УДК 004.89
DOI: 10.31673/2412-9070.2020.046880
Г. О. ГРИНКЕВИЧ, канд. техн. наук, доцент,
Державний університет телекомунікацій, Київ

МЕТОД РОЗГОРТАННЯ АРХІТЕКТУРИ МАШИННОГО НАВЧАННЯ ДЛЯ IOT-ПРИСТРОЇВ НА ОСНОВІ БЕЗСЕРВЕРНОЇ АРХІТЕКТУРИ

Запропоновано метод розгортання архітектури машинного навчання для IoT-пристроїв на основі безсерверної архітектури, названий MLAbosa. Метод MLAbosa може розгортати, планувати та динамічно керувати інструментами передавання даних, програмами для потокової трансляції, інструментами пакетної аналітики та інструментами візуалізації в усьому хмарному спектрі. Ця стаття описує архітектуру MLAbosa, висвітлюючи проблеми, які він вирішує.
Ключові слова: Інтернет речей; якість обслуговування; машинне навчання; штучний інтелект; глибокі нейронні мережі.

Вступ

Інтернет речей (IoT) забезпечує екосистему взаємозв'язаних пристроїв (наприклад, камер спостереження, переносних технологій, промислових давачів, розумних будівельних приладів, систем моніторингу здоров'я, транспортних засобів тощо), які генерують і трансформують великі обсяги даних із великою швидкістю, що потім аналізуються, аби здобути цінну інформацію та ухвалити обґрунтовані рішення для різних доменів розумних додатків. Наприклад, такі інтелектуальні додатки для аналізу даних, як керування продуктами, оптимізація процесів, відеоаналітика, прогнозова аналітика та рекомендаційні системи. Усі зазначені засоби покладаються на поточний та поглиблений аналіз вхідних потоків даних, а також історичних даних.

Зі збільшенням доступності даних із різних джерел та значними вдосконаленнями апаратного забезпечення та мережних вирішень, що роблять обчислення великих даних простішими та сприйнятливішими, з'являються численні бібліотеки та фреймворки машинного навчання, розроблені ще нещодавно для прогнозної аналітики. Відеоаналіз, виявлення об'єктів, програмні засоби розпізнавання мови, автономні машини (наземні, повітряні та підводні дрони), автоматизована дорожня сигналізація, промислова робототехніка — це далеко не повний перелік багатьох реальних додатків, яким потрібні вирішення машинного навчання (**ML** — *machine learning*) як частина своєї аналітики потокових масивів даних або поглибленого аналізу пакетної аналітики. Однак написання коду для завантаження, перетворення та попереднього оброблення даних, а також вибору правильного алгоритму для машинного навчання з подальшим оцінюванням моделі та налаштуванням гіперпараметрів вимагає значного досвіду. Усе зростаюча потреба і особливо перспективи, що відкриває використання прогнозної аналітики для вирішення різноманітних проблем, які мають суспільне та екологічне значення [1; 2], зобов'язує, щоб розроблення моделі ML було доступним навіть для початківців-користувачів (наприклад, лінійного персоналу низових державних структур).

Крім того, існує суєтний ажіотаж, зокрема щодо використання апаратних ресурсів (наприклад, графічних процесорів, TPU) поряд із хмарними інфраструктурними послугами. Робота з цією гетерогенністю потребує досвіду у виборі правильної конфігурації обладнання, яка може підвищити продуктивність та мінімізувати витрати, чого, як правило, не вистачає розробникам машинного навчання.

Отже, вимоги до керування життєвим циклом прогнозної аналітики подвійні:

- швидкий механізм розроблення моделі ML, де метою є допомогти розробникам алгоритмів ML побудувати моделі ML, використовуючи абстракції вищого рівня [3];
- структура швидкого розгортання моделі ML, де метою є допомогти розробникам розгорнути та інтегрувати навчені моделі аналітики на цільовому обладнанні та звільнити розробника від потреби з'ясовувати правильну конфігурацію для своїх робочих процесів ML в інфраструктурі.

З цією метою в статті запропоновано структуру під назвою MLAbosa, яка розкриває проблеми розвитку життєвого циклу, розгортання та керування аналітикою даних у неоднорідному розподіленому середовищі по всьому спектру хмарного туману. Також визначено бачення MLAbosa, його ключові особливості та архітектурні деталі, зокрема сфери застосування, де MLAbosa буде корисним.

