КОНТАМІНАЦІЯ ТЕСТОВИХ НАБОРІВ ЯК ЗАГРОЗА ДОСТОВІРНОСТІ ОЦІНЮВАННЯ АГЕНТНИХ СИСТЕМ

Киселевич Володимир Володимирович
асистент кафедри комп'ютерних наук та інформаційних технологій
Житомирський державний університет імені Івана Франка,
Україна

Постановка проблеми. Оцінювання програмних систем на основі великих мовних моделей спирається переважно на відкриті набори задач, розміщені у публічних репозиторіях. Набір SWE-bench, який містить 2294 задачі з реальних звітів про дефекти та відповідних запитів на злиття з 12 популярних репозиторіїв мовою Python, на момент оприлюднення давав найкращий результат 1,96 відсотка розв'язаних задач для моделі Claude 2 [1]. Показники, заявлені на похідних підмножинах того самого набору, за наступні роки зросли на порядок, і саме ці числа використовують як свідчення практичної придатності агентних систем. Джерело сумніву полягає в тому, що разом із показником змінилися й умови вимірювання: формулювання задач, тести та канонічні виправлення залишаються у відкритому доступі протягом усього періоду, з якого збирають навчальні корпуси наступних поколінь моделей. Тому приріст показника допускає щонайменше два несумісні тлумачення, а саме зростання здатності розв'язувати нові задачі та зростання частки задач, відтворених із запам'ятованого. Розрізнити ці тлумачення за підсумковим числом неможливо, оскільки читач публікації не має доступу ні до навчальних даних моделі, ні до повних траєкторій виконання агента.

Аналіз останніх досліджень. Наведені далі числові оцінки переважно походять із препринтів [1; 2; 3; 4; 5; 6; 7; 8; 9], тому їх слід трактувати як попередні, до завершення рецензування. Прямі ознаки запам'ятовування на найпоширенішому наборі задач з програмної інженерії зафіксовано в рецензованому дослідженні [10]: моделі найвищого рівня визначають шлях до файлу з дефектом з точністю до 76 відсотків, маючи лише текст звіту про дефект і не маючи доступу до структури репозиторію, тоді як на задачах із репозиторіїв, не включених до набору, та сама точність не перевищує 53 відсотків; під час відтворення функцій дослівна подібність за перекриттям п'ятиграм досягає 35 відсотків на підмножинах Verified і Full проти 18 відсотків на споріднених наборах. Незалежний перегляд того самого набору [11] показав, що 32,67 відсотка успішних виправлень одного з поширених агентів містять витік розв'язку у самому формулюванні задачі, а 31,08 відсотка зарахованих виправлень є підозрілими через слабкі тестові набори; після вилучення дефектних задач показник розв'язання впав з 12,47 до 3,97 відсотка, причому понад 94 відсотки задач набору створено раніше за дату відсікання навчальних даних. Масштабне вимірювання витоку на 83 наборах задач з програмної

інженерії [2] дає середні коефіцієнти 4,8, 2,8 і 0,7 відсотка для наборів мовами Python, Java і C або C++, проте окремі набори мають значно вищі значення, зокрема QuixBugs 100 відсотків і BigCloneBench 55,7 відсотка, тобто витік є концентрованим, а не рівномірно розподіленим. Чутливість рейтингів до перебудови задач показано на 51 моделі [3], де перехід від стандартного набору до задач, отриманих керованою мутацією умов, дає середнє падіння 39,4 відсотка з розкидом від 19,6 до 47,7 відсотка та різкі зміни порядку моделей. Контрольований повторний набір арифметичних задач [4] дає падіння точності до 8 відсотків, позитивний зв'язок між імовірністю породження моделлю прикладу з вихідного набору та розривом у показниках (квадрат коефіцієнта кореляції Спірмена становить 0,36), і водночас мінімальні ознаки перенавчання для частини передових моделей, тобто ризик є суттєвим, але не універсальним. Як конструктивну відповідь запропоновано вимірювання на задачах з відомою датою публікації [5], де набір з 400 задач, оприлюднених з травня 2023 по травень 2024 року на трьох платформах змагального програмування, дав змогу оцінити 18 базових і 34 моделі, донавчені на інструкціях з контролем за часом. Дефекти самих наборів задач, не пов'язані з протіканням, оцінено на корпусі з 168 наборів у дев'яти доменах [6]: критичні проблеми (неоднозначність постановки, конфлікти середовища виконання, некоректні еталонні відповіді) виявлено у понад 25,7 відсотка задач, а фільтрація дефектних задач змінює рейтинги моделей і підвищує середню продуктивність на двох поширених наборах на 9,9 і 9,6 відсотка. Окремо слід відзначити межі інструментального виявлення контамінації: систематичний перегляд 50 публікацій виокремив вісім категорій припущень, на яких будують методи виявлення, і показав, що три перевірені припущення дають результати, близькі до випадкового вгадування, на даних попереднього навчання [12]; той самий висновок отримано для атак на визначення належності до навчальної вибірки на моделях від 160 мільйонів до 12 мільярдів параметрів, де видима вразливість пояснюється зміщенням розподілу між часовими діапазонами вибірок, а не властивістю моделі [7]. Методологічні настанови для наборів задач з програмування [8] та узгоджений перелік вимог до звітування в дослідженнях на основі машинного навчання [13] формулюють вимоги до укладачів наборів і авторів експериментів, а огляд складу рейтингових таблиць [9] показує, що серед 79 записів однієї підмножини та 133 записів іншої переважають заявки від індустрії із закритими моделями, що додатково обмежує можливість зовнішньої перевірки.

