УДК 007:004.9
DOI: https://doi.org/10.17721/1812-5409.2025/2.29

Михайло МАХНО[1], канд. техн. наук, доц.
ORCID ID: 0000-0001-5045-1705
e-mail: mykhailo.makhno@knu.ua

Олексій ФЕДОРУС[1], канд. техн. наук, асист.
ORCID ID: 0009-0006-5496-4544
e-mail: oleksii.fedorus@knu.ua

Олександр БОРИСЕНКО[1], асп.
ORCID ID: 0009-0009-7852-3227
e-mail: borysenko@knu.ua

Максим ВЕРЕМЧУК[2], асп.
ORCID ID: 0009-0003-9787-2787
e-mail: mrveremc@uwaterloo.ca

[1]Київський національний університет імені Тараса Шевченка, Київ, Україна
[2]Університет Ватерлоо, Ватерлоо, Онтаріо, Канада

АРХІТЕКТУРА СИСТЕМИ ВИЯВЛЕННЯ БОТІВ У СОЦІАЛЬНИХ МЕРЕЖАХ

Сучасні інформаційні системи потребують ефективної архітектури для забезпечення високої продуктивності, масштабованості та надійності. У представленій статті розглянуто підхід до проєктування архітектури системи, що враховує новітні технологічні рішення та методи оптимізації роботи з великими масивами даних. Запропоновано власну архітектуру системи виявлення ботів, засновану на мікросервісному підході та сучасних методах обробки даних. На відміну від наявних рішень, система акцентує увагу не на розробленні радикально нового методу класифікації, а на ефективному поєднанні вже відомих підходів у єдиній архітектурі.

Розвиток інформаційних технологій вимагає створення ефективних архітектурних рішень, що забезпечують високу продуктивність і надійність програмних систем. Зі зростанням обсягів даних і вимог до швидкодії, класичні підходи до побудови архітектури потребують удосконалення. Дослідження в цій галузі є важливими для розробників й архітекторів програмного забезпечення. Мета пропонованої роботи – розроблення концепції архітектури, що відповідає сучасним вимогам продуктивності, масштабованості та безпеки. Основними завданнями є аналіз наявних підходів, ідентифікація їхніх переваг і недоліків, а також вибудовування ефективної архітектури, що мінімізує витрати ресурсів і підвищує швидкість обробки даних.

У дослідженні застосовано методи аналізу архітектурних рішень, моделювання системи, тестування продуктивності та порівняльний аналіз ефективності різних підходів. Для реалізації архітектури використано сучасні технології, зокрема мікросервісний підхід, контейнеризація та розподілені обчислення.

Запропонована архітектура дає змогу підвищити продуктивність системи шляхом оптимізації обробки запитів та розподілу навантаження між сервісами. Використання контейнеризації та оркестрації допомагає досягти гнучкості у масштабуванні та поліпшує стабільність роботи системи. Аналіз продуктивності показав зменшення затримок обробки запитів й ефективне використання ресурсів серверів.

Розроблена архітектура підтвердила свою ефективність у тестових середовищах і може бути застосована для високонавантажених систем. Перспективи подальших досліджень передбачають інтеграцію штучного інтелекту для автоматичного масштабування й оптимізації роботи сервісів, а також дослідження впливу різних підходів до кешування даних на продуктивність системи.

Ключові слова: *архітектура систем, бот, API, NLP.*

Класифікація відповідно до AMS 2020: 68P05, 68P10, 68P15, 68P20, 68W32.

Вступ

Зі зростанням популярності соціальних мереж активність бот-акаунтів значно збільшилася. Вони використовуються для маніпулювання громадською думкою, поширення дезінформації та фішингу. Згідно з дослідженнями, значна частина інтернет-трафіку генерується автоматизованими системами, проте шкідливі боти займають 32 % від усього трафіку, згідно з (Imperva, 2024), що містить загрозу інформаційній безпеці.

