УДК 004.8
DOI https://doi.org/10.15589/znp2025.1(499).17

AUTOMATED MACHINE LEARNING TECHNOLOGIES FOR NATURAL LANGUAGE PROCESSING SYSTEMS

ОГЛЯД ТЕХНОЛОГІЙ АВТОМАТИЧНОГО МАШИННОГО НАВЧАННЯ ДЛЯ СИСТЕМ З ОБРОБКИ ПРИРОДНОЇ МОВИ

Mykhailo Y. Vyshniak
mykhailo.vyshniak@nure.ua
ORCID: 0000-0002-3240-139X
Mykhailo Y. Pyrozhenko
mykhailo.pyrozhenko@nure.ua
ORCID: 0000-0003-0785-1273

М. Ю. Вишняк,
канд. техн. наук, доцент
М. Ю. Пироженко,
аспірант

Kharkiv National University of Radio Electronics, Kharkiv
Харківський національний університет радіоелектроніки, м. Харків

Abstract. Building high-quality artificial intelligence systems depends on the knowledge of machine learning experts and requires a large amount of work, which prevents the widespread use of these technologies in various fields of human activity. This paper discusses modern approaches and systems of automatic machine learning. Automatic machine learning tools simplify the process of data collection, feature development, model creation and evaluation, and therefore facilitate the implementation of artificial intelligence systems, including natural language processing systems. The article describes the peculiarities of applying machine learning technologies to various natural language processing tasks.
Purpose. Analyzing the state of development of the methodological framework for the development of artificial intelligence systems and automatic machine learning systems, in particular, for the construction of natural language processing systems.
Method. The study was conducted using a systematic approach, methods of abstraction, system analysis, comparison and synthesis.
Results. We have reviewed the stages of creating artificial intelligence systems and the approaches used to build systems using automatic machine learning technology. As a result of this work, an overview of automatic machine learning approaches and systems is presented.
Scientific novelty. We have solved an urgent scientific task, which is to review modern automatic machine learning systems and approaches.
Practical importance. It is the possibility of applying the scientific provisions and conclusions of the study for the development and implementation of artificial intelligence systems; in the process of teaching natural language processing disciplines in higher education institutions; in writing manuals on natural language processing; in conducting applied research.
Key words: automatic machine learning; natural language processing; text analytics; systems; methods; models.

Анотація. Побудова високоякісних систем штучного інтелекту залежить від знань експертів з машинного навчання та потребує великого обсягу роботи, що перешкоджає широкому застосуванню цих технологій в різних галузях діяльності людини. В цій роботі розглядаються сучасні підходи та системи автоматичного машинного навчання. Інструменти автоматичного машинного навчання спрощують процес збору даних, розробки функцій, створення та оцінки моделей, а тому полегшують впровадження систем штучного інтелекту, зокрема систем в галузі обробки природної мови. Також наводяться особливості застосування технологій машинного навчання щодо різних задач з обробки природньої мови.
Мета. Аналіз стану розвитку методологічної бази розробки систем штучного інтелекту та систем автоматичного машинного навчання, зокрема, для побудови систем обробки природної мови.
Методика. Дослідження проводилось з використанням системного підходу, методів абстрагування, системного аналізу, порівняння та синтезу.
Результати. Нами розглянуті етапи ствонення систем штучного інтелекту та підходи, що застосовуються при побудові систем з застосуванням технології автоматичного машинного навчання. В результаті проведеної роботи представлено огляд підходів та систем автоматичного машинного навчання.

Наукова новизна. Було вирішено актуальне наукове завдання, що полягає в огляді сучасних систем та підходів автоматичного машинного навчання.
Практична значимість. Полягає в можливості застосування наукових положень та висновків дослідження для розробки та впровадження систем штучного інтелекту; у процесі викладання дисциплін з обробки природних мов у вищих навчальних закладах; під час написання посібників з обробки природних мов; під час проведення прикладних досліджень.
Ключові слова: автоматичне машинне навчання; обробка природної мови; текстова аналітика; системи; методи; моделі.

ПОСТАНОВКА ЗАДАЧІ

Технології машинного навчання досягли значних успіхів для різних завдань, зокрема в галузі обробки природної мови. Розробка подібних систем значною мірою залежить від експертів з машинного навчання, які виконують наступні завдання: попередня обробка та очищення даних, вибір ознак, вибір моделей, оптимізація гіперпараметрів, розробка та постобробка моделей машинного навчання, аналіз отриманих результатів.

