

УДК 004.8:004.942:658.8

DOI <https://doi.org/10.32782/3041-2080/2025-4-13>

ОПТИМІЗАЦІЯ ГІПЕРПАРАМЕТРІВ XGBOOST ДЛЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ПРОГНОЗУВАННЯ В2В-ЗАМОВЛЕНЬ

Мірошниченко Сергій Олександрович,

студент групи 174-24-1дф

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»

ORCID ID: 0009-0007-4868-3006

Статтю присвячено розробленню методології оптимізації гіперпараметрів моделі машинного навчання XGBoost для інтелектуальних систем прогнозування успішності В2В-замовлень в умовах часових обмежень. Дослідження базується на комплексному порівнянні шести алгоритмів оптимізації, включаючи Random Search, Optuna TPE, Hyperopt TPE, генетичний алгоритм, оптимізацію роєм частинок та послідовну оптимізацію. Експериментальну валідацію проведено на двох наборах історичних даних обсягом 86 794 записи кожен із використанням метрики AUC-PR як цільової функції оптимізації та п'ятикратною стратифікованою крос-валідацією. Результати демонструють, що алгоритм Optuna TPE досягає найвищої ефективності з максимальними значеннями AUC-PR 0,9661 та 0,9780 для досліджуваних датасетів відповідно. Установлено оптимальний часовий інтервал роботи алгоритму в межах 240–360 секунд, після якого подальша оптимізація не забезпечує покращення результатів і може призводити до деградації якості моделі. Застосування оптимізованих гіперпараметрів забезпечило зменшення кількості помилок класифікації на 5,3–6,2 % порівняно з базовими налаштуваннями XGBoost. Дослідження включає детальний аналіз простору пошуку гіперпараметрів та розроблення системи контролю часових обмежень. Розроблена методологія має практичне значення для створення автоматизованих систем підтримки прийняття рішень у В2В-секторі та може бути впроваджена в комп'ютерно-інтегровані технології управління підприємством.

Ключові слова: оптимізація, машинне навчання, XGBoost, гіперпараметри, В2В-замовлення, AUC-PR, Optuna TPE, часові обмеження.

Miroshnichenko Serhii. XGBOOST hyperparameter optimization for intelligent B2B order forecasting systems

This paper focuses on developing a methodology for automated XGBoost hyperparameter optimization in intelligent systems for B2B order success prediction under time constraints. The research is based on comprehensive comparison of six optimization algorithms, including Random Search, Optuna TPE, Hyperopt TPE, genetic algorithm, particle swarm optimization, and sequential optimization. Experimental validation was conducted on two historical datasets with 86,794 records each, using AUC-PR metric as the optimization objective function and five-fold stratified cross-validation. Results demonstrate that Optuna TPE algorithm achieves the highest efficiency with maximum AUC-PR values of 0.9661 and 0.9780 for the studied datasets respectively. The optimal time interval for algorithm operation was established within 240–360 seconds, after which further optimization does not provide improvement and may lead to model quality degradation. Application of optimized hyperparameters ensured a reduction in classification errors by 5.3–6.2 % compared to default XGBoost settings. The study includes detailed analysis of hyperparameter search space and development of a time constraint control system. The methodology incorporates robust preprocessing techniques, including median imputation for missing values, interquartile range outlier detection with winsorization, and robust scaling for numerical features. The developed methodology has practical significance for creating automated decision support systems in the B2B sector and can be integrated into computer-integrated enterprise management technologies.

Keywords: optimization, machine learning, XGBoost, hyperparameters, B2B orders, AUC-PR, Optuna TPE, time constraints.

Вступ. Забезпечення стабільної роботи підприємства в оптимальному режимі потребує ефективного управління складськими запасами, своєчасного виробництва продукції та налагодженого механізму замовлення сировини. Для досягнення цієї мети важливою є можливість передбачення кількості продукції та сировини на складах підприємства. Як показано в [1], передбачення кількості продукції на складі здебільшого спирається

на вже затверджені замовлення. Однак від часу створення замовлення до моменту його затвердження може проходити значний час, що створює необхідність передбачення майбутньої успішності замовлення за допомогою моделей машинного навчання з використанням наявних історичних даних. Для забезпечення максимальної якості моделі відповідно до вибраної метрики та конкретного набору історичних даних необхідним є налаштування

гіперпараметрів [2]. Відповідно до сучасних підходів до розроблення інтелектуальних систем [3], планується періодичне перенавчання моделі з метою актуалізації на нових даних, що постійно надходять у систему [4]. Оптимізація гіперпараметрів є обов'язковою частиною цього процесу, оскільки зміна характеристик даних потенційно вимагає корекції оптимальних значень гіперпараметрів [5]. Водночас робота алгоритмів оптимізації відрізняється значною обчислювальною складністю та створює додаткове навантаження на робочу систему [6; 7]. Це зумовлює необхідність установлення обмеження вказаного впливу та вибору максимально ефективного з урахуванням установлених обмежень методу оптимізації.

Алгоритм XGBoost (eXtreme Gradient Boosting) належить до сімейства градієнтного бустингу та є одним із найефективніших методів машинного навчання для задач класифікації та регресії [8]. Основними його перевагами є висока точність предикції, ефективне використання обчислювальних ресурсів та вбудовані механізми регуляризації для запобігання перенавчанню [9]. Завдяки здатності ефективно обробляти великі обсяги даних, стійкості до перенавчання та можливості працювати з різнорідними даними [10] XGBoost доцільно використовувати для B2B-середовища, де обсяги транзакційних даних можуть бути надзвичайно великими [11]. Автори [11; 12] підкреслюють важливість оптимізації гіперпараметрів для досягнення максимальної ефективності XGBoost у B2B-прогнозуванні. Зокрема, у роботі [11] застосовано 5-кратну перехресну валідацію з пошуком по сітці (Grid Search) для визначення оптимальних гіперпараметрів та наголошено на важливості вибору відповідної метрики оцінки, зокрема площі під кривою точності – повноти (AUC-PR) для задач B2B-прогнозування. Як показує дослідження [13], для задач бінарної класифікації з незбалансованими класами метрика AUC-PR (Area Under the Precision – Recall Curve) є більш інформативною порівняно з традиційною AUC-ROC.

