

## 151 АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

УДК 004.852:004.891:338.27

DOI: 10.31498/2225-6733.53.2.2026.359931

### ПРОГНОЗУВАННЯ УСПІШНОСТІ В2В-ЗАМОВЛЕНЬ В УМОВАХ НЕСТАБІЛЬНОСТІ: ОЦІНКА СТІЙКОСТІ АРХІТЕКТУР МАШИННОГО НАВЧАННЯ ДО КОНЦЕПТУАЛЬНОГО ДРЕЙФУ

**Мірошниченко  
С.О.***аспірант, ст. викладач, ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ  
ПОЛІТЕХНІКА», м. Запоріжжя, ORCID: <https://orcid.org/0009-0007-4868-3006>, e-mail:  
[serhii.miroshnychenko@mipolytech.education](mailto:serhii.miroshnychenko@mipolytech.education)*

Прогнозування успішності В2В-замовлень є критичним завданням у системах планування ресурсів підприємства (ERP). Традиційні моделі пакетного навчання демонструють стійку вразливість до концептуального дрейфу, спричиненого макроекономічними збуреннями або зміною поведінкових патернів контрагентів. У статті порівнюється прогностична ефективність алгоритмів статичного пакетного та потокового навчання на масиві з 86 786 В2В-транзакцій реальної промислової ERP-системи. Бенчмаркінг охопив 15 алгоритмів: оптимізовані ансамблі дерев (XGBoost, LightGBM, Random Forest), лінійні та імовірнісні моделі, а також інкрементальні потокові архітектури, зокрема Adaptive Random Forest з екосистеми River. Оцінювання проводилось у двох режимах: глобальне тестування на стаціонарних вибірках та преквенційна симуляція для відтворення структурних зсувів у даних. Окремо досліджено вплив порядкового та циклічного кодування часових ознак. Встановлено, що циклічне кодування систематично знижує роздільну здатність алгоритмів на базі дерев рішень через ускладнення ортогональних розбиттів; порядкове кодування є оптимальним для таких архітектур. За стаціонарних умов пакетні моделі демонструють вищу глобальну точність (XGBoost, ROC-AUC = 0,9039). Однак під час симуляції макроекономічних збурень дискретність перенавчання знижує Recall пакетної моделі до ~60%, а частку хибних класифікацій підвищує до 35–36%. Потокова модель River ARF\_Preferential завдяки потранзакційній адаптації утримує частку помилок у межах 14–17% та Recall > 0,95. Результати підтверджують наявність компромісу між точністю пакетних і реактивністю потокових моделей, що обґрунтовує розробку гібридних прогностичних архітектур для промислових ERP.

**Ключові слова:** машинне навчання; прогнозування успішності; потокове навчання; концептуальний дрейф; ансамблі дерев рішень; преквенційна симуляція; ERP-системи; обробка часових ознак.

#### Постановка проблеми

Інтеграція ERP-систем з методами машинного навчання відкриває можливості для переходу від ретроспективного аналізу до предиктивного управління продажами. Одним із ключових завдань у цьому контексті є прогнозування успішності замовлень – завчасна ідентифікація ризику відмови або збиткового закриття угоди.

Практичне впровадження класичних ML-алгоритмів стикається з фундаментальною проблемою деградації прогностичної здатності в часі внаслідок концептуального дрейфу. Моделі пакетного навчання оптимізуються на накопиченій історичній вибірці і потребують повного перенавчання при зміні статистичних характеристик вхідних даних. Хоча системи моніторингу дрейфу дозволяють виявляти момент зсуву, перенавчання залишається ресурсомістким і вносить часовий лаг між появою нових патернів і оновленням прогностичної логіки. В умовах зовнішніх макроекономічних збурень або швидкої зміни поведінки контрагентів цей лаг призводить до суттєвого зростання кількості хибних рішень.

Альтернативою є потокові архітектури, здатні оновлюватися інкрементально після кожної нової транзакції. Окремим відкритим питанням залишається вибір методу кодування часових ознак, що може суттєво впливати на стійкість моделей за умов збурень вхідного розподілу. Дослідження цих факторів у промисловому ERP-контексті становить предмет цієї роботи.

#### Аналіз останніх досліджень та публікацій

Сучасні дослідження у сфері предиктивної аналітики В2В-транзакцій охоплюють три відносно ізольовані напрями: бенчмаркінг алгоритмів пакетного навчання, адаптація до концептуального дрейфу методами потокового навчання та кодування часових ознак.

Ансамблеві методи на базі дерев рішень (XGBoost, LightGBM, Random Forest) стабільно перевершують лінійні та метричні алгоритми на гетерогенних табличних даних ERP-систем [1, 2] і визнані стандартом для скорингу В2В-лідів [3]. Їхнім принциповим обмеженням є припущення стаціонарності розподілу ознак. Автори [4] емпірично показали, що за умов зовнішніх макроекономічних збурень точність таких

моделей різко знижується, а регулярне перенавчання стає запізнілим і ресурсомістким.

Для подолання концептуального дрейфу дослідники звертаються до методів потокового навчання, які обробляють транзакції послідовно й оновлюють параметри інкрементально без збереження повної історичної вибірки [5]. Базовим інструментом є Дерева Хефдінга [6]; їхнім розвитком – Adaptive Random Forest (ARF), який поєднує детектор дрейфу ADWIN з механізмом заміни застарілих дерев фоновими [7]. Стандартом реалізації потокових конвеєрів у Python є бібліотека River [8]. Валідація таких моделей здійснюється через преквенційну оцінку: кожен запис спочатку тестує модель, а потім використовується для оновлення її параметрів, що унеможливило витік даних [9, 10].

Часові зміни є невід’ємним компонентом автоматизованих конвеєрів прогнозування [11], проте їх стандартне порядкове представлення проєктує циклічні величини на скалярну вісь, створюючи штучний розрив між останнім і першим значенням циклу [12]. Циклічне кодування через  $\sin/\cos$ -перетворення зберігає метричну неперервність [13]. Однак алгоритми на базі дерев рішень будують ортогональні границі розбиття, тому апроксимація кола в просторі  $\sin/\cos$  вимагає складних ієрархічних структур і може погіршувати якість моделі [12, 14]. Автори [15] зафіксували зростання точності ансамблів при синусоїдальному кодуванні на 12,6%, проте дослідження охоплювало лише енергетичні часові ряди без включення потокових алгоритмів.

