



О. О. Гудь, Н. Е. Кунанець

Національний університет "Львівська Політехніка", м. Львів, Україна

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ РОБОТИ АЛГОРИТМІВ XGBOOST ТА LSTM У ЗАДАЧІ ПРОГНОЗУВАННЯ СПРОМОЖНОСТІ SCRUM-КОМАНД НА РІЗНИХ ГОРИЗОНТАХ

Обґрунтовано доцільність застосування технологій машинного навчання для підвищення точності планування ітерацій у методології Scrum. Виконано експериментальне дослідження на історичних даних реальної DevOps-команди проєкту за 32 завершені спринти, що містили показники продуктивності, календарні обмеження, відпустки, операційне навантаження та контекстні фактори. Проведено систематичний порівняльний аналіз двох класів моделей машинного навчання – градієнтного підсилення XGBoost та рекурентних нейронних мереж LSTM – для прогнозування командної спроможності на різних часових горизонтах. Застосовано методику покрокової валідації з метриками MAE, RMSE, MAPE та тестом Дібольда-Маріано для статистичного порівняння моделей. Встановлено, що модель XGBoost демонструє статистично значущу перевагу на короткотерміновому горизонті ($H = 1$ спринт) із середньою абсолютною похибкою MAE = 8,42 проти MAE = 12,35 для LSTM ($p = 0,005$), забезпечуючи зниження похибки на 31,8 % завдяки ефективному виконанню роботи з табличними ознаками та календарними обмеженнями. Виявлено відсутність статистично значущої різниці між моделями на середньо-терміновому горизонті ($H = 3$ спринти) з MAE = 14,67 для моделі XGBoost та MAE = 14,92 для LSTM ($p = 0,678$), що свідчить про перехідну зону між домінуванням експертно сформованих ознак й автоматично виявлених часових залежностей. Встановлено, що методика покрокової валідації забезпечує темпорально коректне оцінювання узагальнювальної здатності моделей та усуває систематичне оптимістичне зміщення, характерне для стандартного k-fold крос-валідування на часових рядах спринтів. З'ясовано, що нейронна мережа LSTM виявляє статистично значущу перевагу на довготерміновому горизонті ($H = 6$ спринтів) з MAE = 17,26 проти MAE = 21,48 для моделі XGBoost ($p = 0,031$), зменшуючи похибку на 19,6 % через здатність моделювати довготривалі часові залежності, інерційність команди проєкту та сезонні тренди. Результати обґрунтовують доцільність розроблення адаптивної метамоделі, що динамічно вибиратиме або комбінуватиме алгоритми градієнтного підсилення та рекурентних нейронних мереж залежно від горизонту прогнозування командної спроможності та контексту scrum-проєкту для підвищення загальної точності системи прогнозування.

Ключові слова: ітеративне планування обсягу робіт; часові ряди; ансамблеві методи; рекурентні нейронні мережі; покрокова валідація; тест Дібольда-Маріано; адаптивна метамодель.

Вступ / Introduction

Методологія управління проєктами Scrum є одним із найпоширеніших підходів до розроблення програмного забезпечення. Її ітеративна природа дає змогу командам проєкту швидко реагувати на зміни вимог, проте водночас ставить перед менеджерами проєкту складне завдання точного планування обсягу робіт спринту. За даними звіту Standish Group Chaos Report 2020 [11], тільки 31 % проєктів завершуються вчасно та в межах бюджету, тоді як 50 % перевищують планові терміни та кошториси. Однією з основних причин таких відхилень є недостатня точність прогнозування спроможності команд розробників, що призводить до нереалістичних зобов'язань та порушення термінів.

У світовій науковій літературі питання прогнозування командної спроможності у scrum-проєктах активно досліджують. У роботі [2] доведено переваги методів

машинного навчання над традиційними експертними оцінками. У дослідженні [8] продемонстровано ефективність використання рекурентних нейронних мереж LSTM для моделювання послідовностей спринтів. У науковій праці [6] показано високу точність градієнтних ансамблів XGBoost на структурованих наборах даних. У роботі [14] запропоновано метамодельний підхід для динамічного вибору алгоритмів.

Попри значний обсяг досліджень, відсутнє системне порівняння ефективності використання градієнтних методів та рекурентних нейронних мереж на різних часових горизонтах прогнозування командної спроможності за умов реальних проєктів. Не встановлено, за яких умов доцільно використовувати кожен із підходів, що ускладнює вибір раціонального алгоритму для практичного застосування. Це становить актуальність даного дослідження.

Об'єкт дослідження – процеси кількісного оціню-

© Copyright 2026 by the author(s)

Гудь Олександрич, аспірант, кафедра інформаційних систем та технологій.

Email: oleksa.o.hud@lpnu.ua; <https://orcid.org/0009-0007-4298-7332>

Кунанець Наталія Едуардівна, д-р пед. наук, професор, кафедра інформаційних систем та технологій.

Email: natalia.e.kunanets@lpnu.ua; <https://orcid.org/0000-0003-3007-2462>

Цитування за ДСТУ: Гудь О. О., Кунанець Н. Е. Порівняльний аналіз ефективності роботи алгоритмів XGBoost та LSTM у задачі прогнозування спроможності scrum-команд на різних горизонтах. Науковий вісник НЛТУ України. 2026, т. 36, № 1. С. 162–168.