У сучасних виробничих системах критично важливо враховувати обмеження часу за оптимізації гіперпараметрів алгоритмів машинного навчання, оскільки це безпосередньо впливає на практичну застосовність їх у реальних бізнес-сценаріях [14]. У зв'язку з тим, що різні методи оптимізації демонструють суттєві відмінності у швидкості збіжності та ефективності використання доступного часового ліміту, емпіричне порівняння цих підходів є актуальним для практичних застосувань [15; 16].

Метою дослідження є розроблення та експериментальна валідація методології вибору оптимального алгоритму оптимізації гіперпараметрів XGBoost для задачі прогнозування успішності B2B-замовлень в умовах часових обмежень.

Для досягнення поставленої мети визначено такі завдання дослідження:

1. Провести порівняльний аналіз ефективності шести різних алгоритмів оптимізації гіперпараметрів в умовах часових обмежень.
2. Установити залежність між тривалістю процесу оптимізації та якістю отриманих результатів для кожного з досліджуваних алгоритмів.
3. Визначити оптимальні значення гіперпараметрів XGBoost для досліджуваних наборів даних.
4. Кількісно оцінити підвищення якості класифікації під час використання оптимізованих гіперпараметрів порівняно зі стандартними налаштуваннями моделі.
5. Сформулювати практичні рекомендації щодо вибору алгоритму оптимізації залежно від доступного часового ліміту та характеристик конкретної задачі.

Методи та методики дослідження. Дослідження базується на порівнянні шести методів оптимізації гіперпараметрів, що представляють різні підходи до пошуку в просторі параметрів: Random Search [18], Optuna TPE [19], Hyperopt TPE [20], Genetic Algorithm [21], Particle Swarm Optimization [22], Sequential Optimization [23].

Експериментальне дослідження проводилося з використанням відповідних Python-бібліотек на двох наборах історичних даних про успішність B2B-замовлень. Для забезпечення репрезентативності та збалансованості аналізу розміри обох датасетів було стандартизовано до 86 794 записів з однаковим розподілом цільових класів: 54 763 успішні замовлення (клас 1) та 32 031 неуспішне замовлення (клас 0), що відповідає співвідношенню приблизно 63:37, де успішність замовлення визначається тим, чи завершилося воно угодою, і представлено бінарною цільовою ознакою `is_successful`.

Перший датасет містить 23 характеристики, зокрема ідентифікатор замовлення, дату створення, ідентифікатор партнера, суму замовлення, кількість повідомлень та змін, статистичні показники партнера. Другий датасет включає 25 ознак із додатковими часовими характеристиками, інформацією про продавця та джерело замовлення.

Для обробки пропущених значень застосовано метод заповнення медіаною, що забезпечує робастність до екстремальних значень

[23]. Виявлення та обробку статистичних викидів реалізовано методом міжквартильного розмаху з порогом 1,5 [24] із застосуванням техніки вінсоризації замість видалення екстремальних значень [25]. Для нормалізації числових змінних використано Robust Scaler, що базується на медіані та міжквартильному діапазоні [26]. Циклічні часові характеристики оброблено за допомогою синусоїдальних та косинусних перетворень [27; 28], а категоріальні змінні закодовано методом ONE Extended Compact [29]. Для компенсації дисбалансу класів застосовується параметр `scale_pos_weight`, що обчислюється як відношення кількості негативних зразків до позитивних [30].

Розроблена експериментальна система дає змогу тестувати шість алгоритмів оптимізації в умовах обмеженого часового ліміту. Простір пошуку гіперпараметрів XGBoost включає дев'ять ключових параметрів [31], відповідні досліджувані значення для яких наведено в табл. 1.

Простір пошуку Θ визначається як декартовий добуток дискретних множин: $\Theta = \Theta_1 \times \Theta_2 \times \dots \times \Theta_9$, де кожна множина Θ_i містить десять попередньо визначених значень відповідного гіперпараметра. Загальна потужність простору пошуку становить $|\Theta| = 10^9$ унікальних конфігурацій, що створює достатньо складну оптимізаційну задачу для змістовного порівняння різних алгоритмів.

Цільова функція $f: \Theta \rightarrow R$ визначається як середнє значення AUC-PR, обчислене через п'ятикратну стратифіковану крос-валідацію на тренувальній вибірці, що можна математично визначити як [32; 33]:

$$f(\theta) = \frac{1}{5} \sum_{i=1}^5 AUC-PR(M(D_{train}^{(i)}, \theta), D_{val}^{(i)}),$$

де $M(D_{train}^{(i)}, \theta)$ представляє модель XGBoost, навчену на i -й тренувальній частині з параметрами θ ; $D_{val}^{(i)}$ – відповідна валідаційна частина.

Методологія базується на дворівневий перевірці, що передбачає holdout-валідацію та крос-валідацію [34; 35]. Первинне розділення датасету здійснюється з використанням стратифікованого семплування з розміром тестової вибірки 30 % [36]. Для оцінки якості конфігурацій гіперпараметрів застосовується стратифікована 5-кратна крос-валідація виключно на тренувальній вибірці [37].

