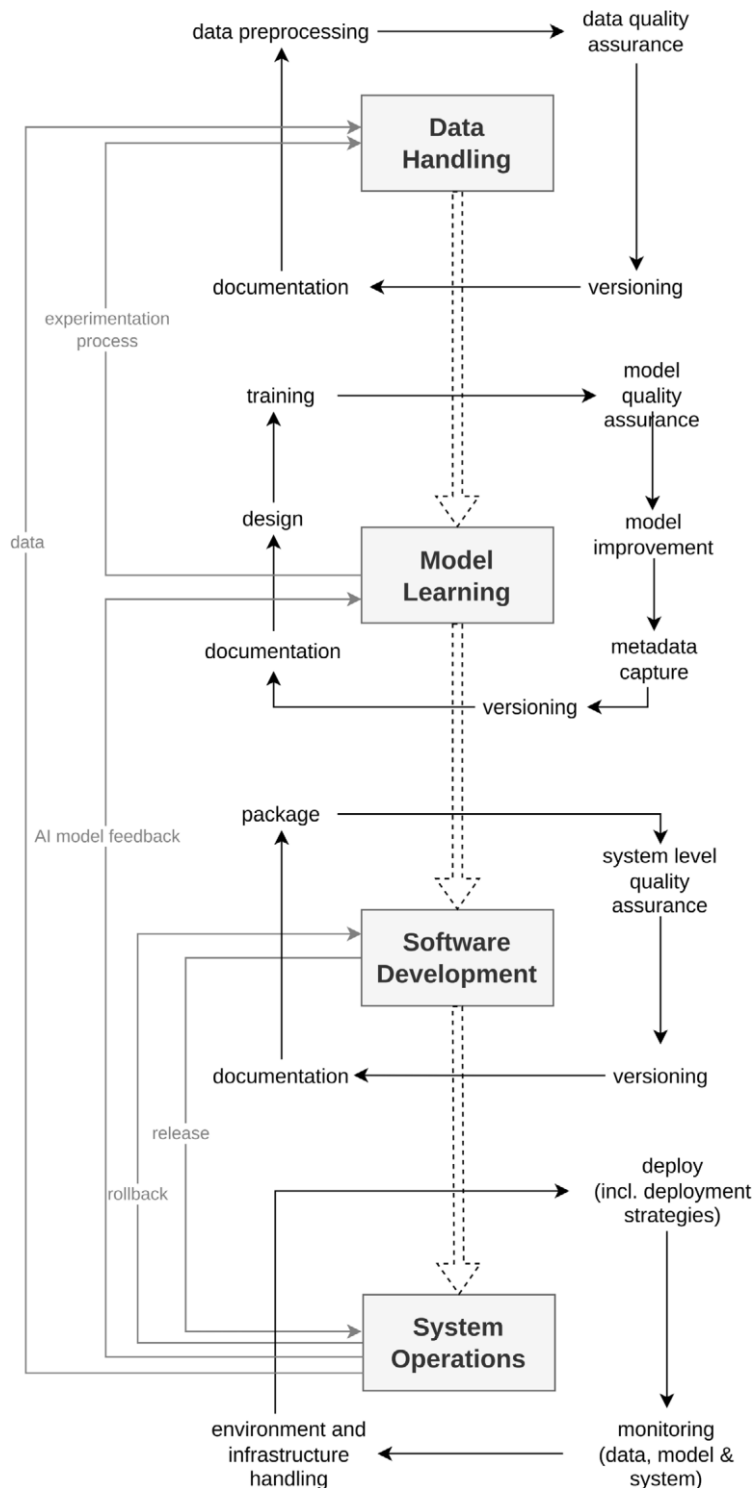


Рисунок 3. Безперервний конвеєр життєвого циклу для програм штучного інтелекту [7]

Процес CI / CD у MLOps включає в себе етапи побудови (build), тестування (test), доставки (delivery) та розгортання (deploy) [2]. Однак, на відміну від традиційного CI / CD, у MLOps також можуть бути додаткові етапи, як-от донавчання моделі. CI / CD у MLOps є частиною архітектури систем MLOps та забезпечує розробникам швидкий зворотний зв'язок у випадку успішності чи невдачі певних етапів, що підвищує загальну продуктивність [2]. Типовими тригерами запуску процесу CI / CD у MLOps на GitHub є події git push та pull_request [12]. Також можуть використовуватися події issue_comment, release та schedule. У статті [7] також досліджено потенційні основні тригери, як-от системи зворотного зв'язку та оповіщення, служба оркестрування за розкладом, традиційні оновлення репозиторію та ручні тригери. Ці тригери запускають виконання конвеєра, який складається з чотирьох етапів: обробка даних; навчання моделі; розробка програмного забезпечення; введення системи в експлуатацію. Етап обробки даних складається з повторюваного наскрізного життєвого циклу завдань, пов'язаних з даними, як-от попередня обробка, забезпечення якості, версіонування та документування. На етапі навчання моделі здійснюється проєктування моделі, її навчання, забезпечення якості, збір метаданих для вдосконалення моделі, керування версіями та документування. Після того, як конвеєр завершує навчання моделі, етап розробки програмного забезпечення готує модель до розгортання за допомогою пакування, забезпечення якості на програмному рівні та версіонування системи. На завершальному етапі введення системи в експлуатацію модель розгортається в конкретному середовищі за допомогою різних стратегій, а також здійснюється моніторинг системи [7].

Особливістю процесу CI / CD у MLOps є необхідність версіонування не лише коду, а й даних та моделей. Це дає змогу забезпечити відтворюваність та можливість відкату до попередніх версій [2]. Серед популярних інструментів для реалізації CI / CD у MLOps в [2] виділяють GitHub Actions та інструменти від хмарних провайдерів AWS CodePipeline, Azure DevOps Pipelines тощо.

На рис. 4 наведено наскрізну архітектуру та робочий процес MLOps із функціональними компонентами та ролями, задіяними на кожному етапі. Розглянемо їх детальніше.

MLOps Project Initiation. На цьому етапі бізнес-стейкхолдер (BS) аналізує проблему та визначає ціль. Науковець з даних (DS) формулює проблему машинного навчання на основі бізнес-цілі. Також визначаються необхідні дані та здійснюється їх первинний аналіз інженером з даних (DE) та науковцем з даних (DS).

Data Engineering Zone. Ця зона включає два підетапи:

Requirements for feature engineering pipeline, на якому визначаються правила для трансформації, очищення та розрахунку нових ознак;

Feature Engineering Pipeline, на якому реалізується конвеєр обробки даних, який отримує дані із різних джерел, застосовує трансформації і завантажує у сховище ознак (Feature store system).

Experimentation. На цьому етапі науковець з даних (DS) здійснює аналіз, підготовку та валідацію даних, а також навчання й валідацію моделі. Найкраща модель зберігається у Model Registry.

ML Production Zone. Ця зона включає автоматизований конвеєр (D. Automated ML Workflow Pipeline), який забезпечує підготовку даних, навчання, валідацію та реєстрацію

моделі у режимі production. Компонент Model Serving розгортає модель, а Monitoring Component забезпечує її постійний моніторинг. У разі виявлення проблем інформація передається через Feedback Loop для ініціювання повторного навчання моделі.

Окрім функціональних компонентів, на рисунку також показані різні ролі та зони їх відповідальності: Business Stakeholder (BS), Data Scientist (DS), Data Engineer (DE), DevOps Engineer (DO), ML Engineer (ML), Software Engineer (SE) та IT Solution Architect (SA).

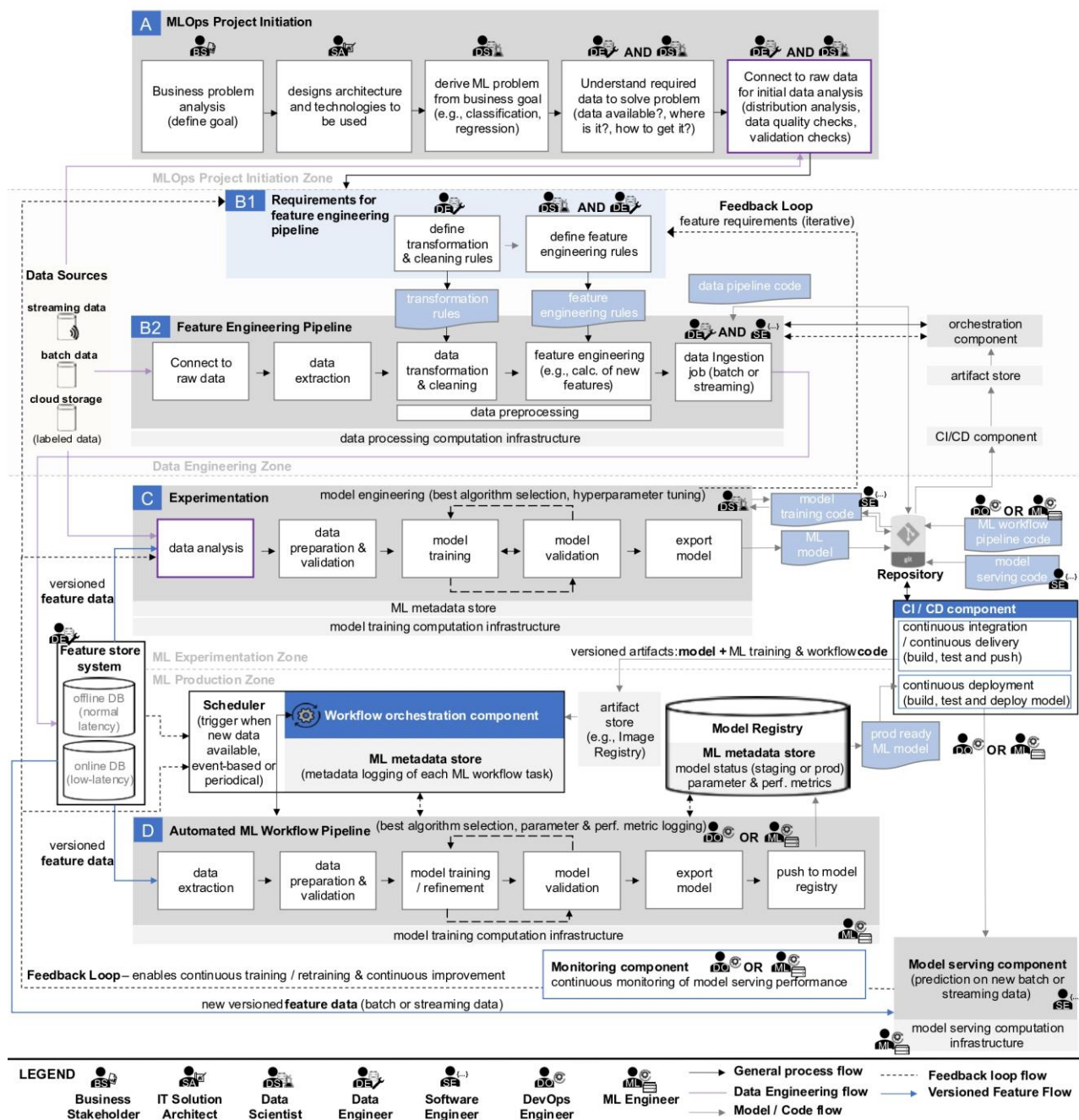


Рисунок 4. Наскрізна архітектура та робочий процес MLOps із функціональними компонентами та ролями [2]

Версіонування моделей та даних

Версіонування є однією з ключових практик MLOps, яка забезпечує відтворюваність (reproducibility) та можливість відстеження (traceability) моделей машинного навчання [2]. Версіонування охоплює дані, код та самі моделі [7]. Версіонування моделі не лише фіксує артефакти моделі версій, але й залежності моделі для відстеження або відтворення різних версій моделі, які швидко змінюються з часом [7]. Метою версіонування даних є гарантування відтворюваності моделей та відповідності регуляторним вимогам. Версіонування даних може реалізовуватись або через збереження знімків даних (data snapshots), або через посилання на оригінальний набір даних. Оскільки традиційні системи керування версіями не можуть впоратися з великими обсягами даних, використовують спеціалізовані інструменти, як-от Data Version Control [7].

У робочому процесі MLOps версіонування моделей відбувається на етапі навчання. Його метою є збереження різних версій моделей разом з метаданими для можливості відкату до попередніх версій та відтворення результатів. Умовами використання версіонування моделей є постійна еволюція моделей з часом та необхідність відстеження залежностей між моделями, даними та кодом [7].

