УДК 004.421.2
DOI https://doi.org/10.15589/znp2025.4(502).34

ANALYSIS AND CLASSIFICATION OF STREAMING MACHINE LEARNING ALGORITHMS FOR REAL-TIME BIG DATA PROCESSING

АНАЛІЗ ТА КЛАСИФІКАЦІЯ ПОТОКОВИХ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ ОБРОБКИ ВЕЛИКИХ ОБСЯГІВ ДАНИХ У РЕАЛЬНОМУ ЧАСІ

Volodymyr I. Pankiv
pankiv.volodymyr@nltu.lviv.ua
ORCID: 0009-0006-2720-0864

В. І. Паньків,
аспірант

Ukrainian National Forestry University, Lviv
Національний лісотехнічний університет України, м. Львів

Abstract. *Purpose*. The paper provides a comprehensive analysis of streaming machine learning approaches focused on real-time processing of large amounts of data, with the aim of systematically outlining key methods, identifying their common principles and differences, and forming a coherent framework for further classification and research in this field.
Method. The analysis focuses on key requirements of the streaming environment, including memory limitations, the need for a single view of data, the dynamics of statistical properties, and the need for low latency computations. The specifics of conceptual drift, its types, and its impact on model stability are considered, as well as approaches to adaptation in changing conditions. Classes of streaming ML methods are systematized: online algorithms, drift adaptation models, streaming ensembles, algorithms for high-speed time series, and the architectural frameworks Apache Flink, Kafka Streams, Spark Structured Streaming, and the River and MOA libraries are analyzed.
Results. It is shown that online approaches provide incremental updates and constant computational complexity, but are sensitive to noise. Modern mechanisms for detecting conceptual changes and reconfiguring models are outlined, including Hoeffding trees and their adaptive modifications. In the context of ensemble approaches, it is demonstrated how combining incremental models allows balancing accuracy, stability, and response speed. For high-frequency flows, the effectiveness of combining statistical online update models with lightweight deep architectures is revealed. It is found that the performance of streaming systems is determined by the combination of algorithmic solutions and computational models with state maintenance, consistent time semantics, and low latency.
Scientific novelty. The generalization demonstrated the relationship between algorithmic mechanisms for adapting to drift, ensemble strategies, and architectural models of streaming processing, which allows us to consider streaming ML systems not only as a set of models, but as integrated computational and algorithmic ecosystems. The prospects for increasing the robustness of models, improving adaptive mechanisms, and standardizing benchmarks are emphasized.
Practical significance. The obtained generalizations are aimed at increasing the efficiency of streaming system design, optimizing algorithms in resource-constrained conditions, improving mechanisms for adapting to conceptual drift, and forming methodological foundations for comparative evaluation of streaming methods in real-world applications of high-speed data analytics.
Key words: drift; adaptation; detection; telemetry; forecasting.

Анотація. *Мета*. У роботі здійснено комплексний аналіз підходів потокового машинного навчання, орієнтованих на обробку великих обсягів даних у режимі реального часу, з метою системного окреслення ключових методів, виокремлення їхніх спільних принципів та відмінностей, а також формування узгодженої основи для подальшої класифікації та дослідження цієї галузі.
Методика. Аналіз зосереджено на ключових вимогах потокового середовища, серед яких обмеження пам'яті, потреба в одноразовому перегляді даних, динамічність статистичних властивостей і необхідність низької латентності обчислень. Розглянуто специфіку концептуального дрейфу, його типи та вплив на стабільність моделей, а також підходи до адаптації у змінних умовах. Систематизовано класи методів потокового ML: онлайн-алгоритми, моделі адаптації до дрейфу, стрімінгові ансамблі, алгоритми для високошвидкісних часових рядів, а також проаналізовано архітектурні фреймворки Apache Flink, Kafka Streams, Spark Structured Streaming та бібліотеки River і MOA.

Результати. Показано, що онлайн-підходи забезпечують інкрементальні оновлення та сталу обчислювальну складність, але є чутливими до шуму. Окреслено сучасні механізми виявлення концептуальних змін і реконфігурації моделей, включно з деревами Hoeffding та їхніми адаптивними модифікаціями. У контексті ансамблевих підходів продемонстровано, як комбінування інкрементальних моделей дозволяє збалансувати точність, стабільність і швидкість реагування. Для високочастотних потоків виявлено ефективність поєднання статистичних моделей онлайн-оновлення з легкими глибокими архітектурами. З'ясовано, що результативність потокових систем визначається поєднанням алгоритмічних рішень та обчислювальних моделей із підтримкою стану, узгодженою семантикою часу й низькою латентністю.
Наукова новизна. Узагальнення продемонструвало взаємозв'язок між алгоритмічними механізмами адаптації до дрейфу, ансамблевими стратегіями та архітектурними моделями потокової обробки, що дозволяє розглядати потокові ML-системи не лише як набір моделей, а як інтегровані обчислювально-алгоритмічні екосистеми. Акцентовано перспективи підвищення робастності моделей, удосконалення адаптивних механізмів і стандартизації бенчмарків.
Практична значимість. Отримані узагальнення спрямовані на підвищення ефективності проєктування потокових систем, оптимізацію алгоритмів у ресурсно обмежених умовах, удосконалення механізмів адаптації до концептуального дрейфу та формування методичних основ для порівняльного оцінювання потокових методів у реальних застосуваннях високошвидкісної аналітики даних.
Ключові слова: дрейф; адаптація; виявлення; телеметрія; прогнозування.