Citation APA: Hud, O. O., & Kunanets, N. E. (2026). Comparative analysis of the efficiency of XGBoost and LSTM algorithms in the problem of forecasting the capacity of scrum teams at different horizons. *Scientific Bulletin of UNFU*, 36(1), 162–168. <https://doi.org/10.36930/40360118>

вання та планування обсягу робіт у гнучких ітеративних проєктах розроблення програмного забезпечення на основі історичних даних продуктивності роботи команд проєкту.

Предмет дослідження – методи і засоби моделювання часових залежностей і структурованих факторів продуктивності команд проєкту, які дадуть можливість встановити їх вплив на точність багатогоризонтного прогнозування спроможності scrum-команд за умов змінного проєктного контексту.

Мета роботи – визначити переваги та обмеження різних класів прогностичних моделей машинного навчання для оцінювання командної спроможності у scrum-проєктах на основі історичних даних виконання проєктних робіт з урахуванням часової структури, контекстних факторів та тривалості прогностичного інтервалу, що забезпечить науково обґрунтований вибір методу валідації та системи метрик оцінювання, адекватних умовам динамічного проєктного середовища з обмеженим обсягом вибірки.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

- проаналізувати сучасні наукові підходи до кількісного оцінювання результативності роботи команд проєкту у гнучкому ітеративному середовищі та визначити їхні переваги й обмеження, що дасть змогу формувати теоретичне підґрунтя для обґрунтованого вибору класу прогностичних моделей машинного навчання для оцінювання командної спроможності, придатних для застосування за умов обмеженого обсягу вибірки та концептуального дрейфу;
- розробити експериментальну методику порівняння роботи алгоритмів XGBoost і LSTM на історичних даних продуктивності команд проєкту з використанням методу покрокової валідації часових рядів виконання спринтів, що дасть змогу забезпечити темпорально коректне оцінювання узагальнювальної здатності моделей за умов послідовного накопичення проєктних даних;
- провести моделювання для кількох інтервалів передбачення та оцінити результати за допомогою метрик абсолютної, квадратичної та відносної похибки, що дасть змогу виявити залежність точності прогнозу від горизонту планування та встановити межі застосовності кожного алгоритму для коротко- і середньотермінового прогнозування спроможності команди;
- визначити переваги й обмеження кожного підходу та обґрунтувати доцільність подальшої побудови метамоделі прогнозування спроможності scrum-команд, що забезпечить теоретичну та емпіричну базу для розроблення гібридної прогностичної системи, здатної адаптивно поєднувати переваги різних класів алгоритмів залежно від контексту проєкту.

Реалізація визначених завдань створює підґрунтя для побудови адаптивних систем підтримки планування обсягу робіт спринту, здатних забезпечувати статистично підтверджену точність прогнозів спроможності scrum-команд, підвищувати обґрунтованість управлінських рішень і сприяти зростанню результативності проєктної діяльності команди виконавців за умов динамічного середовища.

Аналіз останніх досліджень та публікацій. Питання прогнозування зусиль та спроможності виконавців у проєктах, що реалізуються за методологією Scrum, активно розглядають у сучасних наукових дослідженнях. Результати систематичних аналізів наукової літератури у галузі машинного навчання та управління програмни-

ми проєктами [2] свідчать про зростання використання технологій машинного навчання у процесах оцінювання трудомісткості виконання спринтів, розроблення програмного забезпечення загалом та планування ітерацій його виконання зокрема. Доведено, що ML-моделі здебільшого перевершують традиційні методи експертних оцінок, забезпечуючи вищу точність прогнозів і стійкість до варіацій у складі команди проєкту та його динаміки.

У дослідженні [8] автори провели комплексний аналіз застосування методів deep learning для прогнозування командної спроможності у гнучких методологіях розроблення ПЗ. Автори показали, що рекурентні нейронні мережі, зокрема LSTM, демонструють високу ефективність у моделюванні послідовностей спринтів, проте потребують значного обсягу історичних даних для досягнення раціональної точності. Водночас вказано на важливість врахування концептуального дрейфу – поступової зміни поведінки команди проєкту та його контексту з часом. Дослідники відзначають, що градієнтні ансамблі (XGBoost, LightGBM) демонструють стабільно високі результати на структурованих наборах даних [1].

У роботі [6] представлено порівняльний аналіз ансамблевих методів для оцінювання трудомісткості виконання спринтів у scrum-командах. Результати підтверджують, що XGBoost забезпечує кращу інтерпретовність прогнозів спроможності scrum-команд через механізм SHAP values, що дає змогу ідентифікувати найвпливовіші фактори планування обсягу робіт спринту. Це особливо важливо для практичного застосування в IT-індустрії, де прозорість прийняття рішень є критичною.

У контексті часових рядів, нейронні мережі глибокого навчання, особливо рекурентні (LSTM), показують перевагу в аналізі послідовностей виконання завдань [3].

У дослідженні [5] описано застосування трансформерних архітектур для прогнозування velocity команд. Автори показали, що механізми уваги (attention mechanisms) здатні ефективно виявляти довготривалі залежності між спринтами навіть за наявності пропусків у даних. Проте складність навчання та інтерпретації таких моделей обмежує їхнє широке впровадження у практику. У роботі [10] показано, що ансамблеві методи дають можливість підвищити стабільність прогнозів спроможності scrum-команд завдяки об'єднанню кількох алгоритмів, проте жоден окремий метод не демонструє безумовного домінування. Подібні висновки наведено у дослідженнях гібридних рішень, у яких поєднуються текстова семантика user story із табличними ознаками. Автори роботи [14] запропонували метамодельний підхід, що динамічно вибирає між градієнтними та нейромережевими методами залежно від характеристик датасету та горизонту прогнозування командної спроможності. Експериментальна валідація на даних 15 проєктів показала покращення точності на 12-18 % порівняно з використанням окремих базових моделей.

