МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
Фізико-математичний факультет
Кафедра інформатики та прикладної математики

«Допущено до захисту»
Завідувач кафедри
_____ Моісеєнко Н. В.
«____» _____ 2024 р.

Реєстраційний №____
«____» _____ 2024 р.

УПРОВАДЖЕННЯ ПРАКТИК MLOPS ДЛЯ ЕФЕКТИВНОГО РОЗГОРТАННЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

Кваліфікаційна робота студента групи І-20
ступінь вищої освіти «бакалавр»
спеціальності 014.09 Середня освіта (Інформатика)
Ганчука Данила Олеговича

Керівник: доктор педагогічних наук, професор,
старший дослідник
Семеріков Сергій Олексійович

Оцінка:
Національна шкала _____
Шкала ECTS ____ Кількість балів ____
Голова ФК _____ _____
Члени ФК _____ _____
_____ _____
_____ _____
_____ _____

Кривий Ріг – 2024

ЗАПЕВНЕННЯ

Я, Ганчук Данило Олегович, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

Ганчук

ЗМІСТ

ВСТУП

Актуальність теми. У сучасному світі машинне навчання стає все більш важливою технологією, яка знаходить застосування в різноманітних галузях, таких як фінанси, охорона здоров'я, промисловість, роздрібна торгівля тощо. Однак, незважаючи на значний прогрес у розробці алгоритмів та моделей машинного навчання, їх ефективне розгортання у виробничих середовищах залишається складним завданням [19; 22]. Це обумовлено низкою факторів, таких як необхідність забезпечення масштабованості, відтворюваності, безпеки та надійності моделей, а також складністю інтеграції процесів розробки та експлуатації.

Для вирішення цих проблем виникла методологія MLOps (Machine Learning Operations), яка має на меті застосування принципів та практик DevOps до процесів розробки та розгортання моделей машинного навчання [2; 24]. MLOps охоплює широкий спектр практик, таких як автоматизація конвеєрів машинного навчання, версіонування даних та моделей, моніторинг продуктивності моделей, управління експериментами тощо [3; 33]. Дослідження показують, що застосування практик MLOps дозволяє суттєво підвищити ефективність та надійність розгортання моделей машинного навчання у виробничих середовищах [17; 20].

Водночас, незважаючи на значний інтерес до теми MLOps з боку як науковців, так і практиків, в даній галузі все ще існують певні прогалини та невирішені проблеми. Зокрема, відсутні загальноприйняті стандарти та кращі практики впровадження MLOps, недостатньо досліджені питання інтеграції MLOps з іншими підходами (DataOps, ModelOps, AIOps тощо), а також існує потреба в розробці нових інструментів та платформ для автоматизації процесів MLOps [2; 18; 36]

Дана 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.

Об'єкт дослідження – процес розгортання моделей машинного на-

вчання у виробничих середовищах.

Предмет дослідження – практики MLOps, необхідні для ефективного розгортання моделей машинного навчання.

Мета дослідження – визначити та проаналізувати практики MLOps, необхідні для ефективного розгортання моделей машинного навчання.

Відповідно до мети визначено такі основні **завдання дослідження**:

1. Виконати мета-синтез 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.

Методи дослідження:

- *системний аналіз* принципів, процесів та практик MLOps;
- *мета-синтез* систематичних оглядів для узагальнення знань щодо практик MLOps;
- *моделювання* зв'язків між принципами, процесами та практиками MLOps.

Наукова новизна отриманих результатів полягає в наступному:

- *удосконалено*:
  - підхід до визначення та систематизації практик MLOps на основі мета-синтезу систематичних оглядів, що дозволило виявити найбільш ефективні практики для розгортання моделей машинного навчання;
  - класифікацію практик MLOps шляхом виділення основних та додаткових практик, що сприяє більш чіткому розумінню їх призначення та особливостей застосування;

- *набуло подальшого розвитку* моделювання зв'язків між принципами, процесами та практиками MLOps, що дозволило визначити роль та місце окремих практик у загальному процесі розгортання моделей машинного навчання.

Практичне значення отриманих результатів полягає в можливості їх використання організаціями для впровадження або вдосконалення процесів MLOps з метою підвищення ефективності та надійності розгортання моделей машинного навчання у виробничих середовищах. Запропонована схема зв'язків між принципами, процесами та практиками MLOps може бути використана як концептуальна основа для розробки стратегії впровадження MLOps в організації. Виявлені найбільш ефективні практики MLOps можуть слугувати орієнтиром для визначення пріоритетів при впровадженні або вдосконаленні процесів розгортання моделей машинного навчання.

Структура та обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, двох розділів, висновків до них, загальних висновків, списку використаних джерел (37 найменувань), двох додатків. Робота містить 3 таблиці та 21 рисунків. Загальний обсяг кваліфікаційної роботи – 100 сторінок.

РОЗДІЛ 1
МЕТА-СИНТЕЗ ПРАКТИК MLOPS

1.1. Основні поняття дослідження

DevOps (Development & Operations) набуває все більшого поширення, і компанії застосовують його методи в різних галузях [3]. У цьому контексті *MLOps* (Machine Learning & Operations) автоматизують робочі процеси ML (Machine Learning – машинне навчання), такі, як конвеєри, застосовуючи практики DevOps (безперервну інтеграцію/безперервне розгортання (CI/CD – continuous integration/continuous deployment) для проєктів машинного навчання) [3; 9; 20].

Згідно з Calefato, Lanubile, Quaranta [9], ключові практики MLOps можуть бути реалізовані за допомогою GitHub Actions і CML (Continuous Machine Learning – безперервне машинне навчання). Хоча деякі робочі процеси автоматизують завдання ML за допомогою GitHub Actions і CML, наскрізні конвеєри MLOps виробничого рівня в проаналізованих проєктах з відкритим кодом на GitHub зустрічаються рідко. Практики більше зосереджені на звітності та метриках, ніж на перенавчання чи розгортанні.

1.2. 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" [35, с. 1]. Основною метою систематичного огляду є сприяння прийняттю обґрунтованих рішень (таких, що ґрунтуються на доведених фактах) [35, с. 6]. 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, джере-

ла відбіраються 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 року ми здійснили пошуковий запит до бази даних Scopus за назвою статті:

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

Було виявлено 5 документів (табл. 1.1), з яких 3 [2; 3; 20] відносяться до систематичних оглядів.

Таблиця 1.1

Результати пошуку наявних систематичних оглядів у Scopus.

Бібліографічний опис	**Зміст огляду**
A Multivocal Literature Review of MLOps Tools and Features / G. Recupito [et al.] // 2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). 2022. P. 84–91. DOI: 10.1109/SEAA56994.2022.00021	Recupito та ін. [3] провели "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 *визначити інструменти* створення конвеєрів MLOps та проаналізувати їхні основні характеристики та особливості. Автори дослідили функціональність 13 інструментів MLOps та показали, що більшість інструментів MLOps підтримують однакові функції, але застосовують різні підходи, які можуть надавати різні переваги залежно від вимог користувача.

Продовження на наступній сторінці

Продовження таблиці 1.1

Бібліографічний опис	**Зміст огляду**
Lima A., *Monteiro L.*, *Furtado A. P.* MLOps: Practices, Maturity Models, Roles, Tools, and Challenges – A Systematic Literature Review // Proceedings of the 24th International Conference on Enterprise Information Systems - Volume 1: ICEIS. INSTICC. SciTePress, 2022. P. 308–320. ISBN 978-989-758-569-2. DOI: 10.5220/0010997300003179	Lima, Monteiro, Furtado [20] здійснено систематичний огляд літератури з метою визначення практик, стандартів, ролей, моделей зрілості, викликів та інструментів MLOps. 30 статей були відібрані для аналізу. Результати дослідження дозволили зробити висновок, що MLOps все ще перебуває на початковій стадії.
MLOps in Data Science Projects: A Review / C. Haertel [et al.] // 2023 IEEE International Conference on Big Data (BigData). 2023. P. 2396–2404. DOI: 10.1109/BigData59044.2023.10386139	Haertel та ін. [23] надали огляд застосувань MLOps у проєктах Data Science. Автори показали, що при розгляді сучасних підходів MLOps наголос робиться на розробці та розгортанні моделі, тоді як організаційним аспектам (бізнес-розуміння, оцінка) приділяється недостатня увага. Оскільки успіх проєкту Data Science не залежить виключно 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.
Cohen R. Digital Strategy, Machine Learning, and Industry Survey of MLOps // Digital Strategies and Organizational Transformation. 2023. Chap. 8. P. 137–150. DOI: 10.1142/9789811271984_0008. URL: https://tinyurl.com/33z6zpd3	As part of a digital strategy, machine learning (ML) has become a common toolset and capability across many businesses. However, the operational aspects of machine learning (MLOps) are often overlooked for ML projects until they are already installed and being executed in the business environment. This chapter provides a review of MLOps products and vendors to give data scientists the ability to set up the appropriate ML infrastructure in a proactive manner.

Продовження на наступній сторінці

Продовження таблиці 1.1

Бібліографічний опис	**Зміст огляду**
A Joint Study of the Challenges, Opportunities, and Roadmap of MLOps and AIOps: A Systematic Survey / J. Diaz-de-Arcaya [et al.] // ACM Comput. Surv. New York, NY, USA, 2023. Oct. Vol. 56, no. 4. DOI: 10.1145/3625289	Diaz-de-Arcaya та ін. [2] аналізують проблеми, можливості та перспективи впровадження MLOps і AIOps. Автори проаналізували відкриті питання, можливості та тенденції, з якими стикаються організації при впровадженні MLOps та AIOps, фреймворки та 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, таких як технології 5G і 6G, тоді як MLOps більш поширені у традиційних промислових середовищах.

Haertel та ін. [23] та Cohen [13] не є систематичними оглядами, проте отримані у даних роботах результати було взято до уваги при виконанні мета-синтезу [29] для об'єднання (додаток А на с. 88) та тематичного аналізу результатів, отриманих у систематичних оглядах.

Мета-синтез було виконано згідно Chrastina [12, с. 123-125]:

1. *Визначення предмета дослідження*: практики MLOps для ефективного розгортання моделей.

2. *Визначення релевантних джерел*: систематичні літературні огляди [2; 3; 20] та огляд продуктів та постачальників MLOps [13].

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 визначення та групування ключових тем.

5. *Взаємна трансляція результатів різних робіт* через визначення спільної термінології, пояснення протиріч у результатах із різних робіт та узагальнення результатів із різних робіт.

6. *Синтез результатів.*

7. *Оприлюднення мета-синтезу.*

1.3. Ретельне вивчення та визначення зв'язку між роботами

1.3.1. Розподіл оглядів за роками

Дві роботи [3; 20] відносяться до 2022 року, одна [2] – до 2023 року. Водночас джерела, що аналізуються у [3], обмежені 2020 роком, у [20] – 2021, а у [2] – 2023 роком. Крім того, робота [2] згадує роботу [20] як попередню.

1.3.2. Цілі оглядів

Метою мета-синтезу цілей оглядів [2; 3; 20] було виявити спільні та відмінні аспекти щодо загальної спрямованості та завдань цих досліджень.

Спільними аспектами цілей розглянутих оглядів є:

1. Всі огляди [2; 3; 20] спрямовані на дослідження та узагальнення знань щодо методології MLOps, її практик, інструментів та викликів.

2. Огляди [3; 20] мають на меті виявлення та аналіз інструментів MLOps, які використовуються для автоматизації процесів розробки та розгортання моделей машинного навчання.

3. Огляди [2; 20] прагнуть надати розуміння щодо загального стану впровадження практик MLOps в індустрії та академічній сфері.

Відміннними аспектами цілей оглядів є:

1. Огляд [3] більше фокусується на виявленні та аналізі функціональних можливостей інструментів MLOps для створення конвеєрів машинного навчання.

2. Огляд [20] приділяє увагу ширшому колу аспектів MLOps, таких як практики, ролі, моделі зрілості, виклики, на додачу до інструментів.

3. Огляд [2], окрім методології MLOps, також розглядає споріднену концепцію AIOps. Більше уваги приділяється висвітленню можливостей, викликів та майбутніх трендів в обох областях.

Отже, 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.

1.3.3. Дослідницькі питання оглядів

Метою мета-синтезу дослідницьких питань оглядів [2; 3; 20] було виявити спільні та відмінні аспекти щодо основних напрямків дослідження в рамках вивчення методології MLOps.

Спільними аспектами дослідницьких питань розглянутих оглядів є:

1. Всі огляди [2; 3; 20] містять питання щодо інструментів та платформ, які використовуються для впровадження практик MLOps, автоматизації процесів розробки, розгортання та моніторингу моделей машинного навчання.

2. Огляди [2; 20] включають питання стосовно викликів та відкритих проблем, з якими стикаються організації при впровадженні MLOps.

3. Огляди [2; 20] розглядають питання щодо можливостей та майбутніх трендів в області MLOps.

Відміннними аспектами дослідницьких питань оглядів є:

1. Огляд [3] містить більш специфічні питання щодо функціональних можливостей та особливостей інструментів MLOps для створення конвеєрів машинного навчання.

2. Огляд [20] включає питання стосовно ролей та обов'язків спеціалістів, залучених до впровадження MLOps, а також моделей зрілості для оцінки рівня автоматизації процесів розгортання моделей.

3. Огляд [2], окрім MLOps, також розглядає питання, специфічні для методології AIOps, та приділяє увагу поточним і майбутнім сферам застосування цих підходів.

Отже, 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 та фактори, які впливають на впровадження та розвиток MLOps практик в організаціях.

1.3.4. Інформаційні джерела оглядів

Метою мета-синтезу інформаційних джерел оглядів [2; 3; 20] було виявити спільні та відмінні аспекти щодо використаних баз даних, пошукових систем та типів літератури для пошуку релевантних досліджень.

Спільними аспектами інформаційних джерел розглянутих оглядів є:

1. Всі огляди [2; 3; 20] використовували електронні бази даних наукових публікацій для пошуку релевантних досліджень.

2. Огляди [2; 3] включали в пошук як академічні (рецензовані), так і неакадемічні ("сірі") джерела літератури, такі як блоги, веб-сайти, відео, репозиторії коду тощо.

Відмінними аспектами інформаційних джерел оглядів є:

1. Огляд [3] використовував Google Scholar для пошуку наукових публікацій та звичайний пошук Google для "сірої" літератури.

2. Огляд [20] обмежився пошуком лише в академічних базах даних, таких як ACM Digital Library, IEEE Xplore, ScienceDirect та SpringerLink.

3. Огляд [2] використовував декілька баз даних (Scopus, arXiv, Springer, IEEE), але основним джерелом була база даних Scopus від Elsevier.

Отже, розглянуті огляди демонструють різні підходи до вибору інформаційних джерел. Деякі дослідження [2; 3] включають як академічні, так і неакадемічні джерела для отримання більш повної картини щодо практичного застосування MLOps. Інші [20] фокусуються виключно на рецензованих наукових публікаціях. Вибір джерел може впливати на охоплення та тип знайдених досліджень, а отже і на результати та висновки оглядів.

1.3.5. Критерії включення інформаційних джерел до оглядів

Метою мета-синтезу критеріїв включення інформаційних джерел до оглядів [2; 3; 20] було виявити спільні та відмінні аспекти щодо вимог, яким повинні відповідати дослідження для включення в аналіз.

Спільними аспектами критеріїв включення у розглянутих оглядах є:

1. Всі огляди [2; 3; 20] включали дослідження, які безпосередньо стосуються теми MLOps, її практик, інструментів та застосування.

2. Огляди [3; 20] розглядали дослідження, які описують досвід, практики, архітектуру або реалізацію інструментів та процесів MLOps.

Відміннними аспектами критеріїв включення у оглядах є:

1. Огляд [3] включав дослідження, які описують компоненти мінімального життєвого циклу MLOps або представляють досвід та думки експертів щодо MLOps.

2. Огляд [20] додатково включав дослідження, які оцінюють зрілість процесів MLOps, розглядають ролі та обов'язки в життєвому циклі розробки моделей ML, а також визначають виклики в розробці та впровадженні рішень MLOps.

3. Огляд [2] мав більш загальні критерії включення, розглядаючи дослідження, опубліковані з 2018 по 2023 роки, які містять нові ідеї та тісно пов'язані з темою MLOps та AIOps.

Отже, 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 ML. Огляди [2; 20] мають дещо ширші критерії, включаючи також дослідження, пов'язані з оцінкою зрілості, ролями та викликами MLOps.

1.3.6. Критерії виключення інформаційних джерел з оглядів

Метою мета-синтезу критеріїв виключення інформаційних джерел оглядів [2; 3; 20] було виявити спільні та відмінні аспекти щодо характеристик досліджень, які призводять до їх виключення з аналізу.

Спільними аспектами критеріїв виключення у розглянутих оглядах є:

1. Огляди [3; 20] виключали дослідження, які не надають достатньо деталей щодо архітектури, реалізації або застосування інструментів та процесів MLOps.

2. Огляди [2; 20] виключали дослідження, опубліковані не англійською мовою.

Відмінними аспектами критеріїв виключення інформаційних джерел з оглядів є:

1. Огляд [3] виключав дослідження, які просувають комерційні платформи MLOps без надання деталей щодо їх реалізації або використання.

2. Огляд [20] виключав дослідження, які стосуються лише застосування моделей ML без розгляду аспектів MLOps, а також короткі статті, постери та дослідження без доступу до повного тексту.

3. Огляд [2] 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.

Отже, розглянуті огляди застосовують різні критерії виключення для відсіювання досліджень, які не відповідають їхнім вимогам. Спільним є виключення досліджень з недостатнім описом MLOps процесів та інструментів, а також не англомовних публікацій. Відмінності полягають у додаткових критеріях, таких як виключення комерційних платформ без технічних деталей [3], коротких статей та постерів [20], а також матеріалів з обмеженим доступом та низькою кількістю цитувань [2].

1.3.7. Критерії якості інформаційних джерел у оглядах

Метою мета-синтезу критеріїв якості оглядів [2; 3; 20] було виявити спільні та відмінні аспекти щодо вимог до якості та надійності досліджень, включених до аналізу.

Спільними аспектами критеріїв якості у розглянутих оглядах є:

1. Огляди [2; 20] оцінювали якість досліджень на основі повноти опису методології, контексту та результатів.

2. Огляди [2; 20] враховували наявність обґрунтованих доказів та аргументів на підтримку висновків дослідження.

Відміннними аспектами критеріїв якості в оглядах є:

1. Огляд [3] використовував кількісні показники популярності (кількість зірок на Github, переглядів на YouTube) для оцінки якості та релевантності "сірої" літератури.

2. Огляд [20] оцінював, чи представляє дослідження емпіричні результати, а не лише думки експертів, а також чи валідовані результати належним чином.

3. Огляд [2] використовував розширений набір критеріїв, включаючи наявність всебічного огляду літератури, перевірку результатів на прикладах використання, кількість досліджуваних питань, наявність відкритого доступу, публікацію в журналах з високим імпакт-фактором та кількість цитувань.

Отже, розглянуті огляди застосовують різні підходи до оцінки 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 [3], до оцінки емпіричності результатів [20] та врахування бібліометричних показників, таких як імпакт-фактор журналу та кількість цитувань [2].

1.4. Взаємна трансляція результатів різних робіт та синтез результатів

Для взаємної трансляція результатів різних робіт, крім систематичних оглядів [2; 3; 20], був залучений огляд продуктів та постачальників MLOps [13].

1.4.1. Визначення MLOps

Метою мета-синтезу визначень MLOps в оглядах [2; 3; 13; 20] було виявити спільні та відмінні аспекти в розумінні та трактуванні цього поняття.

Спільними аспектами визначень MLOps в розглянутих оглядах є:

1. Всі огляди [2; 3; 13; 20] розглядають MLOps як набір практик, принципів та процесів для автоматизації та управління життєвим циклом моделей машинного навчання.

2. Огляди [2; 3] наголошують на використанні в MLOps підходів та практик з DevOps, таких як безперервна інтеграція, доставка та моніторинг.

3. Огляди [13; 20] підкреслюють роль MLOps в операціоналізації рішень машинного навчання та передачі їх в промислову експлуатацію.

Відмінними аспектами визначень MLOps в оглядах є:

1. Огляд [3] більше фокусується на технічних аспектах MLOps, таких як управління життєвим циклом моделей, автоматизація конвеєрів та моніторинг продуктивності.

2. Огляд [20] розглядає MLOps як набір практик конкретно для операціоналізації рішень науки про дані (data science).

3. Огляд [2] наголошує на використанні в MLOps принципів інженерії програмного забезпечення та машинного навчання для створення продуктів на основі моделей.

4. Огляд [13] розглядає MLOps як окрему сферу, що фокусується на автоматизації життєвого циклу моделей машинного навчання як частини цифрової стратегії компаній.

Отже, 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.

1.4.2. Етапи робочого процесу MLOps

Метою мета-синтезу етапів робочого процесу MLOps в оглядах [2; 3; 13; 20] було виявити спільні та відмінні кроки в життєвому циклі розробки та впровадження моделей машинного навчання.

Спільними етапами робочого процесу MLOps в розглянутих оглядах є:

1. Всі огляди [2; 3; 13; 20] включають етапи збору та обробки даних, розробки та навчання моделей, а також розгортання моделей в робочому середовищі.

2. Огляди [2; 3; 13] виділяють етап моніторингу продуктивності та деградації моделей після розгортання як важливу частину робочого процесу MLOps.

3. Огляди [3; 13] включають етап повторного навчання моделей на основі нових даних або за розкладом як частину життєвого циклу MLOps.

Відмінними аспектами етапів робочого процесу MLOps в оглядах є:

1. Огляд [3] пропонує детальний розподіл етапів робочого процесу MLOps, включаючи кроки вилучення, аналізу, очищення та трансформації даних, а також валідації моделей.

2. Огляд [20] менше фокусується на детальних етапах, а більше на загальних функціях MLOps, таких як збір даних, трансформація, навчання та впровадження моделей.

3. Огляд [2] групує етапи в ширші категорії, такі як управління даними, розподілене навчання, розгортання та моніторинг.

4. Огляд [13] додатково виділяє етапи генерування прогнозів та управління моделями й даними як частину робочого процесу MLOps.

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

1.4.3. Фреймворки та архітектури, що сприяють впровадженню MLOps

Метою мета-синтезу фреймворків та архітектур, що сприяють впровадженню MLOps, в оглядах [2; 3; 13; 20] було визначити найбільш поширені та ефективні підходи і технології в цій сфері.

Спільними фреймворками та архітектурами, що сприяють впровадженню MLOps, згідно з розглянутими оглядами, є:

1. Огляди [2; 3; 20] виділяють платформи та фреймворки з відкритим кодом, такі як MLflow, Kubeflow та TensorFlow Extended (TFX), як ключові компоненти екосистеми MLOps.