Мета. Мета роботи полягає в побудові діагностичного інструмента, який дає змогу читачеві публікації оцінити ризик контамінації тестового набору в чужому експерименті, не маючи доступу ні до навчальних даних моделі, ні до внутрішніх артефактів експерименту. Для цього потрібно виокремити перелік ознак ризику, які можна виявити в тексті публікації, сформулювати мінімальні вимоги до звітування, за яких ризик стає оцінюваним, і зафіксувати умову оцінюваності. Робота продовжує лінію попередніх досліджень автора щодо відтворюваності оцінювання агентних систем, переносячи розгляд з площини повторюваності окремих запусків у площину достовірності самого набору задач.

Виклад основного матеріалу. Запропоновано перелік з десяти ознак ризику контамінації, згрупованих у три категорії за тим, який складник публікації потрібно прочитати для перевірки. Першу категорію утворюють ознаки походження та часового співвідношення набору задач і моделі. Ознака 1 полягає в тому, що дата оприлюднення набору задач передує оголошеній даті відсікання навчальних даних використаних моделей; за відсутності у тексті хоча б однієї з цих дат ознаку слід вважати наявною, оскільки перевірку унеможливлено. Ознака 2 полягає у походженні задач з невеликого числа популярних відкритих репозиторіїв, зафіксованих до дати відсікання, що підтверджено оцінкою понад 94 відсотки задач, створених до цієї дати, для одного з найуживаніших наборів [11], та концентрацією витоку в окремих наборах [2]. Ознака 3 полягає у відсутності роздільного показника для задач, створених до і після дати відсікання, тоді як роздільне вимірювання за датою є технічно досяжним [5], а розрив між вихідним і контрольним наборами вимірюваним [4].

Другу категорію утворюють ознаки, які стосуються змісту задачі та вигляду розв’язку. Ознака 4 полягає у наявності у формулюванні задачі покликання на публічний репозиторій, звіт про дефект або запит на злиття, де вже міститься канонічне виправлення, що виявлено приблизно у третині зарахованих успішних виправлень [11]. Ознака 5 полягає в аномально високій частці точних або майже точних збігів із канонічним виправленням, зокрема у високому перекритті п’ятиграм, яке для контамінованих підмножин удвічі перевищує значення на споріднених наборах [10]. Ознака 6 полягає у відсутності проміжних кроків, які пояснюють розв’язок: агент одразу називає файл, функцію або рядок без діагностичних дій, тоді як така сама здатність зникає на задачах поза набором [10]. Ознака 7 полягає у відчутному падінні показника після незначної перебудови формулювання задачі, що для мутованих наборів становить у середньому 39,4 відсотка [3].

Третю категорію утворюють ознаки протоколу вимірювання. Ознака 8 полягає у відсутності контролю на прихованій або новоствореній підмножині задач, тобто в тому, що всі числа отримано лише на відкритому наборі [5]. Ознака 9 полягає у відсутності повідомлення про контроль якості задач і сили тестів, попри те, що частка дефектних завдань у наборах перевищує чверть [6], а фільтрація суттєво змінює показник [11], і попри наявність готових настанов з перевірними пунктами [8]. Ознака 10 полягає у звітуванні єдиного числа без розподілу за складністю, датою створення задачі та числом спроб, що суперечить узгодженим вимогам до звітування в дослідженнях на основі машинного навчання [13] і не компенсується зовнішнім переглядом рейтингової таблиці, де переважають закриті комерційні розв’язки [9]. Зведення ознак і способів їхньої перевірки за текстом публікації подано в табл. 1.