Існування ботів у соціальних мережах створює низку проблем, пов'язаних із поширенням дезінформації та маніпуляцією громадською думкою. Боти можуть імітувати поведінку реальних користувачів, що ускладнює їхнє виявлення та сприяє неконтрольованому розповсюдженню неправдивої інформації. З розвитком генеративних моделей, створення фейкових новин стало доступнішим, що підсилює проблему дезінформації в соціальних мережах.

Виявлення ботів є складним завданням, що потребує використання різних методів аналізу. Традиційно для цього застосовували аналіз метаданих профілю та лінгвістичні особливості повідомлень. Однак дослідження показують, що аналіз структурних особливостей мережі, зокрема використання структурних убудовувань, може бути ефективнішим у виявленні ботів. Це підтверджує гіпотезу, що локальна соціальна мережа, сформована навколо бот-акаунтів, містить цінну інформацію для їхньої ідентифікації.

Крім того, сучасні системи виявлення ботів стикаються з проблемами масштабованості й узагальненості. Більшість із них базуються на методах машинного навчання з використанням вручну позначених даних, що обмежує їхню здатність виявляти скоординовані дії бот-мереж. Існує потреба в розробленні більш універсальних підходів, що враховують різні аспекти діяльності ботів, включно з аналізом зображень та міжплатформною активністю. Проблему виявлення скоординованих бот-мереж також розглянуто в роботах, присвячених BotNet-детекції (Devle et al., 2021).

ISSN 2518-1378 (Print), ISSN 2708-6038 (Online)

Отже, для ефективної боротьби з ботами та пов'язаними з ними загрозами необхідно поєднувати різні методи аналізу й розробляти нові підходи, що враховують сучасні тенденції в розвитку технологій і поведінки ботів у соціальних мережах. Існуючі методи виявлення ботів передбачають:

- *аналіз поведінкових патернів* (частота публікацій, часові інтервали, аномалії);
- *обробку природної мови* (NLP) для аналізу контенту та визначення аномалій у стилістиці;
- *графові методи* (аналіз мережевої структури зв'язків);
- *гібридні підходи*, що комбінують кілька підходів для підвищення точності (Sen et al., 2018).

Останні дослідження (Pote, 2024) та (Antispoofing Wiki, 2024) показують, що методи на основі графових ембедингів забезпечують вищу точність порівняно з традиційними підходами, що ґрунтуються на NLP та мета-даних акаунтів.

Головна мета дослідження – створення масштабованої системи виявлення ботів у соціальних мережах, що здатна працювати в режимі реального часу, забезпечуючи високу точність класифікації. Важливими аспектами системи є:

- підтримка потокової обробки великих обсягів даних,
- використання графових ембедингів для підвищення точності детекції,
- інтеграція із сучасними базами даних і технологіями контейнеризації,
- масштабування системи при збільшенні навантаження.

Перед початком розроблення системи виявлення ботів необхідно визначити основні загрози, типи ботів і методи, які вони використовують для маскування своєї активності. Боти можуть мати різну мету, зокрема:

- **Маніпулювання громадською думкою** – використання ботів для поширення дезінформації, політичної пропаганди або соціального впливу через масове створення та поширення контенту.
- **Фінансове шахрайство** – передбачає накрутку лайків, переглядів, підписників, а також шахрайські схеми із криптовалютами та фінансовими операціями.
- **Кіберзагрози** – використання ботів для фішингових атак, спаму та розповсюдження шкідливого ПЗ.

Сучасні боти дедалі частіше використовують складні техніки маскування своєї активності, що ускладнює їхнє виявлення:

- **Імітація людської поведінки** – нерівномірні інтервали між публікаціями, взаємодія з іншими користувачами, перемежування автоматичних та ручних дій.
- **Динамічна зміна IP-адрес** – використання VPN, проксі-серверів або ботнет-інфраструктури для приховування справжнього місцезнаходження бота.
- **Персоналізовані облікові записи** – використання реальних або зламаних акаунтів, що мають історію активності та друзів.
- **Автоматизація через ШІ** – використання моделей GPT та аналогічних нейромереж для створення правдоподібного контенту, що важко відрізнити від людського. Також варто враховувати можливість ботів генерувати візуальний контент самостійно, наприклад, за допомогою таких моделей, як Stable Diffusion.

