УДК 004.89 DOI 10.18372/2073-4751.87.21693

Спірідонов А. Ю.,
orcid.org/0009-0005-4071-1733,
e-mail: 2386932@stud.nau.edu.ua

МАТЕМАТИЧНА МОДЕЛЬ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ІНФОРМАЦІЇ НА ОСНОВІ ГІПЕРАВТОМАТИЗАЦІЇ

Національний університет «Київський авіаційний інститут»

Вступ

Попередні публікації[1] окреслили контекст і практичний ефект системи обробки інформації на основі гіперавтоматизації: у першій статті - огляд технологій та архітектурні підходи до інтелектуальної обробки даних, у другій - експериментальну перевірку продуктивності прототипу (RPA+ETL+LLM) на реальних потоках даних. Результати показали істотне скорочення часу обробки та високу релевантність відповідей.

Мета статті - розробити і формалізувати математичну модель системи обробки інформації на гіперавтоматизації, яка інтегрує джерела базових даних (MBASE), результати роботизованого збору (RPA), процедури інтеграції/ETL та донавчання LLM до єдиної інтегральної системи, а також формалізує перетворення цільового запиту користувача у відповідь системи з чітко визначеними метриками якості й часу.

Огляд аналогічних досліджень та існуючих підходів

За останнє десятиліття концепція гіперавтоматизації набула широкого розвитку як інтеграція RPA, машинного навчання, ETL-конвеєрів та великих мовних моделей. Gartner визначає гіперавтоматизацію як стратегічний підхід до автоматизації бізнес-процесів шляхом поєднання різних технологій автоматизації та аналітики даних [2].

Дослідження у сфері RPA та процесної автоматизації

У роботах van der Aalst et al. [3] показано, що традиційна роботизована автоматизація ефективна для структурованих процесів, однак має обмеження у роботі з неструктурованими даними. Автори пропонують інтегрувати RPA з process mining та ML-алгоритмами для підвищення адаптивності систем.

Дослідження Willcocks et al. [3] демонструє економічний ефект RPA у корпоративних середовищах, зокрема скорочення операційних витрат на 30–50%, однак наголошує на необхідності масштабованих архітектур для обробки великих обсягів даних.

Інтеграція ETL та інтелектуальної обробки даних

Класичні ETL-архітектури описані у роботах Kimball та Ross [4], де формалізовано принципи побудови сховищ даних. Однак традиційні підходи не враховують потокові джерела та неструктуровані дані.

У дослідженнях Inmon et al. [5] запропоновано концепцію Data Lake та hybrid ETL, що дозволяє інтегрувати напівструктуровані та неструктуровані дані, що безпосередньо корелює з підходом, використаним у даній роботі.

Використання LLM та RAG-архітектур

Після появи GPT-архітектур [6] великі мовні моделі почали застосовуватися для аналітики та прийняття рішень. Однак класичні LLM мають обмеження щодо актуальності знань.

У роботах Lewis et al. [7] запропоновано Retrieval-Augmented Generation (RAG), що поєднує пошук у зовнішніх базах знань із генеративними моделями. Подібні підходи розвинуто у дослідженнях Izacard et al. [8], де показано суттєве підвищення точності відповідей.

Патентні рішення у сфері гіперавтоматизації

Аналіз патентних баз (USPTO, WIPO) показує активний розвиток систем:

– IBM (US20210326745A1) – інтеграція RPA з ML для автоматичної обробки бізнес-подій;[9]

– UiPath (WO2022178932A1) – інтелектуальна оркестрація ботів із використанням AI;[10]

– Microsoft (US20220176534A1) – гібридні ETL-конвеєри з вбудованою аналітикою.