ПОСТАНОВКА ЗАДАЧІ

Потокові дані стають основою сучасних інформаційних систем, оскільки дедалі більше процесів генерують інформацію у безперервному режимі. На відміну від статичних наборів, такі дані не формують завершених вибірок і не дають можливості повернення до попередніх спостережень. Це ускладнює застосування традиційних методів машинного навчання, орієнтованих на повний і багаторазовий доступ до даних. Умови безперервності, мінливості та обмежених ресурсів створюють потребу в підходах, здатних працювати в середовищі, де інформація постійно оновлюється, а можливості зберігання та обробки залишаються обмеженими. Саме ці особливості формують складності побудови моделей, які здатні навчатися та залишатися коректними в умовах неповної інформації та постійного оновлення даних.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

У сучасних дослідженнях представлено широкий спектр підходів до потокового машинного навчання, зосереджених на обробці великих даних у реальному часі, адаптації до концептуального дрейфу та оптимізації моделей під обмежені ресурси. Так, Дж. Куі, К. Ву, Г. Дінг, Ю. Сюй, С. Феng [1] подають розгорнутий огляд машинного навчання для обробки великих даних, зосереджений на викликах та методах їх подолання. Автори підкреслюють, що традиційні алгоритми не здатні працювати з обсягами, швидкістю й різнорідністю сучасних потокових даних. У статті систематизовано підходи, включаючи глибоке навчання, розподілені моделі, паралельні алгоритми, передавальне навчання (transfer learning) та «online-learning». Окремо розглянуто критичні проблеми великих обсягів даних – масштаб, різноманітність, потоки, невизначеність і низьку цінність – та наведено відповідні методологічні рішення, від ADMM і MapReduce до методів зниження розмірності й обробки потоків.

Детальний огляд проблем навчання з потокових даних у присутності концептуального дрейфу та дисбалансу класів було здійснено Т. Гоенсом та Р. Полікаром [2]. Ці дослідники відзначають, що реальні потоки часто порушують припущення сталості розподілу, а рідкісні події роблять класифікацію нестійкою. У статті систематизовано типи дрейфу, зокрема зміну апріорних і умовних ймовірностей та зміщення межі рішень, а також проаналізовано взаємне підсилення труднощів дрейфу й дисбалансу. Розглянуто три групи методів: адаптивні базові класифікатори (дерева рішень, k-найближчих сусідів, Fuzzy ARTMAP), підходи зі змінним вікном і ваговими коефіцієнтами, а також ансамблеві техніки, здатні враховувати повторюваність концептів.

Джанардан, С. Мехта [3] здійснили огляд класифікації потокових даних із акцентом на проблему концептуального дрейфу. Автори наголошують, що постійні зміни розподілу даних, обмеження пам'яті та одноразовий перегляд унеможливлюють застосування традиційних алгоритмів. Стаття систематизує техніки адаптації: інкрементальне навчання, ансамблі та тригерні методи виявлення змін. Наведено класифікацію алгоритмів за типами моделей (деревні, правилоорієнтовані, сусідські, статистичні, ансамблеві) та їхню здатність реагувати на дрейф. Окремо розглянуто платформи для потокового аналізу й типові набори даних, що використовуються для оцінювання методів.

У роботі Н. АльНуаймі, М. Масуд, М. Серхані, Н. Закі [4] подано систематизований огляд алгоритмів відбору ознак у потокових даних, що виконують класифікацію в умовах високої розмірності та динамічних змін. Наголошено, що традиційні методи відбору ознак не придатні для режиму безперервного надходження даних, де доступний лише одноразовий

перегляд та обмежені ресурси пам'яті. У статті класифіковано підходи до аналізу релевантності та надмірності ознак, розділивши їх на статичні та потокові. Окремо розглянуто виклики, зумовлені масштабом великих даних, нестабільністю результатів, потребою в стійкості та низькою обчислювальною складністю. Також, автори підкреслюють, що ефективний відбір ознак є ключовою умовою для підвищення продуктивності потокових алгоритмів та забезпечення адаптивності моделей у реальному часі.

