УДК 004
DOI https://doi.org/10.32689/maup.it.2024.1.8

Леся ЛЮШЕНКО
кандидат технічних наук, доцент кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», lyushenkol@gmail.com
***ORCID:** 0000-0003-4319-5955*

Ярослав ПЕРЕГУДА
аспірант кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», findershtein@gmail.com
***ORCID:** 0009-0002-7292-7887*

СПОСІБ ПОБУДОВИ ПРОГРАМНИХ ДЕТЕКТОРІВ ДЛЯ ВИЯВЛЕННЯ ПРОГРАМНИХ БОТІВ В СОЦІАЛЬНИХ МЕРЕЖАХ

***Анотація. Мета даної роботи** полягає у детальному дослідженні ефективності використання великих мовних моделей (Large Language Models), або LLM, для виявлення програмних ботів в соціальних мережах. Робота зосереджується на аналізуванні ефективності різних методів виявлення та визначення потенціалу LLM як засобу для підвищення точності та ефективності процесу ідентифікації ботів.*

Дослідження охоплює аналіз трьох основних підходів до виявлення програмних ботів: аналіз метаданих, текстовий аналіз та аналіз графів. Аналізуються як традиційні методи машинного навчання, так і новітні LLM, які використовуються для аналізу великих даних з соціальних мереж. Основною методикою є порівняльний аналіз, який включає використання розширених наборів даних, таких як TwiBot20 і TwiBot-22, для оцінки продуктивності кожного методу з використанням метрик, таких як точність та F1-міра, що дозволяє отримати об'єктивне уявлення про ефективність різних підходів до виявлення ботів.

***Наукова новизна** даної роботи полягає у використанні LLM для аналізу різноманітних видів даних з соціальних мереж для виявлення програмних ботів. Автори розглядають інтеграцію LLM у традиційні методи виявлення, що дозволяє адаптувати процеси виявлення до складної поведінки програмних ботів, забезпечуючи високу точність і ефективність.*

***Висновки.** LLM демонструють високу ефективність у виявленні програмних ботів, проте мають високу обчислювальну вимогливість. Тому актуальним є застосування гібридних підходів, які поєднують LLM з традиційними методами. Така гібридизація дозволить зменшити використання ресурсів і забезпечити більш стійку та адаптовану систему виявлення ботів. Такий підхід може сприяти поліпшенню загальної продуктивності систем виявлення ботів, зменшенню витрат на обчислювальні ресурси та забезпеченню більш точного і ефективного виявлення шкідливих програм у соціальних мережах. Рекомендується подальше дослідження для вдосконалення інтеграції LLM у системи виявлення ботів, особливо в контексті динамічної поведінки соціальних мереж та еволюції програмних ботів.*

***Ключові слова:** великі мовні моделі, нейронні мережі, аналіз метаданих, програмні боти, соціальні мережі.*

Lesya LYUSHENKO, Yaroslav PEREHUDA. METHOD OF BUILDING SOFTWARE DETECTORS FOR DETECTING SOFTWARE BOTS IN SOCIAL NETWORKS

***Abstract. The purpose** of this work is to study in detail the effectiveness of using large language models (LLM) to detect software bots in social networks. The work focuses on analyzing the effectiveness of different detection methods and determining the potential of LLM as a means to improve the accuracy and efficiency of the bot identification process.*

The study covers the analysis of three main approaches to bot detection: metadata analysis, text analysis, and graph analysis. Both traditional machine learning methods and the latest LLM are analyzed for their ability to analyze big data from social networks. The main technique is benchmarking, which involves the use of extended datasets such as TwiBot20 and TwiBot-22 to evaluate the performance of each method using metrics such as accuracy and F1-measure. It provides an objective view of the performance of different approaches to bot detection.

***The scientific novelty** of this work is the use of LLM to analyze various types of data from social networks to detect software bots. The authors consider the integration of LLM into traditional detection methods, which allows adapting detection processes to the complex behavior of software bots, ensuring high accuracy and efficiency.*

***Conclusions.** LLMs demonstrate high efficiency in detecting software bots, outperforming traditional methods by some indicators. However, given the computational demands of LLM, the authors recommend considering hybrid approaches that combine the advantages of LLM with the efficiency of traditional methods to optimize resource usage and provide a more robust and adaptive bot detection system. This approach can improve the overall performance of bot detection systems, reduce computing resource costs, and provide more accurate and effective detection of malicious actors in social networks. Further research is recommended to improve the integration of LLM into bot detection systems, especially in the context of the dynamic behavior of social networks and the evolution of software bots.*

***Key words:** large language models (LLM), neural networks, metadata analysis, software bots, social networks.*

Соціальні мережі змінили принципи комунікації в суспільстві та стали невід'ємною частиною повсякденного життя. Висока популярність таких мереж призвела до створення різних соціальних онлайн-платформ, кожна з яких надає унікальний досвід та забезпечує спілкування людям з однаковими інтересами у режимі реального часу, без географічних та часових обмежень.

