ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА ЕЛЕКТРОННА ІНЖЕНЕРІЯ
INFORMATION AND COMMUNICATION TECHNOLOGIES, ELECTRONIC ENGINEERING

№ 5 (1), 2025

ЕФЕКТИВНІСТЬ ФОРМАТІВ ІНСТРУКЦІЙ LLM ДЛЯ ЗАДАЧ НЕЗБАЛАНСОВАНОСТІ КЛАСІВ ТРЕНУВАЛЬНИХ ДАНИХ У СИСТЕМАХ ПРЕДИКТИВНОГО МОНІТОРИНГУ

А. Луцюк [ORCID: 0009-0000-5622-5215]

Національний університет «Львівська політехніка» вул. С. Бандери, 12, 79013, Львів, Україна

Відповідальний за рукопис: Андрій Луцюк (e-mail: andrii.v.lutsiuk@lpnu.ua).

(Подано 4 березня 2025)

У статті розглянуто підходи до форматування табличних даних (HTML, XML, Markdown, CSV) з метою подальшого генерування синтетичних зразків за допомогою великих мовних моделей (LLM) у задачах предиктивного моніторингу. Оскільки реальні дані часто характеризуються незбалансованістю класів, генерування додаткових зразків дає змогу поліпшити навчальні вибірки, підвищуючи ефективність роботи моделей. При цьому важливим постає питання швидкості обробки та вартості запитів, які значною мірою залежать від того, скільки вхідних токенів потребує обраний формат для форматування табличних даних. У межах дослідження проаналізовано витрати обчислювальних ресурсів і тривалість обробки запитів LLM залежно від формату табличних даних. Хоча, згідно із дослідженнями[1], HTML забезпечує найвищий рівень точності, він водночас вимагає суттєво більшої кількості токенів через формат подання таблиць. Така особливість суттєво збільшує об’єм вхідних даних та загальний час опрацювання запиту. Натомість менш об’ємні формати (Markdown та CSV) потребують значно меншу кількість токенів, пришвидшуючи обробку та знижуючи вартість взаємодії з моделлю. Незначне зменшення точності, в порівнянні з HTML, може виявитися прийнятним компромісом, особливо коли стоїть завдання масштабного розширення набору тренувальних даних задля компенсації нестачі прикладів нештатних станів. Такий підхід є ефективним у системах предиктивного моніторингу, де час реакції та обсяг оброблених даних безпосередньо впливають на швидкість виявлення аномалій та стійкості системи в цілому. Результати дослідження підтверджують, що Markdown і CSV, завдяки меншому об’єму вхідних даних, дають змогу зменшити тривалість обробки запитів та витрати на генерування синтетичних зразків для навчання. У той же час, HTML і XML потенційно залишаються корисними в задачах, де максимально важливим є збереження складної структури й додаткових метаданих, проте ці формати вимагають суттєвіших ресурсів. Таким чином, вибір формату подання табличних даних повинен враховувати вимоги конкретної системи й особливості робочого середовища: від апаратних обмежень і тарифікації за токени до потрібної тривалості обробки запиту.

Ключові слова: *великі мовні моделі, предиктивний моніторинг, форматування інструкцій*
УДК: 621.391

Вступ

У сучасній сфері інформаційно-комунікаційних пристроїв та систем обсяг наявних даних щороку стрімко зростає: від статистичних зведень щодо навантаження на мережу до докладних записів подій (логування), історії викликів і параметрів обладнання [2]. Обробка таких даних потребує великих ресурсів, а нормальна робота систем – постійного моніторингу. Історично, переважна більшість даних в інформаційно-комунікаційних системах зберігається у вигляді

об’ємних таблиць. Водночас, оскільки це реальні дані, що часто характеризуються незбалансованістю класів та недостатньо якісною розміткою, виникають додаткові складнощі під час застосування предиктивного моніторингу, аналізу продуктивності систем й пошуку першопричин відмов в цілому.