Таблиця 1. Ознаки ризику контамінації та спосіб їхньої перевірки за текстом публікації

Ознака ризику	Як перевірити за текстом публікації	Підстава
Набір старший за модель	Порівняти дату набору й дату відсікання даних	[1; 11]
Вузьке коло популярних репозиторіїв	Прочитати опис походження задач	[1; 2]
Немає поділу за датою задачі	Шукати роздільні показники за часом створення	[4; 5]
Протікання розв'язку у формулюванні	Перевірити приклад формулювання на покликання	[11]
Збіг із канонічним виправленням	Шукати оцінку дослівного перекриття	[10]
Успіх без проміжних кроків	Перевірити наявність опублікованих траєкторій	[10]
Падіння після перебудови умови	Шукати результат на мутованих задачах	[3]
Немає затемненої підмножини	Перевірити склад контрольного набору	[5]
Немає перевірки якості задач і тестів	Шукати перелік вилучених задач	[6; 8]
Єдине зведене число	Перевірити розподіл за складністю і спробами	[9; 13]

Джерело: розробка автора

Побудовано також перелік мінімальних вимог до звітування, виконання яких робить ризик контамінації оцінюваним. Перша вимога полягає у зазначенні дати відсікання навчальних даних і точної версії кожної використаної моделі. Друга вимога полягає у поданні результату окремо для підмножини задач, створених після цієї дати, за єдиним для обох підмножин протоколом. Третя вимога полягає у фіксації точної версії набору задач і переліку вилучених задач з обґрунтуванням вилучення для кожної з них. Четверта вимога полягає у публікуванні повних траєкторій виконання, а не лише підсумкового показника, оскільки саме траєкторія дає змогу побачити відсутність діагностичних кроків. П'ята вимога полягає у зазначенні числа спроб на задачу і правила вибору найкращої спроби, бо без цього показник не порівнюваний між роботами. Шоста вимога полягає у повідомленні параметрів висновування та розкиду між повторними запусками, оскільки варіативність запусків впливає на підсумкове число.

Слід підкреслити різницю між усуненням і оцінюваністю ризику. Перелічені вимоги не зменшують частки запам'ятованого в моделі та не очищують навчальні дані, тобто не усувають контамінацію. Їхня дія полягає у переведенні непрозорого числа у величину, для якої читач може побудувати нижню оцінку викривлення. Розраховувати на інструментальне усунення також не варто, бо методи виявлення контамінації спираються на припущення, перевірка яких дає

результати, близькі до випадкового вгадування [7; 12]. Сформульовано умову оцінюваності: ризик контамінації є оцінюваним тоді, коли звіт дає змогу, по-перше, відтворити поділ набору задач на підмножини за датою створення задачі відносно дати відсікання навчальних даних кожної використаної моделі, і, по-друге, прочитати показник окремо на пізнішій підмножині за тим самим протоколом. Обидві частини є необхідними: перша без другої дає лише опис даних, друга без першої не піддається перевірці читачем. Публікація, яка не відповідає цій умові, не дає підстав тлумачити свій показник як міру здатності розв'язувати нові задачі, незалежно від абсолютного значення показника.

Обмеження. Робота не містить власного експерименту: перелік ознак і вимог побудовано на опрацюванні чужих вимірювань, і його прогностична сила щодо конкретної публікації не перевірена. Усі числові значення взято з опрацьованих джерел, зокрема з препринтів [1; 2; 3; 4; 5; 6; 7; 8; 9], тому вони можуть змінитися після рецензування, а частина з них стосується окремих наборів задач і не переноситься на інші домени. Ознаки не є взаємно незалежними, і запропоноване групування у три категорії є класифікаційним припущенням, а не результатом факторного аналізу. Вагові коефіцієнти ознакам не приписано, тому перелік дає якісний, а не числовий рівень ризику, і не дає змоги оцінити ймовірність помилкового зарахування добросовісної публікації до групи ризику. Статистичної перевірки зв'язку між наявністю ознак і величиною завищення показника не виконано, оскільки для цього потрібна вибірка публікацій з відомою істинною величиною контамінації. Вибірка опрацьованих джерел обмежена англомовними працями 2023–2026 років з переважною орієнтацією на задачі програмної інженерії та генерації коду, тому висновки щодо інших предметних областей є гіпотетичними. Умова оцінюваності є необхідною, але не достатньою для достовірності: її виконання не захищає від інших джерел викривлення, зокрема від дефектів самих задач і слабких тестів.