Особливістю методології є контроль часових обмежень, що забезпечує строго рівні умови для всіх порівнюваних алгоритмів. Темпоральний контролер відстежує використання процесорного часу з точністю до мілісекунд та реалізує механізм graceful shutdown за досягнення встановленого ліміту. Система моніторингу періодично перевіряє залишковий час та забезпечує коректне завершення поточних операцій оцінювання, мінімізуючи втрату обчислювальних ресурсів. Як основна метрика для оптимізації використовується AUC-PR [38]. Детермінованість процесу розділення забезпечується фіксацією генератора псевдовипадкових чисел через параметр `random_state=42`. Така практика відповідає принципам відтворюваної науки (reproducible research) та дає змогу проводити коректне порівняння різних методів оптимізації на ідентичних підмножинах даних [39].

Експериментальне середовище характеризується такими технічними специфікаціями: обчислювальна платформа базується на AMD Ryzen 7 7735HS 3.20 GHz; 16 ГБ DDR5 5600 МГц; AMD Radeon Graphics 6 ГБ; програмне середовище Python 3.10 з відповідними науковими бібліотеками.

Результати. Проведено тестування алгоритмів на двох наборах даних для умов обмеження часу від 60 до 580 секунд із кроком 60 с, що дало змогу оцінити динаміку ефективності кожного методу. Загалом виконано 20 експериментів для кожного датасету з визначенням рангу ефективності та значень AUC-PR для кожного алгоритму за кожного часового бюджету (рис. 1, 2).

Таблиця 1

Значення параметрів простору пошуку гіперпараметрів XGBoost

Гіперпараметр	Значення									
<code>n_estimators</code>	50	100	150	200	300	400	500	650	800	1000
<code>learning_rate</code>	0,01	0,03	0,05	0,08	0,1	0,15	0,2	0,25	0,3	0,35
<code>max_depth</code>	3	4	5	6	7	8	9	10	11	12
<code>min_child_weight</code>	1	2	3	4	5	6	7	8	9	10
<code>gamma</code>	0	0,1	0,2	0,3	0,5	0,7	1	1,3	1,6	2
<code>subsample</code>	0,6	0,65	0,7	0,75	0,8	0,85	0,9	0,95	0,98	1
<code>colsample_bytree</code>	0,6	0,65	0,7	0,75	0,8	0,85	0,9	0,95	0,98	1
<code>reg_alpha</code>	0	0,01	0,05	0,1	0,2	0,3	0,5	0,7	1	1,5
<code>reg_lambda</code>	0,1	0,3	0,5	0,7	1	1,3	1,6	2	2,5	3