Поряд із тим, сучасні досягнення в галузі великих мовних моделей (LLM) дозволяють частково розв’язати проблему незбалансованості даних, зокрема шляхом генерування додаткових синтетичних зразків. Утім, навіть за високих показників у завданнях генерації й аналізу природної мови, LLM зазнають труднощів у роботі зі структурованими таблицями у їхньому класичному вигляді. Відтак виникає необхідність у спеціальному форматуванні таких даних, щоб модель могла краще розпізнавати табличну структуру та ефективно генерувати додаткові зразки, одночасно враховуючи обмеження реальних систем.

Дане дослідження присвячене пошуку ефективного способу форматування вхідних табличних даних, що сприяє скороченню часу відповіді великих мовних моделей і зменшує обчислювальні витрати. Такий підхід особливо корисний у завданнях тренування моделей для предиктивного моніторингу, оскільки правильне форматування великих табличних даних покращує процес генерування нових синтетичних зразків і зменшення незбалансованості, що зрештою підвищує ефективність процесу навчання та якість прогнозування та проактивного запобігання відмовам.

2. Аналіз останніх досліджень та публікацій

Низка останніх досліджень що розглядають обробку табличних даних великими мовними моделями концентруються на покращенні точності або семантичному збагаченні структурованої інформації. Зокрема, у роботі авторів Yuan Sui та ін. [1] акцент зроблено на порівнянні форматів подання (HTML, JSON, Markdown тощо) і різноманітних методів покращення інструкцій (role prompting, partition mark, форматні пояснення). Автори демонструють, що детальне налаштування вхідних даних може істотно покращити здатність LLM розуміти структурну логіку таблиць, проте основний показник оцінювання – точність виконання завдань (наприклад, пошук конкретної клітинки, визначення кількості рядків), а не кількісні ресурси, необхідні для досягнення цих результатів.

Водночас у інших подібних роботах [3], зокрема тих, що досліджують різні формати подачі тексту або аналізують процедуру об’єднання текстової та табличної інформації, не фокусується увага на обсягах обчислювальних ресурсів чи вартості обробки. Це стосується і розміру контексту (зокрема кількості токенів), швидкості обробки чи необхідного обсягу пам’яті. Як наслідок, питання про те, наскільки ресурсоємним є подібний підхід на практиці, лишається відкритим. Часто, коли моделі зосереджені на підвищенні точності, витрати часу, пам’яті та фінансових ресурсів для довгих вхідних рядків, великих таблиць чи багатьох ітерацій запитів не аналізуються так детально. Це створює перспективу для глибшого кількісного аналізу, який би розкрив кореляцію між точністю, швидкістю обробки та вартістю масштабування мовних моделей при інтеграції табличних даних.

Для цього у даній статті пропонується підхід, що поєднує порівняльний аналіз форматів вхідних даних із експериментальною оцінкою тривалості обробки запиту й фактично, фінансових витрат. Це дає змогу сформувати компроміс між точністю, швидкістю відповіді та економічною доцільністю, що є критично важливим для масштабних проєктів предиктивного моніторингу.

3. Дослідження форматування вхідних даних для великих мовних моделей

Одним із ключових чинників, що визначають якість та ефективність роботи великих мовних моделей, є спосіб перетворення вхідних даних на текстовий формат. Залежно від того, наскільки добре структуровано вхідний текст і якою є його довжина, модель може або коректно інтерпретувати зміст, або ж частково втратити важливий контекст. У випадку табличних даних це

стає особливо критичним, адже неправильне форматування може спотворити логічні зв'язки між заголовками стовпців і значеннями, призвести до дублювання або втрати контексту. Як наслідок, у задачах генерації синтетичних зразків це призводить до утворення нерелевантних записів, які не відповідають базовому набору даних.

