**УДК 004.056:004.8**

***Чуєва С.О.,***

*студентка ІІ курсу магістратури*
*Харківського національного університету радіоелектроніки*

***Халімов Г.З.,***

*д.т.н., професор,*
*завідувач кафедри безпеки інформаційних технологій*
*Харківського національного університету радіоелектроніки*

# ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ШВИДКОДІЇ ТА ТОЧНОСТІ ЛОКАЛЬНИХ МАЛИХ МОВНИХ МОДЕЛЕЙ У ЗАДАЧАХ ГЕНЕРАЦІЇ XAI-ЗВІТІВ

**Анотація.** Досліджено часові та ресурсні параметри функціонування локальних малих мовних моделей під час формування пояснювальних звітів про кіберзагрози. Проаналізовано, як саме затримки генерації залежать від рівня небезпеки досліджуваного об’єкта та встановлених обмежень контекстного вікна. Доведено, що поєднання семантичної підготовки даних із механізмом локального кешування на основі криптографічних хеш-сум SHA-256 дозволяє досягти високої точності наведених фактів у висновках. Такий підхід забезпечує стабільну роботу аналітичного комплексу на робочих станціях із базовим обсягом оперативної пам’яті від 8 ГБ у локальному режимі без доступу до мережі.

**Ключові слова:** автономні системи захисту, затримка генерації, малі мовні моделі (SLM), обмеження токенів, пояснювальний штучний інтелект (XAI), семантичне структурування даних, кешування SHA-256.

Практична інтеграція технологій пояснювального штучного інтелекту в процеси розслідування кіберінцидентів вимагає оптимального

співвідношення між різними чинниками. Необхідно враховувати як часову ефективність аналізу, так і вичерпність підготовлених висновків. Використання хмарних аналітичних сервісів часто створює загрозу витоку внутрішніх технічних даних організації. Водночас розгортання локальних малих мовних моделей на пристроях користувачів суворо обмежується наявними апаратними потужностями обчислювальної системи [1].

Для оцінювання обчислювального навантаження та правдивості висновків пояснювального модуля у складі розробленого програмного комплексу було проведено експеримент. Дослідження базувалося на локальній малій мовній моделі Gemma 3 з обсягом 4 мільярди параметрів. Тестування здійснювали повністю локально на базі стандартної робочої станції без залучення зовнішніх графічних прискорювачів. Щоб запобігти переповненню контекстного вікна та усунути вигадані факти, первинні статичні індикатори компрометації потребували попередньої нормалізації. Заголовки PE-файлів, імпорти IAT, сигнатури YARA, ентропія Шеннона та нечіткі хеші TLSH були структуровані у форматі JSON [2].

Для оптимізації витрат оперативної пам’яті в коді аналітичного модуля запрограмовано динамічний розподіл ліміту токенів num_predict. Цей розподіл узгоджується з розрахованим рівнем ризику за математичною моделлю Сааті. Для стабілізації результатів коефіцієнт креативності моделі (temperature) було встановлено на рівні 0,1. Практичні результати проведених вимірювань узагальнено в табл. 1.

Аналіз одержаних експериментальних даних показує, що первинне зчитування та детермінований аналіз статичних атрибутів у коді комплексу вимагають мінімального часу – від 0,9 до 2,5 секунди. Основна частина часових витрат йде на етап послідовної генерації тексту мовною моделлю. Для безпечних об’єктів із лімітом до 250 токенів час формування пояснення становить від 54 до 106 секунд.

Таблиця 1 – Експериментальні показники функціонування модуля XAI на базі Gemma 3 (4B)

| Категорія вердикту об'єкта | Обмеження токенів (num_predict) | Коефіцієнт креативності (temperature) | Середній час генерації (с) | Час завантаження з кешу SHA-256 (с) | Використання ОЗП (ГБ) |
|---|---|---|---|---|---|
| БЕЗПЕЧНИЙ | 250 | 0,1 | 54–106 с | менше ніж 0,1 | приблизно 2,9 |
| ПІДОЗРІЛИЙ | 350 | 0,1 | 84–113 с | менше ніж 0,1 | приблизно 2,9 |
| ШКІДЛИВИЙ | 500 | 0,1 | 74–112 с | менше ніж 0,1 | приблизно 2,9 |