Особливістю версіонування моделей, на відміну від традиційного версіонування коду, є необхідність відстеження більшої кількості артефактів та метаданих, а також потреба у більших обсягах пам'яті через постійний розвиток моделей [7]. Окрім самих даних, версіонуванню підлягають залежності, кроки обробки даних та видобуті ознаки (features) [7]. Для останніх часто використовують спеціальні сховища ознак (feature stores). Залежності фіксують зв'язок із пов'язаними елементами, як-от набір даних, вихідний код і конфігураційні файли. До того ж версії моделі зберігають пов'язані з ними файли журналів і результати оцінювання моделі. Це дає змогу перевірити, чи версії моделі постійно вдосконалюються впродовж безперервного життєвого циклу. Оскільки керування версіями моделей штучного інтелекту є складнішим і вимагає більшого обсягу пам'яті через безперервний розвиток, стандартні системи керування версіями, як-от Git, не можуть бути використані як репозиторії моделей. Потенційні альтернативи – це контейнерні реєстри (MLFlow, H2O, DataRobot), де зберігаються версії зображень, або репозиторії моделей, які зберігають версії моделей, включно з кодом, метаданими, результатами тестів і залежностями [7].

Автоматизація конвеєрів машинного навчання

Автоматизація конвеєрів машинного навчання є ключовою практикою MLOps, яка дає змогу спростити та прискорити розробку, тестування й розгортання моделей у робочому середовищі. Ця практика охоплює автоматизацію різних етапів конвеєра машинного навчання, включно зі збором даних, препроцесінгом, розробкою моделі, навчанням, тестуванням, валідацією та розгортанням [7].

Алгоритм роботи автоматизованого конвеєра машинного навчання (рис. 5) складається з 10 етапів. Цикл зворотного зв'язку забезпечує безперервне покращення моделі шляхом повернення до етапу навчання. За потреби моніторинг може ініціювати тригер донавчання моделі, який запускає процес спочатку. Цей автоматизований конвеєр надає можливість

ефективно керувати життєвим циклом моделі машинного навчання, від підготовки даних до розгортання та моніторингу, забезпечуючи безперервне вдосконалення моделі.

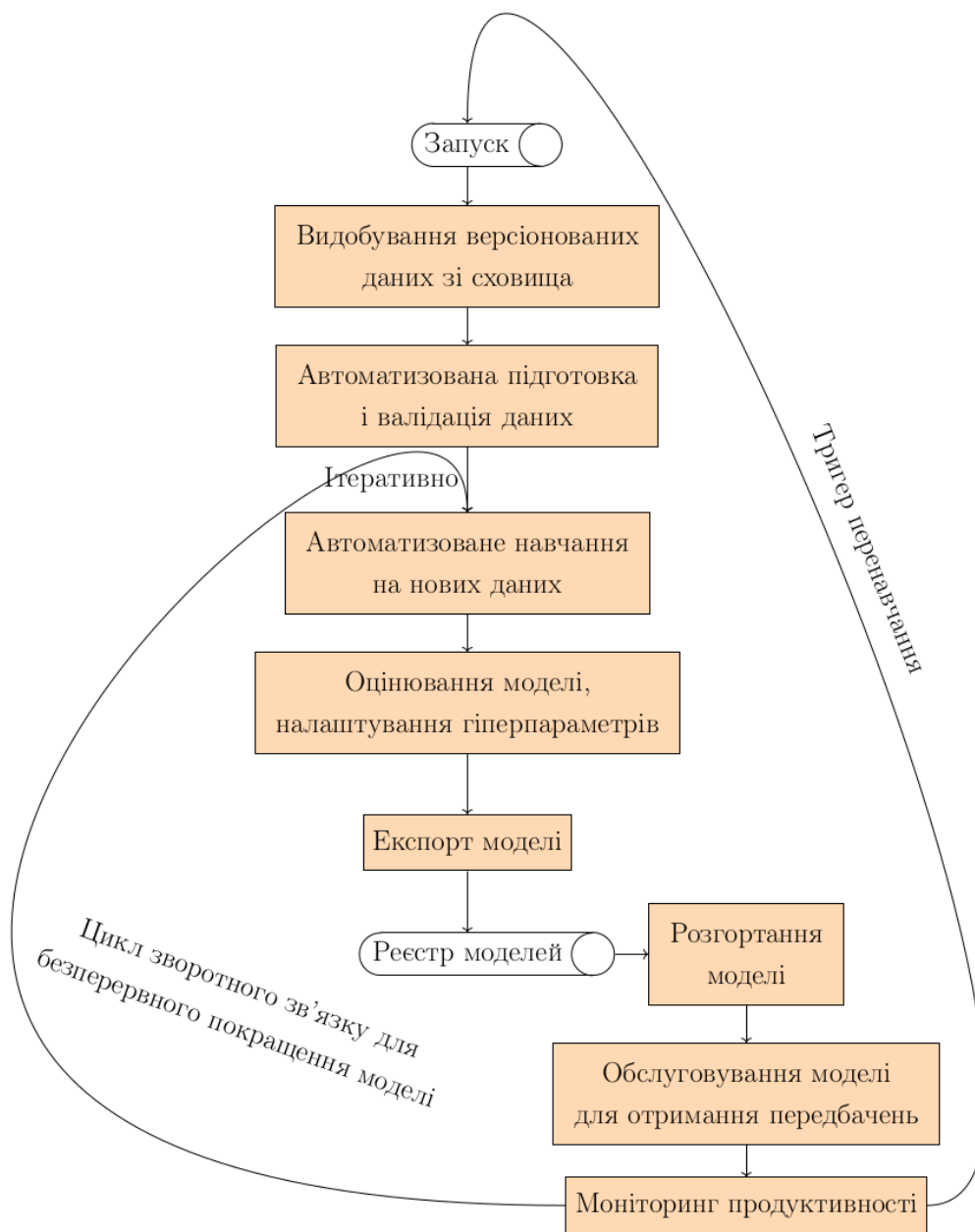


Рисунок 5. Алгоритм роботи автоматизованого конвеєра машинного навчання (за [2])

Особливістю використання автоматизації конвеєрів машинного навчання є врахування гетерогенності моделей, фреймворків та середовищ виконання. Тому кожен крок конвеєра машинного навчання має бути максимально ізольованим та повинен мати чіткі інтерфейси, наприклад, через використання контейнеризації [22]. Також критично важливим є забезпечення можливості відтворення результатів та відслідковування артефактів [23].

Способи автоматизації конвеєрів машинного навчання включають використання систем керування конвеєрами, як-от Kubeflow, TensorFlow Extended [7] або Apache Airflow [6], а

також розробку власних скриптів автоматизації з використанням Jenkins [24] або GitLab CI / CD [2]. Під час цього конвеєр розбивається на окремі кроки, кожен з яких реалізується як код або конфігурація, а потім ці кроки оркеструються та виконуються автоматично [23].

Моніторинг продуктивності моделей

Моніторинг продуктивності моделей машинного навчання є важливою практикою MLOps, яка дає змогу відстежувати роботу моделей у виробничому середовищі, виявляти проблеми та вживати заходів для підтримки їх якості. Ця практика охоплює збір метрик щодо роботи моделі, моніторинг цих метрик у реальному часі та оповіщення у разі виявлення відхилень від норми [25]. Моніторинг продуктивності відбувається на етапі експлуатації моделі, після її розгортання у виробничому середовищі. Він є частиною неперервного циклу MLOps і виконується паралельно з іншими практиками, як-от розробка, тестування та розгортання [25].

Умовами використання моніторингу є наявність розгорнутої моделі машинного навчання, яка обробляє реальні дані та генерує передбачення. До того ж повинна бути налаштована інфраструктура для збору і зберігання даних моніторингу, наприклад, база даних та інструменти візуалізації [25]. Процес моніторингу включає кілька кроків (рис. 6). Спочатку визначаються ключові метрики продуктивності моделі – точність, помилка, латентність тощо. Потім налаштовуються інструменти для збору цих метрик із системи, де розгорнута модель. Далі дані агрегуються та візуалізуються на інформаційних панелях для зручного аналізу. Нарешті, налаштовуються оповіщення, які спрацьовують за виходу метрик за допустимі межі [25].

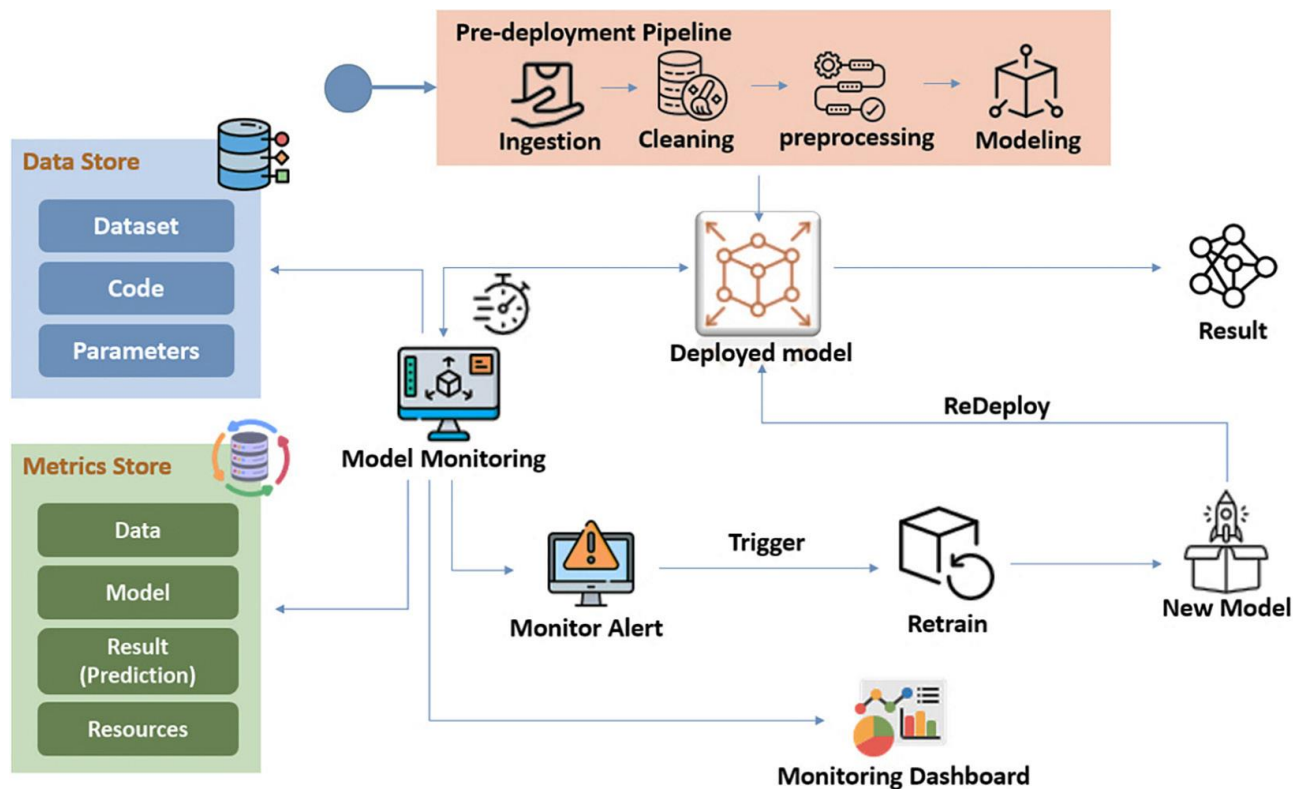


Рисунок 6. Моніторинг у MLOps [25]

Рис. 7 зображує запропоновану в [25] аналогію між моніторингом системи машинного навчання та айсбергом. Вона підкреслює, що “здоров’я системи” машинного навчання залежить не тільки від видимих елементів, як-от надані сервіси (Service), але й від прихованих особливостей, як-от дані (Data) та сама модель (Model). Верхній рівень айсберга представляє “здоров’я системи” (Ecosystem Health), що включає прогнозний дрейф (Prediction Drift) та бізнес-показники ефективності (Business KPI). Метою моніторингу є оцінювання продуктивності, надійності та стабільності цього верхнього рівня. Рівень сервісів включає метрики латентності (Latency), вартості (Cost) та загальної продуктивності системи (System Performance). Рівень даних містить характеристики якості даних (Data Quality), викиди (Outlier Value) та дрейф даних (Data Drift). Найнижчий рівень айсберга – це модель, яка характеризується точністю (Model Accuracy), дрейфом концепції (Concept Drift) та зміщенням моделі (Model Bias).