Основна частина

Зрозуміло, що традиційні хмарні обчислення можуть не забезпечити властивості бажаної якості обслуговування (QoS) цих додатків для аналітики IoT через високу вартість переміщення великих обсягів даних у віддалені хмари та неприпустимі затримки туди-назад під час отримання критичних розумінь проблем домену. Зокрема, такі програми, як реагування на надзвичайні ситуації,

моніторинг стану здоров'я, інтелектуальні помічники потребують аналітичних можливостей у режимі реального часу з низькою затримкою. Хмарні обчислення дають змогу виконувати програми ближче до джерела даних, тим самим усуваючи багато проблем, які постають через необхідність користуватися віддаленою хмарою.

На жаль, розробники додатків для аналітики IoT зазвичай не мають досвіду для забезпечення корисних вирішень щодо розгортання необхідних додатків та динамічного керування ресурсами. Отже, існує нагальна потреба у підході, який позбавив би розробника додатків для аналітики IoT необхідності визначати розміщення компонентів додатків для аналітики, контролювати використання їх ресурсів та перевіряти різні завдання оброблення даних на хмарних пристроях, що забезпечують кращий підхід для підтримання оптимального режиму керування даними по всьому спектру ресурсів від хмари.

Безсерверні обчислення виявляють перспективність у вирішенні цих проблем, оскільки дають можливість розробникам додатків створювати основні компоненти програми, не турбуючись про конфліктуючі деталі інфраструктури. Зокрема, «Функції як послуга» (FaaS) — це концепція для досягнення безсерверних обчислень, що дає змогу розробникам виконувати код залежно від подій без створення або підтримання складної інфраструктури. Перевірка програми аналітики даних IoT розкриває структуру додатка, яка складається з набору таких вільно зв'язаних між собою служб, як передавання даних, потік та пакетне оброблення, машинне навчання як послуга, візуалізація та зберігання [4], де окремі компоненти взаємозв'язані Restful API. Слабозв'язаний характер та характер подій цих додатків роблять їх надзвичайно придатними для розміщення за допомогою безсерверної парадигми.

Ще одна проблема, яка постає перед розробниками додатків для аналітики IoT, полягає у створенні моделі штучного інтелекту/машинного навчання з використанням великих навчальних наборів даних [5]. Це потребує від розробників знання діапазону можливих моделей ML (наприклад, лінійних моделей, дерев вирішень або глибоких нейронних мереж), а також уміння вибирати необхідне з безлічі наявних бібліотек ML та фреймворків. До того ж, вони відповідають і за забезпечення високої якості прогнозування розроблених моделей ML, що значною мірою залежить від вибору характеристик та гіперпараметрів, які самі мають бути налаштовані на офлайн-процес оцінювання. Розробники додатків для аналітики IoT навряд чи будуть експертами в усіх цих напрямках, зокрема і у використанні навчених моделей під час виконання призначених функцій.

Шарове бачення та архітектура. Загальну архітектуру того, як аналітичний додаток може бути розгорнуто за допомогою інженерної моделі MLAbosa зображено на рис. 1. Мотивується сценарій використання ділянкової аналітики за допомогою інтелектуальної системи керування трафіком. Наприклад, камери дорожнього руху постійно нагромаджують відповідне відео, а крайні пристрої, інтегровані з можливостями розпізнавання зображень, замість того, щоб відправляти всі відео в хмару, можуть скористатися корисною інформацією, зокрема щодо обсягу трафіку, швидкості руху автомобілів та дорожніх випадків. На основі даних, зібраних протягом певного періоду, стосовно закономірностей руху та періодів його інтенсивності можна дізнатись через пакетну аналітику, що є обчислювально напруженим процесом, який зазвичай здійснюється в хмарі. Нарешті, інтелектуальна система керування дорожнім рухом, як правило, перебуває у вузлах туману для потреб у режимі реального часу, динамічно регулюючи синхронізацію сигналів світлофорів на основі вивченої моделі ML та аналізуючи дані в режимі реального часу.