2. Огляди [2; 3; 13] підкреслюють важливість використання хмарних обчислювальних платформ та сервісів, таких як AWS, Google Cloud та Azure, для розгортання та масштабування рішень MLOps.

3. Огляди [2; 20] зазначають, що архітектури, засновані на контейнеризації (наприклад, з використанням Docker) та оркестрації контейнерів (наприклад, за допомогою Kubernetes), є ключовими для забезпечення переносимості та масштабованості рішень MLOps.

Відмінними аспектами розглянутих фреймворків та архітектур MLOps в оглядах є:

1. Огляд [3] додатково виділяє платформи для оркестрації конвеєрів MLOps, такі як Apache Airflow, Jenkins та Polyaxon.

2. Огляд [20] згадує специфічні фреймворки та платформи, такі як Kafka-ML та MLModelCI, які використовуються для управління життєвим циклом моделей ML.

3. Огляд [2] розглядає ширший спектр архітектурних підходів, включаючи використання периферійних обчислень (edge computing), безсерверних обчислень (serverless) та архітектур, керованих подіями (event-driven architectures).

4. Огляд [13] фокусується переважно на пропрієтарних платформах та рішеннях від комерційних постачальників, таких як Iguazio, Domino Data Lab, Comet та Valohai.

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

1.4.4. Інструменти MLOps для створення конвеєрів машинного навчання та операціоналізації моделей

Метою мета-синтезу інструментів MLOps для створення конвеєрів машинного навчання та операціоналізації моделей в оглядах [2; 3; 13; 20] було виявити найбільш популярні та функціональні інструменти в цій сфері.

Спільними інструментами MLOps, згаданими в розглянутих оглядах, є:

1. Огляди [3; 20] виділяють MLflow як популярну платформу з відкритим кодом для управління життєвим циклом моделей машинного навчання, експериментами та розгортанням.

2. Огляди [3; 13] згадують хмарні платформи від великих провайдерів, таких як AWS SageMaker, Google Cloud AI Platform, Azure Machine Learning, як інструменти для операціоналізації моделей.

3. Огляди [2; 20] зазначають, що для розгортання моделей часто використовуються інструменти контейнеризації, такі як Docker, та оркестрації, такі як Kubernetes.

Відмінними аспектами розглянутих інструментів MLOps в оглядах є:

1. Огляд [3] надає детальний перелік інструментів для різних етапів конвеєра MLOps, включаючи платформи оркестрації (Apache Airflow, Jenkins, Kubeflow, Polyaxon, Seldon Core та ін.) та розгортання (TensorFlow Extended).

2. Огляд [20] додатково згадує такі інструменти, як Kubeflow, Polyaxon, Comet.ml, Kafka-ML, MLModelCI для управління конвеєрами та розгортання моделей.

3. Огляд [2] більше фокусується на загальних категоріях інструментів, таких як системи управління експериментами, версіонування даних і моделей, автоматизація інфраструктури.

4. Огляд [13] деталізує функціональність популярних комерційних платформ MLOps, таких як Iguazio, Domino Data Lab, Comet, Valohai та ін. (табл. 1.2)

Таблиця 1.2

Популярні платформи та продукти MLOps, а також пов'язані з ними постачальники (на основі [13, с. 141]).

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

1.4.5. Основні функції, що надаються інструментами MLOps

Метою мета-синтезу основних функцій, що надаються інструментами MLOps, в оглядах [2; 3; 13; 20], було визначити ключові можливості та компоненти цих інструментів.

Спільними функціями інструментів MLOps, виділеними в розглянутих оглядах, є:

1. Огляди [2; 3; 20] зазначають, що інструменти MLOps зазвичай надають можливості для відстеження експериментів, версіонування моделей та даних.

2. Огляди [3; 13; 20] підкреслюють важливість функцій автоматизації та оркестрації робочих процесів MLOps, таких як конвеєри навчання та розгортання моделей.

3. Огляди [2; 3; 13] вказують на наявність в інструментах MLOps компонентів для моніторингу продуктивності та деградації розгорнутих моделей.

Відмінними аспектами функцій інструментів MLOps, розглянутих в оглядах, є:

1. Огляд [3] надає детальну класифікацію функцій на три категорії:

    а) *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 (CI/CD);
    - інтерфейси користувача: графічний (GUI), командного рядка (CLI), програмний (API);

    б) *функції управління даними*:

    - передача даних у реальному часі;

- зберігання даних;
- аналіз, очищення та трансформація даних;
- моніторинг даних;
- управління метаданими;
- забезпечення доступу до даних через API;

в) *функції управління моделями*:

- підтримка різних бібліотек та фреймворків машинного навчання;
- відстеження експериментів та версіонування моделей
- реєстр моделей;
- автоматична оптимізація гіперпараметрів;
- тестування моделей (А/Б-тестування);
- виявлення аномалій та дрейфу моделей;
- моніторинг продуктивності моделей;
- управління метаданими моделей;
- розгортання моделей через API.

2. Огляд [20] додатково виділяє такі функції, як автоматична оптимізація гіперпараметрів моделей та підтримка мобільності для розгортання в різних середовищах.

3. Огляд [2] зазначає важливість інтеграції інструментів MLOps з існуючими системами та підтримки колаборативної роботи команд.

4. Огляд [13] детально описує функції комерційних платформ MLOps:

   - *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;
- *генерація 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, такі як оптимізація гіперпараметрів або управління даними. Вибір інструменту з відповідним набором функцій залежить від конкретних потреб і цілей організації в області MLOps.

1.4.6. Способи розгортання моделей машинного навчання у виробничих середовищах

Метою мета-синтезу способів розгортання моделей машинного навчання у виробничих середовищах в оглядах [2; 3; 13; 20] було визначити найбільш поширені підходи та практики в цій сфері.

Спільними способами розгортання моделей машинного навчання у виробничих середовищах, згідно з розглянутими оглядами, є:

1. Огляди [2; 3; 13] зазначають, що моделі часто розгортаються з використанням контейнерних технологій, таких як Docker, що забезпечує мобільність та ізоляцію моделей.

2. Огляди [3; 13; 20] вказують на поширеність розгортання моделей у хмарних середовищах з використанням платформ і сервісів від основних провайдерів, таких як AWS, Google Cloud та Azure.

3. Огляди [2; 3] зазначають, що моделі часто розгортаються як вебсервіси з використанням REST API або інших протоколів для забезпечення доступу до прогнозів у режимі реального часу.

Відмінними аспектами розглянутих способів розгортання моделей в оглядах є:

1. Огляд [3] додатково описує розгортання моделей з використанням платформ оркестрації (Apache Airflow, Jenkins, Kubeflow, MLflow, Polyaxon, Seldon Core, Valohai) для забезпечення автоматичного масштабування та управління контейнерами.

2. Огляд [20] зазначає, що деякі інструменти MLOps, такі як MLflow, Kubeflow та Kafka-ML, мають вбудовані можливості для полегшення розгортання моделей у різних середовищах.

3. Огляд [2] розглядає розгортання моделей не лише в хмарі, але й на периферійних пристроях з використанням спеціальних фреймворків, таких як TensorFlow Lite та Core ML.

4. Огляд [13] надає детальний опис основних етапів та особливості процесу розгортання моделей машинного навчання з використанням конвеєрів CI/CD та підтримкою різних середовищ на прикладі комерційних платформ MLOps:

    а) *створення конвеєру CI/CD для моделей*: платформи MLOps, такі як SageMaker, Azure ML та Databricks, дозволяють створювати конвеєри CI/CD для автоматизації процесу розгортання моделей, які включають етапи збирання, тестування та розгортання моделей, а також відстеження артефактів та управління версіями;

    б) *підтримка різних середовищ розгортання*: платформи MLOps зазвичай підтримують кілька середовищ розгортання, таких як середовища розробки, тестування та виробничі середовища; моделі можуть бути розгорнуті в різних середовищах з використанням відповідних конфігурацій та політик доступу;

    в) *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 (при розгортанні у виробничому середовищі можуть застосовуватись додаткові заходи безпеки та моніторингу);

г) *автоматизація та оркестрація розгортання*: платформи MLOps використовують інструменти автоматизації, такі як Jenkins або GitLab CI/CD, для забезпечення безперервної інтеграції та розгортання моделей, яке може бути налаштоване на автоматичний запуск при певних подіях, таких як оновлення коду моделі або поява нових даних;

д) *моніторинг та управління розгорнутими моделями*: платформи MLOps надають інструменти для моніторингу продуктивності та метрик розгорнутих моделей в режимі реального часу: у разі виявлення проблем або деградації моделі платформа може автоматично ініціювати процес перенавчання або відкату до попередньої версії моделі.

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

1.4.7. Моделі зрілості для оцінки рівня автоматизації розгортання моделей машинного навчання

Lima, Monteiro, Furtado [20] автори посилаються на кілька моделей зрілості для оцінки рівня покращення процесу розробки рішень машинного навчання:

1. Модель зрілості, запропонована Amershi та ін. [31], згадується одночасно у [20] та [2]. Дана модель, заснована на Capability Maturity Model (CMM) та методиці Six Sigma, перевіряє, чи діяльність: (1) має визначені цілі, (2) послідовно реалізована, (3) задокументована, (4) автоматизована, (5) вимірюється і відстежується, і (6) постійно вдосконалюється.

2. Згідно з Dhanorkar та ін. [37], організації можуть бути класифіковані за трьома рівнями зрілості розробки рішень машинного навчання: (1) орієнтовані на дані, (2) орієнтовані на модель, (3) орієнтовані на конвеєр.

3. Lwakatare, Crnkovic, Bosch [21] описують п'ять етапів вдосконалення практик розробки: (1) ручний процес, керований наукою про дані, (2) стандартизований процес експериментально-операційної симетрії, (3) автоматизований процес робочого процесу ML, (4) інтегровані процеси розробки програмного забезпечення та робочого процесу ML, і (5) автоматизований і повністю інтегрований процес робочого процесу CD і ML.

4. Akkiraju та ін. [10] запропоновано адаптацію моделі CMM з визначенням п'яти рівнів зрілості для кожної оціненої здатності: (1) початковий, (2) повторюваний, (3) визначений, (4) керований і (5) оптимізуючий.

Усі систематичні огляди [2; 3; 20] вказують, що рівень автоматизації процесів MLOps є одним з ключових факторів при оцінці зрілості організації в цій сфері. Незважаючи на те, що розглянуті огляди не надають вичерпного опису моделей зрілості MLOps, вони підкреслюють важливість оцінки рівня автоматизації процесів розробки, тестування та розгортання моделей як ключового фактора зрілості організації в цій сфері. Адаптація існуючих моделей зрілості розробки програмного забезпечення до специфіки MLOps може бути ефективним підходом до оцінки та вдосконалення процесів машинного навчання в організації.

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

Метою мета-синтезу ролей та обов'язків, визначених в діяльності з операціоналізації моделей машинного навчання, в оглядах [2; 3; 13; 20] було визначити ключових учасників процесу MLOps та їхні функції.

Спільні ролі та обов'язки, визначені в розглянутих оглядах:

1. Всі огляди [2; 3; 13; 20] згадують про залучення фахівців з даних / наукових співробітників з даних, які відповідають за розробку, навчання та експериментування з моделями машинного навчання.

2. Огляди [2; 3; 20] виділяють роль інженерів з даних / провайдерів даних, які займаються вилученням, обробкою, перетворенням та забезпеченням якості даних для навчання моделей.

3. Огляди [2; 3; 13] зазначають важливість інженерів DevOps, інженерів ML/MLOps та інженерів програмного забезпечення в операціоналізації моделей, автоматизації процесів розгортання, створенні конвеєрів та управлінні середовищами.

4. Огляди [2; 3] підкреслюють роль менеджерів, керівництва та бізнес-зацікавлених сторін у визначенні вимог до моделей щодо їх розгортання, прийнятті рішень та підтримці MLOps стратегії.

5. Огляди [2; 3] 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, розглянуті в оглядах:

1. Огляд [20] додатково виділяє ролі:

   - *фахівець з предметної галузі* (domain specialist) має глибокі знання предметної галузі, відіграє важливу роль в отриманні даних та валідації результатів);
   - *науковець/інженер з обчислювальних наук* (computational scientist/engineer) має високі технічні навички для підготовки середовища для роботи моделей машинного навчання;

- *науковець/інженер з машинного навчання* (ML scientist/engineer) відповідає за проєктування нових моделей машинного навчання, має поглиблені знання статистики та алгоритмів ML;
- *провайдер* (provenance specialist) керує постачанням даних у життєвому циклі розробки рішень машинного навчання, має знання як предметної галузі, так і машинного навчання;
- *менеджер* (manager) оцінює моделі перед їх публікацією;
- *розробник додатків* (application developer) розробляє додатки, в яких будуть працювати створені моделі;
- *керівник з розгортання* (deployment lead) оцінює аспекти, пов'язані з компонентами інфраструктури при розгортанні моделей ML у виробництво.

2. Огляд [2] згадує роль експертів предметної галузі в розміченні даних у специфічних доменах.

Отже, незважаючи на деякі відмінності в деталізації ролей, розглянуті огляди визнають необхідність залучення фахівців з різних сфер – 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. 1.1).

1.4.9. Виклики, що виникають при розгортанні моделей машинного навчання у виробничих середовищах

Метою мета-синтезу викликів, що виникають при розгортанні моделей машинного навчання у виробничих середовищах, в оглядах [2; 3; 13; 20] було визначити найбільш поширені та критичні проблеми в цій сфері. У [3] та [13] явно не наводяться конкретні виклики, однак їх можна визначити опосередковано, базуючись на обговоренні MLOps та автоматизації конвеєрів машинного навчання у [3] та описі різних етапів MLOps і необхідності у відповідних інструментах у [13].

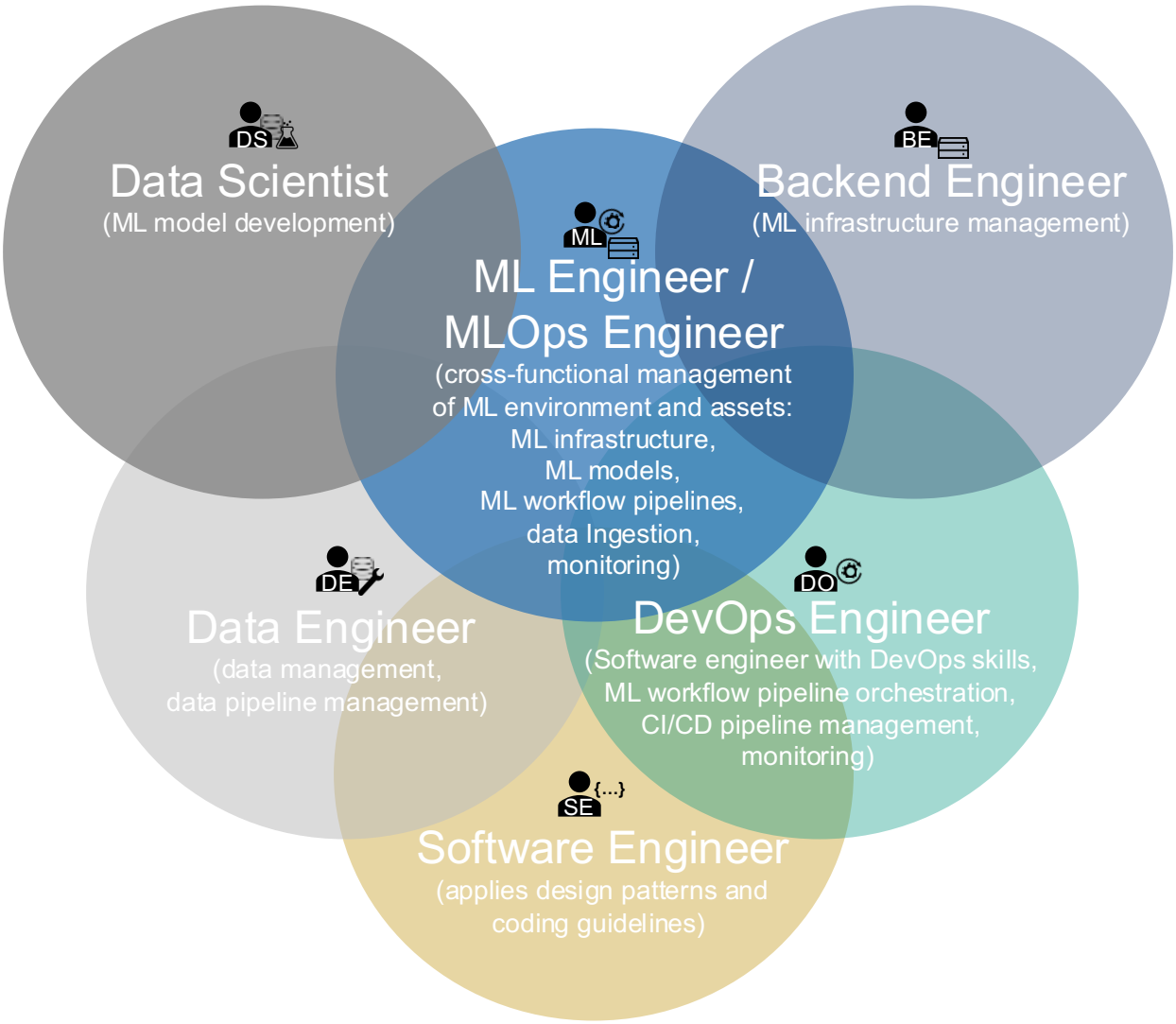

Рис. 1.1. Перетин ролей та обов'язків (за [19, с. 5]).

Спільні виклики, визначені в розглянутих оглядах:

1. Огляди [2; 3; 13; 20] зазначають складність управління життєвим циклом моделей машинного навчання, включаючи 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. Огляди [2; 13; 20] вказують на виклики, пов'язані з моніторингом та підтримкою моделей в виробничому середовищі, включаючи виявлення дрейфу даних та деградації продуктивності моделей.

3. Огляди [3; 20] розглядають виклики, пов'язані з інтеграцією розробки програмного забезпечення із конвеєром машинного навчання.

4. Огляди [2; 3] розглядають виклики, пов'язані з забезпеченням безпеки та конфіденційності даних при розгортанні моделей машинного навчання.

5. Огляди [2; 13] вказують, що якість, доступність, підготовка, маркування та інтеграція даних з різних джерел є значним викликом, що потребує багато часу та ресурсів, та виділяють проблему інтерпретації й пояснення результатів роботи моделей для кінцевих користувачів та бізнес-стейкхолдерів.

Відмінні виклики, розглянуті в оглядах:

1. Огляд [3] підкреслює необхідність автоматизації всіх етапів конвеєру MLOps та інтеграції з існуючими системами і процесами розробки програмного забезпечення.

2. Огляд [20] зазначає проблему вибору та управління інфраструктурою для розгортання моделей, включаючи вибір між хмарними та локальними середовищами.

3. Огляд [2] зазначає проблеми: а) розриву між інженерією програмного забезпечення та навичками машинного навчання – науковці з даних часто не розуміють вимоги певних виробничих середовищ, а розробники програмного забезпечення не мають достатніх навичок машинного навчання; б) ефективного розподілу, паралелізації та оркестрація даних і завдань ML; в) різноманітність обчислювальної інфраструктури.

Розглянуті огляди демонструють, що розгортання моделей машинного навчання у виробничих середовищах пов'язане з низкою викликів, таких як управління життєвим циклом моделей, забезпечення масштабованості та продуктивності, моніторинг та підтримка моделей в 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.

1.4.10. Відкриті питання, виклики та особливості MLOps

Метою мета-синтезу відкритих питань, викликів та особливостей MLOps в оглядах [2; 3; 13; 20] було визначити найбільш актуальні та перспективні напрямки досліджень та розробок у цій галузі. У [3] та [13] безпосередньо не обговорюються відкриті проблеми, виклики та особливості MLOps, однак їх можна виокремити на основі аналізу інструментів MLOps та їх можливостей у [3] та опису компонентів та функцій платформ MLOps [13].

Спільні відкриті питання та виклики MLOps, визначені в розглянутих оглядах:

1. Огляди [2; 3; 13; 20] вказують на потребу в розробці методів та інструментів для забезпечення інтерпретовності, відтворюваності та відповідального використання моделей машинного навчання в контексті MLOps.

2. Огляди [2; 3; 13] підкреслюють важливість розробки підходів до управління даними в MLOps, включаючи забезпечення якості, конфіденційності та безпеки даних.

3. Огляди [3; 20] відзначають необхідність розробки та впровадження стандартів та кращих практик MLOps для забезпечення узгодженості та сумісності між різними інструментами та платформами.

4. Огляд [2; 20] підкреслює важливість людського фактору в MLOps, включаючи необхідність забезпечення ефективної комунікації та співпраці між різними ролями й командами та підготовку кваліфікованих кадрів із крос-функціональними навичками програмування, обробки даних та операційної діяльності.

Особливості MLOps, визначені в розглянутих оглядах:

1. Огляд [3] зазначає, що MLOps має враховувати специфіку процесу розробки моделей машинного навчання, який відрізняється від традиційної розробки програмного забезпечення.

2. Огляд [13] розглядає MLOps в контексті загальної цифрової стратегії організації та підкреслює необхідність узгодження практик MLOps з бізнес-цілями та потребами.

Отже, розглянуті огляди визначають низку відкритих питань та викликів в MLOps, таких як необхідність розробки стандартів та кращих практик, забезпечення інтерпретовності та відповідального використання моделей, а також ефективне управління даними. Особливості MLOps, такі як відмінність від традиційної розробки ПЗ, важливість людського фактору та необхідність інтеграції знань з різних галузей, вимагають врахування при впровадженні практик MLOps в організаціях.

1.4.11. Можливості, майбутні тенденції та сфери застосування MLOps

Метою мета-синтезу можливостей, майбутніх тенденцій та сфер застосування MLOps в оглядах [2; 3; 20] було визначити перспективні напрямки розвитку та потенційні області, в яких практики MLOps можуть принести значну користь. У [3] вони безпосередньо не обговорюються, однак їх можна визначити опосередковано, базуючись на представлених інструментах MLOps та їх можливостях, можна окреслити деякі потенційні напрямки та тренди.

Можливості та майбутні тенденції MLOps, визначені в розглянутих оглядах:

1. Огляди [2; 3; 20] відзначають потенціал розвитку стандартизованих платформ та інструментів MLOps, які дозволять спростити та прискорити впровадження моделей машинного навчання в виробництво.

