

<https://doi.org/10.31891/2219-9365-2023-75-2>

УДК 044

ЛІП'ЯНИНА-ГОНЧАРЕНКО Христина

Західноукраїнський національний університет

<https://orcid.org/0000-0002-2441-6292>

e-mail: kh.lipianina@wunu.edu.ua

ЮРКІВ Христина

Західноукраїнський національний університет

<https://orcid.org/0009-0007-4917-3251>

e-mail: kh.yurkiv@wunu.edu.ua

МЕТОДИ БУСТИНГОВОГО МАШИННОГО НАВЧАННЯ ДЛЯ НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ

Методи бустингового машинного навчання представляють собою потужний інструмент для ефективного прогнозування нестационарних часових рядів. Вони дозволяють створити сильну модель, яка виявляється значно точнішою, ніж окремі слабкі моделі. Один із найвідоміших методів бустингу для нестационарних часових рядів - це XGBoost, який використовує інноваційні характеристики, такі як градієнтний спуск з адаптивним темпом навчання та регуляризація, для досягнення високої точності. Ця стаття розглядає переваги використання методів бустингу для прогнозування нестационарних часових рядів, включаючи високу точність, стійкість до перенавчання та універсальність в застосуванні до різних завдань. Проте, слід враховувати, що методи бустингу також мають певні недоліки, такі як складність і висока вартість навчання. Незважаючи на ці обмеження, методи бустингового машинного навчання залишаються потужним інструментом для вирішення різноманітних завдань прогнозування нестационарних часових рядів, включаючи прогнозування цін на акції, погоди та продажів.

Ключові слова: бустингове машинне навчання, нестационарні часові ряди, прогнозування.

LIPIANINA-HONCHARENKO Khrystyna, YURKIV Khrystyna

West Ukrainian National University

BOOSTING MACHINE LEARNING METHODS FOR NON-STATIONARY TIME SERIES

Most machine learning algorithms assume that the data they are trained on is stationary, meaning that its distribution does not change over time. However, many time series we encounter in the real world are non-stationary, that is, their distribution changes over time. This can make forecasting based on this data difficult.

The non-stationarity of time series can be caused by many factors, such as seasonal fluctuations, trends, cyclicity and shocks. These factors can make forecasting non-stationary time series difficult because they can change the distribution of the data and make it less predictable.

Boosting machine learning algorithms is one approach to solving this problem. They work by training a sequence of weak models and then combining their results to produce a strong model. This can be effective for forecasting non-stationary time series because a strong model can compensate for the limitations of weak models.

Boosting machine learning methods are a powerful tool for effectively forecasting non-stationary time series data. They enable the creation of strong models that are significantly more accurate than individual weak models. One of the most well-known boosting methods for non-stationary time series is XGBoost, which utilizes innovative features such as gradient descent with adaptive learning rates and regularization to achieve high accuracy. This article discusses the advantages of using boosting methods for non-stationary time series forecasting, including high accuracy, resistance to overfitting, and versatility in handling various tasks. However, it should be noted that boosting methods also have certain drawbacks, such as training complexity and high cost. Despite these limitations, boosting machine learning methods remain a powerful tool for addressing a wide range of non-stationary time series forecasting tasks, including stock price prediction, weather forecasting, and sales forecasting.

Keywords: boosting machine learning, non-stationary time series, forecasting.

Постановка проблеми у загальному вигляді

та її зв'язок із важливими науковими чи практичними завданнями

Більшість алгоритмів машинного навчання припускають, що дані, на яких вони навчаються, є стаціонарними, тобто їх розподіл не змінюється з часом. Однак багато часових рядів, з якими ми стикаємося в реальному світі, є нестационарними, тобто їх розподіл змінюється з часом. Це може зробити прогнозування на основі цих даних складним.

Нестационарність часових рядів може бути викликана багатьма факторами, такими як сезонні коливання, тренди, циклічність і шоки. Ці фактори можуть зробити прогнозування нестационарних часових рядів складним, оскільки вони можуть змінити розподіл даних і зробити їх менш передбачуваними.

Бустингові алгоритми машинного навчання є одним із підходів до вирішення цієї проблеми. Вони працюють, навчаючи послідовність слабких моделей, а потім поєднуючи їх результати, щоб отримати сильну модель. Це може бути ефективним для прогнозування нестационарних часових рядів, оскільки сильна модель може компенсувати обмеження слабких моделей.

Аналіз відомих рішень і постановка задачі дослідження

У статті Ferreira та Figueiredo [1] надається загальний огляд методів бустингу в машинному навчанні, з фокусом на використанні слабких і сильних моделей. В дослідженні Tanha та інші [2] розглянуто застосування методів бустингу для багатокласової класифікації на незбалансованих даних, зокрема в мультикласовому контексті. Вони порівнюють різні методи бустингу і визначають, які з них є найкращими для обробки таких даних. У статті Bühlmann та Yu [3] надається загальний огляд бустингу з фокусом на методах AdaBoost та L2Boost для класифікації та регресії. Ця робота надає практичні поради щодо вибору слабких моделей та функцій втрат, а також посилання на програмне забезпечення. У роботі Praveena та Jaiganesh [4] проводиться огляд алгоритмів машинного навчання з фокусом на застосуванні бустингу для підвищення точності передбачення і класифікації. Основний акцент робиться на алгоритмах дерев рішень та методах опорних векторів. В статті Li та Мао [5] розглядається проблема класифікації незбалансованих даних і застосування методів бустингу для її вирішення. Автори подають огляд різних методів бустингу для цієї задачі. У роботі Galar та інших [6] вивчається використання ансамблевих методів для вирішення проблеми незбалансованих класів в класифікації. Автори проводять порівняльний аналіз різних підходів та визначають найбільш ефективні.

Загалом, наше дослідження вирізняється тим, що воно спеціалізується на використанні методів бустингу для нестационарних часових рядів, що є важливою областю дослідження в контексті прогнозування часових рядів зі змінною структурою.

Формулювання цілей статті

Головною метою полягає в тому, щоб надати огляд методів бустингового машинного навчання для нестационарних часових рядів. Це допоможе краще зрозуміти ці методи і їх застосування для прогнозування нестационарних часових рядів. Для досягнення поставленої мети потрібно виконати наступні задачі:

- Провести огляд методів бустингового машинного навчання.
- Детально описати методи бустингового машинного навчання і їх застосування до нестационарних часових рядів.
- Проаналізувати переваги і обмеження цих методів при роботі з нестационарними часовими рядами.

Виклад основного матеріалу

Бустингові алгоритми машинного навчання є хорошим вибором для прогнозування нестационарних часових рядів, оскільки вони мають ряд переваг (див. табл. 1) перед іншими методами, такими як: метод ковзного середнього, метод експоненційного згладжування, метод Холта-Вінтера, метод ARIMA та метод STL.