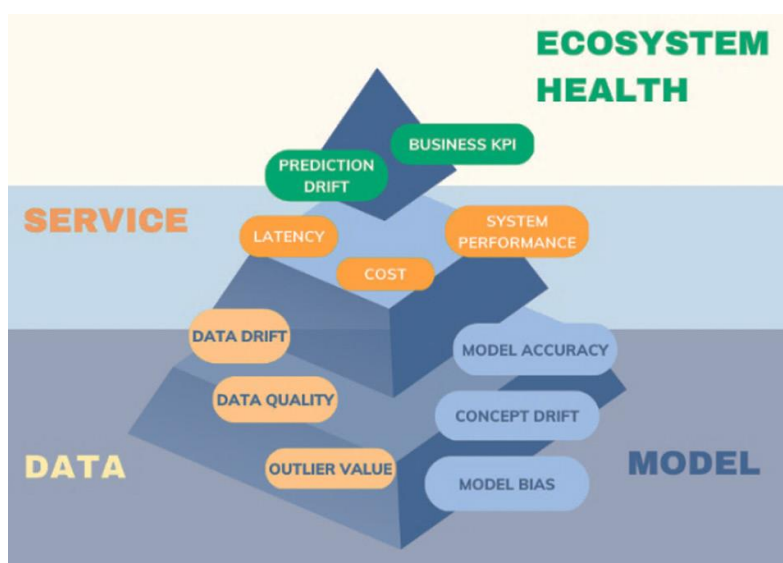


Рисунок 7. Приховані та видимі характеристики, пов’язані з моніторингом [25]

Особливістю моніторингу в контексті MLOps є необхідність відстежувати не лише традиційні метрики продуктивності програмного забезпечення, як то завантаженість процесора та пам’яті, але й специфічні для машинного навчання метрики, наприклад, точність моделювання на нових даних (рис. 8). До того ж MLOps передбачає автоматизацію процесу моніторингу та інтеграцію його в загальний конвеєр розробки й розгортання моделей [25]. Існує низка інструментів для налаштування моніторингу продуктивності моделей машинного навчання. Вони включають платформи з відкритим кодом, як-от Prometheus, Grafana, ELK stack, а також комерційні рішення від хмарних провайдерів, наприклад, AWS CloudWatch, Google Stackdriver, Azure Monitor. Ці інструменти дають змогу зручно збирати, зберігати, візуалізувати метрики та налаштовувати оповіщення [25].

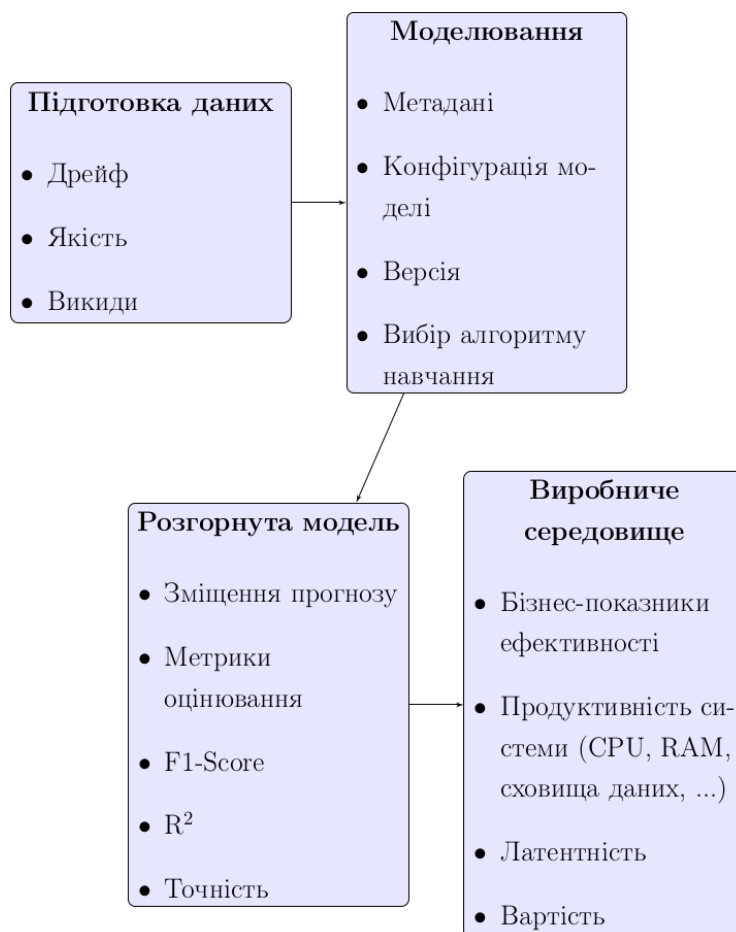


Рисунок 8. Елементи для моніторингу [25]

Управління експериментами

Управління експериментами (experiment tracking) полягає в систематичному збереженні метаданих експериментів машинного навчання [26]. Метадані експерименту можуть включати довільні скрипти для запуску експерименту, файли конфігурації середовища, інформацію про дані для навчання та оцінювання, конфігурації параметрів моделі та навчання, метрики оцінювання, вагові коефіцієнти моделі, візуалізації ефективності (наприклад, матриця сплутувань або ROC-крива) тощо [27]. Управління експериментами (також відоме як реєстрація експериментів) є частиною MLOps, яка орієнтована на підтримку ітеративної розробки моделі – частини життєвого циклу проєкту машинного навчання, де, зокрема, виконується добір гіперпараметрів для досягнення необхідного рівня продуктивності моделі. Управління експериментами тісно переплітається з іншими аспектами MLOps, як-от версіонування даних і моделей. Основною умовою застосування управління експериментами є ітеративний характер процесу розробки та навчання моделі, коли проводиться багато експериментів з різними наборами гіперпараметрів, архітектур моделей та навчальних даних (рис. 9). Це характерно для дослідницьких та прикладних проєктів із машинного навчання.

Найбільш популярними інструментами для управління експериментами є MLflow, Neptune.ai, Weights & Biases (wandb), Guild.ai, Comet.ml та TensorBoard [2]. Вони дають

змогу зберігати у репозиторії інформацію про кожен експеримент – використані дані та їх версії, параметри моделі та її архітектура, метрики ефективності, артефакти моделі тощо.



Рисунок 9. Управління експериментами у життєвому циклі MLOps (за [27])

Розгортання моделей

Розгортання моделей (model deployment) є важливою практикою MLOps, що відбувається на етапі операціоналізації (operationalization) у робочому процесі розробки та впровадження моделей машинного навчання. Ця практика полягає у безпосередньому розміщенні навченої та протестованої моделі машинного навчання у виробниче середовище, де вона може використовуватися для отримання прогнозів на реальних даних [11].

Розгортання може відбуватися кількома різними способами. Моделі, упаковані в контейнери, просто запускаються безпосередньо як окремі сервіси. Однак моделі можуть розгортатися в цільовому середовищі, яке відрізняється від того, де вони були упаковані. В цьому випадку перенесення моделі здійснюється за допомогою push- або pull-шаблонів [11]:

- під час розгортання за pull-шаблоном цільове середовище (хост-додаток, що працює, наприклад, на сервері або периферійному пристрої) періодично запитує оновлення моделі та завантажує їх, коли вони доступні;
- під час розгортання за push-шаблоном цільове середовище сповіщається про доступність нової моделі головним сервером через службу обміну повідомленнями.

Розгортання моделей машинного навчання часто відбувається шляхом їх пакування у контейнери, наприклад, за допомогою Docker. Це дає змогу стандартизувати процес розгортання і гарантувати, що модель буде працювати у тому ж середовищі, що і під час розробки і тестування [11].

Існують такі способи розгортання моделей машинного навчання, які обумовлено вимогами й архітектурою системи [28]:

- модель може бути інтегрована безпосередньо у код застосунку;
- модель може бути розгорнута як окремий сервіс (мікросервіс) із REST API;
- модель може бути завантажена у спеціалізоване середовище для розгортання та масштабування моделей (наприклад, AWS SageMaker).

Розгортання моделей машинного навчання пов'язане з низкою проблем, зокрема з потребами забезпечення малої затримки та високої пропускної здатності сервісів прогнозування [11]. Для їх вирішення використовуються різні методи, як-от адаптивні черги з таймаутом для пакетного прогнозування, кешування, динамічне переключення між моделями різної точності тощо. З інструментів для розгортання моделей машинного навчання використовують системи оркестрації контейнерів (Kubernetes), сервіси хмарних провайдерів (AWS SageMaker, Azure ML), наскрізні платформи MLOps (MLflow, Kubeflow) [3, 28].

Розгортання моделей машинного навчання є критично важливою практикою MLOps, що дає змогу переводити розроблені моделі у робочий стан і використовувати їх для прогнозування. Воно відбувається на етапі операціоналізації та вимагає врахування низки факторів – від способу пакування моделі до забезпечення необхідної продуктивності сервісу прогнозування. Розгортання спирається на сучасні інструменти контейнеризації, оркестрації та автоматизації інфраструктури.

На основі узагальнення матеріалів з [11, 28] на рис. 10 представимо загальну схему розгортання моделей машинного навчання. На схемі використовуються такі позначення:

Модель ML – навчена та протестована модель машинного навчання;

Упаковка – пакування моделі у відповідний формат (наприклад, у контейнер Docker);

Реєстр моделей – реєстр упакованих моделей;

Розгортання – розгортання упакованої моделі у цільовому середовищі (хмара, периферійні пристрої);

Обслуговування – обслуговування моделлю запитів та генерування передбачень;

Моніторинг – відстежування продуктивності моделі та середовища, і за потреби – ініціація повторного навчання моделі.

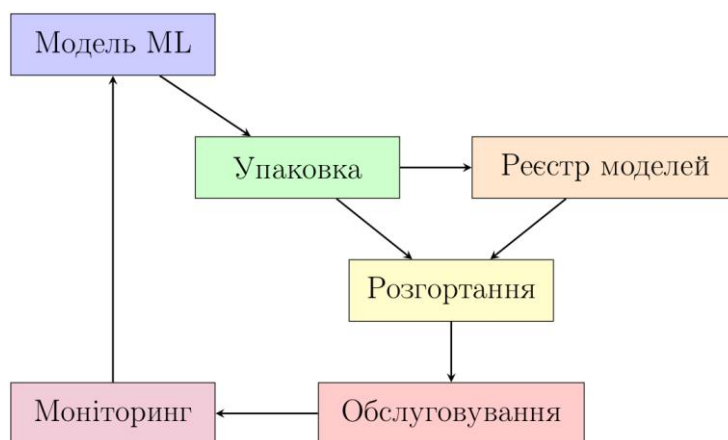


Рисунок 10. Схема розгортання моделей машинного навчання

Управління життєвим циклом

Згідно з [7], безперервне управління життєвим циклом (end-to-end lifecycle management) – це “безперервний конвеєр / потік управління, який описує (автоматичне) виконання певних задач задля забезпечення управління життєвим циклом штучного інтелекту, ... який розпочинається зі збору даних та завершується розгортанням та моніторингом моделі штучного інтелекту”. Метою управління життєвим циклом є уніфікація та стандартизація процесів, що сприяє підвищенню продуктивності розробки машинного навчання моделей та їх надійності у промисловому середовищі.

Ключові особливості управління життєвим циклом у MLOps:

- охоплює усі етапи – збір та підготовку даних, розробку моделі, її навчання, валідацію та розгортання [25];
- застосовується як на ранніх стадіях розробки моделей, так і для їх неперервної підтримки після розгортання у виробничому середовищі [5];
- передбачає версіонування даних, коду та самих моделей для відстеження змін і забезпечення відтворюваності;
- передбачає моніторинг продуктивності моделей;
- автоматизує процеси за допомогою конвеєрів [7].

На рис. 11 показано процес розробки проєкту машинного навчання (конвеєр), який складається як із лінійних, так і з ітеративних кроків. Конвеєр починається з видобутку даних, їх перевірки та підготовки, а потім здійснюється навчання моделі, оцінювання та перевірка (управління експериментами). Після цього модель розгортається у виробничому середовищі [25].

Важливим аспектом життєвого циклу є версіонування моделей та даних, яке відбувається як у конвеєрі управління експериментами, так і у виробничому конвеєрі. Останній включає всі етапи створення моделі, а не лише кінцевий результат конвеєра управління експериментами. Ітераційний характер MLOps надає можливість отримання та підтримання найкращої моделі, а поєднання моніторингу на основі ключових показників ефективності та функції попередження забезпечує проактивне втручання, гарантуючи якість і надійність розгорнутих моделей протягом усього життєвого циклу [25].

Основні інструменти, що використовуються для управління життєвим циклом у MLOps є такими [25]:

- платформи для організації workloads та pipelines в машинному навчанні, як-от MLflow та Kubeflow;
- системи версіонування даних, наприклад, Data Version Control;
- інструменти для моніторингу продуктивності моделей у виробничому середовищі, як-от Neptune.ai.

Комплексний підхід до управління життєвим циклом у MLOps реалізовано в Ease.ML – в системі управління життєвим циклом для спрощення всього процесу розробки [29]. Основна мета Ease.ML полягає у наданні систематичних рекомендацій та автоматизації на всіх етапах життєвого циклу машинного навчання, мінімізуючи зусилля користувачів.