2. Огляди [2; 20] відзначають перспективи інтеграції MLOps з іншими підходами, такими як DataOps, ModelOps та DevSecOps, для забезпечення комплексного управління життєвим циклом моделей машинного навчання.

3. Огляд [20] вказує 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, в міру розвитку галузі.

4. Огляд [2] вказує на: а) можливості застосування практик MLOps в контексті розподіленого та федеративного навчання моделей, що дозволить ефективно використовувати децентралізовані дані; б) залучення бізнес-підрозділів та навчання керівництва принципів MLOps; в) використання таких апаратних платформ, як FPGA та IoT, для покращення продуктивності та конфіденційності

Поточні та майбутні сфери застосування MLOps, визначені в розглянутих оглядах:

1. Огляди [2; 3; 20] відзначають, що MLOps вже активно застосовується в таких галузях, як фінанси, охорона здоров'я, торгівля, маркетинг та виробництво, де моделі машинного навчання використовуються для вирішення реальних бізнес-задач.

2. Огляд [2] вказують на потенціал застосування MLOps в сфері IoT та периферійних обчислень, де моделі машинного навчання можуть бути розгорнуті на пристроях з обмеженими ресурсами, технологій 5G та 6G, навчальній та науковій діяльності.

3. Огляд [20] відзначають перспективи використання MLOps у транспорті та логістиці.

Таким чином, розглянуті огляди окреслюють низку можливостей та тенденцій розвитку MLOps, таких як 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.

Висновки до 1 розділу

У даному розділі було виконано мета-синтез систематичних оглядів [2; 3; 20] та огляду продуктів та постачальників [13] з метою узагальнення

знань щодо впровадження практик MLOps для ефективного розгортання моделей машинного навчання. Основні висновки, отримані в результаті виконання мета-синтезу, є такими:

1. MLOps – підхід для управління, автоматизації та операціоналізації процесів розробки, розгортання та підтримки моделей машинного навчання на основі практик з інженерії програмного забезпечення та DevOps. MLOps базується на наборі принципів, процесів та практик, які забезпечують ефективне розроблення, розгортання та підтримку моделей машинного навчання.

2. Основні етапи життєвого циклу MLOps включають такі процеси: збір та обробку даних, розробку та навчання моделей, розгортання, моніторинг та повторне навчання моделей.

3. Для впровадження MLOps використовуються різні фреймворки та архітектури, такі як платформи з відкритим кодом (MLflow, Kubeflow, TensorFlow Extended), хмарні обчислювальні платформи (AWS, Google Cloud, Azure), контейнеризація (Docker) та оркестрація контейнерів (Kubernetes).

4. Інструменти MLOps надають широкий спектр функцій для підтримки життєвого циклу моделей машинного навчання, з акцентом на такі практики: автоматизація, відстеження експериментів, версіонування, моніторинг та розгортання моделей.

5. Найбільш поширеними способами розгортання моделей машинного навчання у виробничих середовищах є використання контейнерних технологій, хмарних платформ і сервісів, а також розгортання моделей як веб-сервісів.

6. Для оцінки рівня зрілості процесів MLOps в організаціях можуть використовуватися адаптовані моделі зрілості розробки програмного забезпечення, такі як CMM.

7. Успішне впровадження MLOps вимагає залучення фахівців з різних сфер – розробки програмного забезпечення, інженерії даних, машинного навчання, експертів предметної області та менеджменту.

8. Основними викликами при розгортанні моделей машинного навчання у виробничих середовищах є управління життєвим циклом моделей, забезпечення масштабованості та продуктивності, моніторинг та підтримка моделей в реальних умовах використання.

9. Відкритими питаннями та викликами в MLOps є необхідність розробки стандартів та кращих практик, забезпечення інтерпретовності та відповідального використання моделей, ефективне управління даними, інтеграція знань з різних галузей.

10. Основними можливостями та тенденціями розвитку MLOps є 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.

Проведений мета-синтез показав, що MLOps є перспективним підходом для ефективного розгортання моделей машинного навчання у виробничих середовищах, який потребує подальшого дослідження та розвитку для вирішення існуючих викликів та реалізації потенційних можливостей.

РОЗДІЛ 2
АНАЛІЗ ПРАКТИК MLOPS

2.1. Співвідношення принципів, процесів і практик MLOps

MLOps базується на наборі принципів [19, с. 3] та процесів, які забезпечують ефективне розроблення, розгортання та підтримку моделей машинного навчання (практик MLOps).

Принципи MLOps визначають фундаментальні основи проєктування конвейєрів машинного навчання:

- *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;
- *керованість даними*: забезпечення якості, безпеки та конфіденційності даних протягом усього життєвого циклу моделей.

Процеси MLOps визначають порядок дій із проєктування та реалізації конвейєрів машинного навчання:

1. *Визначення бізнес-цілей та вимог*: узгодження цілей розробки моделей машинного навчання з бізнес-стратегією організації.
2. *Збір та підготовка даних*: збір, очищення, трансформація та збагачення даних для навчання моделей.

3. *Розробка та навчання моделей*: вибір алгоритмів, розробка архітектури моделей, навчання та валідація моделей.

4. *Оцінка та тестування моделей*: оцінка продуктивності моделей на тестових даних, проведення тестів на надійність, безпеку та відповідність вимогам.

5. *Розгортання моделей*: пакування моделей з необхідними залежностями, розгортання в цільових середовищах.

6. *Моніторинг та обслуговування моделей*: відстеження продуктивності моделей, виявлення та вирішення проблем, оновлення моделей за потреби.

7. *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 конвейєрів машинного навчання:

- *основні практики MLOps* включають:
  - *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 ML*: 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 забезпечення відповідності вимогам;

- *додаткові практики MLOps* включають:
  - *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 та пояснення поведінки моделей, особливо в регульованих галузях;
  - *управління якістю даних*: моніторинг та забезпечення якості даних, що використовуються для навчання та оцінки моделей, виявлення та обробка аномалій;
  - *управління конфігурацією*: версіонування та управління конфігураціями середовищ, в яких розгортаються моделі, забезпечення узгодженості між різними середовищами;
  - *стратегії розгортання моделей*: вибір та реалізація відповідних 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 та запобігання ризиків, пов'язаних з використанням моделей машинного навчання, забезпечення відповідності регуляторним вимогам.

Рис. 2.1 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, в свою чергу, 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 з такими практиками, як версіонування, автоматизація конвеєрів, управління експериментами та інтерпретовність моделей.

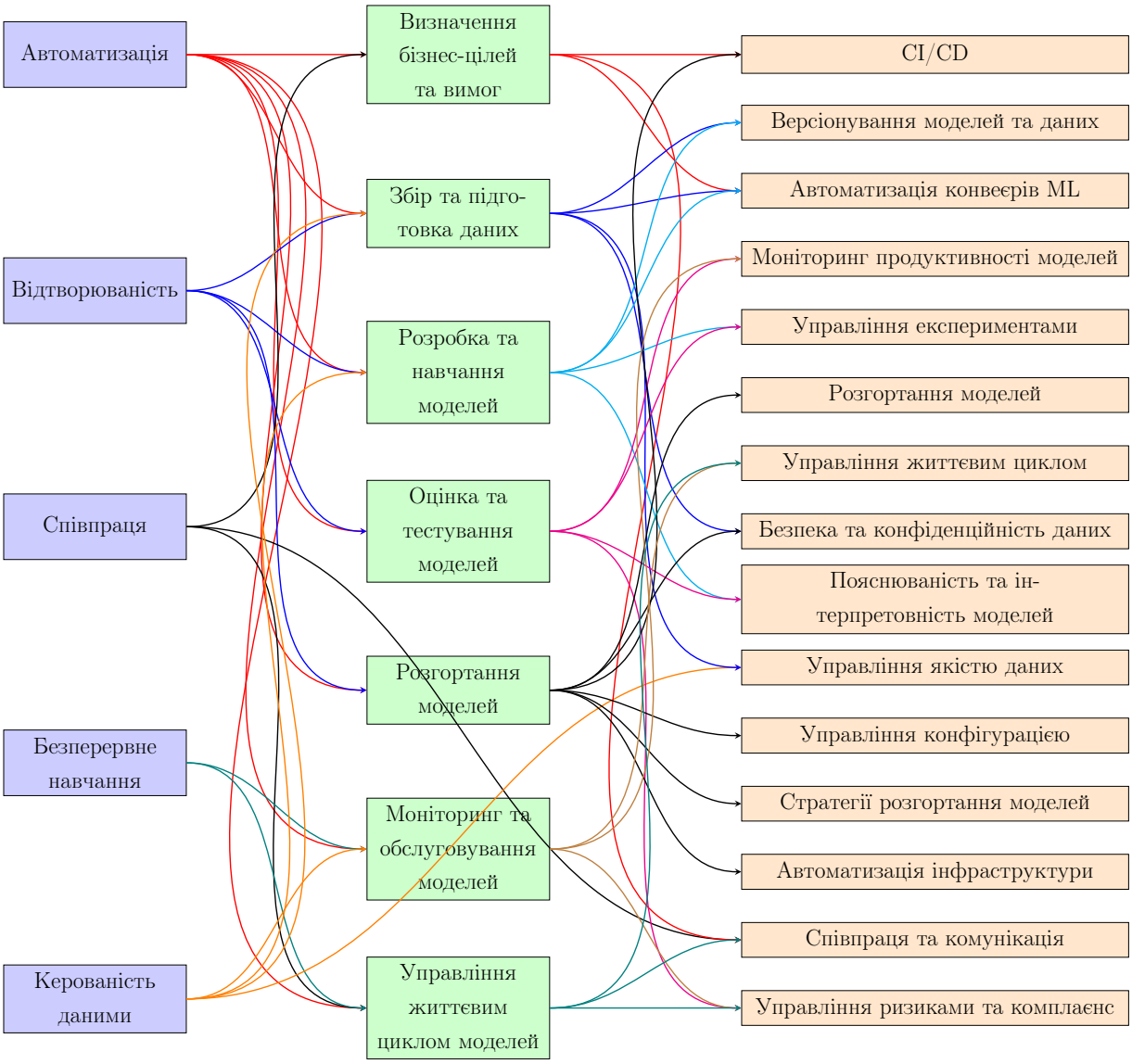

Рис. 2.1. Схема зв'язків між принципами, процесами та практиками MLOps.

2.2. CI/CD

CI/CD (Continuous Integration/Continuous Delivery) – це ключовий елемент/практика/впровадження DevOps для автоматичного тестування та розгортання коду, даних і моделей у виробничому середовищі (рис. 2.2) [33, с. 7]. У MLOps вона розширюється для автоматизації процесу розробки та розгортання моделей ML, включаючи етапи побудови, тестування, доставки та розгортання [19, с. 3-4].

Процес CI/CD у MLOps включає в себе етапи побудови (build), тестування (test), доставки (delivery) та розгортання (deploy) [19, с. 4]. Однак, на відміну від традиційного CI/CD, у MLOps також можуть бути додаткові етапи, такі як перенавчання моделі.

CI/CD у MLOps є частиною архітектури систем MLOps та забезпечує швидкий зворотний зв'язок розробникам щодо успішності чи невдачі певних етапів, підвищуючи загальну продуктивність [19, с. 3-4].

Типовими тригерами запуску процесу CI/CD у MLOps на GitHub є події git push та pull_request [9, с. 4]. Також можуть використовуватися події issue_comment, release та schedule (за розкладом). У статті Steidl, Felderer, Ramler [33] також досліджено потенційні основні тригери, такі як 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: (1) обробка даних, (2) навчання моделі, (3) розробка програмного забезпечення та (4) уведення системи в експлуатацію. Етап обробки даних складається з повторюваного наскрізного життєвого циклу завдань, пов'язаних з даними, таких як попередня обробка, забезпечення якості, версіонування та документування. Етап навчання моделі використовує результати обробки даних та ілюструє завдання, пов'язані з розробкою моделі, такі як проєктування моделі, 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 уведення системи в експлуатацію модель розгортається в конкретному середовищі за допомогою різних стратегій розгортання і здійснюється

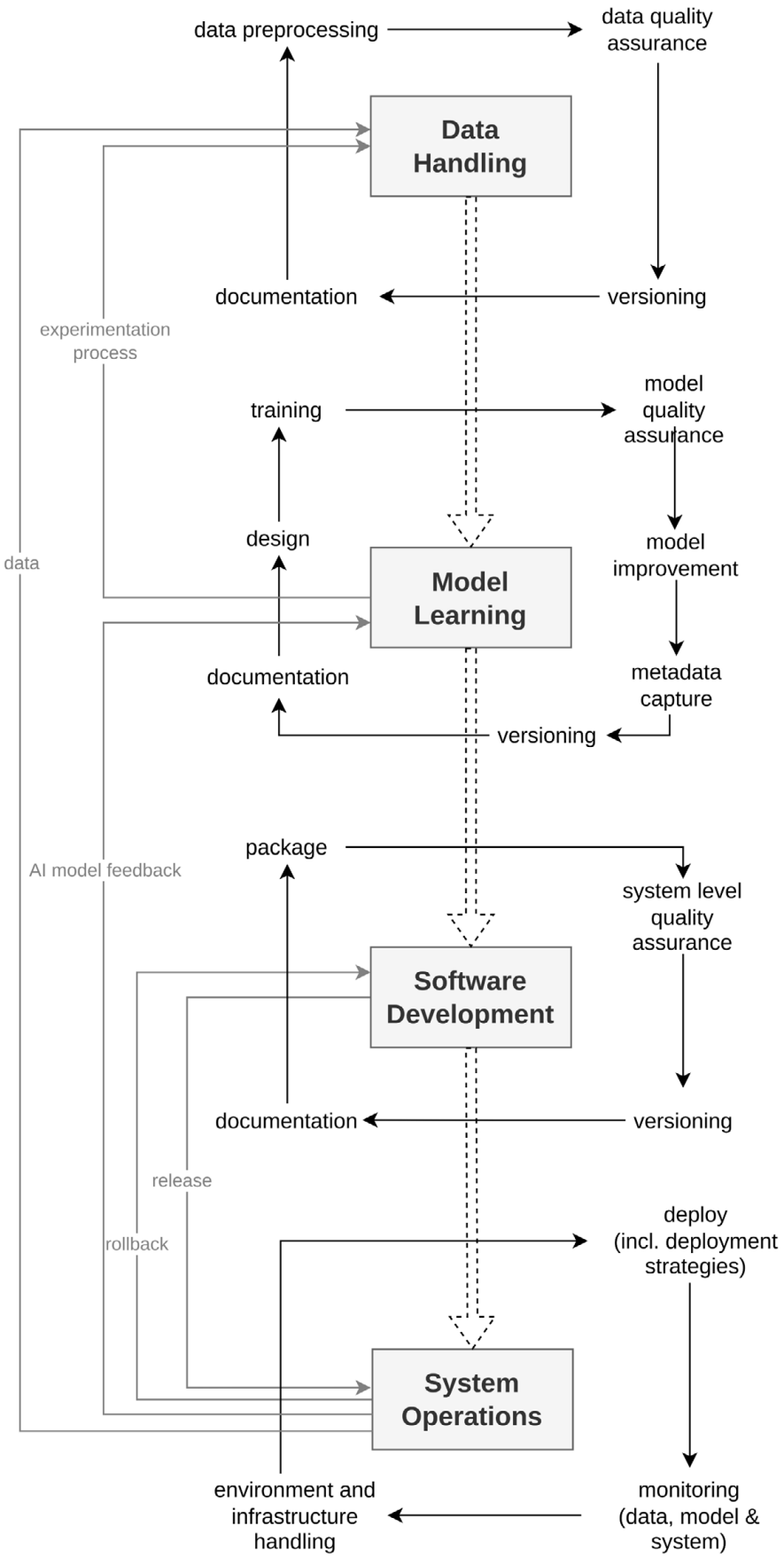

Рис. 2.2. Continuous lifecycle pipeline for AI applications [33, с. 10].

моніторинг системи [33, с. 21].

Особливістю процесу CI/CD у MLOps є необхідність версіонування не лише коду, а й даних та моделей. Це дозволяє забезпечити відтворюваність та можливість відкату до попередніх версій [19, с. 4].

Серед популярних інструментів для реалізації CI/CD у MLOps можна виділити Jenkins [19, с. 3] та GitHub Actions [9; 19] та інструменти від хмарних провайдерів, такі як AWS CodePipeline, Azure DevOps Pipelines тощо.

Рис. 2.3 із статті Kreuzberger, Kühl, Hirschl [19] представляє наскрізну архітектуру та робочий процес MLOps із функціональними компонентами та ролями, задіяними на кожному етапі. Розглянемо детальніше кожну із зон та етапів, зображених на рисунку:

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

2. **Data Engineering Zone**. Ця зона включає два підетапи:

   - **B1. Requirements for feature engineering pipeline**. Визначаються правила для трансформації, очищення та розрахунку нових ознак.
   - **B2. Feature Engineering Pipeline**. Реалізується конвеєр обробки даних, який отримує дані із різних джерел, застосовує трансформації і завантажує у сховище ознак (Feature store system).

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

4. **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).

2.3. Версіонування моделей та даних

Версіонування є однією з ключових практик MLOps, яка забезпечує відтворюваність (reproducibility) та можливість відстеження (traceability) моделей машинного навчання [19, с. 3]. Версіонування охоплює дані, код та самі моделі [33, с. 9]. Версіонування моделі не лише фіксує артефакти моделі версій, але й залежності моделі для відстеження або відтворення різних версій моделі, які швидко змінюються з часом [33, с. 13].

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 впоратись з великими обсягами даних, використовують спеціалізовані інструменти, такі як Data Version Control (DVC) [33, с. 11].

У робочому процесі MLOps версіонування моделей відбувається на етапі навчання моделі. Його метою є збереження різних версій моделей разом з їхніми метаданими для можливості відкату до попередніх версій та відтворення результатів. Умовами використання версіонування моделей є: (1) постійна еволюція моделей з часом; (2) необхідність відстеження залежностей між моделями, даними та кодом [33, с. 12].

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 [33, с. 13].

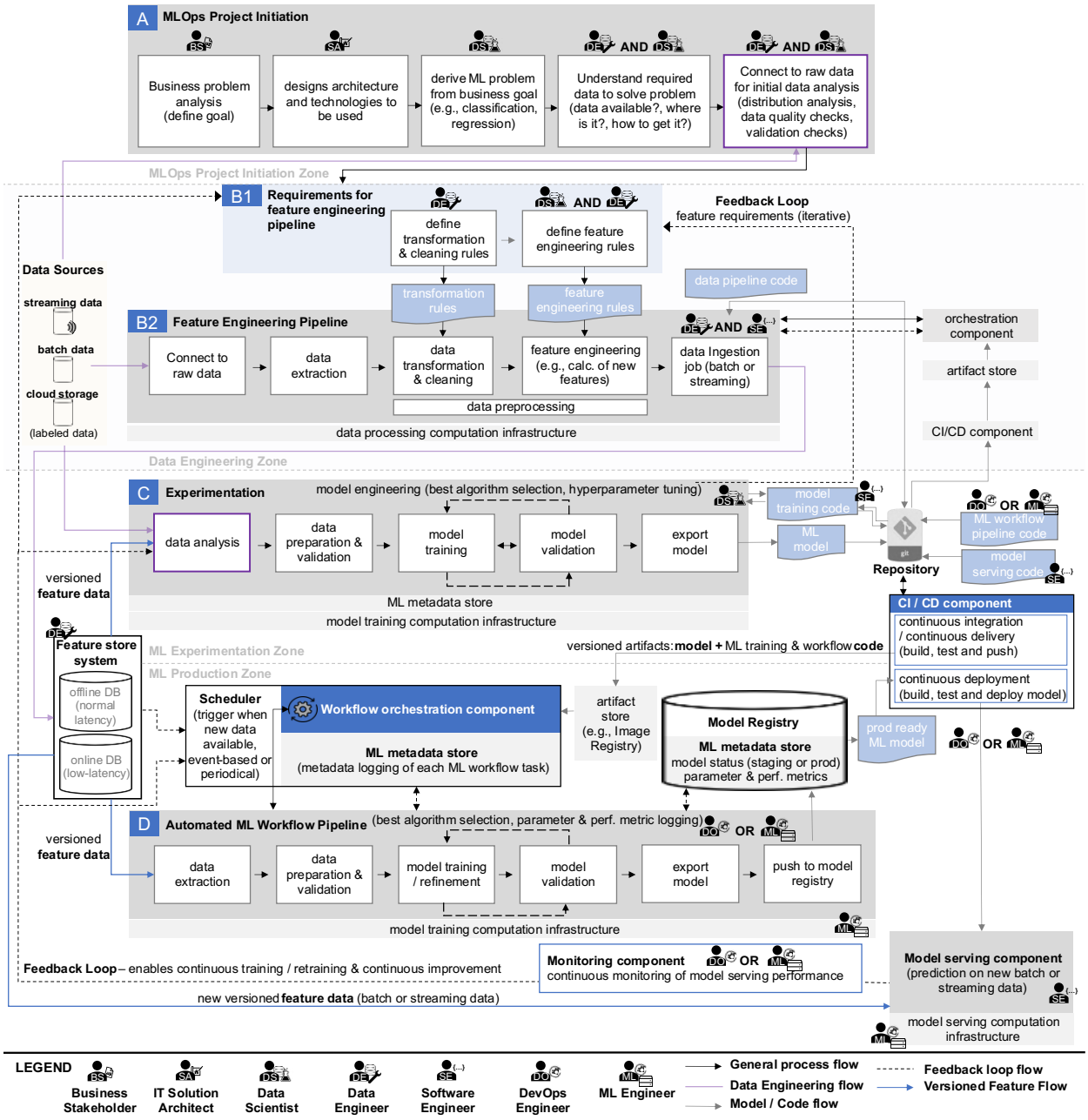

Рис. 2.3. End-to-end MLOps architecture and workflow with functional components and roles [19, с. 6].

Окрім самих даних, версіонуванню підлягають залежності, кроки обробки даних та видобуті ознаки (features) [33, с. 11]. Для останніх часто використовують спеціальні сховища ознак (feature stores).

Залежності моделі фіксують зв'язок із пов'язаними елементами, такими як відповідний набір даних, вихідний код і конфігураційні файли. Крім того, 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, такі як Git, не можуть бути використані як репозиторії моделей. Потенційні альтернативи, такі як MLFlow, H2O і DataRobot - це контейнерні реєстри, де зберігаються версії зображень, або репозиторії моделей, які зберігають версії моделей, включаючи код, метадані, результати тестів і залежності [33, с. 13].

2.4. Автоматизація конвеєрів ML

Автоматизація конвеєрів ML є ключовою практикою MLOps, яка дозволяє спростити та прискорити розробку, тестування та розгортання моделей ML в робочому середовищі. Ця практика охоплює автоматизацію різних етапів конвеєра ML, включаючи збір даних, препроцесінг, розробку моделі, навчання, тестування, валідацію та розгортання [33, с. 2].

