КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.34.1308
УДК 004.056.5

Пучков Олександр Олександрович
кандидат філософських наук, професор,
начальник Інституту спеціального зв'язку та захисту інформації
Національний технічний університетУкраїни
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: 0000-0002-8585-1044
iszzi@iszzi.kpi.ua

Ланде Дмитро Володимирович
доктор технічних наук, професор, завідувач кафедри
Навчальний фізико-технічний інститут
Національнийтехнічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: 0000-0003-3945-1178
dwlande@gmail.com

Субач Ігор Юрійович
доктор технічних наук, професор, завідувач кафедри
Інститут спеціального зв'язку та захисту інформації
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна
ORCID: 0000-0002-9344-713X
igor_subach@ukr.net

МЕТОД КЕРОВАНОГО ІТЕРАТИВНОГО ВИКОНАННЯ ПРОЦЕСІВ У СИСТЕМАХ НА ОСНОВІ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

Анотація. У статті запропоновано метод керованого ітеративного виконання процесів у системах на основі великих мовних моделей (*LLM*), який базується на композиції примітивів «Функція», «Суперцикл» та «Умова». Показано, що одноразове виконання запитів до *LLM* у задачах видобування знань, аналізу текстів та побудови структурованих представлень даних є принципово неповним через стохастичну природу генерації результатів, обмеження контекстного вікна та залежність результату від попереднього контексту. Запропонований підхід реалізує керований ітеративний механізм накопичення результатів із контролем новизни та адаптивною умовою завершення процесу. Розроблено математичну модель динаміки інформаційного приросту, відповідно до якої кількість нових результатів на кожній ітерації демонструє стійку тенденцію до спадання та може бути апроксимована гаусівською функцією. На цій основі обґрунтовано збіжний характер процесу та введено поняття ε-повноти результату, за якого подальші ітерації не забезпечують суттєвого інформаційного приросту. Проведена експериментальна верифікація підтвердила стабільність запропонованого механізму та ефективність адаптивної умови зупинки процесу. Отримані результати створюють методологічну основу для побудови керованих *LLM*-орієнтованих систем, здатних реалізовувати багатоетапні процеси аналізу, видобування знань, формування семантичних структур та підтримки агентних сценаріїв із контрольованою повнотою та передбачуваним завершенням виконання процесу вирішення задачі.

Ключові слова: великі мовні моделі, інтерактивна модель , агентні системи, промпт-інженерія, керування процесами, семантичні мережі, збіжність процесів, адаптивна умова зупинки.

ВСТУП

Стрімкий розвиток великих мовних моделей (Large Language Models, LLM) зумовив суттєву трансформацію підходів до автоматизації інтелектуальної діяльності у сферах аналізу текстів, видобування знань, підтримки прийняття рішень, побудови агентних систем та обробки

неструктурованої інформації. Сучасні LLM демонструють високу ефективність у задачах генерації тексту, семантичного аналізу, класифікації, узагальнення та логічного виведення, що створює передумови для їх інтеграції у складні багатокрокові інформаційно-аналітичні процеси. На практиці LLM дедалі частіше використовуються не як окремі генеративні модулі, а як елементи багатокомпонентних систем, здатних реалізовувати циклічне виконання задач, накопичення проміжних результатів, самокорекцію та адаптивне керування ходом обчислень.

Разом із тим, одноразове виконання запиту до LLM у більшості практичних сценаріїв не забезпечує повноти та стабільності результату. Це обумовлено стохастичною природою генерації результату, обмеженням контекстного вікна, залежністю відповіді від структури промпту, а також здатністю моделей акцентувати увагу лише на частині потенційно значущих залежностей. У результаті складні процеси, пов'язані з видобуванням знань, побудовою семантичних структур, аналізом взаємозв'язків або виконанням багатоетапних агентних сценаріїв, потребують багаторазового звернення до моделі з поступовим уточненням контексту та накопиченням отриманих результатів.

Постановка проблеми. У сучасних дослідженнях активно розвиваються такі підходи, як: iterative prompting, self-refinement, chain-of-thought reasoning, multi-agent orchestration та retrieval-augmented generation, спрямовані на підвищення якості роботи LLM у складних задачах. Проте більшість існуючих рішень орієнтовані переважно на емпіричне покращення результатів і не забезпечують формального опису процесу виконання, критеріїв його завершення та оцінки повноти отриманого результату. У багатьох випадках кількість ітерацій визначається евристично або задається оператором вручну, що ускладнює відтворюваність результатів, контроль обчислювальних витрат та забезпечення стабільності роботи системи.

Особливої актуальності ця проблема набуває в контексті побудови автономних агентних систем на основі LLM, де великі мовні моделі виконують функції планування, аналізу, генерації проміжних рішень та керування послідовністю дій. Відсутність формалізованих механізмів організації циклічного виконання та адаптивних умов завершення обмежує можливість створення надійних LLM-орієнтованих систем із прогнозованою поведінкою та контрольованим рівнем повноти результату.

У зв'язку з цим виникає потреба у розробленні методів керованого ітеративного виконання процесів у системах на основі великих мовних моделей, які б забезпечували накопичення проміжних результатів, контроль новизни отриманої інформації, адаптивне завершення обчислень та формалізоване обґрунтування збіжного характеру процесу. Розв'язання цієї задачі дозволяє перейти від одноразової взаємодії з LLM до побудови керованих багатокрокових процесів із передбачуваною динамікою виконання та можливістю оцінки інформаційного насичення результату.

Аналіз останніх досліджень і публікацій. Дослідження у сфері LLM, промпт-інженерії та побудови керованих агентних систем протягом останніх років сформували новий напрям розвитку інформаційних технологій (IT), орієнтований на інтеграцію генеративних моделей у складні багатоетапні процеси аналізу та прийняття рішень. Одним із фундаментальних напрямів таких досліджень є розроблення методів організації взаємодії з LLM, які забезпечують підвищення якості, повноти та стабільності результатів генерації.

У систематичному огляді методів промпт-інжинірингу [1-3] концепція «Pre-train, Prompt and Predict» визначена як базова парадигма сучасної взаємодії з LLM. У роботі детально проаналізовано підходи до формування промптів, проте питання організації багатоетапного керованого виконання процесів, адаптивного завершення ітерацій та формального опису динаміки накопичення результатів залишаються поза межами дослідження.