Всі зазначені рішення зосереджені на окремих компонентах, однак не пропонують цілісної математичної моделі процесу інтеграції даних, адаптації LLM та формалізації часових метрик.[11]

2. Архітектурна схема обробки даних:

Розглянемо архітектуру запропонованої моделі (рис. 1)

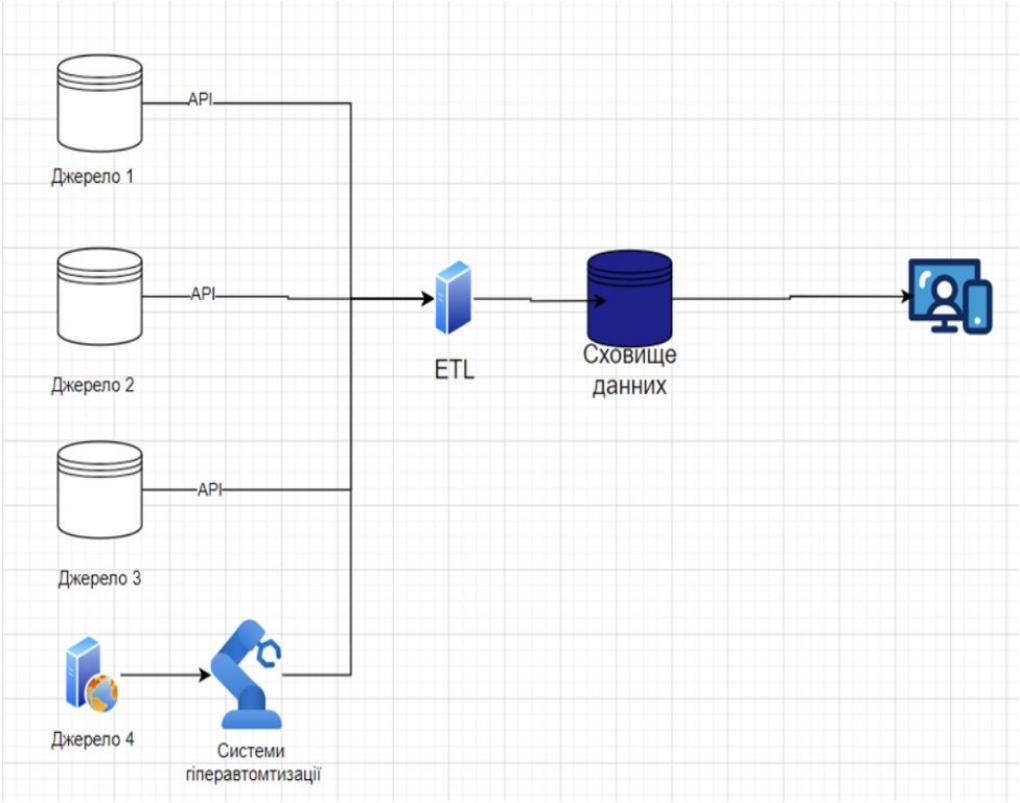

Рис. 1. Архітектура запропонованої моделі

Джерела даних (OLTP-бази та прикладні системи)

Джерело 1, 2, 3, 4 — транзакційні бази даних і пов'язані прикладні системи (CRM/ERP, державні реєстри, месенджери тощо), що обробляють події в реальному часі. На цьому рівні фіксуються первинні операції (замовлення, платежі, довідники), які слугують єдиним «джерелом істини» для подальшої аналітики[12].

Вхід: транзакції, табличні структури, журнали змін.

Вихід: інкрементальні дельти та/або snapshots для подальшої екстракції.

