

КЛАСИФІКАЦІЯ КІБЕРАТАК З ВИКОРИСТАННЯМ МОДЕЛІ FASTTEXT

У статті представлено дослідження парсингу рядків користувацьких агентів (UAS) для машинного навчання та розглянуто особливості програмних бібліотек. Виконано реалізацію точнішого та структурованого визначення властивостей кібератак без складних регулярних виразів. Наведено переваги підходу: покращену якість даних, зменшення витрат на обробку і полегшення інтеграції в ML-пайплайнами. Стаття містить приклади коду та механізми впровадження рішення у проекти, де потрібний аналіз інформації про кібератаки на пристрої клієнтів фінансових установ.

Ключові слова: кібератака, UAS, fastText машинне навчання, ML

Під час взаємодії користувача з вебсистемою, браузер надсилає HTTP-запити на сервер для отримання необхідного вмісту [1, 2], відправлення даних або виконання інших дій. Такі запити зазвичай містять кілька заголовків – це пари «ключ-значення», що описують параметри запиту. User-Agent String (скорочено UAS) є HTTP-заголовком, який описує програмне забезпечення, яке діє від імені користувача (рис. 1).

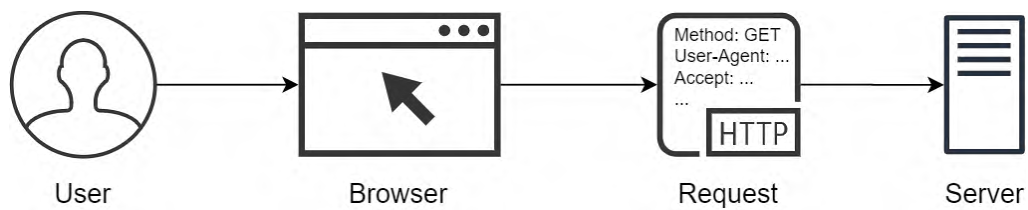


Рис. 1. UAS описує властивості браузера

Початкове призначення UAS полягало в узгодженні контенту, тобто в механізмі визначення найкращого контенту для користувача залежно від інформації у відповідному UAS (наприклад, формат зображення, мова документа, текстове кодування тощо). Дана інформація зазвичай містить дані про середовище [3, 4], в якому працює браузер (пристрій, операційна система та її версія, регіональні налаштування), движок браузера та розкладки та їх версії та інше (рис. 2).

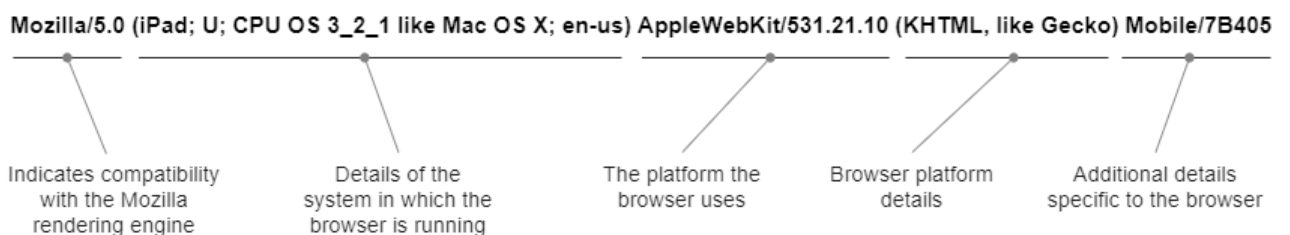


Рис. 2. Приклад типового UAS та його елементів

UAS має багато практичних застосувань. Найпоширенішим є вебаналітика, тобто звітність про склад трафіку для оптимізації ефективності вебсайту. Інше застосування – управління вебтрафіком, що включає блокування небажаних ботів, зменшення навантаження на сайт від непотрібних відвідувачів, запобігання клікфроду (шахрайські кліки) та інші подібні завдання. Завдяки великій кількості інформації, яку вони містять, UAS також можуть слугувати джерелом даних для застосування методів машинного навчання.