Подальший розвиток напрямів технологій візуальної розробки (no-code) та програмування шляхом обробки природної мови (natural-language programming) представлено у роботах [4-9], де запропоновано підходи до організації виконуваних процесів у середовищі LLM на основі композиції примітивів природномовного програмування. Зокрема, у роботі [4] вперше запропоновано безкодовий метод виявлення інформаційно-психологічних впливів на основі LLM, що демонструє можливість використання LLM як елементів керованих аналітичних процесів. У дослідженні [7] розроблено концепцію no-code programming framework із механізмами самокорекції, а роботи [8, 9] розширюють цей підхід на задачі оркестрації агентів та організації складних нелінійних потоків виконання у LLM-середовищах. Водночас зазначені роботи зосереджуються переважно на архітектурних та прикладних аспектах побудови агентних систем і не містять формалізованого математичного обґрунтування збіжності ітеративних процесів та критеріїв їх завершення.

Проблема побудови семантичних мереж та структурованого видобування знань засобами промпт-інженерії розглянута у роботі [10], де запропоновано безкодовий підхід до формування семантичних структур на основі багаторазового аналізу тексту. Дослідження демонструє перспективність

інтерпактивного промптингу (iterative prompting) для побудови графів знань, однак не визначає формальних умов завершення процесу та не розглядає питання оцінки повноти отриманого результату.

Важливим напрямом розвитку LLM-систем є процеси самовдосконалення й самокорекції (self-refinement) та інтерактивного мислення (iterative reasoning). Так, у роботі [11] показано, що LLM здатні виконувати самокорекцію та повторне уточнення результатів, проте одночасно підкреслюється ризик накопичення внутрішніх упереджень та стохастичних похибок. Це підтверджує необхідність розроблення керованих механізмів ітеративного виконання, які б забезпечували контроль якості та стабільності процесу.

У роботі [16] досліджуються автономні LLM-агенти як новий клас інтелектуальних систем, здатних реалізовувати планування, прийняття рішень та виконання складних багатоетапних сценаріїв. Автори підкреслюють перспективність агентних архітектур, однак також зазначають відсутність універсальних механізмів керування ітераційними процесами, адаптивних умов завершення та формалізованих критеріїв інформаційного насичення результату.

Окремий клас досліджень пов’язаний із технологіями генерування, що доповнені пошуком у зовнішніх базах даних (Retrieval-Augmented Generation, RAG), що поєднує LLM із зовнішніми джерелами знань. У роботі [17] наведено систематичний аналіз сучасних RAG-підходів та показано, що інтеграція зовнішнього контексту суттєво підвищує якість генерації. Разом із тим, навіть при використанні RAG-підходів проблема неповноти одноразового видобування знань та необхідність ітеративного накопичення результатів залишається актуальною.

Практичні аспекти інтеграції різнорідних джерел інформації для задач кібербезпеки розглянуто у роботі [18], де запропоновано систему CyberAggregator для агрегації даних із різних інформаційних мереж. Ця система створює основу для побудови LLM-орієнтованих процесів видобування знань із великих масивів неструктурованої інформації.

Таким чином, аналіз сучасних досліджень свідчить про активний розвиток таких підходів, як: iterative prompting, no-code orchestration, self-refinement та agent-based LLM systems. Водночас існуючі роботи переважно орієнтовані на архітектурні або прикладні аспекти використання великих мовних моделей і не забезпечують формалізованого опису керованого ітеративного виконання процесів, математичного обґрунтування їх збіжного характеру та адаптивних критеріїв завершення. Це зумовлює необхідність розроблення методів, здатних забезпечити контрольоване накопичення результатів, оцінку інформаційного приросту та передбачуване завершення процесів у системах на основі LLM.

Метою статті є розроблення методу керованого ітеративного виконання процесів у системах на основі великих мовних моделей та його застосуванні для задачі формування семантичних мереж концептів із неструктурованих текстів. Запропонований підхід спрямований на організацію багатоетапної взаємодії з LLM із накопиченням проміжних результатів, контролем інформаційного приросту та адаптивним завершенням процесу в момент досягнення стану семантичного насичення.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для досягнення поставленої мети у роботі формалізується композиція примітивів «Функція», «Суперцикл» та «Умова», де «Функція» реалізує окремий етап виконання задачі, «Суперцикл» забезпечує ітеративне накопичення результатів із передачею оновленого контексту між кроками, а «Умова» визначає адаптивний критерій завершення процесу на основі оцінки інформаційного приросту. На прикладі задачі формування семантичних мереж досліджується динаміка накопичення нових зв’язків між концептами та розробляється математична модель процесу, що описує зміну кількості нових результатів залежно від номера ітерації.

У межах дослідження виконується емпіричне та аналітичне обґрунтування збіжного характеру ітеративного процесу, вводиться поняття ε-повноти результату та досліджується роль адаптивної умови завершення у забезпеченні структурної стабільності та контрольованої повноти сформованої семантичної мережі. Отримані результати розглядаються як узагальнена методологічна основа для побудови керованих багатоетапних процесів у LLM-орієнтованих системах, що можуть застосовуватися не лише для формування семантичних мереж, а й для задач видобування знань, багатоетапного аналізу текстів, агентних сценаріїв та інших процесів, які потребують ітеративного накопичення та уточнення результатів.

Механізм формування мережі та базові промпти-функції. Суть запропонованого підходу полягає в організації керованого ітеративного процесу взаємодії з LLM, у межах якого результати кожного кроку накопичуються, аналізуються та використовуються як контекст для подальших ітерацій. На відміну від одноразового виконання промпту, запропонований механізм забезпечує поступове уточнення результату,

контроль інформаційного приросту та адаптивне завершення процесу після досягнення стану інформаційного насичення.

У загальному випадку процес складається з послідовної взаємодії трьох базових примітивів: «Функція», «Суперцикл» та «Умова». Така композиція дозволяє реалізувати універсальний механізм керованого ітеративного виконання задач у середовищі LLM незалежно від конкретної предметної області. У межах даного дослідження робота механізму розглядається на прикладі задачі формування семантичних мереж шляхом ітеративного видобування зв'язків між концептами з неструктурованих текстів.