Щоб ефективно виявляти ботів, необхідно використовувати кілька взаємопов'язаних підходів, кожен з яких дає змогу ідентифікувати різні характеристики ботів і їхньої активності. Застосування комплексного аналізу допомагає суттєво підвищити точність класифікації ботів. Таким чином ми отримуємо ще одну вимогу до системи – можливість застосування одночасно кількох підходів для аналізу поведінки. Аналітичний сервіс матиме кілька модулів, зокрема: модуль аналізу поведінкових патернів, аналізу текстів та аналізу мережевих зв'язків.

Перший модуль відповідає за аналіз поведінкових патернів користувачів, зокрема часових рядів їхньої активності та виявлення аномалій у публікаціях.

Другий модуль здійснює обробку контенту акаунтів для виявлення ботів на основі мовних характеристик їхніх дописів. Основні напрямки аналізу:

- *лексичний і синтаксичний* – застосування методів обробки природної мови (NLP) для аналізу структури речень, виявлення надмірного використання шаблонних текстів і повторюваних ключових слів (Tzoumanekas et al., 2024);
- *семантичний* – визначення смислових зв'язків між словами та текстами, що допомагає розпізнавати автоматично згенерований контент. Використовуються моделі Word2Vec, BERT, RoBERTa;
- *стилістичний* – оцінювання характерних особливостей текстів, таких як довжина речень, складність граматики, частота використання певних слів. Боти часто демонструють нестандартну стилістику;
- *тональність й емоційний аналіз* – оцінювання емоційної спрямованості текстів (позитивна, нейтральна, негативна) та виявлення шаблонної емоційної поведінки ботів.

Гібридні глибокі нейронні архітектури, що поєднують рекурентні мережі та векторні подання слів (напр., BiGRU-LSTM із використанням GloVe), демонструють високу ефективність у задачах виявлення ботів у соціальних мережах (Ellaky et al., 2024).

Реалізація другого модуля передбачає використання бібліотек NLP, таких як spaCy, Hugging Face Transformers та алгоритмів машинного навчання (LSTM, Transformer-based models).

Третій модуль дає змогу аналізувати структуру соціальних зв'язків користувачів для виявлення бот-мереж. Основні методи передбачають:

- *графовий аналіз* – використання алгоритмів Node2Vec, Struc2Vec для виявлення схожих патернів взаємодії серед акаунтів;
- *ідентифікація координаційних груп* – дослідження поведінки груп акаунтів, що синхронно взаємодіють із певним контентом (масові ретвіти, спільні підписки, одномоментні публікації);
- *моделювання соціальних графів* – аналіз зв'язків між акаунтами з використанням таких метрик, як центральність, кластеризація, коефіцієнт спільних сусідів;
- *виявлення аномальних зв'язків* – визначення акаунтів з аномально високою взаємодією з іншими ботами або великою кількістю однотипних зв'язків.

1. Опис архітектури системи

Архітектура системи виявлення ботів реалізована у вигляді набору взаємодіючих вебсервісів, що дає змогу забезпечити масштабованість, гнучкість і незалежне оновлення компонентів. В основі архітектури лежить кілька ключових сервісів, кожен з яких виконує спеціалізовані функції (рис. 1).

Система починається із сервісу збору даних, що відповідальний за отримання інформації з різних соціальних мереж, таких як Twitter, Telegram та Facebook. Для цього використовують офіційні API-платформ, а також методи вебскрапінгу. Цей сервіс також застосовує попередню фільтрацію, щоб зменшити обсяг оброблюваної інформації.

