

УДК 004.85+004.896]:629.735

DOI: 10.31673/2412-9070.2023.062836

М. Ю. КУЗЬМІЧ, здобувач;

Т. Б. ГОРДІЄНКО, доктор техн. наук, професор,

Державний університет інформаційно-комунікаційних технологій, Київ

УПРОВАДЖЕННЯ KUBEFLOW MLOps У РОЗПОДІЛЕНІ ІНФОРМАЦІЙНІ СИСТЕМИ МОБІЛЬНИХ АГЕНТІВ ІЗ ПІДВИЩЕНОЮ ФУНКЦІОНАЛЬНОЮ СТІЙКІСТЮ

Наразі машинне навчання є одним з основних інструментів для розв'язання складних завдань у різних сферах діяльності. Популярність машинного навчання спричинена такими факторами, як доступ до великих наборів даних, невисока ціна обчислювальних ресурсів, наявні сервіси на основі хмарних технологій, швидкий прогрес у сферах штучного інтелекту.

Для розроблення, тестування та підтримання інфраструктури системи з даними застосовуються, зокрема, концепції машинного навчання та процесів (Machine Learning and Operation, MLOps) із набором технік для імплементації та автоматичної неперервної інтеграції. Концепцію MLOps розглянуто як інструменти Kubeflow-cloud native — системи з відкритим кодом, що працює на платформі Kubernetes. Досліджено можливості використання підходу MLOps для поліпшення процесів розроблення інформаційних систем машинного навчання. Спроектовано розподілену інформаційну систему на основі мобільних агентів, якими виступають безпілотні літальні апарати (БПЛА), та коротко описано їх потенційну практичну реалізацію за умов реального часу.

Продемонстровано, що написання коду моделі — це не тільки невелика частина серед задач машинного навчання, що впливає на потребу в автоматизації, це наявність повноцінного конвеєра неперервної інтеграції та доставляння застосунку до кінцевого користувача. Проведені дослідження показали, що Kubeflow становить набір різноманітних компонентів із відкритих джерел, що мають високий рівень інтеграції між собою через платформу Kubernetes. Це дає їм змогу бути запущеними на різноманітних пристроях, зокрема на таких мобільних агентах, як БПЛА.

Запропоновано та обґрунтовано використання Mesh-мереж для підвищення функціональної стійкості розподіленої інформаційної системи мобільних агентів. Було спроектовано архітектуру концепції системи на базі дистрибутива Kubernetes k3s, що дає можливість працювати в парадигмі крайових (Edge) обчислень.

Ключові слова: Kubeflow; Kubernetes; MLOps; мобільні агенти; постійне навчання; машинне навчання; Mesh-мережа; системи керування; ефективність керування; розподілена інформаційна система.

ВСТУП

Машинне навчання (Machine Learning, ML) стає критично важливим інструментом для максимізації цінності даних і дає змогу компаніям бути більш інноваційними, ефективними та стійкими. Наразі ML є одним із основних інструментів для розв'язання складних прикладних завдань у різних сферах діяльності. ML є одним із методів функціонування штучного інтелекту, а саме: практичної реалізації його можливостей завдяки створенню алгоритмів для виявлення закономірностей під час аналізу великих даних, та їх подальше використання для самонавчання [1–3]. Основними причинами популярності ML стали такі чинники, як доступ до великих наборів даних, невисока ціна обчислювальних ресурсів, готівні сервіси на основі хмарних технологій, швидкий прогрес у сферах штучного інтелекту.

Сьогодні компанії мають доступ до величезних обсягів даних, які можуть надати цінні відомості. Однак значні обсяги даних не завжди дають змогу це зробити, оскільки компаніям складно здобути цінну інформацію вручну. Це один із прикладів, де ML стає в пригоді. Компанії можуть швидко і точно аналізувати великі обсяги даних, виявляти закономірності й ухвалювати рішення на основі даних, використовуючи алгоритми та методи ML. Алгоритми ML, наприклад, можна застосовувати для виявлення шахрайських фінансових операцій [4], прогнозування відтоку клієнтів, оптимізації керування ланцюгом постачань [5] та автоматизації виробничих процесів [6].

Водночас у розробленні ML існує велика кількість додаткових складнощів, через які безліч ML-систем зазнає невдачі на ранніх стадіях розроблення і не досягає продуктового середовища.

Для розв'язання цих проблем широко використовується концепція спільного підходу до процесів ML та експлуатації, відома як MLOps. Це комплекс технік, спрямованих на впровадження та автоматизацію неперервного інтегрування, доставляння моделей до продуктового середовища та їх навчання. MLOps — це інженерна культура та практика в галузі машинного навчання, спрямована на злагодження розроблення систем ML (Dev) та їх експлуатації (Ops). Суть MLOps полягає в автоматизації та моніторингу на всіх етапах створення системи ML, включно з інтеграцією, тестуванням, випуском, розгортанням та керуванням інфраструктурою [7].