Автоматизоване машинне навчання (AutoML) є можливим рішенням для побудови систем, пов'язаних з глибоким навчанням, без допомоги людини. Сучасні підходи AutoML для різних завдань моделювання мови, особливо коли це стосується великих мовних моделей, досягли значних результатів за останні роки. Через велику кількість напрацювань останніх років виникає потреба аналізу підходів та систем AutoML.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Багато досліджень останніх років були зосереджені на покращенні методів машинного навчання. Створення систем потребувало створення великих навчальних наборів даних. З ростом обсягу цих даних з'явилася потреба у створені автоматизованих методів.

В [1] визначені проблеми розробки систем з застосуванням AutoML для великих мовних моделей. Була вказана проблема зростання вимог до апаратного забезпечення, що ставить питання про те, чи доцільно витрачати такі ресурси. Також була окремо висвітлена проблема неоднозначності у взаємодії системи з користувачем та ситуації коли створена інформація буде виглядати достовірною для неспеціалістів, хоча вона може не відповідати дійсності.

В [2] представлено Extreme AutoML та порівняно його з іншими підходами для різних задач машинного навчання. Extreme AutoML показав кращі результати за низкою ключових метрик для задач класифікації, регресійних задач та різних задач з галузі обробки природної мови.

В [3] також порівнюється продуктивність та ефективність деяких існуючих підходів AutoML. Також в роботі описуються проблеми та обговорюються тенденції та важливі майбутні напрямки досліджень.

В [4] представлено огляд розвитку окремих підходів машинного навчання, глибокого машинного навчання та автоматичного машинного навчання. В дослідженні показано, що деякі традиційні підходи все ще використовуються для класифікації та прогнозування. Методи глибокого навчання, включаючи CNN та RNN, досить поширені в завданнях класифікації зображень та обробці природної мови.

ВІДОКРЕМЛЕННЯ НЕВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ

Системи з обробки природної мови вимагають великої кількості анотованих даних. Набори даних не завжди доступні, особливо для спеціалізованих областей знань або мов, що негативно впливає на продуктивність таких систем. Разробка таких систем також потребує знань з машинного навчання. На сьогодні фактично відсутній огляд систем та підходів автоматичного машинного навчання, який враховував би досягнення, які відбулись у цій галузі за останні роки.

МЕТА ДОСЛІДЖЕННЯ

Аналіз стану розвитку методологічної бази розробки систем штучного інтелекту та систем автоматичного машинного навчання, зокрема, для побудови систем обробки природної мови.

МЕТОДИ, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Дослідження проводилось з використанням системного підходу, методів абстрагування, системного аналізу, порівняння та синтезу. Предмет дослідження – поточний стан розробки систем автоматичного машинного навчання. Об'єкт дослідження – моделі та методи автоматичного машинного навчання.

ОСНОВНИЙ МАТЕРІАЛ

Розробка AutoML системи складається з кількох процесів: підготовка даних, розробки функцій, створення та оцінка роботи моделі системи.

Процес підготовки даних можна представити в трьох аспектах: збір даних, очищення даних та розширення даних. Збір даних – це крок для створення нового набору даних. Процес очищення даних використовується для фільтрації зашумлених даних. Попередня обробка набору даних важлива, оскільки вхідні дані можуть мати багато розбіжностей. Цей етап часто включає видалення розділових знаків, зміну регістру

слів, видалення стоп-слів. Розширення даних важливо для підвищення продуктивності моделі.

Набори даних є критично важливими для навчання. Складно знайти потрібний набір даних для деяких мов або конкретних галузей діяльності. Наприклад, для завдань пов'язаних із медичною допомогою набори можуть бути недоступні через конфіденційність цих даних. Пошук веб-даних є простим способом збору набору даних. Проте результати пошуку можуть бути неправильно позначені та повинні бути відфільтровані. Загальним рішенням є тонке налаштування та фільтрації веб-даних.

Для вирішення цієї проблеми, розробниками використовується метод самомаркування на основі навчання [5]. Якщо показники достовірності міток дуже близькі або мітка з найвищим балом збігається з оригінальною міткою, то ці данні будуть використані як новий навчальний зразок.

Для деяких, конкретних завдань, неможливо перевірити та налаштувати модель на етапі дослідження. Імітація даних є одним із найпоширеніших методів генерації даних. Таким чином, практичний підхід до генерації даних полягає у використанні даних, які максимально відповідають реальним.

