THEORY OF INFORMATION TECHNOLOGIES AND SYSTEMS CONSTRUCTION

ТЕОРІЯ ПОБУДОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА СИСТЕМ

https://doi.org/10.15407/intechsys.2025.02.003
UDC 004.8 + 004.032.26

О.А. УРСАТЬЄВ, канд. техн. наук, старш. наук. співроб., провідн. наук. співроб.,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, м. Київ, 03187, Україна
https://org/0009-0009-8323-0525
aleksei@irtc.org.ua

О.Є. ВОЛКОВ, канд. техн. наук., старш. дослідник, директор,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, м. Київ, 03187, Україна
https://orcid.org/0000-0002-5418-6723
alexvolk@ukr.net

В.Г. ТКАЛЯ, аспірант,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, м. Київ, 03187, Україна
https://orcid.org/0009-0004-7870-3256
advv0207@gmail.com

АВТОМАТИЗОВАНЕ МАШИННЕ НАВЧАННЯ. СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ

Розглянуто автоматизоване машинне навчання як рішення на основі штучного інтелекту для потреби автоматизації наскрізного процесу застосування машинного навчання, тобто проєктування конвеєрів машинного навчання – послідовності кроків, які перетворюють необроблені дані на машинну модель, прийнятну для розгортання у практичному використанні. Присутність людини у цьому циклі має бути значно скорочена або її бажано зовсім виключити. Розглянуто напрям подальшого розвитку штучного інтелекту і автоматизованого машинного навчання та тенденції його розвитку.

Ключові слова: *автоматизоване машинне навчання, демократизація штучного інтелекту, керований даними штучний інтелект, глибоке посилене навчання, трансферне навчання.*

Cite: Урсатьєв О.А., Волков О.Є., Ткаля В.Г. Автоматизоване машинне навчання. Стан та перспективи розвитку. *Information Technologies and Systems*, Київ, 2025, Том 2 (2), 03—33. https://doi.org/ 10.15407/intechsys.2025.02.003

Вступ

На сучасному етапі дані перетворюються на капітал, який безпосередньо бере участь у формуванні та керуванні різними сферами діяльності людини. Яскравим прикладом цього може бути альтернативна цінність даних [1], зосереджена в необмеженому повторному використанні їх. Абсолютна цінність даних може набагато перевищувати ту, яку вдається отримати під час первинного використання. Інноваційні компанії можуть отримати приховану цінність і, потенційно, також величезні переваги, тобто, цінність даних треба розглядати як можливості їхнього подальшого використання, а не лише нинішнього.

Статтю присвячено огляду інформаційних послуг з автоматизації наскрізного процесу застосування та підвищення ефективності методів машинного навчання — сучасні підходи, методи, загалом технології у цій сфері, що конкурують внаслідок розвитку інтелектуальних засобів оброблення інформації, а іноді навіть перевершують експертів з машинного навчання.

Її метою є ознайомити фахівців, що потребують навичок роботи з даними, із завданнями, пов'язаними з обробленням даних, досконалим математичним апаратом та інструментарієм світового рівня для отримання з них інформації та знань.

Машинне навчання як один з інструментів штучного інтелекту

Сучасні платформи оброблення даних розширюють свої можливості, зокрема *Data Science* — на потребу прогностичної або, інакше, предикативної (*Predictive Analytics*) і рекомендаційної (*Prescriptive Analytics*) аналітики [1]. *Data Science* — це міждисциплінарна область, у якій застосовують наукові методи, процеси, алгоритми для отримання знань (*knowledge*) із різноманітних даних та для розуміння їх (*insights*). Застосовуючи методи та теорії в контексті математики, комп'ютерних наук та інформатики із залученням машинного навчання, вона пропонує концепцію об'єднання їх, для аналізування даних та розуміння реальних явищ, що їх представляють. Ця концепція відображає всі аспекти роботи з даними: здатність збирати та обробляти їх, вміти розуміти, отримувати цінність, візуалізувати. Отже, завдання, вирішуване тими, хто займається *Data Science*, полягає в видобуванні знань із даних з метою розуміння того, що містять у собі ці дані, а самі дані не є для *Data Science* предметом цієї науки. Інтелектуальний аналіз даних та Великі Дані є розділами *Data Science*.

В області *Data Science* машинне навчання (*machine-learning*, *ML*), як один з інструментів дослідження даних, охоплює автоматизовані форми статистичних алгоритмів, самонавчальних на даних. *Data science* — це ключова дисципліна в розвитку штучного інтелекту (*AI*),

а *ML* — це основний інструмент *AI*. Поняття «штучний інтелект» охоплює багато різних технологій і ніколи не належав до конкретного типу. Це, радше, «соціотехнічна конструкція» [2] — термін, який застосовують для позначення можливостей машин, що вирішують складні завдання, які донедавна могли розв'язувати лише люди. Нинішній інтерес до *AI*, ажіотаж навколо нього, пов'язаний насамперед із машинним навчанням, що дає алгоритмам змогу навчатися самостійно, не будучи явно запрограмованими, та поглиблювати свій досвід. Машинне навчання в бізнесі спрямоване на створення та навчання моделей. *AI* використовує ці машинні моделі, щоб коригувати ситуацію за певних обставин. Він перебуває на іншому рівні агрегації, ніж *Data Science* та *ML* — на рівні застосунку. Створена прогнозна математична модель процесу і *ML*-моделі мають бути об'днані для інших можливостей спільної роботи, таких як інтерфейс користувача та керування робочим процесом для створення програми *AI*. Структури *AI* дають змогу розширити аналітичні можливості поза традиційну *Data Science* і *ML* (рис.1). Ці рамки охоплюють потужні, сфокусовані інструменти великої аналітики, такі як самонавчання та посилене навчання (в зарубіжній літературі — неконтрольоване навчання (*Unsupervised Learning, UL*) глибоке[1] (*Deep Learning, DL*) та навчання з підкріпленням (*Reinforcement Learning, RL*). Це особливий тип машинного навчання, в якому шари нейронних мереж імітують роботу нашого мозку, і де останнім часом було досягнуто значного прогресу. Технологія була застосована в області маркування зображень, транскрипції голосу, автоматичного перекладу, безпілотних автомобілів і показала дуже хороші результати в порівнянні зі звичайним *ML* [2].

Когнітивні обчислення — це ще один термін, який зазвичай використовується для позначення найскладніших типів технологій штучного інтелекту, які намагаються імітувати людське мислення. Але його точне значення неясне, і багато аналітиків і постачальників уникають цього терміну, оскільки він дуже тісно пов'язаний з *IBM* [2].

[1] Термін «глибоке навчання» виник у 1980-х роках. Цей підхід до створення *AI* передбачає використання моделі нейронних мереж, які сприймають інформацію, що надходить, шарами. Кожен із таких шарів нейронів, об'єднаних в єдину мережу, оцінює отриману інформацію за якоюсь однією ознакою. Отримані дані сумуються мережею видачі кінцевого результату оцінки. Лауреатами премії імені Алана Тьюринґа за 2018 рік стали «хрещені батьки» штучного інтелекту — чені Джеффрі Хінтон, Янн ЛеКун та Йошуа Бенжіо. Зі слів Асоціації обчислювальної техніки, усі троє розробили концептуальні основи та інженерні рішення, які зробили глибинні нейронні мережі наріжним компонентом в обчислювальній техніці, і які сприяли розвитку технологій штучного інтелекту. Саме глибоке машинне навчання з використанням нейромереж стало передовим напрямом у галузі створення *AI*. Премію Тьюринґа отримали розробники штучного інтелекту з *Google* та *Facebook*. Березень, 2019, ФОКУС — *https://focus.ua/technologies/424630-premiyu-tyuringa-poluchili-razrabotchiki-iskusstvennogo-intellekta-iz-google-i-facebook*

Рис. 1. Крива зрілості інноваційних технологій або цикл очікувань (*The Gartner hype cycle*[2])

Автоматизоване машинне навчання (Automated Machine Learning). Машинне навчання досягло значних успіхів, і дедалі більше дисциплін покладаються на нього. Однак ці успіхи визначаються насамперед внеском науковців у розробку теоретичних питань, наприклад [3], і значною мірою залежать від фахівців з *ML*, які обирають парадигми машинного навчання, відповідні функції, робочі процеси, алгоритми та гіперпараметри. Автоматизоване машинне навчання (*AutoML*) — це сфера *ML*, яка спрямована на розроблення рішень, що дають змогу різнобічним спеціалістам у науці, експертам і фахівцям, добре обізнаним з предметною областю ПрО (*citizen data scientist*) та з моделями, генерувати останні, застосовуючи прогнозну чи рекомендаційну аналітику (але основна частина роботи при цьому перебуває поза сферою статистики та аналітики), ефективно створювати точні прогностичні моделі з мінімальним втручанням людського інтелекту. Видобування знань із даних охоплює кілька етапів, таких як: вибір підмножини даних, очищення їх і перетворення, вибір функцій чи проєктування їх, а також застосування відповідних методів аналізу даних для вибору шаблонів (*pattern*) [1], налаштування алгоритму та оптимізацію гіперпараметрів моделі, оцінювання та інтерпретацію результатів і розгортання моделей машинного навчання. *AutoML* підвищує ефективність машинного навчання та прискорює дослідження у цій сфері. Алгоритм *AutoML* подано на рис. 2 [4], а успіх його реалізації вирішальною мірою залежить від експертів

[2] «*The Gartner hype cycle*» — «Цикл хайпа *Gartner*» або «Цикл очікувань» — крива входження або зрілості інноваційних технологій. Цикл очікувань *Gartner* — це графічна презентація, розроблена, використана та брендована американською дослідницькою, консультативною та інформаційною компанією *Gartner*, щоб показати зрілість, впровадження та соціальне застосування конкретних технологій. Цикл очікувань стверджує, що забезпечує графічне та концептуальне подання зрілості нових технологій у п'ять етапів. Модель піддавалася критиці з різних причин, у тому числі через ненаукову точність і використання суб'єктивної термінології — *https://en.wikipedia.org/wiki/Gartner_hype_cycle*

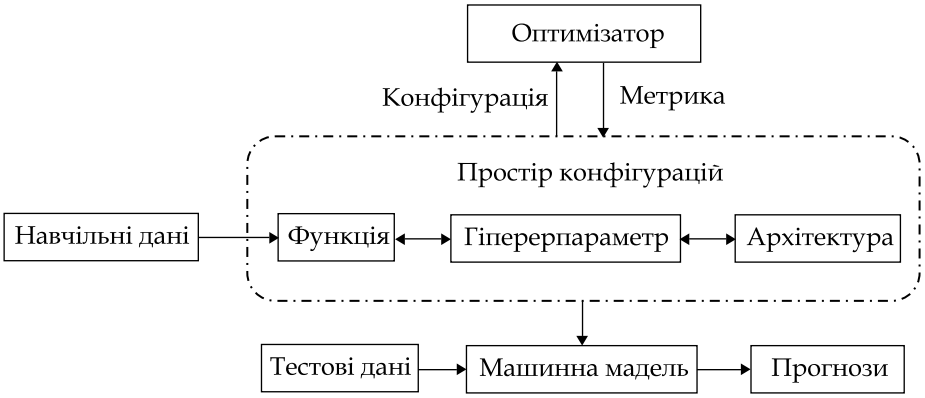

Рис. 2. Алгоритм автоматизованого машинного навчання

з *ML*, здатних виконати наведені завдання, із залученням глибокої експертизі фахівців з ПрО.