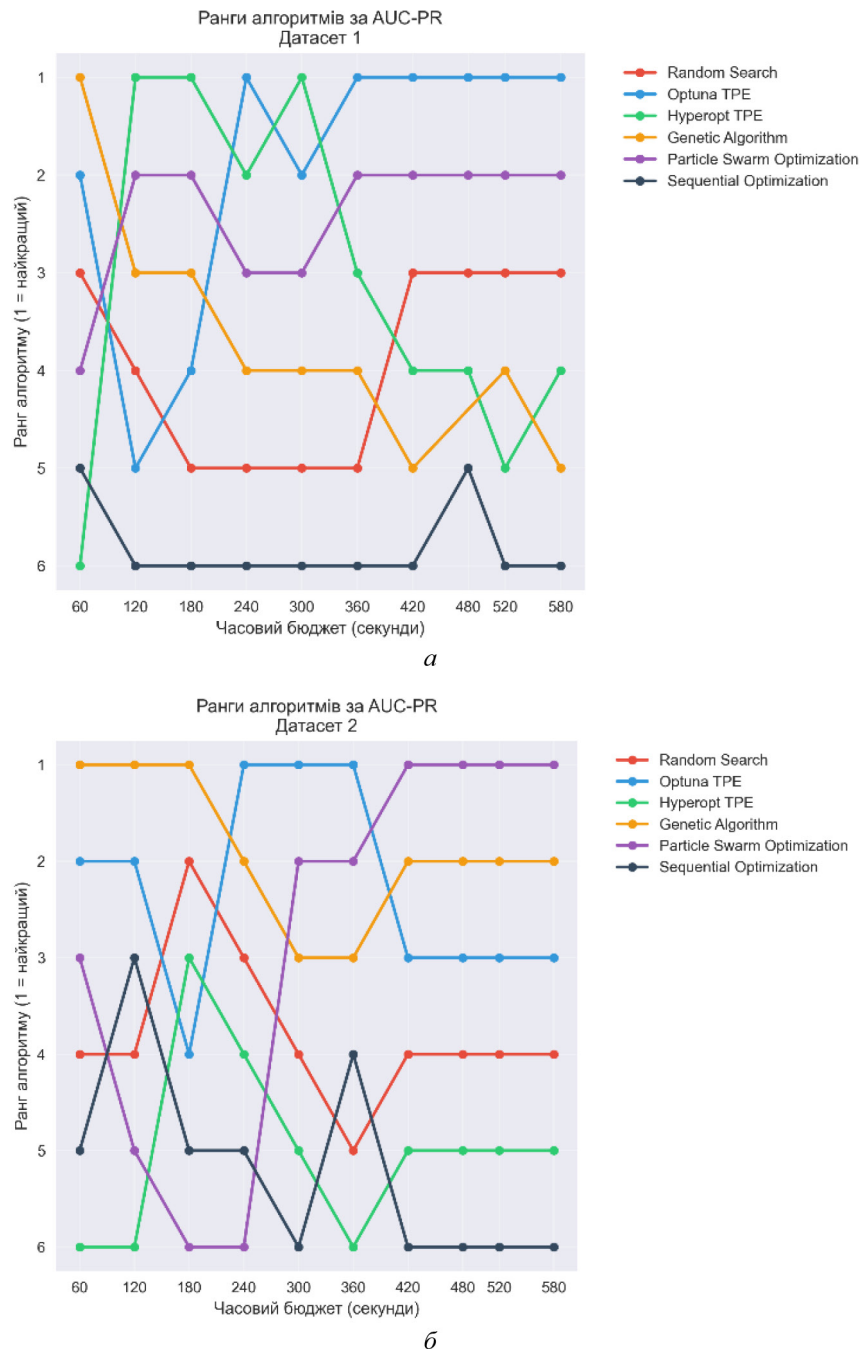


Рис. 1. Ранги алгоритмів за AUC-PR: а – датасет 1; б – датасет 2

За наведеними графіками можна бачити, що Optuna TPE визначено найкращим алгоритмом для обох датасетів. Для першого датасету – перша позиція з AUC-PR = 0,9661 від 360 до 580 с. Для другого датасету – перша позиція з AUC-PR = 0,9780 з 240 с, але після 360 с спостерігається зниження показника, та в підсумку третє місце з AUC-PR = 0,9777. Важливо відзначити, що для другого датасету жоден алгоритм не наблизився до якості Optuna TPE на пікових показниках. Підсумкове перше місце здобув Particle Swarm Optimization з AUC-PR = 0,9779,

що демонструє важливість вчасного зупинення процесу оптимізації.

Для кількісної характеристики отриманих результатів проведено порівняльний аналіз оптимізованих та стандартних гіперпараметрів. Оптиміальні конфігурації для обох датасетів демонструють суттєві відхилення від стандартних налаштувань XGBoost, що наведено в табл. 2.

Для оцінки якості класифікації моделі XGBoost з оптимізованими гіперпараметрами було порівняно з базовою реалізацією. Контрольну групу

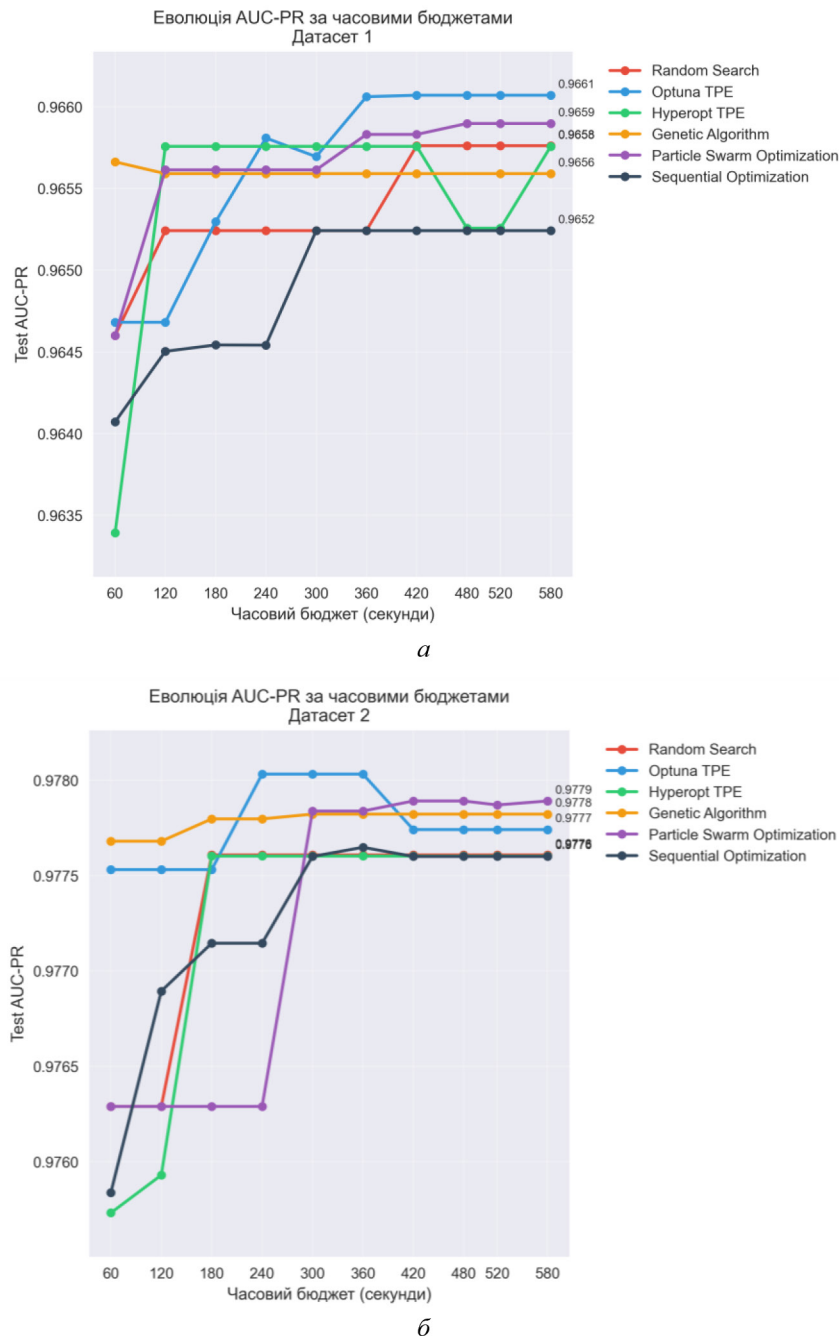


Рис. 2. Еволюція AUC-PR за часовими лімітами: а – датасет 1; б – датасет 2

представляв алгоритм XGBClassifier із базовими налаштуваннями бібліотеки xgboost. Для першого набору даних кількість помилкових класифікацій (False Positives + False Negatives) зменшилася з 2 507 до 2 351 (на 6,2 %). Для другого датасету – з 1 896 до 1 795 (на 5,3 %) відповідно.

