УДК 004.056:004.8:004.774
DOI: https://doi.org/10.17721/ISTS.2025.9.11-17

Сергій БУЧИК, д-р техн. наук, проф.
ORCID ID: 0000-0003-0892-3494
e-mail: buchyk@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Віталій П'ЯТИГОР, асп.
ORCID ID: 0000-0002-7621-1299
e-mail: vp5gor@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

АРХІТЕКТУРА СИСТЕМ ВИЯВЛЕННЯ АВТОМАТИЗОВАНИХ АКАУНТІВ (БОТІВ) У СОЦІАЛЬНИХ МЕРЕЖАХ

Вступ. ***Соціальні мережі є одним з основних джерел отримання інформації. Проте зі зростанням довіри до них зростає і використання автоматизованих акаунтів для поширення дезінформації та впливу на суспільну думку, що є ефективним методом маніпуляції. Для виявлення таких акаунтів застосовують системи виявлення автоматизованих акаунтів. Метою статті є розгляд архітектури систем виявлення автоматизованих акаунтів (ботів) у соціальних мережах і визначення основних напрямів щодо вдосконалення такої архітектури з урахуванням виявлених недоліків.***

Методи. ***Використано методи аналізу, систематизації та узагальнення для визначення недоліків і напрямів удосконалення систем і метод моделювання для розроблення загальної та вдосконаленої архітектури систем.***

Результати. ***Запропоновано вдосконалену архітектуру системи виявлення автоматизованих акаунтів (ботів) у соціальній мережі x.com, що використовує змішану модель аналізу даних за допомогою штучного інтелекту й об'єднує вебскрейпінг та API запити для збору даних.***

Висновки. ***Розглянуто загальну архітектуру систем виявлення автоматизованих акаунтів (ботів) у соціальних мережах і визначено основні недоліки та напрями вдосконалення, серед яких можна виділити обмеження запитів до API від власників соціальних мереж, неструктурований і вільний стиль дописів у соціальних мережах і постійну зміну алгоритмів генерації дописів автоматизованими акаунтами. Запропоновано вдосконалену архітектуру системи виявлення автоматизованих акаунтів ботів, що поєднує вебскрейпінг та API запити для збору даних і дозволяє паралельне виконання аналізу даних соціальних мереж.***

Ключові слова: ***комп'ютерна система, збір даних, аналіз даних, машинне навчання.***

Вступ

Соціальні мережі стали одним з основних джерел інформації, забезпечуючи майже миттєву доставку та часто анонімне використання. Вони стали основним компонентом засобів комунікації для окремих осіб та організацій. Наприклад, у квітні 2024 р. мережа "FacebookC налічувала 3 млрд користувачів, які були активними хоча б раз на місяць (Dixon, 2024).

Зі зростанням впливу соціальних мереж, вони стали зброєю, доступною для будь-кого. Хоча більшість користувачів використовують ці платформи для побутових цілей, існують такі особи й організації, які використовують соціальні мережі для прихованих цілей. Щоб використовувати силу соціальних мереж, окремим особам або організаціям необхідно вплинути на їхні алгоритми трендів і взаємодії з дописами, тому створюють автоматизовані аккаунти або боти. Боти соціальних мереж – це комп'ютерні алгоритми, які створюють контент і взаємодіють із користувачами (Ferrara et al., 2016). Згідно зі статистичними даними, в останньому кварталі 2023 р. з мережі "Facebook" було видалено 691 млн фейкових облікових записів (Dixon, 2025), тоді як за оцінками у 2020 р. у мережі "X" (раніше "Twitter") було 48 млн облікових записів ботів, що становило 15 % усіх облікових записів (Guy et al., 2012). Ця присутність розмиває цілісність інформації, яка поширюється на цих платформах і сприяє створенню середовища, в якому користувачі відчувають невпевненість щодо автентичності інших облікових акаунтів і вмісту.

Одним із способів використання таких мереж ботів є кампанії дезінформації. Вони стали інструментом, який дозволяє державним і недержавним акторам поширювати неправдиві наративи, маніпулювати громадською думкою та дестабілізувати спільноти. Ці боти виглядають як справжні користувачі, що ускладнює їх виявлення та дозволяє їм працювати в такому масштабі, який був би неможливий для самих людей. Такі мережі у великих обсягах задокументовано під час президентських виборів у США 2016 р. (Bessi, & Ferrara, 2016), і ці обсяги зросли після повномасштабного вторгнення росії в Україну (Fredhaim, & Stolze, 2022).

Системи виявлення автоматизованих акаунтів дозволяють виявляти та приймати рішення щодо шкідливої активності, яка здійснюється ботами у соціальних мережах. Ефективне виявлення ботів є критично важливим для забезпечення безпеки користувачів, збереження достовірності інформаційного простору та підтримки довіри до цифрових платформ. Деякі дослідники працюють над класифікацією автоматично створених дописів, а не над виявленням автоматизованих акаунтів (Almerekhi, & Elsayed, 2015). Інші зосереджені на прогнозуванні потенційних цілей атак як підходи захисту (Boshmaf et al., 2016). Але найпопулярнішим механізмом захисту є виявлення облікових акаунтів ботів, індивідуальних чи групових (ботнетів), на ранніх стадіях (створення облікових записів) або після початку їхньої роботи в соціальних мережах.

