УДК 681.518, 004.896 DOI: 10.31673/2412-4338.2023.036679

Кузьміч М. Ю. Гордієнко Т. Б.
Державний університет інформаційно-комунікаційних технологій

ЗАСТОСУВАННЯ ІНСТРУМЕНТУ KUBEFLOW ДЛЯ ІНТЕГРАЦІЇ МАШИННОГО НАВЧАННЯ І ШТУЧНОГО ІНТЕЛЕКТУ В БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТАХ

Анотація: *На сучасному етапі розвитку інформаційних технологій машинне навчання (Mashine Learning, ML) та штучний інтелект (Artificial Intelligence, AI) стають одними із головних інструментів для розв'язання складних прикладних задач у різних сферах діяльності. Для розробки, тестування та підтримки інфраструктури системи із даними застосовуються різні процеси та технології. Особливо актуальним на сьогодні є застосування інструментів з інтеграції ML та AI в керування безпілотними літальними апаратами (БпЛА).*

Зроблено огляд концепції ML та процесів (Machine Learning and Operation, MLOps), що являє собою набір технік для імплементації та автоматичної неперервної інтеграції, а також доставки на продуктове середовище та навчання моделей. Концепцію MLOps розглянуто в розрізі інструментів Kubeflow, що працює на платформі Kubernetes. Досліджено можливості використання сучасних MLOps рішень для покращення процесів розробки інформаційних систем ML. Спроектовано інформаційну систему на базі AI із можливістю постійного навчання. Наведена концепція використання MLOps конвеєру для розв'язання прикладної задачі класифікації об'єктів із відео розвідувальних БпЛА.

Перевірено результати експлуатації моделі в арсеналі Kubeflow із використанням таких факторів покращення як: швидкість розробки, імплементації змін, зменшення часу на пошук проблем, відновлення після глобальних перебоїв, зменшення кількості помилок в моделі. Для практичного аналізу розгорнуто модель, яка перебуває у відкритому доступі, в Kubeflow кластер за допомогою маніфеста застосунку Seldon Core Serving.

Проведені дослідження показали, що Kubeflow складається із набору різноманітних open source компонентів, що мають високий рівень інтеграції між собою через платформу Kubernetes. При цьому Kubeflow надзвичайно ефективно використовує патерн Kubernetes операторів для об'єктів ML. Продемонстровано, що написання коду моделі це невелика частина серед задач ML, що впливає на необхідність автоматизації. Сформовано концепцію повноцінного інформаційного рішення на базі конвеєру неперервної інтеграції, що є фундаментом реалізації концепції постійного навчання. Представлення абстракцій у вигляді окремих ресурсів платформи дозволяє зменшити поріг входу для кінцевого користувача.

Ключові слова: *Kubeflow, MLOps, машинне навчання, штучний інтелект, постійне навчання, безпілотний літальний апарат.*

Kuzmich M. Yu., Gordiyenko T. B.
State University of Information and Communication Technologies

APPLICATION OF THE KUBEFLOW TOOL FOR THE INTEGRATION OF MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE IN UNMANNED AERIAL VEHICLE

Abstract: *At the current stage of information technology development, machine learning (ML) and artificial intelligence (AI) are becoming one of the main tools for solving complex applied*

problems in various fields of activity. Various processes and technologies are used for develop, test, and maintain the infrastructure of the data system. The application of tools for the integration of ML and AI in the management of unmanned aerial vehicles (UAVs) is especially relevant today.

An overview of the ML concept and processes (Machine Learning and Operation, MLOps) was made, which is a set of techniques for implementation and automatic continuous integration, as well as delivery to the product environment and model learning. The concept of MLOps is considered in terms of Kubeflow tools, which work on the Kubernetes platform. The possibilities of using modern MLOps solutions to improve the development processes of ML information systems have investigated. An AI-based information system with the possibility of continuous learning has designed. The concept of using the MLOps pipeline to solve the applied problem of classifying objects from the video of reconnaissance UAVs was presented.

The results of the operation of the model in the Kubeflow arsenal have been checked using such improvement factors as: speed of development, implementation of changes, reduction of time to search for problems, recovery after global interruptions, reduction of the number of errors in the model. A publicly available model was deployed in a Kubeflow cluster using the Seldon Core Serving application manifest for practical analysis.

The conducted research showed that Kubeflow consists of a set of various open source components that have a high level of integration with each other through the Kubernetes platform. At the same time, Kubeflow uses the Kubernetes pattern of operators for ML objects extremely efficiently. It has shown that writing model code is a small part of ML tasks, which affects the need for automation. The concept of a full-fledged information solution based on the continuous integration pipeline, which is the foundation of the implementation of the concept of continuous learning, has formed. Representing abstractions in the form of separate platform resources allows you to reduce the entry threshold for the end user.

Keywords: *Kubeflow, MLOps, machine learning, artificial intelligence, continuous learning, unmanned aerial vehicle.*

1. Вступ

На сучасному етапі розвитку інформаційних технологій наука про дані, аналітика даних, машинне навчання (Mashine Learning, ML) стають одними із головних інструментів для розв'язання складних прикладних задач у різних сферах діяльності. ML є одним із методів функціонування штучного інтелекту (Artificial Intelligence, AI), а саме – практичної реалізації його можливостей шляхом створення алгоритмів для виявлення закономірностей під час аналізу великих даних, та їх подальше використання для самонавчання [1–3]. Загалом, обидві галузі пов'язані з використанням обчислювальних і математичних підходів для покращення прийняття рішень.

Швидкий прогрес у різних галузях досліджень ML (таких як комп'ютерний зір, розуміння природної мови та рекомендаційні системи AI) мотивують більшість компаній до інвестицій у свої команди з обробки даних і можливості ML, щоб розробити прогнозні моделі, які можуть принести бізнес-цінність своїм користувачам. Особливо це питання є актуальним в умовах дослідження війни безпілотних літальних апаратів (БпЛА) за допомогою AI як варіанту надання військовим тактичної переваги над противником та зменшення ризику для людських життів. Використовуючи ML з алгоритмами AI, БпЛА можуть швидко ідентифікувати загрози, реагувати на них, вести спостереження та розвідку. Незважаючи на юридичні та етичні дебати щодо застосування БпЛА із використанням AI, варто зауважити, що ця технологія стає все більш складною, що призводитиме до більш точного націлювання і, відповідно, зменшення ризиків жертв серед цивільного населення та ймовірності супутніх збитків. На сьогодні актуальним питанням є застосування інструментів з інтеграції ML та AI в керування та тактику бою БпЛА.

Для розробки, тестування та підтримки інфраструктури системи із великими даними застосовуються різні процеси та технології.