Однак, такі платформи страждають від наявності програмних ботів, які використовуються для маніпулювання та поширення неправдивої інформації. За даними дослідницького співтовариства

маніпулювання за допомогою програмних ботів фіксується під час широкого спектру різноманітних тематичних дискусій. Аналіз облікових записів програмних ботів у соціальних мережах показує, що дії більшості таких ботів призводять до різнопланових онлайн-загроз, а саме: дезінформація [24], втручання у вибори [11], екстремістські кампанії [17], теорії змови [10] тощо. Додатково програмні боти використовуються для поширення пропаганди, нав'язливої реклами та фейкових новин. Так вплив ботів зафіксовано під час дебатів щодо вакцинації [30], реклами електронних сигарет [1] та дебатів щодо пандемії COVID-19 [23]. Така висока активність програмних ботів викликає занепокоєння дослідницького співтовариства та користувачів соціальних мереж щодо цілісності і правдивості інформації.

Дослідження щодо розуміння та виявлення програмних ботів в соціальних мережах завжди були «гонкою озброєнь» [26]. Перші методи були зосереджені на аналізуванні метаданих користувачів за допомогою класифікаторів машинного навчання [22], тоді як оператори програмних ботів маніпулюють поведінковими характеристиками та метаданими облікового запису, щоб уникнути виявлення [6]. Пізніше з'являються мовні моделі для аналізу текстів всередині облікового запису та в постах [31], тоді як оператори програмних ботів періодично публікують скопійовані у справжніх людей пости, щоб заплутати дані моделі і видавати себе за живих користувачів [6]. Новітні моделі збирають мережеві дані про взаємодію користувачів і аналізують її з використанням нейронних мереж на основі графів [2], тоді як сучасні комплексні програмні боти стежать за користувачами, з якими вони пов'язані тим чи іншим чином та стратегічно обривають зв'язки з ними, щоб бути більш непомітними для таких нейронних мереж [16].

Виникнення великих мовних моделей (Large Language Model) або LLM, призвело до значних проривів у сфері обробки великих масивів даних. Завдяки здатності швидко аналізувати, узагальнювати та витягувати інформацію з різнопланових джерел, LLM змінили підходи до виявлення взаємозв'язків. Це особливо важливо у галузях, де необхідно обробляти великі набори неструктурованих даних для розпізнавання тенденцій, аналізу патернів поведінки та розробки нових теорій тощо. Такі моделі відмінно справляються з різноманітними завданнями, здатні слідувати інструкціям [27], але їх використання має певні ризики [25].

Виявлення програмних ботів є надзвичайно складним завданням, головним чином через зростаючу складність та постійну модифікацію цих ботів. Тому актуальним є дослідження можливості застосування гібридних підходів, які поєднують переваги LLM з ефективністю традиційних методів, щоб оптимізувати використання обчислювальних ресурсів і забезпечити більш стійку та адаптовану систему виявлення програмних ботів в соціальних мережах.

1 Існуючі методи виявлення програмних ботів

Існуючі системи виявлення програмних ботів можна класифікувати за об'єктами аналізу: метадані, тексти, графи.

1.1 Виявлення програмних ботів на основі аналізу метаданих

Методи, які виявляють програмних ботів на основі аналізу метаданих, базуються на обробці даних, отриманих з облікових записів та шаблонів активності користувачів. Як правило, для виявлення програмних ботів використовується машинне навчання або нейронні мережі з алгоритмами класифікації. Даними для аналізування є різноманітні характеристики облікового запису користувача [12].

Розглянемо модель **SGBot** (Scalable and Generalizable Bot), яка використовує технології машинного навчання за парадигмою навчання під наглядом (supervised learning) [22]. Методом класифікації в даній моделі є метод «випадковий ліс» (Random forest) [4]. Для ефективного тренування та роботи даної моделі достатньо мати набір даних, який складається лише з невеликої кількості метаданих та їх похідних даних облікового запису користувача (табл. 1).

Таблиця 1

Характеристики облікового запису потрібні для роботи SGBot

Метадані	Похідні дані
Кількість постів	Частота постів
Кількість підписників	Швидкість зросту кількості підписників
Кількість друзів	Швидкість зросту кількості друзів
Кількість підписок	Швидкість зростання кількості підписок
Кількість груп	Швидкість зросту кількості груп
Чи наявна картинка облікового запису	Співвідношення кількості послідовників до друзів
Чи наявна картинка фону (якщо є така можливість)	Довжина псевдоніму, кількість чисел в псевдонімі, довжина імені, кількість чисел в імені, довжина опису профілю, вірогідність вибору псевдоніму
Чи підтверджений користувач	

Дані характеристики облікового запису аналізуються моделлю і результат аналізу повертається у вигляді дробового числа від 0 до 1. Чим ближче це число до 1, тим більша вірогідність, що даний обліковий запис управляється програмним ботом. Такий формат результату є досить поширеним серед моделей, які використовуються як детектори ботів.

