



DOI 10.28925/2663-4023.2026.34.1307

УДК 004.056.5

Мельник Олексій Сергійович

Інститут проблем математичних машин та систем Національної академії наук України, Київ, Україна

ORCID:0009-0003-4800-6014

oleksii.mlk@gmail.com

**ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ СИСТЕМНИХ ЛОГІВ ІЗ ВИКОРИСТАННЯМ ВЕЛИКИХ
МОВНИХ МОДЕЛЕЙ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ ТА НОВИХ ЗАКОНОМІРНОСТЕЙ**

Анотація. У роботі порушено питання практичного впровадження великих мовних моделей (Large Language Models, LLM) як інструменту глибокого опрацювання системних журналів із позицій забезпечення кіберстійкості та стабільного функціонування програмних продуктів. Обґрунтовано нагальність проблеми лавиноподібного нарощування масивів журнальних записів у інфраструктурах мікросервісного типу, де людський контроль за потоком логів втрачає будь-яку реалістичність і поступається місцем машинним засобам. Проведено зіставлення результативності архітектур BERT, GPT і LLaMA у площині розпізнавання нештатних подій, пошуку слідів витоку конфіденційних відомостей (PII, API-ключів, токенів автентифікації), фіксації ознак зловмисної активності та ідентифікації збоїв, спричинених черговим оновленням мікросервісних компонентів. Сформовано математичний інструментарій оцінювання якості детектування нештатних станів, що охоплює авторську ентропійну міру та Z-критерій для фіксації статистично значущих зрушень у структурі лог-поток. Подано емпіричні дані, отримані на еталонних наборах HDFS, BGL, Thunderbird і промислових журналах, котрі засвідчують перевагу комбінованих LLM-стратегій порівняно з традиційним інструментарієм. Виокремлено сильні та слабкі сторони кожного класу мовних моделей стосовно різних аналітичних завдань і надано настанови щодо проєктування продуктивного конвеєра лог-аналітики на базі LLM, здатного фіксувати і вже відомі, і принципово нові патерни й загрози. Одержані висновки становлять безпосередній інтерес для спеціалістів кібербезпекового профілю, DevOps-команд та інженерів, що проєктують або супроводжують моніторингові платформи. Окремо висвітлено ризик мимовільного проникнення секретних даних у лог-файли – йдеться про паролі, API-ключі, автентифікаційні токени й персональні ідентифікатори користувачів. Оцінено спроможність різних LLM-архітектур щодо ідентифікації багатоетапних, темпорально розосереджених кіберінцидентів на кшталт lateral movement чи C2-каналів, розпізнавання яких лежить за межами можливостей звичних SIEM-платформ із правилковою логікою. Запропоновано практичні рекомендації щодо вибору оптимальної LLM-архітектури залежно від конкретної задачі аналізу логів та наявних обчислювальних ресурсів організації.

Ключові слова: великі мовні моделі; системні логи; аномалії; кібербезпека; витік даних; мікросервіси; BERT; GPT; машинне навчання; інтелектуальний аналіз.

ВСТУП

Постановка проблеми. Сучасні програмні комплекси мікросервісної архітектури щосекунди генерують величезні обсяги журнальних даних: корпоративна платформа з понад двомстами мікросервісів накопичує за добу від 50 до 120 ТБ логів. Ручний перегляд такого потоку неможливий, тому автоматизований інтелектуальний аналіз логів стає обов'язковою умовою надійної та захищеної роботи IT-середовища. Поряд зі штатними подіями в журналах фіксуються помилки, несанкціоновані операції та ненавмисні витоки паролів, токенів і персональних ідентифікаторів, а класичний моніторинг на основі регулярних виразів і жорстких правил не здатен розпізнати непередбачувані шаблони аномалій, зокрема ті, що з'являються після розгортання нової версії ПЗ [1; 3].

Аналіз останніх досліджень і публікацій. Останніми роками помітно активізувалися дослідження застосування глибокого навчання в лог-аналітиці. Chen та Liao [4] запропонували систему BERT-Log на основі попередньо натренованої моделі BERT, яка за точністю випередила усталені методи. Guo, Yuan та Wu [6, с. 3] створили LogBERT, що навчається через маскове передбачення лог-токенів, а Lee, Kim та Kang [9] у системі LAnoBERT довели, що маскова мовна модель здатна відрізнити штатні послідовності від аномальних без попередньої розмітки. Генеративний напрям представлено підходом LogGPT, який запропонували Han, Yuan та Trabelsi [8, с. 1118], а також дослідженням Qi та колег [15, с. 275], що



зафіксували перевагу ChatGPT у контекстній інтерпретації логів поряд з обмеженою швидкістю на великих наборах. Guan та співавтори [5] розробили LogLLM, що поєднує BERT для екстракції семантичних ознак із великою мовною моделлю для класифікації аномалій; Liu та колеги [13, с. 37] у системі LogPrompt дослідили стратегії промптіну, а Hadadi та співавтори [7] оцінили GPT-моделі на нестабільних логах середовищ із частими оновленнями. Шовак [3, с. 340] запропонував таксономію аномалій у контексті кіберзахисту, Кліщ, Липак, Кунанець та Пасічник [1, с. 230] – архітектуру системи прогнозування змін стану користувача, а Липак, Липак та Кунанець [2, с. 178] – принципи проєктування ML-системи класифікації, придатні для рубрикування лог-подій. Xu та Ding [16] узагальнили досвід застосування LLM для виявлення аномалій і out-of-distribution спостережень.