Елементи User-Agent String (UAS) можуть слугувати корисними проксі-ознаками характеристик користувача, таких як стиль життя, технічна обізнаність або навіть рівень добробуту. Наприклад, користувач, який зазвичай відвідує вебсайт із преміального мобільного пристрою, ймовірно, відрізняється від користувача, який заходить на той самий сайт за допомогою браузера Internet Explorer на настільному комп'ютері під керуванням Windows XP. Наявність проксі-ознак таких характеристик, отриманих на основі UAS, є особливо цінною у випадках, коли відсутня будь-яка інша демографічна інформація про користувача (наприклад, коли сайт відвідує нова, неідентифікована особа).

У певних випадках також важливо розрізняти трафік від людей і не-людей (наприклад, ботів). Іноді це зробити досить просто, оскільки автоматизовані вебкраулери використовують спрощений формат UAS, що містить слово «bot» (наприклад, Googlebot/2.1 (+http://www.google.com/bot.html)). Проте деякі краулери не дотримуються цього формату, наприклад, боти Facebook містять у своєму UAS слово facebookexternalhit, і для їхнього виявлення необхідно використовувати словник відповістей.

Відносно простим способом створення ознак для алгоритмів машинного навчання з UAS є застосування парсера, виділення окремих елементів та подальше one-hot кодування. Такий підхід може бути ефективним у простих сценаріях, коли потрібно трансформувати лише найвищі рівні та легко впізнавані елементи UAS у машинозчитувані ознаки. Наприклад, досить просто визначити тип пристрою користувача (мобільний, ПК, сервер тощо). Для такого інжинірингу ознак існує низка високоякісних парсерів, що базуються на регулярних виразах (наприклад, проєкт ua-parser та його реалізації для різних мов програмування).

Однак вищезгаданий підхід швидко стає непридатним, якщо виникає потреба враховувати усі елементи UAS для максимального вилучення інформації. Це пов'язано з двома ключовими причинами. Першою причиною є те, що існуючі рекомендації щодо форматування заголовків User-Agent не є обов'язковими до виконання, і в реальних умовах можна зіткнутися з великою різноманітністю специфікацій UAS. Унаслідок цього забезпечити узгоджений парсинг таких рядків надзвичайно складно. Більше того, щодня з'являються нові пристрої, версії операційних систем та браузерів, що ускладнює підтримку високоякісних парсерів. Другою причиною є те, що кількість можливих елементів UAS і їхніх комбінацій є астрономічно великою. Навіть якщо їх вдалося б закодувати у one-hot форматі, утворена матриця даних була б надзвичайно розрідженою (sparse) та надто великою для обробки на комп'ютерах, які зазвичай використовують фахівці з аналізу даних.

```
mozilla/5.0 (iphone; cpu iphone os 10_3 like mac os x) applewebkit/602.1.50 (khtml, like gecko) crios/56.0.2924.75 mobile/14e5239e
safari/602.1 ruxitsynthetic/1.0 v256748272 t2095300180647406436
mozilla/5.0 (iphone; cpu iphone os 12_4_3 like mac os x) applewebkit/605.1.15 (khtml, like gecko) mobile/15e148 instagram 123.0.0.24.115
(iphone6,2; ios 12_4_3; nl_nl; nl-nl; scale=2.00; 640x1136; 188362626) nw/1
mozilla/5.0 (iphone; cpu iphone os 10_2_1 like mac os x) applewebkit/602.4.6 (khtml, like gecko) mobile/14d27
[fbab/fbios;fbav/84.0.0.36.37;fbbv/52496792;fbdv/iphone7,1;fbmd/iphone;fbns/ios;fbssv/10.2.1;fbss/3;fbcr/3ireland;fbid/phone;fbic/en_gb;fbop/5
;fbrrv/0]
mozilla/5.0 (linux; u; android 8.1.0; fr-fr; redmi s2 build/opml.171019.011) applewebkit/537.36 (khtml, like gecko) version/4.0
chrome/61.0.3163.128 mobile safari/537.36 xiaomi/miuibrowser/10.8.3-g
mozilla/5.0 (ipad; cpu os 10_2_1 like mac os x) applewebkit/602.4.6 (khtml, like gecko) mobile/14d27
[fbab/fbios;fbav/88.0.0.60.70;fbbv/55239788;fbdv/ipad6,4;fbmd/ipad;fbns/ios;fbssv/10.2.1;fbss/2;fbcr/sprint;fbid/tablet;fbic/en_us;fbop/5;fbrrv
/0]
mozilla/5.0 (ipad; cpu os 9_0 like mac os x) applewebkit/601.1.46 (khtml, like gecko) mobile/13a344
[fbab/fbios;fbav/8.0.0.28.18;fbbv/1665515;fbdv/ipad2,1;fbmd/ipad;fbns/iphone os;fbssv/9.0;fbss/1;fbcr/;fbid/tablet;fbic/en_us;fbop/1]
mozilla/5.0 (linux; android 9; oneplus a6000 build/pkq1.180716.001; wv) applewebkit/537.36 (khtml, like gecko) version/4.0
chrome/76.0.3809.111 mobile safari/537.36 [fb_iab/fb4a;fbav/233.0.0.36.117;]
mozilla/5.0 (linux; android 5.1.1; build/lmy49i; wv) applewebkit/537.36 (khtml, like gecko) version/4.0 chrome/59.0.3071.125 safari/537.36
mozilla/5.0 (linux; android 7.0; sm-g925f build/nrd90m; wv) applewebkit/537.36 (khtml, like gecko) version/4.0 chrome/74.0.3729.136 mobile
safari/537.36 jssdk/2 topbuzz/11.2.3 nettype/4g
mozilla/5.0 (macintosh; intel mac os x 10.11; rv:46.0) gecko/20100101 firefox/46.0 | hardenize/v1.1168.1 (https://www.hardenize.com)
```