Фахівці з оброблення даних можуть успішно створювати та навчати модель машинного навчання, яка виявляє ефективність на офлайн-затриманому наборі даних з огляду на наявні навчальні дані для конкретного сценарію використання. Проте головний виклик полягає не тільки в розробленні моделі

© М. Ю. Кузьміч, Т. Б. Гордієнко, 2023

ML, а й у побудові інтегрованої системи ML та забезпеченні її неперервної експлуатації у виробництві. За довгий період роботи у сфері виробничих служб ML компанія Google виявила, що експлуатація систем на основі машинного навчання у виробництві може призводити до низки складнощів. Деякі з цих проблем було узагальнено в документі [8].

Останнім часом зростає зацікавленість до концепції контейнеризації, особливо в контексті керування великим обсягом компонентів сервісу на значних кількостях серверів. Компанія Google, яка операційно керує розгортанням та розвитком широкого спектра сервісів на сотнях тисяч серверів, визначається серед тих, хто вдається до цієї концепції. Основою для цього стала розподілена система керування контейнерами, відома як Kubernetes, яка використовується як засіб контейнеризації та мікросервісної архітектури [7].

У контексті концепції MLOps розумно розглядати інструменти, такі як Kubeflow, котрий є системою з відкритим кодом, спеціально розробленим для хмарного ML. Kubeflow оптимізує процес розгортання робочих циклів ML у середовищі Kubernetes. Його завдання не лише уникати дублювання функціонала інших служб, а й надавати зручний метод для розгортання передових систем із відкритим кодом для ML на різних інфраструктурах, чи то внутрішні сервери виробництва, чи то рішення в публічному хмарному просторі. Тобто, Kubeflow може ефективно працювати там, де працює Kubernetes, надаючи гнучкість та переносність рішень.

З огляду на високу портативність застосунків, розроблених у Kubernetes та Kubeflow, проаналізовано можливість розгортання моделей ML на мобільних агентах, зважаючи на те, що мобільними агентами в запропонованій концепції є безпілотні літальні апарати (БпЛА).

Аналіз літературних джерел і постановка проблеми. Машинне навчання вже доволі ефективно застосовують у різних сферах діяльності, постійно роблячи інвестиції в розроблення якісних та передбачуваних моделей [1–3]. Концепція MLOps — це набір технік для імплементації та автоматичної неперервної інтеграції, доставляння та навчання моделей, яка охоплює основні патерни та антипатерни, на основі яких будують успішний конвеєр автоматизації розроблення ML-систем. Ефективність використання підходу MLOps було розглянуто і доведено в багатьох працях, де сформалізовано основні вимоги до ML-систем та виявлено проблеми [8–12].

Застосування аналітики даних для керування операціями з використанням ML та методології оптимізації, зокрема за алгоритмом «передбачити, потім оптимізувати», розглянуто в [13; 14]. У статтях наведено застосування аналітики даних для керування операціями в трьох основних сферах — керування ланцюгом постачань, керування доходами та операції з охорони здоров'я. Однак у цих працях не розглядаються будь-які конкретні рекомендації щодо покращення або зміни моделі, наприклад, реакцію на зміну ланцюгів постачань.

У [15] досліджено основні концепції, спільні для багатьох алгоритмів ML, зокрема налаштування гіперпараметрів як важливих цілей навчання, незалежно від алгоритмів ML. Обговорено типи ML, їхні параметри та гіперпараметри, навчання моделі, валідацію та тестування. Водночас автори наголошують, що незважаючи на те, що постійно докладаються зусилля, аби вдосконалити ML-алгоритми та зробити їх більш зрозумілими та інтерпретованими, роль викладачів полягає в поліпшенні розуміння людиною ML-алгоритмів.

У праці [16] запропоновано структуру конвеєра оновлення моделі та керування для моделей AI Ops у системах мікросервісів. Це зроблено з метою навчання, інкапсулювання та розгортання моделі, а також для спрощення процесу. Для забезпечення попередньої реалізації фреймворку та перевірки розроблено прототип системи на основі Kubernetes і Gitlab. Процес пакування та розгортання автоматизовано за допомогою технології неперервної інтеграції. Дослідження в цій статті є корисним ресурсом для побудови інтегрованої та оптимізованої системи керування моделлю AI Ops. Однак у запропонованій технології наявний 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 б автоматично адаптувалась під конкретне середовище мікросервісів і не потребувала ручного донавчання.

У статті [17] досліджено проблеми конвеєра Kubeflow, перевірено здійсненність рішення незалежності Kubernetes від Google Cloud Service на прикладі підтримання програми ML на основі Pytorch. Це рішення дає можливість науковцям з оброблення даних створювати програми глибокого навчання на основі Pytorch у розробленні розподілених систем на основі Kubernetes. Запропонований підхід забезпечує стабільну та неперервну роботу програми та розумне планування на кластері, що містить різні обчислювальні ресурси.

У [18] описано можливість використання Mesh-мереж для платформи Kubernetes та потенційні переваги використання оновленого компонента планувальника навантаження з можливістю оцінювати затримки між вузлами. Запропоноване рішення може поліпшити загальну продуктивність розподіленої системи, коли виникатиме потреба у збільшенні кількості вузлів. Однак у статті недостатньо приділено уваги практичним перевагам застосування Mesh-мереж у Kubernetes та дистрибутивам, розробленим для систем у парадигмі крайових обчислень (Edge computing).

