

додаємо перевірки людської взаємодії та застосовуємо LSTM-ядро з privacy-first нормалізацією для ухвалення рішень allow / step-up / block у реальному часі.

Підсумовуючи, мультимодальний fingerprinting із поєднанням клієнтських, мережесих і поведінкових сигналів забезпечує раннє й низьколатентне відсікання ботів без залежності від cookies. LSTM-ядро та короткогоризонтний аналіз послідовностей підвищують чутливість до headless/емуляцій і «low-and-slow» патернів. Політика allow / step-up / block на периферії трафіку зменшує навантаження на бекенд і покращує якість даних аналітики. Очікуємо зниження хибнопозитивних спрацювань за фіксованого рівня виявлення й підвищення стійкості до VPN/проксі та крос-браузерного спуфінгу.

#### Список джерел

1. Zakutynskiy I., Kalishuk O., Iavich M., Nebylytsia V., Yehunko V. AI-based user identification method for web services. URL: <https://www.researchgate.net/publication/392391397> (дата звернення: 20.10.2025).

2. Imperva Bad Bot Report. URL: <https://www.imperva.com/resources/resource-library/reports/bad-bot-report/> (дата звернення: 20.10.2025).

3. Zhao R. Toward the flow-centric detection of browser fingerprinting. *Computers & Security*. 2023. P. 103642. URL: <https://doi.org/10.1016/j.cose.2023.103642> (дата звернення: 20.10.2025).

4. Tracking your browser with high-performance browser fingerprint recognition model / W. Jiang et al. *China Communications*. 2020. Vol. 17, № 3. P. 168–175. URL: <https://doi.org/10.23919/jcc.2020.03.014> (дата звернення: 20.10.2025).

5. A Survey of Browser Fingerprint Research and Application / D. Zhang et al. *Wireless Communications and Mobile Computing*. 2022. Vol. 2022. P. 1–14. URL: <https://doi.org/10.1155/2022/3363335> (дата звернення: 20.10.2025).

#### КАЛІБРУВАННЯ ПАРИ РОБОТИЗОВАНИХ МАНІПУЛЯТОРІВ У СХЕМІ «ВЕДУЧА–ВЕДЕНА» / CALIBRATION OF A LEADER–FOLLOWER PAIR OF ROBOTIC MANIPULATORS

*Волинець Є.А. / Y. Volynets*

Національний університет “Киево-Могилянська Академія” / National University of Kyiv-Mohyla Academy

04655, Київ, вул. Григорія Сковороди, 2, факультет інформатики, кафедра мережних технологій

E-mail: [ye.volynets@ukma.edu.ua](mailto:ye.volynets@ukma.edu.ua)

This work addresses a systems-level reliability gap in leader–follower manipulation: nominally identical commands do not yield identical states. We present a reproducible joint-level calibration that estimates per-joint direction and zero offset from controlled rotations, applies robust angle normalization to avoid wrap-around, and standardizes a small pose set for validation and control. Empirical evaluations indicate that previously significant pose errors are reduced to minor, task-tolerant deviations.

Для багатоланкових роботизованих маніпуляторів із розумними сервоприводами (Dynamixel, Feetech) типовими джерелами похибок є механічні неточності та зазори, люфт, некоректне нульове базування суглобів і порушення орієнтації привода під час складання кінематичних пар. Навіть за однакових алгоритмів керування дві зовні однакові системи можуть суттєво відрізнятися за фактичними координатами, що ускладнює масштабування та обслуговування. У прикладних сценаріях (телеоперування, навчальні стенди) ведучою є дешева, полегшена рука, з якою безпосередньо працює оператор; веденою — дорога, промислова рука, що виконує роботу з об'єктами та має точно відтворювати рухи ведучої. Через відмінності в

кінематичних ланцюгах і жорсткості вузлів навіть малі зсуви нульового базування спричиняють помітну розбіжність траєкторій, тож міжсистемне калібрування є критично необхідним.

Щоб усунути ці відхилення, запропоновано універсальну програмну процедуру калібрування у схемі «ведуча–ведена», яка визначає та компенсує кутові зсуви нульових положень і узгоджує напрям обертання кожного ступеня вільності. Реалізацію виконано мовою Python у вигляді окремого скрипта, що обмінюється даними з маніпуляторами через послідовні інтерфейси (UART/TTL через USB-перетворювач) і формує YAML-параметри калібрування для подальшого використання у системі керування.