Формалізація стану процесу та базових примітивів. Для формального опису процесу введемо поняття стану системи на k-й ітерації:

$$S_k = (H_k, R_k, C_k),$$

де H_k – множина накопичених результатів після *k*-ї ітерації;

R_k – множина результатів, отриманих на поточній ітерації;

C_k – контекст, сформований для наступного звернення до LLM.

У задачі побудови семантичної мережі множина R_k містить нові семантичні зв'язки між концептами, а H_k – накопичену множину всіх виявлених зв'язків.

У свою чергу, формування контексту для *k*-ї ітерації задається оператором:

$$C_k = G\,(X, H_{k-1}),$$

де *X* – вихідний текстовий корпус;

G – оператор формування контексту на основі вихідних даних та історії попередніх результатів.

Базовий примітив «Функція». Основним операційним елементом системи виступає примітив «Функція», який реалізує окремий етап виконання задачі засобами *LLM*. У загальному випадку функція може відповідати за аналіз тексту, генерацію проміжного результату, перевірку умов, класифікацію, пошук залежностей або виконання інших дій у межах багатоетапного процесу.

Базовий примітив «Функція» формалізується як оператор $F : C_k \times H_{k-1} \to \Delta H_k$, який на основі поточного контексту та накопиченої історії результатів генерує множину нових елементів результату.

Оновлення історії результатів виконується за правилом: $H_k = H_{k-1} \bigcup \Delta H_k$, а кількість нових результатів на *k*-й ітерації визначається як:

$$y_k = |\Delta H\_k| \tag{1}$$

Таким чином, примітив «Функція» реалізує відображення поточного контексту та історії накопичених результатів у множину нових елементів знань, тоді як історія H_k забезпечує механізм кумулятивного накопичення інформації між ітераціями.

Для задачі формування семантичної мережі примітив «Функція» реалізує процедуру витягування нових пар концептів, зв'язаних між собою за змістом. Базова текстова реалізація такого примітиву має вигляд:

```
Функція: {
Витягни з цього тексту пари понять, зв'язаних за змістом між собою. Порівняй кожну знайдену пару з історією вже виведених пар понять. Наведи тільки ті пари, яких ще не було в історії, у вигляді списку пар, що поділяються точкою з комою «;», наприклад, «вразливість;кібератака». Додай нові пари до історії виведених. Виведення: номер циклу, кількість нових пар слів.
}
```

У результаті виконання функції формується множина нових результатів поточної ітерації, оновлена історія накопичених результатів та кількісна оцінка інформаційного приросту.

Наявність історії попередніх результатів забезпечує поступове розширення контексту та зменшує ймовірність повторного виведення вже знайдених елементів.

Текстова конструкція промпту у даному випадку виступає прикладом практичної реалізації оператора F у середовищі великої мовної моделі. При цьому конкретний вигляд промпту не є фіксованим

елементом методу та може змінюватися залежно від предметної області, типу задачі та способу організації взаємодії з LLM.

Примітив «Суперцикл». Для організації багатоетапного виконання задачі базова функція інтегрується в ітеративний механізм – «Суперцикл». Його призначення полягає у повторному виконанні функції з урахуванням накопиченого контексту та результатів попередніх кроків.

На кожній ітерації виконується виклик функції, результати інтегруються до накопиченої історії, оцінюється інформаційний приріст та приймається рішення про продовження або завершення процесу.

У контексті задачі побудови семантичної мережі це забезпечує поступове виявлення як явних, так і латентних зв'язків між концептами. Після накопичення первинних зв'язків модель отримує можливість використовувати вже знайдені концепти як додатковий контекст для пошуку менш очевидних залежностей.

У загальному вигляді структура процесу може бути представлена так:

```
Суперцикл початок: {
Функція: {виконання задачі}
Оновлення історії результатів
Оцінка інформаційного приросту
Умова завершення
} Закінчення суперциклу
```

Вивід результату

Формально примітив «Суперцикл» може бути представлений як оператор переходу між станами системи:

$$S_k = \Phi(S_{[k-1]}),$$

де S_{k-1} – стан системи на попередній ітерації;

S_k – стан системи після виконання поточного циклу;

Φ – оператор суперциклу, який включає: формування нового контексту C_k; виконання функції *F*; оновлення історії результатів H_k; оцінювання інформаційного приросту y_k; перевірку умови завершення процесу.

У розгорнутому вигляді один цикл роботи системи може бути поданий як:

$$S_{k-1} \to C_k \to F \to R_k \to H_k \to S_k .$$

Таким чином, примітив «Суперцикл» реалізує керований механізм еволюції стану системи з поступовим накопиченням результатів та уточненням контексту виконання.

Примітив «Умова». Керування завершенням процесу реалізується примітивом «Умова», який визначає критерій припинення ітерацій. На відміну від підходів із фіксованою кількістю повторень, запропонований механізм використовує адаптивний критерій завершення, що базується на оцінці інформаційного приросту.

У задачі формування семантичної мережі таким критерієм виступає кількість нових пар понять, отриманих на поточній ітерації. Якщо кількість нових зв'язків стає меншою за заданий поріг *N*, процес завершується:

$$y_k < N,$$

де: y_k – кількість нових результатів на *k*-й ітерації;

N – порогове значення новизни.

У більш загальному випадку примітив «Умова» може бути формалізований як функціонал завершення:

$$T(S_k) = \begin{cases} 1, y_k < N, \\ 0, \textit{інакше.} \end{cases}.$$

Якщо $T(S_k)=1$, ітеративний процес завершується. В іншому випадку виконується перехід до наступної ітерації суперциклу.

У загальному випадку функціонал завершення може враховувати не лише кількість нових результатів, а й семантичну стабільність, структурну подібність результатів суміжних ітерацій, зовнішні критерії верифікації, а також ресурсні обмеження.

Такий механізм дозволяє автоматично фіксувати момент семантичного насичення, коли подальші ітерації перестають давати суттєвий інформаційний приріст.

Отже, примітив «Умова» може використовувати оцінку структурної стабільності результату, семантичну подібність між ітераціями, зміну метрик якості, зовнішні критерії верифікації, а також комбіновані логічні умови.