Таблиця 1

Порівняння бустингових методів зі звичайними

Характеристика	Бустингові методи	Звичайні методи
Припущення про розподіл даних	Не вимагають припущень про розподіл даних	Вимагають припущень про розподіл даних
Ефективність для нестационарних часових рядів	Висока	Низька
Адаптивність	Висока	Низька
Універсальність	Висока	Низька
Складність	Висока	Низька
Час обчислень	Високий	Низький

Однією з основних переваг бустингу є те, що він може компенсувати обмеження слабких моделей. Слабкі моделі, такі як дерева рішень, можуть бути ефективними для прогнозування статистичних часових рядів, однак вони можуть бути менш точними для нестационарних часових рядів. Бустинг дозволяє компенсувати це, навчаючи послідовність слабких моделей, а потім поєднуючи їх результати, щоб отримати сильну модель.

Іншою перевагою бустингу є те, що він може бути адаптивним. Бустингові алгоритми можуть навчатися на помилках попередніх моделей, щоб покращити точність прогнозів. Це може бути корисно для нестационарних часових рядів, оскільки вони можуть швидко змінюватися з часом.

Нарешті, бустингові алгоритми є досить універсальними. Вони можуть використовуватися для прогнозування різного роду часових рядів, включаючи часові ряди з трендами, сезонними коливаннями та циклічністю.

Отже, бустингові методи мають ряд переваг перед звичайними методами для прогнозування нестационарних часових рядів. Вони не вимагають припущень про розподіл даних, ефективні для нестационарних часових рядів, адаптивні і універсальні. Однак вони також більш складні і вимагають більше часу для обчислень.

Вибір конкретного алгоритму бустингового машинного навчання для прогнозування нестационарних часових рядів залежить від конкретних даних і завдань. Однак загальна рекомендація полягає в тому, щоб використовувати алгоритм, який дозволяє регулювати вагу вкладу кожної слабкої моделі. Далі розглянемо більш детально існуючі методи бустингового машинного навчання, які можуть бути використані для прогнозування нестационарних часових рядів, а саме: XGBoost (eXtreme Gradient Boosting), LightGBM (Light Gradient Boosting Machine), CatBoost (CatBoost), GBRT (Gradient Boosted Regression Trees), AdaBoost (Adaptive Boosting), Gradient Boosting Forest.

XGBoost (eXtreme Gradient Boosting) - це алгоритм машинного навчання, який використовує метод градієнтного бустингу для побудови моделі прогнозування. XGBoost був розроблений Чень Тяньці і його командою в 2016 році.

Математично XGBoost можна описати наступним чином:

Нехай $D = \{(x_i, y_i)\}_{i=1}^n$ - набір даних, де $x_i \in R^d$ - вектор ознак, а $y_i \in Y$ - таргет.

XGBoost - це алгоритм градієнтного бустингу над деревами рішень (рис.1). Градієнтний бустинг - це метод машинного навчання, який створює ансамбль моделей, кожен з яких називається "слабким вчителем". Слабкий вчитель - це модель, яка в середньому краще випадкової, але не обов'язково дуже хорошою.

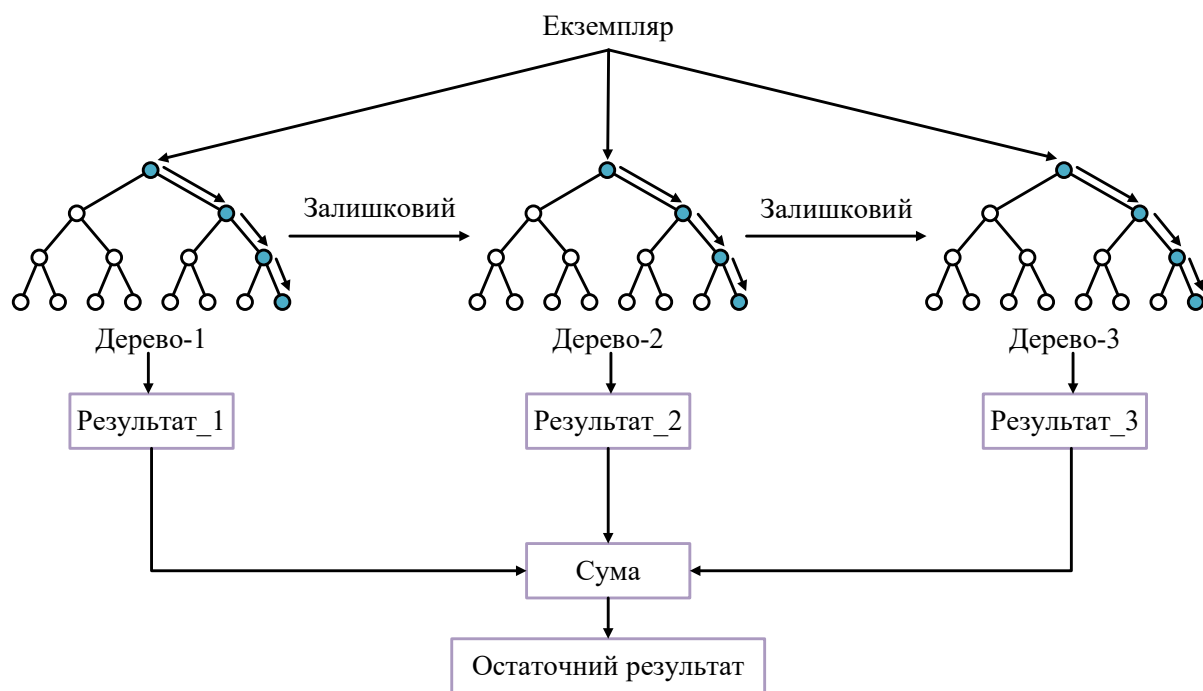


Рис. 1. Структура XGBoost

У XGBoost кожне дерево рішень навчається так, щоб мінімізувати функцію втрат, яка визначається як сума функцій втрат для всіх навчальних прикладів:

$$L(\theta) = \sum_{i=1}^n l(y_i, f(x_i; \theta))$$

де θ - параметри моделі, l - функція втрат, а $f(x_i; \theta)$ - прогноз моделі для прикладу x_i .

XGBoost використовує функцію втрат логістичної регресії для класифікації та квадратичну помилку для регресії.

Для класифікації функція втрат логістичної регресії визначається як:

$$l(y_i, f(x_i; \theta)) = -y_i \log \log (f(x_i; \theta)) - (1 - y_i) \log \log (1 - f(x_i; \theta))$$

де $y_i \in \{0, 1\}$.