Алгоритм роботи автоматизованого конвеєра машинного навчання (рис. 2.4) складається з наступних етапів:

1. Запуск процесу.
2. Видобування версіонованих даних зі сховища.
3. Автоматизована підготовка і валідація даних.
4. Автоматизоване навчання моделі на нових даних (ітеративно).
5. Оцінювання моделі та налаштування гіперпараметрів.
6. Експорт моделі.
7. Збереження моделі в реєстрі моделей.
8. Розгортання моделі.
9. Обслуговування моделі для отримання передбачень.
10. Моніторинг продуктивності моделі.

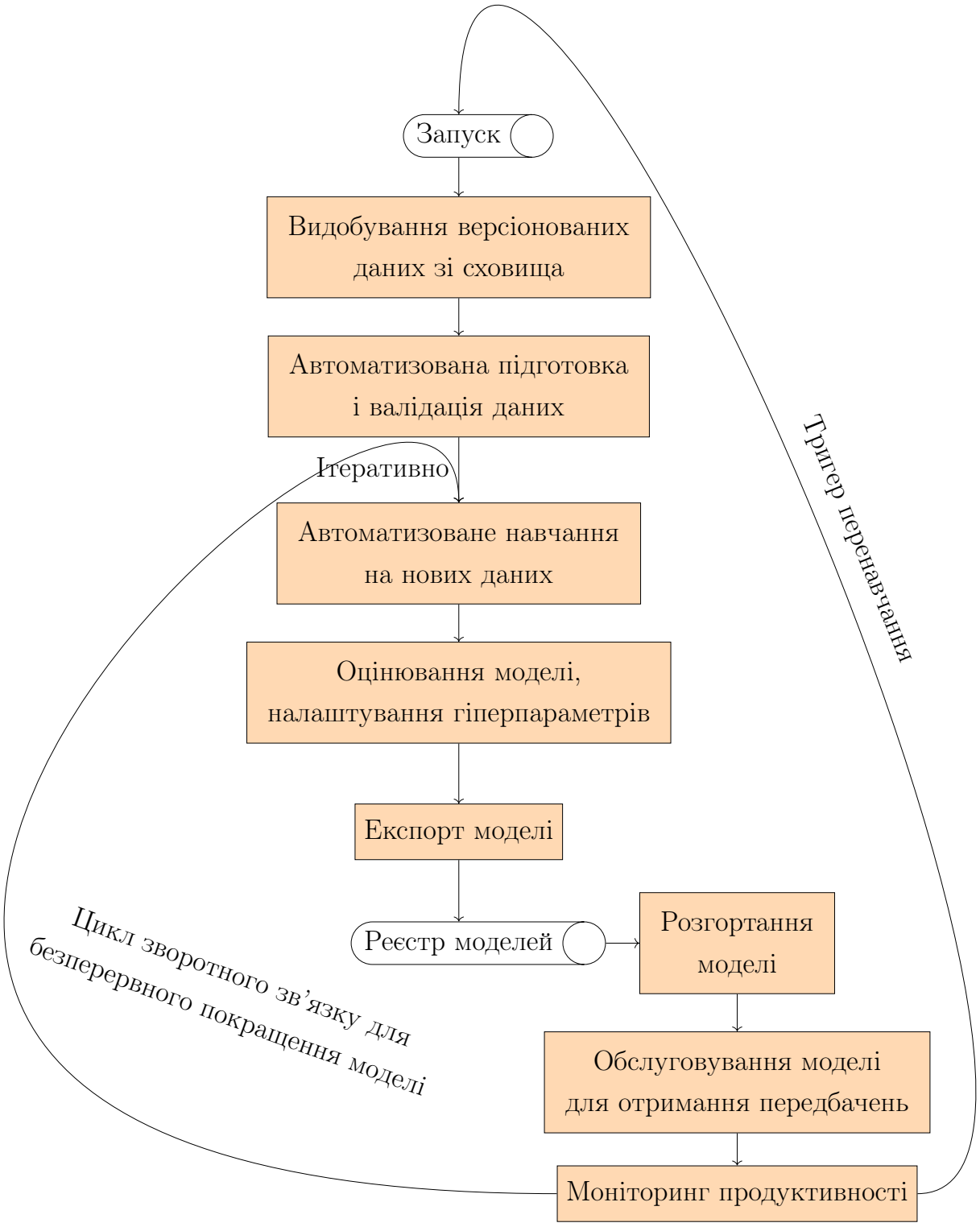

Рис. 2.4. Алгоритм роботи автоматизованого конвеєра машинного навчання (за Kreuzberger, Kühl, Hirschl [19]).

Цикл зворотного зв'язку забезпечує безперервне покращення моделі шляхом повернення до етапу навчання. У разі необхідності, моніторинг може ініціювати тригер перенавчання моделі, який запускає процес спочатку. Цей автоматизований конвеєр надає можливість ефективно керувати жит-

тєвим циклом моделі машинного навчання, від підготовки даних до розгортання та моніторингу, забезпечуючи безперервне вдосконалення моделі. Особливостями використання автоматизації конвеєрів ML є необхідність враховувати гетерогенність моделей, фреймворків та середовищ виконання. Тому кожен крок конвеєра ML має бути максимально ізольованим та мати чіткі інтерфейси, наприклад, через використання контейнеризації [27, с. 6]. Також критично важливим є забезпечення можливості відтворення результатів та відслідковуваності артефактів [6, с. 1706].

Способи автоматизації конвеєрів ML включають використання систем керування конвеєрами, таких як Kubeflow, TFX [33, с. 19] або Apache Airflow [3, с. 4], а також розробку власних скриптів автоматизації з використанням таких інструментів, як Jenkins [34] або GitLab CI/CD [19, с. 2]. При цьому конвеєр розбивається на окремі кроки, кожен з яких реалізується як код або конфігурація, і далі ці кроки оркеструються та виконуються автоматично [6, с. 1706].

2.5. Моніторинг продуктивності моделей

Моніторинг продуктивності моделей машинного навчання є важливою практикою MLOps, яка дозволяє 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 в реальному часі та оповіщення у разі виявлення відхилень від норми [7, с. 128].

Моніторинг продуктивності моделей відбувається на етапі експлуатації моделі, після її розгортання у виробничому середовищі. Він є частиною неперервного циклу MLOps і виконується паралельно з іншими практиками, такими як розробка, тестування та розгортання [7, с. 127].

Умовами використання моніторингу є наявність розгорнутої моделі ML, яка обробляє реальні дані та генерує передбачення. Крім того, повинна бути налаштована інфраструктура для збору та зберігання даних моніторингу, наприклад, база даних та інструменти візуалізації [7, с. 134].

Процес моніторингу включає кілька кроків (рис. 2.5). Спочатку визначаються ключові метрики продуктивності моделі, такі як точність, помилка, латентність тощо. Потім налаштовуються інструменти для збору

цих метрик з системи, де розгорнута модель. Далі дані агрегуються та візуалізуються на інформаційних панелях для зручного аналізу. Нарешті, налаштовуються оповіщення, які спрацьовують при виході метрик за допустимі межі [7, с. 129].

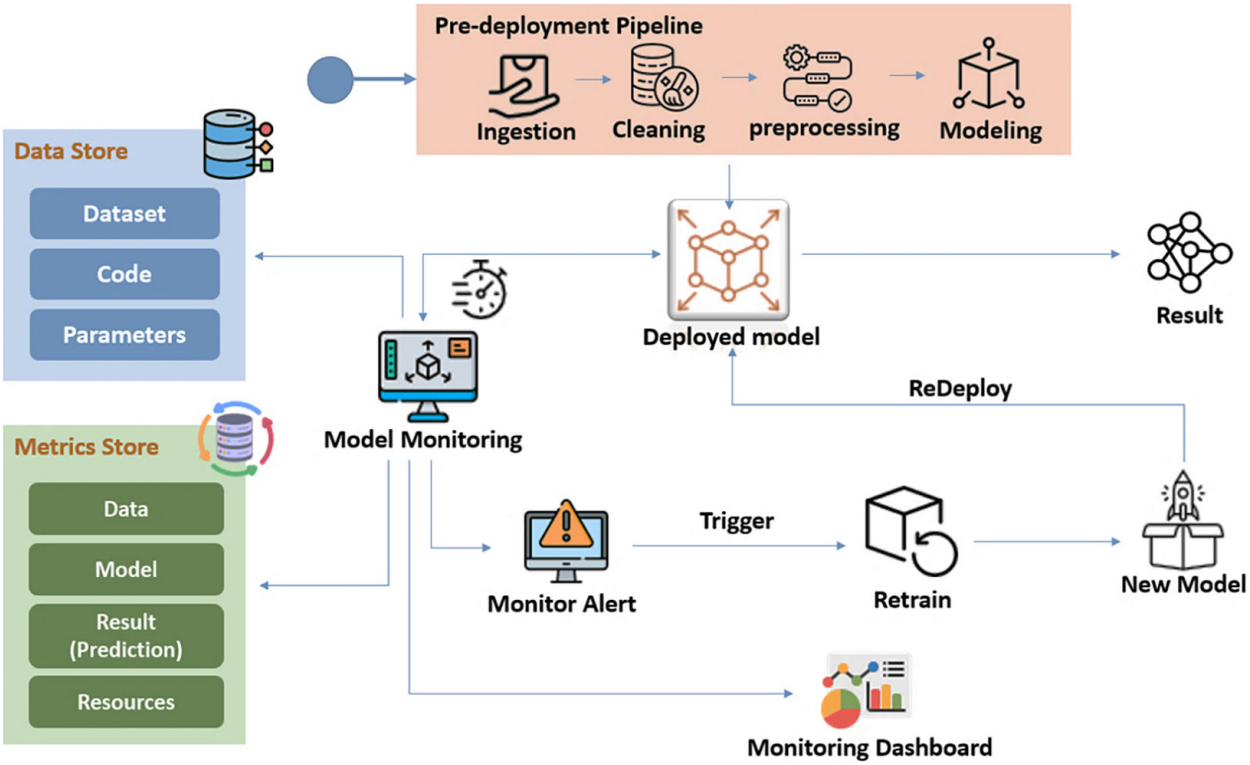

Рис. 2.5. Процес моніторингу у MLOps [7, с. 129].

Рис. 2.6 зображує запропоновану Bodor, Hnida, Daoudi [7] аналогію між моніторингом системи машинного навчання та айсбергом. Ця аналогія підкреслює, що здоров'я ML системи залежить не тільки від видимих елементів, таких як надані сервіси (Service), але й від прихованих особливостей, що складно відслідковувати, таких як дані (Data) та сама модель (Model):

- Верхній рівень айсберга представляє здоров'я екосистеми (Ecosystem Health), що включає такі аспекти, як прогнозний дрейф (Prediction Drift) та бізнес-показники ефективності (Business KPI). Метою є оцінка продуктивності, надійності та стабільності цього верхнього рівня.

- Рівень сервісів включає такі метрики, як латентність (Latency), вартість (Cost) та загальну продуктивність системи (System Performance).

- Рівень даних містить характеристики якості даних (Data Quality), викиди (Outlier Value) та дрейф даних (Data Drift).
- Найнижчий рівень айсберга – модель, яка характеризується точністю (Model Accuracy), дрейфом концепції (Concept Drift) та зміщенням моделі (Model Bias).

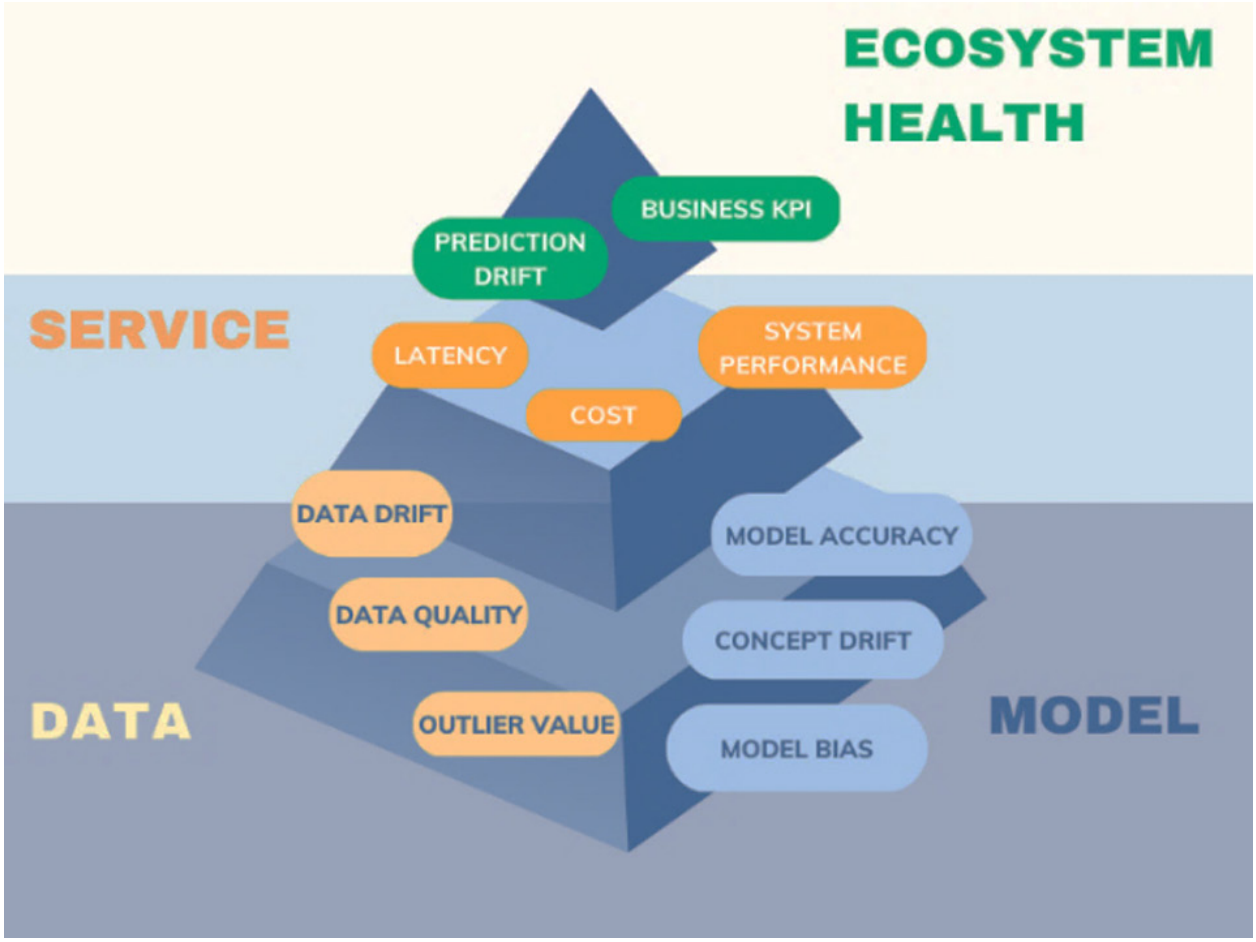

Рис. 2.6. Приховані та видимі характеристики, пов'язані з моніторингом [7, с. 130].

Особливістю моніторингу в контексті MLOps є необхідність відстежувати не лише традиційні метрики продуктивності програмного забезпечення (наприклад, використання процесора та пам'яті), але й метрики, специфічні для ML, такі як точність моделі на нових даних (рис. 2.7). Крім того, MLOps передбачає автоматизацію процесу моніторингу та інтеграцію його в загальний конвеєр розробки та розгортання моделей [7, с. 128].

Існує ряд інструментів для налаштування моніторингу продуктивності моделей ML. Вони включають платформи з відкритим кодом, такі як

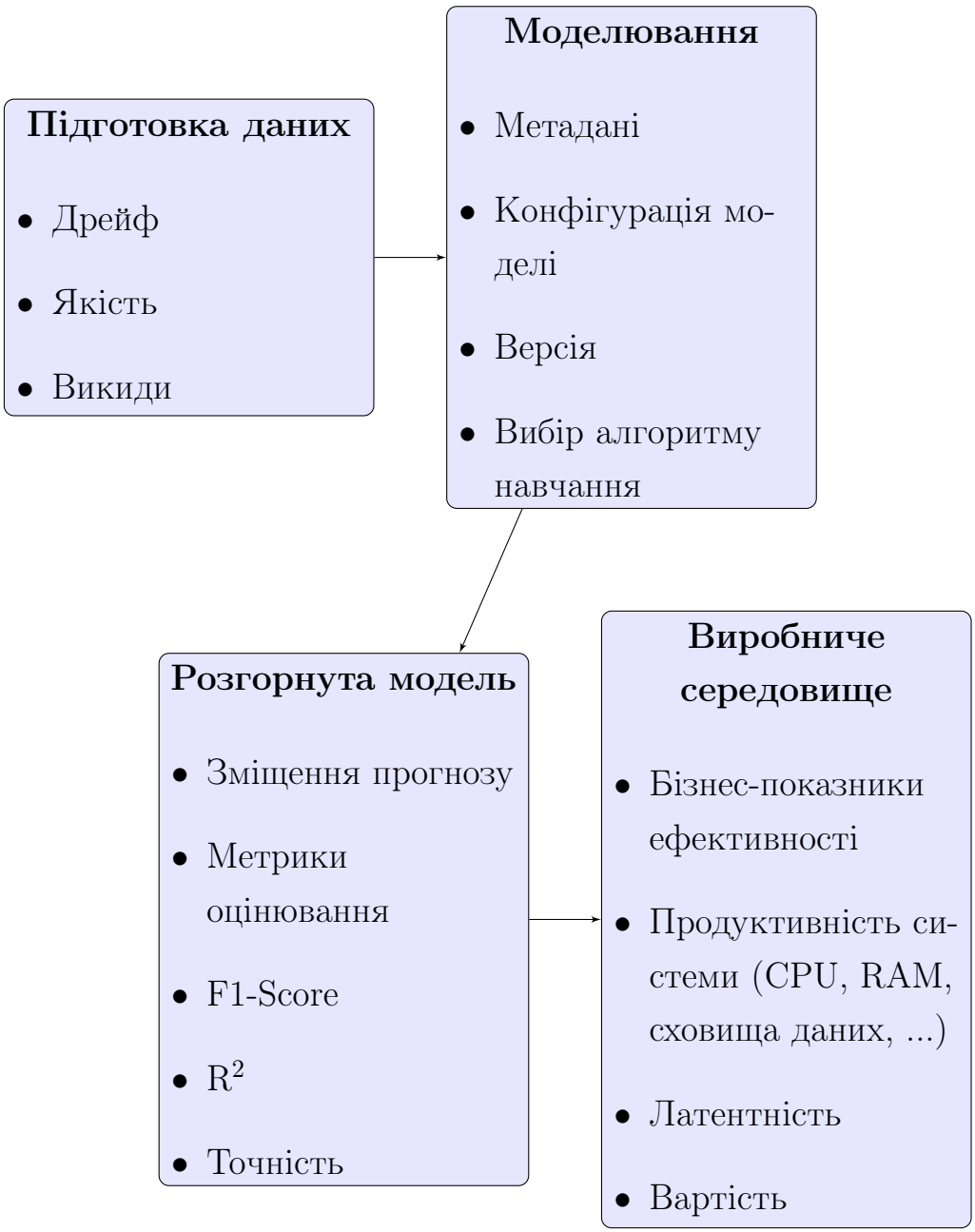

Рис. 2.7. Елементи для моніторингу [7, с. 131].

Prometheus, Grafana, ELK stack, а також комерційні рішення від хмарних провайдерів, наприклад, AWS CloudWatch, Google Stackdriver, Azure Monitor. Ці інструменти дозволяють зручно збирати, зберігати, візуалізувати метрики та налаштовувати оповіщення [7, с. 135].

2.6. Управління експериментами

Згідно Singh [28], управління експериментами (experiment tracking) полягає в систематичному збереженні метаданих експериментів машинного

навчання [28, с. 153].

Czakon, Kluge [14] визначають, что 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 ML, вагові коефіцієнти моделі, візуалізації ефективності (наприклад, матриця помилок або ROC-крива) тощо.

Управління експериментами (також відоме як реєстрація експериментів) є частиною MLOps, орієнтованої на підтримку ітеративної розробки моделі – частини життєвого циклу ML-проєкту, де, зокрема, виконується добір гіперпараметрів для досягнення необхідного рівня продуктивності моделі. Управління експериментами тісно переплітається з іншими аспектами MLOps, такими як 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. 2.8) [14]. Це характерно для дослідницьких та прикладних проектів з машинного навчання.

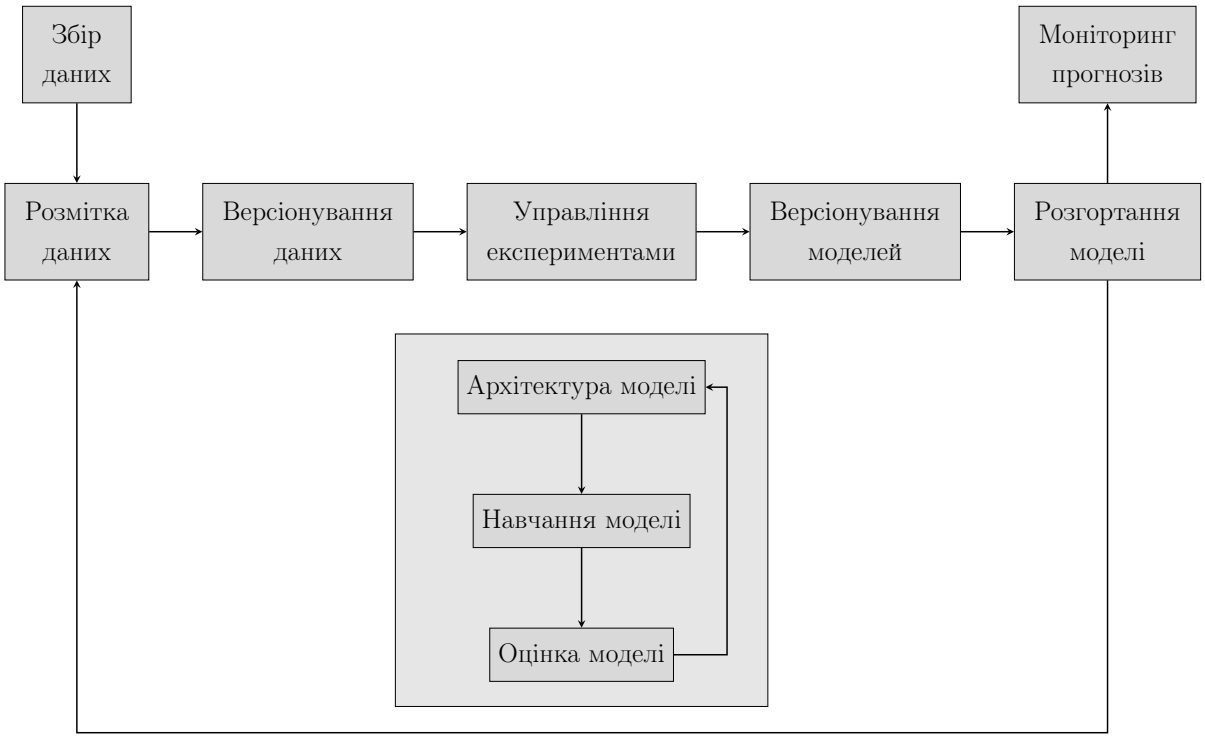

Рис. 2.8. Управління експериментами у життєвому циклі MLOps (за Czakon, Kluge [14]).