Зібрані дані неминуче мають шум який негативно вплине на навчання моделі. Для зменьшеня впливу шуму необхідно очистити отриманні дані.

Зусилля розробників з очищення даних поступово переходять від людської праці до автоматизації збору даних. Дослідниками запропоновано лише очистити невелику підмножину даних [6], однак це вимагає, щоб спеціаліст з обробки даних розробив операції очищення даних, які застосовуються до набору. Кінцева комбінаторна операція очищення даних може складатися з кількох конвеєрних операцій очищення [7].

Концепція гіперпараметричної оптимізації виникла в нейронних мережах, оскільки набори даних, як правило, стають занадто великими для ручного налаштування. Байєсівська оптимізація є одним з варіантів для налаштування гіперпараметрів [8]. Це імовірнісна модель, яка фіксує взаємозв'язок між налаштуваннями гіперпараметрів та їхньою продуктивністю. Ця модель використовується для оцінювання та вибору перспективних налаштувань.

Можна також використовувати техніку метанавчання для автоматизації процесу очищення даних.

Метанавчання – ще одна концепція, що набуває популярності останнім часом [9]. Кожна модель, навчена на наборі даних, робить свій внесок у розуміння даних. Ці результати, об'єднані разом, можуть створити систему знань, яку можна використовувати на подібних наборах даних. Метанавчання використовує такі атрибути, як розмір набору даних, кількість об'єктів та різні інші характеристики об'єктів разом з даними про продуктивність. Метанавчання використовується для пошуку хороших екземплярів моделей навчання на основі знань про попередні завдання [10].

Методи очищення даних застосовуються до фіксованого набору даних. Однак реальний світ щодня створює величезну кількість даних. Чистка даних в безперервному процесі є проблемою, яка потребує окремого вивчення.

Розширення даних [11] також можна розглядати як інструмент для збору даних, оскільки розширення даних дозволяє генерувати нові дані на основі існуючих. Для текстових даних, в [12] запропоновані два підходи для створення додаткових навчальних прикладів: викривлення даних та синтетична надвибірка. Викривлення даних генерує додаткові зразки, застосовуючи перетворення до простору даних. Синтетична надвибірка створює додаткові зразки в просторі ознак. Текстові дані можна доповнити вставкою синонімів або спочатку перекладом тексту на іноземну мову, а потім його перекладом на мову оригіналу.

В [13] представлена політика розширення даних, яка використовує шуми в RNN. Цей підхід добре працює для завдань мовного моделювання та машинного перекладу. В [14] запропонуван метод зворотного перекладу для розширення даних та покращення розуміння прочитаного.

З метою підвищення ефективності пошуку були запропоновані алгоритми з використанням різних стратегій, таких як градієнтний спуск [15], байєсовська оптимізація [8], гіперпараметр онлайн навчання [16], жадібний пошук [17], випадковий пошук [18], та метод без пошуку [19].

Створення ознак – це створення функцій, які допомагають алгоритмам машинного навчання навчатися швидше. У більшості випадків виділення ознак має на меті зменшити розмірність ознак шляхом застосування спеціальних функцій, у той час як побудова ознак використовується для розширення вихідного простору ознак, а метою вибору ознак є зменшення надмірності шляхом вибору більш важливих функцій.

Розробка функцій складається з трьох частин: вибір функцій, вилучення функцій та побудова функцій. Вибір функцій створює підмножину функцій на основі вихідного набору шляхом зменшення нерелевантних або зайвих функцій. Це, як правило, спрощує модель та покращуючи її продуктивність.

Стратегія пошуку для вибору ознак включає три типи методів: повний пошук [20], евристичний пошук [21] та випадковий пошук [22].

Методи оцінки підмножини можна розділити на три групи: методи фільтрації, оболонки та вбудовані методи. Методи фільтрації оцінюють кожну ознаку відповідно до її розбіжності або кореляції, а потім вибирають функції відповідно до порогу. Зазвичай використовуваними критеріями оцінки для кожної ознаки є дисперсія, коефіцієнт кореляції, критерій хі-квадрат та взаємна інформація. Методи оболонки

класифікують набір вибірки з вибраною підмножиною ознак, після чого точність класифікації використовується як критерій для вимірювання якості підмножини ознак. Вбудовані методи – це методи у яких вибір змінних виконується як частина процедури навчання.