ВІДОКРЕМЛЕННЯ НЕВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ

Наявність широкого спектра досліджень у галузі потокового машинного навчання, не вичерпують цілої низки аспектів, які залишаються недостатньо опрацьованими та потребують систематизації. Так, наприклад, у наукових публікаціях розглядаються окремі підходи – адаптація до концептуального дрейфу, відбір ознак у потоках, інкрементальні та ансамблеві моделі, проте відсутній комплексний огляд, що окреслює спільні принципи їхнього функціонування та взаємозв'язок між алгоритмічними, обчислювальними та інфраструктурними компонентами потокових систем. Актуальною також залишається проблема інтегрованого розгляду адаптивних механізмів у контексті ресурсних обмежень реального часу, де більшість робіт зосереджена на окремих моделях, але не охоплює їхню продуктивність у різних потокових середовищах і конфігураціях обчислювальної інфраструктури. Значна кількість досліджень висвітлює окремі класи задач (класифікація, відбір ознак, виявлення дрейфу), однак бракує уніфікованої класифікації, що поєднувала б ці напрями в єдину схему аналізу потокових ML-методів.

МЕТА ДОСЛІДЖЕННЯ

Необхідно системно окреслити ключові підходи потокового машинного навчання, виокремити їхні спільні принципи та відмінності, а також сформувати узгоджену основу для подальшого аналізу й класифікації цієї галузі.

МЕТОДИ, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об'єктом дослідження є процес потокового машинного навчання в умовах безперервного надходження даних, обмежених обчислювальних ресурсів і динамічної змінності статистичних характеристик потоку.

Предметом дослідження є алгоритмічні підходи та інфраструктурні засоби реалізації потокового машинного навчання, які забезпечують інкрементальне оновлення моделей, адаптацію до концептуального дрейфу, низьку латентність обробки та ефективне використання пам'яті.

Для досягнення поставленої мети використано наступні **методи**: метод системного аналізу – для зіставлення архітектурних умов потокового середовища (реального часу, обмеження пам'яті, одноразовість перегляду даних) з алгоритмічними вимогами до моделей онлайн-навчання; аналітично-оглядовий метод – для визначення функціональних властивостей класичних та сучасних онлайн-алгоритмів (Perceptron, Passive-Aggressive, SGD та його модифікації), їх стійкості до шуму та здатності реагувати на концептуальний дрейф; порівняльний аналіз – для виявлення можливостей та обмежень адаптивних дерев рішень (Hoeffding Adaptive Tree, EFHAT), а також потокових ансамблів (OzaBag, Adaptive Random Forest, Streaming Random Patches) щодо латентності, точності, обчислювальної складності та стійкості до нестаціонарності; структурно-функціональний аналіз – для оцінювання ролі фреймворків і спеціалізованих бібліотек (Apache Flink, Spark Structured Streaming, Kafka Streams, River, MOA) у забезпеченні потокового ML, зокрема підтримки стану, семантики часу, подій та механізмів fault-tolerance; метод узагальнення – для формування класифікації потокових методів, визначення перспектив їх застосування в умовах високошвидкісних потоків і швидких структурних змін, включно з моделями на основі адаптивних механізмів глибокого навчання (наприклад, POLA).

ОСНОВНИЙ МАТЕРІАЛ

Потокове машинне навчання висуває низку специфічних вимог, через надходження даних безперервним потоком, обмеженням ресурсів, та динамічністю середовища. Ключовою проблемою є обмеження пам'яті, оскільки потік даних має потенційно необмежений розмір, тоді як доступні ресурси фіксовані [5]. Це зумовлює використання компактних структур, інкрементальних оновлень та мінімального збереження історичних даних.

Ще однією фундаментальною вимогою є одноразовий перегляд спостережень (single-pass) [6]. Потоковий режим передбачає, що кожне спостереження доступне лише в момент надходження і не може бути повторно оброблене. У цьому режимі алгоритм повинен оновлювати модель, використовуючи лише поточний приклад або обмежений контекст, що потребує інкрементальних методів із сталою обчислювальною складністю на кожному кроці. Також у потокових даних типово виникає концептуальний дрейф (concept drift), який може мати раптовий, поступовий, інкрементальний або періодичний характер і стосуватися як розподілу ознак (virtual drift), так і залежності між ознаками та цільовою змінною (real drift) [7]. Це вимагає адаптивних механізмів, здатних своєчасно реагувати на зміни, коригувати модель або повторно використовувати релевантні компоненти у разі повторної появи раніше спостережених концептів. Крім того потокові дані застосовуються у режимах

реального часу, де актуальність прогнозу визначається мілісекундами. Затримка навіть у кілька мілісекунд може призвести до втрати подій або зниження якості прийняття рішень, тому алгоритми мають забезпечувати низьку латентність і негайне інкрементальне оновлення моделі. Таким чином, потокове машинне навчання висуває комплекс вимог, зумовлених безперервністю потоку, обмеженими ресурсами та динамічними змінами розподілів. Алгоритми, що працюють у таких умовах, повинні забезпечувати інкрементальне оновлення моделі, адаптивність до концептуального дрейфу, низьку латентність обробки та ефективне використання пам'яті.