Історія створення *AutoML* сягає початку 2010-х років, коли фахівці в області *ML* почали вивчати способи автоматизації процесу розроблення моделей машинного навчання. У минулому десятилітті сфера досліджень, спрямована на завдання прогресивної автоматизації *AutoML*, була сформульована в [5, 6] як вирішення проблеми автоматизації ефективного виявлення об'ктів (*Object Detection, OD*) та проєктування конвеєрів *ML* – послідовності кроків, які перетворюють необроблені дані на машинну модель, прийнятну для виробництва. Фахівці з даних (*data scientists and researchers; citizen data scientists* — науковці та дослідники даних; фахівці, які добре обізнані з предметною областю) мали знайти правильний спосіб перетворення даних, щоб подавати їх у конвеєр *ML*. Будь-яка реальність (статичний чи динамічний розвиток подій) передбачає підготовку даних або їх попереднє оброблення [1]. Було досягнуто значного прогресу в наскрізному автоматичному *ML* для вирішення складніших завдань, таких як оброблення тексту, зображень, відео та мовлення з використанням методів самонавчання та посиленого навчання. Однак і ці методи мають багато варіантів моделювання та гіперпараметрів [5, 6].

Методи оптимізації гіперпараметрів

Гіперпараметри моделі можуть охоплювати не лише змінні, що відповідають перемиканню між альтернативними моделями, а й варіанти моделювання, такі як параметри попереднього оброблення, топологію та розмір нейронної мережі або параметри регуляризації алгоритму навчання — долучення деяких обмежень з метою запобігти перенавчанню, наприклад, деяких апріорних розподілів на параметри моделі. Оптимізація гіперпараметрів (ОГП, англ. *HPO*) дає змогу суттєво поліпшити продуктивність алгоритмів *ML*, адаптуючи

їх до цієї проблеми. Машинна модель має такий вигляд:

$$y = f(x; \theta) = f(x; a(\theta), \theta).$$

Тут вектор параметра моделі a є неявною функцією вектора гіперпараметра, отриманого для фіксованого значення θ, і навчальних даних *Dtrain*.

Спочатку проблему *AutoML* розв'язували за допомоги оптимізації гіперпараметрів. Це рішення передбачало, що є обраний алгоритм машинного навчання A, який має гіперпараметри $\theta_1, \ldots, \theta_n$ з відповідними областями визначення $\theta_1, \ldots, \theta_n$ і сумісний простір у вигляді багатомірного вектора $\theta = \theta 1 \times \ldots \times \theta 1_n$. Для кожного набору гіперпараметрів $\theta_i \in \Theta_i$ застосовується символ A_θ для позначення алгоритму навчання A із цим налаштуванням. Для позначення втрат під час перевірки (наприклад, рівня помилкової класифікації), якого досягає A_θ на даних *Dvalid* в разі навчання на *Dtrain*, користуються виразом $l(\theta) = L(A_\theta; Dtrain; Dvalid)$. Завдання оптимізації гіперпараметрів полягає в тому, аби знайти $\theta \in \Theta$, що мінімізує $l(\theta)$ [7]. Існує інше визначення, яке для багатьох алгоритмів машинного навчання можна сформулювати як мінімізацію критерію навчання, що включає гіперпараметр [8]. Цей гіперпараметр зазвичай обирається методом спроб та помилок з використанням критерію вибору моделі.

Стандартом де-факто оптимізації гіперпараметрів у машинному навчанні був простий пошук у сітці (*grid-search*), тобто вичерпний пошук серед усіх можливих комбінацій дискредитованої сукупності параметрів, який імовірно, буде обчислювально забороненим у великих просторах пошуку або з великими наборами даних його вельми проблематично використовувати. Для гіперпараметричної оптимізації, особливо в багатовимірних просторах, вважаються ефективнішими випадково вибрані випробування (*random search*), де простір досліджується протягом обмеженого часу, порівняно з пошуком по сітці. Його здійснюють за заздалегідь заданим розподілом імовірностей у просторі пошуку, і він є найпростішим, але досить ефективним на практиці підходом [9]. Є також інші підходи, такі як градієнтний пошук. У [8] наведено методологію оптимізації декількох гіперпараметрів, засновану на обчисленні градієнта критерію вибору моделі стосовно гіперпараметрів. Тут показано, що випадковий пошук є ефективнішим для оптимізації гіперпараметрів, ніж пошук по сітці. У порівнянні з нейронними мережами, налаштованими на чистий пошук по сітці, виявили, що випадковий пошук у тій же області дає змогу знаходити моделі, які є кращими за часом обчислень. Випадковий пошук при тому самому обчислювальному бюджеті знаходить кращі моделі за допомоги ефективного пошуку у більш-менш перспективному конфігураційному просторі. Є ще одне важливе зауваження: для наборів даних лише кілька гіперпараметрів справді

мають значення, але різні гіперпараметри є важливими для різних наборів даних. Це явище робить пошук у мережі поганим вибором для налаштування алгоритмів під нові набори даних [8–10]. Також є інші адаптивні (послідовні) методи, наприклад, еволюційний пошук і баєсівська оптимізація. Хоча ці адаптивні підходи відрізняються тим, як вони визначають, які моделі оцінювати, всі вони намагаються змістити пошук убік конфігурацій, які із більшою імовірністю працюватимуть добре.

Через базові припущення, зроблені завдяки різним методам пошуку, вибір відповідного методу може залежати від простору пошуку. Баєсівські підходи (БП) за допомоги гаусівських процесів для моделювання характеристик ефективності узагальнення, наприклад [11], і градієнтні підходи [12] зазвичай можуть бути застосованими до просторів безперервного пошуку. Баєсівські методи оптимізації часто поєднують з іншими методами, наприклад, такими, як ансамблеві методи, щоб отримати перевагу в деяких ситуаціях з обмеженням за часом [5]. Деякі з цих методів спільно розглядають дворівневу оптимізацію і приймають витрати часу як важливу установку для пошуку гіперпараметрів. БП на основі дерев [13], еволюційні стратегії [14] і випадковий пошук є гнучкішими, їх можна застосовувати до будь-якого простору пошуку. Застосування посиленого навчання до загальних завдань оптимізації гіперпараметрів є обмеженим через складність вивчення політики поведінки над великими просторами безперервної дії [5, 15].

БП, наприклад [13], що моделюють умовну імовірність продуктивності $p(y \mid \theta)$ за метрикою оцінки Y (тобто, точністю тесту), з огляду на конфігурацію набору гіперпараметрів Θ, спрямовані на швидше визначення перспективних змін. Вони забезпечують статистично стійкі (надійні) та виграшні конфігурації гіперпараметра моделі. Налаштування гіперпараметра виконується послідовно та концентрується на перспективних областях простору пошуку. БП вирішують принципово складне завдання одночасного підбору та оптимізації багатомірного простору, зокрема неопуклих функцій, і, можливо, оцінки шуму. Для вирішення цієї проблеми методи БП застосовують евристику для моделювання цільової функції або пришвидшення підпрограм ресурсів. Однак адаптивні методи вибору конфігурації за своєю суттю є послідовними, і їх важко розпаралелювати, що потрібно для пришвидшення обчислень.

Оскільки значення гіперпараметрів можуть ефективно впливати на кінцеву продуктивність робочого процесу, проблема полягає не лише в розмірі пошукового простору, а й у керуванні часом, необхідним для оцінювання кожного можливого рішення. Ця проблема містить і методи пошуку передбачуваних змін властивостей у просторі пошуку і оцінювання *ML*-моделей, тобто визначення виграшної (*scoring*) конфігурації гіперпараметра з набору можливих значень.

Простір пошуку містить цей набір конфігурацій і може вміщувати безперервні або дискретні гіперпараметри.

Байєсівська оптимізація (Bayesian Optimization). Автоматичний вибір робочого процесу було спочатку формалізовано в *Auto-Weka*[3] як завдання вибору комбінованого алгоритму та оптимізації гіперпараметрів (*CASH Combined algorithm selection and hyperparameter optimization*). Оптимізацію вибору моделі виконано за дворівневою програмою із зазначенням нижньої мети $J1$ для навчання параметрів моделі та верхньої $J2$ – для навчання гіперпараметрів θ. Обидва оптимізуються одночасно. Для визначення простору гіперпараметрів використано структуру дерев [16]. Баєсівські процедури оптимізації [5, 9, 11, 15–18]: послідовний алгоритм налаштування моделюванням (*SMAC – Sequential Model-based Algorithm Configuration*), оцінювач Парзена з деревоподібною структурою (*TPE, But Tree-structured Parzen Estimator*) і *Spearmint* (*TPE* не є стандартним баєсівським алгоритмом оптимізації [9]). *SMAC* використовує випадкові ліси для моделювання $p(y|\theta)$ як розподіл Гауса, а потім *Spearmint*, застосовуючи гаусові процеси (*GP*) для моделювання $p(y|\theta)$, виконує вибірку зрізів за гіперпараметрами *GPs*. Було виявлено [9, 19], що деревоподібний метод оцінювання Парзена, який моделює $p(x|y)$ і $p(y)$, а не $p(y|x)$ безпосередньо, перевершує методи баєсівської оптимізації на основі гаусівських процесів для задач структурованої оптимізації з багатьма гіперпараметрами, зокрема дискретні. Основна ідея цих методів полягає в тому, щоб підігнати $J2(\theta)$ до гладкої функції у спробі зменшити дисперсію та оцінити її в тих областях гіперпараметрічного простору, які недостатньо відібрано для направлення пошуку в області високої дисперсії.

Алгоритм налаштування моделюванням *SMAC*, який допоміг на кілька порядків пришвидшити і локальний пошук, і пошуки у дереві на певних розподілах примірників, виявився ефективним для гіпероптимізації параметрів алгоритмів машинного навчання, забезпечив краще масштабування до великих розмірів і дискретних вхідних вимірів. Пошук в об'єднаному просторі алгоритмів і гіперпараметрів дав ефективніші моделі, ніж вибір стандартних алгоритмів і гіперпараметра.

SMAC засновано на випадкових лісах та *Tree Parzen Estimator*, які підходять для оптимізації багатовимірних гіперпараметрів у глибоких нейронних мережах [19]. Мета байєсівської оптимізації – скласти імовірнісну модель і потім використовувати її для ухвалення рішення про те, як слід оцінювати наступну функцію. Принцип полягає у застосуванні інформації, отриманої в результаті попередньої

[3] *Weka* – популярна бібліотека машинного навчання, одна з застосовуваних бібліотек *ML*, яка містить множину методів попереднього оброблення та алгоритмів *ML*, які перевершують інші бібліотеки.

оцінки, що дає змогу знайти мінімум неопуклої складної функції для оптимізації гіперпараметрів [20].