У сучасних дослідженнях відсутні уніфіковані порівняльні експерименти, що одночасно охоплюють широкий спектр пакетних і потокових алгоритмів, обидва методи кодування часових ознак та преквенційну симуляцію на реальних ERP-даних великого масштабу. Вивчення того, як методи кодування часу здатні пом’якшувати агресивний концептуальний дрейф в індустриальних конвеєрах прогнозування замовлень, є фундаментальною метою та обґрунтуванням актуальності запропонованої роботи.

### Мета статті

Метою дослідження є комплексне порівняння ефективності пакетних і потокових архітектур машинного навчання у задачі бінарної класифікації успішності B2B-замовлень на реальних ERP-даних з акцентом на методи кодування часових ознак і стійкість до концептуального дрейфу.

Для досягнення цієї мети вирішувались такі завдання:

- оцінка впливу порядкового та циклічного кодування часових ознак на точність пакетних і потокових алгоритмів;
- проведення преквенційної симуляції виробничого середовища для порівняння динамічної адаптивності алгоритмів в умовах зовнішніх макроекономічних збурень;

– визначення меж ефективності онлайн-алгоритмів River порівняно з пакетними моделями для задачі безперервного ERP-скорингу.

### Матеріали та методи

Емпіричну базу становлять 86 786 B2B-транзакцій реальної промислової ERP-системи. Цільова змінна – бінарний маркер успішності замовлення (`is_successful`): клас «1» відповідає закритій угоді, клас «0» – скасуванню або критичній втраті маржинальності.

Предиктори охоплюють кілька категорій ознак. Фінансові показники включають суму замовлення, середню ціну позиції та історичні метрики контрагента: загальний середній чек, середній чек успішних і неуспішних угод. Кількісні характеристики представлені обсягом і кількістю унікальних позицій та кількістю рядків чеку. Комунікаційні та операційні метрики відображають кількість повідомлень і змін у замовленні як на рівні окремої угоди, так і в розрізі контрагента. Надійність контрагента формалізована через частку успішних угод, загальну кількість замовлень і тривалість співпраці в днях. Часові метрики включають інтервал від попереднього замовлення та мітку часу створення транзакції. Пропущені значення інтервальних метрик імпутовані значенням «-1»; для фіксації першої взаємодії клієнта введено бінарний прапорець `is_first_order`.

На основі наявних предикторів згенеровано додаткові розрахункові ознаки: відношення суми замовлення до середнього чеку контрагента, вартість на рядок чеку, коефіцієнт інтенсивності обговорення та відношення середньої ціни продукту до середнього чеку контрагента.

Часові ознаки (година, день тижня, місяць і день місяця) кодувались двома альтернативними способами. Порядкове кодування представляє компоненти дати як скалярні змінні, що спричиняє штучний розрив на стику циклу: відстань між 23:00 та 00:00 у просторі ознак є максимальною, попри їх хронологічну суміжність. Циклічне кодування усуває цю проблему через тригонометричне відображення за формулою  $\sin(2\pi \cdot t/T)$  та  $\cos(2\pi \cdot t/T)$ , де  $T \in \{24, 7, 12, 31\}$ , зберігаючи метричну неперервність на межах циклу. Обчислення реалізовано засобами бібліотеки NumPy для забезпечення сумісності з обома класами алгоритмів – пакетними ансамблями та потоковими конвеєрами.

До бенчмаркінгу залучено п’ятнадцять алгоритмів, розподілених на дві групи. Пакетна група охоплює одинадцять моделей: ансамблі на базі дерев рішень (XGBoost, LightGBM, Random Forest, Bagging, AdaBoost, Decision Tree), а також MLP, SVM, KNN, Naive Bayes та Logistic Regression. Потокова група представлена чотирма алгоритмами екосистеми River: Hoeffding Tree, Hoeffding Tree Prequential, Adaptive Random Forest та ARF Prequential.

Глобальний бенчмаркінг здійснювався на хронологічно розподіленому масиві: 70% записів – навчальна вибірка, по 15% – валідаційна і тестова. Моделі оцінювались за показниками ROC-AUC, F1, Accuracy, Precision, Recall, часткою хибних класифікацій і часом навчання.

Преквенційна симуляція охоплювала 75 629 транзакцій за період з липня 2018 по листопад 2024 року. До симуляції відібрано XGBoost та River ARF\_Prequential як моделі з найвищим ROC-AUC у своїх класах. Обидві моделі ініціалізовані на масиві даних тривалістю один рік. Надалі XGBoost формував прогнози протягом кожного місяця, після чого перенавчався на всій накопиченій вибірці. River ARF\_Prequential після кожної транзакції спочатку формував прогноз, а потім інкрементально оновлював параметри без доступу до повної історії. Такий дизайн відтворює реальні умови промислового скорингу і дозволяє об'єктивно порівняти поведінку архітектур в

умовах збурення характеристик вхідного потоку даних.

**Виклад основного матеріалу**

Оцінювання продуктивності на стаціонарній вибірці підтверджує перевагу ансамблевих архітектур над лінійними та метричними класифікаторами. За показниками Accuracy (рис. 1) і Precision (рис. 2) найвищі результати серед усіх 15 алгоритмів показав River\_ARF\_Prequential (0,9074 і 0,8994 відповідно) – однак його перевага над пакетними моделями частково зумовлена преквенційним режимом оновлення безпосередньо під час тестування. Серед суто пакетних алгоритмів лідирують Random Forest (Accuracy = 0,8788), LightGBM (0,8627) та XGBoost (0,8589). Традиційні класифікатори – SVM (0,7732), NaiveBayes (0,7617), Logistic Regression (0,6679) – суттєво поступаються за обома показниками.