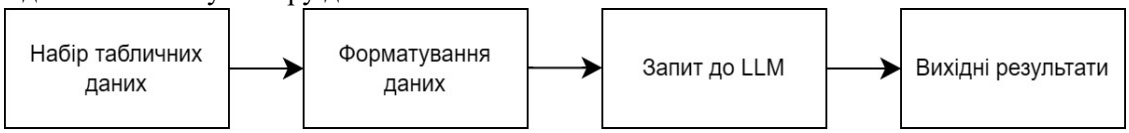

Рис.1 Процес дослідження

У сучасних дослідженнях виділяють кілька найбільш поширених підходів до форматування табличних даних. Цими підходами є HTML, XML, Markdown, CSV. Для того аби їх порівняти, в статті застосовано процес, який представлений на рисунку 1. Дослідження виконуються з використанням частини набору даних LUFlow Network Intrusion Detection Data Set[4]. LUFlow – це потоковий набір даних, що зібраний та промаркований через кореляцію зловмисної поведінки спеціально для тренування та виявлення можливих вторгнень в систему. В таблиці «опис полів набору даних» представлені усі поля набору даних та опис цих полів.

Для дослідження вибрано випадкову частину набору даних, що складається із 300 записів. В кожному з підходів, набір форматується та оцінюється його розмірність в токенах. Токенами називають мінімальні одиниці тексту, з якими працюють мовні моделі, тобто фрагменти, на які алгоритм розбиває вхідний рядок. Для прикладу, одне слово може бути розділене на декілька токенів, залежно від внутрішніх правил сегментації. Оскільки робота відбувається з моделлю GPT 4o-mini, то і кількість токенів розраховуватиметься для цієї моделі. Після цього відбувається передавання даних на вхід великої мовної моделі та розрахунок тривалості обробки запиту. Передавання даних на обробку моделі повторюється по 10 разів для кожного із підходів форматування даних.

Опис полів набору даних

Назва	Опис
src_ip	Анонімізована IP-адреса джерела
src_port	Номер порту джерела
dest_ip	Анонімізована IP-адреса призначення
dest_port	Номер порту призначення
protocol	Номер протоколу за яким працює потік
bytes_in	Кількість байтів, переданих від джерела
bytes_out	Кількість байтів, переданих від призначення
numpktsin	Кількість пакетів даних від джерела до місця призначення
numpktsout	Кількість пакетів від пункту призначення до джерела
entropy	Ентропія в бітах на байт полів даних у потоці
total_entropy	Загальна ентропія в байтах за всіма байтами в полях даних потоку
mean_ipt	Середнє значення міжпакетного часу прибуття потоку
duration	Час тривалості потоку з точністю до мікросекунди
label	Маркування потоку

3.1. Форматування із використанням HTML розмітки

HTML вважається одним із найпоширеніших форматів представлення таблиць у мережі, завдяки чому великі мовні моделі здатні розуміти таку структуру. Це пояснюється тим, що значна

частина даних, на яких навчаються моделі, походить із веб-сторінок, де таблиці формуються саме засобами HTML[5].

Особливістю HTML є чітка система тегів, які однозначно окреслюють логічну побудову таблиці. Такими тегами є <table>, <tr>, <th>, <td>. Така структура й поширеність формату дає низку переваг. По-перше, мовні моделі можуть краще відрізнити заголовки від безпосередніх значень, що зменшує ризик плутанини між категорією та конкретною інформацією. По-друге, за наявності HTML-розмітки простіше обробляти навіть великі обсяги даних: парсери можуть швидко відокремити один рядок від іншого й виділити кожен стовпець із потрібними даними.

Водночас, одним із очевидних недоліків використання HTML є збільшення обсягу тексту, у порівнянні з іншими форматами, на кшталт CSV або Markdown[6]. На рисунку 2 зображено порівняння кількості токенів для кожного із форматів. Підхід з використанням HTML не є найоптимальнішим варіантом з точки зору вхідного об’єму даних, оскільки майже втричі більший від CSV формату, та на 80% об’ємніший від Markdown. Водночас з точки зору швидкості обробки вхідних даних (рисунок 3), LLM, в середньому, обробляє запит із таким набором даних на 64% повільніше, а ніж запити із CSV форматом.