Окремим напрямом розвитку систем прогнозування командної спроможності у сфері планування scrum-проєктів стає застосування метамodelей, що поєднують результати кількох алгоритмів машинного навчання. Суть такого підходу полягає в тому, що результати базових моделей – наприклад, XGBoost, Random Forest або LSTM – агрегуються додатковим рівнем оброблення, який визначає вагомість кожної моделі залежно від

контексту або горизонту прогнозування командної спроможності. Вказано на потенціал адаптивних мета-моделей, які здатні змінювати власні параметри з часом, реагуючи на концептуальний дрейф даних.

Попри широкий спектр досліджень, узагальненого висновку щодо домінування певного алгоритму не існує. Вибір моделі залежить від структури даних, горизонту прогнозування командної спроможності та цілей планування ітерацій scrum-проектів. На основі проведеного аналізу останніх досліджень та публікацій можна стверджувати, що модель XGBoost переважає у короткотермінових прогнозах завдяки стабільності на табличних даних, тоді як LSTM демонструє кращу ефективність за довгих часових інтервалів за рахунок моделювання послідовностей.

Матеріали та методи дослідження. У роботі застосовано методи порівняльного експериментального аналізу, покрокової часово-послідовної валідації та статистичного тестування значущості відмінностей між прогностичними моделями. Дослідження виконано на історичних даних реальної DevOps-команди проекту за 32 завершені двотижневі спринти, що містили показники продуктивності, календарні обмеження, відпустки членів команди та операційне навантаження. Для забезпечення відтворюваності експерименту всі моделі навчалися та тестувалися з фіксованими гіперпараметрами з дотриманням темпоральної послідовності спринтів. Точність прогнозування командної спроможності оцінювалася за метриками MAE, RMSE та MAPE; для статистичного підтвердження значущості різниці між моделями застосовано тест Дібольда-Маріано [3]. Методика дослідження забезпечує можливість повного повторення експерименту іншими дослідниками.

Результати дослідження та їх обговорення / Research results and their discussion

Для оцінювання складності та обсягу роботи використовують показник Story Points (SP) – умовна одиниця вимірювання, яка відображає відносну складність, обсяг і ризики завдання. Зібрано історичні дані за 32 завершені спринти, що включають показники продуктивності, календарні обмеження (робочі дні, свята), відпустки членів команди проекту, операційне навантаження (години дзвінків та інцидентів) та інші контекстні фактори. Задля збереження конфіденційності усі значення продуктивності нормовано сталим коефіцієнтом масштабування (табл. 1).

Для порівняння моделей використано методику покрокової валідації, що відповідає реальним умовам послідовного планування спринтів, яку рекомендують провідні дослідники у галузі прогностичної аналітики [8].

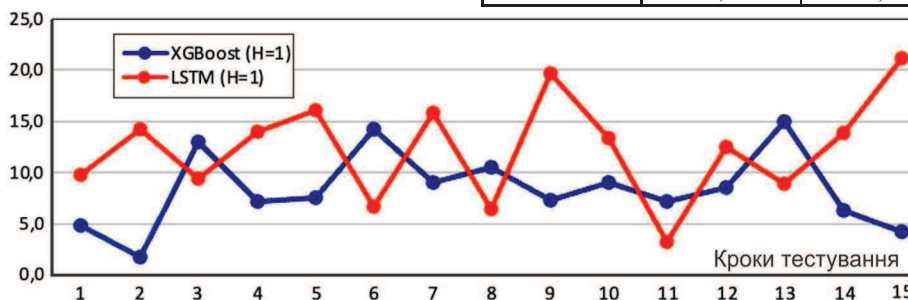


Рис. 1. Абсолютна помилка для кожного прогнозу за покрокового тестування моделей машинного навчання / Absolute error for each prediction in stepwise testing of machine learning models

Модель тренується на всіх попередніх спринтах і прогнозує наступний період (або кілька періодів для середньо- та довготермінових горизонтів). Такий підхід найкоректніший для часових рядів і забезпечує реалістичну оцінку узагальнювальної здатності моделей. Модель XGBoost (англ. *Extreme Gradient Boosting*) – ансамблевий алгоритм, що будує модель у вигляді суми дерев рішень, де кожне наступне дерево мінімізує помилки попередніх [1].

Табл. 1. Приклад історичних даних продуктивності команд проекту / Example of historical performance data for a project team

№	Дата початку	Витрачені, SP	Робочі дні	Відпустки, год	Дзвінки, год	Ковзне середнє, 3 спр.
1	2024-01-10	108	8	13	32	–
2	2024-02-11	107	9	21	15	–
3	2024-03-12	121	8	1	13	115
4	2024-04-13	138	9	0	12	134
5	2024-05-14	142	10	24	18	138
6	2024-06-15	111	8	18	10	110
7	2024-07-16	114	10	32	22	115
8	2024-08-17	129	9	27	21	127
9	2024-09-18	110	10	6	27	112
10	2024-10-19	132	9	21	21	133

У дослідженні модель працювала з табличними ознаками, такими як робочі дні, дата початку спринту, відпустки години, години дзвінків, історичні значення продуктивності та ковзне середнє. Основні гіперпараметри обрано на основі рекомендацій дослідження [6] $\text{max_depth} = 5$, $\text{learning_rate} = 0,05$, $\text{n_estimators} = 300$. LSTM працює тільки з послідовністю попередніх значень SP без ручного формування ознак. Для кожного прогнозу формувалася вхідна послідовність із шести попередніх значень SP, що відповідає рекомендаціям [5] для моделювання коротко- та середньотермінових залежностей.