У праці [19] описано структурну схему системи керування мобільними агентами. Компоненти системи взаємодіють між собою, утворюючи цілісний механізм, який може бути використаний для оптимізації різноманітних завдань та досягнення високого рівня ефективності системи керування мобільними агентами. Проте поточне дослідження не описує рішення з відкритих джерел на базі Kubernetes, що можуть бути використані як основа для впровадження системи, до яких можна віднести легковаговий дистрибутив Kubernetes k3s або систему з відкритим вихідним кодом KubeEdge.

Проведений аналіз показав потребу в продовженні дослідження можливостей сучасних MLOps-рішень для сприяння вдосконалення процесів розроблення інформаційних систем ML [7]. Висока портативність Kubernetes та Kubeflow дає змогу розгортати застосунки на різних системах, тому доцільно розглянути концепції системи на базі дистрибутивів Kubernetes та можливість встановлення на мобільних агентах. Також для підвищення функціональної стійкості розподіленої інформаційної системи мобільних агентів слід звернути увагу на використання безпроводових мереж, на кшталт Mesh-мереж.

Мета і задачі дослідження. Метою дослідження є вдосконалення процесів розроблення інформаційних систем машинного навчання з використанням Kubeflow та можливістю встановлення на мобільних агентах.

Для досягнення мети було поставлено такі завдання:

- дослідити можливість використання Kubeflow для покращення процесів розроблення інформаційних систем машинного навчання;
- сформулювати концепт інтеграції Kubeflow у розподілені інформаційні системи на базі мобільних агентів, беручи до уваги її цілісність та автономність;
- дослідити використання Mesh-мереж для підвищення функціональної стійкості розподілених ML-систем на прикладі систем БПЛА.

ОСНОВНА ЧАСТИНА

Матеріали та методи дослідження щодо підвищення функціональної стійкості MLOps-системи безпілотних літальних апаратів

Під час роботи з ML визначають етапи, які можуть бути виконані як вручну, так і автоматично [7]. Kubeflow, як відкрите джерело для ML, має високий рівень інтеграції із системою оркестрування Kubernetes та значний набір інструментарію для ML-задач [18; 20].

NAME	READY	STATUS	RESTARTS	AGE
admission-webhook-bootstrap-stateful-set-0	1/1	Running	0	8m24s
admission-webhook-deployment-5d9ccb5696-b4k2n	1/1	Running	0	8m21s
alb-ingress-controller-7dcf69ff67-mwx1	1/1	Running	0	9m4s
application-controller-stateful-set-0	1/1	Running	0	8m53s
argo-ui-684bcb587f-rq4j9	1/1	Running	0	9m4s
cache-deployer-deployment-6667847478-p516t	2/2	Running	1	9m4s
cache-server-bd9c859db-svhq8	2/2	Running	0	9m3s
centraldashboard-895c4c768-r4ssh	1/1	Running	0	9m3s
jupyter-web-app-deployment-54758f7bc4-g2svl	1/1	Running	0	9m3s
katib-controller-75c8d47f8c-9n99c	1/1	Running	0	9m3s
katib-db-manager-6c88c68d79-fbnzt	1/1	Running	0	9m3s
katib-mysql-858f68f588-zz7dn	1/1	Running	0	9m3s
katib-ui-68f59498d4-pr15s	1/1	Running	0	9m3s
kfserving-controller-manager-0	2/2	Running	0	8m44s
kubeflow-pipelines-profile-controller-69c94df75b-zfwff	1/1	Running	0	8m2s
metacontroller-0	1/1	Running	0	8m34s
metadata-db-757dc9c7b5-6xvjw	1/1	Running	0	9m2s
metadata-envoy-deployment-6ff58757f6-dkj64	1/1	Running	0	9m2s
metadata-grpc-deployment-76d69f69c8-bwp8b	1/1	Running	2	9m2s
metadata-writer-6d94ffb7df-7d4lg	2/2	Running	0	9m2s
minio-66c9cd74c9-n6659	1/1	Running	0	9m2s
ml-pipeline-54989c9946-m6ga8	2/2	Running	0	9m1s
ml-pipeline-persistenceagent-7f6bf7646-x29rq	2/2	Running	0	9m1s
ml-pipeline-scheduledworkflow-66db7bcf5d-5sb6f	2/2	Running	0	9m1s
ml-pipeline-ui-756b58fb-tcvtr	2/2	Running	0	9m1s
ml-pipeline-viewer-crd-53f59f87db-g29f9	2/2	Running	1	9m
ml-pipeline-visualizationserver-6f9ff4974-jcwc	2/2	Running	0	9m
mpi-operator-77bb5d8f4b-68z2f	1/1	Running	0	9m
mxnet-operator-68b688bb69-xrr24	1/1	Running	0	9m
mysql-7694c6bb87-ghbhv	2/2	Running	0	9m
notebook-controller-deployment-58447d4b4c-ndqkt	1/1	Running	0	9m
nvidia-device-plugin-daemonset-dt792	2/2	Running	0	8m29s
nvidia-device-plugin-daemonset-jbw9z	2/2	Running	0	8m38s
profiles-deployment-78d4549c9c-n45cx	2/2	Running	0	8m59s
pytorch-operator-b79799447-2k6t2	1/1	Running	0	8m59s
seldon-controller-manager-5f5d4fc86c-lht2f	1/1	Running	0	8m59s
spark-operatorsparkoperator-67c6bc65fb-7rn7f	1/1	Running	0	8m59s
tf-job-operator-5c97f4b7-8zr5z	1/1	Running	0	8m59s
workflow-controller-5c7cc7976d-5t8m2	1/1	Running	0	8m59s