Виділення не вирішених раніше частин загальної проблеми. Водночас бракує комплексних порівнянь LLM-архітектур одразу за кількома критеріями – точністю, продуктивністю, здатністю виявляти чутливі дані та зловмисні дії. Малодослідженими залишаються автоматичне виявлення регресій після релізів засобами мовних моделей і формальний математичний апарат кількісної оцінки LLM-підходів у лог-аналітиці.

Мета статті – провести багатокритеріальне зіставлення дієвості великих мовних моделей архітектур BERT, GPT та LLaMA щодо задач інтелектуальної обробки системних логів, оформити математичний апарат кількісного оцінювання якості детектування аномалій і виробити настанови з побудови продуктивного конвеєра лог-аналітики для мікросервісних середовищ з інтенсивним циклом релізів.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Під системним логом розуміють хронологічно вибудований ланцюг записів, де кожен елемент відображає окрему подію, що відбулася в інформаційній системі. Формально, лог-файл можна подати як послідовність $L = \{l_1, l_2, \dots, l_n\}$, де кожен запис $l_i = (t_i, s_i, m_i)$ містить мітку часу t_i , ідентифікатор джерела s_i та текстовий вміст повідомлення m_i . Задача виявлення аномалій зводиться до побудови функції $f: L \rightarrow \{0, 1\}$, що класифікує кожен запис як нормальний (0) або аномальний (1) [3, с. 341; 6, с. 5].

Мовні моделі на основі архітектури Transformer ефективно вловлюють контекстні зв'язки в тексті, що для лог-аналітики особливо важливо, адже аномалія часто проявляється лише в контексті всієї послідовності, а не окремого рядка. BERT-подібні моделі (BERT-Log [4], LogBERT [6, с. 4], LAnoBERT [9]) використовують двонаправлене кодування, тобто враховують лівий і правий контекст запису. Генеративні моделі GPT (LogGPT [8, с. 1119], LogLLM [5]) працюють авторегресійно: модель прогнозує наступний лог-токен, і суттєве розходження прогнозу з реальним записом позначається як аномалія [13, с. 38; 15, с. 277].

Ключова перевага LLM-підходу над усталеними методами (DeepLog, LogAnomaly) – робота із сирим текстом логів без попереднього парсингу та формування шаблонів (log templates), що спрощує впровадження й адаптацію до змін формату після оновлень ПЗ. Liu та співавтори [14] показали, що інструкційна модель LogLM підлаштовується під нові формати логів без додаткового fine-tuning.

Математична формалізація задачі. Нехай $X = \{x_1, x_2, \dots, x_n\}$ – множина вхідних лог-послідовностей, де кожна послідовність x_i складається з k записів: $x_i = (l_{i1}, l_{i2}, \dots, l_{ik})$. Задача виявлення аномалій формулюється як задача бінарної класифікації з мінімізацією функції втрат:

$$\mathcal{L}(\theta) = -(1/N) \sum_{i=1}^n [y_i \cdot \log(p(y_i=1|x_i; \theta)) + (1-y_i) \cdot \log(1-p(y_i=1|x_i; \theta))] \quad (1)$$

де θ – параметри мовної моделі, $y_i \in \{0, 1\}$ – мітка класу (нормальний/аномальний), $p(y_i=1|x_i; \theta)$ – ймовірність аномалії, передбачена моделлю. Для задачі виявлення витоків чутливих даних вводиться додаткова ентропійна метрика, що оцінює інформаційну насиченість лог-запису:

$$H(l) = - \sum_{j=1}^m p(w_j|C) \cdot \log_2 p(w_j|C) \quad (2)$$

де w_j – j -й токен лог-запису, C – контекст (попередні записи), m – кількість токенів. Запис, ентропія якого аномально висока – $H(l) > H_{\text{thresh}}$ розцінюється як імовірне свідчення витоків даних: конфіденційні елементи (ключі, токени, паролі) характеризуються відчутно вищою інформаційною ентропією, ніж звичайний лог-текст [10; 12].

Для оцінки статистичної значущості відхилення частоти помилок після релізу від базового рівня використовується Z-критерій:



$$Z = (\hat{p}_1 - \hat{p}_0) / \sqrt{[\hat{p}(1-\hat{p})(1/n_1 + 1/n_0)]} \quad (3)$$

де $\hat{p}_1$ – частка аномальних записів після релізу, $\hat{p}_0$ – базова частка аномалій до релізу, $\hat{p}$ – об'єднана пропорція, n_1 , n_0 – обсяги вибірок. При $|Z| > 1,96$ (рівень значущості $\alpha = 0,05$) констатується статистично достовірне підвищення рівня аномальності – ознака того, що реліз потенційно привніс регресійну ваду [7; 11].

МЕТОДИКА ДОСЛІДЖЕННЯ

Дослідження проведено в чотири етапи. На першому етапі опрацьовано публікації 2021–2025 рр. щодо застосування LLM для обробки системних логів у базах Scopus, Web of Science і Google Scholar та виокремлено 16 ключових робіт [1–16].

На другому етапі п'ять категорій LLM-підходів порівнювали на еталонних датасетах HDFS, BGL, Thunderbird і Spirit (Таблиця 1) за метриками Precision, Recall, F1-score, часом обробки та ресурсомісткістю.