Метою цієї статті є розгляд архітектури систем виявлення автоматизованих акаунтів (ботів) у соціальних мережах і визначення основних напрямів удосконалення такої архітектури з урахуванням виявлених недоліків.

Об'єктом дослідження є процес виявлення автоматизованих акаунтів (ботів) у соціальних мережах.

Предметом дослідження є архітектури систем виявлення автоматизованих акаунтів (ботів) у соціальних мережах.

Отже, основне завдання полягає в удосконаленні архітектури систем виявлення автоматизованих акаунтів (ботів) у соціальних мережах.

ISSN 2707-1758

Методи

В роботі використано такі методи дослідження.

1. Метод аналізу застосовано для детального розгляду існуючих систем виявлення автоматизованих акаунтів у соціальних мережах. Він дозволив виявити їхні складові частини, принципи роботи та потенційні недоліки.

2. Метод систематизації використано для впорядкування інформації, отриманої в результаті аналізу. Це дозволило класифікувати недоліки існуючих систем і визначити загальні напрями для їхнього вдосконалення.

3. Метод узагальнення, застосування якого дало змогу сформулювати загальні висновки щодо проблем і перспектив розвитку систем виявлення ботів, а також запропонувати ширші напрями для вдосконалення архітектури.

4. Метод моделювання використано для розроблення як загальної архітектури систем виявлення ботів, так і запропонованої удосконаленої архітектури для конкретної соціальної мережі (x.com). Це дозволило візуалізувати запропоновані зміни та їхній потенційний вплив. Отже, основними методами дослідження, використаними в цій роботі, є аналіз, систематизація, узагальнення та моделювання. Ці методи дозволили авторам дослідити існуючий стан проблеми, виявити недоліки та запропонувати власне рішення у вигляді вдосконаленої архітектури системи виявлення автоматизованих акаунтів.

Огляд літератури. У попередніх роботах авторів розглянуто основні етапи систем виявлення автоматизованих акаунтів і проблем, які необхідно розв'язати для розроблення таких систем (П'ятигор, & Бучик, 2025). Було виділено такі етапи: збір даних, попереднє оброблення й аналіз даних. З урахуванням даних, представлених у роботі, можемо подати загальну архітектури систем виявлення автоматизованих акаунтів, що зображена на рис. 1. Додатково виділено етап представлення результатів і компонент бази даних. Успіх таких систем залежить від інтеграції сучасних технологій, врахування нових методів обходу захисту та постійного оновлення моделей, що використовуються для аналізу.

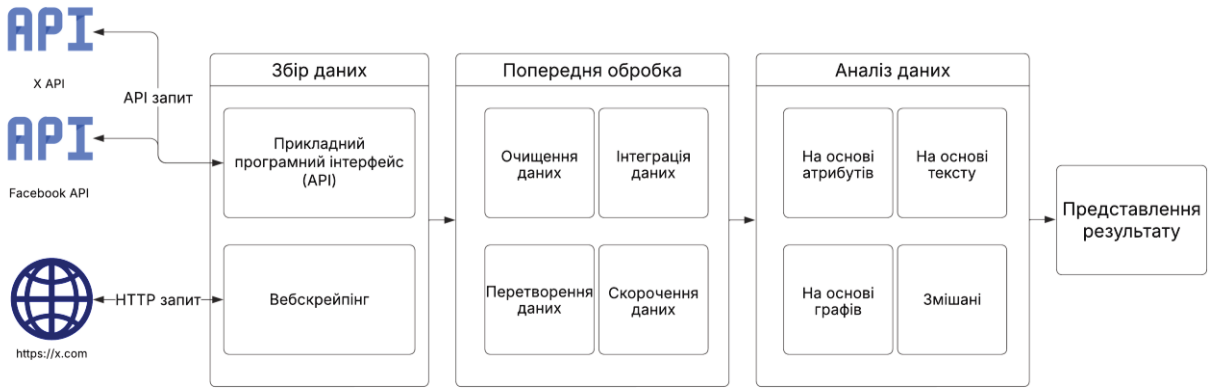

Рис. 1. Загальна архітектура систем виявлення автоматизованих акаунтів у соціальних мережах

На етапі аналізу даних окремим підетапом можна виділити компонент навчання моделі, оскільки більшість із методів використовують засоби машинного навчання для аналізу даних. Цей компонент безсумнівно потрібен у випадку, коли модель не використовувалася до цього і не була навчена, або модель є еволюційною, тобто такою, що постійно навчається. Навчальна модель – це набір даних, який використовується для навчання алгоритму машинного навчання. Він складається з вибірки вихідних даних і відповідних наборів вхідних даних, які впливають на вихід. Навчальну модель застосовують для проходження вхідних даних через алгоритм, щоб співвіднести оброблений вихід із зразком. Результат цієї кореляції використовують для модифікації моделі. Цей ітеративний процес називають "допасовування моделі". Навчання моделі машинною мовою – це процес подачі даних в алгоритм, щоб допомогти визначити та вивчити впливові значення для всіх задіяних атрибутів. Існує кілька типів моделей машинного навчання, з яких найпоширенішими є контрольоване та неконтрольоване навчання.