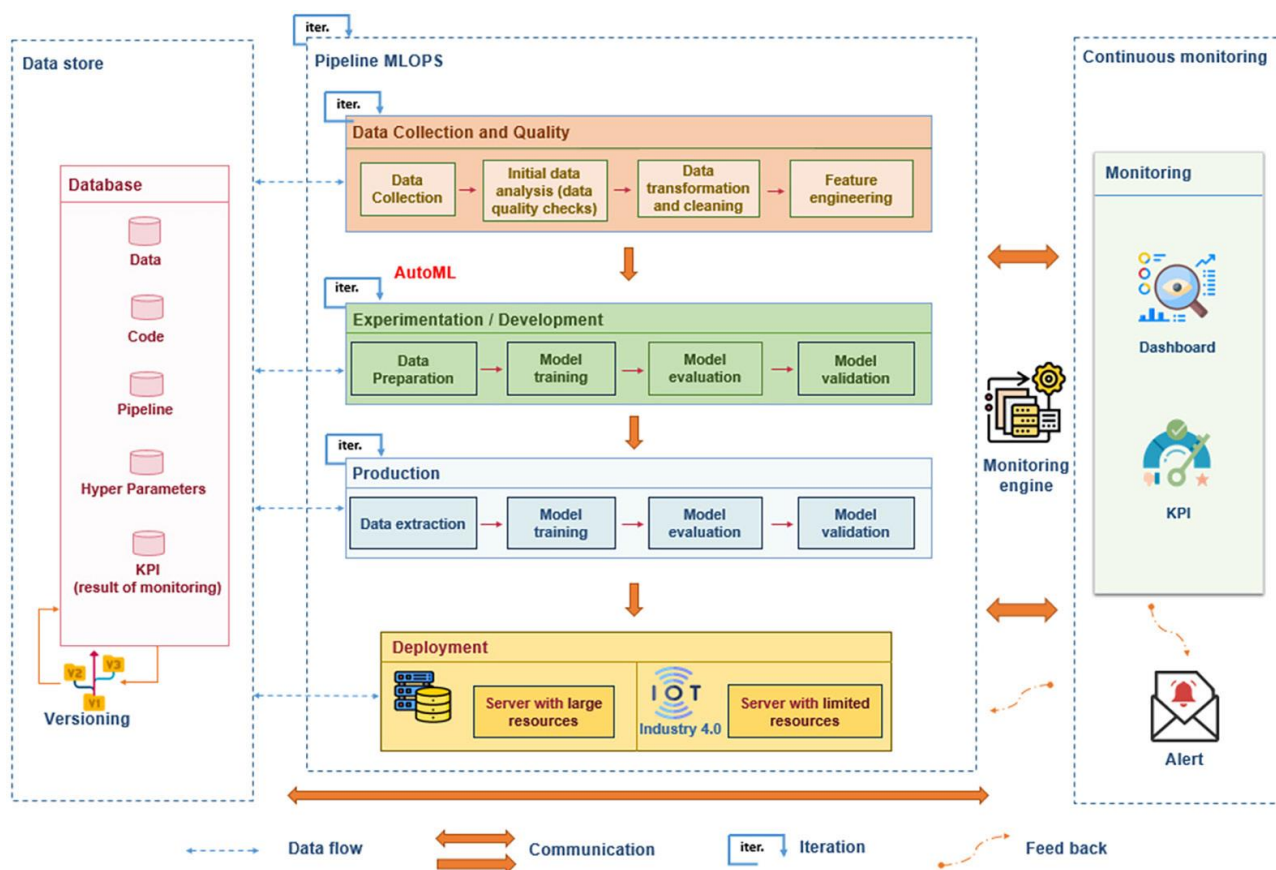


Рисунок 11. Життєвий цикл проекту машинного навчання у MLOps [25]

Ключові особливості Ease.ML є такими:

- 1) *процес із залученням людини*, що включає взаємодію з користувачем у структурованій формі, забезпечуючи можливість внесення користувачем даних і прийняття рішень на критичних етапах;
- 2) *ймовірнісна модель даних*, за якою опрацьовується невизначеність у даних, що може виникати через некоректні дані, слабкий нагляд або через інші причини;
- 3) *інтерактивне середовище* з блокнотами Jupyter, яке надає змогу користувачам виконувати маніпуляції з даними, запускати операції машинного навчання та візуалізувати результати в інтегрованому середовищі;
- 4) *графи лінійності* для відстежування взаємодії та операції користувачів під час усього робочого процесу машинного навчання, що сприяє забезпеченню відтворюваності;
- 5) *автоматичне налаштування якості та рекомендації* для покращення моделей на основі виявлених помилок.

Загальний процес Ease.ML складається з восьми етапів в межах трьох таких підпроцесів:

- Day 0: Pre-ML Subprocess
 1. *Формулювання проблеми* – чітко визначити проблему та цілі машинного навчання.
 2. *Техніко-економічне обґрунтування* – оцінити доцільність використання машинного навчання для вирішення сформульованої проблеми за наявних даних та ресурсів.

- Day 1: AutoML Subprocess
 3. *Підготовка даних* – очищення, попереднє опрацювання та доповнення даних, щоб зробити їх придатними для створення моделей машинного навчання.
 4. *Навчання моделей* – за підготовленими даними навчити моделі з використанням AutoML для автоматизації вибору та налаштування моделей.
 5. *Оцінювання навчених моделей* на валідаційних наборах даних для визначення відповідності критеріям.
 6. *Вибір найкращої моделі* та її розгортання у виробничому середовищі.
- Day 2: Post-ML Subprocess
 7. *Безперервна інтеграція та доставка* – інтеграція моделі у виробниче середовище та налаштування CI / CD конвеєрів для управління оновленнями моделі та моніторингом продуктивності.
 8. *Підтримка моделі* для адаптації до нових даних або умов через постійний моніторинг моделі у виробничому середовищі та виявлення необхідності оновлення або перенавчання моделі.

Безпека та конфіденційність даних

Безпека та конфіденційність даних – це практика захисту інформації та даних, що використовуються в процесах машинного навчання, від несанкціонованого доступу, зловживання та витоку [30]. Вона спрямована на забезпечення цілісності, доступності та конфіденційності даних на всіх етапах життєвого циклу MLOps.

Безпека та конфіденційність даних повинна враховуватися на всіх етапах MLOps, починаючи з визначення проблеми і закінчуючи моніторингом розгорнутої системи. Особливо критичним є забезпечення безпеки на етапах управління даними, розробки та розгортання моделі, оскільки саме тут дані найбільш вразливі [31]. Практика безпеки та конфіденційності даних є обов'язковою у випадках, коли система машинного навчання оперує чутливими даними (персональними, фінансовими, медичними тощо) або розгортається у критично важливих середовищах (охорона здоров'я, автомобільна індустрія, промисловість). Однак навіть для менш чутливих застосувань належний рівень безпеки має бути дотриманий відповідно до нормативних вимог та очікувань користувачів.

На відміну від традиційної розробки програмного забезпечення, у контексті MLOps з'являються нові вектори атак та вразливості, специфічні для машинного навчання (рис. 12). Зокрема, моделі машинного навчання вразливі до атак, як-от отруєння даних (data poisoning), інверсія моделі (model inversion) та атаки з використанням змагальних прикладів (adversarial examples) [30]. Це вимагає застосування спеціалізованих стратегій захисту.

Для дотримання безпеки та конфіденційності даних у MLOps використовуються як стандартні інструменти безпеки (шифрування, автентифікація, журналювання тощо), так і спеціалізовані рішення, орієнтовані на машинне навчання. Прикладами останніх є бібліотеки для безпечного агрегування федеративного навчання, інструменти для тестування моделей на вразливості, фреймворки для конфіденційного навчання на основі гомоморфного шифрування або захищених анклавів [31]. Безпека та конфіденційність даних забезпечується шляхом впровадження різних механізмів контролю та захисту на кожному етапі MLOps. Це

включає, зокрема, шифрування та анонімізацію даних, автентифікацію та контроль доступу, перевірку цілісності, моніторинг активності, реагування на інциденти, а також регулярні перевірки безпеки та тестування на проникнення [30].

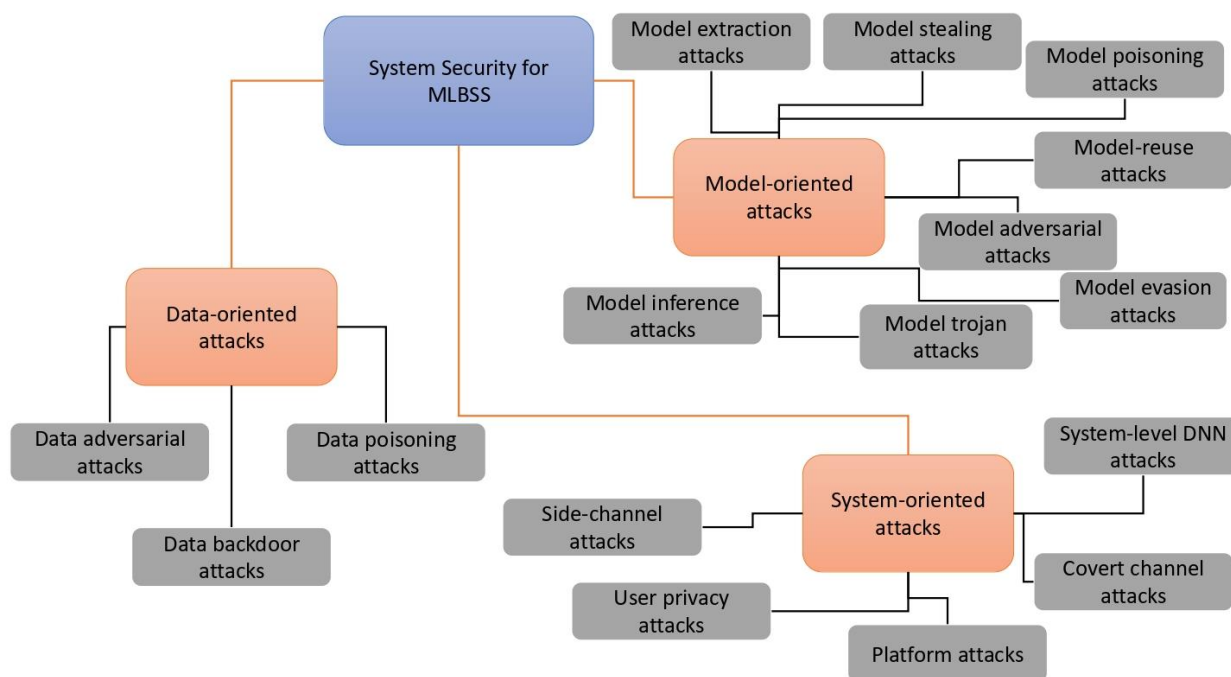


Рисунок 12. Класифікація атак на системи MLOps [30]

На рис. 13 узагальнено подані основні компоненти та практики безпеки у MLOps.

Суттєві компоненти MLOps:

- визначення проблеми (Problem definition) включає розуміння проблеми та вимог до системи;
- управління даними (Data management) охоплює збір, розмітку та верифікацію даних;
- розробка та розгортання моделі (Model construction and deployment) включає вибір, побудову, оптимізацію та оцінювання моделі, а також її розгортання у цільовому середовищі;
- підтримка системи (System maintenance) передбачає моніторинг функціонування розгорнутої системи.

Практики безпеки у MLOps:

- на етапі визначення проблеми – оцінка ризиків (Risk Assessment) та моделювання загроз (Threat Modeling);
- на етапі управління даними – схема потоків даних (Data flow diagram), методологія STRIDE для класифікації загроз, концептуальне моделювання (Conceptual modelling), перевірка даних (Data validation), аналіз якості даних (Data linter) та верифікація даних (Data verification);

- на етапі розробки та розгортання моделі – ідентифікація критеріїв достатності тестування (Test adequacy identification), тестування на змагальних прикладах (Testing against adversarial input), методи “розмиття” (Fuzzing techniques);
- на етапі підтримки системи – засоби моніторингу моделей машинного навчання (MLDEMON, SelfChecker), статичний аналіз (Static Analysis), дослідницькі атаки (Exploratory Attacks), атаки ухилення (Evasion Attacks), атаки отруєння даних (Data Poisoning Attacks), ручне тестування (Manual Testing) та динамічний аналіз (Dynamic Analysis).