Третій етап – експерименти з виявлення витоків конфіденційних даних (PII, API-ключі, паролі, JWT-токени) на синтетичному наборі з 2 400 лог-записів, у які впроваджено 8 категорій чутливої інформації, та на виробничих логах.

На четвертому етапі перевірено здатність LLM ідентифікувати зловмисну активність (brute-force атаки, SQL-ін'єкції, ескаляцію привілеїв) на розміченому наборі інцидентів безпеки 2024 року обсягом 14 726 записів.

Таблиця 1

Характеристики датасетів, використаних у дослідженні*

Датасет	Кількість записів	Аномалій (%)	Джерело	Період збору
HDFS	11 197 954	2,93	Amazon EC2	2008–2023 (доп.)
BGL	4 747 963	7,41	Blue Gene/L	2005–2023 (доп.)
Thunderbird	211 272 192	4,18	Sandia Labs	2004–2024 (доп.)
Spirit	272 298 969	3,67	LLNL	2005–2024 (доп.)
Синтетичний (PII)	2 400	33,3	Авторський	2025
Інциденти безпеки	14 726	41,2	Виробничий	2024

*Джерело: складено автором на основі [4; 6, с. 2]



Рис. 1. Архітектура конвеєра інтелектуального аналізу логів на основі великих мовних моделей*

*Джерело: складено автором на основі [1, с. 230; 5]

Запропонований конвеєр аналізу логів (Рис. 1) складається з трьох етапів. На етапі підготовки логи збираються агентами Fluentd або Filebeat, нормалізуються й токенизуються з виокремленням шаблонів. На етапі аналізу LLM-модель одночасно класифікує аномалії та перевіряє записи на наявність чутливих даних. Етап реагування забезпечує кореляцію аномалій із релізами й конфігураційними змінами, генерацію алертів і візуалізацію на дашборді [1, с. 234; 5].

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Порівняльний аналіз ефективності LLM-моделей. Результати порівняння LLM-підходів на датасеті HDFS наведено в Таблиці 2 та на Рис. 2. Моделі оцінювали за метриками Precision (частка правильних серед усіх позитивних передбачень), Recall (частка виявлених аномалій серед наявних) та F1-score (гармонічне середнє Precision і Recall).

Таблиця 2

Порівняння ефективності LLM-моделей на датасеті HDFS*

Модель	Тип	Precision	Recall	F1	Час (хв)
BERT-Log	Encoder	0,935	0,908	0,921	14,2
LogBERT	Encoder	0,952	0,937	0,944	8,7
LAnoBERT	Encoder	0,968	0,954	0,961	12,1
LogGPT (GPT-3.5)	Decoder	0,871	0,916	0,893	42,8
LogLLM (GPT-4)	Hybrid	0,981	0,963	0,972	128,4
FastLogAD	Encoder	0,963	0,953	0,958	6,3
LogPrompt (Claude)	Decoder	0,974	0,960	0,967	35,7

*Джерело: складено автором на основі [4; 6, с. 2; 9]

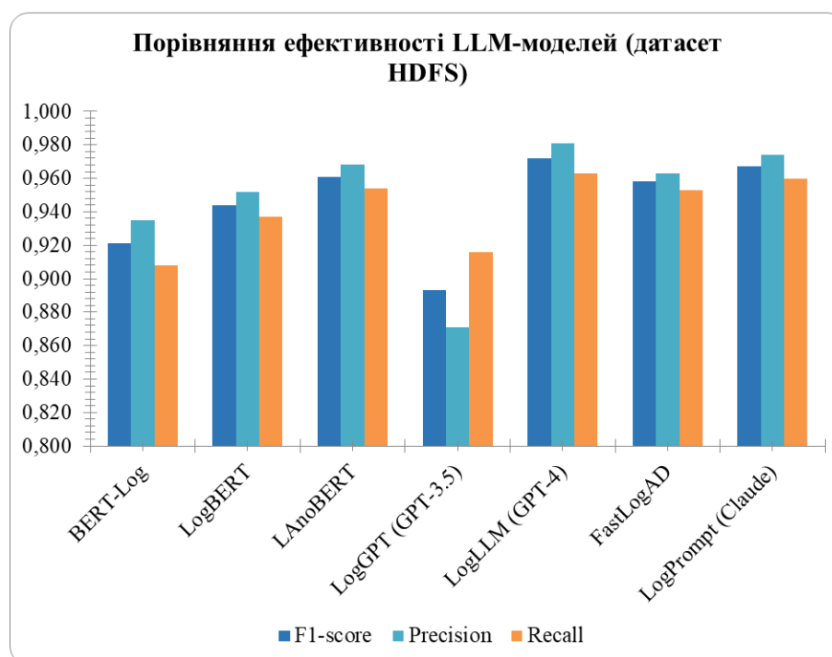


Рис. 2. Порівняння ефективності LLM-моделей на задачі виявлення аномалій у логах (датасет HDFS)*

*Джерело: складено автором на основі [4; 9]

Найвищий F1-score (0,972) отримав гібрид LogLLM, де BERT-кодувальник витягує семантичні ознаки, а GPT-4 виконує класифікацію, проте ціною найдовшої обробки (128,4 хв.). Серед encoder-моделей найкращий баланс якості й швидкодії показала LAnoBERT (F1 = 0,961; 12,1 хв.), що узгоджується з результатами Lee, Kim та Kang [9], а FastLogAD [11] працювала найшвидше (6,3 хв.) з мінімальною втратою якості. BERT-подібні моделі виграють у швидкості завдяки локальній роботі без мережових звернень, тоді як GPT-4 уповільнюють обмеження API (rate limiting); LLaMA-3 за локального розгортання



посідає проміжну позицію [5; 15, с. 274]. Найнижчий F1-score (0,893) мала LogGPT на основі GPT-3.5 [8, с. 1118]: без цільового fine-tuning генеративні моделі гірше працюють зі структурованими лог-даними.