Контрольоване навчання можливе, коли навчальні дані містять і вхідні, і вихідні значення. Кожен набір даних, який має вхідні й очікувані вихідні дані, називають контрольним сигналом. Навчання виконується на основі відхилення обробленого результату від задокументованого результату, коли вхідні дані вводять у модель.

Навчання без нагляду передбачає визначення закономірностей у даних. Потім додаткові дані використовують для підгонки шаблонів або кластерів. Це також ітераційний процес, який покращує точність на основі кореляції з очікуваними шаблонами або кластерами. У цьому методі немає еталонного вихідного набору даних.

За можливості, в системах варто використовувати алгоритми самостійного навчання або навчання без нагляду для підвищення ефективності визначення автоматизованих акаунтів, проте треба пам'ятати, що навчання без нагляду залежить від якості даних, які надаються для навчання. Це означає, що все більшу увагу варто привертати до етапу попереднього оброблення даних, які потім будуть передаватися на навчання й аналіз через велику неузгодженість даних у соціальних мережах.

Іншим опціональним компонентом є база даних. Хоча всі дані, які обробляються на будь-якому з етапів, можна зберігати в оперативній пам'яті, більш практичним підходом є використання бази даних для збереження проміжних результатів. Як мінімум, варто зберігати остаточні результати аналізу даних у базах даних, що дозволить збирати статистичні дані отриманих результатів. Основними моделями баз даних

ISSN 2707-1758

для задач зберігання великої кількості даних є реляційна і NoSQL. Реляційна модель є надійнішою з погляду цілісності даних, але дані, які зберігаються, мають бути представлені в нормальних формах і мати чітко визначений формат. Така база даних може бути використана для зберігання фінальних результатів у форматі – опис даних і результат аналізу, проте для зберігання даних із попередніх етапів, така база даних не є достатньо гнучкою і оптимізація запитів є доволі складною задачею.

NoSQL бази даних або документо-орієнтовані бази даних не мають вимоги до структуризації даних у нормальних формах і в більшому представленні у форматі ключ-значення. Такі бази даних є дуже гнучкими до формату та розміру даних і є відкритими до доповнення блоку даних новою парою – ключ-значення. Це означає, що на кожному етапі оброблення ми можемо додавати результат оброблення як ключ-значення до існуючого запису. Оскільки документи найчастіше мають формат JSON, їх набагато легше читати та розуміти користувачеві. Документи, які зберігаються, також є цілим записом, тобто немає необхідності посилатися на різні таблиці для відображення результату, що є проблемою в реляційних базах. Бази даних документів також мають високу масштабованість. На відміну від реляційних баз даних, де традиційно можна масштабувати лише вертикально, нереляційні бази даних, зокрема і бази даних документів, можна масштабувати горизонтально. Це означає, що бази даних дублюються на кількох серверах, але залишаються синхронізованими.

Останнім етапом оповіщення системи є представлення результату користувачеві. Це може бути і текст у консолі застосунку про те, що користувач є ботом, або додаток, який отримує дані та представляє результат аналізу, впевненість системи в результаті, дані щодо того, на основі яких параметрів було прийнято рішення, тощо. Використання баз даних дають можливість застосунку відображати не тільки останній результат аналізу, а й історичні дані протягом усього часу роботи системи. Також результати можуть бути представлені у таких засобах візуалізації як PowerBI або Google Charts. Це дозволяє генерувати інфографіки та звіти, які є компактним та інформативним способом представити результати роботи системи без необхідності, щоб кінцевий користувач розумів весь обсяг даних.

Результати

З огляду на недоліки, описані вище, можна виділити можливі напрями для вдосконалення таких систем. Деякі з таких напрямів удосконалення авторами представлено у табл. 1, які поділені поетапно.

Таблиця 1

Напрями вдосконалення компонентів системи виявлення автоматизованих акаунтів

Етап	Напрям удосконалення
Збір даних	Оптимізація запитів до API
	Підвищення функціоналу відкритих до змін API та їхньої документації
	Постійна адаптація методів вебскрейпінгу до оновлень дизайну вебсторінок
	Законодавче врегулювання засобів вебскрейпінгу в Україні
Попереднє оброблення	Адаптація методів попереднього оброблення до даних соціальних мереж
Аналіз даних	Постійна адаптація методів до замаскованих даних
	Використання засобів попереднього оброблення для вилучення замаскованих даних
	Засосування хмарних технологій
	Виконання задач аналізу розподіленими системами
	Зменшення обсягу даних або збільшення часу оброблення