Це дозволяє застосовувати запропонований механізм не лише для побудови семантичних мереж, а й для організації інших багатоетапних процесів у системах на основі LLM.

Особливості ітеративного виконання в середовищі *LLM*. Необхідність застосування керованого ітеративного механізму обумовлена архітектурними особливостями сучасних великих мовних моделей. Стохастична природа генерації результату, обмеження контекстного вікна та нерівномірний розподіл уваги призводять до того, що одноразове виконання промпту часто не забезпечує повного охоплення інформаційного простору задачі.

У процесі ітеративного виконання система поступово накопичує результати та формує розширений контекст, який використовується для подальших звернень до моделі. Це дозволяє підвищити повноту результату, зменшити кількість пропущених залежностей, активувати виявлення латентних зв'язків, знизити кількість дублювань та забезпечити контрольований характер завершення процесу.

Таким чином, композиція примітивів «Функція», «Суперцикл» та «Умова» формує універсальний механізм керованого ітеративного виконання процесів у системах на основі LLM, а задача формування семантичних мереж виступає прикладом його практичного застосування.

Математична модель процесу та обґрунтування його збіжного характеру.

Нехай $k \in N$ – номер ітерації примітиву «Суперцикл», а y_k – кількість нових результатів, отриманих на *k*-й ітерації після фільтрації дублікатів відносно накопиченої історії виконання процесу.

Формально величина y_k визначається як потужність множини нових результатів, отриманих на *k*-й ітерації (1).

У задачі формування семантичної мережі під новими результатами розуміються унікальні семантичні зв'язки між концептами, які ще не були виявлені на попередніх кроках.

Тоді загальна кількість накопичених результатів після *M* ітерацій визначається як:

$$Y_M = \sum_{k=1}^{M} y_k, \tag{2}$$

де y_k – інформаційний приріст на *k*-й ітерації;

Y_M – сумарний результат процесу після *M* кроків.

У задачі побудови семантичної мережі, загальна кількість ребер у сформованому графі знань визначається як часткова сума ряду (2), де момент зупинки алгоритму задається умовою $M = \{\min k \in N \mid y_k < N\}$, а N – заданий поріг новизни, що слугує параметром примітиву «Умова».

У загальному випадку запропонована модель може застосовуватися не лише до задачі побудови семантичних мереж, а й до інших процесів у LLM-орієнтованих системах, де кожна ітерація генерує нові елементи результату: факти, концепти, правила, сценарії, висновки або структуровані представлення знань.

Динаміка інформаційного приросту. Експериментальні дослідження показали, що в процесі ітеративного виконання кількість нових результатів із кожною наступною ітерацією має тенденцію до зменшення. Це пояснюється тим, що на початкових етапах модель виявляє найбільш очевидні та семантично виражені залежності, тоді як подальші ітерації спрямовані на пошук менш явних або латентних зв'язків.

У межах проведених експериментів встановлено, що усереднена динаміка інформаційного приросту може бути апроксимована функцією вигляду:

$$y(x) = A \cdot e^{-B(x-x_0)^2}, A > 0, B > 0 \tag{3}$$

де: A – максимальна інтенсивність інформаційного приросту;

B – параметр швидкості спадання;

x_0 – положення максимуму функції;

x – номер ітерації.

Тут, зазначена функція (3) не розглядається як універсальний закон поведінки всіх *LLM*-процесів, а використовується як емпіричний апроксимант динаміки інформаційного приросту для досліджуваного класу задач.

Обґрунтування збіжного характеру процесу. Для аналізу поведінки процесу розглянемо нескінченний ряд $\sum_{k=1}^{\infty} y_k$, який описує накопичення результатів у разі необмеженого продовження ітерацій.

Оскільки емпірично встановлено, що величина y_k демонструє тенденцію до спадання та апроксимується додатною обмеженою функцією (3), для аналізу збіжності можна використати інтегральну оцінку:

$$\sum_{k=1}^{\infty} y_k \le y_1 + \int_1^{\infty} A \cdot e^{-B(x-x_0)^2} dx. \tag{4}$$

Обчислення (4) здійснюється шляхом лінійної заміни змінної та зведення до доповняльної функції помилок erfc (complementary error function) $erfc(z) = \frac{2}{\sqrt{\pi}} \int_1^{\infty} e^{-t^2} dt,$ що дає:

$$\int_1^{\infty} A \cdot e^{-B(x-x_0)^2} dx = A\sqrt{\frac{\pi}{4B}} \cdot erfc\left(\sqrt{B(1-x_0)}\right) < \infty. \tag{5}$$

Відповідно до (5) ряд накопичення результатів має обмежений характер: $Y_M \le Y_{\max}$, де $Y_{\max}$ – деяке скінченне значення.

Це свідчить про те, що процес ітеративного накопичення результатів характеризується збіжною поведінкою: із ростом кількості ітерацій інформаційний приріст зменшується, а сумарний результат поступово наближається до граничного стану насичення.

При цьому слід зазначити, що наведене обґрунтування не є строгим математичним доведенням збіжності будь-якого процесу на основі LLM, а описує збіжний характер процесу для досліджуваного класу задач на основі емпірично встановленої динаміки інформаційного приросту.

Поняття ε-повноти результату. На основі збіжного характеру процесу введемо поняття ε-повноти результату. Вважатимемо, що процес досяг стану ε-повноти, якщо внесок нових результатів на поточній ітерації стає меншим за заданий поріг, тобто виконується умова $y_k < \varepsilon$. Оскільки асимптотична поведінка y_k описується експоненційно спадною гаусівською функцією, то для будь-якого фіксованого ε існує скінченний індекс k_ε, починаючи з якого виконується нерівність $y_k < N(\varepsilon)$.

У задачі побудови семантичної мережі це означає, що більшість семантично значущих зв'язків уже інтегровані до графа, а подальші ітерації переважно генерують дублікати, тривіальні залежності, слабко значущі зв'язки та шумові генерації.

Таким чином, примітив «Умова» реалізує адаптивний механізм завершення процесу, який дозволяє автоматично визначати момент інформаційного насичення без необхідності задавати фіксовану кількість ітерацій.