Алгоритм калібрування пари «ведуча–ведена» для кожного суглоба визначає знак напрямку  $D$  та кутове зміщення нуля  $S$  у градусах, щоб ведена рука відтворювала траєкторію ведучої. На першому кроці обидва маніпулятори проводять до однакової початкової пози, уникаючи кутів сервоприводів поблизу  $0^\circ/360^\circ$  (з порогом  $\approx 20^\circ$ ), і за потреби механічно зміщуючи ближче до  $\sim 180^\circ$ . Далі для кожного суглоба оператор послідовно обертає його на обох маніпуляторах приблизно на  $45^\circ$  в однакових напрямках, фіксуючи знак фактичного обертання; якщо знаки збігаються — напрям вважається узгодженим, якщо ні — для цього суглоба задають інверсію напрямку ( $D = -1$ ). Після визначення напрямку обчислюють кутову різницю між суглобами двох рук у стартовій позі, приводять її до «звичного» діапазону ( $-180^\circ, 180^\circ$ ) без стрибків через  $0^\circ/360^\circ$  окружності і округлюють до найближчого кроку (практично — до кратних сорока п'яти градусів), щоб зменшити вплив люфту та шуму; за потреби різницю додатково стабілізують усередненням по кількох позах. Окремо для веденої руки знімають реперні пози — «початкові», «робоча» — для подальшого використання в конфігурації. Результатом є набір параметрів для кожного суглоба: чи потрібно інвертувати напрям і яку кутову поправку застосувати, плюс зафіксовані опорні пози; ці дані далі використовуються, щоб ведена (дорога) рука точно відтворювала рухи ведучої (дешевої) попри відмінності в механіці.

Результати записуються до конфігураційного файлу у форматі YAML:

```
joint_1:
  offset_deg: -90
  direction: -1
joint_2:
  offset_deg: 45
  direction: 1
...
```

Під час ініціалізації ці зсуви застосовуються як компенсація, а напрям обертання узгоджується зі знаком ведучої руки.

Реалізований підхід використовує офіційний Dynamixel SDK для доступу до сервоприводів (UART/TTL) та модуль взаємодії з кінематичними ланцюгами; забезпечено синхронізоване опитування обох маніпуляторів, нормалізацію кутів із урахуванням меж обертання, виявлення потреби інверсії напрямку та генерацію YAML-файлу з параметрами компенсації. Застосування такої процедури забезпечило узгоджене відтворення рухів між ведучою (дешевою) та веденою (промисловою) руками попри відмінності в механіці; фактичні відхилення зведено до рівня,

прийнятного для надійного копіювання траєкторій, а тривалість підготовки нової системи помітно скорочено порівняно з ручним налаштуванням.

Список джерел:

1. **DYNAMIXEL SDK: Overview** [Електронний ресурс]. — Режим доступу: [https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel\\_sdk/overview/](https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_sdk/overview/)
2. DYNAMIXEL Protocol 2.0 — Communication Manual [Електронний ресурс]. Режим доступу: <https://emanual.robotis.com/>
3. **ROBOTIS-GIT. DynamixelSDK** : репозиторій програмного забезпечення [Електронний ресурс]. — GitHub. — Режим доступу: <https://github.com/ROBOTIS-GIT/DynamixelSDK>
4. **Hugging Face. LeRobot — Documentation** [Електронний ресурс]. — Режим доступу: <https://huggingface.co/docs/lerobot/en/index>

## АВТОМАТИЧНЕ ФОРМУВАННЯ ОНТОЛОГІЇ ТОВАРІВ НА ОСНОВІ АНАЛІЗУ ДАНИХ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

Жежерун О. П., Колесніков А.О.

Національний університет «Києво-Могилянська академія»

вул. Сковороди 2, м. Київ, 04070, Україна, [anton.kolesnikov@ukma.edu.ua](mailto:anton.kolesnikov@ukma.edu.ua)

The article presents a system for automatic generation of product ontology based on analysis of heterogeneous data from multiple e-commerce sources. The system architecture and algorithm for concept extraction from natural language texts without manual synonym dictionary creation are described. The system generated an ontology with 486 concepts and 1216 relationships with F1=95.2% extraction accuracy. The system uses a four-layer hybrid architecture with transformer embeddings (gte-small, 384-dimensional) and HNSW indexing (M=16, efSearch=16). Experimental deployment on 700,000 products from 34 sources in four languages showed F1=95.2% concept extraction accuracy at 13 products per second processing speed. Main advantages: no need for large labeled datasets, automatic multilingual processing without translation dictionaries, ability to supplement ontology with new concepts without retraining. The system can be adapted for other domains: medicine, finance, logistics.

Keywords: ontology engineering, knowledge base, product ontology, natural language processing, transformer embeddings, semantic matching, big data.

Після створення концепції Semantic Web онтологія стала синонімом рішення проблем розуміння природної мови комп'ютерами [1]. Проте ручне створення онтологій потребує значних інтелектуальних ресурсів та швидко застаріває. Щоб знайти рішення, з'явився напрям онтологічної інженерії, який вивчає шляхи автоматизації генерування знань з тексту [2,3].

У процесі роботи розглянуто задачу автоматизованої генерації онтології товарів з використанням гетерогенних даних з 34 джерел електронної комерції чотирма мовами. Побудовано систему, яка формує онтологію з 486 концептів без ручного створення словників.

Традиційні підходи потребують залучення експертів та ручного створення правил, що не масштабується при роботі з мільйонами товарів [3,4]. Сучасні методи можна поділити на три класи: словникові системи (потребують ручних синонімів), правилові системи (не масштабуються), системи на основі машинного навчання (потребують великих розмічених датасетів) [5].

### МЕТОДОЛОГІЯ

Для виокремлення концептів використано трансформерні нейронні мережі, які генерують векторні представлення текстів у багатовимірному просторі. На відміну від класичних підходів (стемінг, лематизація), трансформери захоплюють семантичний зміст та автоматично виявляють близькість концептів у різних мовах без словників перекладу.