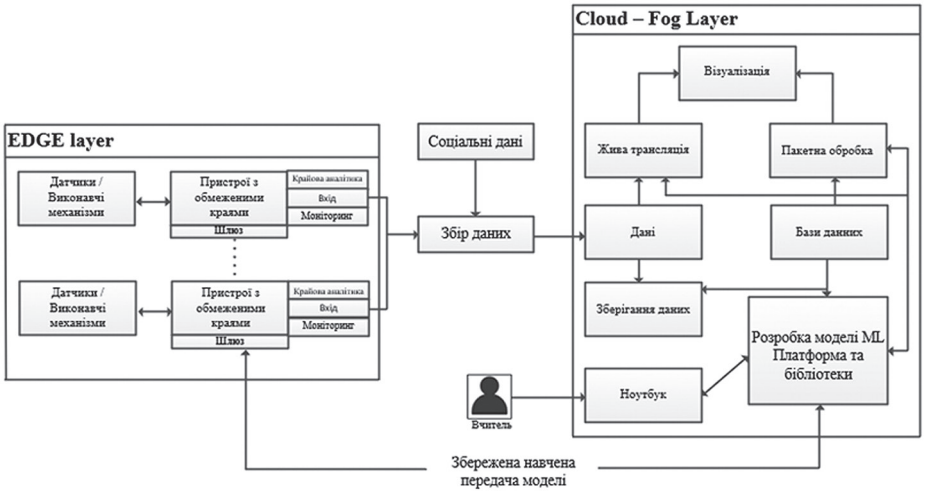

Рис. 1. Узагальнене подання архітектури додатків у моделі MLAbosa

Механізм розгортання MLAbosa за потреби може формувати інструменти передавання даних, засоби оброблення потоків, інструменти пакетної аналітики, платформу машинного навчання та фреймворк на цільовій машині (bare metal та віртуалізовані середовища) (рис. 2). Метод MLAbosa ґрунтується на специфічній для домену мові моделювання (*domain-specific modeling language* — **DSML**), яка надає розробникам пристроїв та розробникам ML інтерфейс користувача з абстракціями вищого рівня, а також на можливостях генерації коду механізму розгортання/керування, розробленого з використанням *Web Generic Modeling Environment* (**WebGME**). І DSML, і двигунці є розширюваними, модулізованими та багаторазовими.

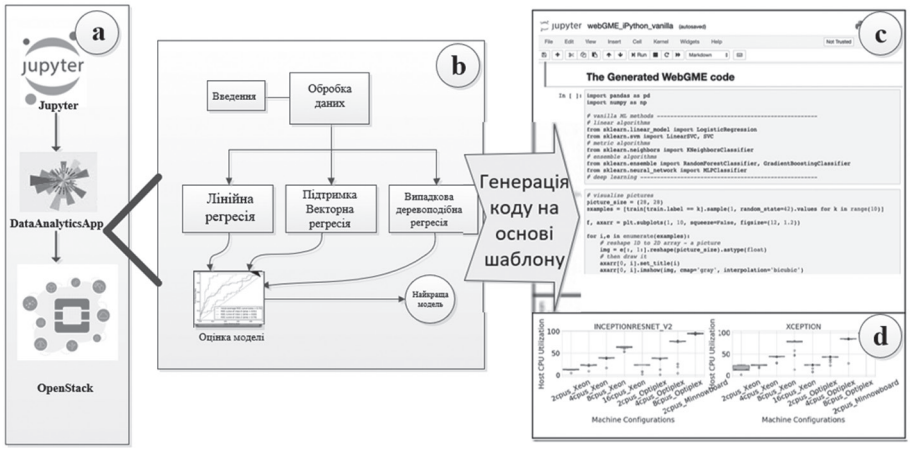

Рис. 2. Механізм розгортання MLAbosa:
***a* — робочий процес розгортання конвеєра ML; *b* — робочий процес розроблення рамок моделі ML;**
***c* — згенерований код у WebGME; *d* — метрики моніторингу ефективності**

Щойно модель ML буде побудовано та оцінено, фреймворк MLAbosa може її зберегти. Метод MLAbosa підтримує рознімну архітектуру, тому надані користувачем специфікації аналізуються та перетворюються на інфраструктуру рівня розгортання як код. Потім робочі процеси користувача ML розгортаються на відповідних машинах через хмарний туман, і безсерверна платформа виконання MLAbosa виділяє потрібні ресурси. Структура моніторингу ресурсів у рамках методу відстежує використання ресурсів і відповідає за ініціювання дій щодо еластичного масштабування ресурсів та міграції завдань, за потреби, для забезпечення якості службових потоків робочого процесу ML.

Метод MLAbosa було розроблено з урахуванням описаних раніше ключових вимог, а отже, підтримує такі функції:

♦ ***структуру розроблення моделі швидкого машинного навчання***. Структура розроблення моделі ML забезпечує швидке та гнучке використання найсучасніших можливостей ML. Вона надає підхід ML Service Encapsulation, використовуючи мікросервіс та архітектуру контейнеризації з підтриманням GPU та API, що абстрагують загальні бібліотеки та фреймворки ML. Вона також забезпечує просту у використанні масштабовану структуру для побудови та оцінювання моделей ML;

♦ ***структуру розгортання моделі швидкого машинного навчання***. MLAbosa забезпечує інтуїтивно зрозумілі абстракції на вищому рівні, щоб приховати складність розгортання та керування інфраструктурою нижчого рівня, а також забезпечує простий у використанні веб-інтерфейс для безпосередніх користувачів. DSML генерує інфраструктурний код «з правильною побудовою», використовуючи контролери обмежень, перш ніж перейти до фактичного розгортання;