Запорукою ефективності ML стали такі чинники, як:

– доступ до великих наборів даних;

– невисока ціна обчислювальних ресурсів;

– готові сервіси на основі хмарних технологій;

– швидкий прогрес у сферах AI, такі як комп'ютерний зір, розуміння природної мови та рекомендаційні системи.

Разом із тим у розробці ML розробки існує велика кількість додаткових складнощів, через це безліч ML систем зазнає невдачі на ранніх стадіях розробки і не досягає продуктового середовища.

Для вирішення цих проблем використовують концепцію спільного підходу до процесів машинного навчання та експлуатації (Machine Learning and Operation, MLOps), яка є набором технік для імплементації та автоматичної неперервної інтеграції, а також доставки на продуктове середовище та навчання моделей. MLOps є інженерною культурою та практикою ML, спрямованою на уніфікацію розробки ML системи (Dev) і їх експлуатацію (Ops). Практика MLOps означає, що ви виступаєте за автоматизацію та моніторинг на всіх етапах створення системи ML, включаючи інтеграцію, тестування, випуск, розгортання та управління інфраструктурою.

Фахівці з обробки даних можуть реалізувати та навчити модель ML із прогнозованою ефективністю на офлайн-затриманому наборі даних, маючи відповідні навчальні дані для свого сценарію використання. Однак справжнє завдання полягає не в побудові моделі ML, а в побудові інтегрованої системи ML і її безперервній експлуатації у виробництві. Завдяки довгій історії виробничих служб ML компанією Google виявлено, що під час експлуатації систем на основі ML у виробництві може бути багато підводних каменів. Деякі з них узагальнено в роботі «Машинне навчання: кредитна картка з високими відсотками технічного боргу» [4].

Тому розробники програмного забезпечення все більше віддають перевагу концепції контейнеризації. Компанія Google є однією з небагатьох, якій необхідно керувати розгортанням і розробкою великої кількості компонентів сервісу на сотнях тисяч серверів. В основі концепції контейнеризації та мікросервісів була розроблена розподілена система керування контейнерами з відкритим кодом під назвою Kubernetes. Останній може не лише підтримувати повну незалежність програм, але й покращувати використання апаратних ресурсів, тому його широко використовують більшість установ.

Таким чином концепцію MLOps доцільно розглядати в розрізі інструментів Kubeflow – cloud native системи з відкритим кодом, що працює на платформі Kubernetes. Проект Kubeflow розроблений для спрощення розгортання робочого циклу ML систем в Kubernetes. Завдання останнього не дублювати інші служби, а забезпечити простий спосіб розгортання найкращих систем з відкритим кодом для ML на різних інфраструктурах, будь то приватні сервери на виробництві або публічні хмарні рішення. Тобто там, де працює Kubernetes, може працювати Kubeflow, що робить його гнучким та портативним.

2. Аналіз літературних даних і постановка проблеми

ML довело свою ефективність в сучасних реаліях, а інвестиції в розробку якісних, передбачуваних моделей виросли в рази. Процес розробки моделі по своїй суті відповідає конвеєру на виробництві, як ланка послідовних або паралельних етапів [1–3]. Концепція MLOps – це набір технік для імплементації та автоматичної неперервної інтеграції, доставки та навчання моделей, яка охоплює основні патерни та антипатерни, на основі яких будують успішний конвеєр автоматизації розробки ML систем. Ефективність використання підходу MLOPs було розглянуто і доведено в багатьох роботах, де було формалізовано основні вимоги до ML систем [4–5].

Але разом із можливостями приходять додаткові складнощі. Дана проблема наряду із іншими, була розкрита у роботі, написаній працівниками Google щодо невидимого технічного боргу в ML системах [6]. Слід зазначити, що хоч термін MLOps не фігурує в згаданій статті, але основні патерни та антипатерни, на основі яких будуються успішний

конвеєр актуальні і зараз. Сучасні ML системи складаються із великої кількості компонентів. ML код є тільки невеликою частиною цієї системи. Для повноцінної роботи ML систем потрібна складна та обширна інфраструктура.

Опис концепцій ML для промислової інженерії та дослідження операцій з використанням реальних програм у вигляді курсів для слухачів розглянуто у роботі [7]. Такі дослідження підвищують інтерес слухачів до ML та оцінки реального впливу, який може мати аналітика. Це сприяє розвитку практичних навичок, а у поєднанні ML та дослідження операцій може бути цінним доповненням до навчальних програм. Однак, наразі обмежена наявність освітньої дослідницької літератури про навчання щодо ML, яку можна використати для розробки курсів, особливо для спеціальностей, не пов'язаних з комп'ютерними науками. Цим проблемам приділено увагу у роботах [8, 9].

У [10] проведені дослідження того як слухачі формують концепцію свого розуміння методу ML без нагляду після взаємодії з інтерактивними візуалізаціями, розробленими з використанням підходу обчислювального когнітивного учнівства, для інтеграції знань концепцій Data Science. А також досліджено як можливості інтерактивної візуалізації підтримують або перешкоджають інтеграції знань слухачів у ML навчання без нагляду. Однак дане дослідження не робить прив'язки до сучасних ML систем розробки, що не дозволяє нам оцінити його із практичної точки зору. Було б доцільно оцінити одну із поточних ML систем (Kubeflow, MLFlow) з точки зору легкості сприйняття для учнів або слухачів.

Завдяки зростаючій доступності даних і досягненнями у ML та методології оптимізації аналітика даних ширше використовується для вирішення проблем управління операціями. Поширений підхід до вирішення реальних проблем операційного менеджменту за алгоритмом «передбачити, потім оптимізувати» розглянуто в [11, 12]. У такому підході ключові параметри оптимізаційної моделі прогнозуються за допомогою ML до або одночасно з вирішенням оптимізаційних моделей. У роботах [11, 12] розглянуто застосування аналітики даних для управління операціями в трьох основних сферах – управління ланцюгом поставок, управління доходами та операції з охорони здоров'я. Однак у цих працях не розглядаються будь-які конкретні рекомендації щодо покращення або зміни моделі, наприклад, як реакцію на зміну ланцюгів поставок.

У [13] розглянуті основні концепції, спільні для багатьох алгоритмів ML, зокрема налаштування гіперпараметрів, як самі по собі є важливими цілями навчання, незалежно від алгоритмів ML. Обговорено типи ML, їхні параметри та гіперпараметри, навчання моделі, валідація та тестування. При цьому автори наголошують, що незважаючи на те, що постійно докладаються зусилля, щоб покращити ML алгоритми та зробити їх більш зрозумілими та інтерпретованими, роль викладачів полягає в покращенні розуміння людиною ML алгоритмів.