Створення функцій – це процес, який створює нові функції з базового простору функцій або необроблених даних для підвищення надійності та можливості узагальнення моделі. Цей процес сильно залежить від досвіду людини. Одними із найбільш часто використовуваних методів є перетворення попередньої обробки, наприклад стандартизація, нормалізація або дискретизація ознак. Операції перетворення для різних типів об'єктів можуть відрізнятися.

Вилучення ознак – це процес зменшення розмірності, який виконується за допомогою деяких функцій відображення. Вилучення ознак є функцією відображення, яку можна реалізувати різними способами.

Найбільш поширеними підходами для вилучення ознак є аналіз головних компонент, аналіз незалежних компонентів, нелінійне зменшення розмірності та лінійний дискримінантний аналіз[23].

Створення моделі можна розділити на проектування (простір пошуку) та оптимізацію. Типи моделей можна загалом розділити на дві категорії: традиційні моделі, такі як машина опорних векторів [24] або k-метод найближчих сусідів [25] та глибока нейронна мережа [26]. Методи оптимізації архітектури визначають, як керувати пошуком. Вони класифікуються на оптимізацію за гіперпараметрами, що передбачає оптимізацію параметрів, пов'язані з навчанням, та оптимізацію архітектури, що передбачає оптимізацію моделі.

Після того, як модель створена, її продуктивність необхідно оцінити. Найпростіший підхід до цього полягає в тому, щоб навчити модель сходитися на навчальному наборі, а потім оцінити продуктивність моделі на перевірочному наборі. Однак, цей спосіб ресурсомісткий.

Очікується, що гарний простір пошуку виключатиме упередженість людини та буде достатньо гнучким, щоб охоплювати більш широкий спектр архітектур моделей.

Простір цілісно структурованих нейронних мереж [27] є одним із найбільш інтуїтивно зрозумілих та простих просторів пошуку. Проте існують кілька недоліків. Чим глибша модель, тим краща її здатність до узагальнення, однак пошук такої мережі обтяжливий. Натомість, модель, що згенерована на невеликому наборі даних, може не відповідати більшому набору даних, що вимагає створення нової моделі для більшого набору даних.

Автоматизовані підходи намагаються автоматизувати процес машинного навчання. Повна автоматизація є кінцевою метою всіх досліджень, проте, це є дуже складним завданням навіть з урахуванням сучасного технологічного прогресу. Розглянемо останні дослідження останніх років.

Auto-WEKA [28] є дослідженням, пов'язаним з автоматизацією процесу навчання. Автори сформулювали проблему у область дослідження під назвою «Комбінований вибір алгоритму та оптимізація гіперпараметрів» (CASH – Combined Algorithm Selection and Hyperp arameter optimization). Вони запропонували концепцію оптимізації шляхом автоматичного вибору алгоритму з Java-пакету WEKA та його гіперпараметрів для заданого набору даних. Auto-WEKA використовує методи байєсівської оптимізації, зокрема, послідовну оптимізацію на основі моделей, яка може працювати як з категоріальними, так і з неперервними гіперпараметрами. Друга версія додала підтримку регресійних алгоритмів та багатопоточність.

Hyperopt-Sklearn [29] має схожість за функціоналом та базовою концепцією на Auto-WEKA, але більше зосереджений на іншій мові програмування. Hyperopt-Sklearn використовує python-пакет Scikit-learn[30]. Hyperopt-Sklearn використовує алгоритми Scikit-learn для навчання та перевірки моделі на основі цільової функції та використовує бібліотеку оптимізації Hyperopt для оптимізації параметрів. Hyperopt-Sklearn дозволяє вирішити проблему Auto-WEKA в масштабованості.

Auto-Sklearn [31] продовжує роботу над проблемою CASH, та вдосконалює існуючі концепції AutoML, такі як байєсівська оптимізація, використовуючи дані про продуктивність минулих спроб на подібних наборах даних та створюючи ансамблі алгоритмів без прив'язки до одного алгоритму як моделі (ансамбль моделей). Мета-навчання здійснюється шляхом збору даних про продуктивність та набір мета-функцій для великих наборів даних. Розуміння моделей та їх продуктивності призводить до ініціювання моделей з гіперпараметрами, які, ймовірно, підходять найкраще, що робить автоматизований процес більш ефективним та швидким. Цей метод використовується разом з байєсівською оптимізацією, щоб отримати найкращі результати від обох підходів. Нечисленні недоліки цієї системи полягали в тому, що вона не підтримувала регресійні алгоритми і погано працювала з великими наборами даних.