Рис. 3. Приклад підмножин UAS

Для подолання цих викликів можна застосувати метод зниження розмірності (dimensionality reduction) та представити UAS у вигляді векторів фіксованої довжини, мінімізуючи втрати інформації. Ця ідея не є новою, і оскільки UAS – це, по суті, текстові рядки, таке представлення можна реалізувати за допомогою методів обробки природної мови (NLP). Алгоритм fastText, розроблений дослідниками з Facebook, забезпечує особливо корисні результати. Цей метод має кілька практичних переваг:

- не потребує багато даних, тобто якісна модель може бути навчена навіть на кількох тисячах прикладів;
- добре працює з короткими та структурованими документами, такими як UAS;
- швидко навчається;
- ефективно обробляє «слова поза словником», тобто здатний створювати осмислені векторні подання навіть для тих рядків, які не зустрічались під час навчання моделі.

Офіційна реалізація алгоритму fastText доступна у вигляді окремої бібліотеки мовою C++ та обгортки для Python. Обидві версії добре задокументовані, легко встановлюються та використовуються. Крім того, популярна бібліотека Python gensim має власну реалізацію цього алгоритму. Нижче опишемо навчання моделі fastText у середовищі R.

Існує декілька обгортки для мови R, що працюють із C++ бібліотекою fastText (зокрема, fastText, fastRtext та fastTextR). Втім, найпростішим способом навчання і використання моделей fastText у R, безперечно, є використання офіційної Python-реалізації через пакет reticulate.

Навчання моделі базуються на вибірці з 200,000 унікальних рядків User-Agent із бази даних whatismybrowser.com (рис. 4). Дані зберігаються у текстовому файлі у форматі *plain text*, де кожен рядок містить один UAS, нормалізований до нижнього регістру, жодної іншої попередньої обробки не застосовується.