Емпірична верифікація та експериментальні результати. Емпірична верифікація запропонованого методу проводилась на прикладі задачі формування семантичних мереж із неструктурованих текстів у сфері кібербезпеки. Обрана предметна область характеризується високою щільністю взаємопов'язаних концептів, наявністю прихованих семантичних залежностей та значною кількістю латентних зв'язків між сутностями, що робить її придатною для дослідження поведінки керованих ітеративних процесів у середовищі LLM.

Вхідні текстові дані були отримані через систему CyberAggregator [18] у режимі *RAG* [17]. Тексти містили інформацію про кібератаки, вразливості, загрози, хакерські угрупування, засоби захисту та пов'язані концепти кібербезпеки. Для забезпечення відтворюваності експериментів використовувався

стандартизований промпт, побудований на композиції примітивів «Функція», «Суперцикл» та «Умова», що реалізовували кероване ітеративне накопичення результатів із контролем новизни на кожному кроці.

У межах експерименту було виконано понад 100 незалежних запусків ітеративного процесу. На кожній ітерації фіксувалась кількість нових результатів y_k, отриманих після фільтрації дублікатів відносно накопиченої історії. У контексті задачі побудови семантичної мережі під новими результатами розумілись унікальні семантичні зв'язки між концептами, які раніше не були виявлені системою.

Усереднення результатів експериментів дозволило встановити характерну динаміку інформаційного приросту. На перших ітераціях система демонструвала максимальну інтенсивність генерації нових зв'язків: середнє значення y_k на перших двох циклах становило близько 10 нових пар понять. Надалі спостерігалося поступове зниження інформаційного приросту: на третьому циклі середнє значення становило 9.3, на четвертому – 6.3, на п'ятому – 4.5, на шостому – 2.8, на сьомому – 1.3, а на восьмому циклі середній інформаційний приріст знижувався до 0.5 нових пар на ітерацію.

Отримані результати підтверджують наявність стійкої тенденції до зменшення кількості нових результатів у процесі накопичення контексту. На початкових етапах LLM ефективно виявляє найбільш очевидні та явно виражені залежності, тоді як подальші ітерації орієнтовані на пошук менш помітних або латентних зв'язків. Після досягнення певного рівня насичення інформаційний приріст стає мінімальним, а подальше виконання процесу переважно призводить до генерації дублікатів або слабко значущих результатів.

На основі отриманих експериментальних даних було виконано нелінійну апроксимацію динаміки інформаційного приросту. Найкращу відповідність експериментальним точкам продемонструвала функція вигляду: $y(x) = 10.6 \cdot e^{-0.0429(x-0.86)^2}$.

Коефіцієнт детермінації моделі перевищував значення $R^2 > 0.95$, що свідчить про високу узгодженість апроксимації з експериментальними даними.

На рис. 1 наведено графічне зіставлення усереднених значень y_k та теоретичної кривої, побудованої на основі отриманої апроксимуючої функції.

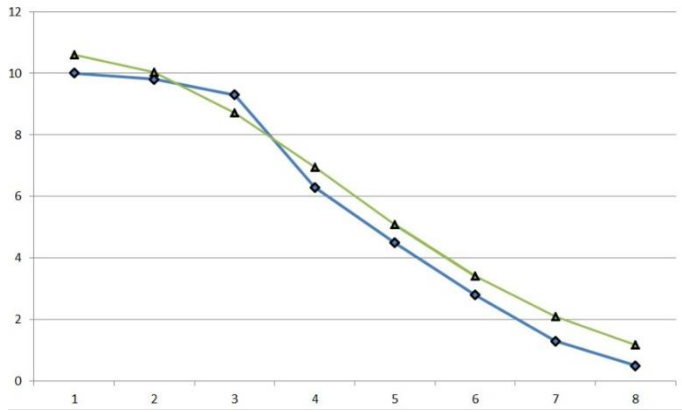

Рис. 1. Експериментально отримані усереднені значення y_k (♦) та теоретична крива (▲)

Як видно з графіка (див. рис. 1), експериментальні значення демонструють стійку відповідність теоретичній моделі на всьому інтервалі ітерацій. Особливістю процесу є наявність двофазної структури: початкової області інтенсивного накопичення нових результатів та подальшого етапу експоненційного спадання інформаційного приросту. Така поведінка узгоджується з особливостями роботи LLM, для яких характерне швидке виявлення найбільш виражених залежностей на ранніх етапах аналізу та поступовий перехід до менш очевидних семантичних зв'язків.

Починаючи з сьомої–восьмої ітерації, значення y_k стабілізувалося на рівні, меншому за порогове значення $N = 2$, що автоматично активувало примітив «Умова» та завершувало виконання «Суперциклу». Це підтверджує ефективність адаптивного механізму завершення процесу без необхідності ручного визначення кількості ітерацій або використання жорстких часових чи ресурсних обмежень.

На рис. 2 наведено приклад сформованої семантичної мережі, отриманої в результаті виконання одного з експериментів. Вузли графа відповідають концептам предметної області, а ребра – виявленим семантичним зв'язкам між ними. Структура мережі містить як локальні тематичні кластери, так і

міжкластерні зв'язки, що відображають складні взаємозалежності між об'єктами досліджуваного інформаційного простору.

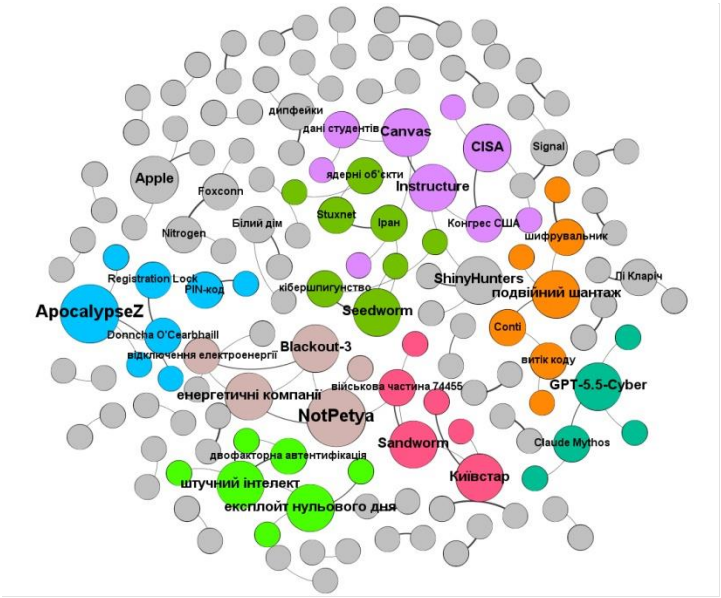

Рис. 2. Приклад сформованої семантичної мережі

Отримані результати підтверджують, що запропонований метод забезпечує керований характер ітеративного виконання процесу та дозволяє реалізувати поступове накопичення результатів із контролем інформаційного приросту. При цьому задача формування семантичних мереж розглядається як приклад практичного застосування запропонованого підходу.