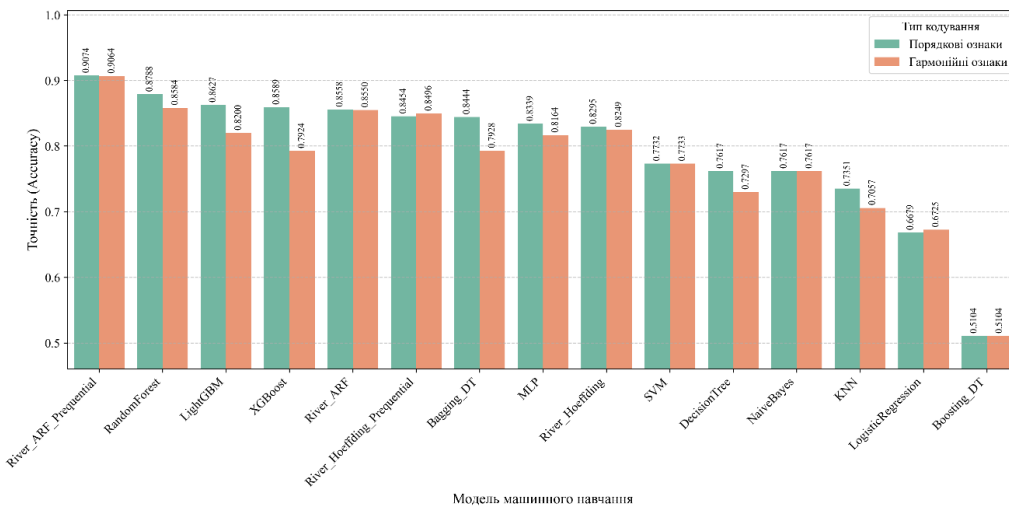


Рис. 1 – Показники моделей: точність (Accuracy)

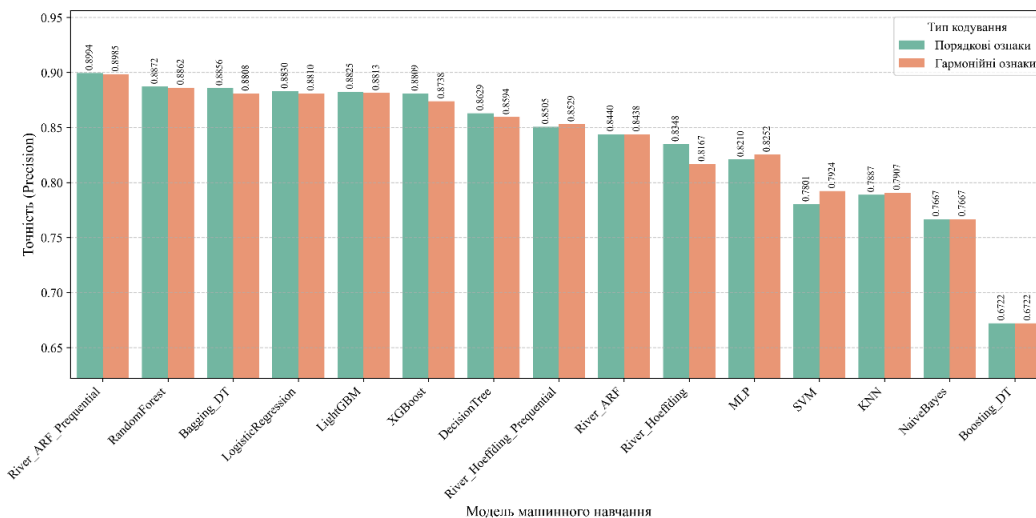


Рис. 2 – Показники моделей: точність (Precision)

Значення метрики Recall розподіляються інакше (рис. 3): найвищу повноту демонструють MLP (0,985) та потокові моделі River ARF (0,982) і ARF\_Prequential (0,9817). Класичні пакетні ансамблі мають помірніші значення: Random Forest – 0,954, LightGBM – 0,934, XGBoost – 0,931. Висока повнота MLP, NaiveBayes (0,9634) і SVM (0,9555) досягається за рахунок підвищеної кількості хибнопозитивних рішень, що знижує їх точність (Precision) до 0,76-0,82 (рис. 2). Потокові архітектури зберігають баланс між обома цими метриками: River\_ARF\_Prequential поєднує Recall 0,9817 з Precision 0,8994.

Інтегральний показник F1-міри (рис. 4) найвищий у River\_ARF\_Prequential (0,9388), далі – Random Forest (0,9192), River\_ARF (0,9078), LightGBM (0,9077), XGBoost (0,9050). За ROC-AUC аналогічна ієрархія

(рис. 5): River\_ARF\_Prequential (0,9482), River\_ARF (0,9143), XGBoost (0,9039), LightGBM (0,9031), Random Forest (0,9024). Найнижчі показники у Boosting\_DT (ROC-AUC = 0,538), що відповідає рівню випадкової класифікації. Частка хибних класифікацій (рис. 6) становить: River\_ARF\_Prequential – 9,3%, Random Forest – 12,1%, LightGBM – 13,7%, XGBoost – 14,1%, тоді як SVM – 22,7%, KNN – 26,5%, Logistic Regression – 33,2%, Boosting\_DT – 49,0%. З точки зору обчислювальних витрат (рис. 7) SVM є непридатним для регулярного перенавчання (1050-1375 с). Найшвидші пакетні алгоритми – XGBoost (~1,8 с), LightGBM (2,6 с), Random Forest (4,7 с). Потокові River\_ARF витрачають 424-432 с через послідовну обробку записів; River\_Hoeffding\_Prequential – лише 18,3 с.

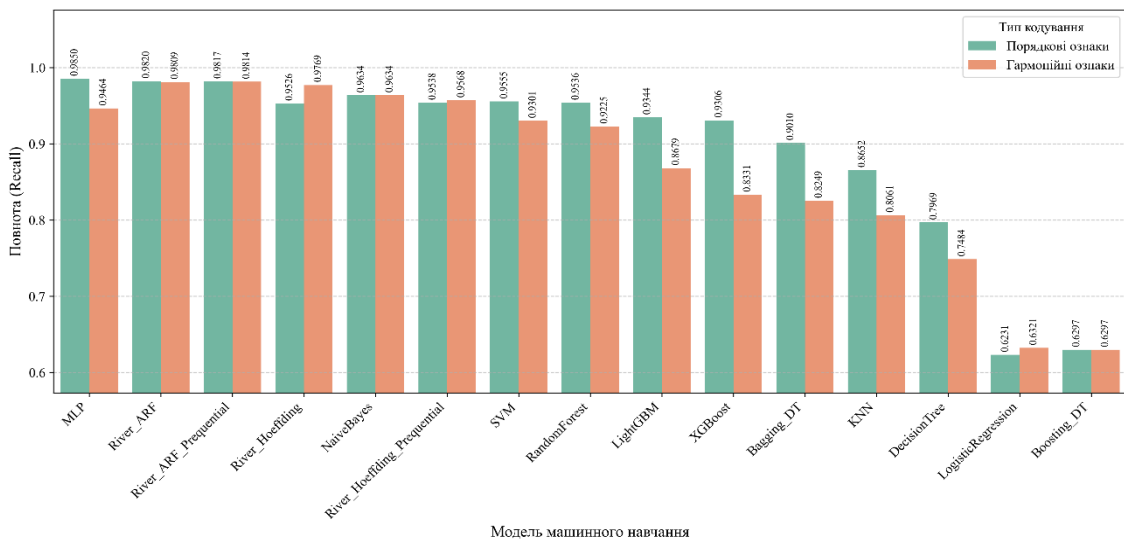


Рис. 3 – Показники моделей: повнота (Recall)