Для регресії квадратична помилка визначається як:

$$l(y_i, f(x_i; \theta)) = (y_i - f(x_i; \theta))^2$$

XGBoost використовує оптимізатор за методом градієнтного спуску для знаходження параметрів моделі, які мінімізують функцію втрат.

На кожному кроці оптимізатор шукає такі параметри дерева, які зменшують значення функції втрат:

$$\theta \leftarrow \theta - \eta \nabla L(\theta)$$

де η - розмір кроку.

XGBoost має ряд додаткових функцій, які роблять його більш ефективним і стабільним, ніж інші алгоритми градієнтного бустингу. До них відносяться:

- L1 і L2 регуляризація: допомагає уникнути перенавчання, обмежуючи розміри параметрів моделі.
- Параметри дерева: XGBoost дозволяє налаштувати різні параметри дерева, такі як максимальна глибина дерева, мінімальна кількість зразків у листі і т.д.
- Параметри оптимізатора: XGBoost дозволяє налаштувати різні параметри оптимізатора, такі як розмір кроку, число ітерацій і т.д.

Отже, XGBoost є одним з найпопулярніших алгоритмів машинного навчання для прогнозування. Він був успішно використаний для різних задач, включаючи прогнозування цін на акції, прогнозування погоди і прогнозування продажів.

LightGBM (Light Gradient Boosting Machine) - це алгоритм машинного навчання, який використовує метод градієнтного бустингу для побудови моделі прогнозування (рис.2). LightGBM був розроблений Microsoft Research в 2016 році.

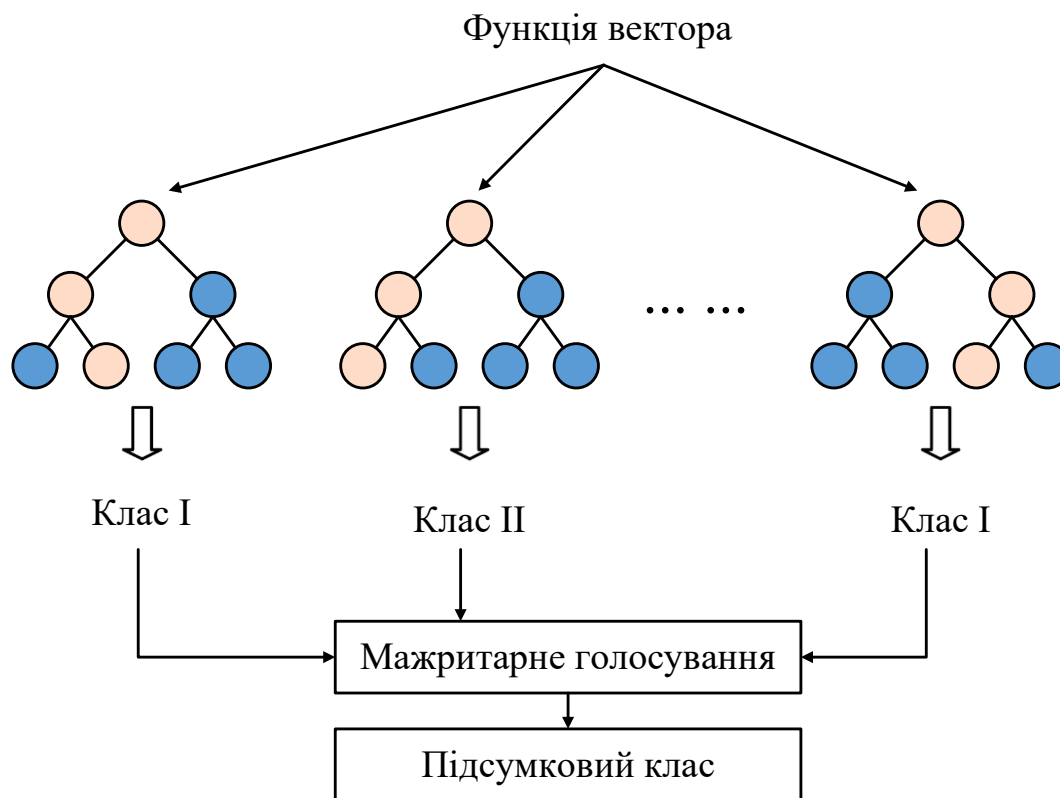


Рис. 2. Структура LightGBM

Математично LightGBM можна описати наступним чином:

Нехай $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ - це набір даних з n прикладами, де x_i - вектор розмірності d , а y_i - цілочисельна мітка. Мета навчання - побудувати модель, яка може передбачати мітку y для нового прикладу x .

LightGBM - це алгоритм градієнтного бустингу, який будує модель як суму над дерев'яними моделями. Кожна дерев'яна модель є одномірним деревом рішень, яке розбиває простір ознак на дві частини.

Для побудови дерева LightGBM використовує метод, заснований на гістограмі. Дані розподіляються по осередкам з використанням гістограмі розподілу. Цей метод можна оптимізувати і для розрідженого набору даних.

Математично процедура побудови дерева LightGBM можна описати наступним чином:

- Для кожного осередку i в гістограмі обчислюється коефіцієнт посилення G_i :

$$G_i = \sum_{j \in i} -y_j \nabla f(y_j)$$

де $f(y)$ - функція втрат, а $\nabla f(y)$ - її градієнт.

- Знаходжується найкращі значення для параметра t_i і значення ознаки v_i , що мінімізують G_i :

$$t_i = \arg G_i(t)$$

$$v_i = \arg G_i(v, t)$$

- Дані в осередку i розбиваються на дві частини за значенням ознаки v_i .
- Повторюємо попередні три кроки для всіх осередків в гістограмі.

Після того, як всі дерева побудовані, прогноз для нового прикладу x визначається як сума прогнозів всіх дерев:

$$\hat{y}(x) = \sum_{i=1}^m f_i(x)$$

де $f_i(x)$ - прогноз i -го дерева.

LightGBM має ряд особливостей, які роблять його ефективним алгоритмом для навчання моделей машинного навчання:

- Гістограмний метод дозволяє ефективно обробляти розріджені набори даних.
- Функція втрат може бути адаптована до різних задач машинного навчання.
- Параметр регуляризації дозволяє попередити перенавчання.

LightGBM є одним з найпопулярніших алгоритмів машинного навчання для прогнозування. Він був успішно використаний для різних задач, включаючи прогнозування цін на акції, прогнозування погоди і прогнозування продажів.

CatBoost (Categorical Boosting) - це алгоритм машинного навчання, який використовує метод градієнтного бустингу для побудови моделі прогнозування (рис.3).