Рис. 1. Список інстальованих компонентів Kubeflow

У дослідженні [7] зроблено огляд основних компонентів Kubeflow версії 1.12, їх доцільність під час формування конвеєра та створено застосунок інтерфейс командного рядка (*Command line interface*, CLI) для цієї потреби. Використання готового CLI-застосунку значно спрощує процес інсталяції, оскільки дає змогу уникнути ручного редагування масивного конфігураційного файлу, зменшуючи поріг входу для користувачів. Список інстальованих компонентів Kubeflow зображено на рис. 1.

Концепт використання Kubeflow для інформаційної системи виявлення об'єктів для БПЛА описано в [7]. Результати наукових досліджень зазначають, що Kubeflow — це різноманітні компоненти з відкритим вихідним кодом, які високо інтегровані через платформу Kubernetes. Також наголошується на високій ефективності Kubeflow у використанні патернів операторів Kubernetes для оброблення об'єктів ML. Особливу увагу приділено тому, що розроблення коду моделі це лише невелика частина в широкому

спектрі завдань ML, що підкреслює актуальність автоматизації. Водночас доведено, що наявність повноцінного конвеєра для неперервної інтеграції та постачання застосунків є ключовою для успішної реалізації цього процесу від початкового розроблення до кінцевого користувача. На основі проведених досліджень було сформовано концепцію інформаційного рішення на основі штучного інтелекту для виявлення ворожих цілей із БпЛА.

Для підвищення функціональної стійкості розподіленої інформаційної системи БпЛА доцільно також розглянути використання безпроводової Mesh-мережі. Актуальність застосування Mesh-мереж визначається розвитком мікроелектроніки, появою безлічі різних пристроїв, здатних працювати автономно довгий час, мають особливість багаторазової зміни режиму (онлайн-находження в мережі і офлайн-виходу з мережі) і потребують обміну інформацією зі своїм оточенням, а можливо і з керівним або інформаційним центром, наприклад для підтримання концепції Інтернету речей (IoT).

Одна з переваг Mesh-мереж — незалежність та стійкість. Чим більша кількість абонентів, тим щільніша та надійніша мережа. Отже, можна завжди розгорнути якісний зв'язок у місцях, де відсутня мережна інфраструктура. Це може стати дуже корисним у регіонах підвищеного ризику, у місцях дикої природи та віддалених населених пунктах, де абсолютно кожен абонентський пристрій (наприклад, смартфон місцевого жителя або станція, встановлена на транспортному засобі дільничного) може брати участь у процесі передавання інформації до адресата.

За задумом, у повноцінних Mesh-мережах не можна перехопити трафік і заборонити поширення інформації. Це, зі свого боку, може суперечити державним законам конкретної країни або регіону. Водночас державні структури та військові відомства через ці причини зацікавлені в освоєнні й організації таких мереж. Топологія мережі адаптивна, спрямована на використання будь-яких доступних каналів для забезпечення повної маршрутизації між усіма учасниками мережі, що робить її надзвичайно стійкою до засобів радіоелектронної боротьби (РЕБ).

Провідними протоколами на сьогоднішній день є Cjdns та Yggdrasil. Іноді Wi-Fi Mesh-мережі називають Cjdns-мережами та Yggdrasil-мережами, оскільки ці протоколи є «двигуном маршрутизації».

Основна ідея полягає в можливості забезпечити функціонально стійку мережу передавання даних для поточної інформаційної системи на основі БпЛА, яка зможе протистояти сучасним РЕБ засобам та іншим завадам і працювати в автономному або напівавтономному вигляді.

До основних ознак функціональної стійкості належать такі:

- структура розподіленої інформаційної системи (РІС) є функціонально стійкою, якщо граф структури однокомпонентний і не має мостів і вузлів з'єднання. Обернене визначення говорить про функціональну нестійкість структури;
- структура РІС є функціонально нестійкою, якщо її граф багатоконпонентний і незв'язний.

Граф Mesh-мережі 5×5 стандарту Інституту інженерів електротехніки та електроніки IEEE 802.11s «Розширений набір послуг (ESS) Mesh Networking», який є частиною стандарту IEEE 802.11 з набору стандартів IEEE, що регулюють протоколи передавання безпроводових мереж [21], зображено на рис. 2.

Отже, за зовнішнім виглядом графа, а саме за кількістю компонентів, наявності мостів і вузлів з'єднання графа можна оцінювати функціональну стійкість структури, тобто закладені в ній властивості протидіяти відмовам і пошкодженням.