2) ETL-процес (Extract, Transform, Load

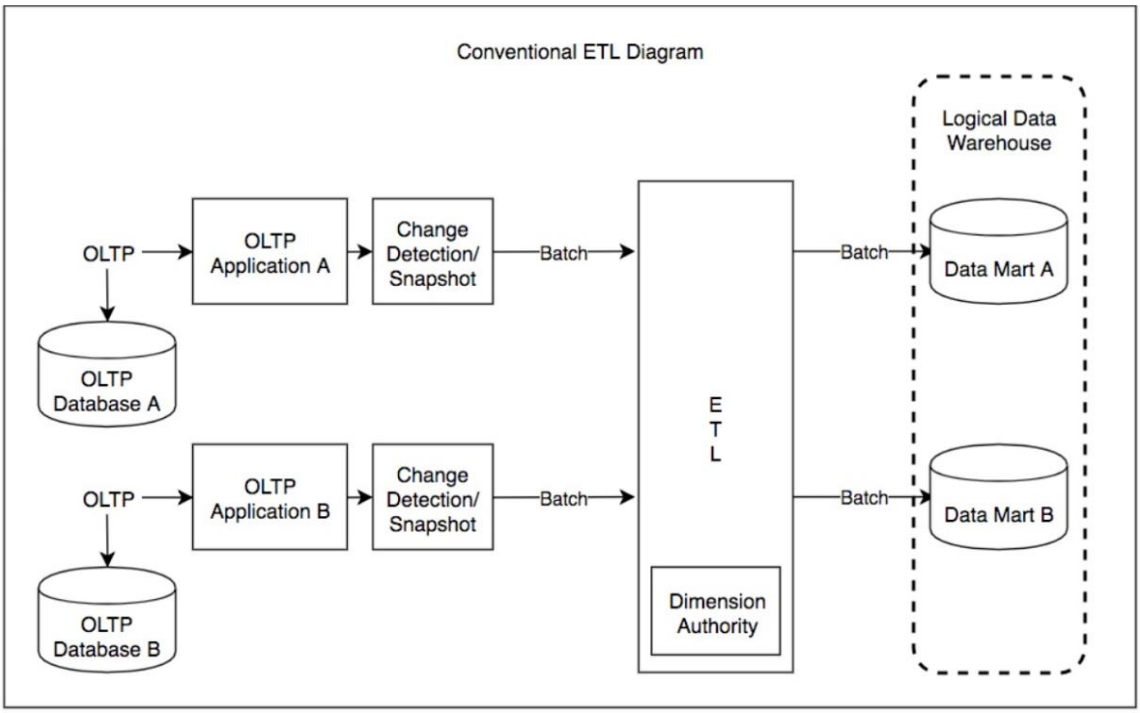

Рис. 2. Типова схема процесу ETL

Extract (Екстракція). На цьому етапі дані витягуються з джерел (бази даних та застосунків) за допомогою:

●За наявності API інтеграція в модуль ETL здійснюється через прямі API-виклики[13].

●Якщо можливості прямої інтеграції немає, використовується RPA-блок для збору даних (приклади: тендерні майданчики, месенджери тощо).

Transform (Трансформація). На етапі трансформації дані приводяться до єдиного формату, відповідного потребам аналітичної системи. Трансформації включають:

●очищення (видалення помилок, пропусків, некоректних значень);

●нормалізацію/денормалізацію;

●Обчислення, агрегації та інші операції, що підвищують якість даних для подальшого аналізу.

Load (Завантаження). На етапі завантаження дані передаються до сховища, яке використовуватиметься для подальшого аналізу й обробки ШІ (LLM). Це можуть бути спеціалізовані аналітичні сховища, зокрема векторні сховища. Завантаження може відбуватися пакетами (batch) або у потоковому режимі (near real-time/streaming).

Векторне сховище

Векторне сховище підключене до великої мовної моделі (LLM) і забезпечує поєднання індексованого пошуку (retrieval) з генеративними можливостями моделі. Вона є джерелом релевантного контенту для формування відповідей та аналітичних довідок.[14]

Інтерфейс користувача

Веб-портал з доступом до лінгвістичної моделі (LLM), з'єднаної з внутрішньою RAG-базою в системі обробки інформації на основі гіперавтоматизації. Інтерфейс забезпечує постановку запитів, керування параметрами вибірки/періодами, а також відображення результатів і звіт відпрацювання запитів.[15]

3.Формалізація математичної моделі

У загальному сенсі математичну модель процесу гіперавтоматизації доцільно переписати в уніфікованій нотації як:

$$PROC_{HIPER}: \{M_{BASE}, M_{RPA}\ ETL(M_{BASE} \cup M_{RPA}) \rightarrow$$

$M_{ETL}\, QUERY(M_{ETL}, lOBJ_{\Delta t}) \rightarrow REQUESTOBJ_{\Delta t}$, де M_{ETL}- с узгоджена множина після витягу/трансформацій/завантаження, а $OBJ_{\Delta t}$- семантичний опис цільового запиту в часовому інтервалі Δt.