У роботі [14] запропонована структура конвеєра оновлення моделі та керування для моделей AIOps у системах мікросервісів. Це зроблено з метою навчання, інкапсулювання та розгортання моделі, а також для спрощення процесу. Для забезпечення попередньої реалізації фреймворку та перевірки розроблений прототип системи на основі Kubernetes і Gitlab. Процес пакування та розгортання автоматизовано за допомогою технології безперервної інтеграції. Ця робота є корисним ресурсом для побудови інтегрованої та оптимізованої системи керування моделлю AIOps. Однак, у запропонованій технології присутній недостатній рівень інтегрованості між ланками та низький рівень використання абстракцій Kubeflow для спрощення моделі. Наприклад, замість написання Dockerfile для моделі можна використати компонент seldon core. Також у роботі не було розкрито концепцію автоматичного донавчання моделі. Адже Kubeflow конвеєр може запускатись по тригеру, яким виступає наявність нових даних для навчання. Ці зміни додали б описаній ML системі більше універсальності та портативності, оскільки вона автоматично адаптувалась під конкретне середовище мікресовервісів і не потребувала ручного донавчання.

Компонент конвеєра, включений у Kubeflow, може створити цілий робочий процес ML. Однак цей компонент усе ще залежить від Google Cloud Platform, тому недостатньо дружній для розробників, які не мають умов для його оренди. У роботі [15] досліджені проблеми конвеєра Kubeflow, перевірено здійсненність рішення незалежності Kubernetes від Google Cloud Service на прикладі підтримки програми ML на основі Pytorch. Це рішення дає можливість науковцям з обробки даних створювати програми глибокого навчання на основі Pytorch у розробці розподілених систем на основі Kubernetes. Запропонований підхід забезпечує стабільну та безперервну роботу програми та розумне планування на кластері, що містить різнорідні обчислювальні ресурси.

У роботі [16] запропонована структура життєвого циклу ML (MLife) – циклічного процесу створення ефективної системи ML. Структура базується на тому факті, що потік даних у MLife є замкнутим циклом, керованим невдалими випадками. Вони найбільше впливають на продуктивність моделі ML, однак також забезпечують найбільшу цінність для подальшої розробки моделі ML. Це дозволяє підприємствам швидко відстежувати їхні можливості ML. Однак, запропонована структура MLife підходить лише для систем ML на їх ранніх стадіях.

У роботі [17] висвітлено дослідження використання БпЛА разом із ML та його алгоритмами в різних областях і регіонах. Автори визначили, що поєднання БпЛА і ML сприяло покращенню моніторингу в режимі реального часу, збору та обробці даних, а також прогнозуванні в комп'ютерних/бездротових мережах, розумних містах, військовій сфері, сільському господарстві тощо. У дослідженні [18] здійснена спроба визначити набір показників для оцінки робочого навантаження на оператора БпЛА в реальному часі для інформування алгоритму динамічного завдання. Для цього було задіяно 20 учасників через симуляцію управління декількома БпЛА зі змінною складністю. Визначені показники були використані як функції в алгоритмі ML для прогнозування робочого навантаження на оператора за кожні 60 секунд. Однак, в цих роботах не деталізовані питання щодо принципів побудови, функціонування та навчання інформаційної системи для використання у БпЛА.

Проведений аналіз показав необхідність розгляду концепції MLOps в розрізі інструментів Kubeflow із використанням низки визначених факторів покращення, зокрема таких як швидкість розробки, зменшення кількості помилок в моделі, часу на пошук проблем тощо. Можливості сучасних MLOps рішень можуть сприяти покращенню процесів розробки інформаційних систем ML, зокрема з алгоритмами штучного інтелекту та можливістю застосування в БпЛА.

3. Мета і задачі дослідження

Метою дослідження є покращення процесів розробки інформаційних систем машинного навчання. Для виконання цієї мети доцільним є використання сучасних MLOps рішень, таких як Kubeflow.

Для досягнення мети розв'язуються такі наукові задачі:

– дослідження можливості використання сучасних MLOps рішень для покращення процесів розробки інформаційних систем машинного навчання з алгоритмами штучного інтелекту та можливістю застосування в безпілотних літальних апаратах;

– перевірка результатів експлуатації моделі в арсеналі Kubeflow із використанням таких факторів покращення: швидкість розробки, імплементації змін, зменшення часу на пошук проблем, відновлення після глобальних перебоїв, зменшення кількості помилок в моделі.

4.1 Матеріали та методи дослідження роботи конвеєра машинного навчання

Під час роботи з ML визначають етапи, які можуть бути виконані як вручну, так і автоматично.

1. Збір даних: пошук та вибір релевантних даних із різних джерел для виконання ML задач.

2. Аналіз даних: проведення розвідувального аналізу – попередній експрес-аналіз даних шляхом їх перетворення та/або представлення у зручному вигляді: графічному, табличному, схемі, діаграмі і т. д.

Результатом буде наступне:

– розуміння структури та характеристик даних, що очікуються моделлю;

– визначення етапів підготовки даних та розробки основних особливостей (feature), необхідних для моделі.

3. Обробка даних: дані мають бути готові для виконання над ними задач машинного навчання. Відбувається їх очищення та розподіл на дані для тренування, валідацію та тестування.

4. Тренування моделі: інженери використовують різноманітні алгоритми та оброблені дані для тренування моделі. Додатково використовують алгоритми для підбору гіперпараметрів для покращення точності моделі.

5. Оцінка моделі: модель оцінюється на основі наявних даних. Результатом є набір метрик з якості моделі.

6. Валідація моделі: перевірка моделі на допустимість для розгортання в потрібне оточення.

7. Експлуатація моделі: перевірена модель розгорнута в кінцевому оточенні. Розгорнута модель може мати наступний вигляд:

– мікросервіс із REST API, що опрацьовує прогнози в режимі реального часу;

– вбудована модель в кінцевий пристрій;

– частина в складній системі прогнозування.

8. Моніторинг моделі: точність та швидкість прогнозування моделі вимірюється для врахування в наступних ітераціях розробки.

Найпростіший MLOps конвеєр зазвичай складається із таких етапів, як: підготовка даних, навчання моделі, валідація моделі і відповідно обслуговування/запуск кінцевої моделі.

Kubeflow – це cloud-native відкритий ресурс-платформа (open-source) для машинного навчання, що розроблена компанією Google [19]. Особливістю цієї платформи є високий рівень інтеграції із системою оркестрації Kubernetes та великий набір інструментарію для ML задач.

Kubeflow розробники організовані в робочі групи із відповідними репозиторіями, що фокусуються на специфічних ML задачах: автоматичне машинне навчання; розгортання; маніфести – файли конфігурацій Kubernetes; Jupiter notebook; конвеєри; тренування.