Модель **SGBot** має значну масштабованість завдяки зосередженню лише на даних облікового запису користувача, до якої зазвичай можна легко отримати доступ без значних затрат на ресурси. Використання меншої кількості характеристик призводить до незначного зниження в точності визначення, проте отримується можливість аналізувати потік облікових записів у режимі реального часу. Проте, слід зазначити, що в певних випадках доречно аналізувати як можна більшу кількість характеристик. Так, аналогічна модель **Botometer** аналізує більше 1000 різноманітних характеристик (метадані, статистичні дані, похідні дані, шаблони поведінки тощо) [32].

Розповсюдження моделей, які працюють на основі аналізу характеристик облікових записів, призвело до нових реалізацій програмних ботів, які є набагато успішнішими в «обмані» таких методів аналізу [6]. Дана програмні боти успішно маніпулюють профілями облікових записів, а їх шаблони поведінки є досить непередбачуваними. Отже, існуючі методи, засновані на аналізі характеристик облікових записів, стикаються з проблемами в точному виявленні цих нових облікових записів програмних ботів [21].

1.2 Виявлення програмних ботів на основі аналізу тексту

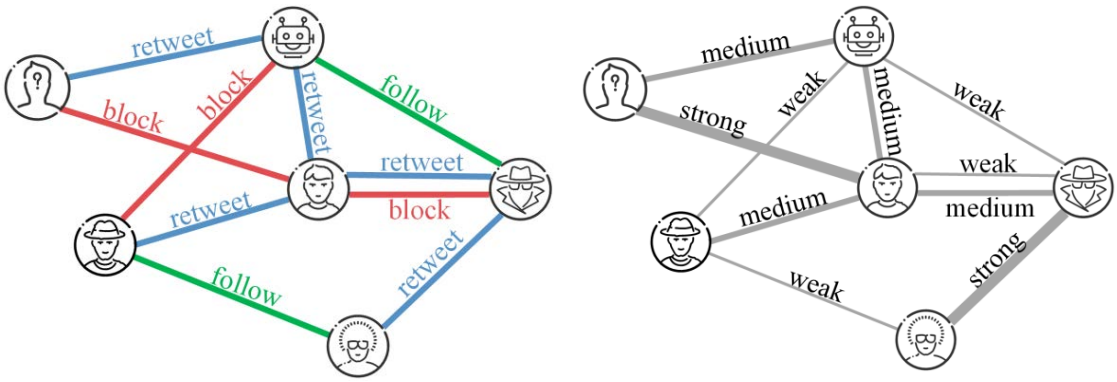
Методи на основі аналізу тексту для виявлення програмних ботів в основному покладаються на методи обробки природньої мови (Natural Language Processing), або NLP. За допомогою NLP зазвичай аналізуються опублікований вміст (пости) та описи облікових записів користувачів. Підходи в цій категорії включають вкладання слів (word embedding) [31], рекурентні нейронні мережі [12], механізми уваги (attention mechanisms) [21] і використання попередньо навчених мовних моделей для кодування постів [9]. Поєднання аналізу постів та аналізу характеристик облікового запису користувача, дозволяє застосували моделі машинного навчання без нагляду [21]

Однією з моделей, яка використовує даний метод є модель для виявлення програмних ботів **RoBERTa** (**R**obustly **O**ptimized **BERT A**pproach) [18]. Дана мовна модель використовується для кодування постів та описів користувачів, закодовані данні яких потім передаються у класифікатор «бот - людина», що заснований на роботі багатошарового перцептрона Румельхарта [19].

Незважаючи на численні існуючі дослідження та вражаючу ефективність методів на основі аналізу тексту, нові облікові записи програмних ботів все ще можуть уникнути виявлення, ділячись викраденим вмістом від справжніх користувачів [6]. Крім того, нещодавні дослідження показали, що покладатися виключно на текстові дані недостатньо для надійного та точного виявлення програмних ботів [5].

1.3 Виявлення програмних ботів на основі аналізу графів

Користувачі та програмні боти в соціальних мережах взаємодіють по-різному та мають різний вплив на інших, що призводить до неоднорідності відношень і впливу (рис. 1).



а) граф відносин користувачів б) граф сили впливів користувачів

Рис. 1. Взаємодія користувачів та програмних ботів в соціальних мережах

Виявлення програмних ботів на основі аналізу графів передбачає аналіз зав'язків та відношень між користувачем та його підписниками, друзями, підписками, спільними групами, тощо. Із таких «об'єктів», зв'язків та відношень можна утворити відповідні структурні графи.

Для виявлення програмних ботів структурні графи аналізуються методами: показників центральності [8], навчання за поданими вузлами (node representation learning) [3], графових нейронних мереж

(Graph Neural Network), або GNN [7]. Комбінування різних методів аналізу графів і текстів [32], а також створення покращених архітектур GNN для аналізу неоднорідних мереж [21], мають значні перспективи для виявлення програмних ботів.