Результати короткотермінового прогнозування командної спроможності ($H = 1$). На горизонті одного спринту (2 тижні) модель XGBoost продемонструвала статистично значущу перевагу над моделлю LSTM. Модель здатна ефективно інтегрувати структуровані табличні ознаки – календарні обмеження, відпустки, операційне навантаження – і швидко адаптуватися до локальних змін контексту, що узгоджується з висновками дослідження [6] про переваги градієнтних методів на коротких горизонтах (табл. 2).

Табл. 2. Порівняння точності моделей для горизонту $H = 1$ / Comparison of model accuracy for horizon $H = 1$

Модель	MAE	RMSE	MAPE, %
XGBoost	8,42	11,26	7,18
LSTM	12,35	15,87	10,63

Візуалізацію точності моделей наведено на рис. 1. Тест Дібольда-Маріано застосовано до послідовності різниць абсолютних втрат, отриманих у покроковій валідації. Для кожного кроку t обчислювалася різниця:

$$d_t = |e_t^{XGB} - e_t^{LSTM}|, \quad (1)$$

де e_t^{XGB} та e_t^{LSTM} – похибки результатів прогнозу відповідних моделей.

Середню різницю втрат визначають як

$$\bar{d} = \frac{1}{T} \sum_{t=1}^T d_t, \quad (2)$$

а статистика тесту має вигляд

$$DM = \frac{\bar{d}}{\sqrt{\hat{\sigma}_d^2 / T}}, \quad (3)$$

де $\hat{\sigma}_d^2$ – оцінка дисперсії послідовності d_t .

Тест Дібольда-Маріано підтвердив статистично значущу перевагу моделі XGBoost: DM-статистика = -2,84, p -value = 0,005 ($p < 0,01$). Модель XGBoost зменшила середню абсолютну похибку на 31,8 % порівняно з моделлю LSTM.

Результати середньотермінового прогнозування командної спроможності ($H = 3$). На горизонті трьох спринтів (6 тижнів) обидві моделі демонструють порівнянню точність. Модель XGBoost зберігає здатність враховувати структуровані фактори, тоді як LSTM починає виявляти часові закономірності у продуктивності команд проекту. Цей результат підтверджує гіпотезу Л. Жанг, К. Чень, Ю. Ліу та Д. Ванг [14] про існування перехідної зони, де доцільним є комбінування різних класів моделей (табл. 3).

Табл. 3. Порівняння точності моделей для горизонту $H = 3$ / Comparison of model accuracy for horizon $H = 3$

Модель	MAE	RMSE	MAPE, %
XGBoost	14,67	19,24	12,54
LSTM	14,92	19,51	12,81

Тест Дібольда-Маріано: DM-статистика = -0,42, p -value = 0,678. Різниця між моделями статистично незна-

чуша ($p > 0,05$), що свідчить про перехідну зону між домінуванням табличних ознак і часових залежностей. Це вказує на доцільність комбінування обох підходів у метамоделі.

Результати довготермінового прогнозування командної спроможності ($H = 6$). На горизонті шести спринтів (12 тижнів) модель LSTM демонструє статистично значущу перевагу над моделлю XGBoost завдяки здатності моделювати довготривалі часові залежності. Модель ефективніше враховує інерційність команди проекту, сезонні тренди та накопичувальні ефекти, що підтверджує висновки попередніх досліджень [8] про переваги рекурентних архітектур для довготермінових прогнозів спроможності scrum-команд (табл. 4).

Табл. 4. Порівняння точності моделей для горизонту $H = 6$ / Comparison of model accuracy for horizon $H = 6$

Модель	MAE	RMSE	MAPE, %
XGBoost	21,48	27,63	18,42
LSTM	17,26	22,14	14,78

Тест Дібольда-Маріано підтвердив статистично значущу перевагу LSTM: DM-статистика = 2,17, p -value = 0,031 ($p < 0,05$). Модель LSTM зменшила середню абсолютну похибку на 19,6 % порівняно з результатами, отриманими за моделлю XGBoost.

Зведені результати статистичного порівняння. Для детальнішої інтерпретації отриманих результатів, окрім агрегованих метрик MAE/RMSE/MAPE, проаналізовано поведінку моделей у покроковому тестуванні ефективності процесу прогнозування. На рис. 2 наведено значення абсолютної похибки прогнозу ($|e|$) для кожного кроку валідації на трьох горизонтах прогнозування командної спроможності: короткотерміновому ($H = 1$), середньотерміновому ($H = 3$) та довготерміновому ($H = 6$). Така візуалізація дає змогу оцінити не тільки середній рівень точності, але й варіативність помилок та наявність локальних "піків", що є критичним у практичному плануванні спринтів.