Далі інформація проходить через сервіс ETL-обробки, що виконує очищення, нормалізацію та трансформацію даних для їх подальшого аналізу. Це передбачає видалення дублікатів (отримання одного повідомлення з різних джерел), стандартизацію форматів й агрегацію даних, що дає змогу підготувати їх для подальшого аналізу.

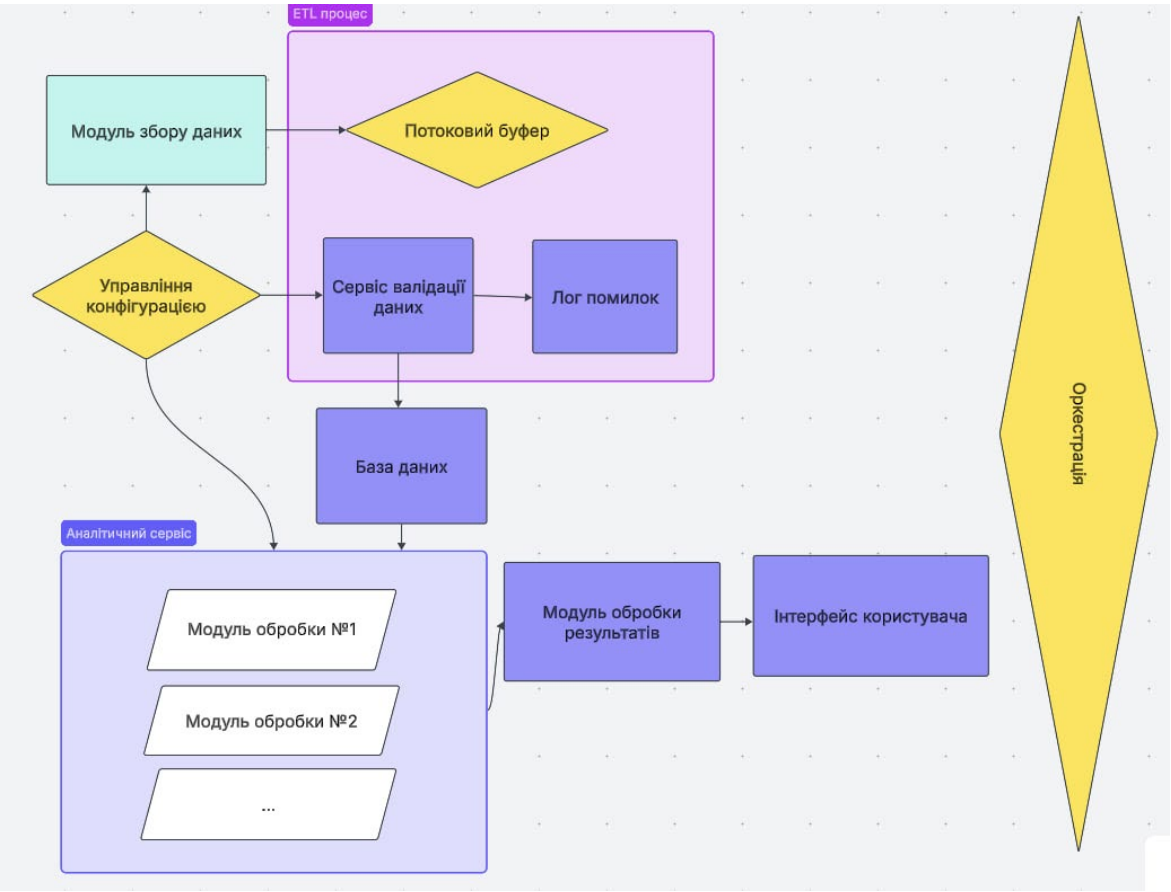

Рис. 1. Загальна схема архітектури системи виявлення ботів у соціальних мережах

Центральним елементом є аналітичний сервіс, який застосовує алгоритми машинного навчання для класифікації ботів. Він складається з декількох основних підмодулів: аналізу поведінкових патернів, аналізу тексту та аналізу мережевих зв'язків. Перший модуль аналізує часові ряди активності акаунтів, виявляючи аномалії та повторювані шаблони, другий застосовує NLP-методи для оцінки змісту та стилістики повідомлень, тоді як третій використовує графові алгоритми для визначення підозрілих зв'язків між акаунтами. Ще в одному з модулів аналізуємо, які особливості користувачів можна використовувати для виявлення ботів. Ці особливості розділяємо на чотири групи:

- **NLP** та **P** – дані профілю й ознаки, отримані з твітів за допомогою обробки природної мови.
- **GF** – прості графові характеристики, такі як центральність вузлів.
- **EMB** – ознаки, отримані із соцмережі за допомогою алгоритмів убудовування (поділяються на класичні та структурні).

Метою цього модуля є не створення нового класифікатора ботів, а порівняння ефективності різних наборів ознак, особливо структурних EMB. Ми показуємо, що боти часто формують локальні соціальні структури, які можна виявити саме за допомогою таких ознак.

Вибір мікросервісної архітектури пояснюють необхідністю забезпечити масштабованість системи та можливість незалежного оновлення окремих компонентів без зупинки роботи всієї інфраструктури. На відміну від монолітних систем, мікросервіси дають змогу швидко адаптуватися до зростання навантаження та спрощують інтеграцію з новими сервісами. Практичні підходи до розгортання масштабованих мікросервісних систем із використанням контейнеризації та оркестрації за допомогою Docker і Kubernetes детально описані в роботі (Lin, 2019).

Використання REST API зумовлене потребою уніфікованої взаємодії між компонентами системи та зовнішніми застосунками. Такий підхід забезпечує простоту інтеграції та високу сумісність із наявними інструментами аналітики.

Поєднання PostgreSQL і DuckDB обрано для оптимізації роботи з різними типами даних: PostgreSQL гарантує надійність і транзакційну цілісність, тоді як DuckDB забезпечує ефективну обробку великих аналітичних запитів.

Elasticsearch інтегровано як інструмент для швидкого пошуку й аналітики в режимі реального часу, що є критично важливим у роботі з високошвидкісними потоками даних.

Для аналітичних модулів використано алгоритми графових убудовувань (Node2Vec, Struc2Vec та ін.), оскільки вони дають змогу автоматично знаходити приховані соціальні структури та взаємозв'язки, що є складним завданням для класичних підходів на основі лише текстового аналізу чи простих статистичних характеристик. Отже, обрана комбінація технологій дає змогу поєднати гнучкість у масштабуванні, надійність зберігання та високу точність виявлення ботів.

Ключові результати:

- Усі чотири набори ознак (**NLP, P, GF** та **EMB**) мають предиктивну силу у виявленні ботів.
- Поєднання класичних і структурних ознак підвищує точність моделей.
- Алгоритми EMB можуть автоматично знаходити корисні особливості, які складно створити вручну.
- Структурні EMB мають вищу точність, ніж класичні методи вбудовування.
- Підвищення розмірності EMB незначно впливає на результат, адже навіть низькорозмірні вбудовування ефективні.
- EMB-моделі демонструють стійкість до шуму в соціальній мережі.

Запропонована система базується на принципах Каппа-архітектури, що передбачає потокову обробку даних у реальному часі та використання єдиної обчислювальної моделі для всіх типів обробки. На відміну від Lambda-архітектури, де існує поділ на пакетний і потоковий рівні, Каппа-архітектура спрощує підтримку системи, оскільки усуває дублювання логіки обробки та надає можливість уникнути повторних обчислень. Це особливо важливо для систем виявлення ботів, що працюють із великими потоками даних із соціальних мереж, де швидкість реакції є критичною. Потокова обробка даних у реальному часі в подібних архітектурах зазвичай реалізується з використанням брокерів повідомлень, таких як Apache Kafka, що підтверджується сучасними промисловими практиками (Statsig, 2024). Використання Каппа-архітектури забезпечує масштабованість, спрощує інтеграцію нових джерел даних і підвищує ефективність оброблення без необхідності додаткової пакетної обробки.