Навчання некерованої (unsupervised) моделі fastText в R є надзвичайно простим, достатньо викликати команду (рис. 4-а):

```
m_unsup <- ft$train_unsupervised(
  input = "./data/train_data_unsup.txt",
  model = "skipgram",
  lr = 0.05,
  dim = 32L, # vector dimension
  ws = 3L,
  minCount = 1L,
  minn = 2L,
  maxn = 6L,
  neg = 3L,
  wordNgrams = 2L,
  loss = "ns",
  epoch = 100L,
  thread = 10L
)
```

а)

```
test_data <- readLines("./data/test_data_unsup.txt")

emb_unsup <- test_data %>%
  lapply(., function(x) {
    m_unsup$get_sentence_vector(text = x) %>%
      t(.) %>% as.data.frame(.)
  }) %>%
  bind_rows(.) %>%
  setNames(., paste0("f", 1:32))

# Printing out the first 5 values
# of the vectors (of size 32)
# that represent the first 3 UAS
# from the test set:

emb_unsup[1:3, 1:5]
##          f1          f2          f3          f4          f5
## 1 0.197 -0.03726 0.147 0.153 0.0423
## 2 0.182 0.00307 0.147 0.101 0.0326
## 3 0.101 -0.28220 0.189 0.202 -0.1623
```

б)

Рис. 4 – а) Навчання некерованої моделі fastText; б) використання функції, що повертає середнє значення векторів для елементів

Аргумент `dim` у наведеній команді визначає розмірність простору векторних подань. Кожен UAS трансформується у вектор розмірності 32. Інші параметри керують процесом навчання моделі (`lr` – швидкість навчання, `loss` – функція втрат, `epoch` – кількість ітерацій у процесі тренування тощо). Повна інформація щодо параметрів міститься у офіційній документації fastText.

Після того як модель навчена, обчислюються векторні подання для нових випадків з набору потенційних кібератак на фінансові установи. Нижче показано використання ключової функції – `m_unsup$get_sentence_vector()`, яка повертає середнє значення векторів для всіх елементів, що складають заданий UAS (рис. 4-б).

Для того, щоб перевірити чи добре працює дана модель необхідно використовувати отримані векторні подання UAS у подальших задачах машинного навчання та оцінити якість отриманого результату. Втім, перед переходом до побудови моделей для конкретних задач доцільно візуально оцінити якість векторних уявлень fastText, зокрема, за допомогою проєкції tSNE. Для цього можна використовувати відомі графіки tSNE, які дозволяють знизити розмірність векторів для їх візуалізації у 2D або 3D просторі. На зображенні (рис. 5) наведено тривимірну tSNE-візуалізацію векторів, обчислених для UAS із вибірки потенційних кібератак на фінансові установи за допомогою моделі fastText. Попри те, що модель була навчена у некерованому режимі, вона змогла згенерувати векторні подання, які відображають важливі характеристики оригінальних UAS. Таким чином, спостерігається чітке розділення точок за видами кібератак.