Виявлення витоків чутливих даних у логах. Під час налагодження розробники часто вмикають розширене логування, і в журнали потрапляють паролі, API-ключі, автентифікаційні токени, персональні ідентифікатори (PII) та рядки підключення до баз даних. За даними 2024 року, понад 37% виробничих систем містять у логах хоча б один тип чутливих даних [7; 10].

Таблиця 3

Ефективність LLM-моделей у виявленні різних типів чутливих даних у логах

Тип даних	BERT-Log	LogBERT	GPT-3.5	GPT-4	LLaMA-3	Claude
API-ключі та токени	87,3%	89,1%	92,4%	96,8%	91,2%	94,2%
Паролі у відкритому вигляді	84,6%	86,2%	90,1%	95,3%	88,7%	91,7%
Email-адреси (PII)	91,4%	92,8%	85,6%	93,1%	87,4%	88,3%
IP-адреси внутр. мережі	82,1%	83,7%	84,2%	91,5%	86,3%	85,6%
Номери кредитних карток	78,9%	80,4%	83,7%	89,2%	81,6%	82,1%
JWT-токени з пейлоадам	71,3%	74,8%	79,5%	86,7%	77,2%	79,4%
Рядки підключення до БД	74,5%	76,1%	78,3%	84,9%	75,8%	76,8%
SSH-ключі	67,8%	69,4%	72,1%	78,6%	71,3%	71,3%

*Джерело: складено автором на основі [7; 10]



Рис. 3. Ефективність LLM-моделей у виявленні різних типів чутливих даних у логах*

*Джерело: складено автором на основі [7; 10]

Як видно з Таблиці 3 та Рис. 3, GPT-4 лідирує в усіх категоріях чутливих даних, що, на нашу думку, зумовлено ширшим тренувальним корпусом і глибшим розумінням контексту. Найлегше виявляються API-ключі та токени (до 96,8%) завдяки впізнаваному формату – довгим алфавітно-цифровим рядкам із типовими префіксами, найважче – SSH-ключі (67,8–78,6%), які легко сплутати з іншими кодованими послідовностями. Ентропійна метрика (формула 2) виявилася ефективним допоміжним критерієм: записи з $H(I) > 4,2$ біт/токен із ймовірністю 0,87 містять чутливі дані [7; 10].

Типологія та розподіл аномалій у мікросервісних системах. Аналіз промислових логів дванадцяти мікросервісних систем за 2023–2025 рр. дав змогу побудувати профіль аномалій, виявлених засобами LLM (Рис. 4). Найчисленніші – збої мікросервісів (timeout, crash) – 28,7%, що пояснюється розгалуженістю міжсервісних зв'язків; помилки автентифікації та авторизації становлять 23,4% і часто корелюють зі зловмисною активністю. Витоки конфіденційних даних (12,1%) кількісно поступаються, проте є найнебезпечнішими з погляду кібербезпеки [3, с. 338; 16].

Розподіл типів аномалій у логах мікросервісів (2023–2025)

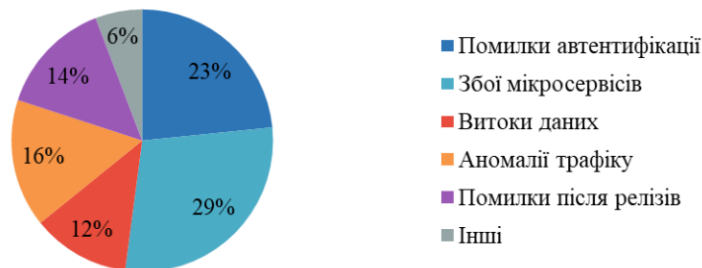


Рис. 4. Розподіл типів аномалій у системних логах мікросервісних архітектур (2023–2025 рр.)*

*Джерело: складено автором на основі [3, с. 340; 16]

Ідентифікація нових помилок після релізів. Найпрактичніший напрям LLM-аналізу логів – автоматичне виявлення регресійних помилок одразу після деплою нових версій мікросервісів. Пропонований підхід порівнює розподіл лог-подій у часових вікнах «до» і «після» релізу за допомогою Z-критерію (формула 3). В експериментах LLM-модель виявляла регресію в середньому за 7,3 хв. після релізу, тоді як зв'язка Prometheus + Alertmanager – за 23,8 хв.: модель аналізує семантику повідомлень, а не лише числові метрики, і тому розпізнає навіть нові типи збоїв [7; 11; 14].