Одним із ключових класів методів, здатних працювати в таких умовах, є онлайн-алгоритми, що виконують інкрементальні оновлення моделі. А. Годішон-Баджоні, Н. Верге, О. Вінтенбергер [8] здійснили дослідження, присвячене стохастичним методам у потоковому середовищі, докладно показано, як алгоритми на основі стохастичного градієнтного спуску (Stochastic gradient descent – SGD) реалізують таке оновлення при наявності залежностей у даних та шумових градієнтів На відміну від пакетних методів, інкрементальний варіант SGD обчислює градієнт за єдиним прикладом або малою партією та одразу коригує параметри моделі, що робить його придатним до сценаріїв з обмеженою пам'яттю та вимогою низької латентності. Потокове середовище, однак, підсилює вплив шуму: градієнтні оцінки можуть бути зміщеними або автокорельованими, а отже модель реагує на локальні флуктуації, що знижує адаптивність до дрейфу. Водночас використання згладжувальних технік, таких як усереднення SGD розроблене Поляком та Руппертом, суттєво підвищує стабільність інкрементальних оцінок.

С. Хоі, Д. Саху та П. Чжао [9] запропнували широку класифікацію онлайн-методів та деталізує властивості класичних алгоритмів послідовного навчання, включно з Perceptron та Passive-Aggressive (PA) підходами. Обидва алгоритми використовують просту схему оновлення за принципом «прогноз – зворотний зв'язок – миттєва корекція» та забезпечують постійну обчислювальну складність на крок. Perceptron оновлює ваги лише у випадку помилки, а PA-алгоритми додатково враховують величину порушення маржі, коригуючи параметри настільки, наскільки це потрібно для негайного усунення помилкової класифікації. Такі методи зберігають ефективність навіть при великій швидкості надходження даних, хоча їхня чутливість до шумових або зміщених спостережень робить їх вразливими до локальних збурень та концептуальних змін.

З огляду на викладене, онлайн-алгоритми становлять один із базових класів методів потокового навчання: вони мінімізують обчислювальні та пам'ятеві витрати, підтримують одноразовий перегляд даних (single-pass) та зберігають високу швидкість оновлення. Їхні ключові переваги – простота, адаптивність і природна сумісність з інкрементальними сценаріями. Основними недоліками є підвищена чутливість до шуму, зміщених градієнтів та дрейфу, що вимагає використання стабілізувальних механізмів або розширених потокових методів у подальших підпунктах класифікації.

Наступною важливою групою методів, що працюють у потокових середовищах, є алгоритми, спеціально розроблені для обробки концептуального дрейфу. Через те що, потокові дані часто характеризуються зміною статистичних властивостей розподілу у часі, вони потребують методів, здатних фіксувати й оперативно реагувати на ці зміни. А. Естебан зі співавторами [10] запропонували низку підходів до виявлення та обробки різних типів концептуального дрейфу, включно з раптовими, поступовими, інкрементальними та повторними змінами, що зумовлюють необхідність адаптивних механізмів у потокових моделях. Ключовим завданням таких механізмів є підтримання актуальності моделі за умов обмеженої пам'яті та одноразового перегляду даних, що передбачає або модифікацію поточної структури, або часткове переорієнтування моделі на новий концепт.

Одним із найбільш досліджених напрямів адаптації є використання потокових дерев рішень, які поєднують локальність оновлень із можливістю реконфігурації структури [11]. У межах цієї групи значну увагу приділено алгоритмам родини Hoeffding Tree, що реалізують статистично обґрунтовані правила сплітів та дозволяють інкрементально освоювати нові концепти. У класичному підході, представленому Hoeffding Adaptive Tree (HAT), процес адаптації базується на виявленні змін у помилках прогнозування за допомогою детекторів дрейфу, таких як ADWIN, і на подальшому побудуванні альтернативних піддерев. Це забезпечує вибір між старою та новою моделлю вузла залежно від їхньої поточної точності, що робить метод ефективним у ситуаціях швидких або повторюваних змін концепту. Розвитком цієї лінії досліджень є Extremely Fast Hoeffding Adaptive Tree (EFHAT), який поєднує адаптивні механізми HAT зі статистично ефективнішим принципом сплітів із EFDT. Запропонований підхід демонструє підвищену швидкість реагування на дрейф завдяки менш консервативній політиці росту дерева, одночасно уникаючи недоліків ревізійної стратегії EFDT, непридатної для нестаціонарних потоків.

Експерименти показують, що EFHAT досягає нижчої «prequential error» помилки порівняно з HAT та EFDT на як стаціонарних, так і на потоках із концептуальним дрейфом, підтверджуючи ефективність поєднання статистичної пластичності з механізмами виявлення змін.

Ще одним окремим класом методів є стрімінгові ансамблі, у яких підвищення точності та стабільності

досягається за рахунок поєднання кількох інкрементальних моделей. На відміну від одиничних класифікаторів, ансамблеві підходи дозволяють поєднувати моделі з різними швидкостями адаптації, забезпечувати коректне оновлення моделі у відповідь на зміни розподілу в потоці даних. Г. Кассалес з колегами [12] представив детальний опис ключових методів «online bagging», включно з OzaBag, OBADWIN, OzaBagASHT, Leveraging Bagging, Adaptive Random Forest та Streaming Random Patches. У класичному OzaBag стохастичне зважування на основі розподілу Пуассона використовується для потокової апроксимації бутстрепінгу (bootstrapping), що дозволяє незалежно оновлювати кожну модель в ансамблі. Розширені варіанти, такі як OBADWIN, вбудовують механізми виявлення дрейфу через ADWIN, тоді як OzaBagASHT поєднує «online bagging» із адаптивними деревами рішень, забезпечуючи локальну реконфігурацію вузлів у разі зміни розподілу.