Щоб зіставити з вашими позначеннями, фіксуємо:

●M_{BASE}- множина даних базової інформаційної системи;

●M_{RPA} - множина даних, отриманих засобами роботизованої автоматизації;

●$ETL(.)$- процедура інтеграції/очищення/узгодження (у т.ч. підготовка до донавчання LLM);

●M_{ETL} - інтегральна множина знань після застосування правил ETL та (за потреби) донавчання LLM;

●$QUERY(M_{ETL}, lOBJ_{\Delta t})$ - операція обробки цільового запиту над інтегрованими даними;

●$REQUESTOBJ_{\Delta t}$ - множина відповідей/висновків, релевантних запиту у вікні Δt.

Для узгодженості подальших розрахунків зручно явно ввести двокрокову інтеграцію:

$$M_{ETL} = ETL(M_{BASE} \cup M_{RPA}) M_{LTE} = \Phi(M_{ETL}; \theta)$$

тоді:

$$REQUESTOBJ_{\Delta t} = f(M_{LTE}, lOBJ_{\Delta t})$$

Детальний розгляд процедури $PROC_{RPA}(I_{RPA})$

Вхід. - I_{RPA} - множина альтернативних джерел щодо базової ІС (державні реєстри, публічні API, сайти, месенджери, тендерні майданчики тощо), кожне з яких має власні політики доступу, формат і частоту оновлення.

Вихід. M_{RPA} - консолідовані «сирі» події/документи з мінімальною нормалізацією (типи полів, часові мітки, джерело, ключі кореляції).

Кроки

●**Планування та оркестрація.** Встановлюються періодичність τ опитувань/читань, SLA на свіжість F_{max} (максимальне відставання), політика ретраїв r.

●**Аутентифікація/доступ.** Ініціалізація секретів (API-keys/OAuth, cookies, токени).

●**Екстракція.** Для кожного джерела $cs \in I_{RPA}$ виконується $Pull_s$ (API/скрапінг/конектор). Результат — множина записів R_s.

●**Первинна валідація.** Фільтри на порожні та явно пошкоджені записи; проставлення source_id, ingest_ts.

●**Легка нормалізація.** Приведення типів (дата/час/числа), локалізацій, кодувань; уніфікація основних полів («контрагент», «сума», «подія», «гео», «медіа-тип» тощо).

●**Контроль якості**. Обчислюються показники покриття Cov, свіжості Fresh, частоти дублікатів Dup та помилок Err:

$$Fresh = (t_{now} - t_x)\, Dup = \frac{|R_s| - |Unique(R_s)|}{|R_s|}$$

●**Журналювання й трасування.** Лінійка даних (data lineage): (s, ingest_ts, |R_s|, Err, Dup).

●**Видача.** $M_{RPA} = \cup_{s \in I_{RPA}} R_s$.

Таким чином:

$$M_{RPA} = PROC_{RPA}(I_{RPA}) M_{BASE} = PROC_BASE_API(\ )$$

Процедура $ETL(M_{BASE} \cup M_{RPA})$

Позначимо оператори E (Extract), T (Transform), L (Load) і C (узгодження вимірів/Dimension Authority). Тоді:

$\widehat{M} = T\big(E(M_{BASE} \cup M_{RPA})\big)$,

$M_{ETL} = L\left(C\left(\widehat{M}\right)\right)$

Ключові інваріанти.

(i) Ідемпотентність: повторний запуск без нових подій не змінює M_{ETL}.

(ii) Конформність вимірів: усі довідники/ієрархії уніфіковані; ключі стабільні у часі.

(iii) Відтворюваність: фіксована конфігурація трансформацій дає однаковий результат на однаковому вході.