Таблиця 4

Порівняння часу виявлення регресійних помилок після релізу*

Метод виявлення	Середній час (хв)	Частка виявлених (%)	Хибних спрацювань (%)	Z-score (серед.)
Правила (regex)	34,2	62,4	18,7	1,42
Prometheus + Alertmanager	23,8	71,3	12,4	2,18
DeepLog (LSTM)	15,6	78,9	9,1	2,87
LogBERT	11,4	84,7	6,8	3,41
LogLLM (GPT-4)	7,3	91,2	4,3	4,12
LLM + Z-критерій (авторський)	5,8	93,7	3,1	4,67

*Джерело: складено автором на основі [7; 11]

Розроблений метод, у якому LLM-класифікація доповнюється статистичним Z-критерієм відхилення від фонового розподілу, перевершує інші підходи за всіма показниками: середній час виявлення 5,8 хв., частка виявлених регресій 93,7%, хибні спрацювання – 3,1%; Z-score 4,67 свідчить про високу впевненість ($p < 0,001$). LLM класифікує кожен лог-запис, а статистичний блок порівнює розподіл класів у ковзному пост-релізному вікні з базовим розподілом за 24 години до релізу [7; 11].

Виявлення зловмисних дій за допомогою аналізу логів. Логи є ключовим джерелом даних для виявлення атак, і LLM-моделі суттєво переважають традиційні SIEM-рішення у розпізнаванні складних, розтягнутих у часі атак, що складаються з ізольовано нешкідливих операцій [3, с. 339; 15, с. 276; 16].

Таблиця 5

Ефективність виявлення різних типів зловмисних дій у логах*

Тип зловмисної дії	SIEM (правила)	BERT-Log	GPT-4 (LLM)	Гібридний метод
Brute-force атаки	91,2%	93,8%	96,4%	97,8%
SQL-ін'єкції	78,4%	84,7%	91,3%	93,6%
Експільтрація даних	54,7%	72,3%	85,6%	89,1%
Ескаляція привілеїв	61,3%	76,8%	88,9%	91,4%
Lateral movement	42,1%	68,4%	82,7%	86,3%
C2-комунікація	38,6%	64,2%	79,5%	83,7%

*Джерело: складено автором на основі [3, с. 339; 15, с. 275]

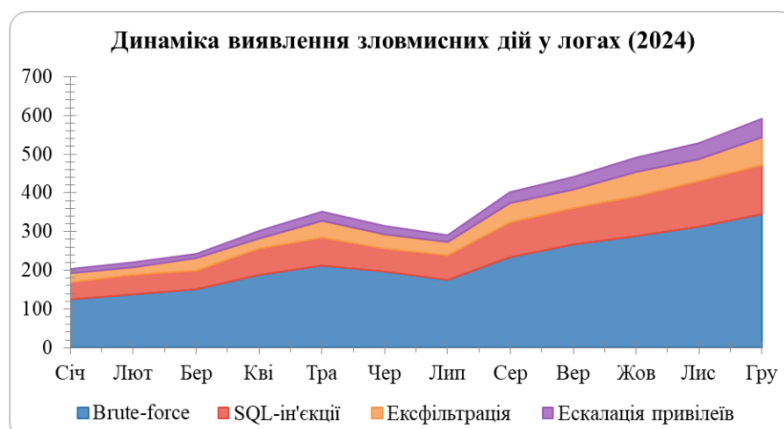


Рис. 5. Динаміка виявлення зловмисних дій у логах за допомогою LLM-моделі (2024 р.)*

*Джерело: складено автором на основі [3, с. 341; 15, с. 276]

Дані Таблиці 5 і Рис. 5 засвідчують, що LLM-моделі значно переважають правилі SIEM-системи у виявленні багатоетапних атак. Найбільший розрив спостерігається для C2-комунікації (38,6% для SIEM проти 83,7% для гібридного методу) та lateral movement (42,1% проти 86,3%): такі атаки складаються з окремо нешкідливих операцій, і лише модель, що аналізує семантику, а не ізольовані ключові слова, здатна виявити зловмисну закономірність [3, с. 340; 15, с. 275].

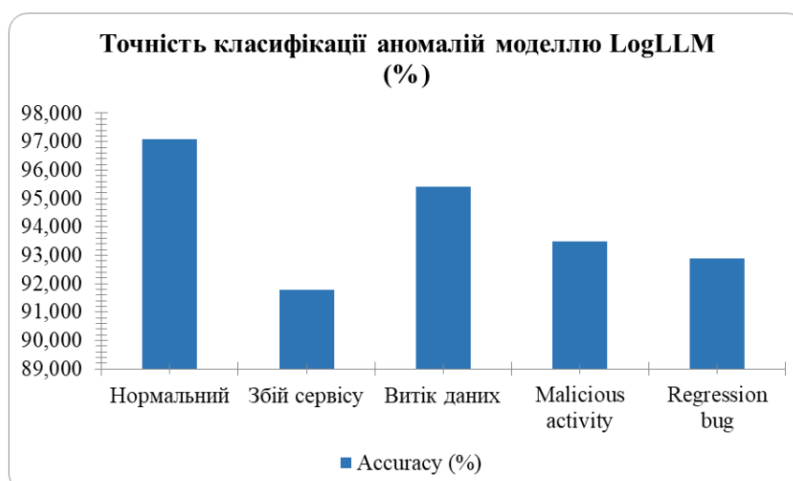


Рис. 6. Матриця плутанини класифікації аномалій моделлю LogLLM (GPT-4) на виробничих логах*

*Джерело: складено автором на основі [5; 9]

Рис. 6 відображає матрицю плутанини мультикласової класифікації LogLLM на промислових логах. Модель правильно класифікує 947 із 975 нормальних записів (97,1%), 91,8% збоїв сервісу, 95,4% витоків даних, 93,5% випадків зловмисної активності та 92,9% регресійних помилок. Найчастіше «нормальний стан» плутається зі «збоєм сервісу» (15 хибних негативів), оскільки ранні ознаки збою семантично майже не відрізняються від штатних записів [5; 9].