Ці результати підкреслюють важливість використання коректної оптимізації гіперпараметрів під час тренування моделей машинного навчання для бізнес-задач.

Висновки. Дослідження демонструє ефективність алгоритму Optuna TPE для оптимізації гіперпараметрів XGBoost в задачі бінарної

класифікації успішності B2B-замовлень. Установлено, що цей метод показує найкращі результати для обох досліджуваних датасетів за використання метрики AUC-PR для крос-валідації та тестування. Критично важливим виявилось питання своєчасної зупинки процесу оптимізації. В умовах експерименту для алгоритму Optuna TPE оптимальний час роботи становить 240–360 секунд, після чого подальша оптимізація не приносить покращень, а в деяких випадках може призвести до погіршення результатів.

Коректно оптимізовані гіперпараметри дали змогу зменшити кількість помилок

Таблиця 2
Порівняння оптимізованих та стандартних гіперпараметрів

Параметр	Стандартні параметри XGBoost	Оптимізовані параметри	
		датасет 1	датасет 2
colsample_bytree	1,0	0,75	0,8
gamma	0,0	1,0	0,5
learning_rate	0,3	0,03	0,03
max_depth	6	12	12
min_child_weight	1	2	2
n_estimators	100	800	1000
reg_alpha	0,0	0,1	0,01
reg_lambda	1,0	0,5	1,0
subsample	1	1	1

у прогнозуванні успішності B2B-замовлень на 5,3–6,2 %, що має суттєвий практичний вплив для бізнес-застосувань.

Основні методологічні обмеження включають фокус на XGBoost та бінарній класифікації, фіксований простір гіперпараметрів та часові обмеження, що можуть впливати на повну конвергенцію окремих методів оптимізації.

Перспектива подальших досліджень охоплює валідацію результатів на інших моделях машинного навчання, розроблення гібридних підходів до оптимізації, дослідження ефективності для регресійних задач та впровадження у виробниче середовище через інтеграцію з ERP-системами.

Результати дослідження мають велике практичне значення для розроблення автоматизованих систем підтримки прийняття рішень у B2B-секторі. Запропонований підхід може бути адаптований для різних типів бізнес-прогнозування з урахуванням специфічних вимог до точності та часових обмежень.

ЛІТЕРАТУРА:

1. Miroshnychenko S. O. Improvement of the existing functionality for forecasting finished goods inventory at the enterprise warehouses based on the stock forecasted report ERP ODOO. *MININGMETALTECH 2024 – THE MINING AND METALS SECTOR: INTEGRATION OF BUSINESS, TECHNOLOGY AND EDUCATION. Volume 2.* 2024. P. 48–51. URL: <https://doi.org/10.30525/978-9934-26-506-8-133>
2. Nagassou M., Mwangi R.W., Nyarige E. A Hybrid Ensemble Learning Approach Utilizing Light Gradient Boosting Machine and Category Boosting Model for Lifestyle-Based Prediction of Type-II Diabetes Mellitus. *Journal of Data Analysis and Information Processing.* 2023. Vol. 11, no. 04. P. 480–511. <https://doi.org/10.4236/jdaip.2023.114025>
3. Miroshnychenko S. O., Koyfman O. O., Miroshnychenko V. I. The relevance of implementing an intelligent decision support system for enterprise resource optimization. *MININGMETALTECH 2024 – THE MINING AND METALS SECTOR: INTEGRATION OF BUSINESS, TECHNOLOGY AND EDUCATION. Volume 2.* 2024. P. 45–47. <https://doi.org/10.30525/978-9934-26-506-8-132>
4. Jawad Z. N., Balázs V. Machine learning-driven optimization of enterprise resource planning (ERP) systems: a comprehensive review. *Beni-Suef University Journal of Basic and Applied Sciences.* 2024. Vol. 13, no. 1. <https://doi.org/10.1186/s43088-023-00460-y>
5. Sibindi R., Mwangi R. W., Waititu A. G. A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices. *Engineering Reports.* 2022. <https://doi.org/10.1002/eng2.12599>
6. Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity / Y.A.Ali et al. *Processes.* 2023. Vol. 11, no. 2. P. 349. <https://doi.org/10.3390/pr11020349>
7. Bayesian Optimization with Additive Kernels for a Stepwise Calibration of Simulation Models for Cost-Effectiveness Analysis / D. Gómez-Guillén et al. *International Journal of Computational Intelligence Systems.* 2024. Vol. 17, no. 1. <https://doi.org/10.1007/s44196-024-00646-x>
8. Chen T., Guestrin C. XGBoost. *KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA. New York, NY, USA, 2016. <https://doi.org/10.1145/2939672.2939785>
9. Luo W. Prediction of Flight Delays Based on the XGboost Model. *2024 International Conference on Computers, Information Processing and Advanced Education (CIPAE)*, Ottawa, ON, Canada, 26–28 August 2024. 2024. P. 104–109. <https://doi.org/10.1109/cipae64326.2024.00024>
10. Abdullhadi S., Al-Qudah D. A., Abu-Salih B. Time-aware forecasting of search volume categories and actual purchase. *Heliyon.* 2024. Vol. 10, no. 3. P. e25034. <https://doi.org/10.1016/j.heliyon.2024.e25034>
11. De T.S., Singh P., Patel A. A Machine learning and Empirical Bayesian Approach for Predictive Buying in B2B E-commerce. *ICMLSC 2024: 2024 The 8th International Conference on Machine Learning and Soft Computing*, Singapore Singapore. New York, NY, USA, 2024. <https://doi.org/10.1145/3647750.3647754>