TPOT [32] представив новий підхід до області автоматичного розпізнавання мови за допомогою генетичного програмування. Хоча він зосереджений лише на алгоритмах класифікації та точності класифікації, він був успішним у структуруванні процесу машинного навчання на набір інкрементних завдань всього робочого процесу, що полегшує автоматизацію завдань одне за одним. TPOT містить три основні завдання: попередньої обробки функцій, вибору функцій, класифікації під наглядом. Завдяки використанню структурованого підходу, TPOT виявився дуже

гнучким, здатним масштабуватися та легко додавати або видаляти більше вузлів. Він також використовував попередні методи мета-навчання та ансамблі моделей, щоб зробити його ефективним рішенням.

Системи AutoML недостатньо розвинені, щоб працювати повністю незалежно, але можуть дати величезний приріст продуктивності, якщо їх координувати із залученням людини. Напівавтоматизовані підходи функціонують в основному як помічник для аналітики даних у завданнях машинного навчання та у якості початкового конструктора моделей, які можуть бути вдосконалені.

AutoCompete [33] – це напівавтоматизоване AutoML рішення, розроблене з метою отримати початкову статистичну модель для змагань з машинного навчання. Після організованого налаштування, AutoCompete має наступні кроки в робочому процесі: ідентифікатор типу даних та роздільник даних, виділення функцій, вибір моделі та гіперпараметрів. Система була здатна автоматично визначати необхідний тип машинного навчання (вибирати алгоритми обробки природної мови для відповідних текстових наборів даних) та уникати надмірної адаптації. Недоліком є зайва зосередженість на конкретному випадку використання.

PennAI [34] – система, що допомагає науковцям знайти найкращі моделі. Ця система має дуже систематизований та чіткий робочий процес, в якому важливу роль відігравало залучення людини. Було впроваджено нову функцію зберігання моделей, створених у попередніх операціях різними користувачами, та надання рекомендацій для нових операцій, а також зручний графічний інтерфейс користувача.

NLPAug [35] — це система, яка об'єднує багато типів операцій доповнення як для текстових, так і для звукових даних. Методи доповнення все ще вимагають від людини вибору операцій доповнення, а потім формування конкретної політики розширення даних для конкретних завдань.

AutoGOAL [36] – система для автоматичного пошуку найкращого способу вирішення заданого завдання. AutoGOAL дозволяє проводити багатоцільову оптимізацію за будь-яким набором цілей, який може надати користувач.

ОБГОВОРЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

AutoML дозволяє неспеціалістам створювати та впроваджувати системи штучного інтелекту, одночасно оптимізуючи робочі процеси для науковців та розробників даних. Розробники можуть створювати, навчати, перевіряти та розгортати генеративні моделі та інші системи глибокого навчання заощаджуючи час на підготовку та попередню обробку даних, розробку функцій та оптимізацію гіперпараметрів.

Розробка архітектури нейронних мереж завжди була трудомістким процесом, що вимагала експертних знань у певній галузі та досвіду. Автоматизоване машинне навчання прагне автоматизувати цей процес, використовуючи алгоритми та обчислювальні ресурси для виявлення ефективних архітектур нейронних мереж та, таким чином, автоматизоване машинне навчання робить машинне навчання більш доступним доступним для широкого кола людей.

Автоматизоване машинне навчання також впливає на галузь з обробку природної мови. Наразі це дозволяє отримати досить якісні моделі в умовах обмеженого бюджету.

ВИСНОВКИ

Останні досягнення у розробці систем штучного інтелекту стали можливими завдяки зростанню обчислювальних потужностей. AutoML є гарною сферою для випробування передових технологій, що дозволяє заощаджувати ресурси на розробку систем. Проаналізувавши результати існуючих рішень, було визначено, що незважаючи на те, що існує багато підходів та систем AutoML не існує універсального рішення. В результаті проведеної роботи представлено огляд цих підходів та систем, що може буди корисним для подальших досліджень.

REFERENCES

[1] Tornede, A., Deng, D., Eimer, T., Giovanelli, J., Mohan, A., Ruhkopf, T., ... & Lindauer, M. (2023). Automl in the age of large language models: Current challenges, future opportunities and risks. arXiv preprint arXiv:2306.08107.