Найбільш популярними інструментами для управління експериментами є MLflow, Neptune.ai, Weights & Biases (wandb), Guild.ai, Comet.ml та TensorBoard [19]. Вони дозволяють зберігати у репозиторії інформацію про кожен експеримент – використані дані та їх версія, параметри моделі та її архітектура, метрики ефективності, артефакти моделі тощо.

2.7. Розгортання моделей

Розгортання моделей (model deployment) є важливою практикою MLOps, що відбувається на етапі операціоналізації (operationalization) у робочому процесі розробки та впровадження ML моделей. 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 [18, с. 1].

Згідно Kolltveit, Li [18], розгортання, тобто перехід упакованої та інтегрованої моделі у стан обслуговування, може відбуватися кількома різними способами. Моделі, упаковані в контейнери, просто запускаються безпосередньо як окремі сервіси. Однак моделі можуть бути розгорнуті в цільовому середовищі, яке відрізняється від того, де вони були упаковані, і в цьому випадку має відбутися перенесення моделі за допомогою push- або pull-шаблону [18, с. 4]:

- при розгортанні 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;

- при розгортанні за push-шаблоном цільове середовище сповіщається про доступність нової моделі головним сервером (наприклад, сервером, на якому була навчена модель) через службу обміну повідомленнями, де повідомлення містить метадані, включаючи місцезнаходження оновленої моделі, або шляхом ініціювання передавання моделі в цільове середовище через певний інтерфейс прийому.

Розгортання моделей ML часто відбувається шляхом їх пакування у контейнери, наприклад, за допомогою Docker. Це дозволяє стандартизува-

ти процес розгортання і гарантувати, що модель буде працювати у тому ж середовищі, що і під час розробки і тестування [18, с. 5].

Існують різні способи розгортання моделей ML у залежності від вимог і архітектури системи [26, с. 68]:

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

Розгортання моделей ML пов'язане з низкою проблем, зокрема забезпеченням малої затримки та високої пропускної здатності для сервісу прогнозування [18, с. 5]. Для їх вирішення використовуються різні методи, як-от адаптивні черги з тайм-аутом для пакетного прогнозування, кешування, динамічне переключення між моделями різної точності тощо.

З точки зору інструментів, для розгортання моделей ML використовують системи оркестрації контейнерів (Kubernetes), сервіси хмарних провайдерів (AWS SageMaker, Azure ML), наскрізні платформи MLOps (MLflow, Kubeflow) [22; 26].

Розгортання моделей ML є критично важливою практикою MLOps, що дозволяє 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 [18; 26] була побудована загальна схема розгортання моделей машинного навчання (рис. 2.9):

1. *Модель ML* – навчена та протестована модель машинного навчання.
2. *Упаковка* – модель пакується у відповідний формат (наприклад, контейнер Docker).

3. *Реєстр моделей* – упакована модель розміщується у реєстрі моделей.
4. *Розгортання* – упакована модель розгортається у цільовому середовищі (хмара, периферійні пристрої).
5. *Обслуговування* – модель обслуговує запити та генерує передбачення.
6. *Моніторинг* – продуктивність моделі та середовища моніториться. За необхідності ініціюється повторне навчання моделі.

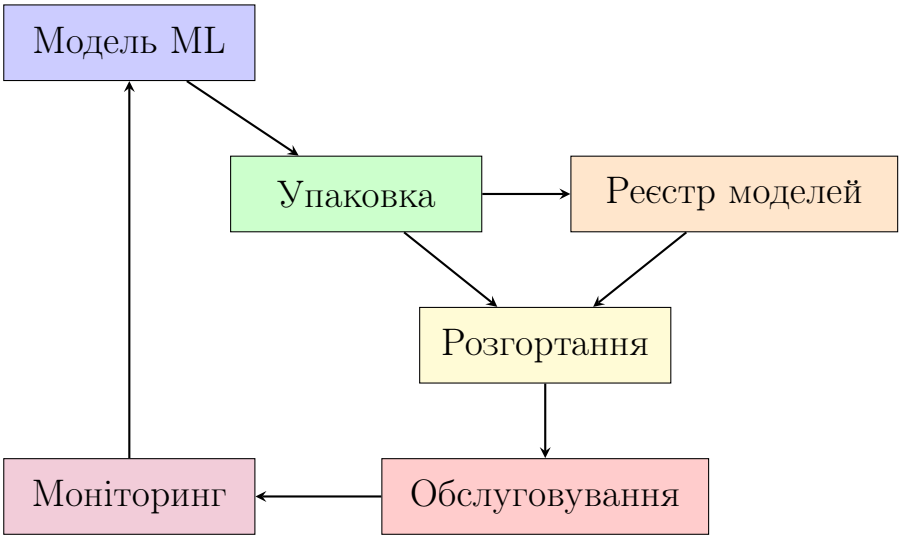

Рис. 2.9. Схема розгортання моделей ML.

2.8. Управління життєвим циклом

Згідно Steidl, Felderer, Ramler [33], безперервне управління життєвим циклом ((end-to-end) lifecycle management) – "безперервний конвеєр/поток управління, який описує (аввтоматичне) виконання певних задач задля забезпечення управління життєвим циклом штучного інтелекту" [33, с. 7], "який розпочинається зі збору даних та завершується розгортанням та моніторингом моделі штучного інтелекту" [33, с. 2].

Метою управління життєвим циклом є уніфікація та стандартизація процесів, що сприяє підвищенню продуктивності розробки моделей ML 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 [7, с. 127] (рис. 2.10);

- застосовується як на ранніх стадіях розробки моделей, так і для їх неперервної підтримки після розгортання у виробничому середовищі [2, с. 9];
- передбачає версіонування даних, коду та самих моделей для відстеження змін та забезпечення відтворюваності;
- передбачає моніторинг продуктивності моделей;
- автоматизує процеси за допомогою конвеєрів [33, с. 8].

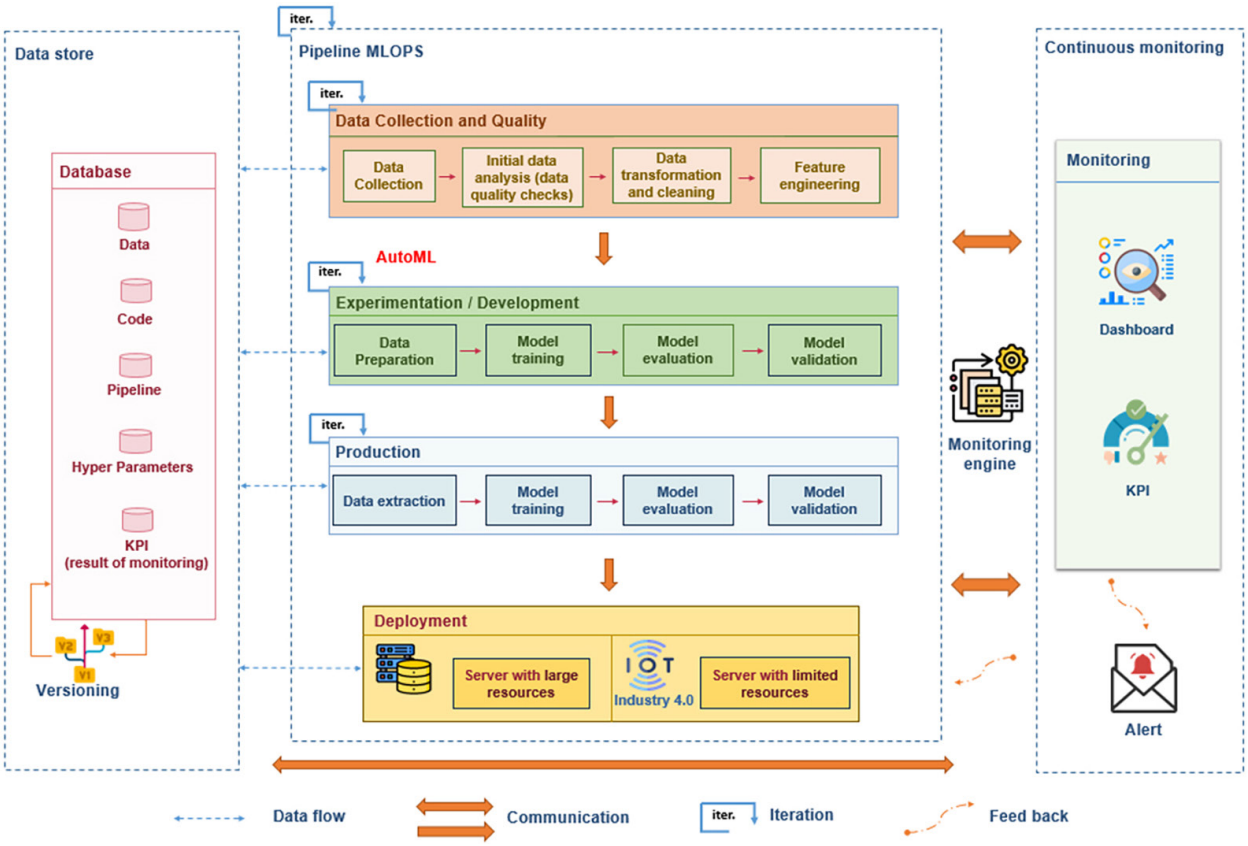

Рис. 2.10. Життєвий цикл проєкту машинного навчання у MLOps (за [7, с. 127]).

На рис. 2.10 показано процес розробки проєкту ML (конвеєр), який складається з ряду кроків, які є як лінійними, так і ітеративними (блок Pipeline MLOps). Конвеєр починається з вилучення даних, перевірки та підготовки, а потім навчання моделі, оцінка та перевірка (управління експериментами). Після цього модель розгортається у виробничому середовищі [7, с. 128].

Важливим аспектом життєвого циклу є версіонування моделей та даних, яке відбувається як у конвеєрі управління експериментами, так й 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 [7, с. 128].

Основні інструменти, що використовуються для управління життвим циклом у MLOps:

- платформи для організації ML workloads та pipelines, такі як MLflow, Kubeflow [7, с. 126];
- системи версіонування даних, наприклад DVC (Data Version Control) [7, с. 126];
- інструменти для моніторингу продуктивності моделей у виробничому середовищі, такі як Neptune.ai [7, с. 126].

Комплексний підхід до управління життєвим циклом у MLOps реалізовано в Ease.ML (`https://ease.ml/`) – системі управління життєвим циклом, призначена для спрощення всього процесу розробки [15]. Основна мета Ease.ML полягає у наданні систематичних рекомендацій та автоматизації на всіх етапах життєвого циклу ML, мінімізуючи зусилля користувачів.

Ключові особливості Ease.ML:

1. *Процес із залученням людини* – Ease.ML включає взаємодію з користувачем у структурованій формі, забезпечуючи можливість внесення користувачем даних і прийняття рішень на критичних етапах.
2. *Ймовірнісна модель даних* – система використовує ймовірнісну базу даних, яка обробляє невизначеність у даних, що може виникати через некоректні дані, слабкий нагляд або інші джерела.

3. *Інтерактивне середовище* – Ease.ML використовує блокноти Jupyter, дозволяючи користувачам виконувати маніпуляції з даними, запускати операції ML та візуалізувати результати в інтегрованому середовищі.

4. *Графи лінійності* – взаємодії та операції користувачів відстежуються у графах лінійності, які представляють весь робочий процес ML і забезпечують відтворюваність.

5. *Автоматичне налаштування якості та рекомендації* – система надає рекомендації для покращення моделей на основі помилок, виявлених у продукції, спрямовуючи користувачів через завдання очищення та збору даних ефективно.

Процес Ease.ML поділений на три підпроцеси і складається з восьми етапів, які охоплюють весь життєвий цикл ML:

- Day 0: Pre-ML Subprocess

    1. *Формулювання проблеми* – чітко визначити проблему та цілі ML).
    2. *Техніко-економічне обґрунтування* – оцінити доцільність ML рішення з наявними даними та ресурсами.

- Day 1: AutoML Subprocess

    3. *Підготовка даних* – очищення, попереднє опрацювання та доповнення даних, щоб зробити їх придатними для навчання моделей ML.
    4. *Навчання моделей*, використовуючи підготовлені дані, з використанням AutoML для автоматизації вибору та налаштування моделей.
    5. *Оцінка продуктивності 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* через постійний моніторинг моделі у виробничому середовищі та необхідні оновлення або перенавчання моделі для адаптації до нових даних або змінних умов.

2.9. 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 [11, с. 1]. 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 [1, с. 8].

Практика безпеки та конфіденційності даних є обов'язковою у випадках, коли система ML 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 забезпечений відповідно до нормативних вимог та очікувань користувачів.

На відміну від традиційної розробки ПЗ, у контексті MLOps з'являються нові вектори атак та вразливості, специфічні для машинного навчання (рис. 2.11). Зокрема, моделі ML вразливі до таких атак, як отруєння даних (data poisoning), інверсія моделі (model inversion) та атаки з використанням змагальних прикладів (adversarial examples) [11, с. 8-15]. Це вимагає застосування спеціалізованих стратегій захисту.

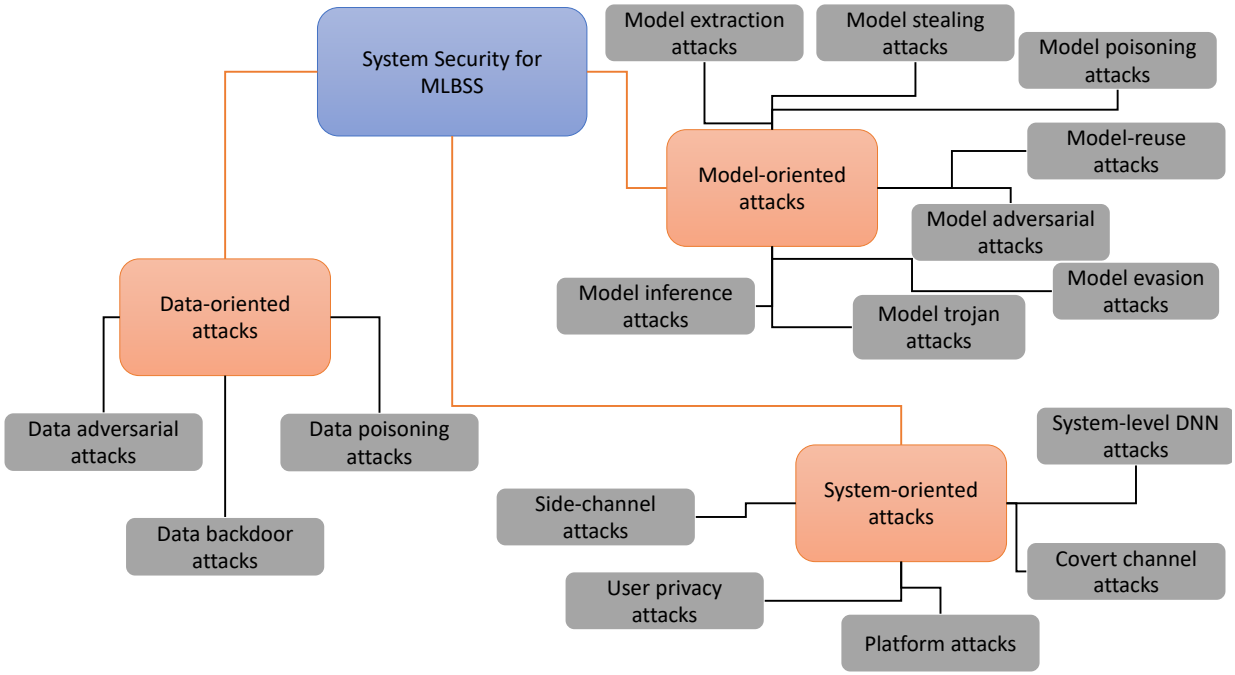

Рис. 2.11. Класифікація атак на системи MLOPs [11, с. 9].

Для забезпечення безпеки та конфіденційності даних в MLOps використовуються як стандартні інструменти безпеки (шифрування, аутентифікація, журналювання тощо), так і спеціалізовані рішення, орієнтовані на ML. Прикладами останніх є бібліотеки для безпечного агрегування федеративного навчання, інструменти для тестування моделей на вразливості, фреймворки для конфіденційного машинного навчання на основі гоморфного шифрування або захищених анклавів [1, с. 10-11].

Безпека та конфіденційність даних забезпечується шляхом впровадження різних механізмів контролю та захисту на кожному етапі MLOps. Це включає, зокрема, шифрування та анонімізацію даних, аутентифікацію та контроль доступу, перевірку цілісності, моніторинг активності, реагування на інциденти, а також регулярні перевірки безпеки та тестування на проникнення [11, с. 4-5].

На рис. 2.12 узагальнено подані у [11, с. 22-25] 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 ве-

рифікацію 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:

1. На етапі визначення проблеми:
    - оцінка ризиків (Risk Assessment);
    - моделювання загроз (Threat Modeling).
2. 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);
    - верифікація даних (Data verification).
3. 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).
4. На етапі підтримки системи:
    - засоби моніторингу моделей ML (MLDEMON, SelfChecker);
    - статичний аналіз (Static Analysis);

- дослідницькі атаки (Exploratory Attacks);
- атаки ухилення (Evasion Attacks);
- атаки отруєння даних (Data Poisoning Attacks);
- ручне тестування (Manual Testing);
- динамічний аналіз (Dynamic Analysis).

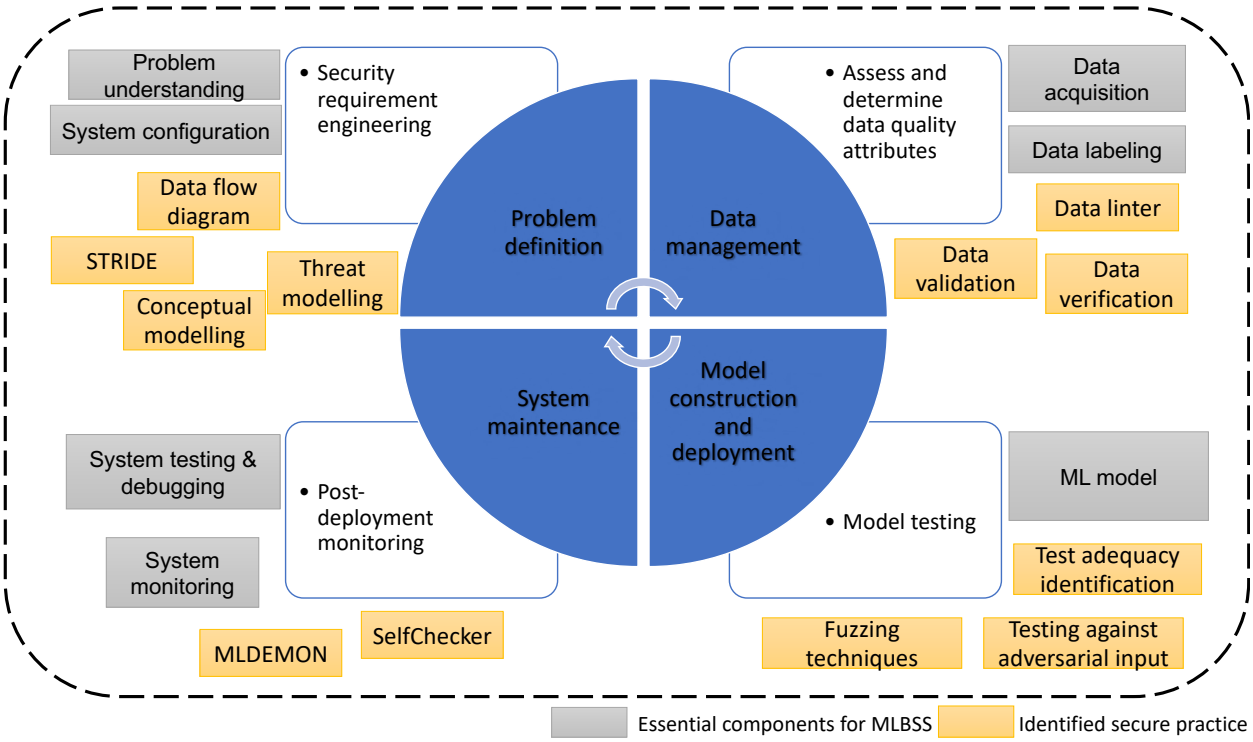

Рис. 2.12. Основні компоненти та практики безпеки у MLOps (за [11, с. 20]).

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

2.10. Пояснюваність та інтерпретовність моделей

Пояснюваність та інтерпретовність (Explainability/interpretability) моделей є важливою практикою MLOps для забезпечення прозорості та довіри до моделей машинного навчання. Пояснюваність відноситься до здатності пояснити та зрозуміти процеси прийняття рішень моделі машинного

навчання [36, с. 66]. Це особливо важливо, коли модель використовується для прийняття рішень, які мають значні наслідки, наприклад, у військових операціях чи правоохоронній діяльності [17].

Пояснюваність як основи довіри до проєкту ML дозволяє користувачам довіряти прогнозу, що підвищує прозорість. 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, з точки зору важливості, повноти і вірності прогнозів або класифікацій (рис. 2.13) [24, с. 63608].