Як згадувалось у роботі (П'ятигор, & Бучик, 2025), одним із недоліків збору даних за допомогою API є обмеження на кількість, частоту запитів або обсяг даних, які повертаються в результаті виконання запиту. Тому системи збору мають бути адаптовані до вимог API. Оптимізація запитів полягає у звуженні обсягу даних, які повертаються API, до лише тих, які необхідні для аналізу, за допомогою використання фільтрів, які надаються власниками API. Це вимагає детального розуміння інтерфейсу, який надає розробник, а також зв'язків між даними. Крім того варто знати, за допомогою яких методів інтерфейсу ці дані можуть бути отримані. Якщо API розміщено у відкритому доступі і не є закритим до редагування, розширення функціоналу API можливе і зможе спростити запити у системах збору даних або додати необхідну інформацію до результату запиту до API, яка до цього вимагала додаткових запитів до API.

Оскільки системи вебскрейпінгу залежать від дизайну вебсторінки, з якої збирається інформація, постійна адаптація до нових версій і змін дизайну є обов'язковим напрямом удосконалення таких систем, якщо система збору є постійною і безперервною. Для забезпечення безперервності, необхідно налагодити доступ до даних без залежності до дизайну вебсторінки, наприклад за допомогою тимчасової зміни методу збору на збір через API, або договорами з власниками соціальних мереж задля отримання попередньої інформації про оновлення сервісу, щоб мати час оновити системи збору.

На відміну від організацій, де як дані, так і ієрархія даних є структурованими, дані із соціальних мереж написано у вільному стилі. З огляду на те, що користувачі соціальних мереж можуть писати все що завгодно, якість даних у таких середовищах коливається від цінних до реклами та непотрібу. Тому засоби попереднього оброблення мають бути адаптовані до специфіки даних, не тільки загальних для всіх соціальних мереж, а також до соціальної мережі, дані якої обробляються в системі. Чим більше джерел даних обробляється, тим складніше уніфікувати та структурувати такі дані.

ISSN 2707-1758

Одним із недоліків етапу аналізу даних, є системні вимоги до апаратного забезпечення для виконання цього аналізу. Залежно від обсягу даних, що обробляються, вимоги можуть змінюватися, але часто аналіз великого обсягу даних вимагає наявності високопродуктивної системи з багатоядерним процесором для розв'язання мультипотокових задач і графічного процесора з великим запасом відеопам'яті та підтримкою високопродуктивних обчислень. Це обмежує можливості розроблення та розширення великих систем на власному апаратному забезпеченні. Можливим удосконаленням забезпечення апаратних вимог до етапу аналізу даних є використання хмарних технологій, якщо дані, які обробляються, дозволяють передачу даних на обробку третім сторонам. Зазвичай провайдери хмарних технологій забезпечують платформу для використання або апаратних ресурсів, або навіть повні сервіси для аналізу даних. Це надає можливість швидкого розширення системи для забезпечення обсягів даних, а також можливості гнучкого регулювання ресурсів, які використовують для аналізу. Така гнучкість також може зменшити початкові витрати на апаратне забезпечення вказаних систем.

На додачу до вертикального розширення систем аналізу даних, хмарні технології можуть забезпечувати і горизонтальне розширення. У розробленні систем аналізу варто враховувати, чи можна поділити процес аналізу на підпроцеси, які можна виконувати паралельно і незалежно. Це дозволяє не тільки зменшити час оброблення даних, але також і навантаження на загальну систему. Зазначимо, що часто горизонтальне розширення є більш швидким і легким процесом порівняно з вертикальним, але розподілення задач на підпроцеси має бути враховане у розробленні таких систем.

Якщо розширення апаратних засобів систем аналізу даних не є можливим, то як оптимизацію систем аналізу можна використати перегляд нефунціональних вимог до систем. Зменшення обсягу даних, що обробляються, або збільшення часу на оброблення даних зможе зменшити системні вимоги до таких систем. Якщо при цьому не змешується точність виявлення акаунтів і система не є критичною, то такі зміни до системи можна вважати вдосконаленням системи.

Визначивши можливі напрями вдосконалення, автори представили на рис. 2 архітектуру системи, яка буде використана для подальших досліджень у сфері виявлення автоматизованих аккаунтів. Ця архітектура має на меті поєднати вебскрейпер з API запитами для збору інформації, а також розподілити етап аналізу на незалежні компоненти, які можуть виконуватися паралельно. Використання хмарних технологій на всіх етапах, а також розподілення етапу аналізу на незалежні підпроцеси, які виконуються паралелльно, дозволить ефективно контролювати ресурси, які використовує система виявлення автоматизованих аккаунтів, і зменшити час на аналіз даних.