Огляд розгортання Kubernetes та Kubeflow у Mesh-мережі

Edge computing — це парадигма розподілених обчислень, яка переносить обчислення і носії даних ближче до джерела даних. У такий спосіб значно знижується затримка, оскільки дані обробляються на місці їх виникнення. Це особливо важливо для застосунків, які працюють у режимі реального часу, зокрема автономні автомобілі та БпЛА, Інтернет речей (IoT) тощо. Парадигма Edge computing також користується популярністю для Kubernetes, де існує тренд його розгортання повністю або частково на Edge пристроях.

Kubernetes початково не був розроблений для роботи із безпроводовими стандартами передавання даних, у тому числі Mesh-мережами та IoT-рішеннями, але сьогодні є дистрибутивні (рішення на основі Kubernetes із певними змінами) саме для тих задач, як KubeEdge, k3s, MicroK8s [22–24]. Структурна схема інформаційної системи буде використовувати k3s, який із трьох рішень користується найбільшою популярністю та виграє за низкою інших характеристик.

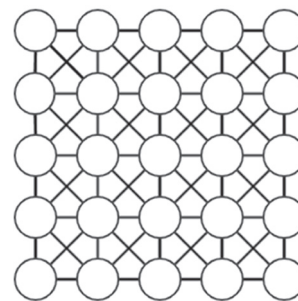


Рис. 2. Граф Mesh-мережі 5×5 стандарту IEEE 802.11s

KubeEdge — є системою з відкритим кодом, яка розширює оркестрування контейнерних програм і керування пристроями на крайові обчислення (Edge), вузли. Він створений на основі Kubernetes і забезпечує підтримання основної інфраструктури для мереж, розгортання програм і синхронізації метаданих між Cloud частиною (хмарні обчислення) та Edge. KubeEdge нативно підтримує спрощений мережний протокол, що працює на TCP/IP (MQTT), дає змогу розробникам створювати спеціальну логіку та забезпечувати зв'язок пристрою з обмеженими ресурсами на Edge. Архітектуру KubeEdge з розділенням системи на Cloud і Edge компоненти наведено на рис. 3 [22].

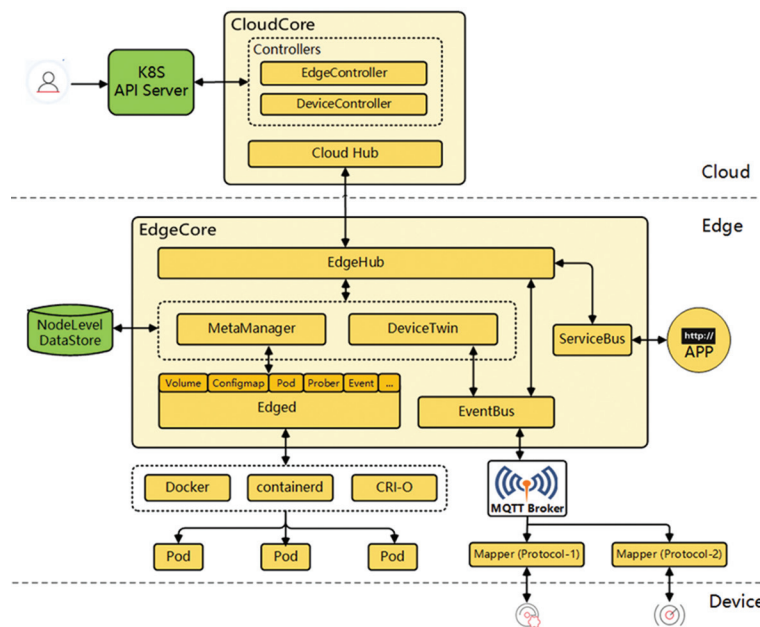


Рис. 3. Архітектура KubeEdge — розділення системи на Cloud і Edge компоненти

До переваг KubeEdge систем та Edge computing парадигми в цілому можна віднести таке:

- **Edge computing:** завдяки бізнес-логіці, що працює на Edge пристроях, набагато більші обсяги даних можна захищати та обробляти локально, де вони створюються. Це зменшує вимоги до пропускну здатності мережі між Edge і Cloud, що підвищує швидкість реагування, зменшує витрати та захищає конфіденційність даних;
- **спрощення процесу розроблення:** розробники можуть писати стандартні застосунки на основі HTTP або MQTT, контейнеризувати їх і запускати будь-де – або на Edge пристроях, або в Cloud середовище – залежно від того, що більше підходить;
- **підтримання Kubernetes:** за допомогою KubeEdge користувачі можуть оркеструвати застосунки, керувати пристроями та відстежувати стан застосунків і пристроїв на Edge вузлах так само, як традиційний кластер Kubernetes у Cloud середовищі;
- **підтримання різноманітності застосунків:** на Edge під час використання правильного інструментарію легко дістати та розгорнути наявні складні програми ML, розпізнавання зображень, оброблення подій та інші високорівневі програми.