Фази конвеєра розробки ML рішень в платформі Kubeflow: а) фаза експериментів, б) фаза експлуатації, зображено на рис. 1.

В дослідженні зроблено огляд основних компонентів Kubeflow і їх доцільність при створенні конвеєра. Для огляду обрано версію Kubeflow 1.12, яка була найновішою на момент написання статті. Процес встановлення є відносно простий, оскільки створено застосунок інтерфейс командного рядка (Command line interface, CLI) для цієї потреби. Основною функціональною задачею CLI застосунку є формування маніфестів для Kubernetes. Після інсталяції компоненти будуть доступні в просторі імен (namespace) kubeflow. В платформі наявна велика кількість інтегрованих компонентів. Однак в роботі розглянуто тільки ті, що мінімально необхідні для побудови довершеного конвеєра.

Використання готового CLI застосунку значно спрощує процес інсталяції, оскільки дозволяє уникнути ручного редагування масивного конфігураційного файлу, зменшуючи поріг входу для користувачів.

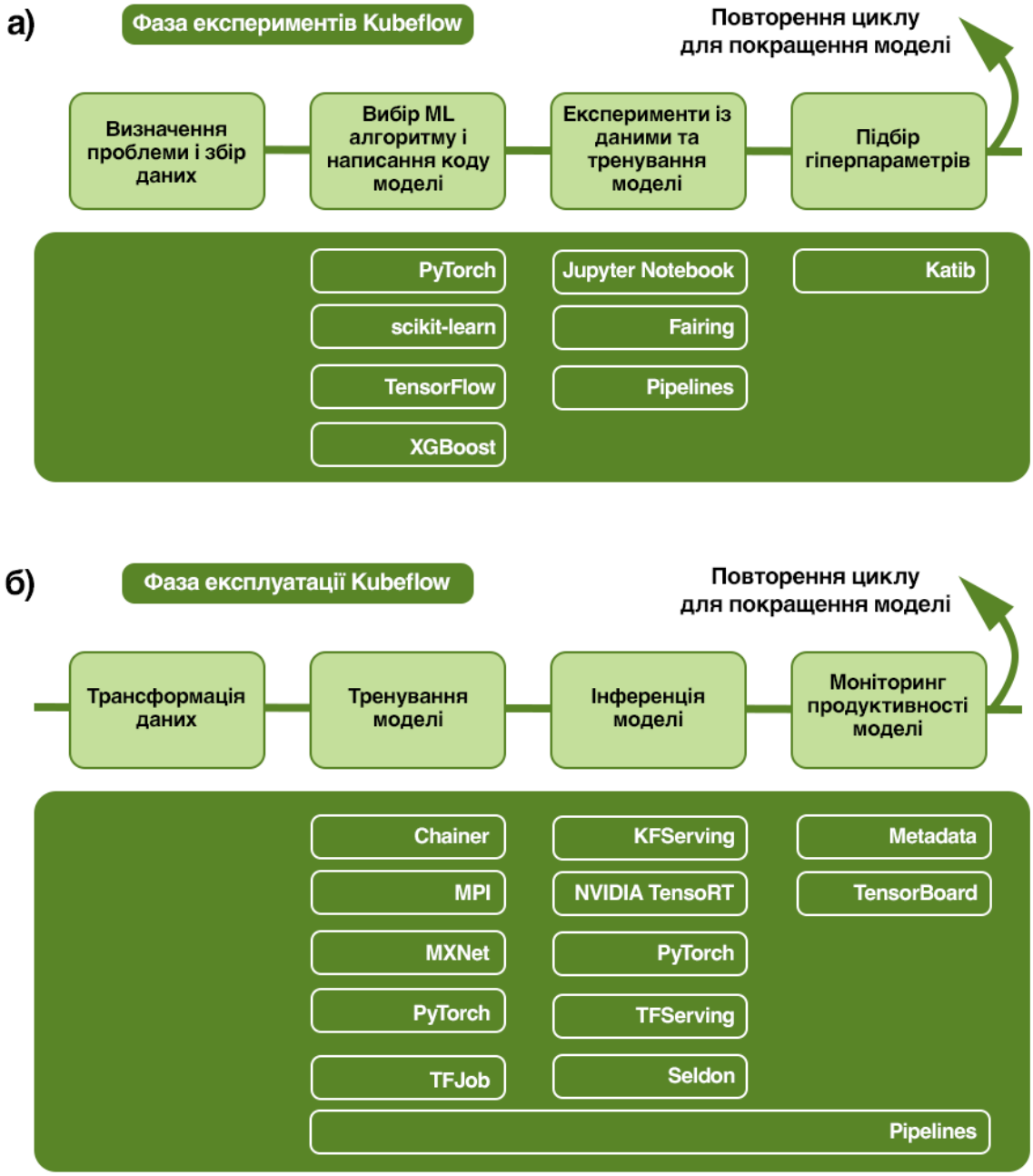

Рис. 1. Фази конвеєра розробки ML рішень в платформі Kubeflow

Список інстальованих нами компонентів Kubeflow наступний: admission-webhook-bootstrap-stateful-set-0, admission-webhook-deployment-5d9ccb5696-b4k2n, alb-ingress-controller-7dcf69ff67-w4wxl, application-controller-stateful-set-0,argo-ui-684bcb587f-rq4j9, cache-deployer-deployment-6667847478-p516t, cache-server-bd9c859db-svhq8, centraldashboard-895c4c768-r4ssh, jupyter-web-app-deployment-54758f7bc4-g2svl, katib-controller-75c8d47f8c-9nr9c, katib-db-manager-6c88c68d79-fbnzt, katib-mysql-858f68f588-zz7dn,katib-ui-68f59498d4-pr15s, kfserving-controller-manager, kubeflow-pipelines-profile-controller-69c94df75b-zfwff, metacontroller-0, metadata-db-757dc9c7b5-6xvjw, metadata-envoy-deployment-6ff58757f6-dkj64, metadata-grpc-deployment-76d69f69c8-bwp8b, metadata-writer-6d94ffb7df-7d4lg, minio-66c9cd74c9-n6659, ml-pipeline-54989c9946-m6gq8, ml-pipeline-persistenceagent-7f6bf7646-x29rq, ml-pipeline-scheduledworkflow-66db7bcf5d-5sb6f, ml-pipeline-ui-756b58fb-tcvtr, ml-pipeline-viewer-crd-58f59f87db-g29f9, ml-pipeline-visualizationserver-6f9ff4974-jgcwr, mpi-operator-77bb5d8f4b-68z2f , mxnet-operator-68b688bb69-xrr24, mysql-7694c6b8b7-ghbhv,

notebook-controller-deployment-58447d4b4c-ndqkt, nvidia-device-plugin-daemonset-dt792, nvidia-device-plugin-daemonset-jbw9z, profiles-deployment-78d4549cbc-n45cx, pytorch-operator-b79799447-2k6t2, seldon-controller-manager-5fc5dfc86c-1ht2f, spark-operatorsparkoperator-67c6bc65fb-7rn7f, tf-job-operator-5c97f4bf7-8zr5z, workflow-controller-5c7cc7976d-5t8m2.