Побудова M_{LTE} та обробка запиту

Інтеграція з LLM. $\Theta' = U(\Theta, M_{ETL})M_{LTE} = \Phi(M_{ETL}; \Theta')$, де $\mathcal{U}$ — оператор донавчання/адаптації (тонке донавчання, інструкційне донавчання, оновлення ембеддинг-індексів), а Φ — побудова представлень/індексів (у т.ч. для RAG).

Обробка запиту. Для запиту $l_{OBJ_{\Delta t}}$у вікні Δt:

$REQUESTOBJ_{\Delta t} = f\left(M_{LTE,\Delta t}, lOBJ_{\Delta t}\right)$

де f — композиція ретрівалу (за потреби R_k) і генерації відповіді LLM із політикою декодування, що сприяють відтворюваності (наприклад, T = 0, beam = 1). Для RAG частина контексту формується як:

$R_k(q) = \sum_{d \in S} cos\ cos\left(e(q), e(d)\right)$ де $e($ $\cdot)$

- функція ембеддингу, q — вектор запиту.

5. Алгоритм функціонування системи

У загальному вигляді алгоритм функціонування інформаційної системи можна відобразити на діаграмі.

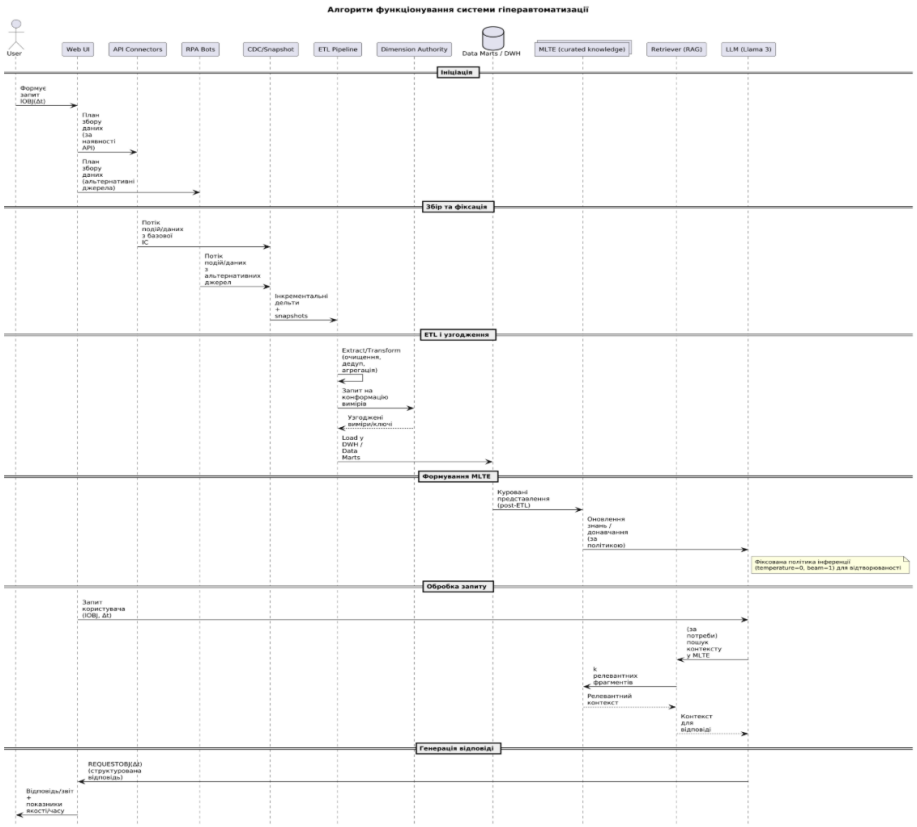

Рис.3. Алгоритм функціонування системи гіпераватоматизації

Збір даних через API та RPA

Система ініціює конектори до базової ІС (API) та альтернативних джерел (RPA-боти для реєстрів, веб-каналів, месенджерів).[16] На цьому кроці фіксуються атрибути походження (source_id), часові мітки (ingest_ts), а також виконується базова валідація та легка нормалізація (типи полів, кодування).