В результаті цих розробок було створено інструментарій для автоматизації конвеєрів машинного навчання методом баєсівської оптимізації, і введено велику бібліотеку тестів для оптимізації гіперпараметрів під назвою *HPOlib*.

В [9, 17] показано, що баєсівські методи оптимізації емпірично перевершують випадковий пошук на кількох тестових завданнях. Однак для задач великої розмірності стандартні баєсівські методи оптимізації працюють аналогічно до випадкового пошуку. Методи, спеціально розроблені для задач великої розмірності, передбачають нижчу ефективну розмірність проблеми[4] або деревоподібне розкладання для цільової функції.

Інші рішення *AutoML* також спрямовано на пошук конвеєрів *ML* з хорошою продуктивністю. Розробки *Auto-Weka*, *Auto-Sklearn*, *PoSH Auto-Sklearn*, *TPOT* та інші вирішують, принаймні, проблему *CASH*.

Байєсівська оптимізація з послідовним поділом наявних ресурсів навпіл (Bayesian Optimization with SH Algorithm). PoSH Auto-Sklearn[5] поєднує в собі попередньо обраний портфель машинних навчальних конвеєрів, побудову ансамблю для досягнення стійкої продуктивності та баєсівську оптимізацію з послідовним поділом наявних ресурсів навпіл (*Bayesian Optimization with Successive Halving*) [21]. Спочатку запускається одна ітерація алгоритму *SHA* (*Successive Halving Algorithm*) із портфоліо машинних навчальних конвеєрів, а потім використовується баєсівська оптимізація для отримання нових конфігурацій гіперпараметра. Ідея, покладена в основу *SHA*, випливає безпосередньо з його назви: рівномірно розподілити бюджет по набору конфігурацій гіперпараметрів, оцінити продуктивність будь-яких змін, відкинути найгіршу половину й повторювати доти, поки не залишиться одна конфігурація. Таким чином, стратегія полягає в усуненні найгіршої половини конфігурацій гіперпараметрів і у збільшенні кількості часу на ітерації процесу, що залишилися. При цьому витрачається значно менше часу, наскільки це можливо, на претендентів (кандидатів) і відкидання непридатних, що дає змогу приділити більше часу перспективним. Алгоритм *SH* виділяє експоненціально більше ресурсів для перспективних конфігурацій. У машинному навчанні таке завдання — вибір із ряду грального автоматів типу «однорукий бандит» того з них, який забезпечить виграш за мінімальної витрати ресурсу (обмеження в грошах і часі) — отримало назву «бандитського» алгоритму або багаторуких бандитів (*multi-ar-*

[4] *Wang Y. et. al., Automatic reconstruction of fault networks from seismicity catalogs including location uncertainty,* 2013, 36 p. — *https://arxiv.org/ftp/arxiv/papers/1304/1304.6912.pdf*

[5] Це ім'я є абревіатурою *PortfolioSuccessiveHalving*, об'єднаної з бібліотекою *ML Auto-sklearn*.

med bandits, *MAB*). Його постановка називається завданням про багаторукого бандита. Останнім часом *SHA* було успішно застосовано до оптимізації гіперпараметрів, демонструючи високу продуктивність за невеликих бюджетів.

SHA дає змогу вказати мінімальний і максимальний бюджет для кожної конфігурації (наприклад, під кутом зору кількості навчальних ітерацій або розміру набору даних). Він починає з мінімального бюджету й ітеративно збільшує бюджет для конвеєрів, які працюють із поточним бюджетом.

Простір конфігурації є підпростором *Auto-sklearncon-figuration*, що дозволяє працювати з бюджетами *SH*: попередні оброблення набору даних (масштабування об'єктів, обчислення пропущених значень, обробка категоріальних значень) і один із чотирьох класифікаторів: метод опорних векторів (*SVM*), випадковий ліс, лінійна класифікація (за допомоги стохастичного градієнтного спуску) або *XGBoost*, що призводить до 37 гіперпараметрів.

Successive Halving покращує оптимізацію гіперпараметра у випадковому пошуку за допомоги ділення та вибору випадково згенерованих конфігурацій гіперпараметрів ефективніше, ніж *Random Search*, не роблячи припущень щодо природи простору конфігурацій. Ефективніший розподіл ресурсу означає, що, застосовуючи той самий набір конфігурацій, *Successive Halving* знайде оптимальну конфігурацію швидше.

Оптимізація гіперпараметрів (Hyperband). *Hyperband* — це алгоритм адаптивного розподілу ресурсів із принципового раннього зупинення оптимізації гіперпараметрів [15, 22]. Другий приклад алгоритмів, що оптимізують розподіл ресурсів — це *Hyperband* і асинхронний *Successive Halving*. На жаль, оригінальний алгоритм *SH* потребує в ролі вхідних даних кількість n конфігурацій. З огляду на деякий кінцевий бюджет B (наприклад, час навчання для вибору конфігурації гіперпараметра), ресурси B/n розподіляються в середньому за конфігураціями. Проте, для фіксованого B, не зрозуміло, чи має апріорі розглядатися (*a*) безліч конфігурацій (велике n) із невеликим середнім часом навчання; або (*b*) невелика кількість конфігурацій (мале n) з тривалішим середнім часом навчання. Інакше кажучи, алгоритму *SH* властива дилема «n проти B/n». Відсутність цієї додаткової інформації перешкоджає застосуванню методу оцінювання конфігурації, тим самим мотивуючи нестохастичну установку для оптимізації гіперпараметрів [21]. У літературі про *MAB* для визначення кумулятивної помилки (*regret*, регрет) алгоритму провісника використовуються алгоритми та здійснюються дослідження і для нестохастичних, і для стохастичних параметрів, тоді як оптимальну схему ідентифікації розглянуто лише в стохастичної постановці. Нестохастичне завдання ідентифікації найкращої якості не здійснювали [21].

Потрібен алгоритм, який за будь-яких значень параметрів забезпечив би теоретично обґрунтоване мінімальне значення *regret*, яке визначають відстанню від оптимального рішення, за максимально короткий час. Це в чистому вигляді є завданням дослідження *MAB*, яке могло би забезпечити теоретичні гарантії та обнадійливі емпіричні результати в налаштуванні гіперпараметра.

Гіперпараметрічну оптимізацію формулюють як проблему нестохастичного бандита з нескінченним озброєнням (*non-stochastic infinite-armed bandit problem*), де визначений ресурс, такий як ітерації для ітераційних алгоритмів, вибірки даних або функції, виділяють для випадково обраних конфігурацій. *Hyperband* розв'язує проблему «*n* проти *B*/*n*», розглядаючи кілька можливих значень *n* для фіксованого *B*. По суті, виконується пошук по сітці за допустимим значенням *n*. З кожним значенням *n* пов'язано мінімальний ресурс *r*, який виділяється всім конфігураціям до того, як деякі з них відкидаються; більше значення *n* відповідає меншому *r* і, отже, агресивнішому ранньому зупиненню. *Hyperband* розширює алгоритм послідовного розподілу навпіл *SH* [21], застосовуючи його як підпрограму організації внутрішнього циклу *Hyperband* (внутрішній цикл викликає *Successive Halving* для фіксованих значень *n* і *r*) з автоматизацією вибору швидкості раннього зупинення за допомоги запускання різних варіантів *SHA*.

Як зазначено в працях [15, 22], *Hyperband* надає надійний, теоретично обґрунтований алгоритм адаптивного розподілу ресурсів і принципового раннього зупинення оптимізації гіперпараметрів. Він спрямований на пришвидшення випадкового пошуку [9], розроблений для нестохастичного настроювання, та автоматично адаптується до невідомого *F* (незалежно від рівня гладкості або структури функції *F*, що оптимізується) без будь-яких параметричних припущень. Випадковий пошук буде асимптотично сходитися до оптимальної конфігурації за будь-якого *F* за допомоги простого аргументу покриття. Швидкість збіжності для випадкового пошуку залежить від гладкості та є експоненційною за кількістю вимірів у просторі пошуку. Це є так само вірним і для баєсівських методів оптимізації без додаткових структурних припущень. Отже, створено новий підхід [15] до оцінювання конфігурації як вирішення проблеми адаптивного розподілу ресурсів випадково обраних конфігурацій гіперпараметрів. Процедура *Hyperband* спирається на принципову стратегію раннього зупинення для розподілу ресурсів, даючи змогу оцінювати на порядки більше конфігурацій, ніж процедури «чорного ящика», такі як баєсівські методи оптимізації. Порівнюється робота цього алгоритму з популярними методами баєсівської оптимізації на наборі завдань оптимізації гіперпараметрів. Спостерігається, що *Hyperband* може забезпечити швидкодію вищою на порядок, порівняно з конкурентами, на множині проблем посиленого навчання та навчання на

основі ядра. Так, *Hyperband* працює у 5—30 разів швидше, ніж баєсівські алгоритми оптимізації. Однак він не годиться для паралельного настроювання під час роботи з великими даними з огляду на схильність відставати в обробленні даних та з тієї ж причини пропущені завдання [15].

Асинхронне послідовне подвійне скорочення (Asynchronous Successive Halving). Сучасні моделі навчання характеризуються великими просторами гіперпараметрів і тривалим часом навчання. Ці властивості зумовлюють необхідність розпаралелювання обчислень і, тим самим, мотивують необхідність розроблення досконалих функцій оптимізації гіперпараметрів в умовах розподілених обчислень, щоб мати можливість знайти добру конфігурацію за розумний час. Користуватися випадковим пошуком і пошуком по сітці за паралельного налаштування, оскільки ці два методи є тривіальними для розпаралелювання та легко масштабуються на будь-яку кількість машин, не є доцільним, оскільки обидва методи є підходами «грубої сили» і погано масштабуються з кількістю гіперпараметрів. За послідовного налаштування застосовують *SH*, добре відомий алгоритм багаторукого бандита (*SHA*). Тоді можна оцінити на кілька порядків більше конфігурацій гіперпараметрів, ніж в разі випадкового пошуку, внаслідок адаптивного розподілу ресурсів для перспективних конфігурацій. Цим досягається швидше дослідження простору пошуку та стандартно висока якість для великих наборів даних. Однак розпаралелити процес обчислень важко, тому що алгоритм приймає набір конфігурацій як вхідні дані, й очікує завершення будь-яких змін у ланцюжку, перш ніж відбудеться перехід до наступної ланки ланцюжка. Щоб усунути це вузьке місце, що створюється синхронними просуваннями, автори [15, 23] налаштовують алгоритм *SH* так, щоб просувати вгору конфігурації наскільки це можливо, замість того, щоб починати з широкого набору конфігурацій і звужувати їх. Такий алгоритм назвали асинхронним алгоритмом послідовного розподілу навпіл (*ASHA*). Автори стверджують, що налаштування обчислювально важких моделей із використанням масового паралелізму є новою парадигмою оптимізації гіперпараметрів.