Висновки. Показники агентних систем вимірюють на відкритих наборах задач, які потрапляють у навчальні корпуси моделей, тому приріст показника не є самодостатнім свідченням поліпшення здатності розв'язувати задачі. Запропоновано перелік з десяти ознак ризику контамінації, згрупованих за походженням набору, змістом задачі та протоколом вимірювання, який читач може застосувати до чужої публікації без доступу до навчальних даних. Побудовано перелік шести мінімальних вимог до звітування і сформульовано умову оцінюваності, за якою ризик є оцінюваним лише тоді, коли звіт дає змогу відтворити поділ задач за датою створення відносно дати відсікання навчальних даних і показує результат окремо на пізнішій підмножині. Розмежування усунення і оцінюваності ризику є ключовим для тлумачення опублікованих чисел, оскільки інструментальне виявлення контамінації наразі не забезпечує надійного результату. Подальші дослідження доцільно спрямувати на перевірку переліку ознак на репрезентативній вибірці публікацій і на приписування ознакам ваг за їхньою діагностичною силою.

Список літератури

1. Jimenez C. E., Yang J., Wettig A. [та ін.] SWE-bench: Can Language Models Resolve Real-World GitHub Issues? : препринт. arXiv:2310.06770. 2023. URL: https://arxiv.org/abs/2310.06770.

2. Zhou X., Weyssow M., Widyasari R. [та ін.] LessLeak-Bench: A First Investigation of Data Leakage in LLMs Across 83 Software Engineering Benchmarks : препринт. arXiv:2502.06215. 2025. URL: https://arxiv.org/abs/2502.06215.

3. Xia C. S., Deng Y., Zhang L. Top Leaderboard Ranking = Top Coding Proficiency, Always? EvoEval: Evolving Coding Benchmarks via LLM : препринт. arXiv:2403.19114. 2024. URL: https://arxiv.org/abs/2403.19114.

4. Zhang H., Da J., Lee D. [та ін.] A Careful Examination of Large Language Model Performance on Grade School Arithmetic : препринт. arXiv:2405.00332. 2024. URL: https://arxiv.org/abs/2405.00332.

5. Jain N., Han K., Gu A. [та ін.] LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code : препринт. arXiv:2403.07974. 2024. URL: https://arxiv.org/abs/2403.07974.

6. Wang J., Bianchi F., Zhu S. [та ін.] Automated Benchmark Auditing for AI Agents and Large Language Models : препринт. arXiv:2605.26079. 2026. URL: https://arxiv.org/abs/2605.26079.

7. Duan M., Suri A., Mireshghallah N. [та ін.] Do Membership Inference Attacks Work on Large Language Models? : препринт. arXiv:2402.07841. 2024. URL: https://arxiv.org/abs/2402.07841.

8. Cao J., Chan Y., Ling Z. [та ін.] Code Benchmarks Should Prioritize Rigor, Reliability, and Reproducibility : препринт. arXiv:2501.10711. 2025. URL: https://arxiv.org/abs/2501.10711.

9. Martinez M., Franch X. What's in a Benchmark? The Case of SWE-Bench in Automated Program Repair : препринт. arXiv:2602.04449. 2026. URL: https://arxiv.org/abs/2602.04449.

10. Li S., Garg S., Moghaddam R. Z. The SWE-Bench Illusion: When State-of-the-Art LLMs Remember Instead of Reason. Proceedings of the IEEE/ACM 48th International Conference on Software Engineering: Software Engineering in Practice. 2026. P. 395–405. DOI: 10.1145/3786583.3786882.

11. Aleithan R., Xue H., Mohajer M. M. [та ін.] SWE-Bench+: Enhanced LLM Coding Benchmark. Proceedings of the 3rd ACM International Conference on AI-Powered Software. 2026. P. 332–339. DOI: 10.1145/3805760.3814924.

12. Fu Y., Uzuner Ö., Yetişgen M. [та ін.] Does Data Contamination Detection Work (Well) for LLMs? A Survey and Evaluation on Detection Assumptions. Findings of the Association for Computational Linguistics: NAACL 2025. 2025. P. 5235–5256. DOI: 10.18653/v1/2025.findings-naacl.291.

13. Kapoor S., Cantrell E. M., Peng K. [та ін.] REFORMS: Consensus-based Recommendations for Machine-learning-based Science. Science Advances. Vol. 10, No. 18. 2024. DOI: 10.1126/sciadv.adk3452.