У більш загальному випадку отримані закономірності можуть бути поширені на інші класи ітеративних процесів у системах на основі LLM, для яких є характерним накопичення проміжних результатів, використання історії попередніх ітерацій, зменшення інформаційного приросту та необхідність адаптивного завершення процесу.

До таких задач можуть належати iterative reasoning, self-refinement, retrieval-augmented analysis, agentic workflows, багатоетапне видобування знань та інші LLM-орієнтовані процеси, що потребують контрольованого накопичення та уточнення результатів.

Роль примітивів у забезпеченні надійності. Ключовою особливістю запропонованого підходу є використання композиції примітивів «Функція», «Суперцикл» та «Умова» для організації керованого ітеративного виконання процесів у системах на основі великих мовних моделей. На відміну від традиційних сценаріїв взаємодії з LLM, де результат формується в межах одноразового виконання промпту, запропонована архітектура реалізує багатоетапний процес із накопиченням контексту, контролем інформаційного приросту та адаптивним завершенням виконання.

У загальній структурі методу примітив «Функція» виконує роль базового операційного елемента, що реалізує окремий етап обробки інформації засобами LLM. Залежно від предметної області та типу задачі функція може відповідати за видобування знань, аналіз тексту, генерацію проміжних висновків, класифікацію, пошук залежностей або інші операції. У задачі формування семантичних мереж цей примітив забезпечує витягування нових зв'язків між концептами та фільтрацію дублікатів відносно накопиченої історії результатів.

Використання історії попередніх ітерацій дозволяє реалізувати механізм кумулятивного накопичення контексту:

$$H_k = H_{k-1} \bigcup \Delta H_k ,$$

де H_k – множина накопичених результатів після *k*-ї ітерації;

ΔH_k – множина нових результатів, отриманих на поточному кроці.

Така модель забезпечує поступове розширення інформаційного простору задачі та створює умови для виявлення латентних або непрямих залежностей, які можуть залишатися недоступними під час одноразового виконання промпту.

Примітив «Суперцикл» реалізує механізм багатоетапного виконання процесу та забезпечує повторне застосування функції з урахуванням накопиченого контексту. На кожній ітерації результати попередніх кроків інтегруються до історії виконання та використовуються як додатковий контекст для подальшого аналізу. Це дозволяє системі поступово уточнювати результат та зменшувати кількість пропущених залежностей.

У контексті задачі формування семантичних мереж така організація процесу забезпечує поступовий перехід від виявлення найбільш очевидних семантичних зв'язків до пошуку менш виражених або латентних залежностей між концептами. У більш загальному випадку аналогічний механізм може використовуватись у задачах iterative reasoning, self-refinement, retrieval-augmented analysis та agentic workflows, де результат формується через послідовне накопичення та уточнення проміжних даних.

Особливу роль у забезпеченні керованості процесу відіграє примітив «Умова», який реалізує адаптивний механізм завершення виконання. На відміну від підходів із фіксованою кількістю ітерацій, запропонований метод використовує оцінку інформаційного приросту як критерій продовження або завершення процесу. У найпростішому випадку умова завершення має вигляд: $y_k < N$, де y_k – кількість нових результатів на k-й ітераці, а N – заданий поріг.

Такий підхід дозволяє автоматично визначати момент інформаційного насичення, коли подальше виконання процесу не забезпечує суттєвого приросту результатів. У задачі побудови семантичної мережі це відповідає ситуації, коли більшість семантично значущих зв'язків уже інтегровані до графа, а нові ітерації переважно генерують дублікати або слабко значущі залежності.

У більш складних сценаріях примітив «Умова» може використовувати комбіновані критерії завершення, що враховують семантичну подібність результатів суміжних ітерацій, стабільність структури сформованих даних, зміну метрик якості, зовнішню верифікацію результатів, а також обмеження часу або обчислювальних ресурсів.

Формально така умова може бути представлена у вигляді логічної комбінації критеріїв:

$$C_k = \left(y_k < N\right) \wedge \left(sim\left(R_k, R_{k-1}\right) > T_{sim}\right),$$

де R_k – результат результат *k*-ї ітерації;

$sim(\cdot)$ – функція оцінки подібності (наприклад, косинусна подібність векторних ембедінгів графів);

$T_{sim} \in [0,1]$ – поріг семантичної стабільності.

Наявність адаптивної умови завершення дозволяє уникнути двох типових проблем LLM-орієнтованих процесів:

- передчасного завершення, коли результат ще не досяг достатнього рівня повноти;
- непродуктивного продовження виконання після досягнення стану інформаційного насичення.

Таким чином, композиція примітивів «Функція», «Суперцикл» та «Умова» забезпечує не лише організацію ітеративного виконання процесу, а й формує механізм його керованості, адаптивності та контрольованого завершення. У задачі побудови семантичних мереж це дозволяє реалізувати поступове накопичення зв'язків між концептами з контролем інформаційного приросту, тоді як у більш загальному випадку запропонований підхід може використовуватись як універсальна основа для побудови керованих багатоетапних процесів у системах на основі великих мовних моделей.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У статті запропоновано метод керованого ітеративного виконання процесів у системах на основі великих мовних моделей, який базується на композиції примітивів «Функція», «Суперцикл» та «Умова». На відміну від традиційних підходів, орієнтованих на одноразове виконання промпту, запропонований метод реалізує механізм багатоетапного накопичення результатів із використанням історії попередніх ітерацій, контролем інформаційного приросту та адаптивним завершенням процесу. Це дозволяє підвищити повноту результату та забезпечити керований характер виконання задач у середовищі LLM.