Однією з моделей виявлення програмних ботів на основі аналізу графів, яка використовує неоднорідності зв'язків в соціальних мережах, є RGT (Relational Graph Transformers) [21]. Дана модель використовує топологічну структуру соціальної мережі і будує граф з неоднорідними відношеннями і впливами. У такому графі користувачі виступають у ролі вершин, а неоднорідні відношення у ролі ребер. Даний граф оброблюється шаром перетворення неоднорідного графа, який використовує методи на основі механізмів уваги. В ході обробки формуються графи інтенсивності впливів вершин, мережа семантичної уваги агрегує графи впливів між користувачами і в результаті шар перетворення неоднорідного графа видає результат аналізу.

Проте існуючі методи виявлення програмних ботів мають свої обмеження, які вимагають значних обчислювальних ресурсів і, найважливіше, великих наборів даних для свого тренування. Нещодавнє оголошення монетизації Twitter API (Twitter 2023), який використовувався для отримання тренувальних даних для моделей, робить зазначені вище методи дорогими в обслуговуванні, підтримці та адаптації до нововведень.

1.4 LLM як детектори програмних ботів

Особливістю LLM є використання нейронних мереж, які навчені на великому обсязі немаркованого тексту. Великі мовні моделі, як правило, працюють краще, ніж звичайні мовні моделі, у широкому діапазоні завдань природної мови, особливо коли вони мають доступ до більшої кількості даних і тонкої настройки, що означає, що вони можуть адаптуватися до різних сфер роботи і сценаріїв.

Існує кілька відмінностей між звичайними мовними моделями та великими мовними моделями. LLM є більш комплексними, ніж звичайні мовні моделі і для їх навчання використовувалася більша кількість різноманітних даних, ніж для звичайної мовної моделі. Це означає, що вони можуть фіксувати більш загальні та різноманітні лінгвістичні знання, але також більше інформаційного шуму та помилок.

Поглибимо розуміння особливостей використання LLM в подальшому дослідженні.

2 Порівняльний аналіз можливостей різних моделей щодо виявлення програмних ботів

Метою даного дослідження є порівняльний аналіз існуючих моделей, які використовуються для виявлення програмних ботів. Результатом дослідження є визначення моделей, які найбільш достовірно розрізняють реальних користувачів від програмних ботів. Особлива увага приділена порівнянню ефективності LLM з іншими моделями та визначення потенціалу LLM як засобу для підвищення точності та ефективності процесу ідентифікації ботів.

Об'єктами дослідження є наступні моделі: SGBot, RoBERTa, RGT, LOBO, BotBuster, Botometer, BotPercent, і LMBot, та три великі мовні моделі: Mistral-7B [15], LLaMA2-70b [13] і ChatGPT. Умовно, їх можна поділити на три групи за видами даних:

1. Моделі, які аналізують тільки один вид даних: SGBot аналізує метадані, RoBERTa – текст, і RGT – графи.
2. Моделі які аналізують два виду даних: BotBuster та LOBO, обидва з яких аналізують метадані та текст.
3. Моделі які аналізують три види даних: метадані, текст та графи. До них відносяться моделі Botometer, BotPercent, LMBot та три моделі на основі LLM: Mistral-7B, LLaMA2-70b і ChatGPT.

Порівняння моделей проводитиметься шляхом порівняння результатів тестування моделей. В результаті тестування визначається точність та F1-міри (F-score, F-measure). Під точністю мається на увазі відсоток достовірно визначених реальних користувачів та відсоток програмних ботів. Під F-мірою мається на увазі одна з мір точності тесту, яка обчислюється через влучність та повноту тесту, де влучність є числом правильно визначених позитивних результатів, поділеним на число всіх позитивних результатів, включно з визначеними неправильно, а повнота є числом правильно визначених позитивних результатів, поділеним на число всіх зразків, які повинно було бути визначено як позитивні [20].

Навчання моделей та їх тестування проводилися на двох наборах даних: TwiBot20 [28] і набагато більш великому та різноплановому наборі TwiBot-22 [29]. Зазначені набори даних містять: метадані, статистичні дані, похідні дані, шаблони поведінки, взаємодію та зв'язки з іншими користувачами, облікові записи реальних користувачів та програмних ботів. Дані збиралися із соціальної мережі Twitter, в час, коли Twitter API ще був доступний для дослідницької роботи. Кожен набір даних розділявся на дві нерівні частини: перший і більший набір даних використовувався для навчання моделей, менший набір – для безпосереднього тестування моделей.

2.1 Навчання моделей LLM для виявлення програмних ботів

Оскільки LLM навчені на більшій кількості різноманітних даних, ніж звичайні мовні моделі і є більш комплексними, для їх подальшого навчання для ефективного виявлення програмних ботів достатньо використовувати так звані запити (prompts). Так, використовуючи спеціально сформовані до LLM

запити, їх можна навчити на прикладах, надавши при цьому відповідний контекст. Для більш ефективного навчання, було вирішено розділити запити відповідно до виду даних, які ці запити будуть оброблювати: метадані, текст та графи (рис. 2).