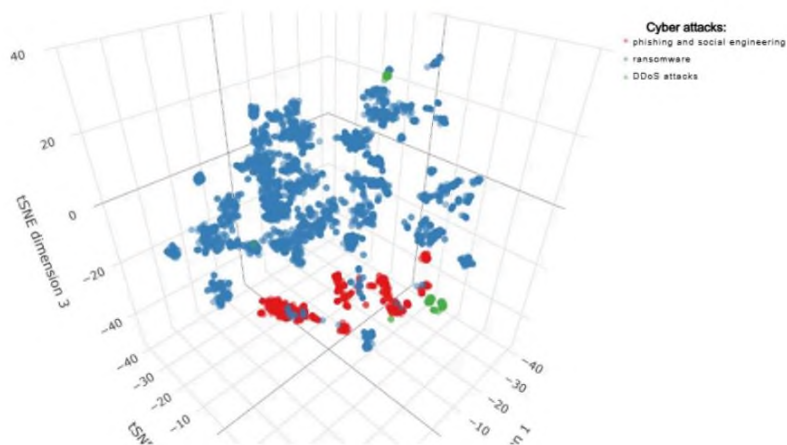


Рис. 5 – 3D візуалізація tSNE векторів, отриманих з некерованої моделі fastText

Навчання керованої (supervised) моделі fastText вимагає наявності розмічених даних. У цьому контексті парсери UAS можуть бути дуже корисними, адже дозволяють швидко отримати мітки для тисяч навчальних прикладів. Алгоритм fastText підтримує як багатокласову, так і мультиетикетну класифікацію. Стандартний формат міток: `__label__ <значення>`. Такі мітки мають передувати кожному документу у навчальному наборі (у випадку багатьох міток – розділені пробілами).

Оскільки ми створюємо векторні подання, які відображають різницю між UAS за типом кібератак на фінансові установи, нижче показано (рис. 6) розмічені дані, які можна використати для навчання відповідної моделі.

```

_label_server mozilla/5.0 (iphone; cpu iphone os 10_3 like mac os x) applewebkit/602.1.50 (KHTML, like Gecko) crios/56.0.2924.75
mobile/14e5239e safari/602.1 ruxitsynthetic/1.0 v256748272 t2095300180647406436
_label_mobile mozilla/5.0 (iphone; cpu iphone os 12_4_3 like mac os x) applewebkit/605.1.15 (KHTML, like Gecko) mobile/15e148 instagram
123.0.0.24.115 (iphone6,2; ios 12_4_3; nl-nl; nl-nl; scale=2.00; 540x1136; 183362626) nw/1
_label_mobile mozilla/5.0 (iphone; cpu iphone os 10_2_1 like mac os x) applewebkit/602.4.6 (KHTML, like Gecko) mobile/14d27
[fbav/fbics;fbav/84.0.0.36.37;fbav/52496792;fbdv/iphone7,1;fbnd/iphone;fbns/ios;fbsv/10.2.1;fbss/3;fbcr/3ireland;fbid/phone;fbic/en_gb;fbop/5
;fbvr/0]
_label_mobile mozilla/5.0 (linux; u; android 8.1.0; fr-fr; redmi s2 build/cpw1.171019.011) applewebkit/537.36 (KHTML, like Gecko)
version/4.0 chrome/61.0.3163.128 mobile safari/537.36 Xiaomi/miuibrowser/10.3.3-g
_label_mobile mozilla/5.0 (ipad; cpu os 10_2_1 like mac os x) applewebkit/602.4.6 (KHTML, like Gecko) mobile/14d27
[fbav/fbics;fbav/84.0.0.60.70;fbav/52496788;fbdv/ipad6,4;fbnd/ipad;fbns/ios;fbsv/10.2.1;fbss/2;fbcr/sprint;fbid/tablet;fbic/en_us;fbop/5;fbvr
/0]
_label_mobile mozilla/5.0 (ipad; cpu os 9_0 like mac os x) applewebkit/601.1.46 (KHTML, like Gecko) mobile/13a344
[fbav/fbics;fbav/8.0.0.28.18;fbav/1665515;fbdv/ipad2,1;fbnd/ipad;fbns/iphone os;fbsv/9.0;fbst/1;fbcr/fbid/tablet;fbic/en_us;fbop/1]
_label_mobile mozilla/5.0 (linux; android 9; oneplus a6000 build/pkq1.180716.001; vv) applewebkit/537.36 (KHTML, like Gecko) version/4.0
chrome/76.0.3809.111 mobile safari/537.36 [fb_iab/fb4a;fbav/233.0.0.36.117;]
_label_mobile mozilla/5.0 (linux; android 5.1.1; build/1my49i; vv) applewebkit/537.36 (KHTML, like Gecko) version/4.0 chrome/59.0.3071.125
safari/537.36
_label_mobile mozilla/5.0 (linux; android 7.0; sm-g925f build/nrd90m; vv) applewebkit/537.36 (KHTML, like Gecko) version/4.0
chrome/74.0.3729.136 mobile safari/537.36 jsjdk/2 topbuzz/11.2.3 nettype/4g
_label_computer mozilla/5.0 (macintosh; intel mac os x 10.11; rv:46.0) Gecko/20100101 Firefox/46.0 | hardenize/v3.1168.1
(https://www.hardenize.com)