Список модулів, що можуть бути інтегрованими в сервіс, детально описані в (Antispoofing Wiki, 2024). Аналітичний сервіс інтегрується з іншими частинами системи через REST API, що забезпечує його масштабованість і легку адаптацію до нових загроз.

Результати аналізу передаються до сервісу обробки інформації та сховища даних, де вони зберігаються у структурованому форматі для подальшого оброблення. Використання реляційних баз даних, таких як PostgreSQL та DuckDB (інтеграція реалізується за допомогою окремого модулю pg_duckdb), забезпечує ефективне зберігання та швидкий доступ до даних, тоді як Elasticsearch дає змогу виконувати пошук й аналітику в реальному часі.

Додатково система використовує MinIO (відкрите S3-сумісне сховище даних) для збереження великих обсягів історичних даних.

Підсумковим результатом аналізу є оцінка ризику, що визначає ймовірність того, що користувач є ботом. Якщо значення оцінки перевищує 95 %, система автоматично відносить акаунт до категорії ботів. Якщо ж оцінка перебуває в діапазоні 70–95 %, то рішення передається людині (оператору системи) для подальшого розгляду. Аналогічно до сучасних систем автопілотів у таксі, коли алгоритм може самостійно керувати у стандартних ситуаціях, а у складних випадках підключається оператор, тут система прагне мінімізувати кількість випадків, коли адміністратору необхідно втручатися вручну.

Для взаємодії з користувачами розроблено UI-сервіс, що надає графічний інтерфейс для адміністраторів та аналітиків. Він розроблений на основі сучасного frontend-фреймворку і дозволяє керувати налаштуваннями системи, переглядати звіти й моніторити активність ботів у соціальних мережах.

Інтеграція UI з backend відбувається через REST API. Завдяки поділу на окремі вебсервіси, система підтримує високу продуктивність, масштабованість і можливість безперервного оновлення без зупинення роботи.

Дискусія і висновки

Отримані результати свідчать, що поєднання різних груп ознак (NLP, профільні характеристики, прості графові метрики та структурні вбудовування) забезпечує вищу точність порівняно з використанням окремих методів, що узгоджується з висновками інших дослідників (Dehghan et al., 2023; Tzoumanekas et al., 2024). На відміну від традиційних рішень, які акцентують увагу на аналізі текстів або метаданих (Sen et al., 2018), запропонована архітектура інтегрує кілька підходів одночасно та забезпечує роботу в режимі реального часу. Важливим аспектом є те, що архітектура реалізує принципи Каппа-архітектури, де всі дані обробляються у потоковому режимі, а результати зберігаються для подальшого аналізу. Це дає змогу уникнути дублювання обчислень і спрощує підтримку системи порівняно з Lambda-архітектурою. Подальші дослідження можуть бути спрямовані на оптимізацію апаратних вимог, що допоможе зменшити залежність від високопродуктивного обладнання. Перспективним напрямком є удосконалення моделей машинного навчання, зокрема впровадження сучасних архітектур нейронних мереж для точнішого аналізу тексту й поведінкових даних.

Важливим завданням також є розширення можливостей модулів збору даних для роботи з новими платформами та впровадження методів динамічного оновлення моделей, що забезпечить їхню адаптацію до нових загроз. Реалізація таких удосконалень допоможе підвищити точність виявлення ботів, розширити функціональні можливості системи й інтегрувати її з іншими технологіями та платформами.

Виявлення ботів у соціальних мережах є складним завданням, що вимагає комплексного підходу. Запропонована система на основі Каппа-архітектури забезпечує високу продуктивність, масштабованість і гнучкість.

Подальший розвиток цієї системи дасть змогу ефективніше протидіяти сучасним ботам, підтримуючи безпеку та якість взаємодії у соціальних мережах. Використання передових методів машинного навчання та потокової обробки даних відкриває нові можливості для виявлення ботів навіть у складних умовах.