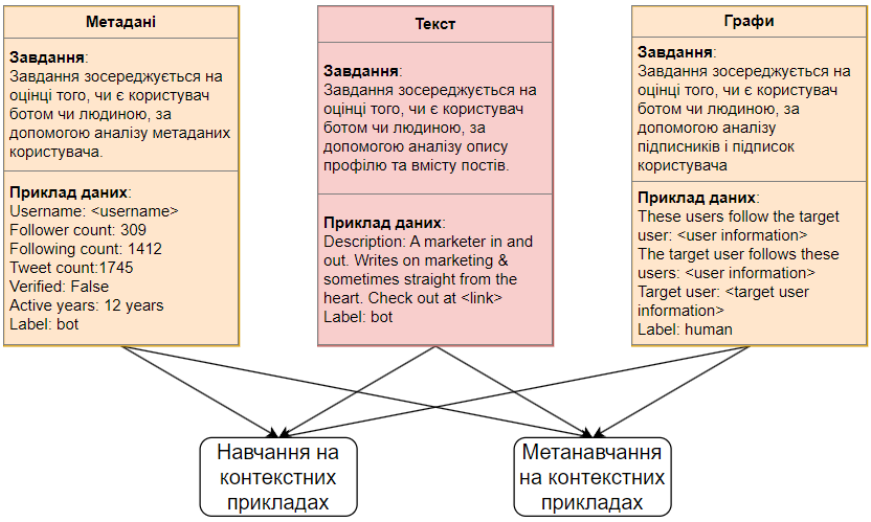


Рис. 2. Схема запитів до LLM для її навчання у виявленні облікових записів програмних ботів

2.1.1 Навчання за допомогою аналізу метаданих

З набору даних випадково вибирається набір із *n* користувачів, метадані яких використовуються для навчання LLM. Метадані облікового запису було послідовно об'єднано у лінійну форму, схожу на природню мову. До кожного запису були надані відповідні мітки про те, чи є користувач програмним ботом чи людиною. Далі, навченій на контекстних прикладах LLM дають запит самій сформувати мітку для наступного користувача (рис. 3).

Для аналізу метаданих облікового запису використовувалася наступні дані: кількість підписників, кількість підписок, кількість постів, чи є обліковий запис підтвердженим та кількість активних років, оскільки саме ці дані найбільше допомагають ідентифікувати соціальних програмних ботів.

2.1.2 Навчання за допомогою аналізу тексту

З набору даних випадково вибирається набір із *n* користувачів, описи облікових записів та зміст постів, яких використовують для навчання LLM.

Evaluate whether a user is a bot or human with the help of several labeled examples. Output the label first and explanation after.

Username: <redacted>. Follower count: 352. Following count: 1432. Tweet count: 1641. Verified: False. Active years: 12 years.

Label: bot

Username: <redacted>. Follower count: 4712064. Following count: 41. Tweet count: 6226. Verified: True. Active years: 14.

Label: human

Username: <redacted>. Follower count: 16491. Following count: 19928. Tweet count: 49652. Verified: False. Active years: 5

Label: ?

Рис. 3. Приклад запиту для навчання LLM на основі аналізу метаданих

До даних постів/описів додається мітка про те, чи є користувач програмним ботом чи людиною, і отриманий набір використовується для навчання LLM на прикладах.

Далі, навченій на контекстних прикладах LLM дають запит самій сформувати мітку для наступного користувача. Для цього їй надається опис цільового облікового запису та вміст його постів. LLM аналізує опис пости і ставить до кожного мітку. Загальним результатом є переважна кількість міток (рис. 4).

Evaluate whether a user is a bot or human with the help of the user's self-written description.
Output the label first and explanation after.
Description: sc/ shenallemoorr ig/ shenallemoore
Label: bot
Description: A marketer in and out. Writes on marketing straight from heart. Check out at <link>
Label: bot
Description: Day 1 Trump supporter. I rode the escalator!
Label: ?

Рис. 4. Приклад запиту для навчання LLM на основі аналізу тексту

2.1.3 Навчання за допомогою аналізу графів

З набору даних випадково вибирається набір із *n* користувачів. Для того, щоб навчити LLM на контекстних прикладах, їй надаються списки із підписників та підписок, кожен з яких має мітку про те, чи є користувач програмним ботом чи людиною. Далі навчена на контекстних прикладах LLM використовує списки підписників та підписок цільового облікового запису, щоб самій сформувати його мітку (рис. 5).

Evaluate whether a user is a bot or human with the help of the user's followers and followings and their labels. Output the label first and explanation after.
These users follow the target user:
<user metadata and description>
Label: bot
The target user follows these users:
<user metadata and description>
Label: human

Target user:
<target user metadata and description>
Label: ?

Рис. 5. Приклад запиту для навчання LLM на основі аналізу графів

2.1.4 Метанавчання на контекстних прикладах

Слід зазначити, що звичайне навчання LLM на контекстних прикладах не є ідеальним способом навчання LLM. Враховуючи специфіку дослідження, було визначено, що більш ефективним методом навчання буде *метанавчання* на контекстних прикладах [14]. Метанавчання на контекстних прикладах має на меті покращити здатність LLM виконувати інструкції шляхом тонкого налаштування LLM до запитів типу {інструкція, вхідні дані, вихідні дані} [27].