Leveraging Bagging збільшує інтенсивність оновлень за рахунок підвищених ваг спостережень, покращуючи реакцію ансамблю на рідкісні події. Adaptive Random Forest застосовує попереджувальні й дрейфові сигнали для селективної заміни дерев, зберігаючи баланс між стабільністю та чутливістю. А метод Streaming Random Patches комбінує випадкові підмножини ознак і часткові потокові вибірки, підвищуючи різноманітність моделей у ресурсно обмежених сценаріях. У сукупності ці методи забезпечують адаптивну та передбачувану роботу ансамблів за високої швидкості надходження даних.

В іншій роботі Г. Кассалеса та колег [13] запропоновано аналіз цих ансамблевих методів з погляду обчислювальних характеристик, зокрема латентності, пропускної здатності та енергоспоживання, що особливо важливо для систем реального часу та обмежених ресурсів. У роботі показано, що методи OzaBag, Leveraging Bagging, OBADWIN, ARF і SRP здатні зберігати високу точність навіть під значним навантаженням, однак вимагають оптимального балансування між частотою оновлень, глибиною окремих моделей і загальною кількістю класифікаторів в ансамблі.

Сучасний розвиток стрімінгових ансамблів доповнюється підходами «online boosting», які пропонують іншу стратегію посилення моделей, орієнтовану на багатопотокові сценарії та асинхронний дрейф. Зокрема, OBAL демонструє механізми динамічного зважування та адаптації до зміщення коваріат, використовуючи GMM-вагові стратегії та перенесення знань між кількома потоками [14]. Це свідчить про поступовий перехід від класичних ансамблів на основі бутстрепінгу до методів, що враховують складні часові та міжпотокові залежності.

Останнім з розглянутих класів методів є алгоритми, орієнтовані на високошвидкісні часові ряди, де дані надходять із значно вищою частотою та характеризуються швидкими локальними змінами. У таких сценаріях стандартні потокові моделі залишаються обмеженими через повільну адаптацію або недостатню чутливість до короткочасних структурних зламів, тому алгоритми цього класу поєднують інкрементальне навчання з механізмами прискореної реакції на дрейф і нерівномірність сигналуСучасні підходи до обробки високошвидкісних часових рядів поєднують статистичні моделі з онлайн-оновленням та легкі глибокі архітектури, орієнтовані на значно вищу частоту сигналу та різку змінність локальних закономірностей. А. Алмейда, С. Брас, С. Сарджento, Ф. Пінту [15] підкреслюють, що для таких сценаріїв дуже значущими є не тільки базові вимоги потокового аналізу, але і спроможність моделі коректно працювати за умов швидких структурних зламів, високого шуму та мілісекундних обмежень на обчислення.

Одним із сучасних прикладів методів цієї групи є POLA (Predicting Online by Learning-rate Adaptation) [16], що пропонує механізм автоматичної адаптації швидкості навчання для рекурентних нейронних мереж у потоковому режимі. Параметри моделі оновлюються за допомогою стохастичного градієнтного спуску, тоді як додатковий коефіцієнт β_t визначає ступінь корисності останнього пакета даних і запобігає перенавчанню на шумових або нестабільних фрагментах потоку. Навчання β_t реалізується через мета-оптимізацію, що імітує схему «interleaved-test-then-train», використовуючи розподіл пакета на «meta-training» і «meta-validation» частини. Такий підхід дозволяє моделі підлаштовувати темп оновлень відповідно до поточної складності сигналу та інтенсивності дрейфу.

Завдяки адаптивній швидкості навчання POLA демонструє стабільно низьку помилку прогнозування у сценаріях із сильно вираженою нестаціонарністю (наприклад, сонячна активність або енергоспоживання). Для високошвидкісних рядів це особливо важливо, оскільки такі дані часто характеризуються складними нелінійними залежностями, швидкими коливаннями та структурними зламами, які класичні методи (зокрема ARIMA-stream або ковзні вікна) не завжди встигають відстежувати. В цілому, моделі для високошвидкісних часових рядів розширюють можливості потокового ML у сценаріях з екстремальною швидкістю надходження даних. Їхня ключова властивість полягає в поєднанні високої чутливості до дрейфу та низької затримки оновлення, що робить їх придатними для складних динамічних середовищ – від енергетичних систем до моніторингу інфраструктури та фінансових ринків.