До переваг саме KubeEdge щодо інших схожих інструментів можна віднести підтримання MQTT брокера з коробки для керування Edge вузлами. MQTT брокер використовує концепцію передавання інформації на основі черги завдань, перевагу якої описано у праці з дослідження системи керування мобільними агентами [19]. Черга завдань у системі керування мобільними агентами підтримує список завдань, які мають бути ними виконані. Цей список упорядковано за пріоритетом або за часом надходження завдання.

Також слід звернути увагу на компонент EdgeMesh [25], що є доповненням до KubeEdge, забезпечуючи просте виявлення сервісів і проксі-функції трафіку для застосунків, тим самим захищаючи складну мережну структуру в Edge сценаріях. Архітектуру високої доступності EdgeMesh загалом спрямовано на граничні сценарії, такі як розподілені динамічні ретрансляційні з'єднання та мережна автономія приватних локальних мереж. Він забезпечує ретрансляцію трафіку для вузлів, щоб гарантувати, що кластерні з'єднання будуть працювати в штатному режимі в граничних сценаріях, наприклад, коли один чи кілька вузлів вийшли з ладу.

Основна ідея високої доступності EdgeMesh полягає в тому, що коли вузол у кластері має можливість ретрансляції, агент EdgeMesh візьме на себе роль ретранслятора для забезпечення передавання трафі-

ку до інших вузлів. Коли кластер ініціалізується або новий вузол приєднується до кластера, система EdgeMesh виявляє та записує вузли в локальній базі на основі механізму mDNS (multicast DNS). Водночас механізм розподіленої хеш-таблиці (DHT) відповідатиме на запити підключення від інших вузлів за межами локальної мережі. Якщо два вузли локальної мережі хочуть бути з'єднані, ретрансляційний вузол може надати їм послуги ретрансляції трафіку та сприяння проникненню в інтрамережу.

K3s — є високодоступним сертифікованим дистрибутивом Kubernetes, розробленим для продуктивних навантажень у віддалених місцях без нагляду, з обмеженими ресурсами або для IoT-пристроїв. З огляду на результати досліджень у контексті Edge computing, де пристрої можуть мати обмежені обчислювальні ресурси, k3s є кращим вибором.

Порівнюючи платформи k3s та KubeEdge, було надано перевагу k3s, зважаючи на такі чинники:

- **спільнота:** k3s і KubeEdge є відкритими продуктами та їх розвиток залежить від активності спільноти, що лежить в основі розроблення цих інструментів. K3s початково розробляла компанія Rancher, а отже, він має велику англійську аудиторію, на відміну від KubeEdge і особливо компонента EdgeMesh, які розробляються китайськими компаніями. Тут засідання робочих груп здебільшого відбуваються китайською мовою, що ускладнює більш глибоке вивчення технології та створює певний мовний бар'єр;
- **автономність:** KubeEdge має чіткий розподіл на Cloud і Edge компоненти, що знижує його автономність та функціональну стійкість;
- **гнучкість:** k3s є більш гнучким у налаштуваннях.

Упровадження Kubeflow MLOps у розподілену інформаційну систему k3s із використанням БпЛА

Концепція використання Kubeflow для інформаційної системи виявлення об'єктів для БпЛА описано в [7]. Далі буде розглянуто більш глибокі дослідження стосовно виконання етапів 7 і 8 конвеєра на Edge пристроях (рис. 4).

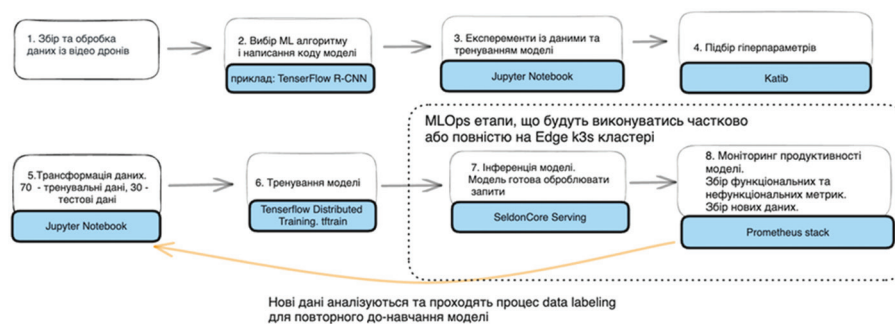


Рис. 4. Схематичне зображення інформаційної системи з можливістю постійного навчання для безпілотних літальних апаратів

Структурна схема інформаційної системи використовуватиме k3s, який із трьох рішень є найбільш доцільним для застосування в розробленій системі. Як і Kubernetes, k3s має схожу архітектуру розділення на службові ноди (Server) та ноди працівники (Agent). На Server ноди встановлено необхідні службові компоненти для роботи кластера та Agent-компонети, що дають їм змогу запускати клієнтські застосунки також. На Agent-ноди службові компоненти не встановлюються, тільки базові (рис. 5).