3.2. Форматування із використанням XML розмітки

XML – гнучкий та широко розповсюджений формат опису структурованих даних, який дозволяє користувачам визначати власні теги під конкретні потреби. На відміну від HTML із фіксованим набором тегів, XML пропонує розробникам повну свободу у визначенні назви та змісту елементів, що надає більшої гнучкості для моделей і прикладних програм. У контексті таблиць, кожен рядок може бути представлено як окремий елемент із вкладеними полями, які відповідають стовпцям. Завдяки цьому структура даних описується прозоро, а дані з різним типом вмісту (наприклад, тексти, числа, дати) можуть бути виділені індивідуальними тегами.

Водночас, XML, як і HTML, може суттєво розширювати обсяг розмітки, адже кожен рядок і кожне поле вимагають відкриття та закриття тегів, що збільшує кількість токенів і, відповідно, обчислювальні витрати. Оскільки у XML немає спеціалізованих елементів для таблиць, логіку рядків і стовпців доводиться визначати додатково, а велика кількість вкладених тегів та атрибутів ускладнює й збільшує тривалість очікування відповіді від моделі.

Рисунок 2 промовисто демонструє проблему XML із збільшенням використання токенів у процесі роботи LLM. Із усіх представлених варіантів, XML є найбільшим. Представлення табличних даних з його допомогою є на 56% об’ємнішим, а ніж найближчий наступний показник, яким є HTML. Тривалість обробки (рисунок 3) складає в середньому 6,3 с.

3.3. Форматування із використанням Markdown

Markdown є одним із найпоширеніших форматів компактної розмітки, що дає змогу представляти дані у вигляді таблиць, використовуючи символи-роздільники, як-от «|» та «-». Такий підхід має перевагу в зручності створення й редагування файлів: навіть без спеціальних інструментів можна легко «окреслити» таблицю, змінити заголовки або додати нові рядки[7]. Крім того, Markdown значно компактніший, ніж HTML або XML, адже не потребує додаткових тегів і складної внутрішньої ієрархії.

Проте, порівняно з повноцінними мовами розмітки, Markdown є менш формальним і немає жорстких правил щодо того, як саме відображати складні комірки, злиття рядків чи вкладену структуру. Це може викликати непорозуміння у великих мовних моделей, особливо коли таблиці великі або містять багаторівневі заголовки[8]. Крім того, під час перетворення таких таблиць у внутрішнє подання LLM є ризик некоректного розрізнення заголовків і самих даних, що потенційно знижує якість аналізу.

Але з точки зору об’єму та використання ресурсів, Markdown є одним із найкращих варіантів. Він використовує невелику токенів (рисунок 2) і забезпечує найнижчу тривалість обробки

моделлю. Окрім того, у досліджені [1], де порівнюють точність із використанням різних форматів, для завдання верифікації фактів на основі таблиці (TabFact), Markdown показав значення 68.4%, у той час як HTML показав 71.3%. Але, слід зазначити, що Markdown використовує на 80% менше токенів для кожного свого запиту у порівнянні із HTML.

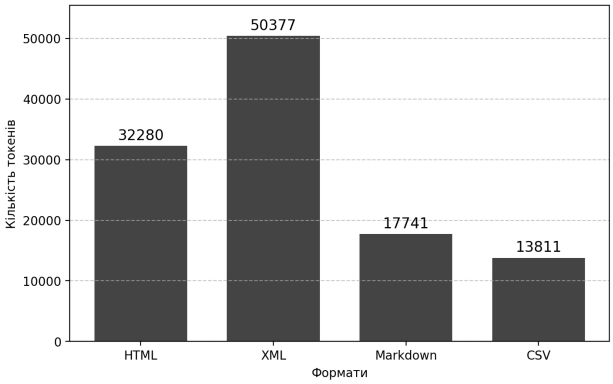

Рис.2 Використання токенів для різних форматів представлення вхідних табличних даних у процесі роботи з LLM

3.4. Форматування із використанням CSV

Comma-Separated Values (CSV) є одним із найпростіших і найпоширеніших форматів зберігання та передачі табличних даних. Його популярність зумовлена передусім мінімальною розміткою – кожен рядок відповідає запису, а стовпці розділяються комами або іншим спеціальним символом (наприклад, «;» або «|»)[9]. Це дає змогу легко формувати та читати файли, а також працювати з різноманітними інструментами обробки даних. У контексті великих мовних моделей використання CSV забезпечує найменшу надлишковість у порівнянні з усіма іншими розглянутими методами форматування, що скорочує загальний розмір тексту і сприяє економії обчислювальних ресурсів[10].