Розглянуті підходи до обробки потокових даних залежать не лише від алгоритмічних рішень, а й від здатності інфраструктури підтримувати обробку подій під високим навантаженням, узгоджені семантики часу, керування станом і стабільну

латентність. У сучасних потокових системах ці вимоги реалізуються на рівні архітектури фреймворків, які визначають модель виконання, механізми відмовостійкості (fault tolerance), підхід до керування станом і ступінь інтеграції з рештою екосистеми даних.

М. Фракгуліс та ін. [17], аналізуючи фреймворки, підкреслюють, що саме вибір архітектури визначає, наскільки ефективно можуть бути застосовані методи прогнозування, виявлення аномалій або високошвидкісного аналізу часових рядів, особливо в умовах реального часу.

Фреймворки загального призначення, такі як Apache Flink, Kafka Streams і Spark Structured Streaming, реалізують різні моделі обробки. С. Хеннінг, А. Фогель, М. Лейхтфрід, О. Ертл, Р. Рабізер [18], порівняли сучасні обчислювальні архітектури для потоків, та прийшли до висновку, що Flink орієнтований на подієву модель, забезпечує низьку латентність та складні семантики часу, включно з підтримкою стану на рівні операторів і механізмами точно-одного оброблення. Kafka Streams, навпаки, є вбудованою бібліотекою поверх Kafka, зосередженою на локальній обробці та мінімізації координації, що робить її придатною для сценаріїв, де важлива тісна інтеграція з топіками й висока доступність. У свою чергу Spark Structured Streaming базується на мікропакетній моделі, що спрощує програмну абстракцію, але водночас збільшує латентність порівняно з подієвими системами. Також відзначено, що попри популярність Spark, він поступається Flink у сценаріях з інтенсивними потоками та високими вимогами до затримки.

Окрему категорію становлять онлайн-бібліотеки, орієнтовані саме на інкрементальне навчання. River (наступник scikit-multiflow) надає універсальний набір онлайн-класифікаторів, регресорів, адаптивних дерев рішень, методів роботи з концептуальним дрейфом і механізмів оцінювання в режимі «prequential». На відміну від фреймворків рівня Flink або Spark, River не управляє розподіленим виконанням, натомість виступає легковаговим ML-шаром, що інтегрується з будь-яким стрімінговим середовищем.

MOA (Massive Online Analysis) має подібну спрямованість, але з акцентом на експериментальні дослідження потокових моделей. Вона розглядається як інструмент для тестування алгоритмів на високих швидкостях потоку, завдяки низькому накладному навантаженню та великій кількості реалізованих дерев, ансамблів і детекторів дрейфу.

Узагальнення результатів показує, що розглянуті інструменти охоплюють різні рівні потокової обробки й виконують взаємодоповнювальні функції.

Apache Flink і Spark Structured Streaming забезпечують інфраструктурний рівень, відповідаючи за розподілене виконання, fault tolerance, керування станом і семантики часу. Kafka Streams діє як інтегрований обробник подій на рівні брокера, оптимізований для локальної обробки та мінімальної координації. River і MOA, своєю чергою, виконують роль онлайн-шару машинного навчання: вони реалізують інкрементальне оновлення моделей, механізми адаптації до концептуального дрейфу й оцінювання в режимі prequential, не беручи на себе питання масштабування або керування потоками. Проведений у роботі порівняльний аналіз підкреслює, що для високонавантажених систем доцільним є поєднання обох рівнів: фреймворк забезпечує обробку подій і підтримку стану, тоді як онлайн-бібліотеки відповідають за адаптивне навчання моделей у реальному часі. У реальних системах потокові методи використовуються в задачах, що потребують постійного оновлення моделей та стабільної роботи під динамічним навантаженням. Так наприклад, сенсорні та IoT-мережі потребують потокових методів для безперервного моніторингу параметрів середовища, обладнання чи інфраструктурних вузлів, що дає змогу оперативно виявляти відхилення без накопичення великих масивів даних [19]. Високошвидкісні потоки на зразок мережевого трафіку або телеметрії потребують механізмів виявлення аномалій, здатних працювати з мілісекундними обмеженнями.

Як переконливо довів П. Чжоу [20] поєднання поточності, обмежених ресурсів і вимоги одноразового перегляду робить інкрементальні моделі критичними для кібербезпеки, де навіть мілісекундні затримки можуть призвести до пропуску атак.

Персоналізовані рекомендаційні системи демонструють подібні вимоги: моделі мають підлаштовуватися до нових патернів поведінки користувача в момент їх появи, щоб утримувати точність прогнозів у динамічних середовищах. Такі сценарії підкреслюють перевагу потокових алгоритмів, які забезпечують адаптивність і сталість роботи, недосяжні для пакетних підходів. Узагальнюючи, сучасні дослідження підкреслюють кілька напрямів розвитку потокового ML. Значна увага приділяється підвищенню здатності моделей зберігати стабільність за наявності шуму й автокореляції, що залишається невирішеною проблемою навіть для стохастичних методів і адаптивних дерев.

