

**МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА  
ІНТЕРНЕТ-КОНФЕРЕНЦІЯ**

**«ТЕНДЕНЦІЇ ТА ПЕРСПЕКТИВИ РОЗВИТКУ  
НАУКИ І ОСВІТИ  
В УМОВАХ ГЛОБАЛІЗАЦІЇ»**



**ВИПУСК 123**

**28 листопада 2025 р.**

**м. Переяслав**

УНІВЕРСИТЕТ ГРИГОРІЯ СКОВОРОДИ  
В ПЕРЕЯСЛАВІ  
РАДА МОЛОДИХ УЧЕНИХ УНІВЕРСИТЕТУ  
ГО «ІНСТИТУТ СУСПІЛЬНОГО РОЗВИТКУ  
І НАУКОВИХ ДОСЛІДЖЕНЬ»

Матеріали

Міжнародної науково-практичної інтернет-конференції  
**«ТЕНДЕНЦІЇ ТА ПЕРСПЕКТИВИ РОЗВИТКУ  
НАУКИ І ОСВІТИ В УМОВАХ ГЛОБАЛІЗАЦІЇ»**

28 листопада 2025 року

Вип. 123

Збірник наукових праць

Переяслав – 2025

- Державна та громадська підтримка: Створення державних програм навчання та консультування МСБ з питань кібербезпеки, розвиток спільнот для обміну досвідом.
- Інтеграція AI в доступні продукти: Впровадження штучного інтелекту в масові антивірусні рішення та платформи безпеки для більш ефективного виявлення аномалій.
- Стандартизація базових заходів безпеки: Розробка простих чек-листів і стандартів, адаптованих саме для потреб МСБ.

### **Висновки.**

#### **Практичні рекомендації для малого та середнього бізнесу.**

1. Розробити базову політику кібербезпеки, яка регулює використання паролів, роботу з даними, оновлення ПЗ.
2. Обов'язково впровадити MFA для всіх сервісів, що його підтримують.
3. Налаштувати автоматичне резервне копіювання і регулярно перевіряти процес відновлення даних.
4. Проводити щоквартальні тренінги з кібербезпеки для всіх співробітників.
5. Використовувати захищені хмарні сервіси замість власних серверів, де це можливо.
6. Обмежити права доступу співробітників до даних і систем, які їм не потрібні для роботи.

### **ДЖЕРЕЛА ТА ЛІТЕРАТУРА**

1. Small Business Administration (SBA). Cybersecurity for Small Businesses. URL: <https://www.sba.gov/business-guide/manage-your-business/cybersecurity-small-businesses>
2. National Institute of Standards and Technology (NIST). Small Business Cybersecurity Corner. URL: <https://www.nist.gov/itl/smallbusinesscyber>
3. Verizon. 2023 Data Breach Investigations Report (DBIR). URL: <https://www.verizon.com/business/resources/reports/dbir/>
4. ENISA (European Union Agency for Cybersecurity). Threat Landscape for Supply Chain Attacks. URL: <https://www.enisa.europa.eu/publications/threat-landscape-for-supply-chain-attacks>
5. Cofense (formerly PhishMe). Phishing Infographic. URL: <https://cofense.com/wp-content/uploads/2023/01/Phishing-Infographic-2023.pdf>
6. Coveware. Q4 2023 Ransomware Payment Amounts. URL: <https://www.coveware.com/blog/2024/1/25/q4-2023-ransomware-marketplace-report>
7. CISA (Cybersecurity and Infrastructure Security Agency). Remote Work Security. URL: <https://www.cisa.gov/secure-our-world/remote-work-security>

*Науковий керівник: кандидат фізико-математичних наук, доцент Козін О. Б.*

**УДК 004.89**

*Роман Гануля, Нестор Кресінський, Юлія Тимошенко, Ігор Козбур  
(Тернопіль, Україна)*

### **ІНТЕЛЕКТУАЛЬНІ ПІДХОДИ ДО АВТОМАТИЗАЦІЇ ДІЙ КОРИСТУВАЧА ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА НА ОСНОВІ МАКРОСИСТЕМ ТА СКРИПТОВИХ МОВ**