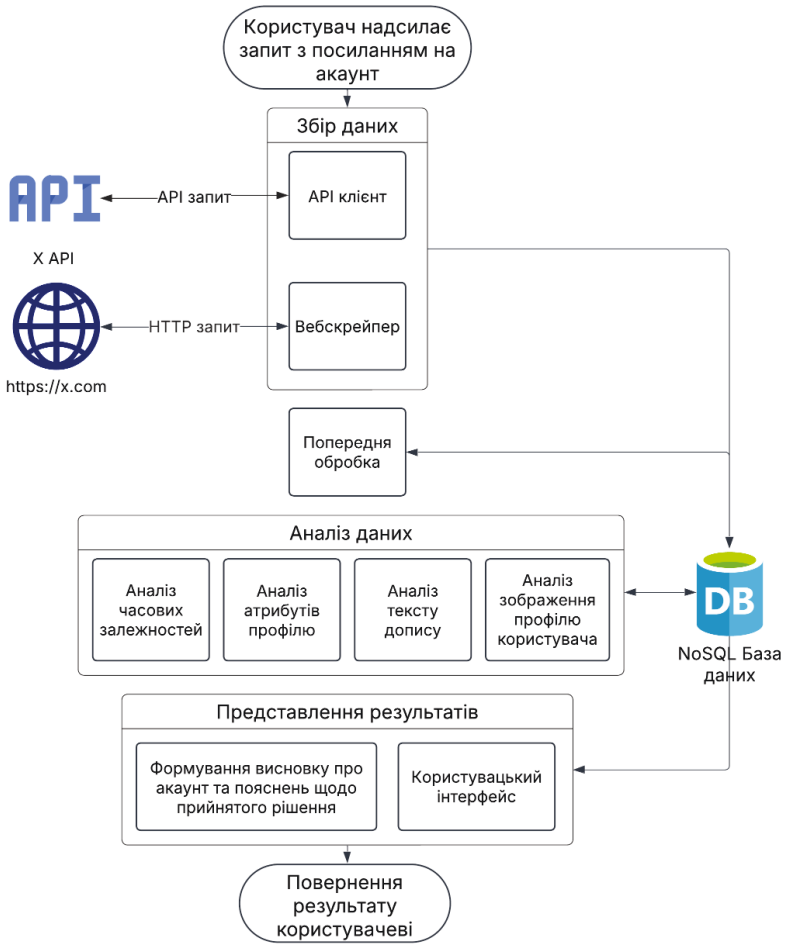

Рис. 2. Запропонована архітектура системи виявлення автоматизованих акаунтів у соціальній мережі "X"

ISSN 2707-1758

Запропонована архітектура системи виявлення автоматизованих акаунтів (ботів) у соціальній мережі "X" складається з кількох основних етапів і компонентів (рис. 2).

На етапі збору даних використовуємо як API клієнт, так і вебскрейпер. Система використовує API соціальної мережі "X" для отримання даних про дописи, користувачів та їхню активність. Користувач надсилає запит із посиланням на акаунт, і API клієнт здійснює API запити для отримання відповідної інформації. На додаток до API, система використовує вебскрейпінг для збору даних безпосередньо з вебсторінок соціальної мережі "X" через HTTP запити. Це необхідно для отримання інформації, яка не надається через публічний API, або для обходу обмежень API на кількість запитів. Вебскрапером також здійснюється очищення даних від непотрібних даних про розмітку та стилі сторінки, скрипти, які присутні на вебсторінці та виділяється корисна інформація. Зібрана інформація зберігається в NoSQL базі даних і контроль передається до етапу попереднього оброблення.

На етапі попереднього оброблення використовують дані, які були зібрані на попередньому етапі і збережені в базі даних. Напряму дані з попереднього етапу не використовують, уся комунікація здійснюється через базу даних. На цьому етапі дані очищають, форматують і зводять до єдиного вигляду для подальшого аналізу. Оброблені дані зберігають в окрему колекцію в базі даних, де кожен запис матиме вигляд профілю користувача з атрибутами, які використовують для аналізу, та списком дописів користувача. Далі контроль передається на етап аналізу даних.

Етап аналізу даних використовує змішаний підхід і включає кілька модулів, які паралельно досліджують різні аспекти поведінки та характеристик користувачів:

- аналіз часових залежностей – досліджується час публікацій, їхня періодичність, інтервали між ними, що може вказувати на автоматизовану діяльність;
- аналіз атрибутів профілю – аналізуються такі характеристики профілю: ім'я користувача, опис, кількість підписників і підписок, дата реєстрації, наявність / відсутність аватара й інша інформація, яка може бути типовою для ботів;
- аналіз тексту допису – зміст, стиль, тематика, наявність певних ключових слів або шаблонних фраз у дописах можуть бути ознаками автоматизованої генерації контенту;
- аналіз зображення профілю користувача – досліджуються характеристики зображення профілю (напр., стандартні зображення, згенеровані штучним інтелектом, або їхня відсутність), що також може бути індикатором бота.

Результат аналізу зберігається у базі даних, запис профілю користувача у базі даних оновлюється з показниками кожного модуля аналізу в числових значеннях. Контроль передається на етап представлення результатів.