Виявлення змін і створення snapshot

Для мінімізації навантаження та збереження історичності застосовуються CDC (Change Data Capture) і періодичні знімки стану. Вихід — інкрементальні

дельти та/або snapshot-и, що передаються в конвеєр ETL разом із журналом змін.

Обробка даних у процесі ETL

Етап Extract уніфікує витяги з API/RPA; Transform виконує очищення, дедуплікацію, узгодження словників/ієрархій, обчислення агрегатів; Load завантажує результати у DWH/предметні дата-марти. Управління вимірами (Dimension Authority) гарантує конформність ключів та стабільність семантики.

Формування інтегральної множини MLTE

На основі узгоджених пост-ETL представлень формується інтегральна множина MLTE (curated knowledge), яка слугує єдиним джерелом для інференції та (за потреби) ретрівалу в RAG-ланцюжку.

Донавчання мовної моделі Llama 3[9]

Виконується адаптація параметрів моделі (тонке/інструкційне донавчання) або оновлення ембеддинг-індексу. Мета — інкорпорувати доменні знання з MLTE і забезпечити відтворювану поведінку (зафіксована політика декодування).

Обробка запиту користувача та генерація відповіді

Запит користувача з інтервалом Δt трансформується у внутрішнє подання; за потреби виконується ретрівал релевантного контексту з MLTE (RAG), після чого LLM генерує структуровану відповідь (класифікація, зведення, пояснення) з урахуванням часової локальності.

Висновок

У роботі запропоновано узгоджену модель гіперавтоматизації

(API/RPA → ETL → MLTE → LLM/RAG), що забезпечує трасованість, відтворюваність та керовану якість даних. Зафіксовано інваріанти інженерної частини , метрики потоку, а також підхід до детермінізації відповідей LLM і оптимізації ретривера. Модель застосовна у різних доменах як маркетингу, комерції та соціології.[17] У наступних наукових роботах буде розглянуто функціонування цієї моделі вже як робочого прототипу: подано опис архітектури виконання, сценаріїв застосування та методики емпіричної оцінки (SLO/SLA, latency, accuracy) на реальних даних.

Література

1. Spiridonov, A. (2026). Concept of information flow analysis technology within OSINT operations in instant messaging systems. Problems of Informatization and Control, 2(86), 134–145. DOI 10.18372/2073-4751.86.21282. Accessed 12.10.2026.

2. Gartner. (2022). Hyperautomation Strategic Roadmap. Gartner Research. Accessed 10.08.2024

3. van der Aalst, W. (2016). Process Mining: Data Science in Action. Springer. ISBN 978-3-662-49850-7. Accessed 08.08.2025

4. Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd ed.). Wiley. ISBN 978-1-118-53080-1. Accessed 12.10.2026.

5. Inmon, W. H., Linstedt, D., & Levins, M. (2019). Data Architecture: A Primer for the Data Scientist (2nd ed.). Academic Press. ISBN 978-0-12-816916-2. Accessed 12.10.2026.

6. Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33, 1877–1901. DOI 10.48550/arXiv.2005.14165. Accessed 12.10.2026.

7. Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems, 33, 9459–9474. DOI 10.48550/arXiv.2005.11401. Accessed 12.10.2026.

8. Izacard, G., Lewis, P., Lomeli, M., et al. (2023). Atlas: Few-shot Learning with Retrieval Augmented Language Models. Journal of Machine Learning Research, 24(251), 1–43. DOI 10.48550/arXiv.2208.03299. Accessed 12.10.2026.

9. Shrivastava, S. (2022). Robotic process automation architectures and processes for hosting, monitoring, and retraining machine learning models (European Patent Application EP 4031942 A1). UiPath, Inc. Accessed 12.10.2026.

10. He, Y., Chaudhuri, S., & Yang, J. (2024). Synthesizing multi-operator data transformation pipelines (U.S. Patent No. 11,880,344 B2). Microsoft Technology Licensing, LLC. Accessed 12.10.2026.

11. Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems (7th ed.). Pearson. ISBN 978-0-13-397077-7. Accessed 12.10.2026.

12. Kleppmann, M. (2017). Designing Data-Intensive Applications: The

Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media. ISBN 978-1-449-37332-0. Accessed 12.10.2026.

13. Pan, J. J., Wang, J., & Li, G. (2024). Survey of vector database management systems. The VLDB Journal, 33, 1591–1615. DOI 10.1007/s00778-024-00864-x. Accessed 12.10.2026.

14. Gao, Y., Xiong, Y., Gao, X., et al. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv preprint. DOI 10.48550/arXiv.2312.10997. Accessed 12.10.2026.

15. Syed, R., Suriadi, S., Adams, M., et al. (2020). Robotic Process Automation: Contemporary themes and challenges. Computers in Industry, 115, 103162. DOI 10.1016/j.compind.2019.103162. Accessed 12.10.2026.

16. Grattafiori, A., Dubey, A., Jauhri, A., et al. (2024). The Llama 3 Herd of Models. Meta AI. DOI 10.48550/arXiv.2407.21783. Accessed 12.10.2026.

17. Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., & Yang, D. (2024). Can Large Language Models Transform Computational Social Science? Computational Linguistics, 50(1), 237–291. DOI 10.1162/coli_a_00502. Accessed 12.10.2026.

Спірідонов А.Ю.

МАТЕМАТИЧНА МОДЕЛЬ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ІНФОРМАЦІЇ НА ОСНОВІ ГІПЕРАВТОМАТИЗАЦІЇ

У статті розроблено та формалізовано математичну модель системи інтелектуальної обробки інформації на основі концепції гіперавтоматизації. Запропонований підхід інтегрує джерела базових даних, інструменти роботизованої автоматизації процесів (RPA), процедури ETL та великі мовні моделі (LLM, зокрема Llama 3) з архітектурою RAG у єдину цілісну систему. Формалізовано двокрокову процедуру інтеграції даних та перетворення цільового запиту користувача у структуровану відповідь із зафіксованими метриками якості й часу. Визначено ключові інваріанти інженерної частини, такі як ідемпотентність, конформність вимірів і відтворюваність результатів. Запропонована модель забезпечує підвищення точності, швидкісних показників та трасованості обробки даних і є універсальною для застосування у сферах OSINT-розвідки, кібербезпеки, маркетингу та комерційного аналізу.

Ключові слова: *відкриті джерела інформації (OSINT), кібербезпека, гіперавтоматизація, інтелектуальна обробка інформації, великі мовні моделі, Llama 3, роботизована автоматизація процесів (RPA), машинне навчання, аналіз даних, штучний інтелект.*

Spiridonov, A.

MATHEMATICAL MODEL OF INTELLIGENT INFORMATION PROCESSING BASED ON HYPERAUTOMATION

The article develops and verifies a mathematical model of an intelligent information processing system based on the hyperautomation concept. The proposed approach integrates basic data sources, Robotic Process Automation (RPA) tools, ETL procedures, and Large Language Models (LLMs, specifically Llama 3) with a RAG architecture into a unified system. A two-step procedure for data integration and transformation of target user requests into structured responses with clearly defined quality and time metrics is formalized. Key engineering invariants, such as idempotency, dimension conformity, and result reproducibility, are established. The model enhances processing accuracy, speed, and data traceability, making it versatile for applications in OSINT intelligence, cybersecurity, marketing, and business analytics.

Keywords: Open Source Intelligence (OSINT), cybersecurity, hyperautomation, intelligent information processing, large language models, Llama 3, robotic process automation (RPA), machine learning, data analysis, artificial intelligence.

Стаття подана до редакції: 11*/09/2026*
Стаття прийнята до опублікування:18/09/2026
Стаття опублікована: 10/11/2026

Стаття поширюється на умовах ліцензії CC BY 4.0