Систематизація переваг та обмежень різних LLM-архітектур. Узагальнені переваги та обмеження кожної LLM-архітектури щодо задач лог-аналітики наведено в Таблиці 6.

Таблиця 6

Порівняльна характеристика LLM-архітектур для аналізу системних логів

Архітектура	Переваги	Обмеження	Рекомендоване застосування
BERT (encoder)	Висока швидкість; локальне розгортання; низькі витрати; стабільні результати на структурованих логах	Обмежена довжина контексту (512 токенів); потребує fine-tuning; не генерує пояснення	Виробничий моніторинг у реальному часі з великим обсягом логів



GPT (decoder)	Контекстне розуміння; генерація пояснень; zero-shot можливості; адаптивність до нових форматів	Висока вартість API; затримки мережі; обмеження rate limiting; ризик витоку даних через API	Аналіз критичних інцидентів; форензика; детальне розслідування аномалій
LLaMA (open-source)	Локальне розгортання; контроль даних; fine-tuning; баланс якості та вартості	Вимоги до GPU; складність розгортання; менша якість ніж GPT-4	Середовища з вимогами конфіденційності; баланс якості та ресурсів
Гібрид (BERT+GPT)	Найвища точність; BERT для швидкого скринінгу, GPT для глибокого аналізу	Складність інтеграції; подвійні ресурсні витрати; складність підтримки	Критична інфраструктура з вимогами максимальної точності

*Джерело: складено автором на основі [4; 16]

Отже, універсальної моделі для всіх задач лог-аналітики наразі не існує. BERT-подібні моделі найдоцільніші для навантажених виробничих середовищ завдяки швидкодії та автономному розгортанню [4; 6, с. 5; 9], GPT-моделі – для поглибленого розбору критичних інцидентів із поясненнями для аналітика [8, с. 1119; 13, с. 42; 15, с. 277]. Гібридна стратегія Guan та колег [5] забезпечує максимальну точність, але потребує значних обчислювальних і фінансових ресурсів, а відкриті моделі LLaMA є економною альтернативою для організацій із жорсткими вимогами до конфіденційності даних [14; 16].

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У роботі досліджено потенціал великих мовних моделей у задачах інтелектуального аналізу системних логів, що дало змогу дійти таких висновків.

Комбіновані LLM-рішення, де encoder-модель (BERT) вилучає ознаки, а decoder-модель (GPT-4) виконує класифікацію, досягли найвищої точності детектування аномалій ($F1 = 0,972$), перевершивши окремі моделі та класичні методи. Водночас encoder-моделі (LAnoBERT, FastLogAD) мають найкращий баланс якості й швидкодії для промислового моніторингу в реальному часі.

LLM-моделі виявляють витoki конфіденційної інформації в логах із точністю від 67,8% (SSH-ключі) до 96,8% (API-ключі), значно перевершуючи regex-підходи. Запропонована ентропійна метрика з порогом $H > 4,2$ біт/токен забезпечує додаткову фільтрацію підозрілих записів з імовірністю виявлення 0,87.

Розроблений метод поєднання LLM-класифікації зі статистичним Z-критерієм виявляє регресійні дефекти після релізів у середньому за 5,8 хв. (проти 23,8 хв. для Prometheus + Alertmanager) із часткою виявлення 93,7% і рівнем хибних спрацювань 3,1%.

У виявленні зловмисної активності LLM-моделі суттєво переважають класичні SIEM-платформи, особливо щодо багатоетапних атак: lateral movement (86,3% проти 42,1%) та C2-комунікація (83,7% проти 38,6%).

Перспективи подальших досліджень пов'язані зі створенням спеціалізованих мовних моделей, натренованих на системних журналах; федеративним навчанням LLM без передачі чутливих даних за межі периметру; адаптивним fine-tuning під специфіку конкретної інфраструктури; поєднанням LLM-аналітики з графами знань для побудови каузальних моделей аномалій, а також мультимодальними підходами, що доповнюють текстові логи числовими метриками й мережевими трейсами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- Кліщ М., Липак Г., Кунанець Н., Пасічник С., Липак Т. Структура інформаційної системи передбачення та інтерпретації зміни стану користувача сервісу. Вісник Національного університету «Львівська політехніка». Інформаційні системи та мережі. 2025. № 17. С. 226–238. URL: <https://doi.org/10.23939/sisn2025.17.226>
- Липак Г., Липак Т., Кунанець Н. Проектування інформаційної системи на основі машинного навчання для збереження та класифікації артефактів документальної спадщини. Вісник Хмельницького національного університету. Технічні науки. 2024. Т. 339, № 4. С. 176–182. URL: <https://doi.org/10.31891/2307-5732-2024-339-4-29>
- Шовак М. І. Таксономія аномалій в контексті кіберзахисту та захисту інформацій. XXI Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених



- «Теоретичні і прикладні проблеми фізики, математики та інформатики». Київ : КПІ ім. Ігоря Сікорського, 2023. С. 338–342. URL: <https://ela.kpi.ua/handle/123456789/62588>
4. Chen S., Liao H. BERT-Log: Anomaly detection for system logs based on pre-trained language model. *Applied Artificial Intelligence*. 2022. Vol. 36, No. 1. Article 2145642. URL: <https://doi.org/10.1080/08839514.2022.2145642>
 5. Guan W., Cao J., Qian S., Gao J., Ouyang C. LogLLM: Log-based anomaly detection using large language models. *arXiv preprint arXiv:2411.08561*. 2024. URL: <https://arxiv.org/abs/2411.08561>
 6. Guo H., Yuan S., Wu X. LogBERT: Log anomaly detection via BERT. 2021 International Joint Conference on Neural Networks (IJCNN). IEEE, 2021. P. 1–8. URL: <https://arxiv.org/abs/2103.04475>
 7. Hadadi F., Xu Q., Bianculli D., Briand L. Anomaly detection on unstable logs with GPT models. *arXiv preprint arXiv:2406.07467*. 2024. URL: <https://arxiv.org/abs/2406.07467>
 8. Han X., Yuan S., Trabelsi M. LogGPT: Log anomaly detection via GPT. 2023 IEEE International Conference on Big Data (BigData). IEEE, 2023. P. 1117–1122. URL: <https://arxiv.org/abs/2309.14482>
 9. Lee Y., Kim J., Kang P. LAnoBERT: System log anomaly detection based on BERT masked language model. *Applied Soft Computing*. 2023. Vol. 146. Article 110689. URL: <https://doi.org/10.1016/j.asoc.2023.110689>
 10. Li A., Zhao Y., Qiu C., Kloft M., Smyth P., Rudolph M., Mandt S. Anomaly detection of tabular data using large language models. *arXiv preprint arXiv:2406.16308*. 2024. URL: <https://arxiv.org/abs/2406.16308>
 11. Lin Y., Deng H., Li X. FastLogAD: Log anomaly detection with mask-guided pseudo anomaly generation and discrimination. *arXiv preprint arXiv:2404.08750*. 2024. URL: <https://arxiv.org/abs/2404.08750>
 12. Liu C., He S., Zhou Q., Li S., Meng W. Large language model guided knowledge distillation for time series anomaly detection. *arXiv preprint arXiv:2401.15123*. 2024. URL: <https://arxiv.org/abs/2401.15123>
 13. Liu Y., Tao S., Meng W., Wang J., Ma W., Zhao Y., Chen Y., Yang H., Jiang Y., Chen X. Interpretable online log analysis using large language models with prompt strategies (LogPrompt). *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension (ICPC)*. 2024. P. 35–46. URL: <https://arxiv.org/abs/2308.07610>
 14. Liu Y., Ji Y., Tao S., He M., Meng W., Zhang S., Sun Y., Xie Y., Chen B., Yang H. LogLM: From task-based to instruction-based automated log analysis. *arXiv preprint arXiv:2410.09352*. 2024. URL: <https://arxiv.org/abs/2410.09352>
 15. Qi J., Huang S., Luan Z., Yang S., Fung C., Yang H., Qian D., Shang J., Xiao Z., Wu Z. LogGPT: Exploring ChatGPT for log-based anomaly detection. 2023 IEEE Int. Conf. on HPCC/DSS/SmartCity/DependSys. IEEE, 2023. P. 273–280. URL: <https://arxiv.org/abs/2309.01189>
 16. Xu R., Ding K. Large language models for anomaly and out-of-distribution detection: A survey. *arXiv preprint arXiv:2409.01980*. 2024. URL: <https://arxiv.org/abs/2409.01980>

**Oleksii Melnyk**

Institute of Mathematical Machines and Systems Problems of the

National Academy of Sciences of Ukraine, Kyiv, Ukraine

ORCID:0009-0003-4800-6014

oleksii.mlk@gmail.com

INTELLIGENT ANALYSIS OF SYSTEM LOGS USING LARGE LANGUAGE MODELS FOR ANOMALY DETECTION AND NEW PATTERN DISCOVERY

Abstract. The present work examines the practical deployment of Large Language Models (LLMs) as a tool for in-depth processing of system logs from the standpoint of cyber-resilience and software dependability. It tackles the pressing challenge posed by the avalanche-like expansion of log data produced by contemporary microservice-based infrastructures, where manual oversight becomes unrealistic and machine-driven methods are indispensable. A multi-criteria comparison of the BERT, GPT, and LLaMA model families is carried out encompassing anomaly detection, identification of sensitive data leaks (PII, API keys, authentication tokens), recognition of adversarial activity, and regression fault detection following microservice releases. A formal mathematical toolkit for quantifying anomaly detection quality is put forward, comprising an author-proposed entropy measure and a Z-test criterion for identifying statistically significant deviations from baseline log-event distributions. Empirical evidence from benchmark datasets (HDFS, BGL, Thunderbird) and industrial logs substantiates the advantage of hybrid LLM strategies over conventional techniques. The research delineates the strengths and weaknesses of each language-model class relative to specific analytical tasks and provides guidelines for engineering productive LLM-powered log-analysis pipelines able to capture both established and previously unseen patterns and threats. The findings carry direct practical relevance for cybersecurity practitioners, DevOps teams, and engineers designing or maintaining monitoring platforms. Particular emphasis is placed on the risk of inadvertent leakage of secrets into log files, encompassing passwords, API keys, authentication tokens, and personally identifiable information (PII). The capacity of various LLM architectures to identify multi-stage, temporally dispersed cyber-incidents – lateral movement and command-and-control (C2) channels in particular – is assessed, revealing that such detection lies beyond the reach of conventional rule-driven SIEM platforms. Actionable guidance is offered for choosing the most suitable LLM architecture given the nature of the analytical task and the computational budget available to a given organization.