Після завершення аналізу даних система формує висновок щодо ймовірності того, що досліджуваний акаунт є автоматизованим, використовуючи отримані значення, що були збережені до бази даних. Формування висновку має бути зваженим, треба враховувати, де певні значення результатів модулів аналізу мають більший чи менший вплив на остаточний результат. Цей висновок також має включати перелік критеріїв, які вказували б на те, що акаунт є автоматизованим, якщо таке рішення було прийняте. Для додаткової прозорості, висновок має також включати відсоток упевненості системи у прийнятому рішенні.

На цьому етапі також присутній компонент користувацького інтерфейсу, який формує відповідь для користувача у форматі вебкомпонента, який буде повернений на сторінку, з якої користувач робив запит.

NoSQL база даних є окремим компонентом системи, який використовують на всіх етапах для збереження проміжних даних, які використовує система. Дані бази даних добре підходять для зберігання неструктурованих або напівструктурованих даних, які часто зустрічаються в соціальних мережах, а також для зберігання великої кількості інформації та оптимізації розподіленого використання даних. База даних має мати дві колекції даних, одна зберігає необроблені дані, які були отримані на етапі збору даних, інша – оброблені записи профілів користувачів, які в ході виконання системи будуть доповнені результатами аналізу.

Як хмарне рішення пропонується використання Microsoft Azure, що має великий спектр сервісів і для побудови системи з використанням лише сервісів даного провайдера, і більш кросплатформні рішення, такі як оркестратор контейнерів Kubernetes. Azure App Service буде використовуватися для розгортання вебінтерфейсу, що отримує початковий запит користувача, відображає результат аналізу та висновки. Як база даних використовуватиметься сервіс Azure Cosmos DB, що є NoSQL базою даних, що має високу швидкість доступу до даних, а також оптимізований для великих обсягів напівструктурованих даних. Azure Functions підходить для реалізації логіки API клієнта та запуску вебскрейпінгових сценаріїв. Хоча правила попереднього оброблення мають бути визначені відповідно до отриманих даних, деякі задачі очищення та зведення даних до єдиного фомату можна покласти на Azure Databricks – платформу для оброблення великих даних, включно з очищенням і зведенням даних до єдиного формату. Оскільки алгоритми аналізу даних будуть власною реалізацію без використання наявних сервісів, для хостингу буде використовуватися Azure Container Instances – сервіс для запуску окремих контейнерів без необхідності в управлінні інфраструктурою, але з підтримкою горизонтального розширення залежно від навантаження. Для загального керування системи буде використовуватися Azure Event Grid – сервіс для маршрутизації подій між компонентами системи, та Azure Logic Apps – платформа автоматизації робочих процесів. Для моніторингу роботи системи, включно зі збором метрик і логів, буде використаний Azure Monitor.

Для розроблення компонентів збору, оброблення, аналізу та формування результатів буде застосована мова програмування Python. Ця мова була вибрана для написання скриптів через наявність великої кількості бібліотек для підтримки аналізу даних, вона забезпечує гнучкість у роботі з різними типами даних і має високу продуктивність завдяки хмарному середовищу. Python добре підходить для оброблення неструктурованих і напівструктурованих даних, які є характерними для соціальних мереж. Скрипти, написані на цій мові, можуть легко обробляти як JSON-дані, отримані через API, так і HTML-дані, зібрані вебскрейпінгом. Вказана мова має великий набір спеціалізованих бібліотек, які роблять аналіз даних ефективним і простим. До основних бібліотек належать такі: Pandas – для роботи з таблицями, оброблення й аналізу даних; NumPy – для обчислень із великими масивами даних. На додачу Python має офіційну підтримку в сервісах Microsoft Azure: Azure SDK для Python для інтеграції з Azure

ISSN 2707-1758

Cosmos DB, Azure Functions, Azure Container Instances та іншими сервісами.

Для розроблення вебсервісу представлення даних буде використаний Flask – мікрофреймворк для веброзробки на мові програмування Python. Він забезпечує простий і гнучкий підхід до створення вебдодатків. Також цей фреймворк повністю підтримується Azure, і його можна використовувати його для створення та розгортання вебдодатків на Azure App Service. Оскільки основною задачею вебзастосунку буде лише отримання запиту користувача та результату з бази даних після виконання аналізу, то цей фреймворк добре підходить для розв'язання цих задач.

Основними характеристиками запропонованої архітектури є:

- гібридний підхід до збору даних – поєднання API запитів і вебскрейпінгу дозволяє отримати повніший обсяг даних, обійшовши можливі обмеження API;
- паралельний аналіз даних – розбиття аналізу на кілька незалежних модулів (часових залежностей, атрибутів профілю, тексту дописів, зображень профілю) дозволяє виконувати їх паралельно, що підвищує ефективність і швидкість роботи системи;
- використання штучного інтелекту: модулі аналізу даних, особливо аналіз тексту та зображень, повинні використовувати методи машинного навчання та штучного інтелекту для виявлення патернів, характерних для ботів.