У межах дослідження задача формування семантичних мереж розглянута як приклад практичного застосування запропонованого підходу. Показано, що використання ітеративного механізму дозволяє поступово накопичувати зв'язки між концептами, зменшуючи кількість пропущених залежностей та забезпечуючи виявлення латентних семантичних зв'язків, які можуть залишатися невиявленими під час одноразового аналізу тексту.

Розроблено математичну модель процесу, у якій інформаційний приріст визначається кількістю нових результатів, отриманих на кожній ітерації. На основі експериментальних даних встановлено, що динаміка інформаційного приросту демонструє стійку тенденцію до спадання та може бути апроксимована гаусівською функцією. Проведений аналіз підтвердив збіжний характер процесу накопичення результатів та наявність стану інформаційного насичення, при якому подальші ітерації не забезпечують суттєвого приросту нових даних.

Важливим результатом роботи є введення поняття ε-повноти результату, яке характеризує стан процесу, за якого інформаційний приріст стає меншим за заданий поріг. На цій основі примітив «Умова» реалізує адаптивний механізм завершення виконання, що дозволяє автоматично визначати момент завершення ітеративного процесу без необхідності задавати фіксовану кількість повторень.

Емпірична верифікація на основі серії експериментів із текстами у сфері кібербезпеки підтвердила ефективність запропонованого підходу. Отримані результати продемонстрували стабільну динаміку зменшення інформаційного приросту та високу узгодженість експериментальних даних з апроксимуючою моделлю. Сформовані семантичні мережі відобразили як локальні тематичні зв'язки, так і складні міжкластерні залежності між концептами предметної області.

Запропонований метод розглядається як узагальнена методологічна основа для організації керованих багатоетапних процесів у системах на основі великих мовних моделей. Окрім задач формування семантичних мереж, підхід може бути застосований у задачах iterative reasoning, self-refinement, retrieval-augmented analysis, agentic workflows, видобування знань та інших процесах, що потребують накопичення проміжних результатів, контролю інформаційного приросту та адаптивного завершення виконання.

Подальші дослідження доцільно спрямувати на автоматичний вибір параметрів адаптивної умови завершення, розроблення методів оцінки семантичної стабільності результатів, інтеграцію зовнішніх механізмів верифікації знань, а також дослідження поведінки запропонованого підходу в мультиагентних та мультидоменних LLM-орієнтованих системах.

ДЕКЛАРАЦІЯ ПРО ШТУЧНИЙ ІНТЕЛЕКТ

Інструменти штучного інтелекту використовувались для мовно-стилістичного редагування тексту та не впливали на науковий зміст, результати та висновки дослідження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys, 55*(9), 1-35. https://doi.org/10.1145/3560815
2. Aghajanyan, A., Okhonko, D., Lewis, M., Joshi, M., Xu, H., Ghosh, G., & Zettlemoyer, L. (2021). *HTLM: Hyper-text pre-training and prompting of language models* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2107.06955
3. Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). *A systematic survey of prompt engineering in large language models: Techniques and applications* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2402.07927
4. Lande, D., Yefremov, K., Soboliev, A., & Pyshnograiev, I. (2025). Devising a code-free method for detecting signs of informational-psychological influences in messages. *Eastern-European Journal of Enterprise Technologies, 5*(2(137)), 55-69. https://doi.org/10.15587/1729-4061.2025.342297
5. Tang, J., Fan, T., & Huang, C. (2025). *AutoAgent: A fully automated and zero-code framework for LLM agents* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2502.05957
6. Irmak, G., & Mahmoud, Q. H. (2026). Design behaviour and interface consistency in generative no-code tools: A systematic literature review. *Computers, 15*(4), Article 238. https://doi.org/10.3390/computers15040238
7. Lande, D., Danyk, Y., & Strashnoy, L. (2026). A no-code programming framework with self-correction based on large language models. In P. Venherskyi, S. Yevseiev, O. Gutik, V. Trushevskyy, & M. Viachalo (Eds.), *Proceedings of the Workshop on Cryptology and Data Security (WCDS 2025), co-*

located with SMICS 2025 (CEUR Workshop Proceedings, Vol. 4191, pp. 22-37). CEUR-WS.org. https://ceur-ws.org/Vol-4191/paper1.pdf

8. Lande, D., & Strashnoy, L. (2025a). *An advanced no-code programming framework for complex problems in LLM environments* [Preprint]. ResearchGate. https://doi.org/10.13140/RG.2.2.25307.89129
9. Lande, D., & Strashnoy, L. (2025b). *Orchestration of no-code agents in the AgentFlow framework* [Preprint]. SSRN. https://doi.org/10.2139/ssrn.5381820
10. Ланде, Д., & Рибак, О. (2025). Безкодовий підхід до побудови семантичних мереж засобами промпт-інжинірингу. *Information Technology and Security*, *13*(2), 264-278. https://doi.org/10.20535/2411-1031.2025.13.2.344712
11. Xu, W., Zhu, G., Zhao, X., Pan, L., Li, L., & Wang, W. (2024). Pride and prejudice: LLM amplifies self-bias in self-refinement. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics: Volume 1. Long Papers* (pp. 15474-15492). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.826
12. Madaan, A., et al. (2023). Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, *36*, 46534-46594. https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf
13. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, *36*, 8634-8652. https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf
14. Kumar, A., et al. (2024). *SCoRe: Self-correcting reasoning in large language models* [Preprint]. arXiv. https://arxiv.org/pdf/2511.09381
15. Zelikman, E., Lorch, E., Mackey, L., & Kalai, A. T. (2024). Self-taught optimizer (STOP): Recursively self-improving code generation. In *First Conference on Language Modeling*. OpenReview. https://openreview.net/forum?id=46Zgqo4QIU
16. Rafique, Z., Wasim, M., Hussain, M., Hussain, M., & Memon, M. I. (2025). From prompt to action: A comprehensive review of LLM autonomous agents. In *2025 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)* (pp. 1-6). IEEE. https://doi.org/10.1109/WiSEE57913.2025.11229863
17. Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., & Li, Q. (2024). A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 6491-6501). Association for Computing Machinery. https://doi.org/10.1145/3637528.3671470
18. Lande, D., Puchkov, O., & Subach, I. (2021). Aggregation of information from diverse networks as the basis for training cyber security specialists on processing ultra large data sets. *Information Technology and Security*, *9*(1), 4-16. https://doi.org/10.20535/2411-1031.2021.9.1.247256