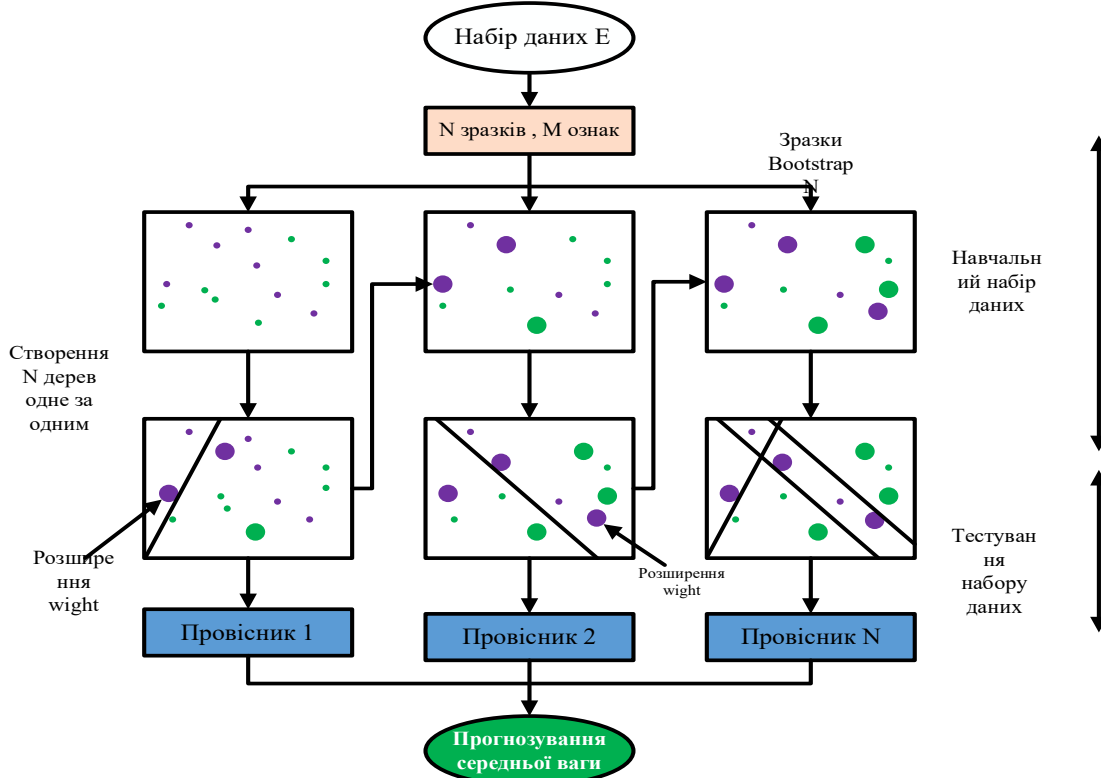


Рис. 3. Структура CatBoost

Математично CatBoost можна описати наступним чином:

Дано набір даних $\{(x_i, y_i)\}_{i=1}^n$, де x_i - вектор ознак розмірності d , а y_i - цілочисельний таргет.

Нехай T - кількість дерев у ансамблі, $f_t(x)$ - функція прогнозу t -го дерева, а w_t - ваговий коефіцієнт t -го дерева.

Тоді функція прогнозу CatBoost для даного набору даних визначається як:

$$f(x) = \sum_{t=1}^T w_t f_t(x)$$

Функції $f_t(x)$ є деревами рішень, які будуються послідовно, починаючи з $t=1$.

Для побудови t -го дерева використовується наступний алгоритм:

- Для кожного з d ознак x_i розраховується ймовірність p_{it} того, що ця ознака буде використана для побудови t -го листа.

- Ознака з найвищою ймовірністю p_{it} обирається як стовбурове значення x_{it} для t -го дерева.

- Набір даних $\{x_i\}_{i=1}^n$ розбивається на дві підмножини $\{x_i\}_{i \in A_t}$ і $\{x_i\}_{i \in B_t}$ залежно від значення x_{it} .

- Для кожного з підмножин $\{x_i\}_{i \in A_t}$ і $\{x_i\}_{i \in B_t}$ будується піддерево.

- Ваги w_t для t -го дерева визначаються таким чином, щоб мінімізувати функцію втрат:

$$L(y, f(x)) = \sum_{i=1}^n l(y_i, f(x_i))$$

де $l(y, f(x))$ - функція втрат, яка визначає похибку між фактичним значенням таргету y_i і прогнозом $f(x_i)$.

Для роботи з категоріальними ознаками CatBoost використовує наступний підхід:

- Для кожної категоріальної ознаки x_i будується окреме дерево.

- В кожному листі дерева для кожної категоріальної ознаки зберігається гістограма кількості об'єктів з кожною категорією.

- При прогнозуванні для даного об'єкта $\{x_i\}$ для кожної категоріальної ознаки x_i вибирається категорія з найбільшою частотою в листі, в якому знаходиться об'єкт.

CatBoost має ряд переваг порівняно з іншими алгоритмами машинного навчання:

- Він добре працює з числовими і категоріальними ознаками.

- Він швидкий і ефективний.

- Він може бути використаний для вирішення різних завдань машинного навчання, таких як класифікація, регресія та оцінка.

CatBoost використовується в багатьох компаніях, включаючи Google, Facebook, Amazon і Microsoft.

CatBoost є одним з найпопулярніших алгоритмів машинного навчання для прогнозування. Він був успішно використаний для різних задач, включаючи прогнозування цін на акції, прогнозування погоди і прогнозування продажів.

Градієнтний бустинг (GBRT) - це метод ансамблевого машинного навчання, який використовує послідовність слабких моделей, щоб побудувати сильну модель (рис.4). Слабкі моделі, які використовуються в GBRT, зазвичай є деревами рішень.

Математично GBRT можна описати наступним чином:

Припустимо, у нас є n прикладів $(x_1, y_1), \dots, (x_n, y_n)$, де $x_i \in R^d$ - вектори характеристик, а $y_i \in R$ - цільові значення. GBRT - це ансамбль з T дерев рішень, де кожне дерево T_t має L_t листів.

Для кожного дерева T_t ми спочатку вибираємо L_t гіперпараметрів $\theta_t = (\theta_{t,1}, \dots, \theta_{t,L_t})$, які визначають структуру дерева. Потім, для кожного прикладу x_i , ми обчислюємо прогноз дерева T_t на цьому прикладі:

$$f_t(x_i; \theta_t) = \sum_{l=1}^{L_t} \beta_{t,l} I(x_i \in R_l^t),$$

де $R_{t,l}$ - l -тий лист дерева T_t , а $\beta_{t,l}$ - ваговий коефіцієнт цього листа.