Простий цикл розробки, де всі дії виконуються вручну, можна розділити на 4 етапи:

1) Обробка даних.

2) Тренування моделі.

3) Оцінка та валідація моделі.

4) Запуск моделі.

Розглянемо процес автоматизації цього конвеєра з допомогою платформи Kubeflow покомпонентно, оскільки якісний конвеєр передбачає наявність певного рівня ізоляції між окремими етапами і чітке розуміння, що приходить на вхід та буде на виході.

Jupyter Notebook – це open-source веб-додаток, який дозволяє створювати та обмінюватися документами, що містять робочий код, рівняння, візуалізації та текст. Notebook використовують для очищення та перетворення даних, чисельного моделювання, статистичного моделювання, візуалізації даних, ML та багато іншого. Спектр використання jupiter notebook є дуже широким та буде використовуватись на всіх етапів конвеєра.

Зберігання метаданих про результати навчання також є важливим етапом розробки ML систем, що дозволяє оцінити різні версії моделі, порівнювати їх між собою. Проєкт Metadata, що є аналогом MLFlow, створений для того, аби вирішити проблеми користувачів із розумінням їх процесів машинного навчання, відслідковуючи і опрацьовуючи метадані, які створені цим процесом. В цьому контексті метадані означають інформацію про моделі, датасет (набір даних) та інші артефакти.

Тренування моделі це процес оптимізації алгоритму залежно від вхідних даних. Kubeflow має в собі оператори, які автоматизують навчання для таких фреймворків машинного навчання, як: MXNet; PyTorch; TensorFlow.

Концепція Kubernetes операторів дозволяє спростити процес навчання та представити процес як окремий kubernetes об’єкт. Складність процесу схована за додатковою абстракцією і, як результат, відносно складна задача розподіленого ML може бути представлена одним kubernetes об’єктом та описана YAML файлом маніфестом.

Налаштування гіперпараметрів (AutoML) – задача машинного навчання з вибору множини оптимальних гіперпараметрів для алгоритму машинного навчання. Katib – проєкт для розробки ML систем, що підтримує оптимізацію гіперпараметрів, пошук нейронної архітектури (NAS). Katib дозволяє задати швидкість навчання, кількість шарів у нейронній мережі та кількість вузлів у кожному шарі.

Kubeflow pipeline є ключовим елементом розробки ML системи об'єднуючи всі компоненти машинного навчання в повноцінний конвеєр (рис. 2). Перевагою Kubeflow pipeline у порівнянні із іншими CI системами є наявність набору програмних бібліотек (SDK) на мові програмування Python, яка покриває основні kubeflow абстракції. Описання конвеєра розробки тією ж мовою програмування, що і код моделі, позитивно впливає на процес, оскільки руйнується абстрактна стіна між розробкою та експлуатацією. Розробкою коду та автоматизацією процесів може займатись один інженер.

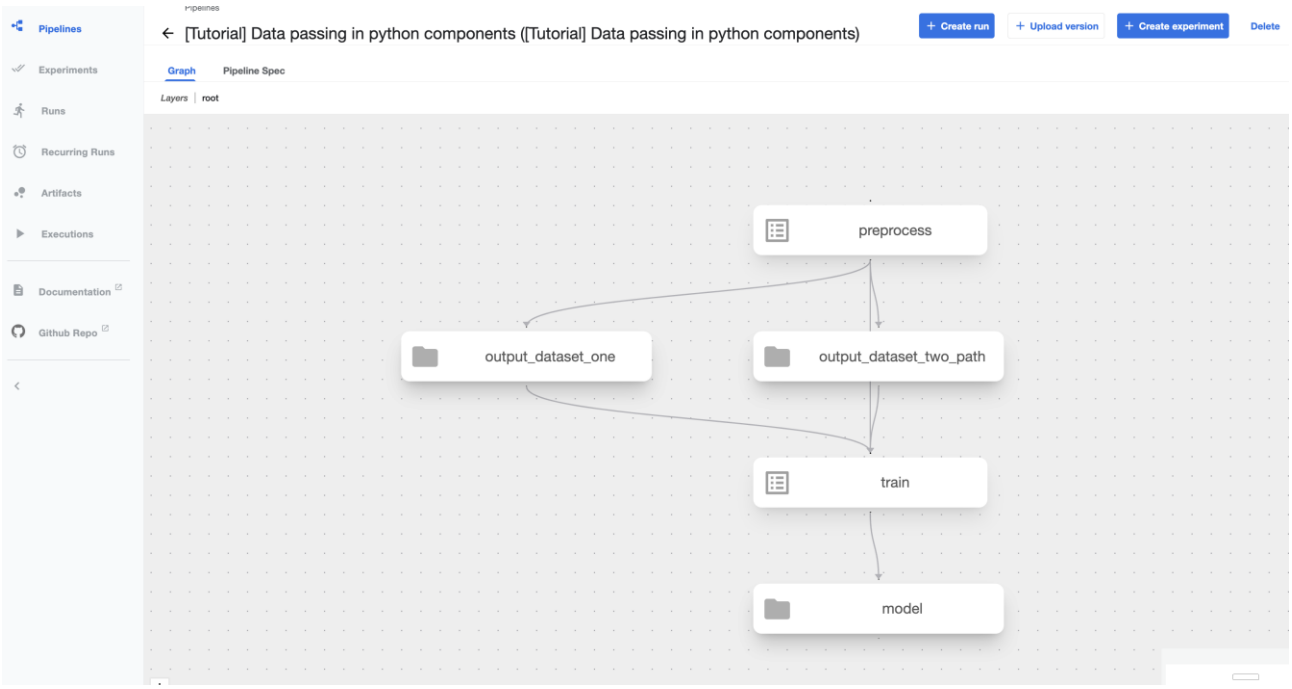

Рис. 2. Приклад візуалізації Kubeflow pipeline

Kubeflow Pipelines SDK включає наступні пакети, основні з них:

– kfp.compiler, включає класи та методи для компіляції конвеєра написаному на Python DSL в маніфести YAML, що застосовуються в Kubernetes;

– kfp.components, включає класи та методи для взаємодії із різними компонентами;

– kfp.dsl, включає предметно-орієнтовану мову програмування (DSL), що використовується для взаємодії із конвеєром та його компонентами;

– kfp.Client, включає Python клієнтські бібліотеки для взаємодії із Kubeflow Pipelines API.