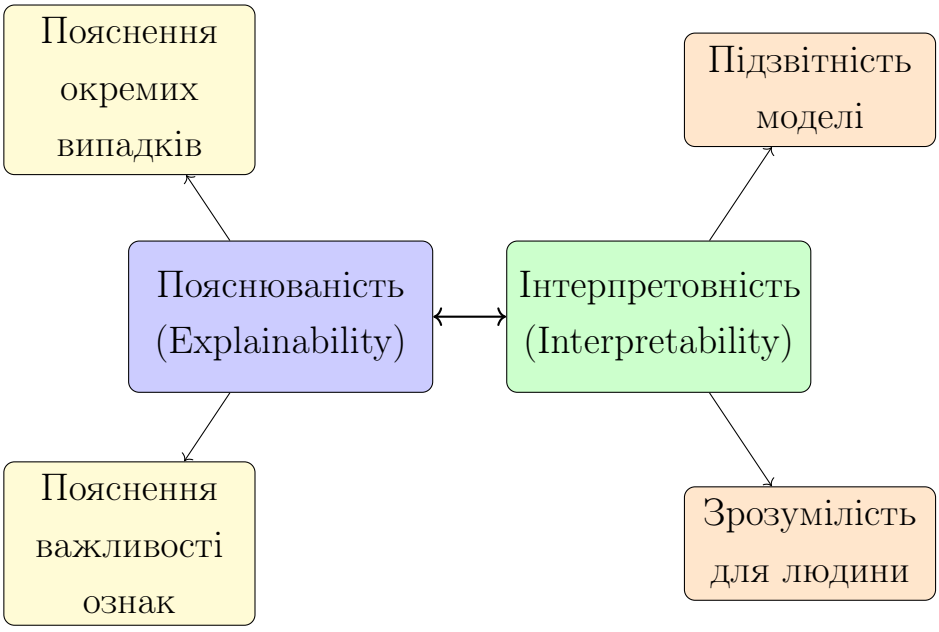

Рис. 2.13. Зв'язок пояснюваності та інтерпретовності у MLOps.

Пояснюваний штучний інтелект (Explainable Artificial Intelligence, XAI) – це дослідницький напрям, який сприяє прийняттю пояснюваних рішень [24, с. 63609]. Пояснюваність можна визначити як "ступінь, до якого людина може зрозуміти причину рішення" [24, с. 63614]. Система ML є пояснюваною, коли легше ідентифікувати причинно-наслідкові зв'язки між входами та виходами системи. Чим більш пояснюваною є модель, тим краще фахівці-практики розуміють внутрішні бізнес-процедури, які відбуваються під час прийняття рішень моделлю. Пояснювана модель не обов'язково перетворюється на модель, яку може зрозуміти людина (внутрішня логіка або процеси, що лежать в основі), але пояснюваність моделі дозволяє користувачеві зміцнити довіру до прогнозів, зроблених

розгорнутою системою [24, с. 63614-63615].

Для досягнення пояснюваності моделей можуть використовуватися різні способи та інструменти, такі як:

- методи на основі атрибуції (Integrated Gradients, Saliency Maps) або пертурбації (SHAP) [30, с. 3], які пояснюють рішення моделі, призначаючи високі оцінки найвпливовішим вхідним параметрам;
- включення механізму уваги (attention mechanism) у моделі, що дозволяє зосередитись на найбільш релевантних станах мережі та входах [30, с. 2];
- використання у процесі навчання комбінованого сигналу винагороди (reward), який включає не тільки цільові показники, але й метрики інтерпретовності [30, с. 4].

2.11. Управління якістю даних

Управління якістю даних є важливою практикою у робочому процесі MLOps. Згідно Steidl, Felderer, Ramler [33], етап обробки даних включає повний життєвий цикл роботи з даними, зокрема попередню обробку, забезпечення якості, версіонування та документування [33, с. 21].

У книзі Халлера [17, с. 77-84] управління якістю даних розглядається в контексті моніторингу та перевірок даних у виробничому середовищі, щоб забезпечити відповідність використовуваних даних реальності.

У життєвому циклі MLOps, заснованому на даних (рис. 2.14), управління якістю даних виконується на наступних етапах:

1. Збір даних – на цьому етапі відбувається створення та отримання даних із різних джерел: дані мають бути релевантними, повними, узгодженими та надійними.
2. Оцінка якості та очищення даних – зібрані дані проходять ретельну перевірку якості за різними метриками, такими як точність, повнота, узгодженість, своєчасність тощо; виявляються та усуваються проблеми з якістю - некоректні значення, пропуски, дублікати, шум; застосовуються техніки очищення даних для покращення їх якості.

3. Доповнення та розмітка даних – очищені дані збагачуються додатковою інформацією та розмічаються відповідно до цільової задачі; на цьому етапі 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; за потреби вносяться корективи для покращення якості даних.

5. Навчання моделі з контролем якості – на основі якісних даних відбувається навчання моделі ML; під час експериментів відстежуються метрики якості моделі та даних, щоб гарантувати стабільність та надійність результатів.

6. Розгортання моделі з забезпеченням якості – модель розгортається у виробничому середовищі лише після ретельного тестування на якісних тестових даних; 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 в даних, зміщення розподілів; за потреби модель донавчається на нових якісних даних.

Щоб гарантувати якість даних упродовж усього життєвого циклу їх використання у MLOps, необхідно застосовувати деякі підходи до валідації та верифікації даних:

- оцінювання якості даних шляхом визначення, наскільки ці дані придатні для досягнення бізнес-цілей (повнота, унікальність, цілісність, валідність, точність, своєчасність) [28, с. 2-3];

- валідація даних у одиночних та перехресних пакетах шляхом порівняння характеристик даних із очікуваною схемою, а також перевірка наявності зсувів даних [33, с. 11];

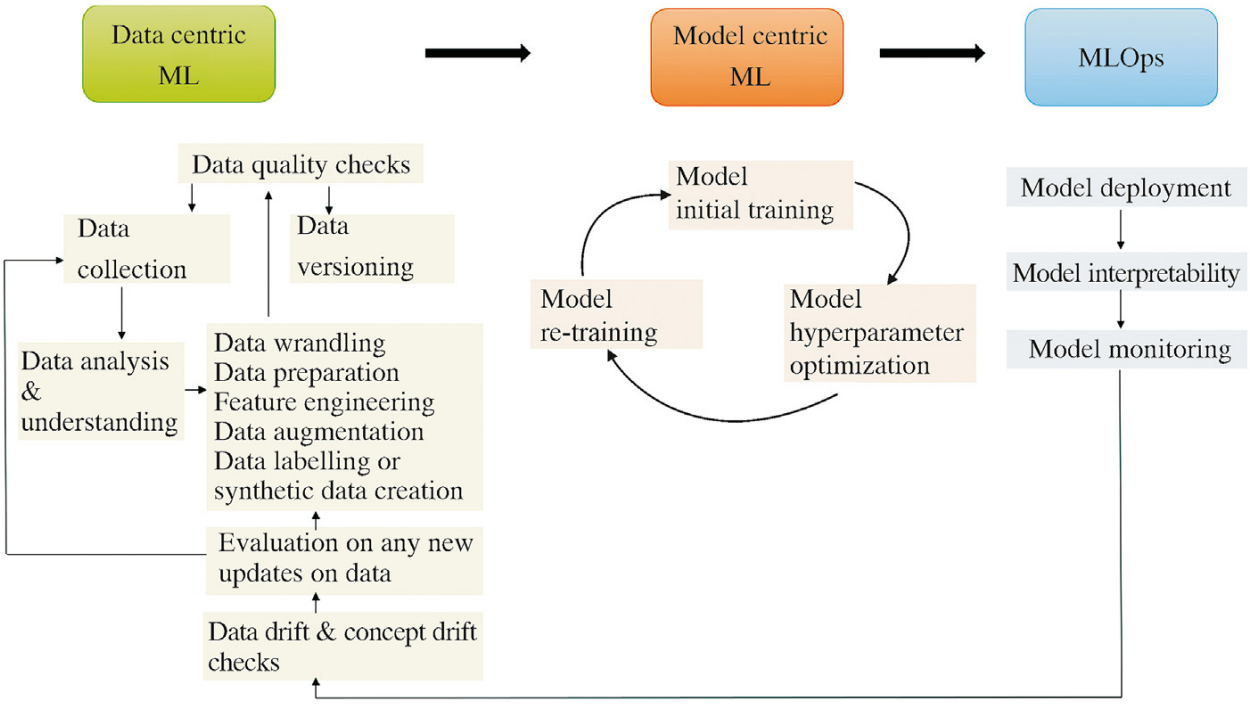

Рис. 2.14. Життєвий цикл MLOps, заснований на даних (за [28, с. 145]).

- автоматичні модульні тести даних на основі схеми для виявлення помилок у даних та запобігання їх проникнення на етап навчання моделі [33, с. 10];
- версіонування даних і пов'язаних артефактів (процедур обробки, метаданих тощо) для відслідковуваності, відтворюваності результатів і дотримання регуляторних вимог [33, с. 11];
- документування даних у обсязі, достатньому для забезпечення веріфікованості моделей [33, с. 12].

Вчасне виявлення і усунення дефектів у даних допомагає запобігти марнуванню обчислювальних ресурсів на неякісних даних [33, с. 11].

У роботі Singh [28] 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 (TFX), в яких є компоненти для валідації даних, такі як SchemaGen та ExampleValidator [33, с. 19].

2.12. Управління конфігурацією

Конфігураційні файли допомагають створювати більш надійне програмне забезпечення, переміщуючи всі жорстко закодовані змінні в спеціальні місця, які можуть бути розділені або організовані на розсуд розробника [16, с. 3].

Godwin, Melvin [16] пропонує шаблон, що підтримує два типи файлів конфігурації – файл config.py, що містить конфігурацію шаблону та може включати додаткові ресурси, такі як бази даних та електронні таблиці, та файли JSON для збереження специфічних змінних програми, таких як порогові значення чи параметри. Файл config.py доступний через інструкції імпорту, а JSON файли викликаються з диску під час виконання [16, с. 3-4].

Yongqiang та ін. [8] розглядають використання єдиної моделі даних на основі мови YANG для уніфікації опису конфігураційних даних і спрощення управління ними. Neptune Labs [25] вказує, що інструменти управління конфігурацією, такі як Ansible (`https://www.ansible.com/`), Puppet (`https://www.puppet.com/`) та Chef (`https://www.chef.io/`), можуть бути використані для автоматизації конфігурації та забезпечення інфраструктури платформ MLOps.

2.13. Стратегії розгортання моделей

Стратегії розгортання моделей:

- реалізуються на фінальному етапі процесу MLOps;
- потребують стандартизації, автоматизації та інкапсуляції моделей;
- застосовуються для поступового переходу нової моделі у виробниче середовище;
- спираються на контейнеризацію.

Peltonen, Dias [26] виокремлють такі переваги використання контейнерів для розгортання моделей [26, с. 68]:

- абстрагування моделей та ізоляція процесів шляхом запуску декількох моделей в окремих контейнерах, що представляють їх залежності під час виконання;

- можливість створювати специфічні для кожної моделі контейнери, що відповідають їхнім вимогам до упаковки;
- призначення на різні процесори (CPU, Mobile GPU тощо);
- можливість виділення окремого контейнера для полегшення функцій пост-обробки;
- для швидкого розгортання можна використовувати репозиторій моделей та контейнерне сховище;
- забезпечує засоби стандартизованого розгортання;
- майже всі існуючі конвеєри неперервної розробки полегшують розгортання моделей у вигляді контейнерів;
- контейнери можуть бути адаптовані до конкретних архітектур периферійних пристроїв;
- контейнери Docker є усталеним стандартом в галузі;
- простота відкату в разі збоїв.

Gunny та ін. [4] детально описують 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. При цьому застосовується сервісна архітектура з NVIDIA Triton Inference Server, що підтримує одночасно розміщення версії розробника та виробничої версії моделі.

Автори виділяють два основні підходи до розгортання моделей [4, с. 11-12].

За традиційного сценарію кожен користувач керує власними ресурсами та версіями моделей. Невідповідності в бібліотеках та залежностях, а також версіях моделей призводять до непослідовних результатів. Зменшені обчислювальні вимоги до виведення (inference) призводять до недовикористання апаратних ресурсів, зображеного зеленими прямокутниками на кожному вузлі (рис. 2.15). Більш складні сценарії розгортання вимагають використання декількох мереж, посилюючи існуючі проблеми.

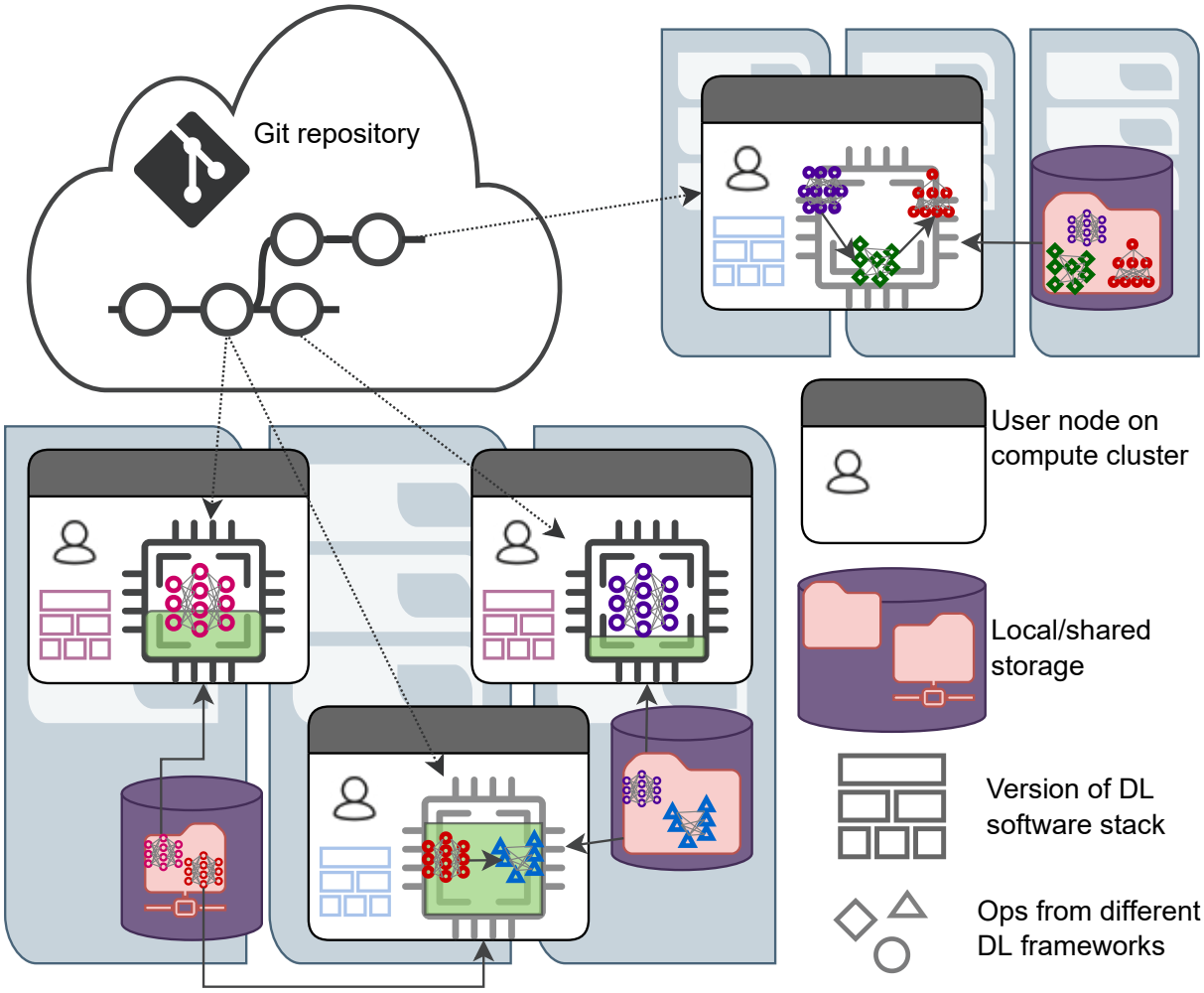

Рис. 2.15. Традиційний сценарій розподіленого розгортання (за [4, с. 12]).

За підходу Inference-as-a-Service централізована служба оркеструє моделі й надає уніфіковані інтерфейси для виклику моделей (рис. 2.16). 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 масштабований спосіб. За такого підходу контейнеризація дозволяє створювати портативні та ізольовані середовища виконання моделей.

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

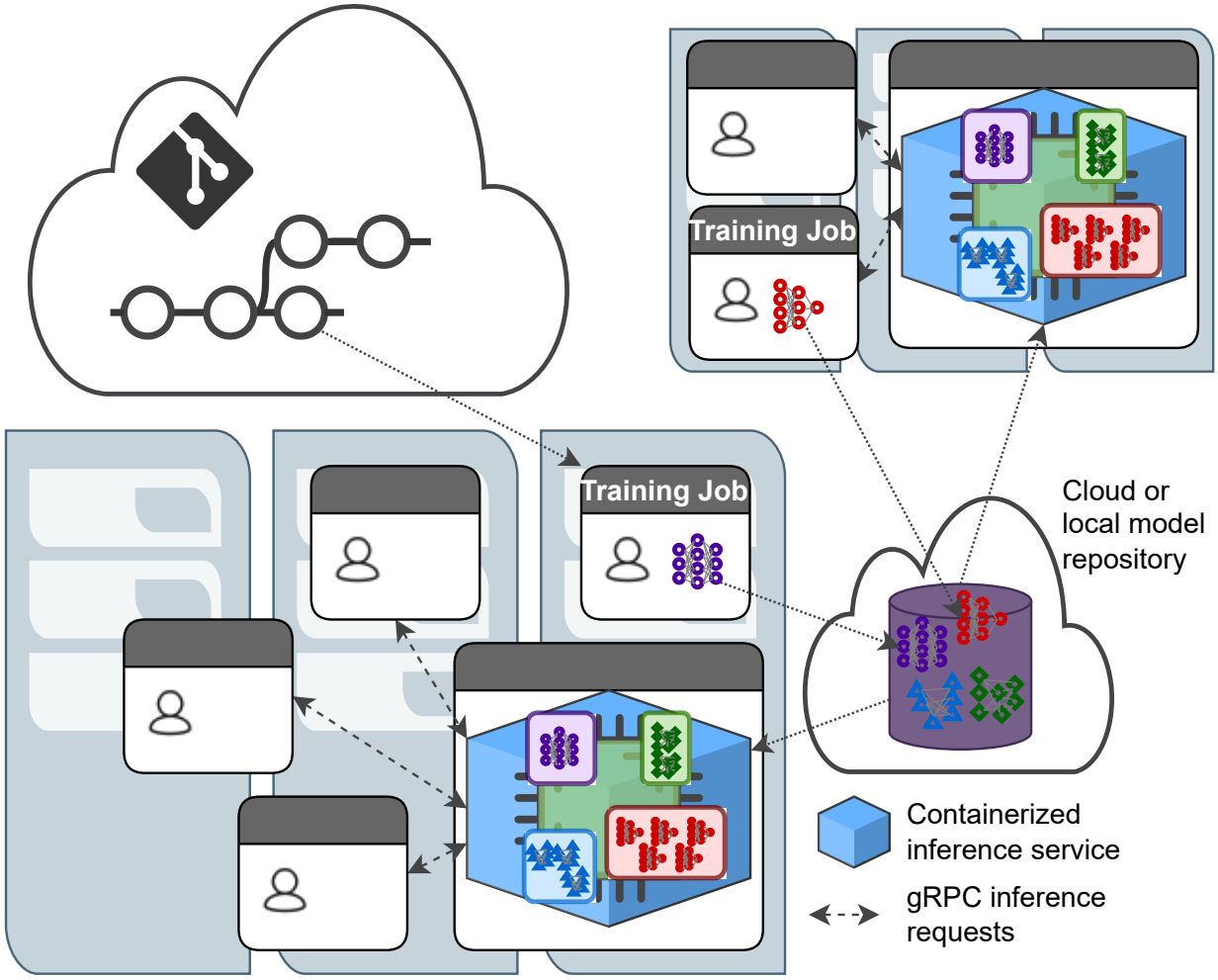

Рис. 2.16. Розгортання за сценарієм Inference-as-a-Service (за [4, с. 12]).

2.14. Автоматизація інфраструктури

Автоматизація інфраструктури (Infrastructure as code) є важливою практикою MLOps, яка дозволяє розглядати інфраструктуру як код для її надійного та ефективного розгортання й управління [5, с. 2]. Автоматизація інфраструктури відноситься до етапу розгортання та моніторингу у процесі MLOps (рис. 2.17). Вона забезпечує надійне створення необхідного оточення для розгортання моделей машинного навчання та автоматизує інфраструктурні завдання [5, с. 3].

Infrastructure as code передбачає опис конфігурації інфраструктури декларативним способом у спеціальних файлах (наприклад, у форматах YAML, JSON) або за допомогою спеціальних мов (наприклад, Terraform) чи інструментів. Ці файли описують бажаний стан інфраструктури [5, с. 4].

Опис налаштувань інфраструктури у вигляді коду дозволяє автоматизувати процес її створення, зміни та управління [5, с. 3]. Це дає мо-

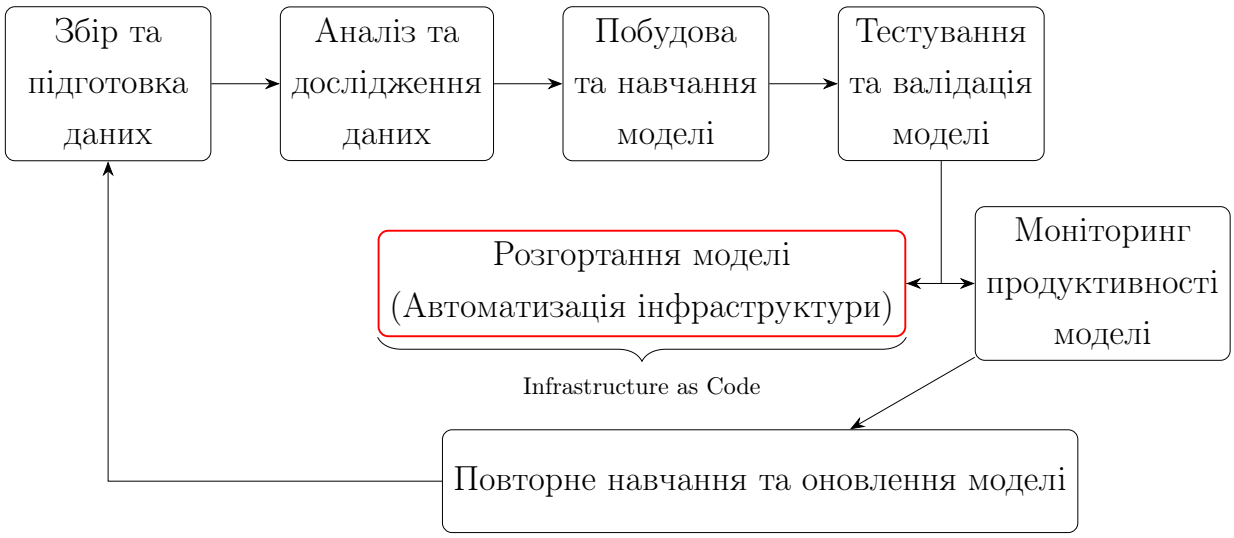

Рис. 2.17. Автоматизація інфраструктури у життєвому циклі MLOps.