Прогноз ансамблю GBRT для прикладу x_i визначається як сума прогнозів всіх дерев:

$$f(x_i) = \sum_{t=1}^T f_t(x_i; \theta_t).$$

Гіперпараметри θ_t для кожного дерева T_t оптимізуються за допомогою алгоритму стохастичного градієнтного спуску. На кожному кроці алгоритму ми вибираємо випадковий приклад x_i і мінімізуємо функцію втрат:

$$L(f(x_i)) = l(y_i, f(x_i)).$$

Де l - функція втрат, яка визначає, наскільки добре модель прогнозує цільові значення.

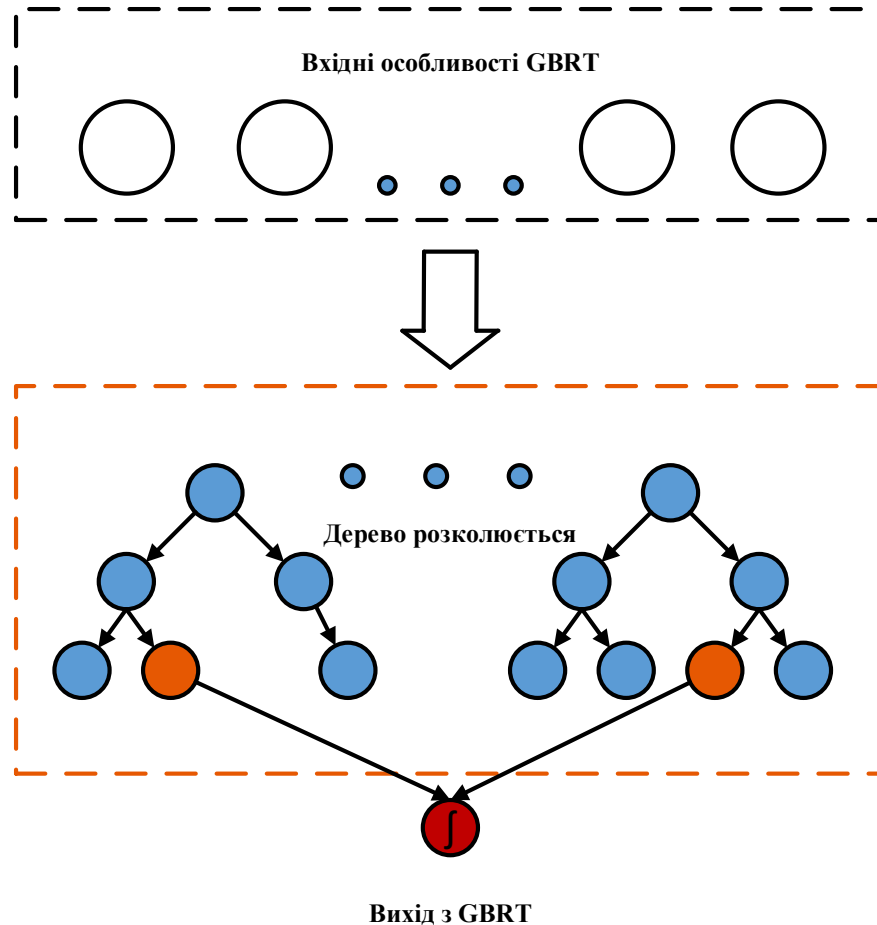


Рис. 4. Структура GBRT

Для GBRT можна використовувати різні функції втрат, такі як квадратична функція втрат, логарифмічна функція втрат або Hinge функція втрат.

Після того, як ми оптимізували гіперпараметри θ_t для всіх дерев, ми можемо використовувати ансамбль GBRT для прогнозування цільових значень для нових прикладів.

GBRT має ряд унікальних характеристик, які відрізняють його від інших алгоритмів бустингу:

- GBRT використовує метод градієнтного спуску з адаптивним темпом навчання. Це дозволяє GBRT швидше навчатися і досягати кращої точності.
- GBRT використовує регуляризацію, щоб запобігти перенавчанню. Це дозволяє GBRT зберігати універсальність і уникати перенавчання на даних навчання.
- GBRT має кілька вдосконалень, які покращують його продуктивність і точність.

GBRT - це потужний алгоритм машинного навчання, який може бути використаний для прогнозування різних задач. Він був успішно використаний для різних задач, включаючи прогнозування цін на акції, прогнозування погоди і прогнозування продажів.

AdaBoost (Adaptive Boosting) - це алгоритм машинного навчання, який використовує метод адаптивного бустингу для побудови моделі прогнозування (рис.5). Адаптивний бустинг - це метод ансамблевого навчання, який послідовно додає слабкі моделі до ансамблю, щоб зменшити функцію втрат.

Математично AdaBoost можна описати наступним чином:

Нехай D - це навчальний набір, що складається з N зразків. Кожний зразок має вигляд (x_i, y_i) , де x_i - це вектор ознак, а y_i - це клас, до якого належить зразок.

Алгоритм AdaBoost працює за наступною схемою:

- Для кожного зразка i присвоюємо початковий вага $w_i = 1/N$.

- Для $m=1,2,\dots,M$:
 - Будуємо слабкий класифікатор h_m , наприклад, дерево рішень або логічну регресію.
 - Визначаємо коефіцієнт a_m , який відповідає за вплив класифікатора h_m на кінцевий класифікатор.
 - Для кожного зразка i оновлюємо його вага:

$$w_i' = w_i \cdot \exp(-a_m y_i h_m(x_i)).$$
- Кінцевий класифікатор визначається як сума слабких класифікаторів, зважених їхніми коефіцієнтами:

$$h(x) = \sum_{m=1}^M a_m h_m(x).$$

Коефіцієнти a_m визначаються таким чином, щоб мінімізувати похибку класифікатора на навчальному наборі. Для цього можна використовувати метод градієнтного спуску. Ось математична формула для визначення коефіцієнта a_m :

$$a_m = \frac{1}{2} \ln \left(\frac{\sum_{i=1}^N w_i' \exp \sum (-y_i h_m(x_i))}{\sum_{i=1}^N w_i'} \right)$$

Ця формула гарантує, що класифікатор h_m буде давати мінімальну похибку на навчальному наборі з урахуванням ваг зразків w_i' .