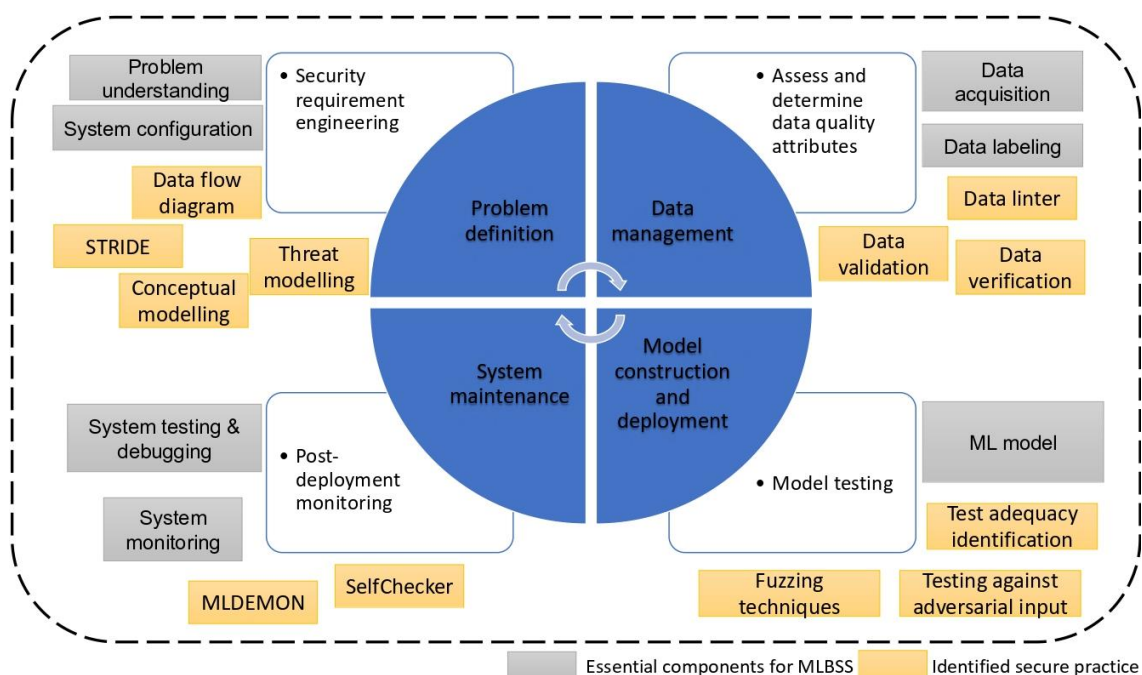


Рисунок 13. Основні компоненти та практики безпеки у MLOps [30]

Отже, практики безпеки у MLOps інтегруються в усі основні етапи життєвого циклу розробки та включають як загальні методи оцінювання та тестування безпеки (моделювання загроз, статичний / динамічний аналіз), так і спеціалізовані підходи, які орієнтовано на особливості систем машинного навчання – тестування на змагальних прикладах, виявлення отруєння даних тощо.

Пояснюваність та інтерпретовність моделей

Пояснюваність та інтерпретовність (Explainability / interpretability) моделей є важливою практикою MLOps для забезпечення прозорості та довіри до моделей машинного навчання. Пояснюваність стосується здатності пояснити та зрозуміти процеси прийняття рішень моделлю машинного навчання [10]. Це особливо важливо для рішень зі значними наслідками, наприклад, у воєнних операціях чи у правоохоронній діяльності [9]. Пояснюваність проєкту машинного навчання дає змогу користувачам довіряти прогнозу. Користувач може перевірити, які фактори вплинули на певні прогнози, що створює додатковий рівень підзвітності. Терміни *пояснюваність* і *інтерпретовність* часто використовуються як

взаємозамінні, однак для MLOPs пояснюваність – це більше, ніж інтерпретовність із погляду важливості, повноти і правильності прогнозів чи класифікацій [4] (рис. 14).

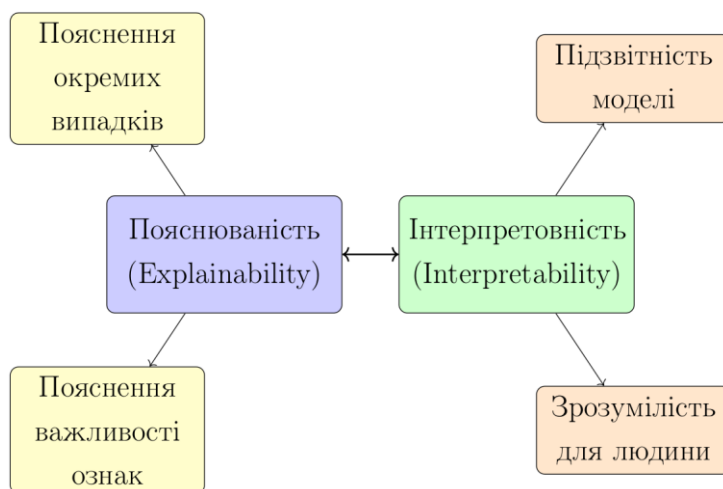


Рисунок 14. Зв'язок пояснюваності та інтерпретовності у MLOps

Пояснюваний штучний інтелект (Explainable Artificial Intelligence, XAI) – це дослідницький напрям, який сприяє прийняттю пояснених рішень [4]. Пояснюваність можна визначити як ступінь, до якого людина може зрозуміти причину ухвалення рішення [32]. Система машинного навчання є пояснюваною, коли легко ідентифікуються причинно-наслідкові зв'язки між входами та виходами системи. Чим більш пояснюваною є модель, тим фахівцям-практикам зрозуміліші внутрішні бізнес-процедури, які відбуваються під час прийняття рішень моделлю. Внутрішню логіку або процеси, що лежать в основі пояснюваної моделі, не обов'язково повністю може розуміти людина, але пояснення від моделі дають змогу користувачеві зміцнити довіру до її прогнозів та висновків [4].

Для досягнення пояснюваності використовуються такі способи та інструменти:

- методи на основі атрибутції (Integrated Gradients, Saliency Maps) або пертурбації (SHAP) [33], які пояснюють рішення моделі, призначаючи високі оцінки найвпливовішим входним параметрам;
- включення механізму уваги (attention mechanism) у моделі, що дає змогу зосередитись на найбільш релевантних станах мережі та входах [33];
- використання у процесі навчання комбінованого сигналу винагороди (reward), який включає не тільки цільові показники, але й метрики інтерпретовності [33].

Управління якістю даних

Управління якістю даних є важливою практикою у робочому процесі MLOps. У книзі [9] управління якістю даних розглядається в контексті моніторингу та перевірок даних у виробничому середовищі, щоб забезпечити відповідність даних реальності. Згідно зі статтею [7], етап обробки даних включає повний життєвий цикл роботи з даними, зокрема попередню обробку, забезпечення якості, версіонування та документування.

У життєвому циклі MLOps, заснованому на даних (рис. 15), управління якістю даних виконується на таких етапах:

- 1) збір даних – створюють та отримують з різних джерел;
- 2) оцінювання якості даних та їх очищення – зібрані дані проходять ретельну перевірку за точністю, повнотою, узгодженістю, своєчасністю та іншими метриками, виявляються та усуваються проблеми з якістю, які обумовлено некоректними значеннями, пропусками, дублікатами та шумом;
- 3) доповнення та розмітка даних – очищені дані збагачуються додатковою інформацією та розмічаються відповідно до цільової задачі, також контролюється якість розмітки даних, щоб уникнути помилок та неточностей;
- 4) аналіз якості даних – розмічені дані аналізуються на предмет якості, перевіряється їх репрезентативність, збалансованість класів, наявність викидів та аномалій і за потреби вносяться корективи;
- 5) навчання моделі з контролем якості – на основі підготовлених даних відбувається навчання моделей з контролем за метриками якості моделей та даних, щоб гарантувати стабільність і надійність результатів;
- 6) розгортання моделі з забезпеченням якості – модель розгортається у виробничому середовищі лише після ретельного тестування на тестових даних належної якості; також здійснюються заходи для підтримки якості даних у виробничому середовищі;
- 7) моніторинг якості даних і моделі – розгорнута модель та дані, що надходять, постійно перевіряються на якість – відстежуються метрики якості, наявність аномалій у даних, зміщення розподілів, і за потреби модель донавчається на нових даних належної якості.

Вчасне виявлення і усунення дефектів у даних допомагає запобігти марнуванню обчислювальних ресурсів на неякісних даних [7]. Для гарантування якості даних упродовж усього життєвого циклу їх використання у MLOps необхідно застосовувати такі підходи до валідації та верифікації даних:

- оцінювання якості даних за повнотою, унікальністю, цілісністю, валідністю, точністю та своєчасністю, щоб встановити їх придатність для досягнення бізнес-цілей [26];
- валідація даних у одиночних та перехресних пакетах шляхом порівняння характеристик даних, а також перевірка наявності зсувів даних [7];
- автоматичні модульні тести даних на основі схеми для виявлення помилок у даних та запобігання їх проникненню на етап навчання моделі [7];
- версіонування даних і пов'язаних артефактів (процедур обробки, метаданих тощо) для відслідковуваності та відтворюваності результатів і дотримання регуляторних вимог [7];
- документування даних у обсязі, достатньому для забезпечення верифікованості моделей [7].

У роботі [26] представлено техніки оцінювання якості великих даних, які допомагають виявити набори даних, що можуть спричинити проблеми і зайві витрати.

Для реалізації практик управління якістю даних можна використовувати фреймворки типу TensorFlow Extended, в яких є компоненти для валідації даних, як-от SchemaGen та ExampleValidator [7].

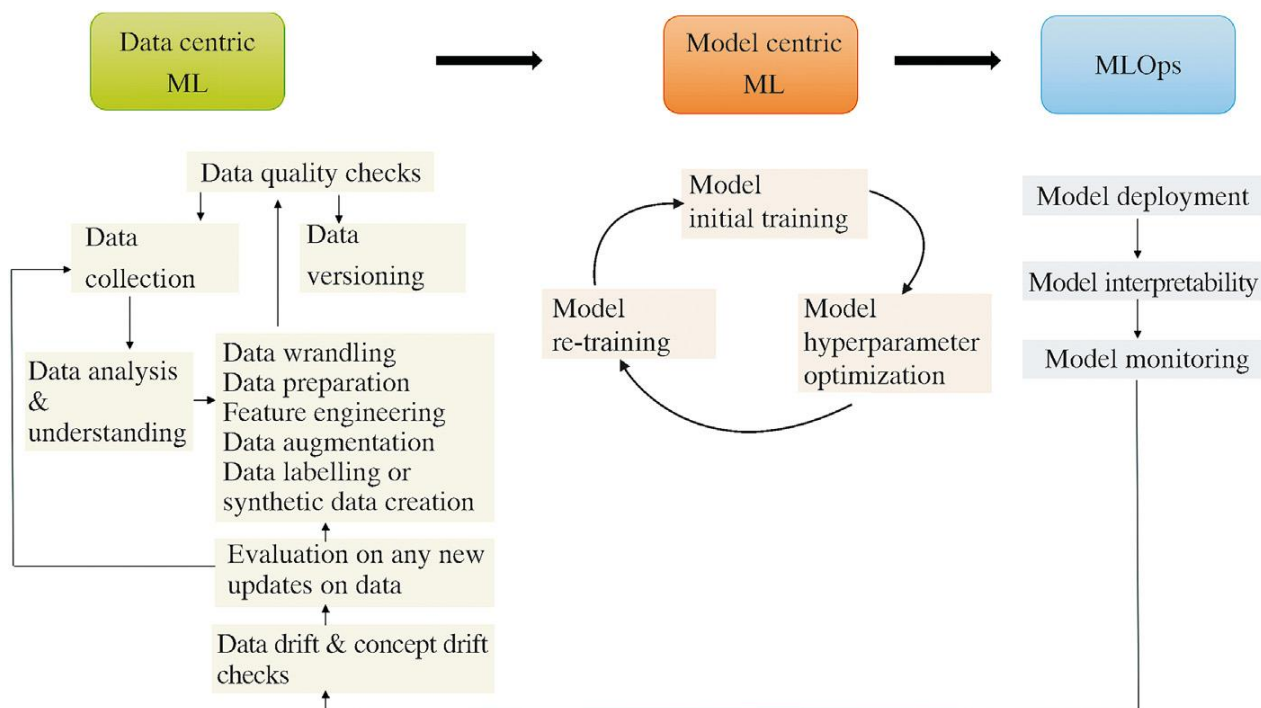


Рисунок 15. Життєвий цикл MLOps, заснований на даних [26]

Управління конфігурацією

Конфігураційні файли надають можливість перемістити всі жорстко закодовані змінні в спеціальні місця, які можуть бути розділені або організовані на розсуд розробника [34]. Це сприяє створенню більш надійного програмного забезпечення.

У [34] запропоновано шаблон, що підтримує 2 типи файлів конфігурації – файл `config.py`, що містить конфігурацію шаблону та може включати додаткові ресурси, як-от бази даних та електронні таблиці, та JSON-файли для збереження специфічних змінних програми, як-от порогові значення чи параметри. Файл `config.py` доступний через інструкції імпорту, а JSON-файли викликаються з диску під час виконання [34].

У [35] розглянуто використання єдиної моделі даних на основі мови YANG для уніфікації опису конфігураційних даних і спрощення управління ними. У [36] зазначено, що інструменти управління конфігурацією, як-от Ansible, Puppet та Chef, можуть використовуватися для автоматизації конфігурування та забезпечення інфраструктури платформ MLOps.