Keywords: large language models; system logs; anomalies; cybersecurity; data leakage; microservices; BERT; GPT; machine learning; intelligent analysis.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Klishch, M., Lypak, H., Kunanets, N., Pasichnyk, S., & Lypak, T. (2025). Structure of information system for predicting and interpreting user service state changes. *Visnyk of Lviv Polytechnic National University. Information Systems and Networks*, 17, 226–238. <https://doi.org/10.23939/sisn2025.17.226>
2. Lypak, H., Lypak, T., & Kunanets, N. (2024). Designing an information system based on machine learning for preservation and classification of documentary heritage artifacts. *Visnyk of Khmelnytskyi National University. Technical Sciences*, 339(4), 176–182. <https://doi.org/10.31891/2307-5732-2024-339-4-29>
3. Shovak, M. I. (2023). Taxonomy of anomalies in the context of cyber defense and information protection. In *XXI All-Ukrainian Scientific and Practical Conference of Students, Postgraduates and Young Scientists "Theoretical and Applied Problems of Physics, Mathematics and Informatics"* (pp. 338–342). Igor Sikorsky KPI. <https://ela.kpi.ua/handle/123456789/62588>
4. Chen, S., & Liao, H. (2022). BERT-Log: Anomaly detection for system logs based on pre-trained language model. *Applied Artificial Intelligence*, 36(1), Article 2145642. <https://doi.org/10.1080/08839514.2022.2145642>
5. Guan, W., Cao, J., Qian, S., Gao, J., & Ouyang, C. (2024). LogLLM: Log-based anomaly detection using large language models. *arXiv preprint arXiv:2411.08561*. <https://arxiv.org/abs/2411.08561>
6. Guo, H., Yuan, S., & Wu, X. (2021). LogBERT: Log anomaly detection via BERT. In *2021 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). IEEE. <https://arxiv.org/abs/2103.04475>
7. Hadadi, F., Xu, Q., Bianculli, D., & Briand, L. (2024). Anomaly detection on unstable logs with GPT models. *arXiv preprint arXiv:2406.07467*. <https://arxiv.org/abs/2406.07467>



8. Han, X., Yuan, S., & Trabelsi, M. (2023). LogGPT: Log anomaly detection via GPT. In 2023 IEEE International Conference on Big Data (BigData) (pp. 1117–1122). IEEE. <https://arxiv.org/abs/2309.14482>
9. Lee, Y., Kim, J., & Kang, P. (2023). LAnoBERT: System log anomaly detection based on BERT masked language model. *Applied Soft Computing*, 146, Article 110689. <https://doi.org/10.1016/j.asoc.2023.110689>
10. Li, A., Zhao, Y., Qiu, C., Kloft, M., Smyth, P., Rudolph, M., & Mandt, S. (2024). Anomaly detection of tabular data using large language models. *arXiv preprint arXiv:2406.16308*. <https://arxiv.org/abs/2406.16308>
11. Lin, Y., Deng, H., & Li, X. (2024). FastLogAD: Log anomaly detection with mask-guided pseudo anomaly generation and discrimination. *arXiv preprint arXiv:2404.08750*. <https://arxiv.org/abs/2404.08750>
12. Liu, C., He, S., Zhou, Q., Li, S., & Meng, W. (2024). Large language model guided knowledge distillation for time series anomaly detection. *arXiv preprint arXiv:2401.15123*. <https://arxiv.org/abs/2401.15123>
13. Liu, Y., Tao, S., Meng, W., Wang, J., Ma, W., Zhao, Y., Chen, Y., Yang, H., Jiang, Y., & Chen, X. (2024). Interpretable online log analysis using large language models with prompt strategies (LogPrompt). In *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension (ICPC)* (pp. 35–46). <https://arxiv.org/abs/2308.07610>
14. Liu, Y., Ji, Y., Tao, S., He, M., Meng, W., Zhang, S., Sun, Y., Xie, Y., Chen, B., & Yang, H. (2024). LogLM: From task-based to instruction-based automated log analysis. *arXiv preprint arXiv:2410.09352*. <https://arxiv.org/abs/2410.09352>
15. Qi, J., Huang, S., Luan, Z., Yang, S., Fung, C., Yang, H., Qian, D., Shang, J., Xiao, Z., & Wu, Z. (2023). LogGPT: Exploring ChatGPT for log-based anomaly detection. In 2023 IEEE Int. Conf. on HPCC/DSS/SmartCity/DependSys (pp. 273–280). IEEE. <https://arxiv.org/abs/2309.01189>
16. Xu, R., & Ding, K. (2024). Large language models for anomaly and out-of-distribution detection: A survey. *arXiv preprint arXiv:2409.01980*. <https://arxiv.org/abs/2409.01980>

Отримано редакцією журналу / Received: 02.03.26

Прорецензовано / Revised: 25.03.26

Схвалено до друку / Accepted: 24.09.26



This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.