```

Рис.6 – Приклад розмічених даних для навчання контрольованої моделі fastText

Команда R для навчання контрольованої моделі fastText на таких розмічених даних схожа на попередню (рис. 8-а):

```

m_sup <- ftstrain_supervised(
  input = "./data/train_data_sup.txt",
  lr = 0.05,
  dim = 32L, # vector dimension
  ws = 3L,
  minCount = 1L,
  minCountLabel = 10L, # min label occurrence
  minn = 2L,
  maxn = 6L,
  neg = 3L,
  wordNgrams = 2L,
  loss = "softmax", # loss function
  epoch = 100L,
  thread = 10L
)

```

а)

```

metrics <- m_sup$test_label("./data/test_data_sup.txt")
metrics["_label_mobile"]
## $__label_ransomware`
## $__label_ransomware`$precision
## [1] 0.998351
##
## $__label_ransomware`$recall
## [1] 0.9981159
##
## $__label_ransomware`$f1score
## [1] 0.9982334

```

б)

Рису.7. – а) Навчання керованої моделі fastText; б) метрики якості для UAS із міткою «ransomware»

Оцінити якість отриманої керованої моделі можна за допомогою розрахунку таких метрик, як точність (precision), повнота (recall) та F1-міра на вибірці. Ці показники можуть бути обчислені як у середньому по всіх мітках, так і для окремих класів. Нижче наведено метрики якості для UAS із міткою «ransomware» (рис. 7-б). Візуальна оцінка tSNE-проекції векторів для даної контрольованої моделі підтверджує її високу якість: спостерігається чітке розділення навчальних прикладів за типом потенційних кібератак на фінансові установи (рис. 8). При навчанні контрольованої моделі алгоритм отримує додаткову інформацію, що дозволяє йому формувати вектори, орієнтовані на конкретне завдання.

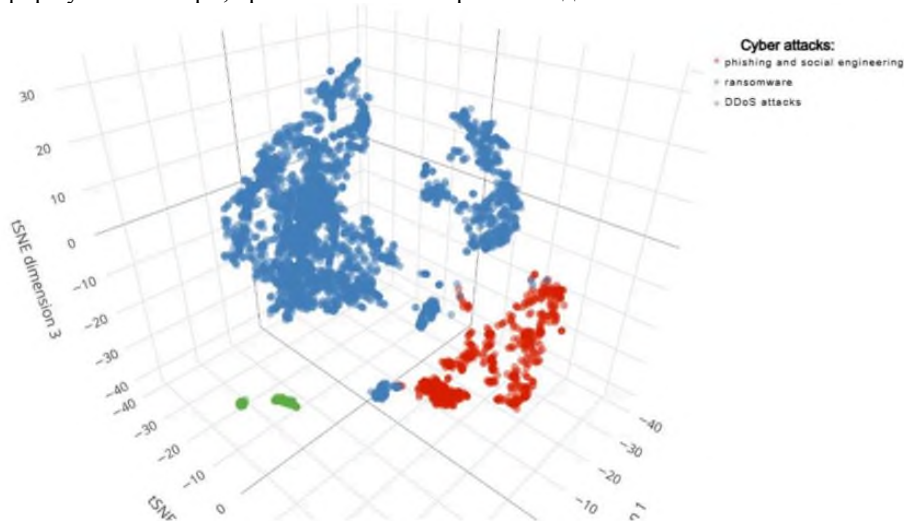


Рис.8. – 3D візуалізація tSNE векторів, отриманих за допомогою контрольованої моделі fastText з мітками, що відповідають типам кібератак

Отже, інформація, яка міститься у рядках UAS, може бути ефективно представлена у вигляді векторних подань низької розмірності. Моделі для створення таких векторів можна навчати як у неконтрольованому, так і в контрольованому режимах. Неконтрольовані вектори є універсальними та можуть бути використані у

будь-яких подальших задачах машинного навчання. Таким чином, можна дійти висновку, що найефективніше навчати моделі на конкретних завданнях. Одним із найбільш ефективних алгоритмів для векторизації UAS є fastText, реалізації якого доступні всіма основними мовами програмування, що використовуються у машинному навчанні.