4.2. Результати експлуатації на прикладі scikitlearn IRIS моделі в арсеналі Kubeflow за допомогою застосунку Seldon Core Serving

Сучасна методологія розробки програмних продуктів базується на принципах неперервного впровадження та не зупиняється коли модель потрапляє в продуктове середовище. Експлуатація моделі в класичному вигляді полягає в написанні коду, який би опрацьовував запити через grpc/http та повертав результат по відкритому порту. Також, при врахуванні технологічних рекомендацій, робота повинна включати створення контейнера, що в більшості дорівнює написанню Dockerfile, маніфести для розгортання в Kubernetes (deployment, service, ingress), розширення функціонала із відкриттям інтерфейсу по якому сторонні сервіси могли б отримувати метрики з роботи сервісу.

Для вирішення задачі експлуатації моделі в арсеналі kubeflow є набір застосунків, таких як Seldon Core Serving, NVIDIA Triton Inference Server, BentoML, TensorFlow Serving.

Seldon Core це – open source платформа для експлуатації моделей машинного навчання в Kubernetes. Seldon Core конвертує модель машинного навчання в REST/GRPC мікросервіс. Seldon підтримує розширення до тисячі екземплярів моделей і надає розширену підтримку із коробки такого функціонала, як метрики для моніторингу, логування запитів, пояснення моделі (explainers), виявлення крайніх випадків (outlier detectors), А/Б тестування, канареєчне розгортання. Зроблено практичний огляд Seldon Core Serving, хоча слід зазначити, що наведені застосунки мають схожий функціонал.

Для практичного аналізу було розгорнуто модель, яка перебуває у відкритому доступі, в Kubeflow кластер за допомогою маніфеста (рис. 3).

```yaml
apiVersion: machinelearning.seldon.io/v1
kind: SeldonDeployment
metadata:
  name: iris-model
  namespace: seldon
spec:
  name: iris
  predictors:
  - graph:
      implementation: SKLEARN_SERVER
      modelUri: gs://seldon-models/v1.14.0-dev/sklearn/iris
      name: classifier
    name: default
    replicas: 1
```

Рис. 3. Маніфест розгортання SeldonDeployment

Під парасолькою SeldenDeployment знаходиться 7 окремих об’єктів Kubernetes. Такий рівень спрощення системи для кінцевого користувача здійснено з використання патерну операторів в Kubernetes. Оператор – це програмний застосунок, що використовує об'єкт custom resources для обслуговування інших застосунків та їх компонентів. В даному випадку в ролі оператора виступає pod-a seldon-controller-manager-5fc5dfc86c-lht2f, що є частиною Kubeflow кластеру. Саме цей компонент разом із базовими Kuberentes контролерами відповідає за те, щоб при створенні custom resources seldondeployments.machinelearning.seldon.io, були створені об’єкти такі як pod, service, deployment, replicaset та istio virtual service. Також, при видалені seldondeployments, всі підпорядковані компоненти будуть автоматично очищені, і кінцевому користувачу не обов’язково пам’ятати про всі вище перелічені абстракції (service, deployment, pod і т. д.).

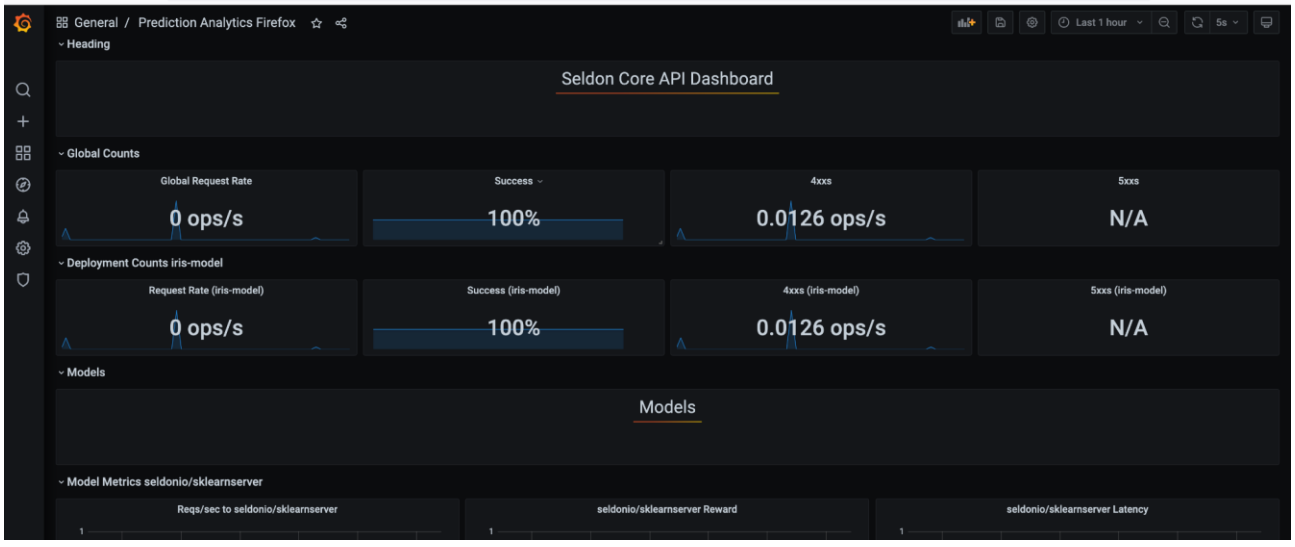

Рис. 4. Моніторингова панель Grafana на основі доступних метрик в SeldonCore

В SeldonCore наявний розширений моніторинг з інтеграціями із Prometheus та Grafana, що можуть бути окремо встановленні із використанням пакетного менеджера helm. Метрики автоматично доступні на застосунки по шляху 8080/metrics і prometheus автоматично їх зчитує за допомогою механізму із назвою service autodiscovery (рис. 4).

4.3. Використання Kubeflow для інформаційної системи виявлення цілей за допомогою безпілотних літальних апаратів

Проведені дослідження показали, що Kubeflow складається із набору різноманітних open source компонентів, що мають високий рівень інтеграції між собою через платформу Kubernetes. При цьому Kubeflow надзвичайно ефективно використовує патерн Kubernetes операторів для об'єктів ML. Продемонстровано, що написання коду моделі це тільки невелика частина серед задач ML, що впливає на необхідність автоматизації – наявність повноцінного конвеєра неперервної інтеграції та доставки додатка до кінцевого користувача.