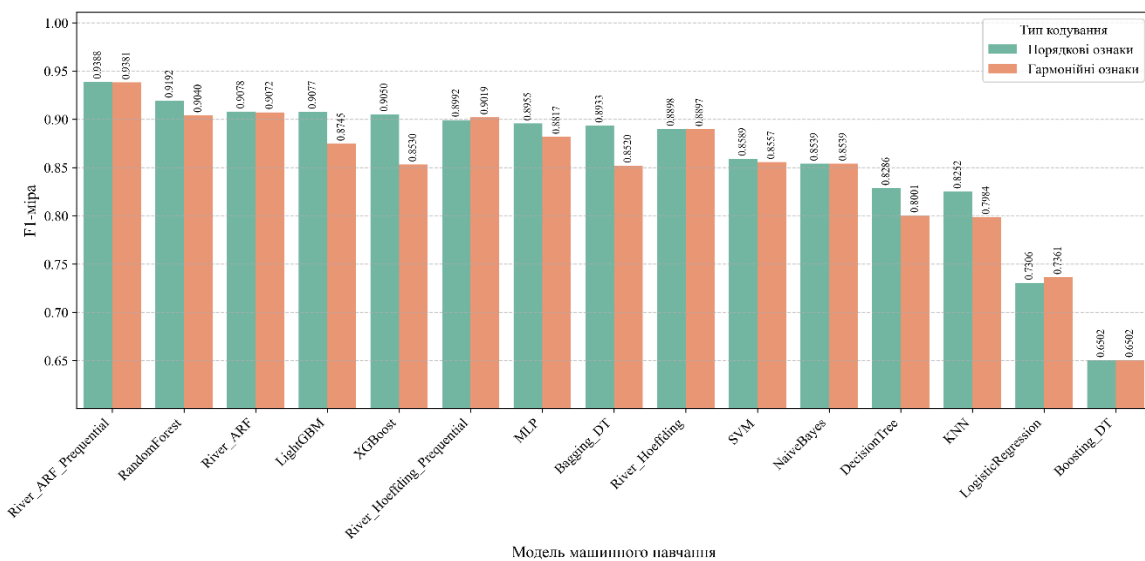


Рис. 4 – Показники моделей: F1-міра

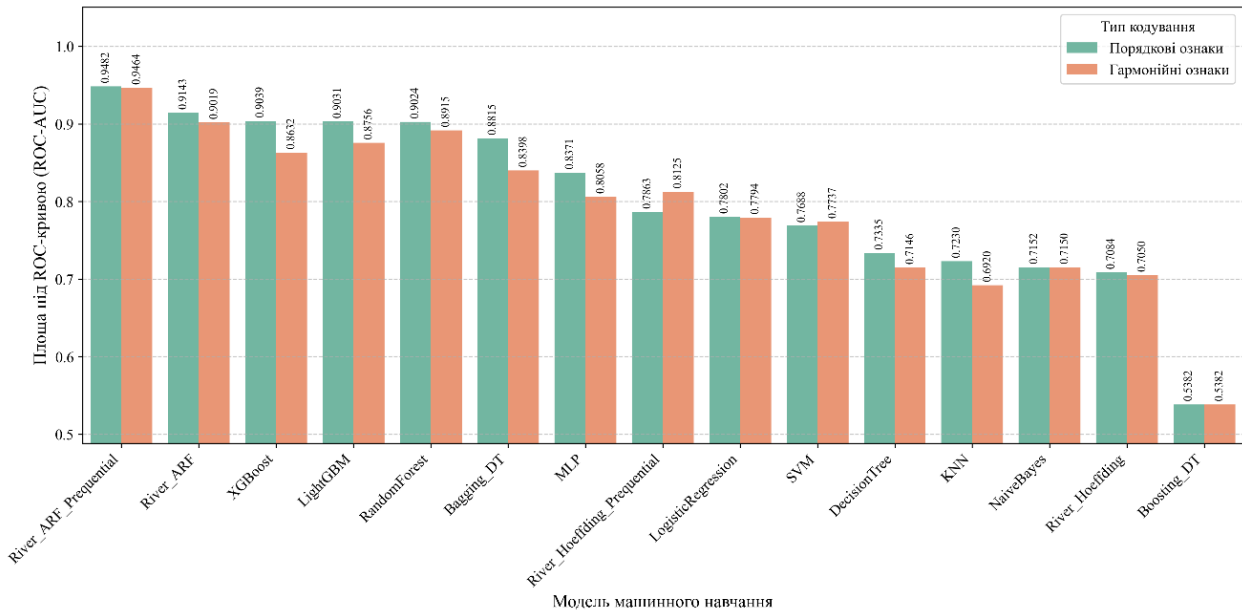


Рис. 5 – Показники моделей: площа під ROC-кривою (ROC-AUC)