*У статті наведено розширений аналіз інтелектуальних підходів до автоматизації дій користувача персонального комп'ютера на основі макросистем, скриптових мов, процес-майнінгу та агентних моделей штучного інтелекту. Порівняно класичні та сучасні інструменти автоматизації, проаналізовано роль HCI та перспективи впровадження GUI-агентів на основі великих мовних моделей у різні сфери діяльності.*

**Ключові слова:** автоматизація, макросистеми, скриптові мови, GUI-агенти, LLM, RPA, процес-майнінг.

*The article provides an extended analysis of intelligent approaches to automating user actions on personal computers using macro systems, scripting languages, process mining and agent-based AI models. Classical automation tools are compared with modern intelligent systems, highlighting the importance of HCI and discussing the prospects for implementing LLM-powered GUI agents in various fields.*

**Key words:** automation, macro systems, scripting languages, GUI agents, LLM, RPA, process mining.

Автоматизація взаємодії користувача з персональним комп'ютером на сьогодні є одним із пріоритетних напрямів розвитку інформаційних технологій, оскільки зростання складності програмного забезпечення та збільшення кількості щоденних операцій вимагає від людини все більшої концентрації та витрат часу. Більшість сучасних робочих процесів у галузі освіти, науки, бізнесу, управління, промисловості та навіть побутових цифрових середовищ включають виконання повторюваних, формалізованих та добре структурованих дій. До них належать обробка інформації у табличних редакторах, переміщення файлів та каталогів, взаємодія з веб-інтерфейсами, введення даних у програмні системи, виконання регламентованих операцій у бухгалтерських, адміністративних та управлінських середовищах. Накопичення цих завдань формує значне навантаження на користувача, що знижує продуктивність, оскільки рутинні операції не залишають достатньо часу на інтелектуальні, творчі чи стратегічні види діяльності.

Потреба автоматизації таких процесів призвела до появи перших інструментів у вигляді макросистем, які дозволяли користувачу записати послідовність виконуваних дій з метою їх відтворення у майбутньому. Це стало революційним кроком свого часу, адже дозволило прискорити роботу без глибокого технічного розуміння механізмів операційної системи чи програмного забезпечення. Проте відсутність адаптивності у макросистем зробила їх придатними лише у стабільних середовищах, де інтерфейси та поведінка програм не змінюються. Макроси реагують виключно на координати й передбачену послідовність подій, що робить їх вразливими до навіть незначних змін: переміщення кнопки, зміни кольорової теми, збільшення масштабу інтерфейсу, оновлення версії програми або нестандартної системної затримки. Вони не здатні аналізувати текст елементів, їх логіку, не можуть розрізнити повідомлення про помилки й не розуміють причинно-наслідкових зв'язків між діями.

Сучасні дослідження, зокрема робота Huang та співавторів [3], демонструють можливість автоматичного виявлення повторюваних шаблонів взаємодії користувачів на основі аналізу потоків дій і логів системної активності. Це формує передумови до створення інтелектуальних макросистем нового покоління, які замість простого запису взаємодії будують узагальнені моделі поведінки, виявляють регулярності та оптимізують послідовності дій, зменшуючи дублювання операцій. Такий підхід переходить від низькорівневого відтворення подій миші та клавіатури до «розуміння цілей» користувача через повторюваність його дій.

Проте справжній прорив стався з розвитком скриптових мов, які дали можливість створювати адаптивні, умовно керовані сценарії, що працюють не лише на рівні запису взаємодії, а на рівні програмної логіки. Мови Python, AutoHotkey, PowerShell, JavaScript та інші інструменти надали користувачу доступ до системних API, файлової системи, мережевих протоколів, бібліотек роботи з інтерфейсом, подій клавіатури та миші, можливостей асинхронної взаємодії та керування зовнішніми процесами. На відміну від класичних макросистем, вони дозволили створювати складні сценарії, здатні реагувати на зміну умов, аналізувати стани системи та відгалужуватися залежно від результатів виконання.