[2] Ratner, E., Farmer, E., Douglas, C., & Lendasse, A. (2024). Extreme AutoML: Analysis of Classification, Regression, and NLP Performance. arXiv preprint arXiv:2412.07000.

[3] He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. Knowledge-Based Systems, 212, 106622.

[4] Nagarajah, T., & Poravi, G. (2019, March). A review on automated machine learning (AutoML) systems. In 2019 IEEE 5th International Conference for Convergence in Technology (I2CT) (pp. 1-6). IEEE.

[5] Gu, Y., & Leroy, G. (2019). Mechanisms for Automatic Training Data Labeling for Machine Learning. International Conference on Interaction Sciences.

[6] Krishnan, S., Wang, J., Franklin, M. J., Goldberg, K., Kraska, T., Milo, T., & Wu, E. (2015). SampleClean: Fast and Reliable Analytics on Dirty Data. *IEEE Data Eng. Bull.*, *38*(3), 59-75.

[7] Ilyas, I. F. (2016). Effective Data Cleaning with Continuous Evaluation. IEEE Data Eng. Bull., 39(2), 38-46.

[8] Frazier, P. I. (2018). Bayesian optimization. In Recent advances in optimization and modeling of contemporary problems (pp. 255-278). Informs.

[9] Lee, H. Y., Li, S. W., & Vu, T. (2022). Meta Learning for Natural Language Processing: A Survey. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 666-684).

[10] Brazdil, P., Vilalta, R., Giraud-Carrier, C., & Soares, C. (2017). Metalearning. In C. Sammut & G. I. Webb (Eds.), Encyclopedia of Machine Learning and Data Mining (pp. 818–823). doi:10.1007/978-1-4899-7687-1_543

[11] Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text data augmentation for deep learning. Journal of big Data, 8(1), 101.

[12] Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). Understanding data augmentation for classification: when to warp?. In 2016 international conference on digital image computing: techniques and applications (DICTA) (pp. 1-6). IEEE.

[13] Xie, Z., Wang, S. I., Li, J., Levy, D., Nie, A., Jurafsky, D., & Ng, A. Y. (2019). Data noising as smoothing in neural network language models. In 5th International Conference on Learning Representations, ICLR 2017.

[14] Yu, A. W., Dohan, D., Luong, M. T., Zhao, R., Chen, K., Norouzi, M., & Le, Q. V. (2018). QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In International Conference on Learning Representations.

[15] Tian, Y., Zhang, Y., & Zhang, H. (2023). Recent advances in stochastic gradient descent in deep learning. Mathematics, 11(3), 682.

[16] Kunstmann, L., Pina, D., Silva, F., Paes, A., Valduriez, P., De Oliveira, D., & Mattoso, M. (2021). Online deep learning hyperparameter tuning based on provenance analysis. Journal of Information and Data Management, 12(5).

[17] Salim, K., Yacine, A., & Akrem, B. M. (2024). Improving greedy adversarial attacks on text classification. International Journal of Information and Computer Security, 25(1-2), 141-166.

[18] Zabinsky, Z. B. (2009). Random search algorithms. Department of Industrial and Systems Engineering, University of Washington, USA, 34.

[19] LingChen, T. C., Khonsari, A., Lashkari, A., Nazari, M. R., Sambee, J. S., & Nascimento, M. A. (2020). UniformAugment: A Search-free Probabilistic Data Augmentation Approach. CoRR, abs/2003.14348.

[20] Ouyang, W., Tombari, F., Mattoccia, S., Di Stefano, L., & Cham, W. K. (2011). Performance evaluation of full search equivalent pattern matching algorithms. IEEE transactions on pattern analysis and machine intelligence, 34(1), 127-143.

[21] Chowdhary, K. R., and K. R. Chowdhary. "Heuristic Search." Fundamentals of Artificial Intelligence (2020): 239-272.

[22] Zabinsky, Z. B. (2009). Random search algorithms. Department of Industrial and Systems Engineering, University of Washington, USA, 34.

[23] Hasan, B. M. S., & Abdulazeez, A. M. (2021). A review of principal component analysis algorithm for dimensionality reduction. Journal of Soft Computing and Data Mining, 2(1), 20-30.

[24] Suthaharan, S., & Suthaharan, S. (2016). Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning, 207-235.