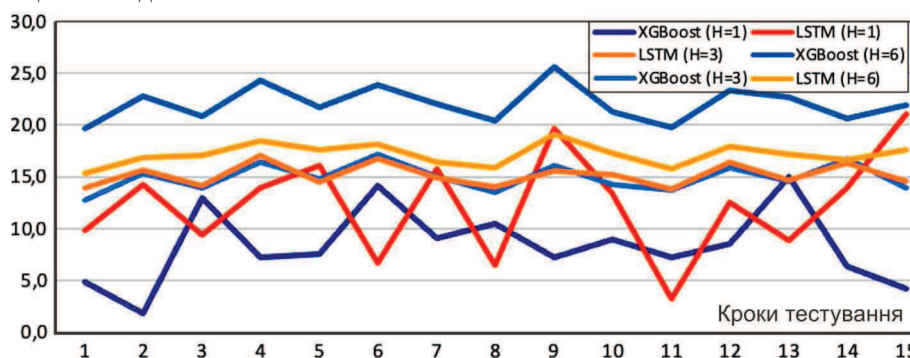


Рис. 2. Порівняльна характеристика абсолютних помилок для кожного прогнозу за покрокового тестування моделей машинного навчання / Comparative characteristics of absolute errors for each forecast in stepwise testing of machine learning models

Як видно з рис. 2, зі збільшенням горизонту прогнозування командної спроможності змінюється характер помилок та домінування моделей. На короткому горизонті ($H = 1$) крива XGBoost здебільшого розташована нижче за моделлю LSTM, що відповідає кращій здатності ансамблевої моделі враховувати контекстні табличні ознаки (календарні обмеження, відпустки, операційне навантаження). На середньому горизонті ($H = 3$) спостерігається зближення кривих і відсутність стійкої переваги однієї моделі, що узгоджується з результатами статистичного порівняння та вказує на "перехідну зону". На довготерміновому горизонті ($H = 6$) помилки

моделі LSTM стають відносно стабільнішими порівняно з моделлю XGBoost, що можна пояснити здатністю рекурентних архітектур моделювати накопичувальні часові залежності та інерційність продуктивності команд проекту.

Незважаючи на наочність покрокових кривих (див. рис. 2), візуальний аналіз сам по собі не дає змоги зробити остаточні висновки щодо статистичної значущості відмінностей між моделями. Для формального підтвердження спостережених тенденцій виконано статистичне порівняння прогнозів за допомогою тесту Дібольда-Маріано для кожного горизонту прогнозування командної

спроможності. Узагальнені результати статистичної перевірки наведено в табл. 5.

Табл. 5. Статистична значущість різниці між моделями на різних горизонтах / Statistical significance of the difference between models at different horizons

Горизонт	Краща модель	DM-статистика	<i>p</i> -value
$H = 1$ (1 спринт)	XGBoost	-2,84	0,005**
$H = 3$ (3 спринти)	<i>Parusnet</i>	-0,42	0,678
$H = 6$ (6 спринтів)	LSTM	2,17	0,031*

Примітка: * $p < 0,05$; ** $p < 0,01$

Отримані значення DM-статистики та *p*-value підтверджують зміну домінування моделей залежно від горизонту прогнозування командної спроможності: модель XGBoost має статистично значущу перевагу на короткому горизонті, на середньому горизонті різниця є незначущою, тоді як на довготерміновому інтервалі переважає модель LSTM.

Для узагальнення цієї тенденції та більш компактного подання зміни точності моделей залежно від горизонту прогнозування командної спроможності побудовано зведену діаграму середніх значень MAE (рис. 3).

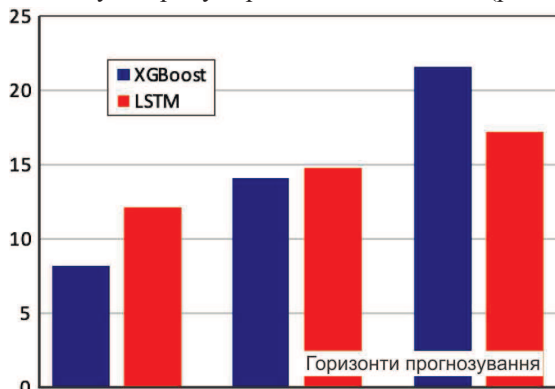


Рис. 3. Точність моделей прогнозування командної спроможності залежно від горизонту / Accuracy of team capability prediction models depending on the horizon

Як видно з рис. 3, зі збільшенням горизонту прогнозування командної спроможності спостерігається зміна домінування класу моделей: модель XGBoost переважає на короткому горизонті, тоді як LSTM демонструє кращі результати на довготерміновому інтервалі.

Аналіз результатів дослідження. Інтерпретація виявлених закономірностей. Експериментальні результати демонструють чітку зміну домінування моделей залежно від часового горизонту прогнозування командної спроможності, що можна пояснити фундаментальними відмінностями у принципах роботи градієнтних ансамблів та рекурентних нейронних мереж. Модель XGBoost, будучи ансамблем дерев рішень, ефективно працює з табличними ознаками через механізм розщеплення вузлів на основі інформаційного критерію. Це дає змогу моделі швидко виявляти локальні залежності між календарними обмеженнями, відпустками та продуктивністю команди проєкту на короткотерміновому горизонті. Натомість модель LSTM, завдяки архітектурі з вентилями забування та запам'ятовування, здатна накопичувати інформацію про довготривалі тренди продуктивності, що стає критичним під час прогнозування на шість спринтів вперед.