жливість повністю відтворювати оточення, швидко розгортати ресурси та уникати помилок ручного налаштування [5, с. 4]. Даний підхід доцільно використовувати для створення складних динамічних інфраструктур, коли потрібна висока повторюваність та узгодженість оточень, для підвищення надійності та зниження витрат часу на ручні налаштування [5, с. 2, 4].

Автоматизований підхід дозволяє тестувати інфраструктуру як код, застосовувати до неї практики з розробки ПЗ (перегляд коду, версіонування тощо). Це сприяє підвищенню стабільності та безпеки, дозволяє оперативно відслідковувати та усувати проблеми в інфраструктурі [5, с. 4].

Для реалізації Infrastructure as code у Azure DevOps застосовуються такі інструменти, як Azure Resource Manager (ARM) Templates, Terraform, Ansible, Chef [5, с. 3-4]. Вони дозволяють описувати інфраструктуру у вигляді коду та автоматизовано створювати чи змінювати її.

2.15. 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 увагу на тому, як крос-функціональні команди, такі як аналітики даних, системні оператори, а також інженери з даних та програмного забезпечення, співпрацюють через узгоджений процес [33, с. 7].

Сутність практики співпраці та комунікації полягає у налагодженні ефективної взаємодії та обміну інформацією між різними командами, залученими до процесу розробки та впровадження ML-моделей – командою науки про дані (data science), розробки (development), операційної діяльності (operations) та бізнес-підрозділами. Ця практика є важливою складовою етапу розробки та впровадження робочого процесу MLOps.

Як зазначають Kreuzberger, Kühl, Hirschl [19], MLOps передбачає тісну співпрацю між командами науковців з даних (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 [19, с. 2]. Ефективна комунікація необхідна на всіх етапах життєвого циклу моделей ML, починаючи від визначення бізнес-цілей та закінчуючи моніторингом моделей у виробничому середовищі. Без налагодженої співпраці розробка ML-рішень може затягуватись, виникатимуть конфлікти інтересів та непорозуміння між учасниками [32, с. 3].

Схема (рис. 2.18) відображає основних учасників процесу MLOps та напрямки їх взаємодії:

1. Команда Data Science готує дані та розробляє моделі машинного навчання.

2. Команда Development інтегрує розроблені моделі у програмні продукти.

3. Команда Operations розгортає моделі у виробничому середовищі та забезпечує їх підтримку.

4. Команда Business визначає цілі та вимоги до ML-рішень, а також отримує результати від команд Data Science та Operations.

Для забезпечення ефективної комунікації у практиці MLOps застосовують такі підходи та інструменти [19, с. 4]:

- використання інструментів для спільної роботи та обміну знаннями, таких як Slack, Trello, GitLab wiki;

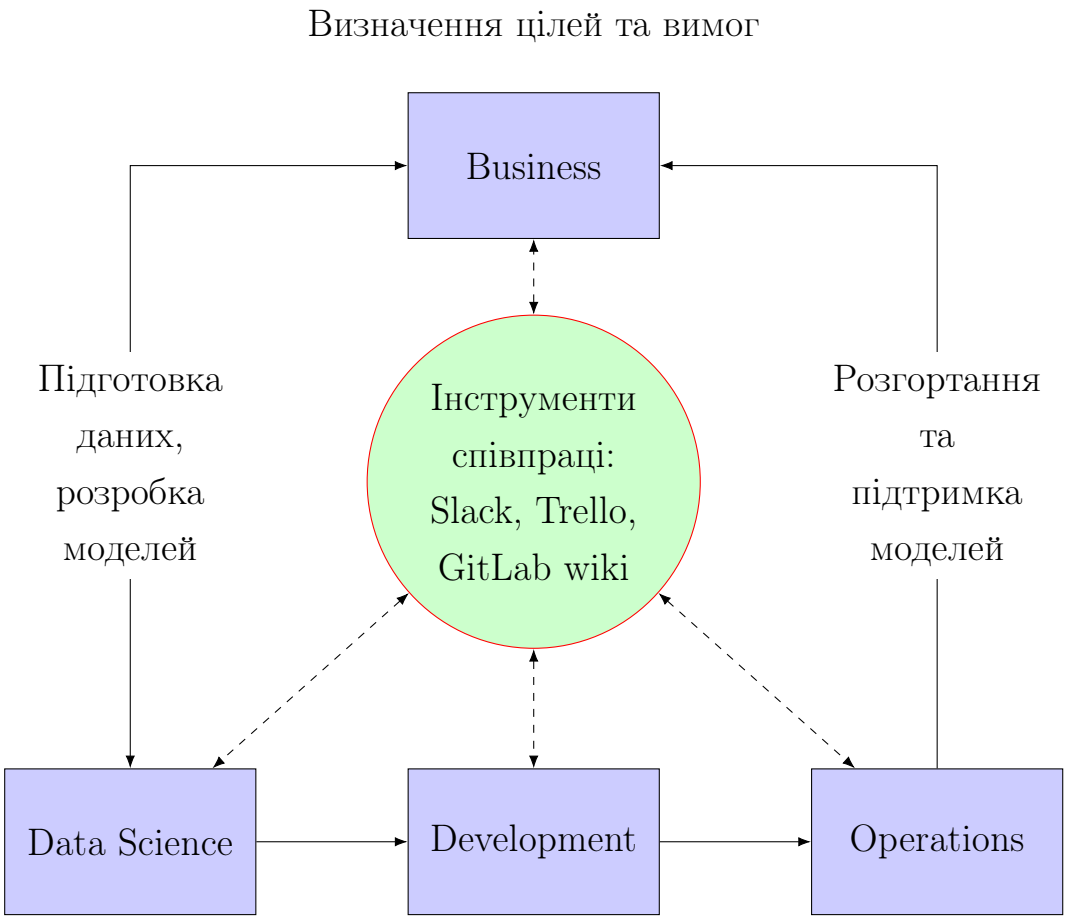

Рис. 2.18. Схема співпраці та комунікації у MLOps.

- регулярні зустрічі між командами для обговорення статусу, проблем та планів;
- чітке визначення ролей та зон відповідальності учасників процесу MLOps;
- використання систем контролю версій (Git) для спільної роботи над кодом та моделями;
- автоматизація процесів CI/CD для забезпечення прозорості та відтворюваності розробки.

2.16. Управління ризиками та комплаєнс

Оскільки моделі ML часто приймають важливі рішення, що впливають на людей, управління ризиками та комплаєнс є критично важливою

практикою MLOps, сутність якої полягає у забезпеченні відповідності розроблюваних ML-моделей та систем нормативним вимогам та стандартам (compliance) та управлінні потенційними ризиками, пов'язаними з їх розробкою та експлуатацією.

Ця практика має наскрізний характер та проявляється на різних етапах життєвого циклу моделі ML. Управління ризиками в контексті MLOps передбачає виявлення та зменшення потенційних ризиків, пов'язаних з моделями ML, таких як упередженість даних, порушення конфіденційності, погіршення точності моделі з часом тощо. Комплаєнс означає забезпечення відповідності моделей нормативним вимогам, наприклад, щодо захисту даних [33, с. 18-19]. Основною умовою використання цієї практики є наявність нормативно-правових вимог чи галузевих стандартів, яким повинна відповідати система ML. Прикладами можуть бути вимоги GDPR щодо захисту персональних даних чи сертифікації у сфері охорони здоров'я [33, с. 5]. Steidl, Felderer, Ramler [33] вказують, що аспекти комплаєнсу мають враховуватись під час підготовки даних, навчання та валідації моделі, а також розгортання та моніторингу [33, с. 18].

Способи використання цієї практики включають регулярні перевірки якості даних, тестування моделей на упередженість та дискримінацію, впровадження контролю доступу та шифрування даних, документування архітектури моделі та процесу розробки, моніторинг продуктивності моделі після розгортання [33, с. 11-12]. Так, забезпечення комплаєнсу досягається шляхом [33, с. 18]:

- контроль якості та походження даних, що використовуються для навчання моделей, з метою уникнення порушення регуляторних вимог;
- документування та версіонування моделей та даних для забезпечення відтворюваності результатів та аудиту;
- верифікацію інтегрованих до виробничого середовища моделей на відповідність вимогам;
- безперервний моніторинг розгорнутих моделей для своєчасного виявлення потенційних порушень чи некоректної поведінки.

Провідними засобами забезпечення практики управління ризиками

та комплаєнс є системи контролю версій для відстеження змін в даних та коді моделі, засоби тестування для виявлення проблем у моделях, системи моніторингу для відстеження точності моделей в реальному часі [33, с. 12, 14]. Водночас інтерв'ю з практиками, проведені Steidl, Felderer, Ramler [33], виявили, що забезпечення комплаєнсу ML-систем в різних доменах (наприклад, в охороні здоров'я) є серйозним викликом через брак усталених методик та програмних інструментів. Водночас, досягнення комплаєнсу є обов'язковою умовою для отримання необхідних сертифікацій та дозволів регуляторів.

Схема на рис. 2.19 відображає три основні етапи MLOps: роботу з даними навчання моделі та розгортання системи (операціоналізація). На кожному етапі присутні блоки, які позначають потенційні ризики та заходи з забезпечення відповідності. Дана схема ілюструє наскрізний характер практики управління ризиками та комплаєнсу в MLOps, показуючи її присутність та взаємозв'язки на кожному етапі життєвого циклу моделі машинного навчання.

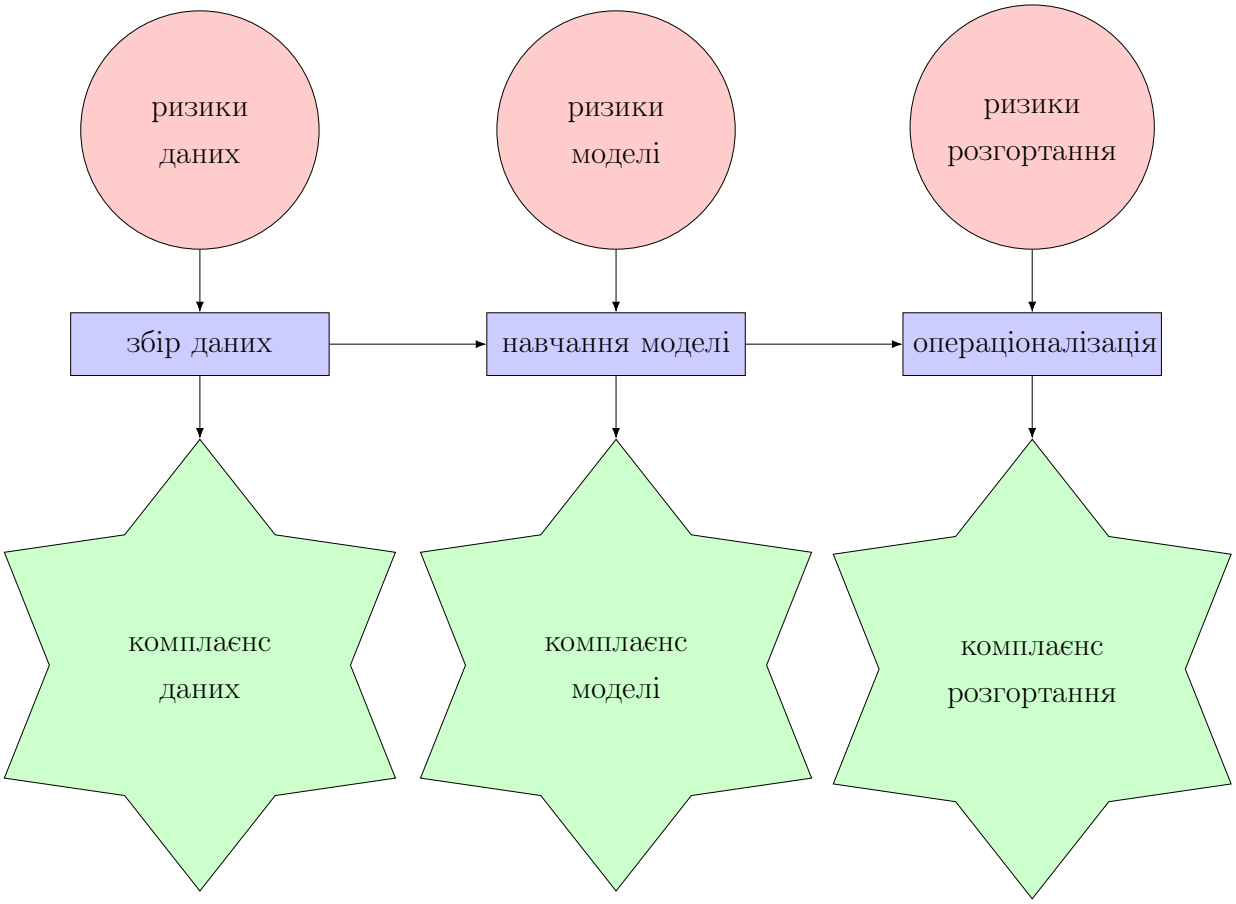

Рис. 2.19. Схема практики управління ризиками та комплаєнсу у MLOps.

Рис. 2.20 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, яка CI/CD, розгортання моделей, безпека та конфіденційність даних, управління конфігурацією, стратегії розгортання моделей та автоматизація інфраструктури.

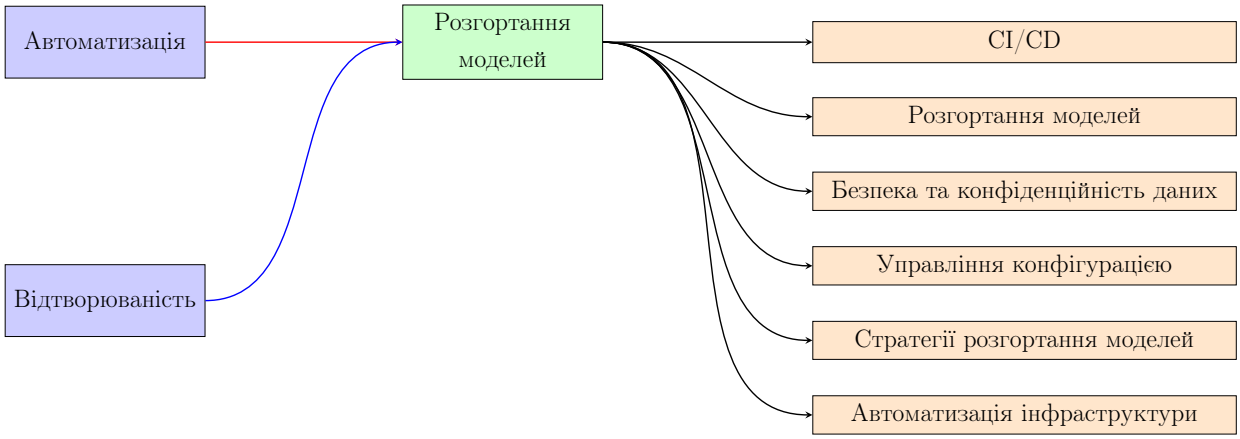

Рис. 2.20. Схема зв'язків між принципами, процесами та практиками MLOps для розгортання моделей.

Висновки до 2 розділу

У даному розділі було проаналізовано ключові практики MLOps, необхідні для ефективного розгортання моделей машинного навчання. Основні висновки, отримані в результаті аналізу, є такими:

1. MLOps базується на наборі принципів, процесів та практик, які забезпечують ефективне розроблення, розгортання та підтримку моделей машинного навчання. Ключовими принципами MLOps є автоматизація, відтворюваність, співпраця, безперервне навчання та керованість даними.

2. Основні практики MLOps включають: безперервну інтеграцію та доставку (CI/CD), версіонування моделей та даних, автоматизацію конвеєрів ML, моніторинг продуктивності моделей, управління експериментами, розгортання моделей та управління життєвим циклом.

3. Додаткові практики MLOps, такі як безпека та конфіденційність даних, пояснюваність та інтерпретовність моделей, управління якістю даних, управління конфігурацією, стратегії розгортання моделей, автоматизація інфраструктури, співпраця та комунікація, управління ризиками та комплаєнс, є важливими для забезпечення надійності, відповідності вимогам та ефективності процесів MLOps.

4. Застосування практик MLOps дозволяє автоматизувати та стандартизувати процеси розробки, розгортання та підтримки моделей машинного навчання, що підвищує ефективність та надійність ML-рішень у виробничому середовищі.

5. Успішне впровадження практик MLOps вимагає використання відповідних інструментів та платформ, таких як системи управління експериментами, версіонування даних і моделей, інструменти автоматизації інфраструктури та моніторингу, а також налагодження ефективної співпраці між різними ролями та командами, залученими до процесу розробки та впровадження моделей машинного навчання.

Проведений аналіз показав, що застосування практик MLOps є критично важливим для успішного розгортання моделей машинного навчання у виробничих середовищах. Впровадження цих практик дозволяє підвищити ефективність, надійність та відтворюваність процесів розробки та експлуатації ML-рішень, що є необхідною умовою для їх успішного використання в реальних бізнес-задачах.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досягнуто поставленої мети – визначено та проаналізовано практики MLOps, необхідні для ефективного розгортання моделей машинного навчання. Основні висновки, отримані в ході дослідження, є такими:

1. Виконано мета-синтез систематичних оглядів для узагальнення знань щодо практик MLOps. Проведений мета-синтез показав, що MLOps є перспективним підходом для ефективного розгортання моделей машинного навчання у виробничих середовищах, який потребує подальшого дослідження та розвитку для вирішення існуючих викликів та реалізації потенційних можливостей.

2. Запропоновано схему зв'язків між принципами, процесами та практиками MLOps. Дана схема ілюструє взаємозв'язки між ключовими принципами, етапами процесу розробки та впровадження моделей машинного навчання, а також основними практиками MLOps, які застосовуються на кожному етапі.

3. Виявлено найбільш ефективні практики MLOps для розгортання моделей, які включають: безперервну інтеграцію та доставку (CI/CD), версіонування моделей та даних, автоматизацію конвеєрів ML, моніторинг продуктивності моделей, управління експериментами, розгортання моделей та управління життєвим циклом, безпеку та конфіденційність даних, пояснюваність та інтерпретовність моделей, управління якістю даних, управління конфігурацією, стратегії розгортання моделей, автоматизацію інфраструктури, співпрацю та комунікацію, управління ризиками та комплаєнс.

Отримані результати мають як теоретичне, так і практичне значення. Теоретичне значення полягає в узагальненні та систематизації знань щодо практик MLOps, необхідних для ефективного розгортання моделей машинного навчання. Практичне значення отриманих результатів полягає в можливості їх використання організаціями для впровадження або вдосконалення процесів MLOps з метою підвищення ефективності та надійності розгортання моделей машинного навчання у виробничих середовищах.

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

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “If security is required”: engineering and security practices for machine learning-based IoT devices / N. K. Gopalakrishna [та ін.] // Proceedings of the 4th International Workshop on Software Engineering Research and Practice for the IoT. — Pittsburgh, Pennsylvania : Association for Computing Machinery, 2023. — С. 1—8. — (SERP4IoT ’22). — DOI: 10.1145/3528227.3528565.

2. A Joint Study of the Challenges, Opportunities, and Roadmap of MLOps and AIOps: A Systematic Survey / J. Diaz-de-Arcaya [та ін.] // ACM Comput. Surv. — New York, NY, USA, 2023. — Жовт. — Т. 56, № 4. — DOI: 10.1145/3625289.

3. A Multivocal Literature Review of MLOps Tools and Features / G. Recupito [та ін.] // 2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). — 2022. — С. 84—91. — DOI: 10.1109/SEAA56994.2022.00021.

4. A Software Ecosystem for Deploying Deep Learning in Gravitational Wave Physics / A. Gunny [та ін.] // Proceedings of the 12th Workshop on AI and Scientific Computing at Scale Using Flexible Computing Infrastructures. — Minneapolis, MN, USA : Association for Computing Machinery, 2022. — С. 9—17. — (FlexScience ’22). — DOI: 10.1145/3526058.3535454.

5. Automating Tiny ML Intelligent Sensors DevOPS Using Microsoft Azure / C. Vuppalapati [та ін.] // 2020 IEEE International Conference on Big Data (Big Data). — 2020. — С. 2375—2384. — DOI: 10.1109/BigData50022.2020.9377755.

6. *Bachinger F.*, *Zenisek J.*, *Affenzeller M.* Automated Machine Learning for Industrial Applications – Challenges and Opportunities // Procedia Computer Science. — 2024. — Т. 232. — С. 1701—1710. — DOI: https://doi.org/10.1016/j.procs.2024.01.168.

7. *Bodor A.*, *Hnida M.*, *Daoudi N.* Machine Learning Models Monitoring in MLOps Context: Metrics and Tools // International Journal of Interactive

Mobile Technologies (iJIM). — 2023. — Груд. — Т. 17, № 23. — pp. 125—139. — DOI: 10.3991/ijim.v17i23.43479.

8. Building Network Domain Knowledge Graph from Heterogeneous YANG Models / D. Yongqiang [та ін.] // Journal of Computer Research and Development. — 2020. — Т. 57, № 4. — С. 699—708. — DOI: 10.7544/issn1000-1239.2020.20190882.

9. *Calefato F.*, *Lanubile F.*, *Quaranta L.* A Preliminary Investigation of MLOps Practices in GitHub // Proceedings of the 16th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement. — Helsinki, Finland : Association for Computing Machinery, 2022. — С. 283—288. — (ESEM '22). — DOI: 10.1145/3544902.3546636.

10. Characterizing Machine Learning Processes: A Maturity Framework / R. Akkiraju [та ін.] // Business Process Management. Т. 12168 / за ред. D. Fahland [та ін.]. — Cham : Springer International Publishing, 2020. — С. 17—31. — (Lecture Notes in Computer Science). — ISBN 978-3-030-58666-9. — DOI: 10.1007/978-3-030-58666-9_2.

11. *Chen H.*, *Babar M. A.* Security for Machine Learning-based Software Systems: A Survey of Threats, Practices, and Challenges // ACM Comput. Surv. — New York, NY, USA, 2024. — Лют. — Т. 56, № 6. — DOI: 10.1145/3638531.

12. *Chrastina J.* Meta-synthesis of qualitative studies: background, methodology and applications // NORDSCI Conference proceedings. Т. 1. — Saima Consult Ltd, 2018. — (NORDSCI Conference). — DOI: 10.32008/nordsci2018/b1/v1/13.