References

1. Chandre, P., Gumaste, S., Wangikar, A., & Deshmukh, S. (2024). Adaptive Behavioral Authentication for Fraud Detection: Leveraging Real-Time User Behavior to Enhance Financial Security. In 2024 First International Conference on Data, Computation and Communication (ICDCC) (pp. 539–545). 2024 First International Conference on Data, Computation and Communication (ICDCC). IEEE. <https://doi.org/10.1109/icdcc62744.2024.10961554>.
2. Tandon, A., Anitha, C., Kataria, A., Mohammed, N. Q., Al-Khuzai, M. Y., & Almulla, A. A. (2024). Allometry Authentication in the Field of Finance: Creation of Well Secured System using AI Algo Based Systems. In 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 962–967). 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). IEEE. <https://doi.org/10.1109/icacite60783.2024.10617124>
3. Aburbeian, A. M., & Fernández-Veiga, M. (2024). Secure Internet Financial Transactions: A Framework Integrating Multi-Factor Authentication and Machine Learning. AI, 5(1), 177–194. <https://doi.org/10.3390/ai5010010>
4. Burlachenko, I. S., Savinov, V. Y., Tohoiev, O. R., & Zhuravska, I. M. (2021). THE CLOUD GNSS DATA FUSION APPROACH BASED ON THE MULTI-AGENT AUTHENTICATION PROTOCOLS' ANALYSIS IN THE CORPORATE LOGISTICS MANAGEMENT SYSTEMS. Radio Electronics, Computer Science, Control, (4), 95-105.

DOI 10.34132/mspc2025.01.06.18

Dmytro Zavorotnii
Ivan Burlachenko

CLASSIFICATION OF CYBERATTACKS USING THE FASTTEXT MODEL

The paper presents a research of parsing User Agent Strings (UAS) for machine learning and examines the features of relevant software libraries. A more accurate and structured method for identifying cyberattack characteristics is implemented without relying on complex regular expressions. The advantages of the proposed approach are highlighted, including improved data quality, reduced processing costs, and easier integration into machine learning (ML) pipelines. The article provides code examples and outlines implementation mechanisms for projects requiring the analysis of cyberattack data on client devices in financial institutions.

Keywords: cyberattack, UAS, fastText, machine learning, ML

Відомості про авторів / Information about the Authors

Дмитро Заворотній, студент 4-го курсу спеціальності «Комп'ютерні науки» факультету комп'ютерних наук, Чорноморського національного університету імені Петра Могили, м. Миколаїв, Україна. E-mail: devzavorotniydmytro@gmail.com orcid: <https://orcid.org/0009-0005-3895-8964>

Dmytro Zavorotnii, fourth year student, specialty Computer Science, Computer Science Faculty, Petro Mohyla Black Sea National University, Mykolaiv, Ukraine. E-mail: devzavorotniydmytro@gmail.com, orcid: <https://orcid.org/0009-0005-3895-8964>

Іван Бурлаченко, старший викладач кафедри комп'ютерної інженерії, факультету комп'ютерних наук, Чорноморського національного університету імені Петра Могили, м. Миколаїв, Україна. E-mail: ivan.burlachenko@chmnu.edu.ua, orcid: <https://orcid.org/0000-0001-5088-6709>.

Ivan Burlachenko, Senior Lecturer of the Computer Engineering Department of Computer Science Faculty of the Petro Mohyla Black Sea National University, Mykolaiv, Ukraine. E-mail: ivan.burlachenko@chmnu.edu.ua, orcid: <https://orcid.org/0000-0001-5088-6709>.