У галузі адаптації до дрейфу триває пошук балансів між швидкістю реагування та стабільністю моделей, особливо для ансамблевих підходів, де збільшення різноманітності нерідко погіршує продуктивність у ресурсно обмежених сценаріях. Для високошвидкісних часових рядів відкритими залишаються питання оптимального керування швидкістю навчання й запобігання локальному перенавчанню, що демонструють методи на кшталт POLA.

Архітектурні роботи відзначають потребу в стандартизованих бенчмарках для оцінювання затримки, пропускної здатності та узгодженості семантики часу в різних фреймворках. Крім того, відсутня

уніфікована методологія порівняння систем онлайн-навчання, що ускладнює відтворюваність і порівнянність результатів.

ВИСНОВКИ

Проведений аналіз дозволив систематизувати підходи потокового машинного навчання та виокремити їх основні класи, що застосовуються для обробки великих обсягів даних у реальному часі. Визначено ключові групи методів: онлайн-алгоритми, моделі адаптації до концептуального дрейфу, стрімінгові ансамблі та алгоритми для високошвидкісних часових рядів, кожна з яких вирішує специфічні завдання в умовах обмежених ресурсів і динамічних змін розподілів. Проаналізовано особливості їх роботи, механізми оновлення та вимоги до інфраструктури. Показано, що ефективність потокових систем значною мірою залежить від поєднання алгоритмічних рішень із відповідними архітектурами обробки даних. Сучасні дослідження акцентують на підвищенні робастності моделей до шуму, удосконаленні механізмів адаптації до дрейфу та розробленні бенчмарків, необхідних для порівняльної оцінки потокових методів.

REFERENCES

[1] Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing, 2016*, 1–16. https://doi.org/10.1186/s13634-016-0355-x

[2] Hoens, T. R., & Polikar, R. (2012). Learning from streaming data with concept drift and imbalance: An overview. *Progress in Artificial Intelligence, 1*(1), 1–14. https://doi.org/10.1007/s13748-011-0008-0

[3] Janardan, & Mehta, S. (2017). Concept drift in streaming data classification: Algorithms, platforms and issues. *Procedia Computer Science, 122*, 804–811. https://doi.org/10.1016/j.procs.2017.11.440

[4] AlNuaimi, N., Masud, M. M., Serhani, M. A., & Zaki, N. (2022). Streaming feature selection algorithms for big data: A survey. *Applied Computing and Informatics, 18*(1–2), 113–135. https://doi.org/10.1016/j.aci.2019.01.001

[5] Read, J., & Žliobaitė, I. (2025). Learning from data streams: An overview and update [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2212.14720

[6] Hu, L., Li, W., Lu, Y., & Hu, C. (2024). Scalable concept drift adaptation for stream data mining. *Complex & Intelligent Systems, 10*, 6725–6743. https://doi.org/10.1007/s40747-024-01524-x

[7] Suárez-Cetrulo, A. L., Quintana, D., & Cervantes, A. (2023). A survey on machine learning for recurring concept drifting data streams. *Expert Systems with Applications, 213*, 118934. https://doi.org/10.1016/j.eswa.2022.118934

[8] Godichon-Baggioni, A., Werge, N., & Wintenberger, O. (2023). Learning from time-dependent streaming data with online stochastic algorithms [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2205.12549

[9] Hoi, S. C. H., Sahoo, D., Lu, J., & Zhao, P. (2018). Online learning: A comprehensive survey [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.1802.02871

[10] Esteban, A., Cano, A., Zafra, A., & Ventura, S. (2024). Hoeffding adaptive trees for multi-label classification on data streams [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2410.20242

[11] Manapragada, C., Salehi, M., & Webb, G. I. (2022). Extremely fast Hoeffding adaptive tree. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE. https://doi.org/10.1109/ICDM54844.2022.00042

[12] Cassales, G., Gomes, H., Bifet, A., Pfahringer, B., & Senger, H. (2021). Improving the performance of bagging ensembles for data streams through mini-batching. *Information Sciences, 580*, 260–282. https://doi.org/10.1016/j.ins.2021.08.085

[13] Cassales, G., Gomes, H., Bifet, A., Pfahringer, B., & Senger, H. (2022). Balancing performance and energy consumption of bagging ensembles for the classification of data streams in edge computing [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2201.06205

[14] Yu, E., Lu, J., Zhang, B., & Zhang, G. (2024). Online boosting adaptive learning under concept drift for multistream classification [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2312.10841

[15] Almeida, A., Brás, S., Sargento, S., & Pinto, F. C. (2023). Time series big data: A survey on data stream frameworks, analysis and algorithms. *Journal of Big Data, 10*, 83. https://doi.org/10.1186/s40537-023-00772-5

[16] Zhang, W. (2021). POLA: Online time series prediction by adaptive learning rates [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2102.08907

[17] Fragkoulis, M., Carbone, P., Kalavri, V., & Katsifodimos, A. (2024). A survey on the evolution of stream processing systems. *The VLDB Journal, 33*, 507–541. https://doi.org/10.1007/s00778-023-00863-0