Існування перехідної зони на горизонті $H = 3$, де статистична різниця між моделями відсутня ($p = 0,678$), вказує на баланс між впливом контекстних факторів і часових залежностей. Цей результат узгоджується з концепцією "ефективної довжини пам'яті" LSTM-мереж, яка у цьому дослідженні становить приблизно 6 попередніх спринтів. При цьому горизонті моделі ще недостатньо для повного використання переваг рекурентної архітектури, проте горизонт вже достатньо великий, щоб табличні ознаки втратили частину своєї прогностичної сили.

Теоретичне обґрунтування переваг моделей. Перевагу моделі XGBoost на короткотерміновому горизонті можна пояснити природою градієнтного підсилення як методу, орієнтованого на мінімізацію помилок через ітеративне додавання слабких базових учнів. Кожне наступне дерево у композиції навчається на залишкових помилках попередніх, що забезпечує швидку адаптацію до локальних особливостей даних. Механізм регуляризації моделі XGBoost (через параметри *max_depth*, *min_child_weight*, *gamma*) запобігає перенаванчання навіть за наявності великої кількості ознак, що важливо у роботі з календарними обмеженнями та метриками операційного навантаження.

Модель LSTM демонструє перевагу на довготермінових горизонтах завдяки архітектурі комірок пам'яті з керованими вентилями. Вентиль забування дає змогу моделі селективно "відкидати" застарілу інформацію, тоді як вентиль входу регулює надходження нових даних. Це створює механізм накопичення контекстної інформації про інерційність команди проєкту та сезонні коливання продуктивності. Водночас модель LSTM потребує достатнього обсягу навчальних даних для налаштування великої кількості вагових параметрів, що пояснює її нижчу ефективність на короткому горизонті, де історія спринтів обмежена.

Обговорення результатів дослідження. У дослідженні [6] автори провели порівняльний аналіз алгоритмів машинного навчання для оцінювання трудомісткості виконання спринтів у scrum-командах і показано, що модель XGBoost забезпечує вищу точність на структурованих табличних даних завдяки ефективному виконанню роботи з контекстними ознаками. Отримані у цьому дослідженні результати узгоджуються з цим висновком: модель XGBoost демонструє статистично значущу перевагу на короткотерміновому горизонті ($H = 1$, MAE = 8,42, $p = 0,005$), де домінують саме структуровані ознаки – календарні обмеження, відпустки та операційне навантаження команди проєкту.

У дослідженні [8] автори продемонстрували ефективність використання рекурентних нейронних мереж LSTM для моделювання послідовностей спринтів у гнучких методологіях і вказали на здатність цих моделей враховувати концептуальний дрейф поведінки команди проєкту. Результати цього дослідження підтверджують цей висновок на довготерміновому горизонті ($H = 6$), де модель LSTM досягає зниження похибки на 19,6 % ($p = 0,031$) завдяки моделюванню інерційності та сезонних трендів продуктивності.

У роботі [14] автор запропонував метамодельний підхід для динамічного вибору між градієнтними та нейромережевими методами залежно від характеристик даних і горизонту прогнозування. Це дослідження емпі-

рично підтверджує доцільність такого підходу: різниця між раціональним та нераціональним вибором алгоритму становить 31,8 % для $H = 1$ та 19,6 % для $H = 6$, що обґрунтовує практичну цінність адаптивної метамоделі.

У дослідженні [7] автори розробили модель оцінювання прогресу і темпу виконання робіт у спринтах на основі даних реального Agile-інструменту та порівняли ефективність різних алгоритмів машинного навчання для прогнозування стану спринту. Автори встановили, що жоден алгоритм не демонструє стабільно кращих результатів у всіх контекстах, а точність прогнозування істотно залежить від складу та якості вхідних ознак. Цей висновок узгоджується з результатами цього дослідження, де перевагу моделі XGBoost на горизонті $H = 1$ можна пояснити саме ефективним використанням структурованих контекстних ознак.

У роботі [9] автори представили емпіричні моделі оцінювання трудомісткості Agile-процесів на основі реальних проєктних даних і показали, що контекстні організаційні фактори – склад команди, зовнішні обмеження, рівень зрілості процесів – є ключовими предикторами точності. Результати цього дослідження підтверджують цей висновок: внесення до моделі XGBoost операційного навантаження та календарних обмежень забезпечило статистично значуще зниження MAE ($p = 0,005$) порівняно з моделлю LSTM, яка оперує винятково послідовністю попередніх значень SP.

У дослідженні [12] автори провели реплікаційне дослідження методів автоматизованого оцінювання трудомісткості на масштабованому наборі даних (31 960 задач) і встановили, що збільшення обсягу даних не гарантує підвищення точності – вирішальну роль відіграє контекстуальна релевантність ознак і відповідність методу валідації реальним умовам планування. Цей висновок безпосередньо узгоджується з результатами цього дослідження: покрокова валідація на 32 спринтах із темпорально коректним розподілом даних забезпечила реалістичну оцінку узагальнювальної здатності моделей, уникаючи систематичного оптимістичного зміщення стандартного k -fold крос-валідування.