Внесок авторів: Олександр Борисенко – написання, оригінальна чернетка; Михайло Махно – концептуалізація, методологія; Олексій Федорус – модифікація підходів і редагування; Максим Веремчук – модифікація підходів і редагування.

Джерела фінансування. Публікацію частково профінансовано Київським національним університетом імені Тараса Шевченка.

Список використаних джерел
Antispoofing Wiki. (2024). *Detecting Bot-Generated Fake News in Social Media*. Antispoofing Wiki. https://antispoofing.org/detecting-bot-generated-fake-news-in-social-media/
Dehghan, A., Siuta, K., Skorupka, A., Dubey, A., Betlen, A., Miller, D., Xu, W., Kamiński, B., & Prałat, P. (2023). *Detecting bots in social-networks using node and structural embeddings*. Journal of Big Data, 10(119). https://doi.org/10.1186/s40537-023-00796-3
Devle, A. C., Jose, J. A., Saraswathula, A. S., Mehta, S., Srivastava, S., Kona, S., & Daggumalli, S. (2021). *BotNet detection on social media*. arXiv. https://doi.org/10.48550/arXiv.2110.05661
Ellaky Z., Benabbou F., Matrane Y., & Qaqa S. (2024). *A Hybrid Deep Learning Architecture for Social Media Bots Detection Based on Bigru-LSTM and Glove Word Embedding*. IEEE Access, 12, 100278-100294. https://doi.org/10.1109/ACCESS.2024.3430859
Imperva. (2024). *Bad Bot Report.* Imperva. https://www.imperva.com/resources/resource-library/reports/2024-bad-bot-report/
Lin J. (2019). *Deploying a scalable web application with Docker and Kubernetes*. Medium. https://medium.com/better-practices/deploying-a-scalable-web-application-with-docker-and-kubernetes-a5000a06c4e9
Pote, M. (2024). *Computational Propaganda Theory and Bot Detection System: Critical Literature Review*. arXiv. https://doi.org/10.48550/arXiv.2404.05240
Sen, I., Aggarwal, A., Mian, S., & Singh, S. (2018). Worth its Weight in Likes: Towards Detecting Fake Likes on Instagram. In Proceedings of the 10th ACM Conference on Web Science, *WebSci 2018: The Reality of Social Media* (pp. 205–209). Association for Computing Machinery. http://dx.doi.org/10.1145/3201064.3201105
Statsig. (2024). *Real-time data processing with Apache Kafka*. Satsig. https://www.statsig.com/perspectives/real-time-data-processing-with-apache-kafka
Tzoumanekas, G., Chatzianastasis, M., Ilias, L., Kiokes, G., Psarras, J., & Askounis, D. (2024). *A Graph Neural Architecture Search Approach for Identifying Bots in Social Media*. Front. Artif. Intell. https://doi.org/10.48550/arXiv.2411.16285