Варто окремо відзначити, що підозрілі об’єкти вимагають трохи більше часу для генерації від 84 до 113 секунд ніж під час аналізу шкідливих об’єктів, де показники становлять від 74 до 112 секунд. Така невелика різниця пояснюється тим, що підозрілі файли зазвичай містять суперечливі індикатори. Наприклад, наявність дійсного цифрового підпису може поєднуватися з потенційно небезпечними викликами системних функцій чи нетиповою структурою окремих секцій. Подібні суперечності змушують мовну модель формувати розширені логічні ланцюжки міркувань для обґрунтування функцій подвійного призначення. Натомість для виражено шкідливих об’єктів технічні докази небезпеки через сигнатури YARA або деструктивні правила CAPA є очевидними та однозначними. Це дозволяє моделі швидко формувати короткий та остаточний висновок. У таких випадках генерація припиняється автоматично без обов’язкового вичерпання повного ліміту у 500 токенів.

Важливою складовою забезпечення швидкодії розробленої системи є підсистема оперативного кешування на основі хеш-сум SHA-256. У разі повторного виявлення раніше проаналізованого об’єкта комплекс ідентифікує збіг і завантажує готовий пояснювальний звіт менш ніж за 0,1 секунди. Такий підхід повністю знімає повторне обчислювальне навантаження з центрального процесора [3].

Споживання оперативної пам’яті локальним сервером мовної моделі Gemma 3 (4B) у середовищі Ollama залишається стабільним. Воно становить приблизно 2,9 ГБ при загальному системному навантаженні в межах від 4,5 до 5,5 ГБ з урахуванням графічного інтерфейсу та процесів операційної системи. Це підтверджує можливість повноцінного функціонування розробленого комплексу на стандартних робочих станціях із базовим обсягом оперативної пам’яті від 8 ГБ без залучення додаткового серверного обладнання. Завдяки фіксації низького коефіцієнта креативності досягається чітка передбачуваність відповідей. Це унеможливлює появу вигаданих фактів і забезпечує відповідність сформованого звіту наданим вихідним даним.

Запропонований підхід довів свою практичну ефективність. Інтеграція семантичної підготовки даних, локальної мовної моделі та механізму кешування забезпечує високу точність аналізу загроз. Системні ресурси в автономному режимі при цьому використовуються раціонально. Розроблене архітектурне рішення суттєво знижує навантаження на фахівців із кібербезпеки. Це досягається завдяки автоматичному перетворенню складних технічних параметрів у зрозумілі пояснення та покрокові інструкції. Чітке структурування вхідних даних у структурі JSON та оптимізація параметрів генерації мінімізують ризик виникнення вигаданих фактів мовної моделі. Як результат, замість некоректної інформації система формує обґрунтовані аналітичні звіти та сценарії реагування. Отримані дані свідчать про доцільність розгортання таких автономних систем у закритих мережах державних установ та на об’єктах критичної інфраструктури.

## Список використаних джерел

1. Yang A., Kang F., Bu W. TinySecGPT: Small-Parameter LLMS Can Outperform Large-Parameter LLMS in Cybersecurity. *Computers, Materials & Continua*. 2026. Vol. 87, No. 2. P. 39. DOI: https://doi.org/10.32604/cmc.2025.073979.
2. Hiriyanna S., Zhao W. Multi-Layered Framework for LLM Hallucination Mitigation in High-Stakes Applications: A Tutorial. *Computers*. 2025. Vol. 14. P. 332. DOI: https://doi.org/10.3390/computers14080332.
3. Malware Scanner PRO: A Cloud-Assisted Desktop Application for Real-Time File Threat Detection Using VirusTotal API / V. Shyamkunwar, D. Kumar, S. Tiwari, J. Chopra. *International Journal of Research Publication and Reviews*. 2026. Vol. 7, No. 4. P. 8555–8557. URL: https://ijrpr.com/uploads/V7ISSUE4/IJRPR62789.pdf (дата звернення: 26.10.2026).