Скриптові системи відкрили нові можливості автоматизації на рівні операційної системи: робота з реєстром Windows, запуск і моніторинг процесів, читання логів,

керування службами, обробка винятків, взаємодія з COM-об'єктами, робота з HTTP-запитами та JSON-даними, автоматизація завдань у хмарних API, а також інтеграція з зовнішніми модульними бібліотеками. Наприклад, AutoHotkey дозволяє не лише надсилати натискання клавіш або рухи миші, а й виконувати повноцінний аналіз вмісту вікон через WinAPI, працювати з буфером обміну, створювати гарячі клавіші та динамічні GUI-форми для керування складними макрологіками. PowerShell, у свою чергу, забезпечує високорівневу інтеграцію з усією інфраструктурою Windows та Azure, дозволяючи автоматизувати сотні внутрішніх процедур, недоступних для макросистем.

Python, як універсальна мова програмування, стала інструментом для автоматизації практично будь-якого рівня, від системного до мережевого. З її допомогою створюють сценарії для збору інформації з вебсторінок (через бібліотеки BeautifulSoup, Selenium або Playwright), здійснюють автоматичне формування звітів (через pandas, docx або openpyxl), обробляють зображення (через Pillow чи OpenCV), будують локальні сервери, ініціюють хмарні сервіси та взаємодію з API різних платформ. Наприклад, Python-скрипт здатний автоматично запустити хмарний сервіс, пройти авторизацію через OAuth, зчитати дані з REST API, обробити їх у DataFrame, сформувати звіт у PDF через ReportLab, зберегти його на диск, синхронізувати з Google Drive та відправити електронною поштою користувачу – усе це без участі людини.

Особливо вагомою перевагою скриптів стало те, що вони дозволяють обробляти виняткові ситуації. Якщо макрос, записаний користувачем, стикається з неочікуваним повідомленням або зміною в інтерфейсі, він просто зупиняється. Натомість скрипт забезпечує роботу механізмів обробки помилок (try/except), часових затримок, перевірки умов (if/else), циклічних спроб виконання дії, логування та відновлення після збоїв. Це робить скриптові системи значно більш стійкими до непередбачуваних ситуацій, ніж макроси.

Однак, незважаючи на свою гнучкість, такі системи залишаються обмеженими у ситуаціях, коли доступ до API відсутній або коли програма не надає способу отримати структуровані дані. У таких випадках скрипти змушені імітувати взаємодію на поверхневому рівні, – через рухи миші, симуляцію клавіш або OCR. Це робить скриптову автоматизацію залежною від стабільності інтерфейсу. На додаток, складність створення скриптів зростає пропорційно складності програм з якими потрібно взаємодіяти. Якщо інтерфейс має численні вкладені меню, контекстні елементи, спливаючі модальні вікна або динамічний вміст, вартість розробки скриптів значно зростає.

Ще одним обмеженням є те, що скриптові інструменти не здатні «розуміти» графічний інтерфейс на семантичному рівні. Вони можуть зчитати текст, можуть визначити положення елемента через селектори (наприклад, у браузерній автоматизації), але не можуть інтерпретувати логіку інтерфейсу так, як це робить людина: зрозуміти, що певна кнопка відповідає за підтвердження операції, а інша – за перехід до нового етапу. Скриптові системи працюють на рівні низькорівневих дій і не мають когнітивного компонента.

Таким чином, хоча скриптові мови створили значний крок уперед у гнучкості й адаптивності автоматизації, вони все ж залишаються залежними від стабільності інтерфейсу, наявності API та передбачуваності поведінки програм. У динамічних або складних середовищах вони можуть потребувати значних зусиль підтримки, а у випадках зміни логіки роботи програм – повного переписування сценаріїв.