Ще однією особливістю є навчання LLM на відносно невеликому наборі даних. Незважаючи на те, на якому наборі даних проводився результат тестування Twibot-20 чи Twibot-22, для навчання або метанавчання на контекстних прикладах з кожного набору відповідно бралися дані лише однієї тисячі облікових записів. Це відносно мало, якщо порівнювати з іншими моделями, для навчання яких потрібно декілька тисяч, або навіть сотень тисяч облікових записів. Незважаючи на, здавалось, малий набір навчальних даних, наданих LLM, вони показали достовірні результати при тестуванні, ознайомитися з якими можна далі.

2.2 Результати порівняльного аналізу моделей

Результати тестування та порівняльного аналізу моделей були розділені на три групи відповідно до кількості видів даних які аналізує та чи інша модель.

2.2.1 SGBot, RoBERTa та RGT

До першої групи моделей належать SGBot, RoBERTa та RGT. Результати їх тестування наведені в таблиці 2.

Таблиця 2

Результати тестування моделей SGBot, RoBERTa та RGT

Назва моделі	Набір даних Twibot-20		Набір даних Twibot-22	
	Точність, %	F1-міра	Точність, %	F1-міра
SGBot	81.6	0.847	62.3	0.394
RoBERTa	75.5	0.732	63.3	0.431
RGT	86.6	0.880	50.9	0.509

За результатами тестування модель RGT показала себе краще, ніж дві інші моделі на наборі даних Twibot-20, і гірше на наборі даних Twibot-22. На другому наборі даних у даної моделі все ще найкращий показник F1-міри, проте вона значно відстає від інших моделей по критерію точності. Це обумовлено тим, що набір даних Twibot-22 є більш великим і різноплановим, а отже надає більш широкий спектр даних для аналізу даних облікового профілю. Таким чином, модель RGT краще працює на обмежених наборах даних, в той час як модель RoBERTa має кращі результати на більш широких наборах даних. Схожу тенденцію також можна побачити в результатах тестування інших моделей.

2.2.2 BotBuster та LOBO

До другої групи моделей належать BotBuster та LOBO. Результати їх тестування наведені в таблиці 3.

Таблиця 3

Результати тестування моделей BotBuster та LOBO

Назва моделі	Набір даних Twibot-20		Набір даних Twibot-22	
	Точність, %	F1-міра	Точність, %	F1-міра
BotBuster	77.2	0.811	62.7	0.439
LOBO	76.2	0.806	55.2	0.197

За результатами тестування зазначимо, що модель BotBuster перевершує модель LOBO на обох наборах даних: Twibot-20 та Twibot-22.

2.2.3 LMBot, BotPercent, Botometer, Mistral-7B, LLaMA2-70b та ChatGPT

До третьої та найбільшої групи моделей належать LMBot, BotPercent, Botometer та три моделі на основі LLM: Mistral-7B, LLaMA2-70b та ChatGPT. Слід зазначити, що для моделі ChatGPT представлені результати тестування як при звичайному навчанні на контекстних прикладах, так і при метанавчанні.

Результати тестування моделей показують, що ChatGPT, навчений за допомогою метанавчання на контекстних прикладах, має кращі показники, ніж інші моделі на обох наборах даних Twibot-20 та Twibot-22 (таблиця 4).

Хоч результати тестування мають значні відмінності в залежності від набору даних, як було сказано раніше, це обумовлюється різницею між цими наборами даних. Отже ChatGPT, навчений на наборі даних Twibot-20 за допомогою простого навчання на контекстних прикладах, має значно гірші результати, ніж при метанавчанні. Аналогічна ситуація і при тестування на наборі даних Twibot-22, хоча в даному випадку різниця порівняно не така велика. Це, в свою чергу, ще раз показує перевагу метанавчання перед звичайним навчання для тренування LLM.

Таблиця 4

Результати тестування моделей LMBot, BotPercent, Botometer, Mistral-7B, LLaMA2-70b та ChatGPT

Назва моделі	Набір даних Twibot-20		Набір даних Twibot-22	
	Точність, %	F1-міра	Точність, %	F1-міра
LMBot	85.6	0.876	-	-
BotPercent	84.5	0.864	73.1	0.726
Botometer	53.1	0.531	75.5	0.585
Mistral-7B	60.9	0.573	58.2	0.534
LLaMA2-70B	66.2	0.658	66.8	0.685
ChatGPT	63.2	0.557	73.5	0.705
ChatGPT (метанавчання)	89.9	0.914	76.9	0.792

2.2.4 RoBERTa, RGT, BotBuster та ChatGPT

Об'єднаємо результати таблиць 6-7, виокремивши моделі, які показали себе найкраще в кожній із трьох груп. Отримаємо результати тестування моделей RoBERTa, RGT, BotBuster та ChatGPT в таблиці 5.