ASHA використовує паралелізм і агресивне раннє зупинення для вирішення великомасштабних завдань оптимізації гіперпараметрів. Чималі емпіричні результати показали, що *ASHA* перевершує наявні сучасні методи оптимізації гіперпараметрів, масштабується лінійно з кількістю робочих вузлів у розподілених налаштуваннях, і годиться для масового паралелізму.

Еволюційна оптимізація. Інші методи пошуку застосовують еволюційні алгоритми. Ці підходи враховують сукупність налаштувань гіперпараметрів, модифікують і усувають безперспективні налаштування відповідно до їхньої оцінки перехресної перевірки. Але по-

при разючі результати, вони потребують величезної кількості обчислювальних ресурсів, і їх важко масштабувати.

TPOT (*Tree-Based Pipeline Optimization Tool*) [14] — деревоподібний інструментарій проєктування й оптимізації конвеєрів *ML*, працює як і *Auto-Sklearn*, у тандемі з бібліотекою *scikit-learn*, описуючи себе як її оболонку на *Python*. У його основі лежить платформа *DEAP*[6], модульна структура (*a modular framework*), яка допомагає реалізовувати процеси, описані еволюційними алгоритмами (*EA*). Реалізація застосовує генетичне програмування (*GP*) — еволюційну техніку обчислень для автоматичного конструювання комп'ютерних програм — із залученням *ES* (*Evolution strategy*)[7] стратегії.

Успішні розробки у сфері еволюційної багатоцільової оптимізації (*Multi-Objective MO*) і застосування концепції парето-оптимальності до машинного навчання підвищують продуктивність також і традиційних методів одноцільового машинного навчання, створюючи різноманітні множинні парето-оптимальні моделі під час побудови ансамблів моделей. Ці моделі відповідають теоретичним вимогам хороших ансамблів. Даючи змогу користувачеві генерувати ансамблі і вибирати моделі відповідно до потреб, багатоцільовий підхід тим самим прагне поліпшити алгоритми машинного навчання, долаючи спільні проблеми балансу компромісу між точністю та різноманітністю. У цьому сенсі *MO* має перевагу перед одноцільовими методами, оскільки під кутом зору багатоцільової оптимізації не має єдиної моделі навчання, яка могла б одночасно вирішувати різні завдання. Багатоцільова оптимізація на основі принципа Парето — єдиний спосіб впоратися з конфліктними цілями [24, 25].

---

[6] *DEAP* (*Distributed Evolutionary Algorithms в Python*) — це еволюційне обчислювальне середовище для швидкого прототипування та тестування ідей. Вільно доступне, з великою документацією (http://deap.gel.ulaval.ca), *DEAP* є проєктом з відкритим вихідним кодом під ліцензією *LGPL*. Еволюційні обчислення (*EC*) — складна область із дуже різноманітними методами та механізмами, де навіть добре спроєктовані структури можуть стати доволі складними, щоб розглянути можливість внесення змін. Мова програмування *Python* забезпечує необхідний зв'язок для збирання складних *EC*-систем, надаючи практичні інструменти для швидкого створення прототипів для користувача еволюційних алгоритмів, де кожен крок процесу є настільки явним (як псевдокод), легко прочитуваним і зрозумілим, наскільки це можливо. Це також надає великого значення і компактності, і зрозумілості коду — [*Fortin F.-A., Rainville F.-M. D., Gardner M.-A., Parizeau M. and Gagn C., Jul. 2012, DEAP: Evolutionary Algorithms Made Easy, Journal of Machine Learning Research, vol.* 13, *pp.* 2171–2175].

[7] *ES* — евристичний метод оптимізації в розділі еволюційних алгоритмів, базований на адаптації та еволюції. Розроблено в 1964 році німецьким ученим І. Рехенберге і розвинено в подальшому Швефелом Х.-П. та ін — [*Schwefel Hans-Paul. Cybernetic Evolution as Strategy for Experimental Research in Fluid Mechanics (in German). Diploma Thesis. Hermann Föttinger-Institute for Fluid Mechanics, Technical University of Berlin, March 1965. H.-G. Beyer and H.-P. Schwefel, Evolution strategies. A comprehensive introduction, Natural Computing, vol. 1, no. 1, pp.* 352, *Mar.* 2002].

Об'єднання *GP* і оптимізації за Парето дає змогу автоматично створювати високоточні та компактні *ML*-конвеєри внаслідок того, що еволюційні алгоритми на основі цього принципа успішно застосовують для оцінювання багатьох рішень, тобто послідовності конвеєрів. Ця особливість призводить до повільної оптимізації, тому що навчання є трудомістким процесом. У робочому процесі *AutoML* є попереднє оброблення функцій (*Feature Preprocessing*) — це очікуваний крок у конвеєрах машинного навчання, бо з'являється можливість змінювати функції[8] так, щоб спростити набір даних і забезпечити продуктивність моделі машинного навчання. *TPOT* оптимізує конвеєри, зокрема низку препроцесорів функцій і моделей *ML*, з метою вибору найбільш відповідних і ефективніших конвеєрів у контрольованій задачі класифікації. Побудова з них деревоподібних конвеєрів — дерев *GP* — відбувається за допомоги об'єднання операторів машинного навчання, що їх використовують як примітиви генетичного програмування. Кожен оператор конвеєра *ML* (примітив *GP*) у ньому відповідає алгоритму машинного навчання. Усі реалізації алгоритмів машинного навчання взято з *scikit-learn* (крім *XGBoost*).

У генетичному програмуванні кожне рішення (або кандидат) подано як дерево, яке складається з вузлів і листя. Кожен внутрішній вузол називається примітивом, а кожен листок — терміналом. Примітив розглядають як функцію, термінал — як аргумент або константу. Примітиви та термінали визначають залежно від розв'язуваної проблеми. У разі *TPOT* примітиви, алгоритми *ML* або методи та термінали попереднього оброблення є гіперпараметрами. Кореневий вузол дерева — обов'язково алгоритм *ML*, який використовують як остаточну оцінку. Інші вузли діють як оператори, що беруть дані на вхід і надають на виході перетворені дані. Дерева можуть складатися з різних гілок, де дві гілки об'єднуються через спеціальний примітив, поданий *TPOT*. Злиття об'єднує висновок із попередніх вузлів.

Отже, шлях вирішення проблеми оптимізації конвеєра машинного навчання — оператори машинного навчання, які використовують як примітиви генетичного програмування, що ґрунтуються на дереві конвеєрів, використовують для об'єднання примітивів в робочі навчальні конвеєри машин і алгоритм *GP*, який використовують для модифікації деревоподібних конвеєрів.

Під час роботи з великими наборами даних *TPOT* є дуже повільним, тому що розмір навчальної вибірки і кількість оцінюваних

[8] *Scikit-feature*: репозиторій вибору об'єктів, зокрема функцій, з відкритим вихідним кодом на *Python*. Ґрунтується на пакеті машинного навчання *scikit-learn* і двох наукових обчислювальних пакетах *Numpy* і *Scipy*. *Scikit-feature* містить приблизно 40 популярних алгоритмів вибору функцій, зокрема традиційні алгоритми та деякі алгоритми структурних і потокових функцій — *https://www.kdnuggets.com/2016/03/scikit-feature-open-source-feature-selection-python.html*

кандидатів є суттєвими. Еволюційні алгоритми зазвичай повільно сходяться, що робить *TPOT* нездатним масштабуватися до великих наборів даних. Відома робота [26], в якій для пришвидшення дослідження простору пошуку та поліпшення продуктивності за великих наборів даних застосовують алгоритм *Successive Halving* послідовного поділу навпіл, використовують у задачах *Multi-Armed Bandit*. При цьому *TPOT-SH* — швидше рішення *AutoML* — скорочує час, якого потребують еволюційні алгоритми, щоб знайти конвеєри *ML* для великих наборів даних.

Сфера застосовності AutoML

З викладеного випливає, що *AutoML* — це потреба автоматизації наскрізного процесу застосування машинного навчання до реальних проблем. Крім складності автоматизації багатьох завдань з оброблення даних, суть *AutoML* полягає в допомозі спеціалістам з даних та звільненні їх від тягаря повторюваних і менш вимогливих завдань, щоб вони могли витрачати свій час на складніші, творчі та важкі для автоматизації завдання. Фахівці *data scientists* через брак *AutoML* мають виконати налаштування алгоритму та оптимізацію гіперпараметрів, щоб максимізувати прогнозовану продуктивність своєї остаточної моделі машинного навчання. Оскільки багато із цих кроків часто виходять за рамки можливостей *Citizen Data Scientist* — експертів у ПрО, *AutoML* для дедалі більших потреб застосування *ML* було запропоновано як рішення на основі штучного інтелекту. Нині системи *AutoML* можуть швидко створювати прогнозні моделі, що забезпечують оптимальну продуктивність. Однак їх охоплення, як і раніше, є вузьким, а їхній справжній потенціал досі ще не використовується. Є три обмеження наявних систем *AutoML*: складні типи даних (1), знання предметної області (2) і самонавчання та посилене навчання (3) [27].

На рис. 3 показано типовий командний робочий процес фахівців з даних *TDSP* (*Team Data Science Process*) — це гнучка ітеративна методологія оброблення даних, яка дає змогу ефективно надавати рішення прогнозної аналітики та інтелектуальних програм [28]. Вона акцентує обмеженість сфер, у яких нині використовується *AutoML* [27].

1. *Складні типи даних.* Інструменти *AutoML* можна значною мірою класифікувати за сценарієм застосування або форматом навчальних даних. Більшість із них було спроєктовано для роботи з найпоширенішим і визнаним типом даних, (згідно з опитуванням *Kaggle*[9])

[9] *Kaggle* — головний організатор конкурсів з дослідження даних, а також соціальна мережа спеціалістів з обробки даних і машинного навчання. *KDnuggets* (*nugget* — самородок) — провідний сайт з науки про дані, машинне навчання, штучний інтелект й аналітику було засновано у квітні 2010 р. Григорієм П'ятецьким-Шапіро. *KD* (*Knowledge Discovery*) — означає «Знаходження (відкриття) знань» — *https://www.kdnuggets.com/about/index.htm*