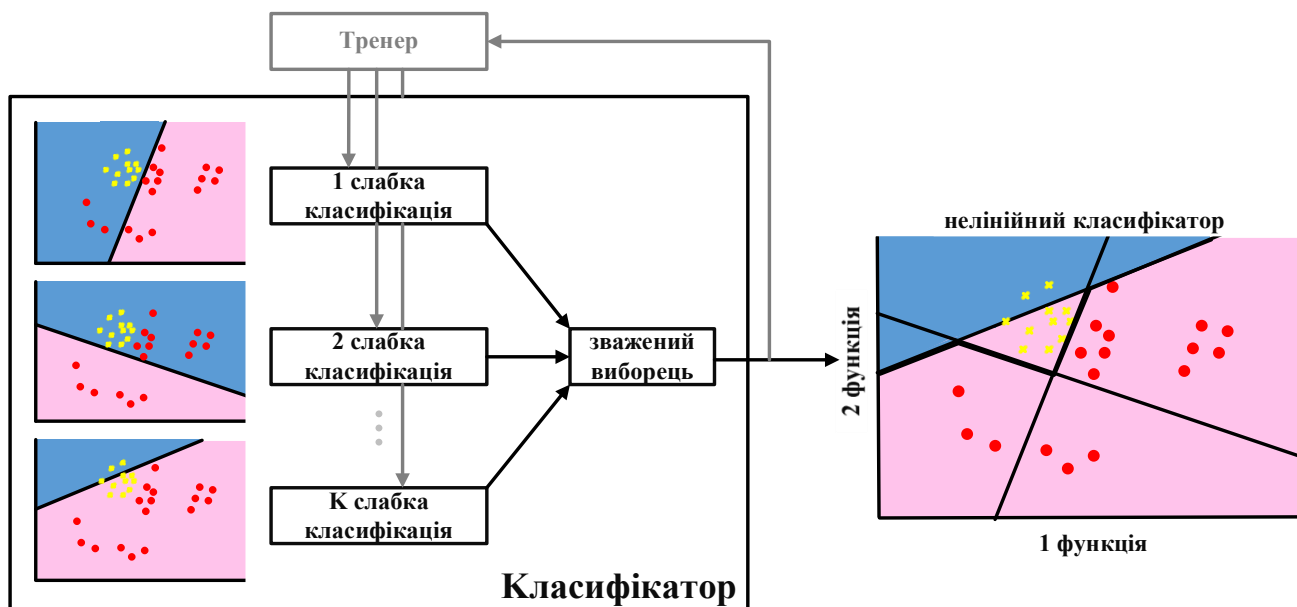


Рис. 5. Структура AdaBoost

На практиці алгоритм AdaBoost зазвичай працює добре з різними типами слабких класифікаторів. Однак найкращі результати досягаються, якщо використовувати слабкі класифікатори, які добре працюють з асиметричними розподілами даних. Наприклад, дерева рішень є хорошим вибором для AdaBoost, оскільки вони можуть навчитися розпізнавати нелінійні залежності в даних.

Даний алгоритм має ряд переваг:

- Він може використовувати будь-які слабкі класифікатори, що робить його досить універсальним.
- Він є стабільним і менш схильний до перенавчання, ніж деякі інші методи машинного навчання.
- Він може бути легко адаптований для роботи з нерівномірно розподіленими даними.

Однак, цей алгоритм також має деякі недоліки:

- Він може бути досить повільним для великих наборів даних.
- Він може бути чутливим до шуму в даних.

AdaBoost - це потужний алгоритм машинного навчання, який може бути використаний для прогнозування різних задач. Він був успішно використаний для різних задач, включаючи прогнозування цін на акції, прогнозування погоди і прогнозування продажів.

Градiєнтний бустинг лісів (GBDT) - це метод ансамблевого машинного навчання, який використовує послiдовнiсть слабких моделей, щоб побудувати сильну модель (рис.6). Слабкі моделі, які використовуються в GBDT, зазвичай є деревами рішень.

Математично GBDT можна описати наступним чином:

Нехай x буде входом в GBDT, а y - дійсною вихідною величиною. Нехай $f_1(x), f_2(x), \dots, f_T(x)$ - це T дерев рішень, які формують ансамбль. Тоді передбачення GBDT для x буде наступним:

$$f(x) = \sum_{t=1}^T f_t(x)$$

Дерева $f_1(x), f_2(x), \dots, f_T(x)$ формуються послiдовно, починаючи з пустого дерева $f_1(x)=0$. На кожному кроці t дерево $f_t(x)$ формується таким чином, щоб зменшити похибку $(y-f(x))$ на основі попередніх $t-1$ дерев:

$$f_t(x) = E[(y - f(x) - \sum_{i=1}^{t-1} f_i(x))^2]$$

Оптимізація цієї функції може бути виконана за допомогою методу градiєнтного спуску.

На кожному кроці t , ми можемо визначити градiєнт похибки як:

$$g(x) = 2(y - f(x) - \sum_{i=1}^{t-1} f_i(x))$$

Градiєнт $g(x)$ показує, в якому напрямку ми повинні змінити дерево $f_t(x)$, щоб зменшити похибку.

Дерево $f_t(x)$ може бути сформовано за допомогою будь-якого алгоритму формування дерева рішень, такого як ID3, C4.5 або CART. Однак, в GBDT часто використовується алгоритм формування дерева CART, оскільки він дозволяє контролювати складність дерева за допомогою параметра C .

GBDT є потужним алгоритмом машинного навчання, який може бути використаний для вирішення різних завдань, таких як класифікація, регресія та прогнозування. Він є одним з найпопулярніших алгоритмів машинного навчання, і він використовується в багатьох комерційних продуктах, таких як Amazon Recommendations і Google Ads.