Таблиця 5

Результати тестування моделей RoBERTa, RGT, BotBuster та ChatGPT

Назва моделі	Набір даних Twibot-20		Набір даних Twibot-22	
	Точність, %	F1-міра	Точність, %	F1-міра
RoBERTa	75.5	0.732	63.3	0.431
RGT	86.6	0.880	50.9	0.509
BotBuster	77.2	0.811	62.7	0.439
ChatGPT (метанавчання)	89.9	0.914	76.9	0.792

В результаті велика мовна модель ChatGPT, навчена за допомогою метанавчання на контекстних прикладах, показала найкращі результати серед усіх трьох груп моделей. При правильному методі навчання вона має найкращі результати точності та F1-міри незалежно від набору даних, який використовувався для навчання та тестування моделі.

Отже, результати порівняльного аналізу показали що великі мовні моделі мають високий потенціал щодо виявлення програмних ботів. Для свого навчання вони не потребують відносно широкого набору даних порівняно з іншими моделями, і при правильному методі навчання вони можуть досягти високої ефективності роботи, вираженої в точності визначення програмних ботів та людей серед користувачів.

3 Покращення системи виявлення програмних ботів

Моделі LLM є потужними сучасними інструментами, які можуть досягти чудових результатів у багатьох сферах та задачах, без потреби у обширних наборах навчальних даних. З іншої сторони моделі LLM вимагають більше обчислювальних ресурсів, таких як пам'ять, час та обчислювальних потужностей для навчання та роботи. Інші моделі є менш вимогливими до ресурсів, ніж великі мовні моделі. Це означає, що вони можуть працювати швидше й дешевше. Тому незважаючи на позитивні результати, використання моделей на основі LLM в тому вигляді, в якому вони існують на даний момент, не є оптимальним.

З урахуванням вищезазначеного, має сенс комбінований підхід в використанні досліджених методів. На (рис. 6) запропонована принципова схема системи детектора ботів, що використовує моделі LLM разом з іншими моделями для виявлення програмних ботів. Основа даної системи полягає у використанні моделей LLM лише у випадках, коли інші моделі мають складності у ідентифікації користувача. Розглянемо даний метод на прикладі моделі SGBot. Результат роботи даної моделі видається у вигляді дробового числа від 0 до 1. Чим ближче це число до 0,5 тим більше у моделі виникало складностей при ідентифікації користувача. У таких випадках можна використати LLM модель для уточнення результату моделі SGBot**.** Діапазон, при якому використовується модель LLM слід підібрати так, щоб збільшити точність результату ідентифікації і при цьому не потребувати значних обчислювальних ресурсів постійно. В результаті, сформована система буде мати підвищену точність з незначним збільшенням потреб в обчислювальних ресурсах.

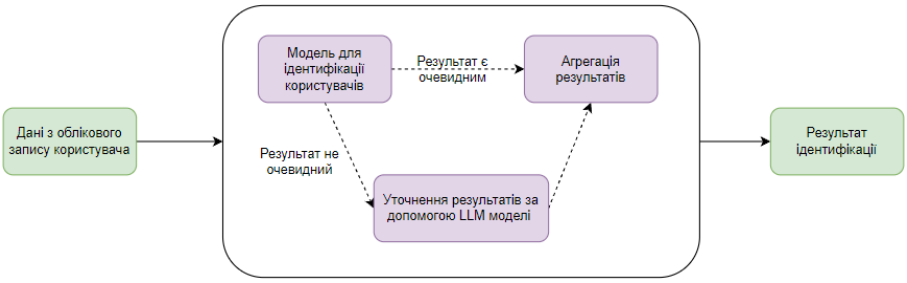


Рис. 6. Принципова схема запропонованої системи визначення програмних ботів

Висновки

Наразі існують різноманітні методи виявлення програмних ботів на основі машинного навчання та нейронних мереж. Кожен із методів підходить для вирішення конкретних сценаріїв виявлення програмних ботів, що забезпечує оптимальні рішення для конкретних випадків використання. Точність цих методів значно знижується в загальніших сценаріях виявлення програмних ботів, що охоплюють, зокрема, різні часові періоди, теми обговорення та мови.

Експерименти на двох широко поширених наборах даних демонструють, що виявлення програмних ботів на основі LLM може досягти високої точності, незважаючи на низьку кількість навчальних даних. Проте, для своєї роботи LLM моделі вимагають значних обчислювальних ресурсів, і, беручи до уваги

кількість користувачів і відповідну величину потоків даних у різноманітних соціальних мережах, використання таких моделей не є оптимальним.

Для вирішення цієї проблеми була запропонована схема системи виявлення програмних ботів. Основа запропонованої системи полягає у використанні ресурсоємних моделей LLM лише у випадках, коли інші моделі мають складності щодо ідентифікації користувача.