Практичне застосування цього інструменту можна знайти в багатьох галузях через його гнучкість та універсальність. На основі проведених досліджень було сформовано концепцію інформаційного рішення на основі AI для виявлення ворожих цілей із БпЛА. Схематичне зображення інформаційної системи із можливістю постійного навчання для БпЛА зображено на рис. 5.

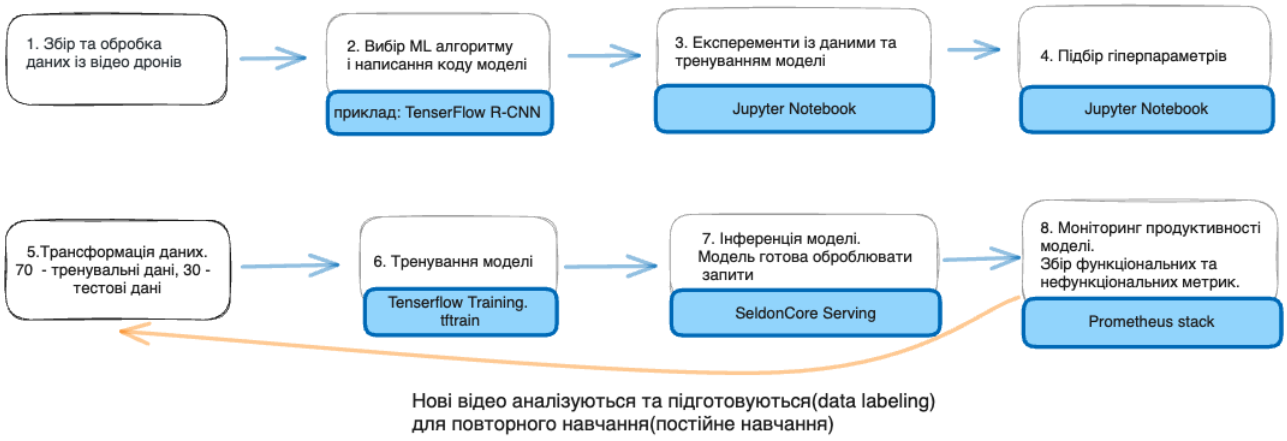

Рис. 5. Схематичне зображення інформаційної системи із можливістю постійного навчання для безпілотних літальних апаратів

Основні етапи ML конвеєру в розрізі нашої моделі.

1. Збір та обробка даних із відео БпЛА – це фото- і відеоматеріали різної якості та з різних джерел. Пріоритетні цілі покадрово визначаються оператором, на основні чого буде навчатись модель.

2. Вибір ML алгоритму і написання коду моделі – для нашого випадку підходить модель на основі R-CNN (Region-Based Convolutional Neural Network), що вирішує задачу класифікації об'єктів на фото чи відео.

3. Експерименти із даними та тренування моделі – інженери використовують різноманітні алгоритми та оброблені дані для тренування моделі.

4. Підбір гіперпараметрів – задача ML з вибору множини оптимальних гіперпараметрів для алгоритму ML. Гіперпараметр є параметром, значення якого використовується для керування процесом навчання. На відміну від значень інших параметрів (наприклад, вагових коефіцієнтів), які потрібно вивчити.

5. Трансформація даних – перетворення даних у формат, який потрібен системі навчання. Щоб переконатися, що модель поводиться узгоджено під час навчання та

прогнозування, процес трансформації має бути однаковим на експериментальній і продуктовій фазах. Для цього процесу доцільно використовувати такі компоненти як kubeflow pipelies та Jupiter Notebook.

6. Тренування моделі – це процес, у якому алгоритм ML подається навчальними даними, з яких він може навчатися. В цьому випадку для тренування доцільно використати компонент Kubeflow tftrain (Tenserflow training). Однією із переваг tfrain є наявність розподіленого навчання, що дозволяє розподілити навантаження навчання на декілька серверів кластера. За допомогою цього можна значно скоротити час навчання. Потенційне перенавчання моделі можна виконувати на регулярній основі щоб адаптуватись до змін та постійно покращувати ефективність.

7. Інференція моделі – це процес запуску моделі в продуктове середовище з використанням SeldonCore serving (рис. 6). Переваги використання цього інструменту описані вище. Також було досліджено, що платформа Kubernetes, на базі якої працює SeldonCore serving, дозволяє гнучко змінювати кількість робочих реплікацій для обробки високого навантаження.

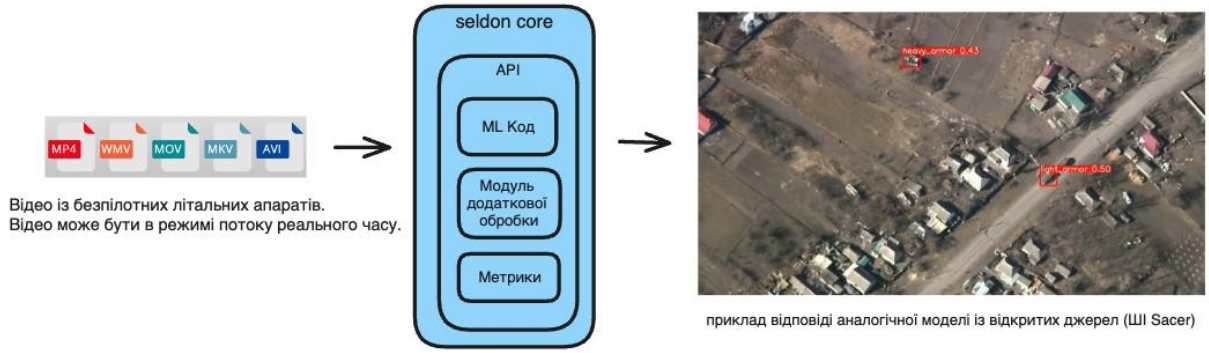

Рис. 6. Схематичне зображення інференції моделі із SeldonCore

8. Моніторинг описується як процес, що включає збір, аналіз та інтерпретацію показників (метрик) об’єкта спостереження. На основі поточних метрик, які зберігаються та обробляються в Prometheus стеку, можна приймати рішення про продуктивність моделі або збільшувати або зменшувати кількість реплік залежно від навантаження. Оброблені відео, на основі яких модель показала низьку ефективність, тобто пріоритетні цілі не були виявлені або були неправильно класифіковані, потрібно проаналізувати та обробити оператором для перенавчання моделі (етап 5). За рахунок цього модель буде постійно покращувати свою ефективність та швидко адаптуватись під зміни навколишнього середовища.