Представлену систему, її етапи та компоненти можна представити у вигляді множин. Визначимо множину етапів E, яка складається з етапів збору даних, попереднього оброблення, аналізу даних і представлення даних. Множина компонентів K складається з усіх компонентів системи: користувацький інтерфейс, база даних, API клієнт, вебскрапер, засоби попереднього оброблення й обрані засоби аналізу даних. Також етапи системи залежать від результатів попередніх етапів, тому їх також треба включити в модель. Для кожного з етапів визначимо компоненти, які він використовує і побудуємо їх відображення. Цю залежність подано у формулі (1).

$$\begin{aligned} f(\mathrm{E}_1) &= \{\mathrm{K}_1, \mathrm{K}_2, \mathrm{K}_3, \mathrm{K}_4\}, \\ f(\mathrm{E}_2) &= \{\mathrm{K}_2, \mathrm{K}_5, \mathrm{result}(\mathrm{E}_1)\}, \\ f(\mathrm{E}_3) &= \{\mathrm{K}_2, \mathrm{K}_6, \mathrm{K}_7, \mathrm{K}_8, \mathrm{K}_9, \mathrm{result}(\mathrm{E}_2)\}, \\ f(\mathrm{E}_4) &= \{\mathrm{K}_1, \mathrm{K}_2, \mathrm{result}(\mathrm{E}_3)\}, \end{aligned} \quad (1)$$

де E_1 – етап збору даних; E_2 – етап попереднього оброблення; E_3 – етап аналізу даних; E_4 – етап представлення даних; K_1 – компонент користувацького інтерфейсу; K_2 – компонент бази даних; K_3 – компонент API клієнта; K_4 – компонент вебскрейпінгу; K_5 – компонент засобів попереднього оброблення даних; K_6 – компонент засобів аналізу часових залежностей; K_7 – компонент засобів аналізу атрибутів профілю; K_8 – компонент засобів аналізу тексту дописів; K_9 – компонент аналізу зображення профілю користувача; result(E_1) – результат виконання етапу збору даних; result(E_2) – результат виконання етапу попереднього оброблення; result (E_3) – результат виконання етапу аналізу даних.

Отже, загальну систему можна описати як об'єднання всіх етапів, відповідних їм компонентів і результатів виконання попередніх етапів. Це об'єднання представлено у формулі (2).

$$S = \bigcup_{\mathrm{E}_i \in \mathrm{E}} \left(\mathrm{E}_i \times f(\mathrm{E}_i)\right). \quad (2)$$

Таке представлення допомагає чітко поділити етапи системи та компоненти, від яких залежить кожен з етапів. Це робить модель простою для розуміння, аналізу й оброблення, дозволяє оптимізувати структуру системи та передбачати наслідки змін. Описана модель полегшує розширення, модифікацію та автоматизацію.

Дискусія і висновки

За результатами дослідження встановлено, що архітектура систем виявлення ботів включає кілька етапів: збір даних, їхнє попереднє оброблення, аналіз характеристик користувачів і класифікацію облікових записів за допомогою алгоритмів машинного навчання. Додатково в системах можливими компонентами можуть бути засоби навчання моделі, бази даних і засоби представлення даних користувачеві. Доповнено перелік недоліків на кожному з етапів, порівняно з преставленим у попередній роботі авторів (П'ятигор, & Бучик, 2025), та наведено основні напрями вдосконалення.

З урахуванням виявлених недоліків запропоновано вдосконалену архітектуру системи виявлення автоматизованих акаунтів, що поєднує вебскрейпінг та API запити для безперервного збору даних і використовує хмарні технології для гнучкого управління ресурсами, які використовує система, і дозволяє виконувати аналіз даних паралельно. Система також була представлена у вигляді множин, що дозволяє візуалізувати залежності компонентів від етапів, що спрощує планування та розроблення вказаної системи. Представлена система у подальшому буде використана для виконання досліджень у галузі виявлення автоматизованих акаунтів та імплементована для оцінювння її ефективності у реальних умовах.

Внесок авторів: Віталій П'ятигор – концептуалізація, методологія, написання (оригінальна чернетка); Сергій Бучик – написання (перегляд і редагування), визначення методів дослідження, опис запропонованої архітектури.

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

П'ятигор, В., & Бучик, С. (2025). Проблеми виявлення автоматизованих аккаунтів в соціальних мережах. У С. Ю. Даков, & О. С. Торошанко (Ред.), *Проблеми кібербезпеки інформаційно-телекомунікаційних систем: зб. матеріалів доповідей та тез* (с. 124–126). Київський національний університет імені Тараса Шевченка.