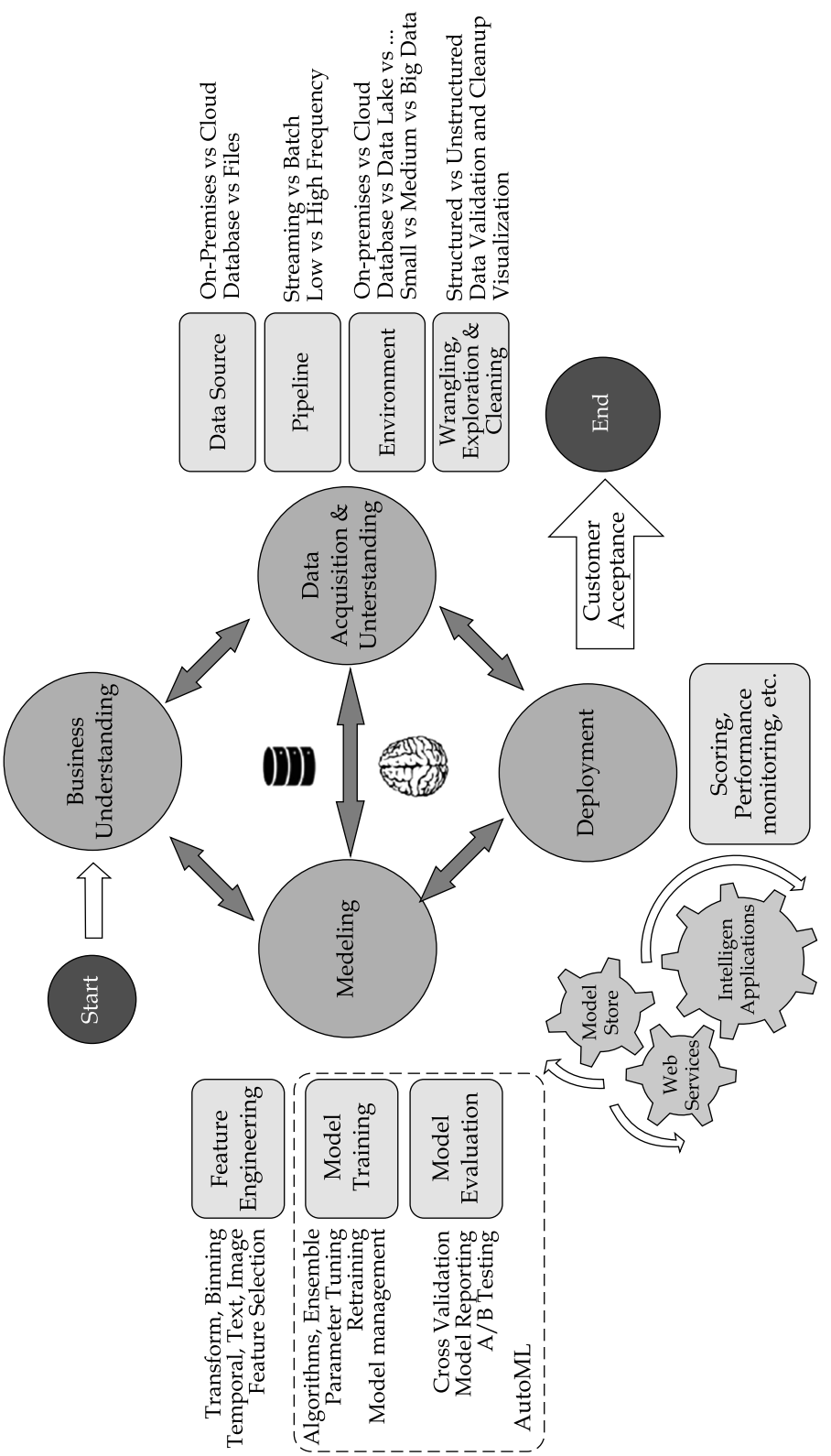

Рис. 3. Позиція *AutoML* у робочому процесі за даними *TDSP* та основні функційні вузли цього процесу

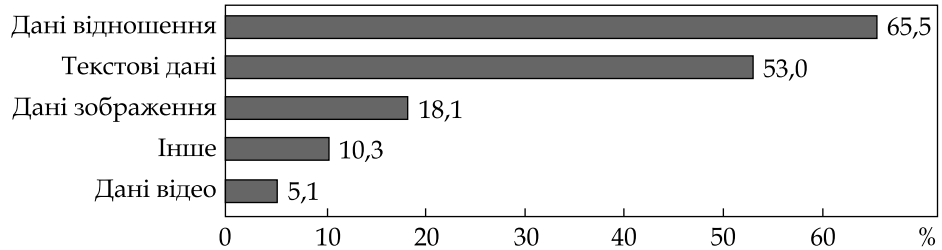

Рис. 4. Найпоширеніші типи даних систем *AutoML*

ML and Data Science Survey 2017 року [27], реляційні структуровані табличні дані — *IID* (таблиці з рядками та стовпцями), подані числами або категоріальним змінними (номінальні дані, наприклад, стать, колір шкіри, і порядкові дані, наприклад, рівень освіти). Іншим традиційним типом даних, переважно у фінансовій та роздрібній торгівлі, є дані з тимчасової залежністю. Флагманський продукт *H2O Driverless AI*[10] компанії *H2O.ai*[11] містить інструменти для автоматичної побудови прогнозних моделей часових рядів. Його, як і його попередника *H2O AutoML*, зорієнтовано на табличні дані *IID*, він підтримує дані часових рядів і необроблений текст. Деякі системи *AutoML* також було розширено для оброблення неструктурованих даних, а саме тексту та зображень (наприклад, *Google Cloud AutoML Natural Language, Google Cloud AutoML Vision, AutoKeras, DataRobot*), оскільки цей тип даних є вельми поширеним (рис. 4).

У разі текстових даних може бути застосовано інструмент загального призначення, такий як *H2O Driverless AI*, або спеціально розроблений для тексту, такий як *Google Cloud Natural Language*. Для вирішення з відкритим вихідним кодом, застосовують алгоритм *H2O Word2Vec* (або будь-який інший інструмент оброблення тексту з відкритим вихідним кодом), щоб перетворити текст на числовий формат, який може бути застосовано *H2O AutoML* (безпосередня підтримка текстових даних у *H2O AutoML* — його дорожня карта).

Мережеві та веб-дані, з іншого боку, є двома складнішими типами даних. Ці типи даних й досі не є предметом розгляду *AutoML*, що обмежує тип проблем, які розв'язують за допомоги *AutoML*. Однак, імовірно найближчим часом, автоматичні засоби оброблення цих типів даних відображатимуть міру розвитку інструментарію систем *AutoML*.

2. *Знання предметної області.* Системи *AutoML* вважають синонімом вибору моделі та налаштування гіперпараметра, і є лише невеликою частиною пазла у видобуванні знань у базах даних (*the knowledge*

[10] *H2O Driverless AI*, *https://www.h2o.ai/products/h2o-driverless-ai/*

[11] *H2O*.ai — компанія-розробник програмного забезпечення з Силіконової долини, яка вивела на ринок нові технології для просування *AI*, є творцем *H2O* — провідної платформи з відкритим вихідним кодом для дослідження даних і машинного навчання. Рішення *H2O* підтримує швидкі алгоритми *ML*, розподілені в оперативній пам'яті, і забезпечує самонавчання та посилене навчання.

discovery in databases, KDD) підприємства. Ці два етапи найлегше автоматизувати, враховуючи об'єктивність і послідовність їхніх кроків у завданнях *ML*-навчання. Проте, один із ключових компонентів для створення адекватних моделей *ML* часто ігнорувався в системах *AutoML* — це розроблення функцій. Проєктування функцій (це більше, ніж наука/знання), імовірно, це можливість, яка пропонує найблагодатніший ґрунт для розквіту людської творчості. Створення функцій фахівцями, кожен із яких намагається моделювати з метою розкриття значущих аспектів процесу, потребує уяви, творчості та досвіду в предметній області. Отже, продуктивність може варіюватися, якщо розроблення функцій виконують різнопрофільні фахівці. Розроблення функцій фахівцем також залежить від проблем і типу об'єктів, які можна відтворити, часто обмежене вхідним набором даних. Як наслідок, це один із важких, тривалих і трудомістких етапів будь-якого проєкту *data science*, поряд із очищенням і попереднім обробленням даних. Проте, достатньо хороші моделі може бути побудовано за допомоги прийняття загальнішої структури для створення ознак, що не залежать від набору даних (наприклад, *Deep Feature Synthesis* (*DFS*) — синтез посилених особливостей) [27].

Однак деякі платформи *AutoML* пропонують автоматичне проєктування функцій (наприклад, *DataRobot, H2O Driverless AI*). Проте жодна з них не може автоматично долучати знання предметної області до процесу *ML*, і це залишається виняткова людською функцією. Цікавим прикладом розв'язання цього питання є підхід, реалізований у *KNIME* [29, 30], так звана *Guided Analytics* — керована аналітика для автоматизації машинного навчання. Завдання її полягає в тому, щоб дозволити фахівцям, які працюють з даними, створювати інтерактивні системи, інтерактивно допомагаючи бізнес-аналітикам у їхньому прагненні знайти нове розуміння даних та передбачити майбутні результати.

Попри все це, заслуговують на увагу спроби автоматизувати складну та трудомістку задачу розроблення функцій, де секретним компонентом отримання високоякісних моделей у багатьох реальних задачах як і раніше залишається знання предметної області. Майбутній розвиток *AutoML* має бути зосереджений на створенні складніших методів для долучення певних знань в автоматично створювані функції. В ідеалі має бути розроблено складніші методи для долучення специфічних для предметної області знань в автоматично створювані функції із застосуванням закономірностей і залученням міждисциплінарної команди до розроблення продуктів *AutoML*. Системи *AutoML* також мають пропонувати можливість комбінування функцій, що генеруються автоматично, з функціями, створеними фахівцями, це свідчило б про їхню гнучкість [27].

3. *Самонавчання* (*UL*) *та посилене навчання* (*RL*). Глибокі нейронні мережі (*DNN*) демонструють дуже високу продуктивність під час

вирішення багатьох завдань машинного навчання, але вони дуже чутливі до налаштування своїх гіперпараметрів. При навчанні нейронної мережі необхідно налаштувати множину гіперпараметрів (швидкість навчання, розмір пакета, швидкість відсіву тощо), проте одним із найзначніших чинників, що впливають на продуктивність моделі, є мережева архітектура. В області завдань класифікації зображень (див. рис. 4) найчастіше використовують метод *AutoML*, який називають «пошуком нейронної архітектури» (*neural architecture search, NAS*). Є кілька інструментів із відкритим вихідним кодом для пошуку нейронної архітектури, зокрема реалізація ефективного пошуку нейронної архітектури (*efficient neural architecture search, ENAS*) — наприклад, *TensorFlow* і *Auto-Keras*, яка виконує ефективний *NAS* із застосуванням байєсівської оптимізації гіперпараметрів. *SaaS Google Cloud* пропонує *AutoML Cloud Vision* і *Cloud Natural Language*. Вони виконують пошук *NAS* для класифікації зображень і тексту відповідно. Цим інструментам потрібні необроблені графічні або текстові дані безпосередньо, і тому вони не надаються для типових числових чи категоріальних табличних даних. Двома областями, в яких *Deep Neural Networks* поліпшили продуктивність порівняно з традиційними методами машинного навчання, є класифікація зображень і оброблення природної мови (*natural language processing, NLP*) [27, 31].

Сучасний стан візуального розпізнавання об'єктів характеризується застосуванням згорткових нейронних мереж (*CNN*). Їх ємністю можна керувати, варіюючи глибину та ширину мережі, у порівнянні зі стандартними нейронними мережами прямого поширення з шарами однакового розміру. Також вони роблять задовільні припущення про природу зображень (а саме, про стаціонарність статистики та локальність піксельних залежностей) [32]. *CNN* мають набагато менше зв'язків і параметрів, тому їх легше навчати, а їхні теоретично найкращі результати продуктивності, ймовірно, є не гіршими. Кількість гіперпараметрів, що налаштовуються, зменшується. У [33] наведено, як приклад, гіперпараметри діапазону архітектур, зокрема у повністю пов'язаних мережах (*Fully Connected Networks*), великих згорткових (*Large Convolutional*) і маленьких згорткових (*Small Convolutional*) нейронних мережах *CNN*. Простір гіперпараметрів змодельовано за зразком архітектури в [32].