[25] Sun, B., Du, J., & Gao, T. (2009, November). Study on the improvement of K-nearest-neighbor algorithm. In 2009 International Conference on Artificial Intelligence and Computational Intelligence (Vol. 4, pp. 390-393). IEEE.

[26] Kamath, U., Liu, J., & Whitaker, J. (2019). Deep learning for NLP and speech recognition (Vol. 84, p. 1). Cham, Switzerland: Springer.

[27] Ren, W., Pan, J., Zhang, H., Cao, X., & Yang, M. H. (2020). Single image dehazing via multi-scale convolutional neural networks with holistic edges. International Journal of Computer Vision, 128, 240-259.

[28] Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2017). Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. Journal of Machine Learning Research, 18(25), 1-5.

[29] Komer, B., Bergstra, J., & Eliasmith, C. (2019). Hyperopt-sklearn. Automated machine learning: methods, systems, challenges, 97-111.

[30] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

[31] Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., & Hutter, F. (2022). Auto-sklearn 2.0: Hands-free automl via meta-learning. Journal of Machine Learning Research, 23(261), 1-61.

[32] Radecic, D. (2021). Machine Learning Automation with TPOT: Build, validate, and deploy fully automated machine learning models with Python. Packt Publishing Ltd.

[33] Thakur, A., & Krohn-Grimberghe, A. (2015). AutoCompete: A Framework for Machine Learning Competition. ArXiv, abs/1507.02188.

[34] La Cava, W., Williams, H., Fu, W., Vitale, S., Srivatsan, D., & Moore, J. H. (2020). Evaluating recommender systems for AI-driven biomedical informatics. Bioinformatics. doi:10.1093/bioinformatics/btaa698

[35] Ma, E. (2019). NLP Augmentation. Retrieved from https://github.com/makcedward/nlpaug

[36] Estévez-Velarde, S., Gutiérrez, Y., Almeida-Cruz, Y., & Montoyo, A. (2020). General-purpose hierarchical optimisation of machine learning pipelines with grammatical evolution. Information Sciences. doi:10.1016/j.ins.2020.07.035

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] Tornede, A., Deng, D., Eimer, T., Giovanelli, J., Mohan, A., Ruhkopf, T., ... & Lindauer, M. (2023). Automl in the age of large language models: Current challenges, future opportunities and risks. arXiv preprint arXiv:2306.08107.

[2] Ratner, E., Farmer, E., Douglas, C., & Lendasse, A. (2024). Extreme AutoML: Analysis of Classification, Regression, and NLP Performance. arXiv preprint arXiv:2412.07000.

[3] He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. Knowledge-Based Systems, 212, 106622.

[4] Nagarajah, T., & Poravi, G. (2019, March). A review on automated machine learning (AutoML) systems. In 2019 IEEE 5th International Conference for Convergence in Technology (I2CT) (pp. 1-6). IEEE.

[5] Gu, Y., & Leroy, G. (2019). Mechanisms for Automatic Training Data Labeling for Machine Learning. International Conference on Interaction Sciences.

[6] Krishnan, S., Wang, J., Franklin, M. J., Goldberg, K., Kraska, T., Milo, T., & Wu, E. (2015). SampleClean: Fast and Reliable Analytics on Dirty Data. IEEE Data Eng. Bull., 38(3), 59-75.

[7] Ilyas, I. F. (2016). Effective Data Cleaning with Continuous Evaluation. IEEE Data Eng. Bull., 39(2), 38-46.

[8] Frazier, P. I. (2018). Bayesian optimization. In Recent advances in optimization and modeling of contemporary problems (pp. 255-278). Informs.

[9] Lee, H. Y., Li, S. W., & Vu, T. (2022). Meta Learning for Natural Language Processing: A Survey. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 666-684).

[10] Brazdil, P., Vilalta, R., Giraud-Carrier, C., & Soares, C. (2017). Metalearning. In C. Sammut & G. I. Webb (Eds.), Encyclopedia of Machine Learning and Data Mining (pp. 818–823). doi:10.1007/978-1-4899-7687-1_543

[11] Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text data augmentation for deep learning. Journal of big Data, 8(1), 101.

[12] Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). Understanding data augmentation for classification: when to warp?. In 2016 international conference on digital image computing: techniques and applications (DICTA) (pp. 1-6). IEEE.