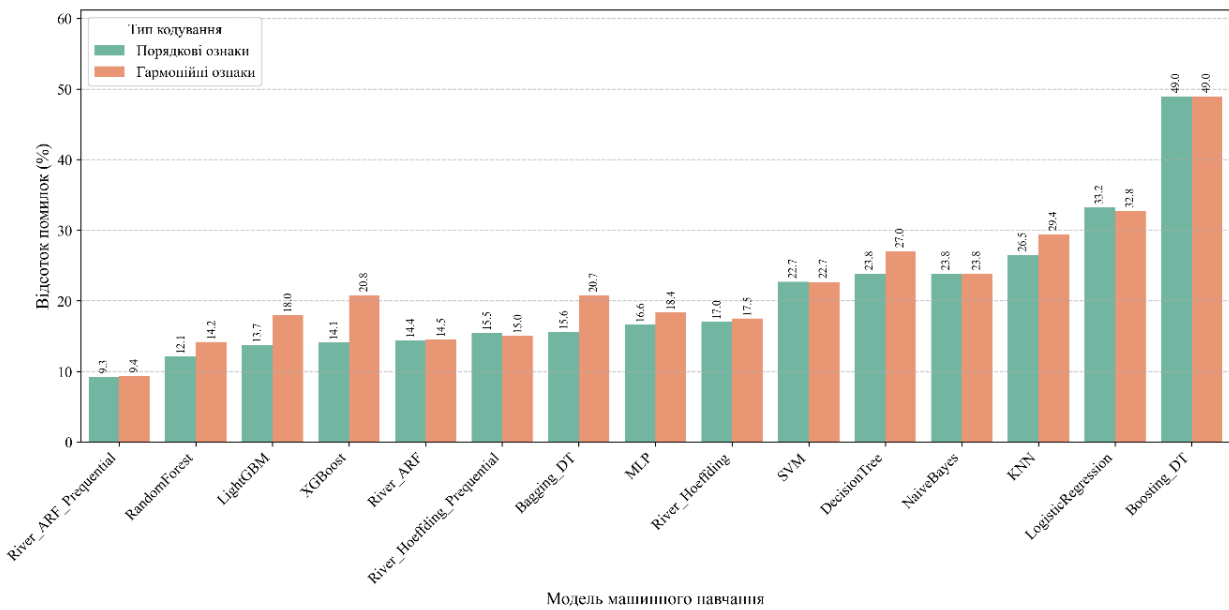


Рис. 6 – Показники моделей: відсоток помилок

Результати тестування виявили гетерогенну реакцію алгоритмів на спосіб кодування часових ознак (рис. 4, 5). Для архітектур на базі дерев рішень циклічне кодування систематично знижує прогностичну якість: ROC-AUC XGBoost падає з 0,9039 до 0,8632, LightGBM – з 0,9031 до 0,8756, Random Forest – з 0,9024 до 0,8915, Bagging\_DT – з 0,8815 до 0,8398; F1 XGBoost знижується з 0,9050 до 0,8530. Механізм цього ефекту пов'язаний з ортогональним характером розбиттів дерев рішень: апроксимація тригонометрично заданих просторів вимагає надмірно глибоких

ієрархічних структур, що підвищує схильність моделей до перенавчання.

Реакція окремих архітектур є протилежною (рис. 3, 4): F1 River\_Hoeffding\_Prequential незначно зростає з 0,8992 до 0,9019, Recall базового River\_Hoeffding підвищується з 0,9526 до 0,9769. Показники NaiveBayes і Boosting\_DT повністю інваріантні до типу кодування (рис. 1-6). Таким чином, для ансамблів дерев рішень, що становлять основу сучасного ERP-скорингу, порядкове кодування є кращим вибором; циклічне кодування може бути корисним лише для окремих потокових або імовірнісних архітектур.



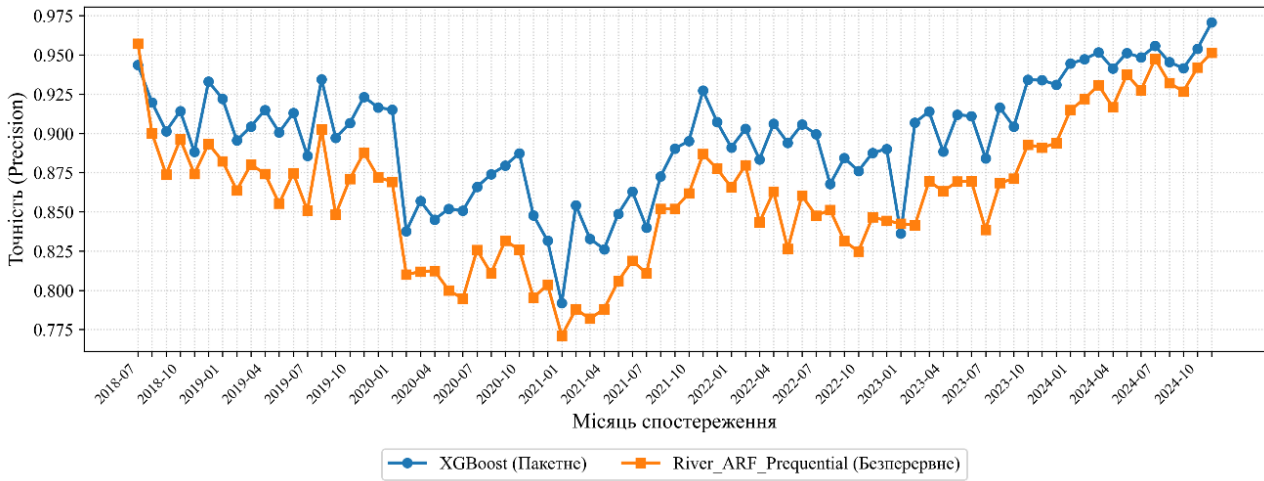


Рис. 9 – Динаміка зміни показника: точність (Precision)



Рис. 10 – Динаміка зміни показника: повнота виявлення (Recall)