Попри привабливі якості *CNN* та відносну ефективність локальної архітектури, застосування їх у великих масштабах до зображень із високою роздільною здатністю, є занадто витратним. Сучасні графічні процесори у поєднанні з високо-оптимізованою реалізацією 2*D*-згортки є достатньо потужними, щоб полегшити навчання дуже великих *CNN*, а набори даних, такі як *ImageNet*, містять достатньо розмічених прикладів для навчання таких моделей без серйозного перенавчання. Ідеться про велику глибоку згорткову нейронну мережу, яка має 60 мільйонів параметрів і 500 000 нейронів, складається

з п'яти згорткових шарів, за деякими з яких ідуть шари максимального пулу, і двох глобально пов'язаних шарів із фінальним *softmax* із 1000 шляхів. Для прискорення навчання застосовувалися ненасичені нейрони та ефективна реалізація згорткових мереж на графічному процесорі. Щоб зменшити перенавчання у глобально пов'язаних шарах, застосували оригінальний метод регуляризації [32].

Традиційний *AutoML,* концентруючись на контрольованих задачах машинного навчання, які потребують маркованих, розмічених даних (*labelled data*) як вхідних, не передбачає складніші випадки навчання: самонавчання та посиленого *RL*-навчання. Навчання без учителя спрямовано на відкриття закономірностей з урахуванням даних, коли вірогідна інформація є недоступною. Немає чіткої міри успіху, яку можна застосовувати для оцінювання якості результатів самонавчання, оскільки немає строго обґрунтованого факту (немає точки відліку), з яким було би можливо порівнювати (наприклад, чи є невелике зображення на рентгенограмі пухлиною тощо). В результаті складніше судити про ефективність різних підходів, оскільки немає прямого способу порівняти їх. Ця суб'єктивність у визначенні і важлива роль експертних знань під час процесу є двома ймовірними причинами, з яких наявні системи *AutoML* не охоплюють цей підхід. Проте, з огляду на те, що у світі більшість даних не мають маркування, системи *AutoML* стали б іще кориснішими, якби їх застосування було розширено внаслідок автоматичного коригування машинних моделей. Посилене навчання, коли програмні агенти навчаються виконувати завдання методом спроб та помилок, отримують зворотний зв'язок від власних дій. Якщо дія є кроком до досягнення мети, то агент отримує винагороду. Інакше це карається. Таким чином, агент навчається на своїх помилках і вдосконалюється з досвідом. Подібно до навчання з учителем, при посиленому навчанні існує певна міра успіху, яка робить це завдання автоматизації *ML* реальним. Однак не було запропоновано жодної системи *AutoML* для автоматизації процесу самонавчання [27].

Подальший розвиток штучного інтелекту та *AutoML*. Тенденції

Наразі штучний інтелект застосовують у більшості областей знань. Він швидко розвивається та поширюється на кілька нових сфер, таких як мультимодальні застосунки та мобільні пристрої, що призводить до зростання технологічного розвитку небаченими раніше темпами. До 2021 року турбота про прозору й гідну довіру до *AI* призвела до того, що концепція відповідального та зрозумілого *AI* почала набувати дедалі більшого значення, допомагаючи зрозуміти, як працює *AI*. Ще одна сфера, в якій спостерігається сильне зростання — це гіперавтоматизація з використанням множини взаємозалежних тех-

нологій, таких як штучний інтелект та машинне навчання. Еволюція інтелектуальної автоматизації процесів та її впровадження забезпечить компаніям більшу гнучкість, принісши більше користі, ніж просто економія часу, підвищивши операційну ефективність і допомагаючи спростити процеси. Останнім часом напрямок розвитку *AI* полягає в прагненні покращити інтерпретованість моделей. Отже, дослідження дедалі більше рухатимуться вбік гібридних моделей між символічним *AI* з високим рівнем розуміння для людей та реальними статистичними глибокими нейронними мережами — рішення, яке забезпечить довіру учасників до надійного запровадження відповідальних систем [34].

В умовах певного успіху сучасних технологій і швидкого зростання досліджень та застосунків у галузі штучного інтелекту виникають і безпрецедентні можливості, і обґрунтовані побоювання з приводу його потенційного неправильного використання. Цю заяву зробила відомий вчений *Isabelle Guyon*[12] у рамках проєкту *HUMANIA*[13] [35]. Це занепокоєння є спільним. У роботі [36] авторами «вноситься певна помірність в уявлення, що виникло останніми роками, що проблему розпізнавання образів, яку традиційно вважають одним з розділів штучного інтелекту, цілковито вичерпано проблемою машинного навчання. Наука про штучний інтелект схильна зараз до серйозного випробування внаслідок надміру захопленої реклами наукових досягнень, адресованої переважно за межі наукової спільноти. Не можна вирішувати будь-яке складне завдання без будь-яких інтелектуальних зусиль щодо його дослідження» [3]. Два погляди на одну проблему, спільна занепокоєність — не ввести в оману, не нашкодити собі та суспільству — але шляхи її подолання є різними. Висновок другої роботи не потребує пояснень, а у першій він формулюється так: нагальна потреба зробити *AI* доступнішим для використання ширшим шаром населення. У контексті опублікованого документа «*HUMANIA. Artificial Intelligence for All*» [35] висловлено прагнення допомогти розв'язати це питання. Це має стати важливим фактором економічного зростання та сприяти зміцненню демократизації

[12] *Isabelle Guyon* — Директор, науковий співробітник *Google Research* (з жовтня 2022 р.). Завідувач кафедри штучного інтелекту (*PR EX1*) та дослідник *INRIA*, машинного навчання та оптимізації (команда *TAU*), Міждисциплінарна лабораторія цифрових наук (*LISN*) Університету Париж-Саклі, Франція (з 2015 року — професор). Співголова програми *NeurIPS* 2016 та генеральний голова *NeurIPS* 2017; потім член правління *NeurIPS*. З 2003 року організатор змагань з машинного навчання. Використання завдань як напряму досліджень у таких галузях, як причиново-наслідковий зв'язок, комп'ютерний зір, автоматичне машинне навчання та фізика високих енергій.

[13] This is the *ANR-19-CHIA*-0022 *project, a* 4*-year chair for the democratization of AI, started in September* 2020. *Publications on HAL. It is funded by L'agence Nationale de la Recherche* (*ANR*). По суті, це короткий виклад проєкту *ANR-19-CHIA*-0022, його мета, концепція, основні напрями наукових досліджень.

штучного інтелекту. Значення *AutoML* у сфері штучного інтелекту є неоціненним. Він демократизував процес створення та впровадження моделей машинного навчання, зробивши його доступним для ширшої аудиторії з різним рівнем знань у галузі машинного навчання. Автоматизуючи складні та трудомісткі завдання, *AutoML* значно знижує вхідний бар'єр для непрофесіоналів, які бажають використовувати можливості *AI* у своїх застосунках [37].

Дослідження [35] спрямовано на зниження потреби у фаховому досвіді під час реалізації алгоритмів розпізнавання образів та моделювання, включно з посиленим навчанням, у різних галузях застосування (медицині, інженерії, соціальних науках, енергомережах тощо) із застосуванням кількох модальностей (зображень, відео, тексту, часових рядів тощо). Цьому сприятиме організація наукових конкурсів проєктів (або змагань) з автоматизованого машинного навчання. Кульмінацією цих зусиль стане конкурс *AutoRL* (посиленого автоматичного навчання — *Automated Reinforcement Learning*). Проєкти готуватимуть спільноту до складніших та різноманітніших умов, постійно зменшуючи необхідність втручання людини у процес моделювання. «...Залучаючи наукове співтовариство як ціле у розв'язання складних завдань, ми ефективно помножимо на важливий чинник наше державне фінансування для вирішення таких складних задач *AutoML...*» [35].

AutoML змінив підхід до штучного інтелекту, демократизуючи його розроблення та прискорюючи темпи інновацій. Його еволюція, значення, реальне застосування підкреслюють ключову роль *AutoML* у формуванні майбутнього *ML* та *AI*. Цей підхід ґрунтується на низці фундаментальних робіт [35], в одній із яких запропоновано розробити систематичну й уніфіковану методологію для організації та використання наукових завдань у викладанні та дослідженнях у галузі *ML*, а саме штучного інтелекту, керованому даними (*data-driven Artificial Intelligence*). На сьогодні завдання стають дедалі популярнішими як засіб просування нових досягнень завдяки залученню вчених різного віку і в академічних колах, і за їхніми межами. Це може бути формою *citizen science*[14] громадянської науки. Є надія, що такий внесок спільноти в науку посприяє відтворенню досліджень та демократизації штучного інтелекту. Проте, є небезпека, що коли дані (або показники, що використовуються для визначення завдань), містять упередженість або артефакти, результат таких зусиль може бути в кращому випадку марним, а в гіршому — шкідливим для репутації громадянської науки [35].

[14] Експерт в ПрО у *Gartner* «*citizen data scientists*» — це дослівно цивільно-громадянські спеціалісти з даних. *Gartner* визначає *Citizen Data Scientist* як «особу, яка створює або генерує моделі, що використовують прогнозну або рекомендаційну аналітику, але чия основна робоча функція перебуває поза цариною статистики та аналітики» — 20 квітня 2018 р [1].

Робота має створити формальні основи для організації викликів і надання спільноті корисних інструкцій. У поєднанні з інструментами викликів, які буде розроблено, це обіцяє стати корисним внеском до здобутків наукової спільноти. Теоретичні фундаментальні внески мають спиратися на експериментальний дизайн і теорію ігор, а також на практичні емпіричні результати, що будуть отримані в результаті аналізу тестування альтернативних протоколів виклику в реальних змаганнях [35].

Орієнтоване на дані автоматизоване самонавчання (*DUL*)

Попри нещодавні успіхи глибокого навчання у застосунках [38], проєктування надійних глибоких нейронних мереж залишається складним завданням і вимагає практичного досвіду та знань, а головне, великої кількості даних [39–41]. В останні кілька років багато зусиль було спрямовано на вибір архітектур і гіперпараметрів, а також на навчання складних глибоких мереж, що відбувалося через множину циклів проб і помилок з використанням багатьох евристик. Пришвидшений попит на вирішення завдань глибинного навчання породжує потребу вдосконалити автоматизацію проєктування цих рішень [42]. Зазвичай, грубе оброблення завдань шляхом подачі до нейронних мереж ще більших наборів даних є привабливим рішенням, чому сприяють успіхи, досягнуті великими компаніями, такими як *Google*[15]. Отже, стає дедалі очевиднішим, що необхідний зсув парадигми в машинному навчанні, а не зосередження уваги на модельно-орієнтованій автоматизації машинного навчання: наприклад пошук нейронної архітектури та оптимізація гіперпараметрів, а також вирішення проблеми отримання якісніших даних. Однак збирання, оброблення, очищення, усунення спотворень та попереднє оброблення даних [1] можуть фактично стати вузьким місцем *AutoML* з інтенсивним застосуванням людської праці. Робота спрямована на вирішення проблеми ефективнішого використання (в автоматичному режимі) вже зібраних даних для отримання значних якісних та кількісних переваг машинного навчання, орієнтованого на дані [35], оскільки глибинні нейронні мережі, як відомо, прагнуть даних.