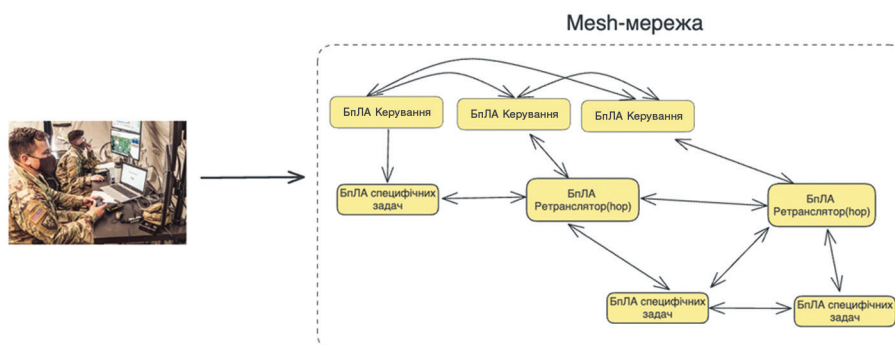


Рис. 5. Схематичне зображення розподіленої інформаційної Mesh-системи

Розглянемо запропоновану поточну систему на рис. 5 більш детально. БпЛА Керування (Server) — це БпЛА, що мають достатню кількість ресурсів для виконання службових задач розподіленої системи Kubernetes та Kubeflow компонентів для, наприклад, інференції моделі, як Seldon Core і/або запуску

інших ресурсоемних застосунків для контролю над усією системою. Відповідно вартість цих БпЛА буде вищою. Доцільно мати хоча б три БпЛА керування для досягнення високої доступності та можливості розподілення навантаження між вузлами.

БпЛА специфічних задач (Agent) та БпЛА ретранслятори (Agent) — це службові БпЛА, що мають слабші характеристики, ніж службові, та виконують вузькоспеціалізовані завдання, що будуть надіслані до них службовим вузлом.

Прикладом використання цієї концепції може бути таке: БпЛА специфічних задач виконує функції баражувального боеприпасу, де модель вибору та розпізнавання і доведення до цілі виконуються на БпЛА керування, а надійність та стійкість мережі забезпечується використанням БпЛА ретрансляторів. Тобто така система дає змогу знизити собівартість «розумного» баражувального боеприпасу, перемістивши роботу «штучного інтелекту» на інший вузол, що може бути використаний багаторазово.

Обговорення результатів побудови конвеєра Kubeflow та розгортання моделі на БпЛА з огляду на взаємодію між вузлами

Результатом дослідження є формування та обґрунтування конвеєра з використанням компонентів платформи Kubeflow, що схематично показано на рис. 4. На основі чого можна дійти висновку, що Kubeflow є довершеною платформою для розроблення та впровадження програмних продуктів, що базуються на ML.

Наявність великого функціонала, доступного з коробки, робить Kubeflow ефективним інструментом розроблення та впровадження ML-моделей на підприємствах, зокрема з невеликим штатом інженерів. Проте чимала кількість абстракцій у системі потребує від кінцевого користувача глибоких знань у предметній сфері, коли необхідно поглибитись на один або кілька рівнів нижче для кастомізації.

Використання Kubernetes як платформи для нашого конвеєра робить нашу систему портативною, що дає можливість переносити та використовувати різні рішення та моделі машинного навчання без прив'язки до конкретного середовища чи платформи. Описана модель буде ефективно працювати як у хмарному середовищі, так і в парадигмі Edge Computing.

Наведена концепція використання MLOps конвеєра придатна для розв'язання прикладної задачі класифікації об'єктів із відеорозвідувальних БпЛА на основі системи мобільного агента, що працюють як повноцінна розподілена система.

Застосування Mesh-мереж запропоновано для досягнення високої функціональної стійкості інформаційної системи БпЛА, що робить її використання можливим за умов ведення радіоелектронної боротьби.

ВИСНОВКИ

1. Досліджено можливості використання Kubeflow для покращення процесів розроблення інформаційних систем ML. Виведено на передній план той факт, що розроблення коду моделі лише невелика частина в широкому спектрі завдань ML, що підкреслює актуальність автоматизації. Водночас доведено, що наявність повноцінного конвеєра для неперервної інтеграції та постачання застосунків є ключовою для успішної реалізації цього процесу від початкового розроблення до кінцевого користувача.

2. Розроблено концепцію конвеєра ML для моделі виявлення об'єктів для БпЛА з можливістю постійного донавчання. Це сприяє швидкій адаптації інформаційної системи до змін у навколишньому середовищі та специфіки завдань.

3. На основі проведено аналізу основних дистрибутивів Kubernetes, що використовуються для Edge Computing, обґрунтовано переваги та доцільність використання k3s для поточної системи. Запропоновано використання Mesh-мереж для БпЛА, які є розподіленою інформаційною системою на базі дистрибутива Kubernetes k3s та Kubeflow, що працюють у парадигмі Edge Computing. Доведено, що використання Mesh-мереж дає змогу досягти високої функціональної стійкості інформаційної системи БпЛА за умов ведення радіоелектронної боротьби.

Список використаної літератури

1. Aykol M., Herring P., Anapolsky A. Machine learning for continuous innovation in battery technologies // *Nat. Rev. Mater.* 2020. 5, 10. P. 725–727.