Поява великих мовних моделей (LLM) докорінно змінила можливості автоматизації. Такі LLM як GPT (OpenAI), DeepSeek (DeepSeek), Grok (xAI), Llama (Meta), Gemini (Google) та Claude (Anthropic) та інші аналогічні за функціоналом, навчені на величезних масивах текстів, мають здатність виконувати загальні когнітивні функції, – інтерпретувати інструкції, аналізувати контекст, прогнозувати наступні кроки та оцінювати правильність дій. LLM можна використовувати не лише для генерації тексту, а й як високорівневий мозковий центр інтелектуальних програмних агентів. Дослідження Zhang та ін. [2]

описують GUI-агентів на основі LLM, які здатні взаємодіяти з комп'ютером у спосіб, максимально наближений до людського. Вони розпізнають вміст екрана, аналізують структуру інтерфейсу, знаходять потрібні елементи, розуміють логіку розмітки, реагують на діалогові вікна та повідомлення, інтерпретують форми і функції кнопок.

Унікальність LLM-агентів полягає в їхній здатності працювати з інтерфейсом не на основі координат, а на основі семантики [7]. Наприклад, агент може натиснути кнопку не тому, що вона знаходиться за координатами X,Y, а тому, що вона виконує операцію створення нового документа, незалежно від її зовнішнього вигляду або місця. Він може прочитати повідомлення про помилку, визначити її причини, проаналізувати належність до поточного завдання і самостійно запропонувати шлях вирішення. Такі можливості кардинально розширюють межі автоматизації, дозволяючи виконувати завдання у більшості середовищ, котрі не передбачали жодної автоматизації.

Важливим аспектом є здатність LLM працювати на основі людської природної мови. Використання текстових інструкцій дає змогу виконувати автоматизацію у форматі «завдання → рішення», без необхідності для користувача навчатися програмуванню. Це значно розширює коло людей, які можуть використовувати автоматизацію, включаючи гуманітаріїв, менеджерів, студентів та представників адміністративних професій. Користувач може просто сказати: «Відкрий цю папку, знайди всі PDF-файли за останній місяць, перейменуй їх за вказаним шаблоном і відправ на мою електронну пошту». Агент виконає це як автономний виконавець. Більше того, якщо агент зіткнеться з перешкодою, він може самостійно уточнити завдання, поставивши питання [1].

Не менш важливим для автоматизації є врахування впливу людсько-комп'ютерної взаємодії (HCI) в подібних системах. Робота Hornbæk, Kristensson та Oulasvirta [1] підкреслює, що ефективні інтерфейси повинні враховувати можливості та обмеження людини, а саме, обсяг короткочасної пам'яті, швидкість прийняття рішень, втомлюваність, когнітивне навантаження. Інтелектуальні агенти повинні враховувати поведінкові моделі користувачів та їх індивідуальні звички, від розташування вікон до послідовності дій. Системи, котрі не враховують HCI, стають менш передбачуваними і можуть порушити логіку роботи користувача.

Окремо варто розглянути роботизовану автоматизацію процесів (RPA), котра вже багато років використовується для оптимізації рутинних процесів. [5] Проте класичний RPA обмежений жорстко прописаними сценаріями, що робить його чутливим до змін інтерфейсу та логіки системи. Інтеграція RPA з процес-майнінгом, як показано в роботі El-Gharib та Amyot [4], дозволяє автоматично аналізувати логи взаємодій і будувати узагальнені моделі бізнес-процесів. Це дає можливість не просто відтворювати сценарії, а й виявляти проблеми, – дублювання кроків, неефективні рішення та надмірні операції. У поєднанні з LLM та GUI-агентами такі системи здатні створювати автоматизовані адаптивні цифрові робочі процеси, що змінюються в залежності від зовнішніх умов [6]. Синергія LLM-агентів, процес-майнінгу, скриптових мов, комп'ютерного зору та макросистем дозволяє створювати інтелектуальні системи автоматизації нового покоління. Вони не лише виконують завдання, але й розуміють контекст, адаптуються до умов, оптимізують логіку своєї роботи, працюють у нестандартних інтерфейсах, виправляють власні помилки, формують моделі поведінки користувача та навчаються на історії взаємодій. У перспективі це може призвести до появи повністю автономних цифрових помічників, здатних виконувати більшість операцій без участі людини.

Підсумовуючи, інтелектуальні підходи до автоматизації користувацьких дій формують нову парадигму взаємодії людини з комп'ютером. Вони дозволяють поєднати класичні інструменти автоматизації з можливостями штучного інтелекту, забезпечуючи високий рівень адаптивності, автономності та надійності. Така автоматизація не лише спрощує виконання завдань, але й змінює саму структуру цифрової праці, суттєво підвищуючи ефективність у різних сферах діяльності.