Головний недолік CSV полягає в обмеженій семантиці даних: структура таблиці не має чіткої прив’язки до назв стовпців (окрім того, що їх можна винести в заголовок), а вкладені чи злиті комірки не можуть бути коректно представлені. Для моделі, що намагається зрозуміти логіку таблиці, відсутність явних тегів іноді ускладнює розрізнення заголовків, рядків і стовпців, особливо коли в даних трапляються пропуски або спеціальні символи (на кшталт лапок чи додаткових ком). Внаслідок цього великі мовні моделі можуть плутатися у визначенні меж стовпців і підтипів комірок, що призводить до потенційних помилок під час аналізу.

Експеримент із CSV засвідчив найкращі результати у використанні вхідних токенів (рисунок 2) та близький результат із Markdown за часом обробки (рисунок 3). Для завдання TabFact, Mardown показав значення 68.4%, у той час як CSV – 70.26%. А вже у іншому досліді (HybridQA), Mardown показав себе краще відносно CSV та мав значення 45.88% проти 45.02%. Обидва підходи загалом поступаються HTML за точністю, проте між собою є співставними.

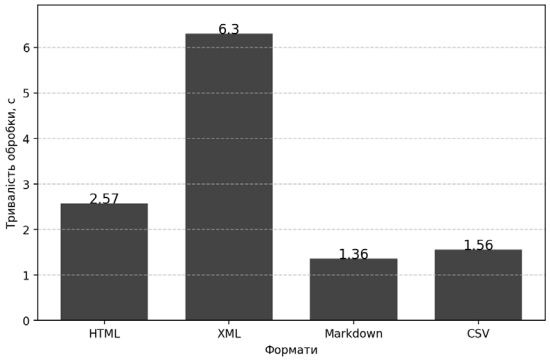

Рис.3 Тривалість обробки запиту до LLM для різних форматів представлення вхідних табличних даних

Висновки

Проведене дослідження різних форматів представлення табличних даних (HTML, XML, Markdown, CSV) в контексті генерування синтетичних зразків для навчання моделей предиктивного моніторингу інформаційно-комунікаційних систем продемонструвало, що HTML, хоч і дає найвищу точність [1], потребує суттєвих обчислювальних ресурсів через більший обсяг використання токенів. Натомість формати з нижчою надлишковістю на кшталт Markdown і CSV споживають менше токенів, що сприяє швидшій обробці й знижує вартість запитів у середовищах із тарифікацією за кількістю токенів чи за тривалістю обробки.

Для предиктивних моніторингових систем, де першорядну роль відіграють оперативне генерування й аналіз великих масивів даних, Markdown та CSV можуть стати оптимальним компромісом: показники точності виявляються дещо нижчими, ніж у HTML, проте здобута вигода у швидкості та економічній доцільності суттєво підвищує ефективність навчання моделей. Такий баланс між рівнем точності та помірним споживанням ресурсів робить згадані формати перспективними для впровадження великих мовних моделей у контексті компенсації незбалансованості класів вхідних даних для навчання моделей у системах предиктивного моніторингу.

Список використаних літературних джерел

[1]. Sui, Y., Zhou, M., Zhou, M., Han, S. and Zhang, D. (2024), "Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study", Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), 4–8 March, Mérida, Yucatán, Mexico. ACM.

[2]. Луцюк, А.В., 2024. "Предиктивний моніторинг інформаційно-комунікаційних систем за допомогою спеціалізованої моделі машинного навчання". Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки, 35(74)(6, ч.1), с. 129–135.