5. Обговорення результатів побудови конвеєру Kubeflow та розгортання моделі із використанням SeldonCore Serving

Результатом дослідження є формування та обґрунтування конвеєра із використанням компонентів платформи Kubeflow, що схематично було показано на рис. 1. На основі чого можна зробити висновок, що Kubeflow являє собою довершену платформу для розробки та впровадження програмних продуктів, що базуються на ML.

Представлення абстракцій у вигляді окремих ресурсів платформи дозволяє зменшити поріг входу для кінцевого користувача. Це дозволяє включити деякі компоненти системи в курс навчальної дисципліни “Системи підтримки прийняття рішень”, у т. ч. для спеціальностей, не пов’язаних з комп’ютерними науками.

Наявність обширного функціонала доступного із коробки робить Kubeflow ефективним інструментом розробки та впровадження ML моделей на підприємствах, в тому числі із

невеликим штатом інженерів. З іншого боку велика кількість абстракцій в системі вимагає від кінцевого користувача глибоких знань в предметній області, коли необхідно поглибитись на один або декілька рівнів нижче для кастоматизації.

Наведена концепція використання MLOps конвеєру для розв'язання прикладної задачі класифікації об’єктів із відео розвідувальних БпЛА є першою спробою представити комплексну систему машинного навчання як цілісний об’єкт, що може формувати додаткову цінність для продукту.

6. Висновки

За результатами дослідження визначено можливість використання сучасних MLOps рішень для покращення процесів розробки інформаційних систем машинного навчання. Доведено доцільність використання Kubeflow для створення MLOps конвеєру. Сформовано схематичний цілісний конвеєр розробки та експлуатації системи штучного інтелекту. Важливо звернути увагу на питання цілісності системи, оскільки коли один із етапів недостатньо розглядається або не розглядається взагалі, вся система деградує. Наведена концепція використання MLOps конвеєру для розв’язання прикладної задачі класифікації об’єктів із відео розвідувальних БпЛА є першою спробою представити комплексну систему машинного навчання як цілісний об’єкт. В нашому баченні наявність конвеєру формує додаткову цінність для продукту та є незамінним для моделей, які працюють в умовах непостійного середовища.

У результаті дослідження проведено практичне використання компонента Kubeflow Seldon Core Serving для розгортання тестової моделі. Це дозволило досягнути кращих практик індустрії експлуатації моделі з використанням спрощених абстракцій. До виявлених функціональних переваг віднесені A/B тестування, глибинний моніторинг внутрішніх метрик, сумісних із системою Prometheus, кращий рівень захисту від вразливостей, що доступні із коробки, тобто ті, що не потребують написання додаткового коду.

7. Список використаної літератури

1. Muratahan Aykol, Patrick Herring, and Abraham Anapolsky. Machine learning for continuous innovation in battery technologies. *Nat. Rev. Mater.* 2020, 5, 10, P. 725–727.

2. Mahendra Kumar Gourisaria, Rakshit Agrawal, G M Harshvardhan, Manjusha Pandey, and Siddharth Swarup Rautaray. Application of Machine Learning in Industry 4.0. *Machine Learning: Theoretical Foundations and Practical Applications*. Springer. 2021, P. 57–87.

3. Ana De Las Heras, Amalia Luque-Sendra, and Francisco Zamora-Polo. Machine learning technologies for sustainability in smart cities in the post-covid era. *Sustainability.* 2020, 12, 22, P. 9320.

4. Google Cloud. MLOps: Continuous Delivery and Automation Pipelines in Machine Learning. URL: https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning.

5. Dominik Kreuzberger, Niklas Kühl, Sebastian Hirschl. Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *Computer Science: Machine Learning*. 2023, Vol. 11, P. 31866.

6. Hidden Technical Debt in Machine Learning Systems. Part of Advances in Neural Information Processing Systems 28. Authors: D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, Dan Dennison. NIPS. 2015, 1486, 9 p.

7. Justin J. Boutilier, Timothy C. Y. Chan. Introducing and Integrating Machine Learning in an Operations Research Curriculum: An Application-Driven Course. *INFORMS Transactions on Education*. 2023, Vol. 23, 2, P. 64-83.

8. Fiebrink R. Machine Learning Education for Artists, Musicians, and Other Creative Practitioners. *ACM Transactions on Computing Education*. 2019, Vol. 19, 4, Art. No. 31, P. 1–32.

9. Sulmont E., Patitsas E., Cooperstock J. R. What Is Hard about Teaching Machine Learning to Non-Majors? Insights from Classifying Instructors' Learning Goals. *ACM Transactions on Computing Education*, 2019, Vol. 19, 4, Art. No. 33, P. 1–16.

10. Sánchez- Peña M, Vieira C and Magana A. Data science knowledge integration: Affordances of a computational cognitive apprenticeship on student conceptual understanding. *Computer Applications in Engineering Education*. 2022, 31, P. 239-259.

11. Elmachtoub A. N., Grigas P. Smart "Predict, then Optimize." *Management Science*. 2022, Vol. 68, 1, P. 9–26.

12. Mišić V. V., Perakis G. Data analytics in operations management: A review. *Manufacturing Service Operations Management*. INFORMS. 2020, 22 (1), P. 158–169.

13. Hazzan O., Mike K. Core Concepts of Machine Learning. Guide to Teaching Data Science. 2023, P. 209-22.

14. Ruibo Chen, Yanjun Pu, Bowen Shi, Wenjun Wu. An automatic model management system and its implementation for AIOps on microservice platforms. *The Journal of Supercomputing*. 2023, 79, P. 11410–11426.

15. Haoyu Cai, Chao Wang, Xuehai Zhou. Deployment and verification of machine learning tool-chain based on kubernetes distributed clusters. *CCF Transactions on High Performance Computing*. 2021, 3, P. 157–170.

16. Cong Yang, Wenfeng Wang, Yunhui Zhang, Zhikai Zhang, Lina Shen… MLife: a lite framework for machine learning lifecycle initialization. *Machine Learning*. 2021, 110, P. 2993–3013.

17. Asharul Islam Khan, Yaseen Al-Mulla. Unmanned Aerial Vehicle in the Machine Learning Environment. *Procedia Computer Science*. 2019, Vol. 160, P. 46-53.

18. Monfort Samuel S., Ciara M. Sibley, Joseph T. Coyne. Using machine learning and real-time workload assessment in a high-fidelity UAV simulation environment. Proceedings SPIE: Next-Generation Analyst IV. 2016, Vol. 9851, 98510B.

19. Kubeflow. Kubeflow Architecture. URL: https://www.kubeflow.org/docs/started/architecture/.

20. Bondarchuk A., Dibrivniy O., Grebenyk V., Onyshchenko V. Motion Vector Search Algorithm for Motion Compensation in Video Encoding / 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology, PIC S and T 2021 - Proceedings, 2021, p. 345–348