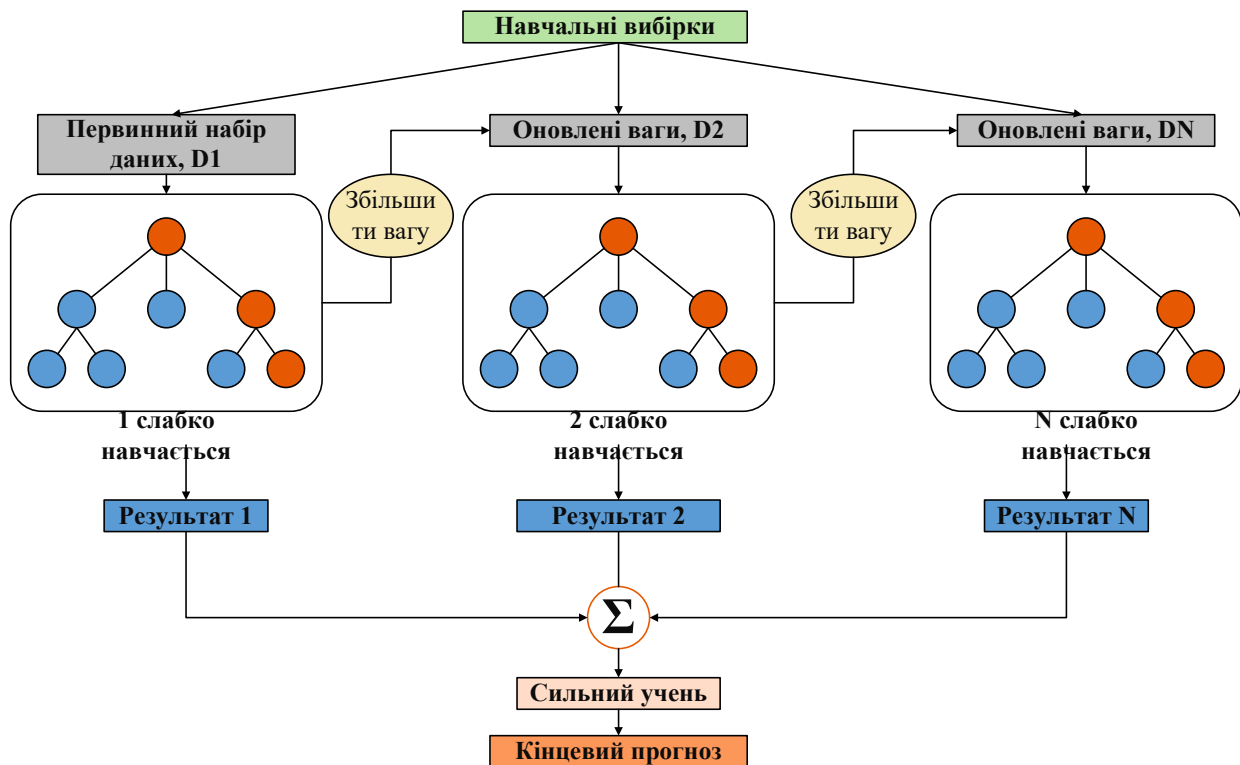


Рис. 6. Структура GBDT

Кілька важливих деталей про GBDT:

- GBDT є алгоритмом ансамблю, що означає, що він об'єднує прогнози декількох моделей, щоб отримати більш точне передбачення.
- GBDT є алгоритмом градієнтного спуску, що означає, що він оновлює свої параметри, щоб зменшити похибку.
- GBDT є алгоритмом з надмірним навчанням, що означає, що він може бути надто складним для певної множини даних. Однак, GBDT можна стабілізувати за допомогою різних методів, таких як регуляризація і випадковий вибір.

Отже, проведемо порівняння популярних бустингових методів (див. табл. 2). XGBoost, LightGBM, CatBoost, GBRT, AdaBoost і Gradient Boosting Forest - це популярні методи машинного навчання на основі дерев рішень. Вони мають схожий принцип роботи, але відрізняються деякими деталями реалізації.

XGBoost і LightGBM - це найпопулярніші методи з цього списку. Вони відрізняються високою швидкістю роботи і точністю. CatBoost також є швидким і точним методом, але він спеціалізований на задачі класифікації. GBRT - це більш універсальний метод, який може використовуватися як для регресії, так і для класифікації. AdaBoost - це відносно старий метод, який все ще може бути ефективним в деяких випадках. Gradient Boosting Forest - це простіший метод, який може бути хорошим вибором для початківців.

Одним з ключових факторів, який слід враховувати при виборі методу, є тип задачі. Якщо потрібно задачу регресії, то XGBoost, LightGBM або GBRT можуть бути хорошими варіантами. Якщо потрібно задачу класифікації, то XGBoost, LightGBM, CatBoost або GBRT можуть бути хорошими варіантами. Якщо потрібно простий метод, то Gradient Boosting Forest може бути хорошим вибором.

Також слід враховувати складність гіперпараметрів і швидкість роботи методу. Якщо є обмеження на час і ресурси, то можна вибрати більш простий метод, такий як Gradient Boosting Forest. Якщо немає обмежень, то потрібно вибрати більш складний метод, такий як XGBoost або LightGBM.

Таблиця 2

Порівняння бустингових методів

Особливість	XGBoost	LightGBM	CatBoost	GBRT	AdaBoost	GBF
Тип	Регресія	Регресія	Класифікація	Регресія	Класифікація	Регресія
Основа	Дерева рішень					
Ознаки	Числові, категорійні		Категорійні	Числові, категорійні		
Гіперпараметри	Багато					
Швидкість	Швидка			Повільна		
Стійкість до надмірного навчання	Висока			Середня		
Застосовність	Загальна					
Ітеративність	Так					
Метод оптимізації	Стохастичний градієнтний спуск					
Регуляризація	Так					
Обробка пропусків	Так					
Паралелізація	Так					
Швидкість	Швидка	Дуже швидка	Швидка			
Точність	Висока	Дуже висока	Висока			
Перенавчання	Можливе		Менше можливе	Можливе		
Складність hyperparameters	Висока	Середня	Середня	Висока	Висока	Середня
Популярність	Дуже популярна	Популярна				

Висновки та перспективи подальшого розвитку у даному напрямі

Методи бустингового машинного навчання є потужним інструментом для прогнозування нестационарних часових рядів. Вони дозволяють побудувати сильну модель, яка може бути більш точною, ніж окремі слабкі моделі.

Одним із найпопулярніших методів бустингу для нестационарних часових рядів є XGBoost. XGBoost використовує ряд унікальних характеристик, які дозволяють йому досягати високої точності, включаючи метод градієнтного спуску з адаптивним темпом навчання і регуляризацію.

Переваги використання методів бустингового машинного навчання для нестационарних часових рядів:

- Висока точність: методи бустингу можуть досягати високої точності прогнозування нестационарних часових рядів.
- Стійкість до перенавчання: методи бустингу, як правило, є досить стійкими до перенавчання, що є важливою характеристикою для нестационарних часових рядів.
- Універсальність: методи бустингу можуть бути використані для різних задач прогнозування нестационарних часових рядів.

Однак, слід зазначити, що методи бустингу також мають деякі недоліки:

- Висока складність: методи бустингу можуть бути досить складними для розуміння і використання.

- Висока вартість: методи бустингу можуть бути досить дорогими для навчання.

Незважаючи на ці недоліки, методи бустингового машинного навчання є потужним інструментом для прогнозування нестационарних часових рядів. Вони можуть бути використані для вирішення широкого кола задач, включаючи прогнозування цін на акції, прогнозування погоди і прогнозування продажів.