У зв'язку з цим було визначено кілька напрямів досліджень під алгоритмічним, теоретичним та практичним кутом зору. Зокрема

[15] Директор *Google* з досліджень штучного інтелекту *Peter Norvig* заявив: «У нас немає алгоритмів краще, ніж у когось іншого» і ще «у нас просто більше даних». Однак після скандалів, що викривають необачні рішення, що приймаються машинами, що навчаються, той же *Peter Norvig* переглянув свою думку і сказав: «Більше даних краще, ніж розумні алгоритми, але якісніші дані краще, ніж більше даних».

пропонується оцінити можливість об'єднання даних завдяки включенню кількох наборів даних з різних областей або з однієї області. Це уможливить входження у проблему машинного навчання, орієнтованого на дані, та дасть змогу вирішувати фундаментальні питання, такі як подолання різних типів помилок (випадкових, епістемічних, алеаторичних[16], систематичних тощо) у даних в автоматизованому режимі, що виникають через поганий збір та підготовку даних для аналізу. Окрім того, з метою обґрунтування теоретичних методів фільтрації, повторного балансування або корекції даних, необхідно провести аналіз джерел появи помилок/шуму/упередженості[17] в даних, та здійснити модифікацію поточних концепцій випадкових та епістемічних помилок через відсутність навчальних даних (зовсім відсутніх, або вичерпаних у конкретних областях простору даних), алеаторних помилок (помилка між прогнозованим та фактичним значенням) через внутрішні обмеження даних (наприклад, погане подання даних або погана попередня обробка; погане внесення пропущених значень), систематичних помилок тощо.

Зважаючи на результати окремих праць, є можливість імпортувати методи підвищення якості у глибоке навчання та об'єднати їх зі збільшенням даних [35, 43]. Однак необхідно проаналізувати покращення меж узагальнення[18], внаслідок використання інформації, отриманої за допомоги збільшення даних або застосування інших алгоритмів, орієнтованих на дані (наприклад, використання симетрії в даних, а також двоїстості між вибором моделі та вибором даних), з точки зору залежних від даних меж продуктивності [44], які сприяють поширеному в машинному навчанні спрощеному припущенню, що навчальні та перевірочні або тестові набори мають дотримуватися того ж розподілу.

Під кутом зору практики такі напрями можуть охоплювати: поєднання відбору даних і вибору моделі, вивчення питань, що стосуються упередженості в даних, зрозумілість модельних рішень щодо вибору репрезентативних вибірок даних, *treating multi-view* або *multimodal data* та прилаштування методів до суперкомп'ютерів, щоб надати вченим-прикладникам ефективні комплексні алгоритми навчання, орієнтовані на дані.

Ми очікуємо, що цей напрям досліджень матиме вплив у багатьох сферах застосування, де глибоке навчання довело свою ефективність.

[16] Системні науки та інформатика: збірник доповідей II науково-практичної конференції «Системні науки та інформатика», 4–8 грудня 2023 року, Київ. К., НН ІПСА КПІ ім. Ігоря Сікорського, 2023. 416 с. *http://mmsa.kpi.ua/sites/default/files/systemni_nauky_ta_informatyka_2023.pd*f

[17] Загалом, це схильність сприймати інформацію, що відповідає переконанням дослідників, та ігнорувати протилежну.

[18] Помилка узагальнення оцінює якість машинної моделі, тобто, наскільки добре модель узагальнює дані, яких раніше не було.

Це дасть змогу використовувати набори даних обмеженого розміру або низької якості для ефективного навчання таких моделей або ж об'єднувати дані з багатьох джерел, які збираються в різних умовах і потенційно є погано відкаліброваними. Враховуючи поточний дисбаланс зусиль дослідницької спільноти між підходами, орієнтованими на моделі та дані, ми сприятимемо зміні парадигм і, сподіваємося, зробимо плідний внесок у цій сфері, яка перебуває зараз у зародковому стані [35].

У сфері машинного навчання, зокрема самонавчання та трансферне навчання

Сучасна тенденція у сфері штучного інтелекту значною мірою покладається на системи, здатні навчатися на прикладах, таких як моделі глибокого навчання (*DL*), які є сучасним втіленням штучних нейронних мереж. Попри те, що за останні роки на ринок вийшли численні програми (зокрема, безпілотні автомобілі, автоматизовані помічники, служби бронювання та чат-боти, вдосконалення пошукових систем, рекомендації та реклама, а також застосунки для охорони здоров'я тощо), моделі *DL* все ще важко розгорнути в нових програмах. Зокрема, потрібна величезна кількість навчальних прикладів, години навчання *GPU* та висококваліфіковані інженери для ручного налаштування їхніх архітектур. Цей напрям дослідження сприятиме зменшенню бар'єрів входу до використання моделей *DL* для нових програм, що є кроком до «демократизації *AI*».

Дослідження [35] полягає в розробці нових підходів до трансферного навчання (*TL*), заснованих на модульних архітектурах *DL*. Трансферне навчання охоплює всі методи пришвидшення навчання завдяки використанню попередніх подібних завдань. Наприклад, використання попередньо навчених мереж, цілковито або частково (модульність), є ключовою тактикою *TL*.

У цьому контексті важливим є наступне. Під технічним кутом зору поточні обмеження попереднього навчання охоплюють: (T1) у багатьох доменах немає доступних попередньо навчених мереж через брак великих наборів даних у пов'язаних доменах; (T2) нові архітектури мереж, такі як «графові нейронні мережі» (*GNN*), не легко піддаються попередньому навчанню; (T3) крім простого перенавчання останнього рівня та тонкого налаштування внутрішніх рівнів, засоби повторного використання попередньо навчених мереж у нових контекстах розроблено недостатньо. Ці три проблеми ускладнюють дослідницькі можливості для ефективного використання попередньо набутих знань, симуляторів даних та/або доповнення даних, а також розробки нових алгоритмів і архітектур, які навчаються модульним способом для повторного використання. Наприклад, з точки зору фундаментальних досліджень, модульність і успадкування попере-

дньо підготовлених навчальних модулів у біологічно інспірованих системах навчання є гострою темою *AI* [35].

У цьому контексті пропонується дослідити новий підхід до перенесення навчання, який ми називаємо «Глибоке модульне навчання» [35]. Треба вирішувати проблему навчання великих штучних нейронних мереж, архітектура яких є модульною, та чиї модулі в кінцевому підсумку можна використовувати повторно. Можливим підходом до проблеми буде застосування багаторівневих алгоритмів оптимізації, спрямованих на оптимізацію всієї системи (досягнення мети вищого рівня) за умови, що модулі досягають мети нижчого рівня (повторне використання). Одна з наукових цілей полягатиме в тому, щоб поставити під сумнів гіпотезу про те, що модульність є важливою для систем навчання, оскільки вона прискорює навчання, роблячи можливою ефективну форму передачі навчання, центральну функційність *AI*. Декілька припущень можуть керувати цим дослідженням, зокрема такі:

(*P1*) Принцип економності або «бритва Оккама», втілений у сучасній теорії навчання як «регулярізація», яка стверджує, що «з двох теорій, рівноцінно потужних у відтворенні спостереження, слід віддавати перевагу простішій». Насправді модульні архітектури, що мають ідентичні підмодулі, мають менше настроюваних параметрів, і тому можуть вважатися менш складними, ніж, наприклад, повністю підключені усі модулі.

(*P*2) Гіпотеза вродженості: здатність розв'язувати завдання є комбінацією вроджених і набутих навичок. Чи властиво інтелектуальним системам більшою мірою покладатися на набуті навички, наприклад мову, а не успадковувати їх? Чи правда, що мову можна повністю вивчити «з нуля»?

(*P*3) Індукція, дедукція, концептуалізація та причиновість: чи ґрунтуються інтелектуальні системи навчання на модульності для концептуалізації, засвоєння мови та причиново-наслідкового висновку? Практичне застосування цієї структури на практиці є можливим у таких областях як біомедицина (наприклад, молекулярна токсичність або ефективність), екологія, економетрика, розпізнавання мови, обробка природної мови, обробка зображень або відео тощо [35].

Висновки

На початку нинішнього століття турбота про прозорі рішення та гідну довіру до штучного інтелекту стала причиною того, що концепція відповідального та зрозумілого *AI* почала набувати дедалі більшого значення, допомагаючи зрозуміти, як працює *AI*. Останнім часом напрям розвитку *AI* полягав у прагненні покращити інтерпретованість моделей. Тому дослідження дедалі інтенсивніше рухатимуться в бік гібридних моделей між символічним *AI* з високим рівнем розуміння,

та реальними статистичними глибинними нейронними мережами — рішення, яке забезпечить довіру учасників до надійного запровадження відповідальних систем.

Якісне вирішення зазначених у цьому огляді проблем роботи з даними з сучасним аналітичним інструментарієм їх дослідження із застосуванням множини взаємозалежних технологій, таких як штучний інтелект та машинне навчання, успішне проникнення в їхню суть на рівні отримання знань при високій автоматизації процесу як цілого.

Значення *AutoML* у сфері штучного інтелекту є неоціненним. Він демократизував процес створення та впровадження моделей машинного навчання, зробивши його доступним для ширшої аудиторії з різним рівнем знань у галузі машинного навчання. Автоматизуючи складні та трудомісткі завдання, *AutoML* значно знижує вхідний бар'єр для непрофесіоналів, які бажають використовувати можливості *AI* у своїх застосунках.

AutoML змінив підходи до штучного інтелекту. Його еволюція від традиційного *AutoML* до *AutoDL* та *AutoRL,* його значення, реальне застосування наголошують на ключовій ролі *AutoML* у формуванні майбутнього *AI* та *ML*. Однак надмірна залежність від автоматизації без фундаментального розуміння концепцій машинного навчання може призвести до неоптимального вибору моделювання та обмеженого розуміння даних.

І останнє. Не слід нехтувати застереженням [35, 36] щодо науки про штучний інтелект, яка стоїть зараз перед серйозним викликом внаслідок надміру захопленої реклами наукових досягнень, адресованої переважно за межі наукової спільноти. Неможна вирішувати будь-яке складне завдання без будь-яких інтелектуальних зусиль щодо його дослідження. Це стосується також автоматизації підбору функцій та моделі, що вимагають залучати знання та обмеження предметної галузі у процес їх розроблення, гарантуючи, що остаточна модель відповідатиме вимогам ПрО.