Almerekhi, H., & Elsayed, T. (2015). Detecting Automatically-Generated Arabic Tweets. *Lecture Notes in Computer Science,* 9460, 122–134. https://doi.org/10.1007/978-3-319-28940-3_10

Bessi, A., & Ferrara, E. (2016). *Social bots distort the 2016 U.S. Presidential election online discussion.* First Monday. https://doi.org/10.5210/fm.v21i11.7090

Boshmaf, Y., Logothetis, D., Siganos, G., Lería, J., Lorenzo, J., Ripeanu, M., Beznosov, K., & Halawa, H. (2016). Íntegro: Leveraging victim prediction for robust fake account detection in large scale OSNs. *Computers & Security,* 61, 142–168. https://doi.org/10.1016/j.cose.2016.05.005

Dixon, S. J. (2024, July 10).*Biggest social media platforms by users 2024.* Statista. https://www.statista.com/statistics/272014/global-social-/networks-ranked-by-number-of-users/

Dixon, S. J. (2025, February 6). *Facebook fake account deletion per quarter 2024*. Statista. https://www.statista.com/statistics/1013474/facebook-fake-account-removal-quarter/

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM,* 59(7), 96–104. https://doi.org/10.1145/2818717

Fredhaim, R., & Stolze, M. (2022, May 24). *Robotrolling 2022 (1).* NATO Strategic Communications Centre of Excellence. https://stratcomcoe.org/publications/robotrolling-20221/243

Guy, S., Ratzki-Leewing, A., Bahati, R., & Gwadry-Sridhar, F. (2012). Social Media: A Systematic Review to Understand the Evidence and Application in Infodemiology. *Lecture Notes of the Institute for*

ISSN 2707-1758

Computer Sciences, Social-Informatics and Telecommunications Engineering, 91, 1–8. https://doi.org/10.1007/978-3-642-29262-0_1

References

Almerekhi, H., & Elsayed, T. (2015). Detecting Automatically-Generated Arabic Tweets. *Lecture Notes in Computer Science,* 9460, 122–134. https://doi.org/10.1007/978-3-319-28940-3_10

Bessi, A., & Ferrara, E. (2016). *Social bots distort the 2016 U.S. Presidential election online discussion.* First Monday. https://doi.org/10.5210/fm.v21i11.7090

Boshmaf, Y., Logothetis, D., Siganos, G., Lería, J., Lorenzo, J., Ripeanu, M., Beznosov, K., & Halawa, H. (2016). Íntegro: Leveraging victim prediction for robust fake account detection in large scale OSNs. *Computers & Security,* 61, 142–168. https://doi.org/10.1016/j.cose.2016.05.005

Dixon, S. J. (2024, July 10).*Biggest social media platforms by users 2024.* Statista. https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

Dixon, S. J. (2025, February 6). *Facebook fake account deletion per quarter 2024.* Statista. https://www.statista.com/statistics/1013474/facebook-fake-account-removal-quarter/

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM,* 59(7), 96–104. https://doi.org/10.1145/2818717

Fredhaim, R., & Stolze, M. (2022, May 24). *Robotrolling 2022 (1).* NATO Strategic Communications Centre of Excellence. https://stratcomcoe.org/publications/robotrolling-20221/243

Guy, S., Ratzki-Leewing, A., Bahati, R., & Gwadry-Sridhar, F. (2012). Social Media: A Systematic Review to Understand the Evidence and Application in Infodemiology. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering,* 91, 1–8. https://doi.org/10.1007/978-3-642-29262-0_1

Pyatigor, V., & Buchyk, S. (2025). Problems of detecting automated accounts in social networks. In S. Yu. Dakov, & O. S. Toroshanko (Eds.), *Problems of cybersecurity of information and telecommunication systems: collection of materials of reports and abstracts* (pp. 124–126). Taras Shevchenko National University of Kyiv [in Ukrainian].

Отримано редакцією журналу / Received: 21.05.2025
Прорецензовано / Revised:29.05.2025
Схвалено до друку / Accepted:22.06.2025

Serhii BUCHYK, DSc (Engin.), Prof.
ORCID ID: 0000-0003-0892-3494
e-mail: buchyk@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Vitalii PIATYHOR, PhD Student
ORCID ID: 0000-0002-7621-1299
e-mail: vp5gor@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

SOCIAL MEDIA AUTOMATED ACCOUNT (BOT) DETECTION SYSTEM ARCHITECTURE

Background. ***Social networks are one of the main sources of information. However, as trust in them grows, so does the use of automated accounts to spread disinformation and influence public opinion, which is an effective method of manipulation. Automated account detection systems are used to detect such accounts. The purpose of the article is to analyze the architecture of automated account (bot) detection systems in social networks and to identify the main directions for improving such architecture, taking into account the identified shortcomings.***

Methods. ***Methods of analysis, systematization, and generalization were used to identify shortcomings and areas for system improvement, and a modeling method was used to develop generalized and improved system architecture.***

Results. ***An improved architecture of a system for detecting automated accounts (bots) in the social network x.com is proposed, which uses a mixed model of data analysis using artificial intelligence and combines web scraping and API queries for data collection.***

Conclusions. ***The general architecture of automated account (bot) detection systems in social networks was analyzed and the main shortcomings and areas for improvement were identified, among which the following can be distinguished: restrictions on API requests from social network owners, unstructured and free style of posts in social networks, and constant changes in post generation algorithms by automated accounts. An improved architecture of the automated account (bot) detection system was proposed, which combines web scraping and API requests for data collection and allows parallel execution of social network data analysis.***

Keywords: ***computer system, data collection, data analysis, machine learning.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758