Отже, внаслідок обговорення результатів дослідження з'ясовано, що результати цього дослідження не тільки узгоджуються з висновками наявної наукової літератури, але й доповнюють її систематичним статистичним підтвердженням зміни домінування класів моделей залежно від горизонту прогнозування – аспекту, який у попередніх роботах розглядали тільки частково або без застосування формальних статистичних тестів порівняння прогностичної точності. Виявлена перехідна зона на середньотерміновому горизонті ($H = 3$, $p = 0,678$) є оригінальним внеском, що відкриває перспективу розроблення адаптивної метамоделі прогнозування спроможності scrum-команд.

Отже, внаслідок виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – отримала подальший розвиток методика моделювання часових залежностей та структурованих факторів продуктивності команд проєкту, яка, на відміну від наявних методик, враховує явище "перехідної зони" на середньотерміновому горизонті ($H = 3$ спринти) та передбачає застосування тесту Дібольда-Маріано для статис-

тичного підтвердження значущості відмінностей між класами моделей на кожному горизонті прогнозування, що дає можливість встановити кількісні межі домінування градієнтних методів і рекурентних нейронних мереж та підвищити точність багатогоризонтного прогнозування спроможності scrum-команд за умов змінного проєктного контексту.

Практична значущість результатів дослідження – можна застосовувати у практиці управління scrum-проєктами як науково обґрунтований інструмент вибору алгоритму прогнозування спроможності команди залежно від горизонту планування обсягу робіт спринту: модель XGBoost – для короткотермінового горизонту ($H = 1$), комбінацію методів – для середньотермінового ($H = 3 - 4$), модель LSTM – для довготермінового ($H \geq 5$), а методику покрокової валідації з тестом Дібольда-Маріано – для автоматичного визначення раціонального алгоритму в системах автоматизованого планування обсягу робіт scrum-проєктів.

Висновки / Conclusions

Визначено переваги та обмеження різних класів прогностичних моделей командної спроможності на основі історичних даних виконання робіт проєкту з урахуванням часової структури, контекстних факторів та тривалості прогностичного інтервалу, що забезпечить науково обґрунтований вибір алгоритму прогнозування залежно від горизонту планування обсягу робіт спринту та створить теоретичне підґрунтя для розроблення адаптивної метамоделі, здатної динамічно комбінувати переваги градієнтних методів і рекурентних нейронних мереж у змінному проєктному контексті. За результатами проведеного дослідження можна зробити такі основні висновки.

1. Проведено аналіз сучасних наукових праць щодо застосування методів машинного навчання у плануванні scrum-проєктів та виявлено ключові обмеження наявних підходів. Встановлено, що попри значний обсяг досліджень у цій області, відсутнє систематичне порівняння ефективності використання градієнтних методів та рекурентних нейронних мереж на різних часових горизонтах прогнозування командної спроможності за умов реальних проєктів. Не встановлено, за яких умов доцільно використовувати кожен із підходів, що ускладнює вибір раціонального алгоритму для практичного застосування у процесі планування спринтів.
2. Розроблено експериментальну методику порівняння роботи алгоритмів XGBoost і LSTM на історичних даних продуктивності команд проєкту з використанням методу покрокової валідації. Методика передбачає послідовне тренування моделей на всіх попередніх спринтах з прогнозуванням наступного періоду, що відповідає реальним умовам планування обсягу робіт спринту. Застосовано метрики MAE, RMSE та MAPE для оцінювання точності прогнозів спроможності scrum-команд, а також тест Дібольда-Маріано для статистичного підтвердження значущості різниці між моделями.
3. Проведено моделювання для трьох прогностичних горизонтів і оцінено результати за метриками MAE, RMSE, MAPE та тестом Дібольда-Маріано. Встановлено, що на короткотерміновому горизонті ($H = 1$ спринт) модель XGBoost демонструє статистично значущу перевагу з $MAE = 8,42$ проти $MAE = 12,35$ для LSTM ($p = 0,005$), забезпечуючи зниження похибки на 31,8 %. На середньотерміновому горизонті ($H = 3$ спринти) моделі демонструють паритетну ефективність з $MAE = 14,67$ для моделі XGBoost та $MAE = 14,92$ для LSTM ($p = 0,678$), що свідчить про перехідну зону між доміну-

ванням різних класів моделей. На довготерміновому горизонті ($H = 6$ спринтів) LSTM виявляє статистично значущу перевагу з $MAE = 17,26$ проти $MAE = 21,48$ для моделі XGBoost ($p = 0,031$), зменшуючи похибку на 19,6 %.

4. Визначено переваги й обмеження кожного підходу та обґрунтовано доцільність подальшої побудови метамоделі прогнозування командної спроможності. Модель XGBoost ефективно інтегрує структуровані табличні ознаки (календарні обмеження, відпустки, операційне навантаження) та швидко адаптується до локальних змін контексту на короткотермінових горизонтах. LSTM здатна моделювати довготривалі часові залежності, інерційність команди проєкту та сезонні тренди на довготермінових горизонтах завдяки архітектурі з вентилями забування та запам'ятовування. Існування перехідної зони на середньому горизонті, де жоден з методів не має статистично значущої переваги, обґрунтовує доцільність розроблення адаптивної метамоделі, що динамічно вибиратиме або комбінуватиме алгоритми градієнтного підсилення та рекурентних нейронних мереж для підвищення точності прогнозування обсягу робіт спринту залежно від горизонту прогнозування командної спроможності та контексту scrum-проєкту для підвищення загальної точності системи прогнозування.