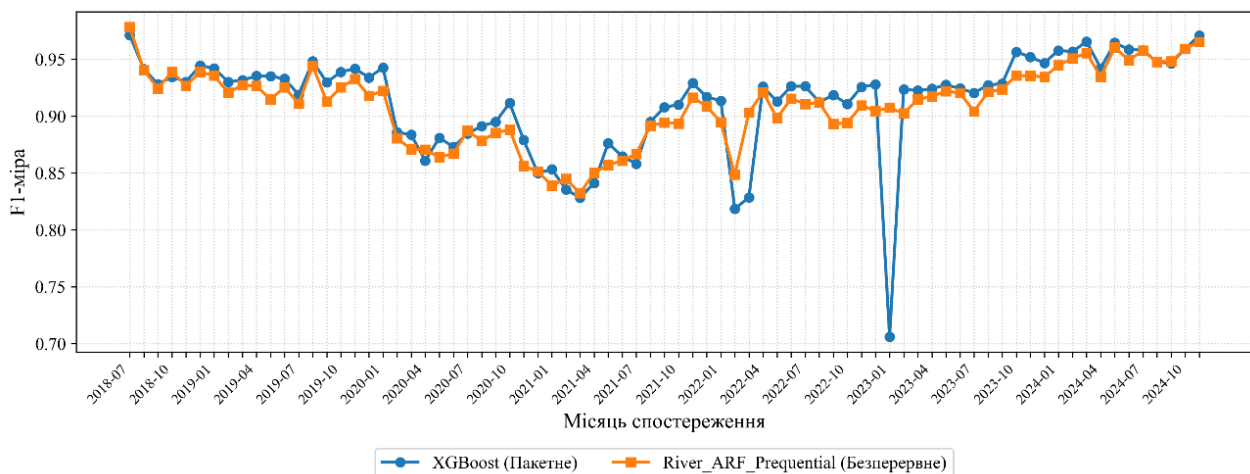


Рис. 11 – Динаміка зміни показника: F1-міра

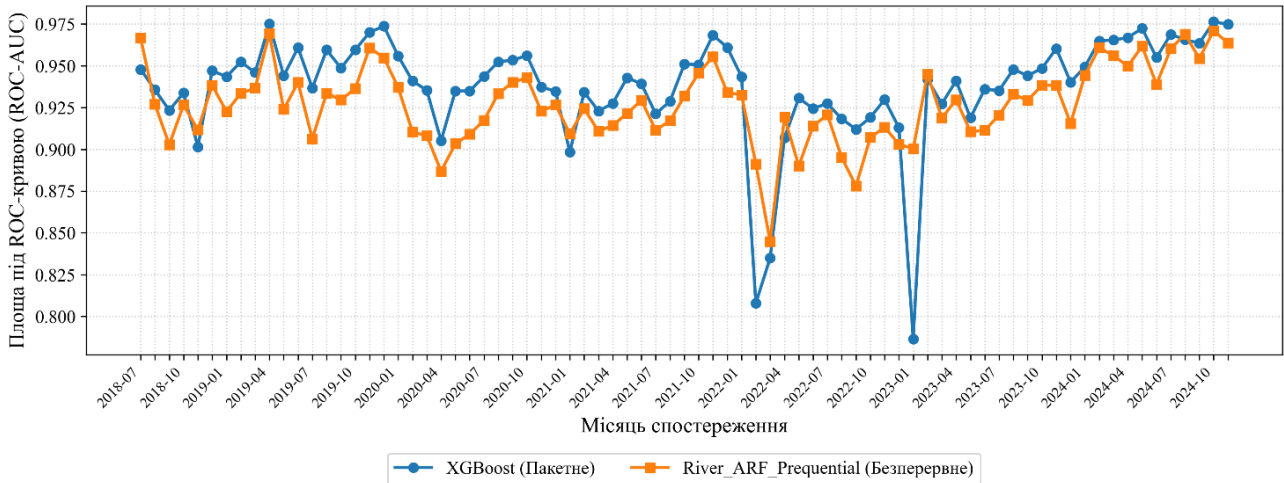


Рис. 12 – Динаміка зміни показника: площа під ROC-кривою (ROC-AUC)

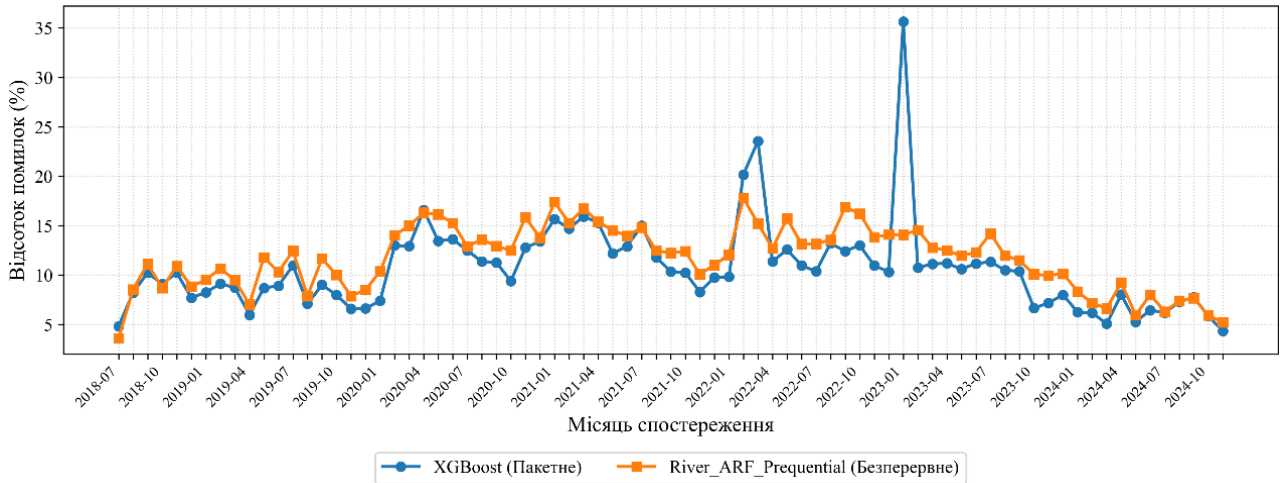


Рис. 13 – Динаміка зміни показника: відсоток помилок

Найбільш виражене збурення вхідного розподілу зафіксовано на початку 2023 року (рис. 8-13). XGBoost зазнав суттєвої деградації: Accuracy впала до 0,65, Recall – до 0,61, F1 – до 0,70, ROC-AUC знизився з 0,93 до 0,78-0,79, частка помилок досягла 35-36%. Модель хибно відхиляла близько 40% потенційно успішних замовлень. Механізм деградації полягає у тому, що XGBoost продовжував класифікувати нові транзакції відповідно до застарілих статистичних закономірностей аж до чергового циклу перенавчання. River\_ARF\_Prequential за той самий період утримував Accuracy на рівні 0,85-0,87, Recall > 0,95, F1 ≈ 0,90-0,91, частку помилок 14–15% завдяки безперервному потранзакційному оновленню структури ансамблю. Після згасання збурення (кінець 2023-2024 р.) обидві моделі відновили показники до рівня F1 ≈ 0,95-0,96.

Отримані результати засвідчують, що перевага пакетної архітектури за інтегральними метриками формується переважно в стабільні періоди. В умовах

збурення характеристик вхідного потоку XGBoost генерує 35-36% хибних рішень проти 14-15% у потокової моделі. Високий показник Precision пакетного підходу досягається ціною масового хибного відхилення легітимних угод, тоді як потокова архітектура забезпечує прийнятну якість рішень упродовж усього горизонту спостереження, включно з періодами збурень

**Висновки**

У дослідженні проведено порівняльний аналіз ефективності пакетних і потокових архітектур машинного навчання на задачі скорингу B2B-транзакцій у реальному ERP-середовищі.