Стратегії розгортання моделей

Стратегії розгортання моделей реалізуються на фінальному етапі процесу MLOps і застосовуються для поступового переходу моделі у виробниче середовище. Вони спираються на контейнеризацію і потребують стандартизації, автоматизації та інкапсуляції моделей. У [28] зазначають такі переваги використання контейнерів для розгортання моделей:

- абстрагування моделей та ізоляція процесів шляхом запуску декількох моделей в окремих контейнерах, що представляють їх залежності під час виконання;
- можливість створити специфічні для кожної моделі контейнери, що відповідають їх вимогам до упаковки;

- призначення на різні процесори – CPU, Mobile GPU тощо;
- можливість виділення окремого контейнера для полегшення функцій постобробки;
- для швидкого розгортання можна використовувати репозиторій моделей та контейнерне сховище;
- забезпечує засоби стандартизованого розгортання;
- майже усі конвейси неперервної розробки полегшують розгортання моделей у вигляді контейнерів;
- контейнери можна адаптувати до конкретних архітектур периферійних пристроїв;
- контейнери Docker є усталеним стандартом у галузі;
- простота відкату в разі збоїв.

У [37] детально описано цикл розробки та стратегію розгортання моделей на прикладі застосунку DeepClean. Нова версія моделі спочатку розгортається як версія розробника, проходить валідацію в умовах, подібних до виробничих, і тільки після цього замінює поточну модель у виробничому середовищі. Застосовується сервісна архітектура з NVIDIA Triton Inference Server, що підтримує одночасне розміщення версії розробника та виробничої версії моделі.

У [37] виділено 2 основні підходи до розгортання моделей. За традиційного сценарію кожен користувач керує власними ресурсами та версіями моделей. Невідповідності в бібліотеках та залежностях, а також у версіях моделей призводять до непослідовних результатів. Зменшені обчислювальні вимоги до виведення (inference) призводять до недовикористання апаратних ресурсів, що на рис. 16 виділено салатовою заливкою на кожному вузлі. Більш складні сценарії розгортання вимагають використання декількох мереж, що посилює проблеми.

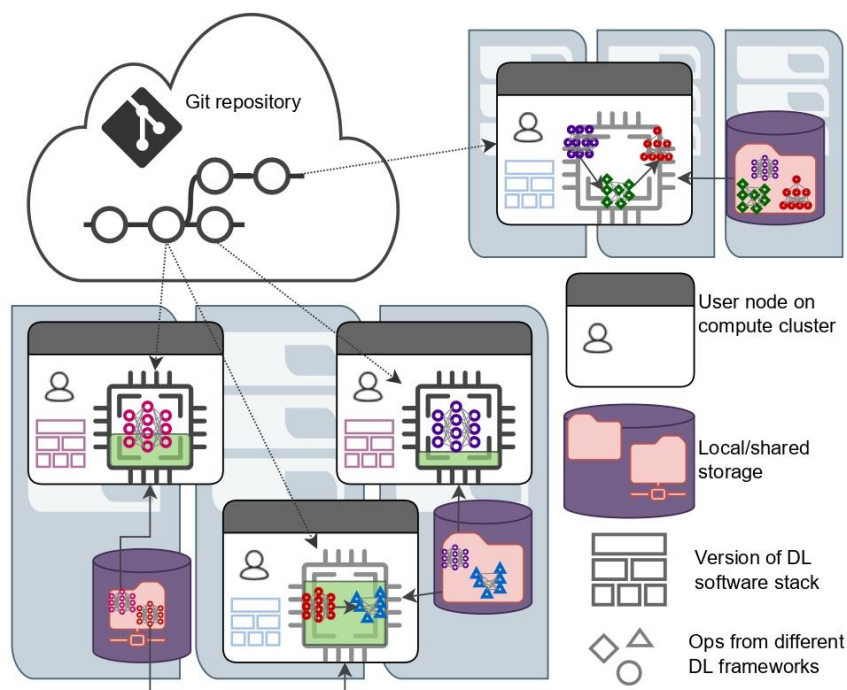


Рисунок 16. Традиційний сценарій розподіленого розгортання [37]

За підходу Inference-as-a-Service централізована служба оркеструє моделі й надає уніфіковані інтерфейси для виклику моделей (рис. 17). Централізоване сховище моделей синхронізує всіх користувачів і підтримує їх в актуальному стані. Конвеєри надсилають gRPC-запити на виведення до сервісу за допомогою стандартизованих API, які абстрагують деталі реалізації самого виконання виведення. Виведення виконується контейнеризованим сервісом, який може ефективно планувати асинхронне виконання моделей, максимально використовуючи обчислювальні можливості апаратних засобів у портативний та у масштабований способи. За такого підходу контейнеризація надає можливість створювати портативні та ізольовані середовища виконання моделей.

Вибір правильної стратегії розгортання моделі дає змогу мінімізувати операційні витрати, забезпечити узгоджену роботу моделі для всіх користувачів та полегшити моніторинг і контроль версій моделі.

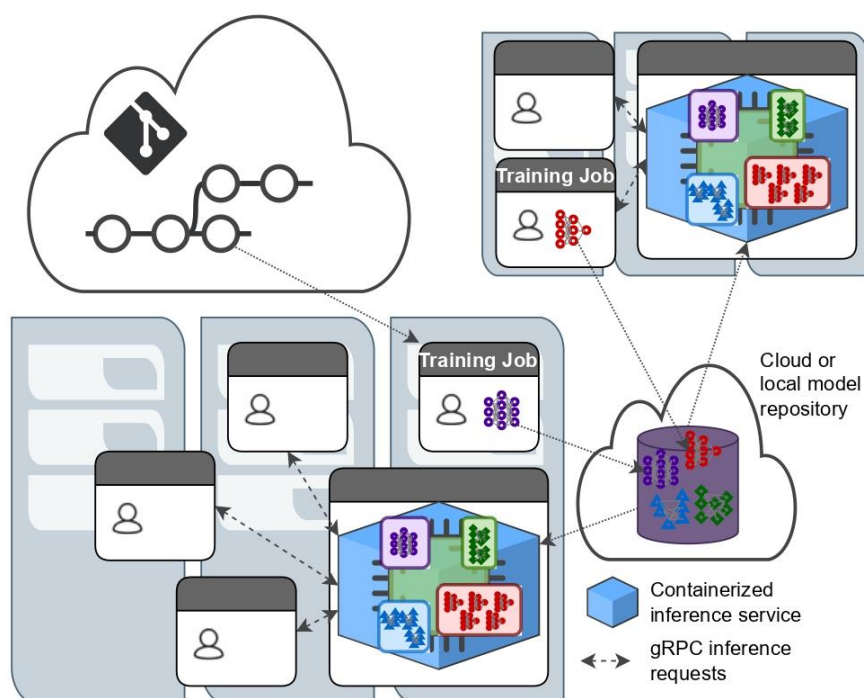


Рисунок 17. Розгортання за сценарієм Inference-as-a-Service [37]

Автоматизація інфраструктури

Автоматизація інфраструктури (Infrastructure as code) є важливою практикою MLOps, яка дає змогу розглядати інфраструктуру як код для її надійного та ефективного розгортання й управління [38]. Автоматизація інфраструктури стосується етапу розгортання та моніторингу у процесі MLOps (рис. 18). Вона забезпечує надійне створення необхідного оточення для розгортання моделей машинного навчання та автоматизує інфраструктурні завдання [38].



Рисунок 18. Автоматизація інфраструктури у життєвому циклі MLOps

Infrastructure as code передбачає опис конфігурації інфраструктури декларативним способом у спеціальних файлах (наприклад, у форматах YAML чи JSON) або за допомогою спеціальних мов (наприклад, Terraform) чи інструментів. Ці файли описують бажаний стан інфраструктури [38].

Опис налаштувань інфраструктури у вигляді коду дає змогу автоматизувати процес її створення, зміни та управління [38]. Це допомагає повністю відтворити оточення, швидко розгортати ресурси та уникати помилок ручного налаштування [38]. Такий підхід доцільний під час створення складних динамічних інфраструктур, коли потрібна висока повторюваність і узгодженість оточень, а також для підвищення надійності та зниження витрат часу на ручні налаштування [38]. Автоматизований підхід дає змогу тестувати інфраструктуру як код, застосовувати до неї практики з розробки програмного забезпечення – перегляд коду, версіонування тощо. Це сприяє підвищенню стабільності та безпеки і допомагає оперативно відслідковувати та усувати проблеми в інфраструктурі [38].

Для реалізації Infrastructure as code у Azure DevOps застосовуються інструменти Azure Resource Manager (ARM) Templates, Terraform, Ansible та Chef [38]. Вони дають змогу описати інфраструктуру у вигляді коду та автоматизовано створювати чи змінювати її.

Співпраця та комунікація

Співпраця та комунікація між різними стейкхолдерами є ключовою практикою MLOps для успішного впровадження проєктів машинного навчання та штучного інтелекту в організаціях. MLOps акцентує на тому, як кросфункціональні команди з аналітиків даних, системних операторів та інженерів з даних та програмного забезпечення співпрацюють через узгоджений процес [7]. Сутність цієї практики полягає у налагодженні ефективної взаємодії та обміну інформацією між різними командами, залученими до процесу розробки і впровадження моделей машинного навчання – командами науки про дані (data science), розробки (development), операційної діяльності (operations) та бізнес-підрозділами. Така практика є важливим складником етапу розробки та впровадження робочого процесу MLOps.

MLOps передбачає тісну співпрацю між командами науковців з машинного навчання, які займаються підготовкою даних та розробкою моделей, інженерами-розробниками, які відповідають за інтеграцію моделей у виробниче середовище, та операційними командами, які забезпечують розгортання та підтримку моделей [2]. Ефективна комунікація необхідна на всіх етапах життєвого циклу моделей машинного навчання, починаючи від визначення бізнес-цілей та закінчуючи моніторингом моделей у виробничому середовищі. Без налагодженої співпраці розробка може затягуватись, виникатимуть конфлікти інтересів та непорозуміння між учасниками [39].

Рис. 19 відображає основних учасників процесу MLOps та напрями їх взаємодії:

- команда Data Science готує дані та розробляє моделі машинного навчання;
- команда Development інтегрує розроблені моделі у програмні продукти;
- команда Operations розгортає та підтримує моделі у виробничому середовищі;
- команда Business визначає цілі та вимоги до рішень на базі машинного навчання, а також отримує результати від команд Data Science та Operations.

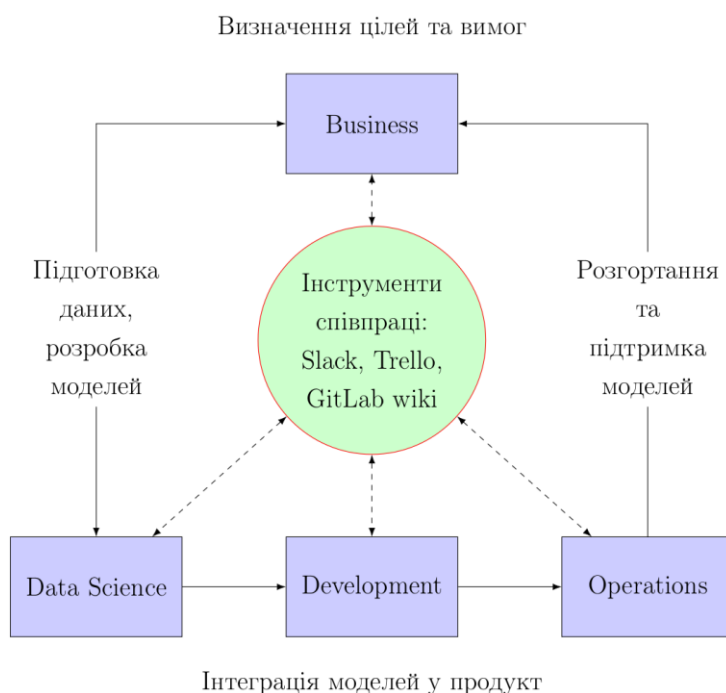


Рисунок 19. Схема співпраці та комунікації у MLOps

Для забезпечення ефективної комунікації у практиці MLOps застосовують такі підходи та інструменти [2]:

- використання інструментів для спільної роботи та обміну знаннями, як-от Slack, Trello, GitLab wiki;
- регулярні зустрічі між командами для обговорення статусу, проблем та планів;
- чітке визначення ролей та зон відповідальності учасників процесу MLOps;
- використання систем контролю версій для спільної роботи над кодом та моделями;
- автоматизація процесів CI / CD для прозорості та відтворюваності розробки.

Управління ризиками та комплаєнс

Оскільки моделі машинного навчання часто приймають важливі рішення, що впливають на людей, управління ризиками та комплаєнс є критично важливою практикою MLOps. Вона полягає у забезпеченні відповідності розроблених моделей та систем машинного навчання нормативним вимогам і стандартам (compliance) та управлінні потенційними ризиками їх розробки та експлуатації. Ця практика має наскрізний характер та проявляється на різних етапах життєвого циклу моделі машинного навчання. Управління ризиками в контексті MLOps передбачає виявлення та зменшення потенційних ризиків, пов'язаних із моделями машинного навчання, як-от упередженість даних, порушення конфіденційності, погіршення точності моделі з часом тощо.