Література

1. Ferreira, A.J., Figueiredo, M.A.T. (2012). Boosting Algorithms: A Review of Methods, Theory, and Applications. In: Zhang, C., Ma, Y. (eds) Ensemble Machine Learning. Springer, New York, NY. https://doi.org/10.1007/978-1-4419-9326-7_2
2. Tanha, J., Abdi, Y., Samadi, N. et al. Boosting methods for multi-class imbalanced data classification: an experimental review. J Big Data 7, 70 (2020). <https://doi.org/10.1186/s40537-020-00349-y>
3. Bühlmann, P., & Yu, B. (2009). Boosting. WIREs Computational Statistics, 2(1), 69–74. <https://doi.org/10.1002/wics.55>
4. Praveena, M., & Jaiganesh, V. (2017). A Literature Review on Supervised Machine Learning Algorithms and Boosting Process. International Journal of Computer Applications, 169(8), 32–35. <https://doi.org/10.5120/ijca2017914816>
5. Li, Q., Mao, Y. A review of boosting methods for imbalanced data classification. Pattern Anal Applic 17, 679–693 (2014). <https://doi.org/10.1007/s10044-014-0392-8>
6. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 4, pp. 463-484, July 2012, doi: 10.1109/TSMCC.2011.2161285.
7. Chen, T., Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). ACM.
8. Wang W., Chakraborty G., Chakraborty B. Predicting the Risk of Chronic Kidney Disease (CKD) Using Machine Learning Algorithm. *Applied Sciences*. 2020. Vol. 11, no. 1. P. 202. URL: <https://doi.org/10.3390/app11010202> (date of access: 31.08.2023).
9. Chen, T., He, T., Benesty, J., Wang, X., Wang, S., & Chen, Z. (2016). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). ACM.
10. An Earthquake Forecast Model Based on Multi-Station PCA Algorithm / Y. Liu et al. *Applied Sciences*. 2022. Vol. 12, no. 7. P. 3311. URL: <https://doi.org/10.3390/app12073311> (date of access: 31.08.2023).
11. Prokhorov, D., Gusev, G., Vorontsov, A., & Yashunin, D. (2018). CatBoost: Gradient Boosting with Categorical Features. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1452-1461). ACM.
12. Machine Learning-Based Intelligent Prediction of Elastic Modulus of Rocks at Thar Coalfield / N. M. Shahani et al. *Sustainability*. 2022. Vol. 14, no. 6. P. 3689. URL: <https://doi.org/10.3390/su14063689> (date of access: 31.08.2023).
13. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232.
14. Feature selection and hyper parameters optimization for short-term wind power forecast / H. Huang et al. *Applied Intelligence*. 2021. Vol. 51, no. 10. P. 6752–6770. URL: <https://doi.org/10.1007/s10489-021-02191-y> (date of access: 31.08.2023).
15. Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In Proceedings of the 13th International Conference on Machine Learning (pp. 148-156). Morgan Kaufmann Publishers Inc.
16. Wang Z., Zhang J., Verma N. Realizing Low-Energy Classification Systems by Implementing Matrix Multiplication Directly Within an ADC. *IEEE Transactions on Biomedical Circuits and Systems*. 2015. P. 1. URL: <https://doi.org/10.1109/tbcas.2015.2500101> (date of access: 31.08.2023).
17. Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367-378.
18. Lei H., Zhao H., Ao T. A two-step merging strategy for incorporating multi-source precipitation products and gauge observations using machine learning classification and regression over China. *Hydrology and Earth System Sciences*. 2022. Vol. 26, no. 11. P. 2969–2995. URL: <https://doi.org/10.5194/hess-26-2969-2022> (date of access: 31.08.2023).

References

1. Ferreira, A.J., Figueiredo, M.A.T. (2012). Boosting Algorithms: A Review of Methods, Theory, and Applications. In: Zhang, C., Ma, Y. (eds) Ensemble Machine Learning. Springer, New York, NY. https://doi.org/10.1007/978-1-4419-9326-7_2

2. Tanha, J., Abdi, Y., Samadi, N. et al. Boosting methods for multi-class imbalanced data classification: an experimental review. *J Big Data* 7, 70 (2020). <https://doi.org/10.1186/s40537-020-00349-y>
3. Bühlmann, P., & Yu, B. (2009). Boosting. *WIREs Computational Statistics*, 2(1), 69–74. <https://doi.org/10.1002/wics.55>
4. Praveena, M., & Jaiganesh, V. (2017). A Literature Review on Supervised Machine Learning Algorithms and Boosting Process. *International Journal of Computer Applications*, 169(8), 32–35. <https://doi.org/10.5120/ijca2017914816>
5. Li, Q., Mao, Y. A review of boosting methods for imbalanced data classification. *Pattern Anal Applic* 17, 679–693 (2014). <https://doi.org/10.1007/s10044-014-0392-8>
6. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463-484, July 2012, doi: 10.1109/TSMCC.2011.2161285.
7. Chen, T., Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM.
8. Wang W., Chakraborty G., Chakraborty B. Predicting the Risk of Chronic Kidney Disease (CKD) Using Machine Learning Algorithm. *Applied Sciences*. 2020. Vol. 11, no. 1. P. 202. URL: <https://doi.org/10.3390/app11010202> (date of access: 31.08.2023).
9. Chen, T., He, T., Benesty, J., Wang, X., Wang, S., & Chen, Z. (2016). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM.
10. An Earthquake Forecast Model Based on Multi-Station PCA Algorithm / Y. Liu et al. *Applied Sciences*. 2022. Vol. 12, no. 7. P. 3311. URL: <https://doi.org/10.3390/app12073311> (date of access: 31.08.2023).
11. Prokhorov, D., Gusev, G., Vorontsov, A., & Yashunin, D. (2018). CatBoost: Gradient Boosting with Categorical Features. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1452-1461). ACM.
12. Machine Learning-Based Intelligent Prediction of Elastic Modulus of Rocks at Thar Coalfield / N. M. Shahani et al. *Sustainability*. 2022. Vol. 14, no. 6. P. 3689. URL: <https://doi.org/10.3390/su14063689> (date of access: 31.08.2023).
13. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232.
14. Feature selection and hyper parameters optimization for short-term wind power forecast / H. Huang et al. *Applied Intelligence*. 2021. Vol. 51, no. 10. P. 6752–6770. URL: <https://doi.org/10.1007/s10489-021-02191-y> (date of access: 31.08.2023).
15. Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning* (pp. 148-156). Morgan Kaufmann Publishers Inc.
16. Wang Z., Zhang J., Verma N. Realizing Low-Energy Classification Systems by Implementing Matrix Multiplication Directly Within an ADC. *IEEE Transactions on Biomedical Circuits and Systems*. 2015. P. 1. URL: <https://doi.org/10.1109/tbcas.2015.2500101> (date of access: 31.08.2023).
17. Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367-378.
18. Lei H., Zhao H., Ao T. A two-step merging strategy for incorporating multi-source precipitation products and gauge observations using machine learning classification and regression over China. *Hydrology and Earth System Sciences*. 2022. Vol. 26, no. 11. P. 2969–2995. URL: <https://doi.org/10.5194/hess-26-2969-2022> (date of access: 31.08.2023).