Щодо впливу методів кодування часових ознак, емпірично підтверджено, що циклічне sin/cos-кодування систематично знижує якість алгоритмів на базі дерев рішень через несумісність тригонометричних просторів з ортогональними розбиттями. Для ансамблів XGBoost, LightGBM і Random Forest оптимальним є порядкове кодування. Незначний приріст від

циклічного кодування можливий лише для окремих потокових архітектур типу River\_Hoeffding; імовірнісні класифікатори інваріантні до типу кодування.

За результатами глобального тестування встановлено ієрархію ефективності моделей. Серед пакетних алгоритмів найвищу дискримінаційну здатність показали XGBoost (ROC-AUC = 0,9039), LightGBM (0,9031) та Random Forest (0,9024) із часткою помилок 12–14%. Лінійні класифікатори та слабко налаштовані дерева виявились непридатними для промислового скорингу, генеруючи 33–49% хибних рішень. Серед потокових алгоритмів River\_ARF\_Prequential показав ROC-AUC = 0,9482, F1 = 0,9388 та лише 9,3% помилок.

Преквенційна симуляція виявила вразливість пакетних архітектур до збурень вхідного розподілу даних. Під час структурного збурення 2023 року Recall XGBoost знизився до ~60%, частка помилок зросла до 35–36%, що відповідало хибному відхиленню ~40% легітимних угод. Натомість River\_ARF\_Prequential утримував Recall > 0.95 та частку помилок 14–17% завдяки потранзакційному оновленню структури ансамблю.

Отримані результати підтверджують об'єктивний компроміс між максимальною точністю пакетних ансамблів за стаціонарних умов та адаптивністю потокових моделей в умовах збурень. Перспективним напрямом подальших досліджень є розробка гібридних архітектур, що поєднують розрізнявальну потужність градієнтних бустингів з інкрементальними механізмами потокового навчання задля забезпечення стійкого скорингу за будь-яких умов функціонування системи

#### Перелік використаних джерел

- [1] Yang Y., Wang H. Random Forest-Based Machine Failure Prediction: A Performance Comparison. *Applied Sciences*. 2025. Vol. 15, no. 16. Article 8841. DOI: <https://doi.org/10.3390/app15168841>.
- [2] Abd El-Aziz R. A., Hassan A. S. 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. DOI: <https://doi.org/10.1186/s43088-023-00460-y>.
- [3] Machine Learning and Deep Learning Models for Demand Forecasting in Supply Chain Management: A Critical Review / K. Douaioui et al. *Applied System Innovation*. 2024. Vol. 7, no. 5. Article 93. DOI: <https://doi.org/10.3390/asi7050093>.
- [4] Kraus A., van der Aa H. Machine learning-based detection of concept drift in business processes. *Process Science*. 2025. Vol. 2, no. 1. DOI: <https://doi.org/10.1007/s44311-025-00012-w>.
- [5] Suárez-Cetrulo A. L., Quintana D., Cervantes A. A survey on machine learning for recurring concept drifting data streams. *Expert Systems with Applications*. 2022. Volume 213, Part A. Article 118934. DOI: <https://doi.org/10.1016/j.eswa.2022.118934>.
- [6] Domingos P., Hulten G. Mining high-speed data streams. *Knowledge discovery and data mining: Proceedings of the sixth ACM SIGKDD international conference, Boston, Massachusetts, United States, 20–23 August 2000*. Pp. 71–80. DOI: <https://doi.org/10.1145/347090.34710>.
- [7] Adaptive random forests for evolving data stream classification / H. M. Gomes et al. *Machine Learning*. 2017. Vol. 106, no. 9–10. Pp. 1469–1495. DOI: <https://doi.org/10.1007/s10994-017-5642-8>.
- [8] River: machine learning for streaming data in Python / J. Montiel et al. *Journal of Machine Learning Research*. 2021. Vol. 22, no. 110. Pp. 1–8.
- [9] S. B. Jadhav, D. V. Kodavade, V. D. Kulkarni. Data Stream Learning Evaluation: Experimenting with Prequential Approach Over Real Data Streams. *Journal of Information Systems Engineering and Management*. 2025. Vol. 10, no. 9s. Pp. 230–237. DOI: <https://doi.org/10.52783/jisem.v10i9s.1206>.
- [10] Machine learning for streaming data / H. M. Gomes et al. *ACM SIGKDD Explorations Newsletter*. 2019. Vol. 21, no. 2. Pp. 6–22. DOI: <https://doi.org/10.1145/3373464.3373470>.
- [11] Review of automated time series forecasting pipelines / S. Meisenbacher et al. *WIREs Data Mining and Knowledge Discovery*. 2022. Vol. 12, no. 6, p. e1475. DOI: <https://doi.org/10.1002/widm.1475>.
- [12] Lewinson E. Three Approaches to Encoding Time Information as Features for ML Models. *NVIDIA Technical Blog*. 2022. URL: <https://developer.nvidia.com/blog/three-approaches-to-encoding-time-information-as-features-for-ml-models/> (дата звернення: 20.02.2026).
- [13] Chakraborty D., Elzarka H. Advanced machine learning techniques for building performance simulation: a comparative analysis. *Journal of Building Performance Simulation*. 2019. Vol. 12, No. 2. Pp. 193–207. DOI: <https://doi.org/10.1080/19401493.2018.1498538>.
- [14] Mahajan T., Singh G., Bruns G. An Experimental Assessment of Treatments for Cyclical Data. *Computer Science Conference for CSU Undergraduates, California State University Northridge*, 2021. Pp. 1–5.
- [15] Bansal A., Balaji K., Lalani Z. Temporal Encoding Strategies for Energy Time Series Prediction. arXiv:2503.15456. 2020. Preprint. DOI: <https://doi.org/10.48550/arXiv.2503.15456>.