♦ ***підтримання передавання моделі ML***. MLAbosa забезпечує інтелектуальний спосіб передавання навченої моделі на цільові машини (по всьому спектру хмари та краю туману) як модуль ML для умовиводу. Модуль ML може бути розміщено на крайніх пристроях або на шарі Cloud або Fog для реального або глибокого аналізу даних, що залежить від вимог користувача та аналізу потужності;

♦ ***розширюваність та багаторазовість***. MLAbosa реалізований модулізовано, і кожний модуль легко використовувати повторно завдяки архітектурі plug and play. У такий самий спосіб нове апаратне підтримання може бути об'єднано з MLAbosa стандартизованим способом.

Висновки

Оскільки аналітика, заснована на IoT, стає все більш досконалою, розробники виявляють недостатність досвіду в широкому діапазоні навичок і водночас перевантажені безліччю фреймворків, бібліотек, протоколів, мов програмування та обладнання, доступних для проектування та мож-

ливості розгорнути ці аналітичні програми. Тому запропонований метод розгортання архітектури машинного навчання для IoT-пристроїв на основі безсерверної архітектури, названий MLAbosa, зменшуватиме навантаження ЦОД, що реалізують хмарні технології.

Список використаної літератури

*1. **Інтелектуальні** машини на службі сучасного суспільства. URL:*

https://www.everest.ua/intelektualni-mashyny-na-sluzhbi-suchasnogo-suspilstva-2/. (дата звернення 15.01.2020).

*2. **Deep** learning applications and challenges in big data analytics / M. Najafabadi, F. Villanustre, M. Khosh [et al.] // Journal of Big Data. 2015. № 2(1):1. URL:*

*3. **Бойко С.** С чего начать работу с ML и DL. Обзор лучших библиотек. URL:*

https://dou.ua/lenta/articles/best-libraries-to-start-with-ml/. (дата звернення 23.02.2019).

*4. **Сервисы** Amazon ML: что такое AWS SageMaker. URL:*

https://senior.ua/articles/servisy-amazon-ml-chto-takoe-aws-sagemaker (дата звернення) 15.03.2020).

*5. **10 прикладів,** як штучний інтелект може змінити ваш спосіб життя. URL:*

https://www.radiosvoboda.org/a/29015231.html (дата звернення 23.05.2019).

А. А. Гринкевич

МЕТОД РАЗВЕРТЫВАНИЯ АРХИТЕКТУРЫ МАШИННОГО ОБУЧЕНИЯ ДЛЯ IOT-УСТРОЙСТВ НА ОСНОВЕ БЕССЕРВЕРНОЙ АРХИТЕКТУРЫ

Предложен метод развертывания архитектуры машинного обучения для IoT-устройств на основе бессерверной архитектуры, названный MLAbosa. Метод MLAbosa может разворачивать, планировать и динамично управлять инструментами передачи данных, программами для потоковой трансляции, инструментами пакетной аналитики и инструментами визуализации во всем облачном спектре. Эта статья описывает архитектуру MLAbosa, освещая проблемы, которые он решает.

Ключевые слова: Интернет вещей; качество обслуживания; машинное обучение; искусственный интеллект; глубокие нейронные сети.

G. Grynkevych

METHOD OF DEVELOPING MACHINE LEARNING ARCHITECTURE FOR IOT-DEVICES BASED ON SERVERLESS ARCHITECTURE

With the advent of IoT and microservice architectures, a multitude of intelligent distributed applications have emerged in which IoT devices collect, transform, and analyze data in large volumes and at high speed. A large number of these programs require robust, predictive analytics in real time, which require threading workflows closer to the data source, as well as dynamic resource management decisions. Moreover, predictive analytics requires the developer to create robust deep learning models, which, in turn, requires them to develop functions, find and configure parameters, and select machine learning models, which takes not only time, but also requires high experience level.

The proliferation of machine learning libraries and frameworks, data ingestion tools, streaming and batch processing engines, rendering techniques, and the myriad of available hardware platforms further exacerbate these challenges.

To overcome these complex challenges faced by developers of intelligent IoT applications, this article proposes a method for deploying machine learning architecture for IoT devices based on a serverless architecture called MLAbosa. The MLAbosa method can deploy, plan and dynamically manage data transfer tools, streaming software, batch analytics tools, and visualization tools across the cloud spectrum. This article describes the architecture of the MLAbosa method, highlighting the problems it solves.

Keywords: Internet of Things; quality of service; machine learning; artificial intelligence; deep neural networks.