Список використаних джерел:

1. Allem J.-P., Ferrara E. The importance of debiasing social media data to better understand e-cigarette related attitudes and behaviors. *Journal of medical Internet research*. 2016. Vol. 18. № 8.
2. BIC: Twitter bot detection with text-graph interaction and semantic consistency. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics / Lei Z. et al. Canada, 2023. Vol. 1. P. 10326–10340.
3. Bot2Vec: A general approach of intra-community oriented representation learning for bot detection in different types of social networks. Information Systems / Pham P. et al. 2022. Vol. 103. DOI: 10.1016/j.is.2021.101771.
4. BotOrNot: A system to evaluate social bots. Proceedings of the 25th International Conference Companion on World Wide Web / Davis C.A. et al. 2016. P. 273–274.
5. BotRGCN: Twitter bot detection with relational graph convolutional networks. Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining / Feng S. et al. IEEE, 2021. P. 236–239.
6. Cresci S. A decade of social bot detection. *Communications of the ACM*. 2020. Vol. 63. № 10. P. 72–83.
7. Detect me if you can: Spam bot detection using inductive representation learning. Companion proceedings of the 2019 World Wide Web conference / Ali A.S. et al. 2019. P. 148–153.
8. Detecting bots in social-networks using node and structural embeddings. *Journal of Big Data* / Dehghan A. et al. 2023. Vol. 10. № 1. P. 1–37.
9. Dukic D., Keca D., Stipic D. Are you human? detecting bots on twitter using BERT. 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA). IEEE, 2020. P. 631– 636.
10. Ferrara E. What types of covid-19 conspiracies are populated by twitter bots? First Monday, 25(6), 2020. DOI: 10.5210/fm.v25i6.10633.
11. Howard P.N., Kollanyi B., Woolley S. Bots and automation over twitter during the US election. *Computational propaganda project : working paper series*. 2016. № 21(8).
12. Kudugunta S., Ferrara E. Deep neural networks for bot detection. *Information Sciences*. 2018. № 467. P. 312–322.
13. Llama 2: Open foundation and fine-tuned chat models / Touvron H. et al. 2023. (Preprint arXiv:2307.09288).
14. MetaICL: Learning to learn in context. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies / Min S. et al. 2022. P. 2791–2809.
15. Mistral 7b / Jiang A.Q. et al. 2023. (Preprint arXiv:2310.06825).
16. Multi-modal social bot detection: Learning homophilic and heterophilic connections adaptively. Proceedings of the 31st ACM International Conference on Multimedia / Li S. et al. ACM, 2023 P. 3908–3916.
17. Predicting online extremism, content adopters, and interaction reciprocity. Social Informatics / Ferrara E. et al. Bellevue, 2016. P. 22–39.
18. Roberta: A robustly optimized BERT pretraining approach / Liu Y. et al. 2019. (Preprint arXiv:1907.11692).
19. Rumelhart D.E., Hinton G.E., Williams R.J. Learning Internal Representations by Error Propagation, Parallel Distributed Processing, Explorations in the Microstructure of Cognition. *Biometrika* / ed. Rumelhart D.E., McClelland J. 1986. Vol. 1. № 71. P. 599-607.
20. Sasaki Y. The truth of the F-measure. *Teach tutor mater*. 2007. Vol. 1. № 5. P. 1-5.
21. Satar: A self-supervised approach to twitter account representation learning and its application in bot detection. Proceedings of the 30th ACM International Conference on Information & Knowledge Management / Feng S. et al. ACM, 2021. P. 3808–3817.
22. Scalable and generalizable social bot detection through data selection. *Proceedings of the AAAI conference on artificial intelligence* / Yang K.-C. et al. 2020. Vol. 34. P. 1096–1103.
23. Shahi G.K., Dirkson A., Majchrzak T.A. An exploratory study of COVID-19 misinformation on Twitter. *Online social networks and media*. 2021. № 22.
24. Social bot-aware graph neural network for early rumor detection. Proceedings of the 29th International Conference on Computational Linguistics / Huang Z. et al. Gyeongju, 2022. P. 6680– 6690.
25. Taxonomy of risks posed by language models. Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency / Weidinger L. et al. ACM, 2022. P. 214–229.
26. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. Proceedings of the 26th international conference on world wide web companion / Cresci S. et al. 2017. P. 963–972.
27. Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems / Ouyang L. et al. 2022. № 35.
28. Twibot-20: A comprehensive twitter bot detection benchmark. Proceedings of the 30th ACM International Conference on Information & Knowledge Management / Feng S. et al. 2021. P. 4485–4494.
29. Twibot-22: Towards graph-based twitter bot detection. Advances in Neural Information Processing Systems / Feng S. et al. 2022. Vol. 35. P. 35254–35269.
30. Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate. *American journal of public health* / Broniatowski D.A. et al. 2018. Vol. 108. № 10. P. 1378–1384.
31. Wei F., Nguyen U.T. Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. 2019 First IEEE International conference on trust, privacy and security in intelligent systems and applications (TPSISA). IEEE, 2019. P. 101–109.
32. Yang K.-C., Ferrara E., Menczer F. Botometer 101: Social bot practicum for computational social scientists. *Journal of Computational Social Science*. 2022. Vol. 5. № 2. P. 1511–1528.