BATCH VS. STREAMING MACHINE LEARNING FOR B2B ORDER SUCCESS FORECASTING:  
ROBUSTNESS TO CONCEPT DRIFT IN INDUSTRIAL ERP SYSTEMSMiroshnychenko  
S.O.postgraduate student, senior lecturer, TECHNICAL UNIVERSITY "METINVEST  
POLYTECHNIC" LLC, Zaporizhzhia, ORCID: <https://orcid.org/0009-0007-4868-3006>, e-  
mail: [serhii.miroshnychenko@mipolytech.edu](mailto:serhii.miroshnychenko@mipolytech.edu)

Predicting B2B order success is a critical task in Enterprise Resource Planning (ERP) systems, directly affecting operational efficiency and commercial outcomes. Traditional batch machine learning models assume stationarity of the underlying data distribution, rendering them vulnerable to concept drift arising from macroeconomic disturbances or shifts in counterparty behaviour. This study compares static batch and continuous streaming machine learning architectures for B2B order success classification on a dataset of 86,786 transactions from an operational industrial ERP system. Fifteen algorithms were benchmarked: optimised tree ensembles (XGBoost, LightGBM, Random Forest), linear and probabilistic classifiers, and incremental streaming models from the River ecosystem, notably the Adaptive Random Forest. Evaluation combined a global stationary benchmark with a strictly prequential simulation reproducing structural data shifts under real production conditions. The impact of linear-ordinal versus harmonic-cyclic temporal feature encoding was assessed across both algorithm classes. Harmonic-cyclic encoding systematically degraded decision tree-based models due to incompatibility between trigonometric feature spaces and orthogonal split boundaries, confirming ordinal encoding as the optimal choice for such architectures. Under stationary conditions, XGBoost achieved the highest batch model ROC-AUC of 0.9039, while River ARF\_Prequential led overall with ROC-AUC 0.9482 and a misclassification rate of 9.3%. During the prequential simulation, XGBoost retained a marginal aggregate advantage; however, at the peak drift episode in early 2023, its Recall dropped to approximately 0.61 and misclassification rate reached 35–36%, implying erroneous rejection of nearly 40% of legitimate orders. River ARF\_Prequential sustained Recall above 0.95 and a misclassification rate of 14–15% throughout the same period through continuous incremental updates. The findings confirm a trade-off between the peak accuracy of batch ensembles under stationary conditions and the adaptive resilience of streaming models under distributional disturbances, motivating the development of hybrid forecasting architectures for industrial ERP pipelines.

**Keywords:** machine learning; success forecasting; streaming learning; concept drift; decision tree ensembles; prequential simulation; ERP systems; temporal feature engineering.

## References

- [1] M. N. Hasan et al., "Random Forest-Based Machine Failure Prediction: A Performance Comparison," *Applied Sciences*, vol. 15, no. 16, article 8841, 2025. doi: [10.3390/app15168841](https://doi.org/10.3390/app15168841).
- [2] R. A. Abd El-Aziz, and A. S. Hassan, "Machine learning-driven optimization of enterprise resource planning (ERP) systems: a comprehensive review," *Beni-Suef University Journal of Basic and Applied Sciences*, vol. 13, no. 1, 2024. doi: [10.3390/asi7050093](https://doi.org/10.3390/asi7050093).
- [3] G. Villegas-Ch, D. García-Ortiz, and S. Luján-Mora, "Machine Learning and Deep Learning Models for Demand Forecasting in Supply Chain Management: A Critical Review," *Applied System Innovation*, vol. 7, no. 5, article 93, 2024. doi: [10.3390/asi7050093](https://doi.org/10.3390/asi7050093).
- [4] A. Kraus, and H. van der Aa, "Machine learning-based detection of concept drift in business processes," *Process Science*, vol. 2, no. 1, 2025. doi: [10.1007/s44311-025-00012-w](https://doi.org/10.1007/s44311-025-00012-w).
- [5] A. L. Suárez-Cetrulo, D. Quintana, and A. Cervantes, "A survey on machine learning for recurring concept drifting data streams," *Expert Systems with Applications*, vol. 213, part A, article 118934, 2023. doi: [10.1016/j.eswa.2022.118934](https://doi.org/10.1016/j.eswa.2022.118934).
- [6] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. 6th ACM SIGKDD Int. Conf. «Knowledge Discovery and Data Mining»*, Boston, MA, USA, August 20–23, 2000 2000, pp. 71–80. doi: [10.1145/347090.347107](https://doi.org/10.1145/347090.347107).
- [7] H. M. Gomes et al., "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, no. 9–10, pp. 1469–1495, 2017. doi: [10.1007/s10994-017-5642-8](https://doi.org/10.1007/s10994-017-5642-8).
- [8] J. Montiel et al., "River: machine learning for streaming data in Python," *Journal of Machine Learning Research*, vol. 22, pp. 1–8, 2021.
- [9] S. B. Jadhav et al., "Data Stream Learning Evaluation: Experimenting with Prequential Approach Over Real Data Streams," *Journal of Information Systems Engineering & Management*, vol. 10, no. 9s, pp. 230–237, 2025. doi: [10.52783/jisem.v10i9s.1206](https://doi.org/10.52783/jisem.v10i9s.1206).
- [10] H. M. Gomes et al., "Machine learning for streaming data: state of the art, challenges, and opportunities," *SIGKDD Explorations*, vol. 21, no. 2, 2019. doi: [10.1145/3373464.3373470](https://doi.org/10.1145/3373464.3373470).
- [11] S. Meisenbacher et al., "Review of automated time series forecasting pipelines," *WIREs Data Mining and Knowledge Discovery*, vol. 12, no. 6, p. e1475, 2022. doi: [10.1002/widm.1475](https://doi.org/10.1002/widm.1475).
- [12] E. Lewinson, "Three Approaches to Encoding Time Information as Features for ML Models," *NVIDIA Technical Blog*, 2022. [Online]. Available: <https://developer.nvidia.com/blog/three-approaches-to-encoding-time-information-as-features-for-ml-models>.

Accessed: February 20, 2026.

- [13] D. Chakraborty and H. Elzarka, "Advanced machine learning techniques for building performance simulation: a comparative analysis," *Journal of Building Performance Simulation*, vol. 12, no. 2, pp. 193–207, 2019. doi: **10.1080/19401493.2018.1498538**.
- [14] T. Mahajan, G. Singh, and G. Bruns, "An Experimental Assessment of Treatments for Cyclical Data," in Proc. of the Computer Science Conference for CSU Undergraduates, California State University Northridge, 2021, pp. 1-5.
- [15] A. Bansal, K. Balaji, and Z. Lalani, "Temporal Encoding Strategies for Energy Time Series Prediction," 2020, arXiv:2503.15456. doi: **10.48550/arXiv.2503.15456**.

Стаття надійшла 26.12.2025  
Стаття прийнята 15.01.2026  
Стаття опублікована 26.03.2026

**Цитуйте цю статтю як:** Мірошніченко С. О. Прогнозування успішності B2B-замовлень в умовах нестабільності: оцінка стійкості архітектур машинного навчання до концептуального дрейфу. *Вісник Приазовського державного технічного університету. Серія: Технічні науки*. 2026. Вип. 53, том 2. С. 75–85. DOI: <https://doi.org/10.31498/2225-6733.53.2.2026.359931>.