13. *Cohen R.* Digital Strategy, Machine Learning, and Industry Survey of MLOps // Digital Strategies and Organizational Transformation. — 2023. — Гл. 8. С. 137—150. — DOI: 10.1142/9789811271984_0008. — URL: https://tinyurl.com/33z6zpd3.

14. *Czakon J.*, *Kluge K.* ML Experiment Tracking: What It Is, Why It Matters, and How to Implement It. — 05.2024. — URL: https://neptune.ai/blog/ml-experiment-tracking.

15. Ease.ML: A Lifecycle Management System for MLDev and MLOps / L. A. Melgar [та ін.] // 11th Conference on Innovative Data Systems Research, CIDR 2021, Virtual Event, January 11-15, 2021, Online Proceedings. — 2021. — URL: https://www.cidrdb.org/cidr2021/papers/cidr2021_paper26.pdf.

16. *Godwin R. C.*, *Melvin R. L.* Toward efficient data science: A comprehensive MLOps template for collaborative code development and automation // SoftwareX. — 2024. — Т. 26. — DOI: 10.1016/j.softx.2024.101723.

17. *Haller K.* Managing AI in the enterprise: Succeeding with AI projects and MLOps to build sustainable AI organizations. — 2022. — С. 1—214. — DOI: 10.1007/978-1-4842-7824-6.

18. *Kolltveit A. B.*, *Li J.* Operationalizing machine learning models: a systematic literature review // Proceedings of the 1st Workshop on Software Engineering for Responsible AI. — Pittsburgh, Pennsylvania : Association for Computing Machinery, 2023. — С. 1—8. — (SE4RAI '22). — DOI: 10.1145/3526073.3527584.

19. *Kreuzberger D.*, *Kühl N.*, *Hirschl S.* Machine Learning Operations (MLOps): Overview, Definition, and Architecture // IEEE Access. — 2023. — Т. 11. — С. 31866—31879. — DOI: 10.1109/ACCESS.2023.3262138.

20. *Lima A.*, *Monteiro L.*, *Furtado A. P.* MLOps: Practices, Maturity Models, Roles, Tools, and Challenges – A Systematic Literature Review // Proceedings of the 24th International Conference on Enterprise Information Systems - Volume 1: ICEIS. — INSTICC. SciTePress, 2022. — С. 308—320. — ISBN 978-989-758-569-2. — DOI: 10.5220/0010997300003179.

21. *Lwakatare L. E.*, *Crnkovic I.*, *Bosch J.* DevOps for AI – Challenges in Development of AI-enabled Applications // 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). — 2020. — С. 1—6. — DOI: 10.23919/SoftCOM50211.2020.9238323.

22. MLOps - Definitions, Tools and Challenges / G. Symeonidis [та ін.] // 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). — 2022. — С. 0453—0460. — DOI: 10.1109/CCWC54503.2022.9720902.

23. MLOps in Data Science Projects: A Review / C. Haertel [та ін.] // 2023 IEEE International Conference on Big Data (BigData). — 2023. — С. 2396—2404. — DOI: 10.1109/BigData59044.2023.10386139.

24. MLOps: A Taxonomy and a Methodology / M. Testi [та ін.] // IEEE Access. — 2022. — Т. 10. — С. 63606—63618. — DOI: 10.1109/ACCESS.2022.3181730.

25. *Neptune Labs.* MLOps Landscape in 2024: Top Tools and Platforms. — 2024. — URL: https://neptune.ai/blog/mlops-tools-platforms-landscape.

26. *Peltonen E., Dias S.* LinkEdge: Open-sourced MLOps Integration with IoT Edge // Proceedings of the 3rd Eclipse Security, AI, Architecture and Modelling Conference on Cloud to Edge Continuum. — Ludwigsburg, Germany : Association for Computing Machinery, 2023. — С. 67—76. — (ESAAM '23). — DOI: 10.1145/3624486.3624496.

27. SensiX++: Bringing MLOps and Multi-tenant Model Serving to Sensory Edge Devices / C. Min [та ін.] // ACM Trans. Embed. Comput. Syst. — New York, NY, USA, 2023. — Листоп. — Т. 22, № 6. — DOI: 10.1145/3617507. — URL: https://doi.org/10.1145/3617507.

28. *Singh P.* Systematic review of data-centric approaches in artificial intelligence and machine learning // Data Science and Management. — 2023. — Т. 6, № 3. — С. 144—157. — DOI: https://doi.org/10.1016/j.dsm.2023.06.001.

29. *Sipe T. A., Curlette W. L.* A meta-synthesis of factors related to educational achievement: a methodological approach to summarizing and synthesizing meta-analyses // International Journal of Educational Research. — 1996. — Т. 25, № 7. — С. 583—698. — DOI: 10.1016/S0883-0355(96)80001-2.

30. SliceOps: Explainable MLOps for Streamlined Automation-Native 6G Networks / F. Rezazadeh [та ін.] // IEEE Wireless Communications. — 2024. — С. 1—7. — DOI: 10.1109/MWC.007.2300144.

31. Software Engineering for Machine Learning: A Case Study / S. Amershi [та ін.] // 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). — 2019. — С. 291—300. — DOI: 10.1109/ICSE-SEIP.2019.00042.

32. *Sothilingam R.*, *Pant V.*, *Yu E. S. K.* Using i* to Analyze Collaboration Challenges in MLOps Project Teams // Proceedings of the 15th International iStar Workshop (iStar 2022) co-located with 41th International Conference on Conceptual Modeling (ER 2022), Virtual Event, Hyderabad, India, October 17, 2022. Т. 3231 / за ред. A. Maté, T. Li, E. J. T. Gonçalves. — CEUR-WS.org, 2022. — С. 1—6. — (CEUR Workshop Proceedings). — URL: https://ceur-ws.org/Vol-3231/iStar22%5C_paper%5C_1.pdf.

33. *Steidl M.*, *Felderer M.*, *Ramler R.* The pipeline for the continuous development of artificial intelligence models—Current state of research and practice // Journal of Systems and Software. — 2023. — Т. 199. — С. 111615. — DOI: 10.1016/j.jss.2023.111615.

34. Structure Learning and Hyperparameter Optimization Using an Automated Machine Learning (AutoML) Pipeline / K. Filippou [та ін.] // Information. — 2023. — Т. 14, № 4. — С. 232. — DOI: 10.3390/info14040232.

35. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews / M. J. Page [та ін.] // BMJ. — 2021. — Т. 372. — n71. — DOI: 10.1136/bmj.n71.

36. Unlabeled learning algorithms and operations: overview and future trends in defense sector / E. e Oliveira [та ін.] // Artificial Intelligence Review. — 2024. — Т. 57, № 3. — DOI: 10.1007/s10462-023-10692-0.

37. Who needs to know what, when?: Broadening the Explainable AI (XAI) Design Space by Looking at Explanations Across the AI Lifecycle / S. Dhanorkar [та ін.] // Proceedings of the 2021 ACM Designing Interactive Systems Conference. — Virtual Event, USA : Association for Computing Machinery, 2021. — С. 1591—1602. — (DIS '21). — DOI: 10.1145/3461778.3462131.

Додаток А.
Використання великої мовної моделі Claude 3 Sonnet для аналізу систематичних оглядів

Запити були створені 10.05.2024. До кожного запиту додавався файл у форматі PDF із текстом статті. Від чатботу очікувався звіт за поданим планом. Запити складались з універсальної та варіативних частин.

Універсальна частина запиту:

`Використовуючи доданий файл оглядової статті як джерело даних, напиши розгорнутий звіт про неї за таким планом. Використовуй українську для відповіді.`

Варіативні частини запиту:

1. `1. Year of publication.`
   `2. Research objective (paper aim).`
   `3. Research questions.`
2. `4. Information sources (databases, etc.)`
   `5. Inclusion criteria.`
   `6. Exclusion criteria.`
   `7. Quality criteria.`
3. `8. MLOps definition (if any).`
   `9. MLOps workflow stages (if any).`
4. `10. What frameworks and architectures facilitate MLOps?`
5. `11. What MLOps tools can be used to build ML pipelines for Continuous Deployment? What tools are used in the activities for operationalizing machine learning models?`
6. `12. What are the main features offered by MLOps tools?`
7. `13. How are machine learning models deployed in production environments?`

8. 14. What maturity models are used to assess the level of automation in deploying machine learning models?

9. 15. What roles and responsibilities are identified in the activities of operationalization of machine learning models?

10. 16. What challenges are encountered with regard to deploying machine learning models in production environments?

11. 17. What are the open issues, challenges and particularities in MLOps?
    18. What are the opportunities and future trends in MLOps? What are the current and future fields in which MLOps is thriving?

Додаток Б.
Результати аналізу систематичних оглядів

Таблиця Б.1

Результати аналізу систематичних оглядів [2; 3; 20].

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Рік публікації	2022	2022	2023
Мета дослідження	Виявити інструменти, які дозволяють створювати конвеєри MLOps для безперервного розгортання. Проаналізувати основні характеристики та функції цих інструментів MLOps, щоб надати всебічний огляд їхньої цінності.	Огляд існуючої літератури з метою виявлення практик, стандартів, ролей, моделей зрілості, викликів та інструментів, що використовуються для автоматизації діяльності з впровадження моделей машинного навчання в промислову експлуатацію (MLOps).	Основною метою цього систематичного огляду літератури є надати розуміння щодо впровадження методологій MLOps та AIOps як у промисловості, так і в академічному середовищі. Автори прагнуть висвітлити проблеми, можливості та майбутні тенденції в цих областях.

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Дослідницькі питання	Ми відповідаємо на на наступне основне питання дослідження: Які інструменти та можливості дозволяють розробникам створювати програмні системи з підтримкою ML? Яке було деталізовано на два підпитання: RQ1. Які інструменти MLOps можна використовувати для створення конвеєрів машинного навчання для безперервного розгортання? RQ2. Які основні функції пропонують інструменти MLOps?	RQ1: Як моделі машинного навчання розгортаються у виробничих середовищах? RQ2: Які моделі зрілості використовуються для оцінки рівня автоматизації розгортання моделей машинного навчання? RQ3: Які ролі та обов'язки визначено у діяльності з впровадження моделей машинного навчання? RQ4: Які інструменти використовуються у діяльності з впровадження моделей машинного навчання? RQ5: З якими викликами стикаються при розгортанні моделей машинного навчання у виробничих середовищах?	Q1: Які відкриті проблеми, виклики та особливості в MLOps та AIOps? Q2: Які можливості та майбутні тенденції в MLOps? Q3: Які можливості та майбутні тенденції в AIOps? Q4: Які платформи та архітектури сприяють MLOps та AIOps? Q5: Які поточні та майбутні галузі, в яких процвітають MLOps та AIOps?

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Джерела інформації	Google Scholar - для пошуку наукової літератури, таких як журнали, книги та дисертації. Пошук Google - для пошуку так званої "сірої" літератури, такої як блог-пости, сайти розробників, вебінари, GitHub репозиторії та відео на YouTube. Використання як академічних (білої літератури), так і професійних (сірої літератури) джерел дозволило авторам досліджувати MLOps з різних точок зору - теоретичної та практичної.	Автоматичний пошук проводився в наступних електронних базах даних досліджень: ACM Digital Library, IEEE Xplore, ScienceDirect та SpringerLink.	Для пошуку релевантних статей автори використали декілька баз даних та репозиторіїв, включаючи arXiv, Springer, IEEE. Однак основним джерелом була база даних Scopus від видавництва Elsevier, оскільки вона містить метадані та анотації багатьох публікацій.

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Критерії включення	Дослідження обговорює компоненти мінімального наскрізного робочого процесу MLOps. Дослідження обговорює практику MLOps або застосунків на основі машинного навчання. Дослідження стосується реалізації інструменту(ів) MLOps. Дослідження описує досвід, думки або практики щодо конвеєрів MLOps.	Дослідження, що стосуються машинних операцій з навчанням (MLOps) загалом. Дослідження, що оцінюють життєвий цикл рішень машинного навчання. Дослідження, що стосуються моделей зрілості процесу машинного навчання. Дослідження, що аналізують ролі та обов'язки, залучені до розробки та впровадження рішень машинного навчання. Дослідження, що охоплюють інструменти для розгортання рішень машинного навчання. Дослідження, що визначають виклики для розробки та розгортання моделей машинного навчання.	До аналізу включалися статті, опубліковані між 2018 та 2023 роками, визначені за пошуковими запитами, що містять нові ідеї та тісно пов'язані з темою MLOps та AIOps.

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Критерії виключення	Дослідження не надає деталей щодо проектування або реалізації інструменту(ів) MLOps. Дослідження пропонує лише проектування певного компоненту конвеєрів машинного навчання. Дослідження не надає або не посилається на деталі щодо автоматизації машинного навчання. Дослідження посилається на комерційні платформи, які пропонують застосунки MLOps для просування своїх послуг розробки та розгортання.	Дослідження, не опубліковані англійською мовою. Дослідження, що стосуються застосування моделей машинного навчання. Короткі статті або постери. Дослідження, не пов'язані з операціями машинного навчання. Дослідження, до контенту яких немає доступу. Статті, що не стосуються дослідницьких питань.	Виключалися публікації не англійською мовою, відкликані публікації, публікації невідповідних видавництв, матеріали за передплатою без доступу та статті з недостатньою кількістю цитувань (залежно від року публікації).

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Критерії якості	Репозиторій має бути не менше 100 зірок Відео на YouTube має бути переглянуто не менше 1000 разів	Чи повідомляє дослідження однозначні відкриття на основі доказів та аргументів? Чи представляє дослідження дослідницький проект, а не експертну думку? Чи повністю описано контекст, що аналізується, в дослідженні? Чи чітко визначені цілі дослідження? Чи належним чином валідовано результати дослідження?	Всебічний огляд літератури та виявлення прогалин Перевірка результатів на прикладі використання Кількість дослідницьких питань, що розглядаються Публікація за відкритою ліцензією Тип публікації (журнал/інше) Публікація в журналі з високим імпакт-фактором Кількість цитувань
Визначення MLOps (якщо є)	MLOps - це практика, яка допомагає моделювати, розробляти та експлуатувати життєвий цикл машинного навчання, спираючись на принципи та практики DevOps.	Набір практик та принципів для операціоналізації рішень науки про дані, який використовується для автоматизації впровадження моделей машинного навчання в операційне середовище	MLOps використовує машинне навчання, DevOps та інженерію даних з метою переведення систем машинного навчання у виробництво, полегшуючи створення машинних продуктів.

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Етапи робочого процесу MLOps (якщо є)	Вилучення даних для інтеграції Аналіз даних Очищення даних, трансформація та інжиніринг ознак для розділення даних Навчання моделі Валідація моделі для оцінки якості Розгортання моделі в цільових середовищах Моніторинг моделі	Збір даних Трансформація даних Безперервне навчання моделі Безперервне впровадження моделі Представлення результатів Моніторинг рішень машинного навчання	Управління даними Розподілене навчання Розгортання Моніторинг Повторне навчання Акцентується необхідність керування життєвим циклом та ключовими компонентами ІІ-додатків за допомогою спеціальних платформ та інструментів.

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Фреймворки та архітектури, що сприяють впровадженню MLOps	Конвеєри безперервного навчання, розгорнуті через CI/CD Платформи орхестрації TensorFlow Extended (TFX) Хмарні платформи машинного навчання	MLflow Kubeflow Polyaxon Comet.ml Kafka-ML MLModelCI	HPC Хмарні обчислення Платформи Edge/IoT Безсерверні архітектури Фреймворки для інтеграції Методи автоматичного маркування даних Проактивне управління інцидентами Платформи для оркестрації Семантично-підсилені конвеєри Архітектури для розподіленого навчання та розгортання Фреймворки для моніторингу Мови програмування Контейнеризовані рішення Програмні засоби AutoML Використання API Глибоке навчання та нейромережі

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Інструменти MLOps для створення конвеєрів машинного навчання та операціоналізації моделей	Хмарні платформи машинного навчання (AWS SageMaker, AzureML, Google AI Platform) Платформи оркестрації (Apache Airflow, Jenkins, Kubeflow, MLflow, Polyaxon, Seldon Core, Valohai) Фреймворки конфігурації (TensorFlow Extended, Gitlab)	MLflow - ця відкрита платформа має компоненти, такі як MLflow Projects та MLflow Model Registry Kubeflow Kafka-ML MLModelCI	Контейнеризовані рішення (наприклад, Docker) Серверлес обчислення (AWS Lambda, Azure Functions тощо) Інструменти безперервної інтеграції/безперервного розгортання (CI/CD) Моніторинг процесів та подій Автоматизація MLOps за допомогою AutoML Контейнеризація для пакування залежностей моделей. Технології API для розгортання як веб-сервіс.
Основні функції, що надаються інструментами MLOps	Загальні функції, пов'язані з усіма фазами конвеєрів машинного навчання Функції управління даними Функції управління моделями	Відстеження експериментів Пакування та версіонування моделей Управління проектами ML Реєстр моделей Безперервна інтеграція та доставка Моніторинг моделей Керування гіперпараметрами Портативність	Версіонування даних, моделей та коду Оркестрація та автоматизація потоку роботи Моніторинг процесів та подій Інтеграція з хмарною та периферійною інфраструктурою Контейнеризація Автоматизація MLOps за допомогою AutoML

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Способи розгортання моделей машинного навчання у виробничих середовищах	Хмарні платформи машинного навчання (AWS SageMaker, AzureML, DotScience, Google AI Platform) Платформи оркестрування (Apache Airflow, Jenkins, Kubeflow, MLflow, Polyaxon, Seldon Core, Valohai) TensorFlow Extended (TFX)	MLOps розглядається як набір практик та принципів для операціоналізації Деякі інструменти MLOps, такі як MLflow, Kubeflow та Kafka-ML Визначається роль "Deployment Lead"	Розгортання у хмарі Використання хмарних обчислювальних ресурсів Забезпечення ізоляції Гібридні підходи Розгортання на периферії/крайових пристроях Розгортання безпосередньо на пристроях IoT та мобільних пристроях. TensorFlow Lite та Core ML. Подолання обмежень Контейнеризовані розгортання Упакування моделей Docker. Серверлес архітектури Розгортання функцій ML як сервісу Зменшення витрат Розгортання через API

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Моделі зрілості для оцінки рівня автоматизації розгортання моделей машинного навчання	Підтримка неперервної інтеграції та неперервного розгортання (CI/CD) Можливість автоматичного налаштування Повна автоматизація процесів управління моделями	Модель зрілості, запропонована Amershi та ін. [31] Dhanorkar та ін. [37] класифікують організації за трьома рівнями зрілості Lwakatare, Crnkovic, Bosch [21] описують п'ять етапів вдосконалення практик розробки Akkiraju та ін. [10] пропонують адаптацію Моделі зрілості можливостей (CMM)	В статті не визначено конкретних моделей зрілості для оцінки автоматизації розгортання моделей машинного навчання, але наголошується на важливості адаптації практик розробки програмного забезпечення до сфери машинного навчання.
Ролі та обов'язки, визначені в діяльності з операціоналізації моделей машинного навчання	Наукові співробітники з даних - відповідають за розробку та навчання Інженери з даних - відповідають за вилучення, обробку, перетворення та забезпечення якості даних Інженери DevOps - відповідають за автоматизацію процесів розгортання та управління операційними середовищами Менеджери продуктів та бізнес-зацікавлені сторони - забезпечують вимоги до моделей та беруть участь у прийнятті рішень	Фахівець з предметної області - має глибокі знання предметної галузі Науковець з обчислювальних наук та інженер - має високі технічні навички Науковець з машинного навчання та інженер - відповідає за проектування Провайдер - керує постачанням даних Менеджер - оцінює моделі Розробник додатків - розробляє додатки Керівник розгортання - оцінює аспекти	Розробники програмного забезпечення Фахівці з даних/науковці з даних Операційні інженери Експерти з предметної галузі Керівництво/зацікавлені сторони

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Виклики, що виникають при розгортанні моделей машинного навчання у виробничих середовищах	Складність управління Забезпечення узгодженості Інтеграція різних інструментів Автоматизація всіх етапів Моніторинг продуктивності моделей Масштабування інфраструктури Забезпечення безпеки	Інтеграція розробки Впровадження практик MLOps/AIOps Для моделей машинного навчання потрібен моніторинг Визначення інфраструктурних компонентів Розгортання та управління версіями	Управління життєвим циклом ML Розрив між інженерією та навичками машинного навчання Управління даними Розподілене та паралельне виконання - Різноманітність обчислювальної інфраструктури Моніторинг Роз'яснюваність
Виклики, що виникають при розгортанні моделей машинного навчання у виробничих середовищах	Відсутність стандартизації Забезпечення портативності Налаштування та інтеграція Повна автоматизація Зрозумілість	Інтеграція процесів розробки Впровадження практик MLOps Необхідно виходити за рамки аналізу прототипів моделей Ретельний моніторинг Визначення інфраструктури Адресація питань масштабованості Необхідність версіонування Автоматизація Управління життєвим циклом Інтеграція з DevOps	Брак кваліфікованих кадрів Проблеми з управлінням даними Складність оркестрації Різнорідність апаратного забезпечення Необхідність безперервного моніторингу Відсутність роз'яснюваності Проблеми масштабування та забезпечення продуктивності

Продовження на наступній сторінці

Продовження таблиці Б.1

Об'єкт порівняння	**Огляд Recupito та ін. [3]**	**Огляд Lima, Monteiro, Furtado [20]**	**Огляд Diaz-de-Arcaya та ін. [2]**
Можливості, майбутні тенденції та сфери застосування MLOps	Стандартизація практик та інструментів Покращення підтримки середовищ. Удосконалення автоматизації та технологій Інтеграція MLOps з DevOps та DevSecOps. Підвищення уваги до управління даними Галузі застосування Фінансові послуги та банкінг Охорона здоров'я та біотехнології Виробництво та Інтернет речей Роздрібна торгівля та електронна комерція Телекомунікації Транспорт та логістика	Очікується зростання попиту на інструменти та платформи MLOps Галузі, де MLOps активно розвивається - фінанси, охорона здоров'я, промисловість, роздрібна торгівля, транспорт та логістика. Можливий розвиток спеціалізованих моделей зрілості Інтеграція MLOps з концепціями DevSecOps, MLSecOps Поява нових ролей та компетенцій	Залучення бізнес-підрозділів Більша увага до життєвого циклу ML Кращі практики управління даними Використання нових апаратних платформ Застосування контейнерів, безсерверних технологій Розробка інструментів версіонування Галузі, де MLOps процвітає: Традиційна промисловість Інноваційні галузі Академічні дисципліни Зв'язок та мережеві технології Охорона здоров'я наукова діяльність.