12. Enhancing customer retention in telecom industry with machine learning driven churn prediction / A. Sikri et al. *Scientific Reports*. 2024. Vol. 14, no. 1. <https://doi.org/10.1038/s41598-024-63750-0>
13. Saito T., Rehmsmeier M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*. 2015. Vol. 10, no. 3. P. e0118432. <https://doi.org/10.1371/journal.pone.0118432>
14. Applications of Machine Learning in Real-Time Control Systems: A Review / X. Zhao et al. *Measurement Science and Technology*. 2024. <https://doi.org/10.1088/1361-6501/ad8947>
15. Optimizing Machine Learning Algorithms for Landslide Susceptibility Mapping along the Karakoram Highway, Gilgit Baltistan, Pakistan: A Comparative Study of Baseline, Bayesian, and Metaheuristic Hyperparameter Optimization Techniques / F. Abbas et al. *Sensors*. 2023. Vol. 23, no. 15. P. 6843. <https://doi.org/10.3390/s23156843>
16. Bassi D., Singh H. A Comparative Study on Hyperparameter Optimization Methods in Software Vulnerability Prediction. *2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST)*, Mohali, India, 17–18 December 2021. 2021. <https://doi.org/10.1109/iccmst54943.2021.00046>
17. RandomizedSearchCV. *scikit-learn*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html (date of access: 29.06.2025).
18. optuna.samplers.TPESampler – Optuna 4.4.0 documentation. *Optuna: A hyperparameter optimization framework – Optuna 4.4.0 documentation*. URL: <https://optuna.readthedocs.io/en/stable/reference/samplers/generated/optuna.samplers.TPESampler.html> (date of access: 29.06.2025).
19. Hyperopt Documentation. *hyperopt.github.io ~ Hyperopt Project Home*. URL: <https://hyperopt.github.io/hyperopt/> (date of access: 29.06.2025).
20. DEAP documentation – DEAP 1.4.3 documentation. *DEAP documentation – DEAP 1.4.3 documentation*. URL: <https://deap.readthedocs.io/en/master/> (date of access: 29.06.2025).
21. Welcome to PySwarms's documentation! – PySwarms 1.3.0 documentation. *Welcome to PySwarms's documentation! – PySwarms 1.3.0 documentation*. URL: <https://pyswarms.readthedocs.io/en/latest/> (date of access: 29.06.2025).
22. spotpython. *GitHub Pages*. URL: <https://sequential-parameter-optimization.github.io/spotPython/> (date of access: 29.06.2025).
23. Joel L. O., Doorsamy W., Paul B. S. A Review of Missing Data Handling Techniques for Machine Learning. *International Journal of Innovative Technology and Interdisciplinary Sciences*. 2022. Vol. 5, no. 3. P. 971–1005. <https://doi.org/10.1515/IJITIS.2022.5.3.971-1005>
24. A Modern Introduction to Probability and Statistics / F. M. Dekking et al. London : Springer London, 2005. <https://doi.org/10.1007/1-84628-168-7>
25. Bringing clarity to investment decisions | MSCI. URL: https://www.msci.com/eqb/methodology/meth_docs/MSCI_GIMIVGMethod_Feb2021.pdf (date of access: 29.06.2025).
26. RobustScaler. *scikit-learn*. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html> (date of access: 29.06.2025).
27. Chakraborty D., Elzarka H. Advanced machine learning techniques for building performance simulation: a comparative analysis. *Journal of Building Performance Simulation*. 2018. Vol. 12, no. 2. P. 193–207. <https://doi.org/10.1080/19401493.2018.1498538>
28. CyclicalFeatures – 1.8.3. *Feature-engine – 1.8.3*. URL: https://feature-engine.trainindata.com/en/1.8.x/user_guide/creation/CyclicalFeatures.html (date of access: 29.06.2025).
29. Categorical Features Transformation with Compact One-Hot Encoder for Fraud Detection in Distributed Environment / I. Ul Haq et al. *Communications in Computer and Information Science*. Singapore, 2019. P. 69–80. https://doi.org/10.1007/978-981-13-6661-1_6
30. Notes on Parameter Tuning – xgboost 3.0.2 documentation. *XGBoost Documentation – xgboost 3.0.2 documentation*. URL: https://xgboost.readthedocs.io/en/stable/tutorials/param_tuning.html#handle-imbalanced-dataset (date of access: 29.06.2025).
31. XGBoost Parameters – xgboost 3.1.0-dev documentation. *XGBoost Documentation – xgboost 3.0.2 documentation*. <https://xgboost.readthedocs.io/en/latest/parameter.html> (date of access: 29.06.2025).
32. K-Fold Cross-Validation: An Effective Hyperparameter Tuning Technique in Machine Learning on GNSS Time Series for Movement Forecast / N. Le et al. *Recent Research on Geotechnical Engineering, Remote Sensing, Geophysics and Earthquake Seismology*. Cham, 2024. P. 377–382. https://doi.org/10.1007/978-3-031-43218-7_88
33. Soper D. S. Greed Is Good: Rapid Hyperparameter Optimization and Model Selection Using Greedy k-Fold Cross Validation. *Electronics*. 2021. Vol. 10, no. 16. P. 1973. <https://doi.org/10.3390/electronics10161973>