References
Antispoofing Wiki. (2024). *Detecting Bot-Generated Fake News in Social Media*. Antispoofing Wiki. https://antispoofing.org/detecting-bot-generated-fake-news-in-social-media/
Dehghan, A., Siuta, K., Skorupka, A., Dubey, A., Betlen, A., Miller, D., Xu, W., Kamiński, B., & Prałat, P. (2023). *Detecting bots in social-networks using node and structural embeddings*. Journal of Big Data, 10(119). https://doi.org/10.1186/s40537-023-00796-3
Devle, A. C., Jose, J. A., Saraswathula, A. S., Mehta, S., Srivastava, S., Kona, S., & Daggumalli, S. (2021). *BotNet detection on social media*. arXiv. https://doi.org/10.48550/arXiv.2110.05661
Ellaky Z., Benabbou F., Matrane Y., & Qaqa S. (2024). *A Hybrid Deep Learning Architecture for Social Media Bots Detection Based on Bigru-LSTM and Glove Word Embedding*. IEEE Access, 12, 100278-100294. https://doi.org/10.1109/ACCESS.2024.3430859
Imperva. (2024). *Bad Bot Report.* Imperva. https://www.imperva.com/resources/resource-library/reports/2024-bad-bot-report/
Lin J. (2019). *Deploying a scalable web application with Docker and Kubernetes*. Medium. https://medium.com/better-practices/deploying-a-scalable-web-application-with-docker-and-kubernetes-a5000a06c4e9
Pote, M. (2024). *Computational Propaganda Theory and Bot Detection System: Critical Literature Review*. arXiv. https://doi.org/10.48550/arXiv.2404.05240
Sen, I., Aggarwal, A., Mian, S., & Singh, S. (2018). Worth its Weight in Likes: Towards Detecting Fake Likes on Instagram. In Proceedings of the 10th ACM Conference on Web Science, *WebSci 2018: The Reality of Social Media* (pp. 205–209). Association for Computing Machinery. http://dx.doi.org/10.1145/3201064.3201105
Statsig. (2024). *Real-time data processing with Apache Kafka*. Satsig. https://www.statsig.com/perspectives/real-time-data-processing-with-apache-kafka
Tzoumanekas, G., Chatzianastasis, M., Ilias, L., Kiokes, G., Psarras, J., & Askounis, D. (2024). *A Graph Neural Architecture Search Approach for Identifying Bots in Social Media*. Front. Artif. Intell. https://doi.org/10.48550/arXiv.2411.16285

Отримано редакцією журналу / Received: 31.03.25
Прорецензовано / Revised: 07.10.25
Схвалено до друку / Accepted: 10.10.25

Mykhailo MAKHNO[1], PhD (Engin.) Assoc. Prof.
ORCID ID: 0000-0001-5045-1705
e-mail: mykhailo.makhno@knu.ua

Oleksii FEDORUS[1], PhD (Engin.), Assist.
ORCID ID: 0009-0006-5496-4544
e-mail: oleksii.fedorus@knu.ua

Oleksandr BORYSENKO[1], PhD Student
ORCID ID: 0009-0009-7852-3227
e-mail: borysenko@knu.ua

Maksym VEREMCHUK[2], PhD Student
ORCID ID: 0009-0003-9787-2787
e-mail: mrveremc@uwaterloo.ca

[1]Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
[2]University of Waterloo, Waterloo, Ontario, Canada

ARCHITECTURE OF A SOCIAL MEDIA BOT DETECTION SYSTEM

Modern information systems require efficient architectures to ensure high performance, scalability, and reliability. This article presents an approach to system architecture design that incorporates the latest technological solutions and methods for optimizing the processing of large data sets. The paper proposes an original architecture of a bot detection system based on the microservices paradigm and modern data processing techniques. Unlike existing solutions, the proposed system does not aim to develop a radically new classification method but focuses on the effective integration of well-established approaches within a unified architecture.

The advancement of information technologies requires the development of architectural solutions that guarantee high performance and reliability of software systems. With the increasing volume of data and growing demands for processing speed, traditional architectural approaches require refinement. Research in this field is important for software developers and system architects.

The aim of this study is to develop an architectural concept that meets modern requirements for performance, scalability, and security. The main objectives include analyzing existing approaches, identifying their advantages and drawbacks, and designing an efficient architecture that minimizes resource consumption and increases data processing speed.

The study employed methods of architectural analysis, system modeling, performance testing, and comparative evaluation of different approaches. For the implementation of the architecture, modern technologies were used, including the microservices paradigm, containerization, and distributed computing.

The proposed architecture improves system performance by optimizing request processing and distributing workloads across services. The use of containerization and orchestration enables flexible scalability and enhances system stability. Performance analysis has shown reduced request processing latency and efficient utilization of server resources.

The developed architecture has proven its effectiveness in test environments and can be applied to high-load systems. Future research directions include the integration of artificial intelligence for automatic scaling and service optimization, as well as studying the impact of different caching strategies on overall system performance.

K e y w o r d s : ***system architecture, bot, API, NLP.***

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.