#### ДЖЕРЕЛА ТА ЛІТЕРАТУРА

1. Hornbæk K., Kristensson P. O., Oulasvirta A. Introduction to Human – Computer Interaction. Oxford University Press, 2025. 412 p.
2. Zhang C., Li Y., Zhang J., et al. Large Language Model-Brained GUI Agents: A Survey. arXiv preprint arXiv:2401.00000, 2024. 78 p.
3. Huang F., Feng J., Chen X., et al. Automatic Macro Mining from Interaction Traces at Scale. arXiv preprint arXiv:2304.00000, 2023. 25 p.
4. El-Gharib N. M., Amyot D. Robotic Process Automation Using Process Mining. arXiv preprint arXiv:2205.00000, 2022. 30 p.
5. Van der Aalst W. M., Bichler M., Heinzl A. Robotic Process Automation. Business & Information Systems Engineering, 2018, Vol. 60, pp. 269–272.
6. Jimenez-Ramirez A., Reijers H. A., Barba I., Del Valle C. A method to improve the early stages of the robotic process automation lifecycle. In: CAiSE 2019. Springer, 2019. pp. 446–461.
7. Arsan D., Zaidi A., Sagar A., Kumar R. App-Based Task Shortcuts for Virtual Assistants. In: Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21). ACM, 2021. pp. 1089–1099.

УДК 004.8:504.3.054

*Баянали Досжанов, Нұрбек Ерғали  
(Қызылорда, Қазақстан)*

#### МАШИНАЛЫҚ ОҚЫТУ АРҚЫЛЫ АУА САПАСЫН БОЛЖАУ

*Бұл мақалада ауа сапасын болжау мәселесі қарастырылады. Қоршаған ортаның көрсеткіштері (температура, ылғалдылық, жел жылдамдығы, газ концентрациясы) негізінде Python тілінде машиналық оқыту әдістерін қолдану арқылы ауа сапасының индексі (AQI) болжау жолдары көрсетіледі. Мақалада деректерді алдын ала өңдеу, модельді құру және нәтижелерді талдау кезеңдері сипатталады. Сонымен қатар, ауа сапасын болжаудың қоғам үшін маңызы мен болашақтағы қолданылу мүмкіндіктері талқыланды.*

***Кілт сөздер:** ауа сапасы, AQI, машиналық оқыту, болжау, Python, деректерді алдын ала өңдеу, қоршаған орта параметрлері, газ концентрациясы, регрессия модельдері, экологиялық мониторинг.*

*This article discusses the problem of forecasting air quality. Based on environmental indicators (temperature, humidity, wind speed, gas concentration), it demonstrates the use of machine learning methods in Python to predict the Air Quality Index (AQI). The article describes the stages of data preprocessing, model development, and result analysis. In addition, the importance of air quality forecasting for society and its potential future applications are examined.*

***Keywords:** air quality, AQI, machine learning, forecasting, Python, data preprocessing, environmental parameters, gas concentration, regression models, environmental monitoring.*

Ауа сапасы – адамның денсаулығына, өмір сүру деңгейіне және экологиялық тұрақтылыққа тікелей әсер ететін маңызды фактор. Атмосфераның ластануы қазіргі таңда ірі қалаларда көлік қозғалысының артуы, өнеркәсіптік кәсіпорындардың жұмысы және климаттың өзгеруі салдарынан күрделене түсуде. Бұл жағдай халықтың денсаулығына кері әсер етіп, тыныс алу және жүрек-қан тамыр жүйесінің ауруларының көбеюіне әкелуде. Соңғы жылдары жасанды интеллект пен машиналық оқыту технологиялары үлкен деректерді (Big Data) талдау арқылы ауа сапасын болжауда тиімді құрал ретінде кеңінен қолданылып келеді [1]. Мұндай әдістер ауа құрамындағы зиянды газдардың (CO<sub>2</sub>, NO<sub>2</sub>,