34. Awwalu J., Ogwueleka F. On Holdout and Cross Validation: A Comparison between Neural Network and Support Vector Machine. *International Journal of Trend in Research and Development*. 2019. Vol. 6, no. 2. P. 235–239.
35. Vaxign-ML: supervised machine learning reverse vaccinology model for improved prediction of bacterial protective antigens / E. Ong et al. *Bioinformatics*. 2020. Vol. 36, no. 10. P. 3185–3191. <https://doi.org/10.1093/bioinformatics/btaa119>
36. Särndal C.-E., Swensson B., Wretman J. Model Assisted Survey Sampling. New York, NY : Springer New York, 1992. <https://doi.org/10.1007/978-1-4612-4378-6>
37. Heart Disease Prediction Evaluation of Machine Learning Models with PSO-Optimized K-Fold Cross-Validation / N. Gupta et al. 2024 *IEEE Region 10 Symposium (TENSYP)*, New Delhi, India, 27–29 September 2024. 2024. P. 1–6. <https://doi.org/10.1109/tensymp61132.2024.10752215>
38. Sofaer H. R., Hoeting J. A., Jarnevich C. S. The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*. 2019. Vol. 10, no. 4. P. 565–577. <https://doi.org/10.1111/2041-210x.13140>
39. Shenouda J., Bajwa W. U. A Guide to Computational Reproducibility in Signal Processing and Machine Learning [Tips & Tricks]. *IEEE Signal Processing Magazine*. 2023. Vol. 40, no. 2. P. 141–151. <https://doi.org/10.1109/msp.2022.3217659>

REFERENCES:

1. Miroshnychenko, S. O. (2024). Improvement of the existing functionality for forecasting finished goods inventory at the enterprise warehouses based on the stock forecasted report ERP ODOO. In *MININGMETALTECH 2024 – The mining and metals sector: Integration of business, technology and education* (Vol. 2, pp. 48–51). <https://doi.org/10.30525/978-9934-26-506-8-133>
2. Nagassou, M., Mwangi, R. W., & Nyarige, E. (2023). A hybrid ensemble learning approach utilizing Light Gradient Boosting Machine and Category Boosting Model for lifestyle-based prediction of Type-II diabetes mellitus. *Journal of Data Analysis and Information Processing*, 11(4), 480–511. <https://doi.org/10.4236/jdaip.2023.114025>
3. Miroshnychenko, S. O., Koyfman, O. O., & Miroshnychenko, V. I. (2024). The relevance of implementing an intelligent decision support system for enterprise resource optimization. In *MININGMETALTECH 2024 – The mining and metals sector: Integration of business, technology and education* (Vol. 2, pp. 45–47). <https://doi.org/10.30525/978-9934-26-506-8-132>
4. Jawad, Z.N., & Balázs, V. (2024). Machine learning-driven optimization of enterprise resource planning (ERP) systems: A comprehensive review. *Beni-Suef University Journal of Basic and Applied Sciences*, 13(1). <https://doi.org/10.1186/s43088-023-00460-y>
5. Sibindi, R., Mwangi, R. W., & Waititu, A. G. (2022). A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices. *Engineering Reports*. <https://doi.org/10.1002/eng2.12599>
6. Ali, Y.A., Awwad, E.M., Al-Razgan, M., Maarouf, A., Usman, A., & Marzouk, R. (2023). Hyperparameter search for machine learning algorithms for optimizing the computational complexity. *Processes*, 11(2), 349. <https://doi.org/10.3390/pr11020349>
7. Gómez-Guillén, D., Anguita-Ruiz, A., Alcalá, R., & Herrera, F. (2024). Bayesian optimization with additive kernels for a stepwise calibration of simulation models for cost-effectiveness analysis. *International Journal of Computational Intelligence Systems*, 17(1). <https://doi.org/10.1007/s44196-024-00646-x>
8. Chen, T., & Guestrin, C. (2016). XGBoost. *KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA. <https://doi.org/10.1145/2939672.2939785>
9. Luo, W. (2024, August 26–28). Prediction of flight delays based on the XGboost model. *2024 International Conference on Computers, Information Processing and Advanced Education (CIPAE)*, Ottawa, ON, Canada (pp. 104–109). <https://doi.org/10.1109/cipae64326.2024.00024>
10. Abdullhadi, S., Al-Qudah, D.A., & Abu-Salih, B. (2024). Time-aware forecasting of search volume categories and actual purchase. *Heliyon*, 10(3), e25034. <https://doi.org/10.1016/j.heliyon.2024.e25034>
11. De, T.S., Singh, P., & Patel, A. (2024). A machine learning and empirical Bayesian approach for predictive buying in B2B e-commerce. *ICMLSC 2024: 2024 The 8th International Conference on Machine Learning and Soft Computing*, Singapore. <https://doi.org/10.1145/3647750.3647754>
12. Sikri, A., Harshwardhan, B., Sangeeta, & Parashar, D. (2024). Enhancing customer retention in telecom industry with machine learning driven churn prediction. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-63750-0>

13. Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3), e0118432. <https://doi.org/10.1371/journal.pone.0118432>
14. Zhao, X., Chen, L., Li, Y., Wang, H., Liu, J., & Zhang, M. (2024). Applications of machine learning in real-time control systems: A review. *Measurement Science and Technology*. <https://doi.org/10.1088/1361-6501/ad8947>
15. Abbas, F., Ali, S., Amjad, M., Shah, M.I., Kousar, R., Afzal, Z., Amin, M., Waheed, A., & Dong, S. (2023). Optimizing machine learning algorithms for landslide susceptibility mapping along the Karakoram Highway, Gilgit Baltistan, Pakistan: A comparative study of baseline, Bayesian, and metaheuristic hyperparameter optimization techniques. *Sensors*, 23(15), 6843. <https://doi.org/10.3390/s23156843>
16. Bassi, D., & Singh, H. (2021, December 17–18). A comparative study on hyperparameter optimization methods in software vulnerability prediction. *2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST)*, Mohali, India. <https://doi.org/10.1109/iccmst54943.2021.00046>
17. RandomizedSearchCV. (n.d.). *scikit-learn*. Retrieved June 29, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
18. optuna.samplers.TPESampler – Optuna 4.4.0 documentation. (n.d.). *Optuna: A hyperparameter optimization framework – Optuna 4.4.0 documentation*. Retrieved June 29, 2025, from <https://optuna.readthedocs.io/en/stable/reference/samplers/generated/optuna.samplers.TPESampler.html>
19. Hyperopt documentation. (n.d.). *hyperopt.github.io ~ Hyperopt Project Home*. Retrieved June 29, 2025, from <https://hyperopt.github.io/hyperopt/>
20. DEAP documentation – DEAP 1.4.3 documentation. (n.d.). *DEAP documentation – DEAP 1.4.3 documentation*. Retrieved June 29, 2025, from <https://deap.readthedocs.io/en/master/>
21. Welcome to PySwarms's documentation! – PySwarms 1.3.0 documentation. (n.d.). *Welcome to PySwarms's documentation! – PySwarms 1.3.0 documentation*. Retrieved June 29, 2025, from <https://pyswarms.readthedocs.io/en/latest/>
22. spotpython. (n.d.). *GitHub Pages*. Retrieved June 29, 2025, from <https://sequential-parameter-optimization.github.io/spotPython/>
23. Joel, L. O., Doorsamy, W., & Paul, B.S. (2022). A review of missing data handling techniques for machine learning. *International Journal of Innovative Technology and Interdisciplinary Sciences*, 5(3), 971–1005. <https://doi.org/10.1515/IJITIS.2022.5.3.971-1005>
24. Dekking, F. M., Kraaikamp, C., Lopuhaä, H. P., & Meester, L. E. (2005). *A modern introduction to probability and statistics*. Springer London. <https://doi.org/10.1007/1-84628-168-7>
25. Bringing clarity to investment decisions | MSCI. (2021, February). Retrieved June 29, 2025, from https://www.msci.com/eqb/methodology/meth_docs/MSCI_GIMIVGMethod_Feb2021.pdf
26. RobustScaler. (n.d.). *scikit-learn*. Retrieved June 29, 2025, from <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html>
27. Chakraborty, D., & Elzarka, H. (2018). Advanced machine learning techniques for building performance simulation: A comparative analysis. *Journal of Building Performance Simulation*, 12(2), 193–207. <https://doi.org/10.1080/19401493.2018.1498538>
28. CyclicalFeatures – 1.8.3. (n.d.). *Feature-engine – 1.8.3*. Retrieved June 29, 2025, from https://feature-engine.trainindata.com/en/1.8.x/user_guide/creation/CyclicalFeatures.html
29. Ul Haq, I., Haq, Q. N., Gondal, T. M., Khan, M. K., & Alarfi, I. M. (2019). Categorical features transformation with compact one-hot encoder for fraud detection in distributed environment. In *Communications in Computer and Information Science* (pp. 69–80). Springer. https://doi.org/10.1007/978-981-13-6661-1_6
30. Notes on parameter tuning – xgboost 3.0.2 documentation. (n.d.). *XGBoost Documentation – xgboost 3.0.2 documentation*. Retrieved June 29, 2025, from https://xgboost.readthedocs.io/en/stable/tutorials/parameter_tuning.html#handle-imbalanced-dataset
31. XGBoost parameters – xgboost 3.1.0-dev documentation. (n.d.). *XGBoost Documentation – xgboost 3.0.2 documentation*. Retrieved June 29, 2025, from <https://xgboost.readthedocs.io/en/latest/parameter.html>
32. Le, N., Phan, T., Nguyen, L., Le, D., & Nguyen, V. (2024). K-fold cross-validation: An effective hyperparameter tuning technique in machine learning on GNSS time series for movement forecast. In *Recent Research on Geotechnical Engineering, Remote Sensing, Geophysics and Earthquake Seismology* (pp. 377–382). Springer. https://doi.org/10.1007/978-3-031-43218-7_88
33. Soper, D. S. (2021). Greed is good: Rapid hyperparameter optimization and model selection using greedy k-fold cross validation. *Electronics*, 10(16), 1973. <https://doi.org/10.3390/electronics10161973>

34. Awwalu, J., & Ogwueleka, F. (2019). On holdout and cross validation: A comparison between neural network and support vector machine. *International Journal of Trend in Research and Development*, 6(2), 235–239.
35. Ong, E., Wang, H., Wong, M.U., Sommer, M. O. A., & He, Y. (2020). Vaxign-ML: Supervised machine learning reverse vaccinology model for improved prediction of bacterial protective antigens. *Bioinformatics*, 36(10), 3185–3191. <https://doi.org/10.1093/bioinformatics/btaa119>
36. Särndal, C.-E., Swensson, B., & Wretman, J. (1992). *Model assisted survey sampling*. Springer. <https://doi.org/10.1007/978-1-4612-4378-6>
37. Gupta, N., Sharma, D., Garg, S., & Singh, P. (2024, September 27–29). Heart disease prediction evaluation of machine learning models with PSO-optimized K-fold cross-validation. *2024 IEEE Region 10 Symposium (TENSYP)*, New Delhi, India (pp. 1–6). <https://doi.org/10.1109/tensymp61132.2024.10752215>
38. Sofaer, H. R., Hoeting, J. A., & Jarnevich, C. S. (2019). The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*, 10(4), 565–577. <https://doi.org/10.1111/2041-210x.13140>
39. Shenouda, J., & Bajwa, W.U. (2023). A guide to computational reproducibility in signal processing and machine learning [Tips & Tricks]. *IEEE Signal Processing Magazine*, 40(2), 141–151. <https://doi.org/10.1109/msp.2022.3217659>