Комплаєнс означає забезпечення відповідності моделей нормативним вимогам, наприклад, щодо захисту даних [7]. Основною умовою використання цієї практики є наявність нормативно-правових вимог чи галузевих стандартів, яким повинна відповідати система машинного навчання. Прикладами можуть бути вимоги GDPR щодо захисту персональних даних чи сертифікації у сфері охорони здоров'я [7]. Аспекти комплаєнсу мають враховуватись під час підготовки даних, навчання та валідації моделі, а також під час розгортання та моніторингу. Способи використання цієї практики включають регулярні перевірки якості даних, тестування моделей на упередженість і дискримінацію, впровадження контролю доступу та шифрування даних, документування архітектури моделі і процесу розробки, моніторинг продуктивності моделі після розгортання [7].

Провідними засобами забезпечення практики управління ризиками та комплаєнсу є системи контролю версій для відстеження змін у даних та коді моделі, засоби тестування для виявлення проблем у моделях, системи моніторингу для відстеження точності моделей у реальному часі [7]. Інтерв'ю з практиками виявили, що забезпечення комплаєнсу ML-систем у різних доменах (наприклад, в охороні здоров'я) є серйозним викликом через брак усталених методик та програмних інструментів [7]. Водночас досягнення комплаєнсу є обов'язковою умовою для отримання необхідних сертифікацій та дозволів регуляторів.

Рис. 20 відображає 3 основні етапи MLOps: роботу з даними, навчання моделі та розгортання системи (операціоналізацію). На кожному етапі наявні блоки, які позначають потенційні ризики та заходи із забезпечення відповідності. Схема ілюструє наскрізний характер практики управління ризиками та комплаєнсу в MLOps, показуючи взаємозв'язки на кожному етапі життєвого циклу моделі машинного навчання.

Рис. 21 показує взаємозв'язки між ключовими принципами, процесом розгортання та основними практиками MLOps, які застосовуються на етапі розгортання моделей машинного навчання. Принципи автоматизації і відтворюваності впливають на процес розгортання моделей, який пов'язаний з наступними практиками MLOps, як-от CI / CD, розгортання моделей, безпека та конфіденційність даних, управління конфігурацією, стратегії розгортання моделей та автоматизація інфраструктури.

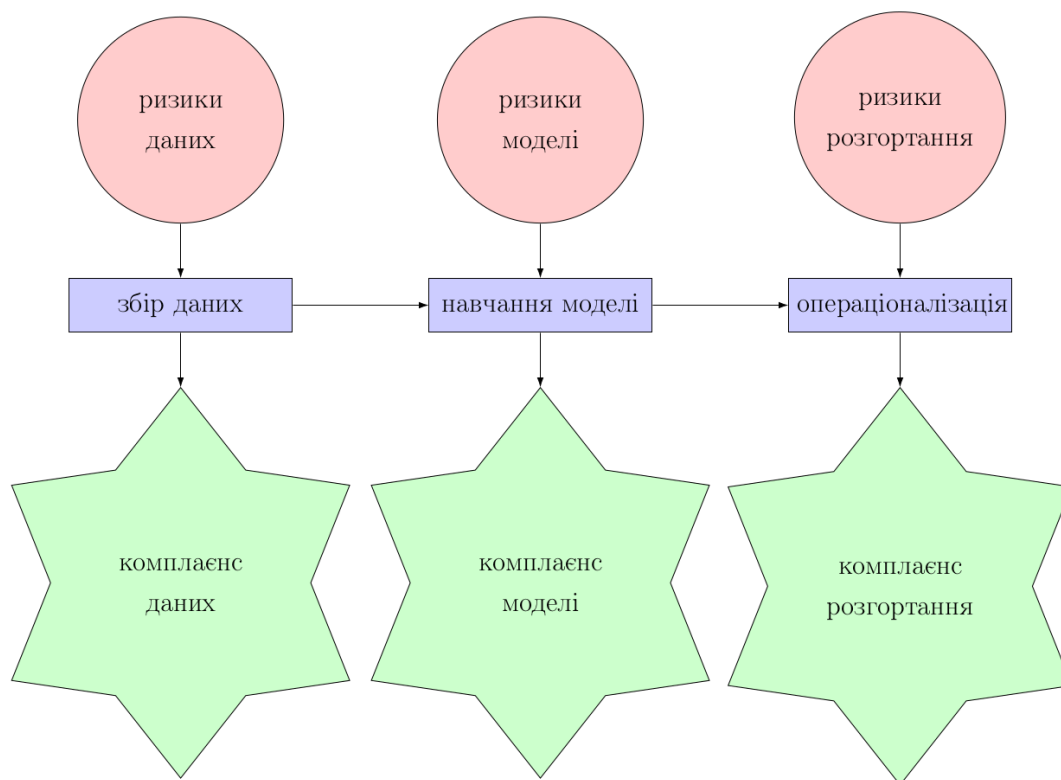


Рисунок 20. Схема практики управління ризиками та комплаєнсу у MLOps

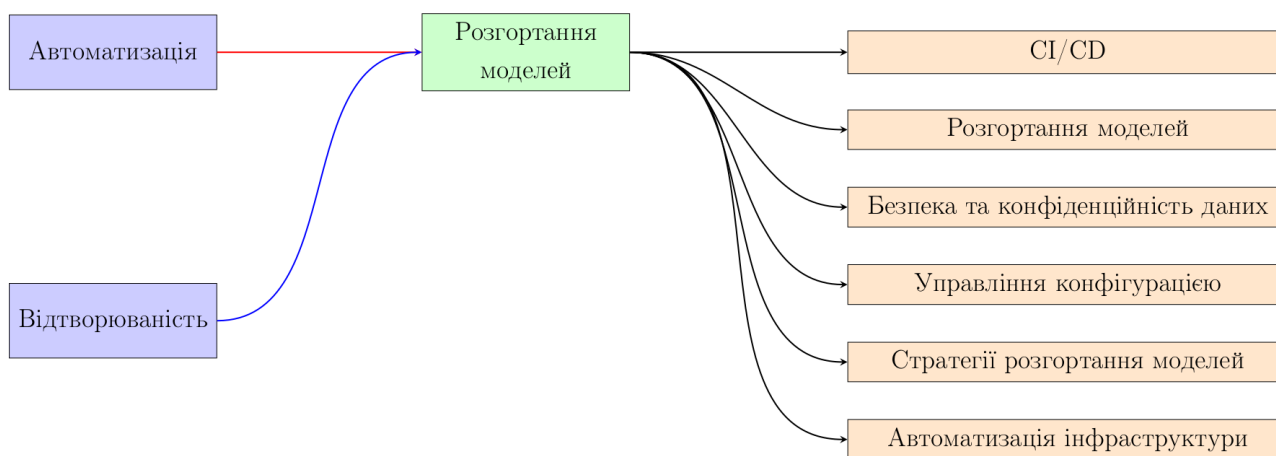


Рисунок 21. Зв'язки принципів, процесів та практик MLOps під час розгортання моделей

Висновки

Здійснено метасинтез систематичних оглядів [5, 6, 8] та огляду продуктів і постачальників [15] з метою узагальнення знань про впровадження практик MLOps для ефективного розгортання моделей машинного навчання. За результатами метасинтезу сформовано 10 висновків.

1. MLOps – це підхід до управління, автоматизації та операціоналізації процесів розробки, розгортання та підтримки моделей машинного навчання на основі практик програмної інженерії та DevOps. MLOps базується на наборі принципів, процесів і практик, що

забезпечують ефективну розробку, розгортання та підтримку моделей машинного навчання.

2. Основні етапи життєвого циклу MLOps включають процеси збору та обробки даних, розробки та навчання моделей, розгортання, моніторинг та донавчання моделей.
3. Для впровадження MLOps використовуються різні фреймворки та архітектури, як-от платформи з відкритим кодом MLflow, Kubeflow та TensorFlow Extended, хмарні обчислювальні платформи AWS, Google Cloud та Azure, контейнеризація (Docker) і оркестрація контейнерів (Kubernetes).
4. Інструменти MLOps пропонують широкий спектр функцій для підтримки життєвого циклу моделі машинного навчання, з акцентом на автоматизацію, відстеження експериментів, версіонування, моніторинг та розгортання моделей.
5. Найпоширенішими способами розгортання моделей машинного навчання в робочих середовищах є використання технологій контейнеризації, хмарних платформ і сервісів, а також розгортання моделей як вебсервісів.
6. Для оцінювання рівня зрілості процесів MLOps в організаціях можуть використовуватися адаптовані моделі зрілості розробки програмного забезпечення, як-от Capability Maturity Model.
7. Успішне впровадження MLOps вимагає залучення фахівців із різних сфер – з розробки програмного забезпечення, інженерії даних, машинного навчання, предметних експертів і менеджменту.
8. Основними викликами під час розгортання моделей машинного навчання в робочих середовищах є управління життєвим циклом моделі, забезпечення масштабованості та продуктивності, моніторинг і підтримка моделей у реальних умовах.
9. Відкритими питаннями та викликами в MLOps є необхідність розробки стандартів і найкращих практик, забезпечення інтерпретованості та відповідального використання моделей, ефективне управління даними та інтеграція знань із різних сфер.
10. Основними можливостями та тенденціями розвитку MLOps є створення стандартизованих платформ, застосування в контексті розподіленого навчання та інтеграція з іншими підходами до управління життєвим циклом даних і моделей. Поточні та майбутні сфери застосування MLOps включають широкий спектр галузей, від фінансів і охорони здоров'я до Інтернету речей та обробки природної мови.

Проведений метасинтез показав, що MLOps є перспективним підходом для ефективного розгортання моделей машинного навчання в робочих середовищах, який вимагає подальших досліджень і розробок для вирішення поточних викликів та реалізації потенційних можливостей.

Також проаналізовано ключові практики MLOps, які необхідні для ефективного розгортання моделей машинного навчання. За результатами аналізу сформовано 5 висновків.

1. MLOps базується на наборі принципів, процесів і практик, що забезпечують ефективну розробку, розгортання та підтримку моделей машинного навчання. Ключовими принципами MLOps є автоматизація, відтворюваність, співпраця, постійне навчання та управління даними.

2. Основні практики MLOps включають безперервну інтеграцію та доставку (CI / CD), версіонування моделей і даних, автоматизацію конвеєрів машинного навчання, моніторинг продуктивності моделей, управління експериментами, розгортання моделей і управління життєвим циклом.
3. Додаткові практики MLOps, як-от безпека даних і конфіденційність, пояснюваність і інтерпретованість моделей, управління якістю даних, управління конфігурацією, стратегії розгортання моделей, автоматизація інфраструктури, співпраця та комунікація, управління ризиками й відповідність вимогам, є важливими для забезпечення надійності, відповідності вимогам та ефективності процесів MLOps.
4. Застосування практик MLOps дає змогу автоматизувати та стандартизувати процеси розробки, розгортання й підтримки моделей машинного навчання, що підвищує ефективність і надійність рішень у робочому середовищі.
5. Успішне впровадження практик MLOps вимагає використання відповідних інструментів і платформ, як-от системи управління експериментами, версіонування даних і моделей, автоматизація інфраструктури та інструменти моніторингу, а також налагодження ефективної співпраці між різними ролями й командами, залученими до розробки та впровадження моделей машинного навчання.

Аналіз показав, що застосування практик MLOps є критично важливим для успішного розгортання моделей машинного навчання в робочих середовищах. Впровадження таких практик сприяє підвищенню ефективності, надійності і відтворюваності процесів розробки та експлуатації рішень на базі машинного навчання, що є необхідною умовою для їх успішного використання в реальних бізнес-задачах.

Внаслідок комплексного дослідження ідентифіковано та проаналізовано практики MLOps, які необхідні для ефективного розгортання моделей машинного навчання. Основні висновки, отримані під час дослідження, є такими:

- 1) MLOps є перспективним підходом для ефективного розгортання моделей машинного навчання в робочих середовищах, який вимагає подальших досліджень і розробок для вирішення поточних викликів і реалізації можливостей.
- 2) запропонована діаграма взаємозв'язків між принципами, процесами та практиками MLOps, яка наочно ілюструє їх взаємодію на кожному етапі процесу розробки та впровадження моделей машинного навчання;
- 3) визначено найбільш ефективні практики MLOps для розгортання моделей, які включають безперервну інтеграцію та доставку (CI / CD), версіонування моделей і даних, автоматизацію конвеєрів машинного навчання, моніторинг продуктивності моделей, управління експериментами, розгортання моделей і управління життєвим циклом, безпеку даних та конфіденційність, пояснюваність і інтерпретованість моделей, управління якістю даних, управління конфігурацією, стратегії розгортання моделей, автоматизацію інфраструктури, співпрацю та комунікацію, управління ризиками та відповідність вимогам.