[18] Henning, S., Vogel, A., Leichtfried, M., Ertl, O., & Rabiser, R. (2024). ShuffleBench: A benchmark for large-scale data shuffling operations with distributed stream processing frameworks [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2403.04570

[19] Wu, Y., & Yusof, Y. (2024). Emerging trends in real-time recommendation systems: A deep dive into multi-behavior streaming processing and recommendation for e-commerce platforms. *Journal of Internet Services and Information Security, 14*(4), 45–66. https://doi.org/10.58346/JISIS.2024.I4.003

[20] Zhou, P. (2025). A survey of streaming data anomaly detection in network security. *PeerJ Computer Science*, 1–23. https://doi.org/10.7717/peerj-cs.3066

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] Qiu, J., Wu, Q., Ding, G., Xu, Y., Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing, 2016*, 1–16. https://doi.org/10.1186/s13634-016-0355-x

[2] Hoens, T. R., Polikar, R. (2012). Learning from streaming data with concept drift and imbalance: An overview. *Progress in Artificial Intelligence, 1*(1), 1–14. https://doi.org/10.1007/s13748-011-0008-0

[3] Janardan, Mehta, S. (2017). Concept drift in streaming data classification: Algorithms, platforms and issues. *Procedia Computer Science, 122*, 804–811. https://doi.org/10.1016/j.procs.2017.11.440

[4] AlNuaimi, N., Masud, M. M., Serhani, M. A., Zaki, N. (2022). Streaming feature selection algorithms for big data: A survey. *Applied Computing and Informatics, 18*(1–2), 113–135. https://doi.org/10.1016/j.aci.2019.01.001

[5] Read, J., Žliobaitė, I. (2025). Learning from data streams: An overview and update [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2212.14720

[6] Hu, L., Li, W., Lu, Y., Hu, C. (2024). Scalable concept drift adaptation for stream data mining. *Complex & Intelligent Systems, 10*, 6725–6743. https://doi.org/10.1007/s40747-024-01524-x

[7] Suárez-Cetrulo, A. L., Quintana, D., Cervantes, A. (2023). A survey on machine learning for recurring concept drifting data streams. *Expert Systems with Applications, 213*, 118934. https://doi.org/10.1016/j.eswa.2022.118934

[8] Godichon-Baggioni, A., Werge, N., Wintenberger, O. (2023). Learning from time-dependent streaming data with online stochastic algorithms [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2205.12549

[9] Hoi, S. C. H., Sahoo, D., Lu, J., Zhao, P. (2018). Online learning: A comprehensive survey [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.1802.02871

[10] Esteban, A., Cano, A., Zafra, A., Ventura, S. (2024). Hoeffding adaptive trees for multi-label classification on data streams [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2410.20242

[11] Manapragada, C., Salehi, M., Webb, G. I. (2022). Extremely fast Hoeffding adaptive tree. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE. https://doi.org/10.1109/ICDM54844.2022.00042

[12] Cassales, G., Gomes, H., Bifet, A., Pfahringer, B., Senger, H. (2021). Improving the performance of bagging ensembles for data streams through mini-batching. *Information Sciences, 580*, 260–282. https://doi.org/10.1016/j.ins.2021.08.085

[13] Cassales, G., Gomes, H., Bifet, A., Pfahringer, B., Senger, H. (2022). Balancing performance and energy consumption of bagging ensembles for the classification of data streams in edge computing [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2201.06205

[14] Yu, E., Lu, J., Zhang, B., Zhang, G. (2024). Online boosting adaptive learning under concept drift for multistream classification [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2312.10841

[15] Almeida, A., Brás, S., Sargento, S., Pinto, F. C. (2023). Time series big data: A survey on data stream frameworks, analysis and algorithms. *Journal of Big Data, 10*, 83. https://doi.org/10.1186/s40537-023-00772-5

[16] Zhang, W. (2021). POLA: Online time series prediction by adaptive learning rates [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2102.08907

[17] Fragkoulis, M., Carbone, P., Kalavri, V., Katsifodimos, A. (2024). A survey on the evolution of stream processing systems. *The VLDB Journal, 33*, 507–541. https://doi.org/10.1007/s00778-023-00863-0

[18] Henning, S., Vogel, A., Leichtfried, M., Ertl, O., Rabiser, R. (2024). ShuffleBench: A benchmark for large-scale data shuffling operations with distributed stream processing frameworks [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2403.04570

[19] Wu, Y., Yusof, Y. (2024). Emerging trends in real-time recommendation systems: A deep dive into multi-behavior streaming processing and recommendation for e-commerce platforms. *Journal of Internet Services and Information Security, 14*(4), 45–66. https://doi.org/10.58346/JISIS.2024.I4.003

[20] Zhou, P. (2025). A survey of streaming data anomaly detection in network security. *PeerJ Computer Science*, 1–23. https://doi.org/10.7717/peerj-cs.3066

Дата надходження статті до редакції: 25.11.2025
Дата затвердження статті до друку: 15.12.2025
Опубліковано: 30.12.2025