References

1. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>
2. Choetkierikul, M., Dam, H. K., Tran, T., & Ghose, A. (2018). Predicting delivery capability in iterative software development. *IEEE Transactions on Software Engineering*, 44(6), 551–573. <https://doi.org/10.1109/TSE.2017.2693989>
3. Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3), 253–263. <https://doi.org/10.1080/07350015.1995.10524599>
4. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
5. Kumar, A., Sharma, R., & Singh, P. (2023). Transformer-based velocity prediction for agile software development teams. *Journal of Systems and Software*, 198, article ID 111587. <https://doi.org/10.1016/j.jss.2022.111587>
6. Maleki, N., Zakerolhosseini, A., & Derakhshanfar, P. (2022). Ensemble methods for effort estimation in Scrum projects: A comparative study. *Information and Software Technology*, 146, article ID 106875. <https://doi.org/10.1016/j.infsof.2022.106875>
7. Peralta, C., Rodrigues, M., & García-Martínez, R. (2024). Sprint Management in Agile Approach: Progress and Velocity Evaluation Applying Machine Learning. *Information*, 15(11), article ID 726. <https://doi.org/10.3390/info15110726>
8. Rajbahadur, G. K., Wang, S., Oliva, G. A., Kamei, Y., & Hassan, A. E. (2021). The impact of using deep learning models for effort estimation in agile software development. *Empirical Software Engineering*, 26(4), 82. <https://doi.org/10.1007/s10664-021-09992-4>
9. Rosa, W., Clark, B. K., Madachy, R. J., & Boehm, B. W. (2022). Empirical Effort and Schedule Estimation Models for Agile Processes in the US DoD. *IEEE Transactions on Software Engineering*, 48(8), 3117–3130. <https://doi.org/10.1109/TSE.2021.3112743>
10. Shepperd, M., & MacDonell, S. (2012). Evaluating prediction systems in software project estimation. *Information and Software Technology*, 54(8), 820–827. <https://doi.org/10.1016/j.infsof.2011.12.008>
11. Standish Group. (2020). *CHAOS Report 2020: Beyond Infinity*. The Standish Group International. URL: <https://hennypotman.wordpress.com/wp-content/uploads/2021/01/project-success-qrc-standish-group-chaos-report-2020.pdf>
12. Tawosi, V., Moussa, R., & Sarro, F. (2022). Agile Effort Estimation: Have We Solved the Problem Yet? Insights From A Replication Study. <https://doi.org/10.48550/arXiv.2201.05401>
13. Torskyi, O. I., & Hrytsiuk, Y. I. (2025). Application of machine learning to enhance the efficiency of automated software testing. *Scientific Bulletin of UNFU*, 35(4), 142–149. <https://doi.org/10.36930/40350416>
14. Zhang, L., Chen, X., Liu, Y., & Wang, J. (2024). Adaptive meta-model for sprint capacity forecasting: A multi-horizon approach. *IEEE Transactions on Software Engineering*, 50(2), 348–365. <https://doi.org/10.1109/TSE.2023.3289456>

O. O. Hud, N. E. Kunanets

Lviv Polytechnic National University, Lviv, Ukraine

COMPARATIVE ANALYSIS OF THE EFFICIENCY OF XGBOOST AND LSTM ALGORITHMS IN THE PROBLEM OF FORECASTING THE CAPACITY OF SCRUM TEAMS AT DIFFERENT HORIZONS

This study demonstrates how machine learning technologies can enhance sprint planning accuracy in scrum-based software development. An experimental investigation was conducted using historical data from a production DevOps team across 32 completed sprints, incorporating productivity metrics, calendar constraints, team member vacations, operational workload, and contextual factors. The research methodology involved systematic comparative analysis of two fundamentally different machine learning model classes: XGBoost gradient boosting and LSTM recurrent neural networks for forecasting team capacity across varying time horizons. Walk-forward validation methodology was applied with MAE, RMSE, and MAPE metrics, employing the Diebold-Mariano test for statistical model comparison. The experimental results revealed distinct performance patterns across different forecasting horizons. For short-term forecasting ($H = 1$ sprint), XGBoost demonstrated statistically significant superiority with $MAE = 8.42$ versus $MAE = 12.35$ for LSTM ($p = 0.005$), consequently achieving 31.8 % error reduction through effective integration of structured tabular features and calendar constraints. At the medium-term horizon ($H = 3$ sprints), both models exhibited comparable performance with $MAE = 14.67$ for XGBoost and $MAE = 14.92$ for LSTM ($p = 0.678$), indicating a transition zone where neither tabular features nor temporal dependencies dominate. It was established that walk-forward validation ensures temporally correct assessment of model generalization capacity and eliminates systematic optimistic bias inherent in standard k -fold cross-validation applied to sprint time series data. Moreover, for long-term forecasting ($H = 6$ sprints), LSTM showed statistically significant advantage with $MAE = 17.26$ versus $MAE = 21.48$ for XGBoost ($p = 0.031$), achieving 19.6 % error reduction by capturing long-term temporal dependencies, team inertia, and seasonal trends. These complementary strengths justify the development of an adaptive meta-model that dynamically selects or combines algorithms depending on forecasting horizon and project context. The proposed approach provides Scrum project managers with a scientifically grounded tool for algorithm selection based on planning goals, enabling direct implementation in automated sprint workload planning systems.

Keywords: iterative workload planning; time series; ensemble methods; recurrent neural networks; walk-forward validation; Diebold-Mariano test; adaptive metamodel.