Теоретична значущість отриманих у статті результатів полягає в узагальненні та систематизації знань про практики MLOps, які необхідні для ефективного розгортання моделей машинного навчання. Практична значущість результатів полягає в можливості їх

використання організаціями для впровадження або вдосконалення процесів MLOps з метою підвищення ефективності та надійності розгортання моделей машинного навчання в робочих середовищах.

Подальші дослідження можуть бути спрямовані на розробку детальних рекомендацій щодо впровадження окремих практик MLOps в організаціях, створення нових інструментів і платформ для автоматизації та управління життєвим циклом моделей машинного навчання, а також на вивчення ефективності практик MLOps у різних галузях і сферах застосування моделей машинного навчання.

Література

1. Zahorodko, P. V., Semerikov, S. O., Soloviev, V. N., Striuk, A. M., Striuk, M. I., & Shalatska, H. M. (2021). Comparisons of performance between quantum-enhanced and classical machine learning algorithms on the IBM Quantum Experience. *Journal of Physics: Conference Series*, 1840, 012021. <https://doi.org/10.1088/1742-6596/1840/1/012021>
2. Kreuzberger, D., Kühl, N., & Hirschl, S. (2023). Machine learning operations (MLOps): overview, definition, and architecture. *IEEE Access*, 11, 31866–31879. <https://doi.org/10.1109/ACCESS.2023.3262138>
3. Symeonidis, G., Nerantzis, E., Kazakis, A., & Papakostas, G. A. (2022). MLOps – definitions, tools and challenges. In *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 0453–0460). <https://doi.org/10.1109/CCWC54503.2022.9720902>
4. Testi, M., Ballabio, M., Frontoni, E., Iannello, G., Moccia, S., Soda, P., & Vessio, G. (2022). MLOps: A taxonomy and a methodology. *IEEE Access*, 10, 63606–63618. <https://doi.org/10.1109/ACCESS.2022.3181730>
5. Diaz-de Arcaya, J., Torre-Bastida, A. I., Zárate, G., Miñón, R., & Almeida, A. (2023). A joint study of the challenges, opportunities, and roadmap of MLOps and AIOps: A systematic survey. *ACM Computing Surveys*, 56, 84. <https://doi.org/10.1145/3625289>
6. Recupito, G., Pecorelli, F., Catolino, G., Moreschini, S., Nucci, D. D., Palomba, F., & Tamburri, D. A. (2022). A multivocal literature review of mlops tools and features. In *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 84–91). <https://doi.org/10.1109/SEAA56994.2022.00021>
7. Steidl, M., Felderer, M., & Ramler, R. (2023). The pipeline for the continuous development of artificial intelligence models – current state of research and practice. *Journal of Systems and Software*, 199, 111615. <https://doi.org/10.1016/j.jss.2023.111615>
8. Lima, A., Monteiro, L., & Furtado, A. P. (2022). MLOps: practices, maturity models, roles, tools, and challenges – a systematic literature review. In *Proceedings of the 24th International Conference on Enterprise Information Systems – Volume 1: ICEIS* (pp. 308–320). <https://doi.org/10.5220/0010997300003179>
9. Haller, K. (2022). *Managing AI in the enterprise: Succeeding with AI projects and MLOps to build sustainable AI organizations*. Apress Berkeley, CA. <https://doi.org/10.1007/978-1-4842-7824-6>

10. e Oliveira, E., Rodrigues, M., Pereira, J. P., Lopes, A. M., Mestric, I. I., & Bjelogrljic, S. (2024). Unlabeled learning algorithms and operations: overview and future trends in defense sector. *Artificial Intelligence Review*, 57, 66. <https://doi.org/10.1007/s10462-023-10692-0>
11. Kolltveit, A. B., & Li, J. (2023). Operationalizing machine learning models: A systematic literature review. In *Proceedings of the 1st Workshop on Software Engineering for Responsible AI* (pp. 1–8). <https://doi.org/10.1145/3526073.3527584>
12. Calefato, F., Lanubile, L., & Quaranta, L. (2022). A preliminary investigation of MLOps practices in GitHub. In *Proceedings of the 16th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 283–288). <https://doi.org/10.1145/3544902.3546636>
13. Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372, n71. <https://doi.org/10.1136/bmj.n71>
14. Haertel, C., Staegemann, D., Daase, C., Pohl, M., Nahhas, A., & Turowski, K. (2023). MLOps in data science projects: A review. In *2023 IEEE International Conference on Big Data (BigData)* (pp. 2396–2404). <https://doi.org/10.1109/BigData59044.2023.10386139>
15. Cohen, R. (2023). Digital strategy, machine learning, and industry survey of MLOps. In *Digital Strategies and Organizational Transformation* (pp. 137–150). https://doi.org/10.1142/9789811271984_0008
16. Sipe, T. A., & Curlette, W. L. (1996). A meta-synthesis of factors related to educational achievement: A methodological approach to summarizing and synthesizing meta-analyses. *International Journal of Educational Research*, 25, 583–698. [https://doi.org/10.1016/S0883-0355\(96\)80001-2](https://doi.org/10.1016/S0883-0355(96)80001-2)
17. Chrastina, J. (2018). Meta-synthesis of qualitative studies: Background, methodology and applications. In *NORDSCI Conference proceedings* (Vol. 1). <https://doi.org/10.32008/nordsci2018/b1/v1/13>
18. Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., & Zimmermann, T. (2019). Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)* (pp. 291–300). <https://doi.org/10.1109/ICSE-SEIP.2019.00042>
19. Dhanorkar, S., Wolf, C. T., Qian, K., Xu, A., Popa, L., & Li, Y. (2021). Who needs to know what, when?: Broadening the explainable AI (XAI) design space by looking at explanations across the AI lifecycle. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference* (pp. 1591–1602). <https://doi.org/10.1145/3461778.3462131>
20. Lwakatare, L. E., Crnkovic, I., & Bosch, J. (2020). DevOps for AI – challenges in development of AI-enabled applications. In *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)* (pp. 1–6). <https://doi.org/10.23919/SoftCOM50211.2020.9238323>
21. Akkiraju, R., Sinha, V., Xu, A., Mahmud, J., Gundecha, P., Liu, Z., Liu, X., & Schumacher, J. (2020). Characterizing machine learning processes: A maturity framework. In D. Fahland, C. Ghidini, J. Becker, & M. Dumas (eds.), *Business Process Management* (Vol. 12168, pp. 17–31). Springer International Publishing. https://doi.org/10.1007/978-3-030-58666-9_2

22. Min, C., Mathur, A., Acer, U. G., Montanari, A., & Kawsar, F. (2023). SensiX++: Bringing MLOps and multi-tenant model serving to sensory edge devices. *ACM Transactions on Embedded Computing Systems*, 22, 98. <https://doi.org/10.1145/3617507>
23. Bachinger, F., Zenisek, J., & Affenzeller, M. (2024). Automated machine learning for industrial applications – challenges and opportunities. *Procedia Computer Science*, 232, 1701–1710. <https://doi.org/10.1016/j.procs.2024.01.168>
24. Filippou, K., Aifantis, G., Papakostas, G. A., & Tsekouras, G. E. (2023). Structure learning and hyperparameter optimization using an automated machine learning (AutoML) Pipeline. *Information*, 14, 232. <https://doi.org/10.3390/info14040232>
25. Bodor, A., Hnida, M., & Daoudi, N. (2023). Machine learning models monitoring in MLOps context: Metrics and tools. *International Journal of Interactive Mobile Technologies (iJIM)*, 17, 125–139. <https://doi.org/10.3991/ijim.v17i23.43479>
26. Singh, P. (2023). Systematic review of data-centric approaches in artificial intelligence and machine learning. *Data Science and Management*, 6, 144–157. <https://doi.org/10.1016/j.dsm.2023.06.001>
27. Czakon, J., & Kluge, K. (2024). ML experiment tracking: What it is, why it matters, and how to implement it. <https://neptune.ai/blog/ml-experiment-tracking>
28. Peltonen, E., & Dias, S. (2023). LinkEdge: Open-sourced MLOps integration with IoT edge. In *Proceedings of the 3rd Eclipse Security, AI, Architecture and Modelling Conference on Cloud to Edge Continuum* (pp. 67–76). <https://doi.org/10.1145/3624486.3624496>
29. Melgar, L. A., Dao, D., Gan, S., Gürel, N. M., Hollenstein, N., Jiang, J., ... Zhang, C. (2021). Ease.ML: A lifecycle management system for MLDev and MLOps. In *11th Conference on Innovative Data Systems Research, CIDR 2021*. https://www.cidrdb.org/cidr2021/papers/cidr2021_paper26.pdf
30. Chen, H., & Babar, M. A. (2024). Security for machine learning-based software systems: A survey of threats, practices, and challenges. *ACM Computing Surveys*, 56, 151. <https://doi.org/10.1145/3638531>
31. Gopalakrishna, N. K., Anandayuvraj, D., Detti, A., Bland, F. L., Rahaman, S., & Davis, J. C. (2023). “If security is required”: engineering and security practices for machine learning-based IoT devices. In *Proceedings of the Fourth International Workshop on Software Engineering Research and Practice for the IoT* (pp. 1–8). <https://doi.org/10.1145/3528227.3528565>
32. Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267, 1–38. <https://doi.org/10.1016/j.artint.2018.07.007>
33. Rezazadeh, F., Chergui, H., Alonso, L., & Verikoukis, C. (2024). SliceOps: Explainable MLOps for streamlined automation-native 6G networks. *IEEE Wireless Communications*, 31, 224–230. <https://doi.org/10.1109/MWC.007.2300144>
34. Godwin, R. C., & Melvin, R. L. (2024). Toward efficient data science: A comprehensive MLOps template for collaborative code development and automation. *SoftwareX*, 26, 101723. <https://doi.org/10.1016/j.softx.2024.101723>

35. Yongqiang, D., Xin, W., Yongbo, L., & Wang, Y. (2020). Building network domain knowledge graph from heterogeneous YANG models. *Journal of Computer Research and Development*, 57, 699–708. <https://doi.org/10.7544/issn1000-1239.2020.20190882>
36. Neptune Labs. (2024). MLOps landscape in 2024: Top tools and platforms. <https://neptune.ai/blog/mlops-tools-platforms-landscape>
37. Gunny, A., Rankin, D., Harris, P., Katsavounidis, E., Marx, E., Saleem, M., Coughlin, M., & Benoit, W. (2022). A software ecosystem for deploying deep learning in gravitational wave physics. In *Proceedings of the 12th Workshop on AI and Scientific Computing at Scale Using Flexible Computing Infrastructures* (pp. 9–17). <https://doi.org/10.1145/3526058.3535454>
38. Vuppalapati, C., Ilapakurti, A., Chillara, K., Kedari, S., & Mamidi, V. (2020). Automating tiny ML intelligent sensors DevOPS using Microsoft Azure. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 2375–2384). <https://doi.org/10.1109/BigData50022.2020.9377755>
39. Sothilingam, R., Pant, V., & Yu, E. S. K. (2022). Using i* to analyze collaboration challenges in MLOps project teams. In A. Maté, T. Li, & E. J. T. Gonçalves (eds.), *Proceedings of the 15th International iStar Workshop (iStar 2022) co-located with 41th International Conference on Conceptual Modeling (ER 2022)* (pp. 1–6). https://ceur-ws.org/Vol-3231/iStar22_paper_1.pdf

Рукопис отримано – 04/03/2025 р.; прийнято до публікації – 26/03/2025 р.

MLOps engineering: A meta-synthesis of tools, practices and architectures for machine learning automation

Danylo Hanchuk, Serhiy Semerikov

Abstract

Automating the end-to-end lifecycle of machine learning models is critical for their effective operationalization in production environments. Various tools, frameworks and architectures have emerged to support Machine Learning Operations (MLOps) practices. This paper presents a meta-synthesis of existing reviews to provide a comprehensive overview of enabling technologies for MLOps. The capabilities and features offered by popular commercial and open-source MLOps platforms are compared. Patterns in MLOps architecture and design philosophies are identified. The paper examines the role of containerization, orchestration, configuration management, and infrastructure automation in ML-pipelines. Approaches for model deployment on cloud and edge are also discussed. The following main results are obtained: 1) a meta-synthesis of systematic reviews was conducted to summarize knowledge about MLOps practices; it was determined that MLOps is a promising approach for effective deployment of machine learning models that requires further research; 2) relationships between MLOps principles, processes, and practices were analyzed. A diagram of the interconnections between key principles, stages of model development and implementation, and main MLOps practices is proposed; 3) the most effective MLOps practices for model deployment were identified – continuous integration/delivery, model and data versioning, ML pipeline automation, performance monitoring, experiment management, lifecycle management, data security and privacy, model explainability, data quality management, configuration management, deployment strategies, infrastructure automation, collaboration, risk management. The results obtained have theoretical significance in generalizing and systematizing knowledge about MLOps practices and practical significance for implementing and improving MLOps processes in organizations.

Keywords: MLOps, automation, tools, frameworks, architecture, model deployment, ML-pipelines, meta-synthesis.