[3]. Aghajanyan, A., Okhonko, D., Lewis, M., Joshi, M., Xu, H., Ghosh, G. and Zettlemoyer, L. (2022) 'HTLM: Hyper-Text Pre-Training and Prompting of Language Models', 10th International Conference on Learning Representations (ICLR 2022), 25-29 April.

[4]. Mills, R. (2025) "LUFlow Network Intrusion Detection Data Set", Kaggle [Data set]. Available at: https://doi.org/10.34740/KAGGLE/DSV/11027911 (Accessed: 15 February 2025).

[5]. Chen, W. (2023) "Large Language Models Are Few(1)-Shot Table Reasoners", Findings of the Association for Computational Linguistics: EACL 2023, 2 April.

[6]. Dong, H., Cheng, Z., He, X., Zhou, M., Zhou, A., Zhou, F., Liu, A., Han, S. and Zhang, D. (2022) 'Table Pre-training: A Survey on Model Architectures, Pre-training Objectives, and Downstream Tasks', Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22), 23–29 July, Vienna, Austria.

[7]. Eisenschlos, J.M., Gor, M., Müller, T. and Cohen, W.W. (2021) "MATE: Multi-view Attention for Table Transformer Efficiency", Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021), 7–11 November, Punta Cana, Dominican Republic. Association for Computational Linguistics.

[8]. Herzig, J., Nowak, P.K., Müller, T., Piccinno, F. and Eisenschlos, J. (2020) 'TaPas: Weakly supervised table parsing via pre-training', Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), 5–10 July, pp. 4320–4333. Association for Computational Linguistics.

[9]. Hulsebos, M., Demiralp, Ç. and Groth, P. (2023) 'GitTables: A Large-Scale Corpus of Relational Tables', Proceedings of the ACM on Management of Data, 1(1), pp. 1–17.

[10]. Iida, H., Thai, D., Manjunatha, V. and Iyyer, M. (2021) 'TABBIE: Pretrained Representations of Tabular Data', arXiv preprint. Available at: https://doi.org/10.48550/arXiv.2105.02584 (Accessed: 15 February 2025).

EFFICIENCY OF LLM INSTRUCTION FORMATS FOR CLASS IMBALANCE PROBLEMS IN TRAINING DATA FOR PREDICTIVE MONITORING SYSTEMS

Andrii Lutsiuk

Lviv Polytechnic National University, S. Bandery Str., 12, 79013, Lviv, Ukraine

The article examines approaches to formatting tabular data (HTML, XML, Markdown, CSV) for the subsequent generation of synthetic samples using large language models (LLM) in predictive monitoring tasks. Since real-world data are often characterized by class imbalance, generating additional samples helps improve training datasets, thereby enhancing the effectiveness of models. At the same time, an important issue arises regarding processing speed and query costs, which largely depend on how many input tokens are required by the chosen format for tabular data representation. The study analyzes computational resource consumption and query processing time for LLMs depending on the tabular data format. Although, according to research [1], HTML provides the highest level of accuracy, it also requires a significantly larger number of tokens due to its table representation format. This characteristic considerably increases the volume of input data and the overall query processing time. In contrast, less bulky formats (Markdown and CSV) require significantly fewer tokens, speeding up processing and reducing the cost of interaction with the model. A slight reduction in accuracy compared to HTML may be an acceptable trade-off, especially when the goal is to significantly expand the training dataset to compensate for the lack of examples of non-standard conditions. This approach proves to be effective in predictive monitoring systems, where response time and the volume of processed data directly affect the speed of anomaly detection and overall system resilience. The study results confirm that Markdown and CSV, due to their smaller input data volume, help reduce query processing time and the costs associated with generating synthetic training samples. At the same time, HTML and XML remain potentially useful in tasks where preserving complex structures and additional metadata is of utmost importance, although these formats require significantly more resources. Thus, the choice of a tabular data representation format should take into account the specific system requirements and operational environment characteristics, ranging from hardware limitations and token-based pricing to the required query processing time.

Keywords: *large language model, predictive monitoring, prompt formatting*