2. *Application of Machine Learning in Industry 4.0 / Mahendra Kumar Gourisaria, Rakshit Agrawal, G. M. Harshvardhan [et al.] // Machine Learning: Theoretical Foundations and Practical Applications. Springer. 2021. P. 57–87.*

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

4. *Kshetri Nir. Blockchain's Roles in Meeting Key Supply Chain Management Objectives // International Journal of Information Management. 2018. 39. P. 80–89. URL: <https://doi.org/10.1016/j.ijinfomgt.2017.12.005>.*
5. *Li Y., Wang B. A study on customer churn of commercial banks based on learning from label proportions // In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018. IEEE: Piscataway Township, NJ, USA. P. 1241–1247.*
6. *Zhang B., Shi H., Wang H. Machine Learning and AI in Cancer Prognosis, Prediction, and Treatment Selection: A Critical Approach // Dove Medical Press. 2023. Vol.16. P. 1779–1791.*
7. *Кузьміч М. Ю., Гордієнко Т. Б. Застосування інструменту kubeflow для інтеграції машинного навчання і штучного інтелекту в безпілотних літальних апаратах // Телекомунікаційні та інформаційні технології. 2023. Вип. 3 (80). С. 65–78.*
8. *Machine Learning: The High-Interest Credit Card of Technical Debt / D. Sculley, G. Holt, D. Golovin [et al.] // SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop). 2014.*
9. *Kreuzberger D., Kühl N., Hirschl S. Machine Learning Operations (MLOps): Overview, Definition, and Architecture.*
10. *Hidden Technical Debt in Machine Learning Systems / D. Sculley, G. Holt, D. Golovin [et al.] // Part of Advances in Neural Information Processing Systems 28 (NIPS 2015).*
11. *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.*
12. *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.*
13. *Elmachtoub A. N., Grigas P. Smart. Predict, then Optimize // Management Science. 2022. Vol. 68, 1. P. 9–26.*
14. *Mišić V. V., Perakis G. Data analytics in operations management: A review // Manufacturing Service Operations Management. INFORMS. 2020. 22 (1). P. 158–169.*
15. *Hazzan O., Mike K. Core Concepts of Machine Learning. Guide to Teaching Data Science. 2023. P. 209–22.*
16. *An automatic model management system and its implementation for AIOps on microservice platforms / Ruibo Chen, Yanjun Pu, Bowen Shi, Wenjun Wu // The Journal of Supercomputing. 2023, 79, P. 11410–11426.*
17. *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.*
18. *Implementing a Network-Aware Kubernetes Scheduler on top of a Mesh Network / J. Moeyersons, B. Stamper, B. Volckaert, F. De Turck // IEEE Cloud Summit. 2022.*
19. *Яровий О. В., Завгородній В. В. Розробка системи керування мобільними агентами // Таврійський науковий вісник. Серія: Технічні науки. 2023 (3). С. 29–36. URL: <https://doi.org/10.32782/tnv-tech.2023.3.4>*
20. *Kubeflow. Kubeflow Architecture. URL: <https://www.kubeflow.org/docs/started/architecture/>*
21. *IEEE 802.11s Mesh Networking.*
22. *Документи KubeEdge. URL: <https://kubedge.io/>.*
23. *Документи k3s. URL: <https://k3s.io/>.*
24. *Документи microk8s. URL: <https://microk8s.io/>.*
25. *Документи Edgemesh. URL: <https://edgemesh.netlify.app/guide/ha.html#introduction-to-basic-principles>.*

M. Yu. Kuzmich, T. B. Gordienko

IMPLEMENTATION OF KUBEFLOW MLOPS WITH A DISTRIBUTED INFORMATION SYSTEM OF MOBILE AGENTS WITH INCREASED FUNCTIONAL STABILITY

Currently, Machine Learning is one of the main tools for solving complex tasks in various fields of activity. The popularity of Machine Learning is caused by such factors as access to large data sets, the low price of computing resources, ready-made services based on cloud technologies, and rapid progress in the fields of artificial intelligence.

To develop, test and support the infrastructure of systems with data, in particular, the concepts of machine learning and processes (Machine Learning and Operation, MLOps) with a set of techniques for implementation and automatic continuous integration are applied. The concept of MLOps is considered in terms of Kubeflow tools — a cloud native system with open source code running on the Kubernetes platform. The possibilities of using the MLOps approach to improve the development processes of machine learning information systems are investigated. A distributed information system based on mobile agents acting as unmanned aerial vehicle (UAVs) is designed and their potential practical implementation in real-time conditions is briefly described.

It has been demonstrated that writing model code is only a small part of the tasks of Machine Learning, which affects the need for automation — the presence of a full-fledged pipeline of continuous integration and delivery of the application to the end user. Conducted studies have shown 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. This allows them to be launched on a variety of devices, including mobile agents such as UAVs.

The use of mesh networks to increase the functional stability of the distributed information system of mobile agents is proposed and substantiated. The architecture of the system concept was designed based on the Kubernetes k3s distribution, which allows you to work in the edge computing paradigm.

Keywords: Kubeflow; Kubernetes; MLOps; mobile agents; continuous learning; machine learning; Mesh network; management systems; management efficiency; distributed information system.