ЛІТЕРАТУРА

1. Oursatyev O. Data Research in Industrial Data Mining Projects in the Big Data Generation Era. *Control Systems and Computers*, 2023, Issue 3, 33–54. [In Ukrainian: Дослідження даних у промислових data-mining-проєктах в епоху генерації великих даних] https://doi.org/10.15407/csc.2023.03.033
2. Elliott T. Intelligence Called?! — Innovation Evangelism. URL: https://timoelliott.com/blog/2017/06/what-is-artificial-intelligence-called.html [Accessed: 11 June 2019].
3. Schlesinger M., Hlavac V. Ten Lectures on Statistical and Structural Pattern Recognition. *Computational Imaging and Vision*. Kluwer Academic Publishers, Dordrecht, Boston London, 2002, Vol. 24, 520 p. https://doi.org/10.1007/978-94-017-3217-8
4. GitHab. Awesome-AutoML-Papers. What is AutoML? URL: https://github.com/hibayesian/awesome-automl-papers [Accessed: 3 March 2024]

5. Guyon I., et al. Design of the 2015 ChaLearn AutoML Challenge. URL: http://haralick.org/ML/automl_ijcnn15.pdf [Accessed: 01 Apr. 2024]
6. Guyon I. et al., Analysis of the AutoML Challenge Series 2015–2018. URL: https://link.springer.com/chapter/10.1007/978-3-030-05318-5_10 [Accessed: 01 Apr. 2024]
7. Domhan T., Springenberg J.T., Hutter F. Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves. *24th International Conference on Artificial Intelligence IJCAI 2015*, Buenos Aires, Argentina, 3460–3468.
8. Bengio, Y. Gradient-Based Optimization of Hyperparameters. *Neural Computation,* 2000, Vol. 12 (8), 1889–1900. https://doi.org/10.1162/089976600300015187
9. Bergstra, J., Bengio, Y., Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research,* 2012, Vol. 13, 281–305. URL: http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf
10. Budjac R., Nikmon M., Schreiber P., Zahradníková B., Janáčová, D. Automated machine learning overview. *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, 2019, Vol. 27 (45), 107–112. https://doi.org/10.2478/rput-2019-0033
11. Snoek J., Larochelle H., Adams Ryan P. Practical Bayesian Optimization of Machine Learning Algorithms**.** *ArXiv*, 2012, 206.2944, 1–9. https://doi.org/10.48550/arXiv.1206.2944
12. Bengio Yo., Ducharme R., Vincent P., Janvin C. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 2003, Vol. 3, 1137–1155. URL: https://dl.acm.org/ doi/10.5555/944919.944966
13. Hutter F., et al. Sequential Model-Based Optimization for General Algorithm Configuration. *Learning and intelligent optimization: 5th international conference*, LION 5, Rome, Italy, 2011, 1–15. URL: https://ml.informatik.uni-freiburg.de/papers/11-LION5SMAC.pdf
14. Olson R., Moore J. TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning. *JMLR: Workshop and Conference Proceedings AutoML Workshop (ICML2016),* 2016, 66–74. URL: http://proceedings.mlr.press/v64/olson_tpot_2016.pdf
15. Li Liam. *Towards Efficient Automated Machine Learning*. 2020, 184 p. URL: https://www.ml.cmu.edu/research/phd-dissertation-pdfs/thesis_li_liam.pdf
16. Thornton C., Hutter F. et al. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. *19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD-2013), 2013,* 847–855. https://doi.org/ 10.1145/2487575.2487629
17. Thornton C., Hutter F. et al. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13), 2013, 847-855. URL: https://www.cs.ubc.ca/~hoos/Publ/ThoEtAl13.pdf [Accessed: 01 Apr. 2024]
18. Feurer M. et al. Practical Automated Machine Learningfor the AutoML Challenge 2018, *ICML 2018 AutoML Workshop*. URL: https://ml.informatik.uni-freiburg.de/papers/18-AUTOML-AutoChallenge.pdf
19. Bergstra J., et al. Algorithms for Hyper-Parameter Optimization. *Part of Advances in Neural Information Processing Systems (NIPS 2011), 2011, Vol. 24, 1-9*. URL: https://papers.nips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf
20. Sun L., et al. Automatic Neural Network Search Method for Open Set Recognition. *2019 IEEE Int. Conf. Image Process,* Beijing, China Fujitsu Laboratories Ltd., Kawasaki, Japan, 2019. https://doi.org/10.1109/ICIP.2019.8803605
21. Jamieson K., Talwalkar A. Non-stochastic Best Arm Identification and Hyperparameter Optimization, Feb 2015, 1–13. URL: https://arxiv.org/pdf/1502.07943.pdf

22. Li L., Jamieson K., Rostamizadeh A., Talwalkar A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research,* 2018б Vol. 18, 1-52. URL: https://arxiv.org/pdf/1603.06560.pdf
23. Li L., Jamieson K., Rostamizadeh A., Gonina E., et al. A System for Massively Parallel Hyperparameter Tuning. *3-rd MLSys Conference*, Austin, TX, USA, 2020, 1–17. URL: https://arxiv.org/pdf/1810.05934.pdf
24. Jin Y. et al. *Multi-Objective Machine Learning.* Berlin Heidelberg: Springer, 2006. Vol. 16., 657 p. URL: https://link.springer.com/content/pdf/ bfm%3A978-3-540-33019-6%2F1.pdf
25. Jin Y., Sendhoff B. Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies. *IEEE transactions on systems, man, and cybernetics – part C: applications and reviews,* 2008, Vol. 38 (3), 397–415. https://doi.org/10.1109/TSMCC.2008.919172
26. Parmentier L., Nicol O., Jourdan L., Kessaci M. TPOT-SH: A Faster Optimization Algorithm to Solve the AutoML Problem on Large Datasets. *IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, 471–478. https://doi.org/10.1109/ICTAI.2019.00072
27. Oliveira M. 3 Reasons Why AutoML Won't Replace Data Scientists Yet. 2019. URL: https://www.kdnuggets.com/2019/03/why-automl-wont-replace-data-scientists.html
28. Adopting a data science process framework. URL: https://microsoft.github.io/azureml-ops-accelerator/1-MLOpsFoundation/2-SkillsRolesAndResponsibilities/1-AdoptingDSProcess.html [Accessed 6 Jun. 2025]
29. Berthold M. Principles of Guided Analytics. 2018. URL: https://www.knime.com/blog/principles-of-guided-analytics
30. Tamagnini P., Schmid S., Dietz C. How to Automate Machine Learning. 2019. URL: https://www.knime.com/blog/how-to-automate-machine-learning
31. LeDell E. The different flavors of AutoML. 2018. URL: https://www.h2o.ai/blog/the-different-flavors-of-automl/
32. Krizhevsky A., Sutskever I., Hinton G. Imagenet classification with deep convolutional neural networks. *NIPS,* 2012, 1097–1105.
33. Domhan T., Springenberg J.T., Hutter F. Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves. *24th International Conference on Artificial Intelligence [IJCAI 2015]*, 2015, 3460–3468.
34. Piatetsky Gregory, KDnuggets on December 10, 2021 in 2022 Predictions. Main 2021 Developments and Key 2022 Trends in AI, Data Science, Machine Learning Technology – https://www.kdnuggets.com/2021/12/trends-ai-data-science-ml-technology.html
35. Guyon I. HUMANIA. Artificial Intelligence for All – https://guyon.chalearn.org/projects/humania
36. Gritsenko V.I., Schlesinger M.I. Interaction of pattern recognition machine thinking and learning problems. International Scientific Technical Journal «Problems of Control and Informatics», 3, 108-136. URL: http://jnas.nbuv.gov.ua/article/UJRN-0001259001 [In Russian].
37. AI Glossary. Automated Machine Learning Automl, December 24, 2023. URL: https://www.larksuite.com/en_us/topics/ai-glossary/automated-machine-learning-automl
38. LeCun Y., BengioY., Hinton G. Deep learning. *Nature*, 2015, Vol. 521 (7553), 436-444. https://doi.org/10.1038/nature14539
39. Alom M. Z. et al. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 2019, Vol. 8 (3), 292-357. https://doi.org/10.3390/electronics8030292
40. Deng J. et al. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

41. Sun C. et al. Revisiting unreasonable effectiveness of data in deep learning era. *IEEE international conference on computer vision*, 2017, 843–852. https://doi.org/10.1109/ICCV.2017.97
42. Assunçao F. et al. Evolving the topology of large scale deep neural networks. *European Conference on Genetic Programming,* Springer, 2018, 19–34. https://doi.org/10.1007/978-3-319-77553-1_2
43. Cubuk, Ekin D. et al. Autoaugment: Learning augmentation policies from data. *CoRR*, 2018, Vol. 1, 1-14. https://doi.org/10.48550/arXiv.1805.09501
44. Vapnik V. *Statistical Learning Theory.* Wiley-Interscience publication, Wiley, 1998, 736 p. URL: https://books.google.fr/books?id=GowoAQAAMAAJ [Accessed: 01 Apr. 2024]

Отримано 10.06.2024

O.A. Oursatyev, PhD (Engineering), Senior Research Associate,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
https://orcid.org/ 0009-0009-8323-0525
aleksei@irtc.org.ua

O.Ye. Volkov, PhD (Engsneering), Senior Researcher, Director,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
https://orcid.org/0000-0002-5418-6723
alexvolk@ukr.net

V.H. Tkalya, PhD Student,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
https://orcid.org/0009-0004-7870-3256
advv0207@gmail.com

AUTOMATED MACHINE LEARNING.
STATE AND PROSPECTS DEVELOPMENT

Introduction. The task of automated machine learning is considered as one of the tasks of artificial intelligence for the automation of the end-to-end process of designing machine learning pipelines. In this case, human presence should be significantly reduced or, preferably, completely excluded. The article provides an overview of information services for the automation of the end-to-end process of applying and improving the efficiency of machine learning methods — modern approaches, methods, technologies in this area, competing due to the development of intelligent means of information processing, and sometimes even surpassing machine learning experts.

The purpose of the paper is to familiarize specialists in need of data skills with tasks related to data processing, advanced mathematical apparatus and world-class tools for obtaining information and knowledge.

Methods. Machine learning methods are studied. machine learning is considered as an artificial intelligence-based solution for automating the end-to-end process of applying machine learning, i.e. designing machine learning pipelines — a sequence of steps that transform raw data into a machine model suitable for deployment in practical use. The direction of further development of artificial intelligence and automated machine learning and its development trends are considered.

Results. Automated machine learning is considered as an AI-based solution for the needs of automating the end-to-end process of applying machine learning to design machine learning pipelines. This is a sequence of steps that transform raw data into a machine model adopted for deployment in practical use. The direction

of further development of artificial intelligence and automated machine learning, as well as its development trends, are considered.

Conclusion. The conducted research has shown the importance of automated machine learning in the field of artificial intelligence. This approach has democratized the process of creating and implementing machine learning models, making it accessible to a wider audience with different levels of knowledge in the field of machine learning. By automating complex and time-consuming tasks, automated machine learning significantly reduces the entry barrier for non-professionals who want to use the capabilities of artificial intelligence in their applications.

***Keywords**: AutoML, Artificial Intelligence for All, data-driven Artificial Intelligence, Deep Learning, DL, Reinforcement Learning, RL. Transfer Learning, TL.*