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023

Olexandr Puchkov
PhD in Philosophy, Professor
Head of the Institute of Special Communication and Information Protection
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: 0000-0002-8585-1044
iszzi@iszzi.kpi.ua

Dmytro Lande
Doctor of Technical Sciences, Professor,
Head of the Department Educational and Scientific Physico-Technical Institute
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: 0000-0003-3945-1178
dwlande@gmail.com

Ihor Subach
Doctor of Technical Science, Professor, Head of the Department
Institute of Special Communications and Information Protection
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: 0000-0002-9344-713X
igor_subach@ukr.net

A METHOD FOR CONTROLLED ITERATIVE PROCESS EXECUTION IN SYSTEMS BASED ON LARGE LANGUAGE MODELS

Abstract. This paper proposes a method for the controlled iterative execution of processes in systems based on large language models (LLMs), which is based on the composition of the 'Function', 'Supercycle' and 'Condition' primitives. It is shown that a single execution of queries to an LLM in tasks of knowledge extraction, text analysis and the construction of structured data representations is fundamentally incomplete due to the stochastic nature of result generation, the limitations of the context window and the dependence of the result on the preceding context. The proposed approach implements a controlled iterative mechanism for accumulating results with novelty control and an adaptive termination condition. A mathematical model of the dynamics of information gain has been developed, according to which the number of new results at each iteration exhibits a steady downward trend and can be approximated by a Gaussian function. On this basis, the convergent nature of the process is substantiated and the concept of ε-completeness of a result is introduced, under which further iterations do not provide significant information gain. Experimental verification has confirmed the stability of the proposed mechanism and the effectiveness of the adaptive termination condition. The results obtained provide a methodological basis for constructing controllable LLM-oriented systems capable of implementing multi-stage processes of analysis, knowledge extraction, semantic structure formation, and support for agent-based scenarios with controlled completeness and predictable termination.

Keywords: large language models, interactive models, agent-based systems, prompt engineering, process control, semantic networks, process convergence, adaptive stopping condition.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys, 55*(9), 1-35. https://doi.org/10.1145/3560815
2. Aghajanyan, A., Okhonko, D., Lewis, M., Joshi, M., Xu, H., Ghosh, G., & Zettlemoyer, L. (2021). *HTLM: Hyper-text pre-training and prompting of language models* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2107.06955
3. Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). *A systematic survey of prompt engineering in large language models: Techniques and applications* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2402.07927

4. Lande, D., Yefremov, K., Soboliev, A., & Pyshnograiev, I. (2025). Devising a code-free method for detecting signs of informational-psychological influences in messages. *Eastern-European Journal of Enterprise Technologies, 5*(2(137)), 55-69. https://doi.org/10.15587/1729-4061.2025.342297
5. Tang, J., Fan, T., & Huang, C. (2025). *AutoAgent: A fully automated and zero-code framework for LLM agents* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2502.05957
6. Irmak, G., & Mahmoud, Q. H. (2026). Design behaviour and interface consistency in generative no-code tools: A systematic literature review. *Computers, 15*(4), Article 238. https://doi.org/10.3390/computers15040238
7. Lande, D., Danyk, Y., & Strashnoy, L. (2026). A no-code programming framework with self-correction based on large language models. In P. Venherskyi, S. Yevseiev, O. Gutik, V. Trushevskyy, & M. Viachalo (Eds.), *Proceedings of the Workshop on Cryptology and Data Security (WCDS 2025), co-located with SMICS 2025* (CEUR Workshop Proceedings, Vol. 4191, pp. 22-37). CEUR-WS.org. https://ceur-ws.org/Vol-4191/paper1.pdf
8. Lande, D., & Strashnoy, L. (2025a). *An advanced no-code programming framework for complex problems in LLM environments* [Preprint]. ResearchGate. https://doi.org/10.13140/RG.2.2.25307.89129
9. Lande, D., & Strashnoy, L. (2025b). *Orchestration of no-code agents in the AgentFlow framework* [Preprint]. SSRN. https://doi.org/10.2139/ssrn.5381820
10. Ланде, Д., & Рибак, О. (2025). Безкодовий підхід до побудови семантичних мереж засобами промпт-інжинірингу. *Information Technology and Security, 13*(2), 264-278. https://doi.org/10.20535/2411-1031.2025.13.2.344712
11. Xu, W., Zhu, G., Zhao, X., Pan, L., Li, L., & Wang, W. (2024). Pride and prejudice: LLM amplifies self-bias in self-refinement. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics: Volume 1. Long Papers* (pp. 15474-15492). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.826
12. Madaan, A., et al. (2023). Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems, 36*, 46534-46594. https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf
13. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems, 36*, 8634-8652. https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf
14. Kumar, A., et al. (2024). *SCoRe: Self-correcting reasoning in large language models* [Preprint]. arXiv. https://arxiv.org/pdf/2511.09381
15. Zelikman, E., Lorch, E., Mackey, L., & Kalai, A. T. (2024). Self-taught optimizer (STOP): Recursively self-improving code generation. In *First Conference on Language Modeling*. OpenReview. https://openreview.net/forum?id=46Zgqo4QIU
16. Rafique, Z., Wasim, M., Hussain, M., Hussain, M., & Memon, M. I. (2025). From prompt to action: A comprehensive review of LLM autonomous agents. In *2025 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)* (pp. 1-6). IEEE. https://doi.org/10.1109/WiSEE57913.2025.11229863
17. Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., & Li, Q. (2024). A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 6491-6501). Association for Computing Machinery. https://doi.org/10.1145/3637528.3671470
18. Lande, D., Puchkov, O., & Subach, I. (2021). Aggregation of information from diverse networks as the basis for training cyber security specialists on processing ultra large data sets. *Information Technology and Security, 9*(1), 4-16. https://doi.org/10.20535/2411-1031.2021.9.1.247256

Отримано редакцією журналу / Received: 26.01.26
Прорецензовано / Revised: 06.02.26
Схвалено до друку / Accepted: 24.09.26