[13] Xie, Z., Wang, S. I., Li, J., Levy, D., Nie, A., Jurafsky, D., & Ng, A. Y. (2019). Data noising as smoothing in neural network language models. In 5th International Conference on Learning Representations, ICLR 2017.

[14] Yu, A. W., Dohan, D., Luong, M. T., Zhao, R., Chen, K., Norouzi, M., & Le, Q. V. (2018). QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In International Conference on Learning Representations.

[15] Tian, Y., Zhang, Y., & Zhang, H. (2023). Recent advances in stochastic gradient descent in deep learning. Mathematics, 11(3), 682.

[16] Kunstmann, L., Pina, D., Silva, F., Paes, A., Valduriez, P., De Oliveira, D., & Mattoso, M. (2021). Online deep learning hyperparameter tuning based on provenance analysis. Journal of Information and Data Management, 12(5).

[17] Salim, K., Yacine, A., & Akrem, B. M. (2024). Improving greedy adversarial attacks on text classification. International Journal of Information and Computer Security, 25(1-2), 141-166.

[18] Zabinsky, Z. B. (2009). Random search algorithms. Department of Industrial and Systems Engineering, University of Washington, USA, 34.

[19] LingChen, T. C., Khonsari, A., Lashkari, A., Nazari, M. R., Sambee, J. S., & Nascimento, M. A. (2020). UniformAugment: A Search-free Probabilistic Data Augmentation Approach. CoRR, abs/2003.14348.

[20] Ouyang, W., Tombari, F., Mattoccia, S., Di Stefano, L., & Cham, W. K. (2011). Performance evaluation of full search equivalent pattern matching algorithms. IEEE transactions on pattern analysis and machine intelligence, 34(1), 127-143.

[21] Chowdhary, K. R., and K. R. Chowdhary. "Heuristic Search." Fundamentals of Artificial Intelligence (2020): 239-272.

[22] Zabinsky, Z. B. (2009). Random search algorithms. Department of Industrial and Systems Engineering, University of Washington, USA, 34.

[23] Hasan, B. M. S., & Abdulazeez, A. M. (2021). A review of principal component analysis algorithm for dimensionality reduction. Journal of Soft Computing and Data Mining, 2(1), 20-30.

[24] Suthaharan, S., & Suthaharan, S. (2016). Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning, 207-235.

[25] Sun, B., Du, J., & Gao, T. (2009, November). Study on the improvement of K-nearest-neighbor algorithm. In 2009 International Conference on Artificial Intelligence and Computational Intelligence (Vol. 4, pp. 390-393). IEEE.

[26] Kamath, U., Liu, J., & Whitaker, J. (2019). Deep learning for NLP and speech recognition (Vol. 84, p. 1). Cham, Switzerland: Springer.

[27] Ren, W., Pan, J., Zhang, H., Cao, X., & Yang, M. H. (2020). Single image dehazing via multi-scale convolutional neural networks with holistic edges. International Journal of Computer Vision, 128, 240-259.

[28] Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2017). Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. Journal of Machine Learning Research, 18(25), 1-5.

[29] Komer, B., Bergstra, J., & Eliasmith, C. (2019). Hyperopt-sklearn. Automated machine learning: methods, systems, challenges, 97-111.

[30] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

[31] Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., & Hutter, F. (2022). Auto-sklearn 2.0: Hands-free automl via meta-learning. Journal of Machine Learning Research, 23(261), 1-61.

[32] Radecic, D. (2021). Machine Learning Automation with TPOT: Build, validate, and deploy fully automated machine learning models with Python. Packt Publishing Ltd.

[33] Thakur, A., & Krohn-Grimberghe, A. (2015). AutoCompete: A Framework for Machine Learning Competition. ArXiv, abs/1507.02188.

[34] La Cava, W., Williams, H., Fu, W., Vitale, S., Srivatsan, D., & Moore, J. H. (2020). Evaluating recommender systems for AI-driven biomedical informatics. Bioinformatics. doi:10.1093/bioinformatics/btaa698

[35] Ma, E. (2019). NLP Augmentation. Retrieved from https://github.com/makcedward/nlpaug

[36] Estévez-Velarde, S., Gutiérrez, Y., Almeida-Cruz, Y., & Montoyo, A. (2020). General-purpose hierarchical optimisation of machine learning pipelines with grammatical evolution. Information Sciences. doi:10.1016/j.ins.2020.07.035

Дата надходження статті до редакції: 25.02.2025
Дата затвердження статті до друку: 16.03.2025