КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (33), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.33.1122
УДК 004.7:004.93:004.8

Редько Денис Вадимович
аспірант кафедри інженерії програмного забезпечення та кібербезпеки
Державний торговельно-економічний університет, Київ, Україна
ORCID: 0009-0003-5827-264X
d.redko@knute.edu.ua

Десятко Альона Миколаївна
доктор філософії, завідувач кафедри інженерії програмного забезпечення та кібербезпеки
Державний торговельно-економічний університет, Київ, Україна
ORCID: 0000-0002-2284-3418
desyatko@gmail.com

Байтума Бісарінов Жаканович
доктор філософії «Комп'ютерні науки», кафедра інформаційних систем
Казахський національний університет імені Аль-Фарабі,
Алматинський університет енергетики та телекомунікацій
ORCID: 0000-0002-2218-0749
baituma_bai@gmail.com

АНСАМБЛЕВА КЛАСТЕРИЗАЦІЯ МЕРЕЖЕВОГО ТРАФІКУ НА ОСНОВІ КОНСЕНСУСНОГО ПІДХОДУ

Анотація. В умовах стрімкого зростання обсягів мережевого трафіку та ускладнення корпоративних інформаційних систем (ІС) особливу актуальність набуває розробка ефективних методів аналізу й кластеризації даних. Сучасні підходи до обробки мережевого трафіку вимагають комплексного розгляду численних параметрів та характеристик, що зумовлює необхідність удосконалення існуючих методів кластерного аналізу. У роботі запропоновано вдосконалений підхід до ансамблевої кластеризації мережевого трафіку та методику побудови узгодженої матриці подібності для інтеграції результатів різних алгоритмів кластеризації на основі експоненціальної залежності з метою посилення різниць у вагах алгоритмів, що суттєво підвищує точність кінцевої кластеризації. Реалізована в рамках дослідження на мові Python програмна система об'єднує кілька методів кластеризації, що дозволяє досягти значно більшої стабільності результатів за рахунок використання консенсусного підходу, ефективність якого підтверджена результатами масштабних обчислювальних експериментів. У ході дослідження та обчислювальних експериментів продемонстровано, що ключовою перевагою розробленого підходу є підвищена стійкість до викидів порівняно з традиційними методами (KMeans, DBSCAN), що застосовуються для кластерного аналізу мережевого трафіку компанії, а також більш збалансована кластеризація для складних багатовимірних даних. Також запропоновано залежність для розрахунку ваг алгоритмів із використанням експоненціальної функції, що дає змогу комплексно підходити до інтеграції результатів різних методів кластеризації. Розроблене програмне рішення суттєво розширює методи аналізу мережевого трафіку та забезпечує ефективний практичний інструментарій для підвищення продуктивності роботи корпоративних інформаційних систем. Запропонований підхід може бути успішно адаптований для розв'язання широкого спектра задач аналізу даних, що потребують обробки великих обсягів багатовимірної інформації.

Ключові слова: мережевий трафік, великі дані, методи аналізу даних, кластеризація, колективні рішення, ансамблеві моделі.

ВСТУП

Сучасні методи аналізу та обробки даних стають невід'ємною частиною оптимізації бізнес-процесів, особливо у сфері інформаційних технологій (ІТ), де динамічний розвиток веб-ресурсів і онлайн-послуг супроводжується постійно зростаючим обсягом трафіку компаній. Водночас ефективне управління та аналіз трафіку компанії є ключовим елементом для досягнення високих показників продуктивності, підвищення якості обслуговування клієнтів і забезпечення безпеки корпоративної

інфраструктури [1-4]. В умовах різноманіття джерел даних, часових коливань навантаження та багатовимірності характеристик корпоративного мережевого трафіку актуальним завданням є його кластеризація – процес групування даних, що належать до різних категорій, з метою виявлення прихованих закономірностей та оптимізації витрат ресурсів [5-7]. Одним із найбільш перспективних підходів до вирішення даної задачі є ансамблева кластеризація, яка поєднує переваги різних методів і моделей для досягнення більш точних й стійких результатів [6, 7]. Ансамблі методів кластеризації сприяють підвищенню стабільності результатів аналізу та прогнозування, зменшенню чутливості до викидів і поліпшенню інтерпретованості отриманих кластерів. З огляду на все вищезазначене, у статті розглядаються окремі практичні аспекти програмної реалізації методів ансамблевої кластеризації, що застосовуються для аналізу трафіку компанії, з акцентом на їх ефективності в умовах багатозадачності та відмінностей у характеристиках даних для аналізу.

Постановка проблеми. У процесі кластеризації мережевого трафіку компаній для аналітиків часто виникає необхідність об'єднувати результати, отримані за допомогою різних алгоритмів, для досягнення більш стабільних кластерів, що особливо характерно для складних і багатовимірних даних, де кожен алгоритм може виявляти різні аспекти структури даних, що, своєю чергою, може призводити до невизначеності або зниження точності кластеризації. Отже, головним завданням нашого дослідження є розробка ефективного методу формування консенсусного розбиття на основі кількох алгоритмів кластеризації. Після отримання кластерів за допомогою різних методів ми застосуємо новий підхід для визначення ступеня узгодженості цих кластерів. Для цього побудуємо узгоджену матрицю подібності, яка відображатиме, наскільки класифікації різних алгоритмів збігаються. На основі цієї матриці надалі ми реалізуємо колективний алгоритм мовою Python, який поєднує результати різних методів для досягнення оптимальних кластерів, мінімізуючи вплив чутливості кожного з алгоритмів до їхніх внутрішніх параметрів. У запропонованому підході застосовується експоненціальна залежність для посилення відмінностей у вагах, що, на нашу думку, дозволяє більш точно враховувати значущість кожного алгоритму залежно від його внеску в загальне розбиття. Водночас урахування ймовірнісної залежності між характеристиками роботи ансамблю та показниками якості кластеризації дає змогу підвищити точність розбиття та зменшити помилки, пов'язані з недоліками окремих методів. Отже, ключова проблема полягає в інтеграції результатів декількох алгоритмів кластеризації в єдину структуру, яка буде більш точною, стійкою до змін у даних і чутливою до ключових особливостей трафіку. Розв'язання цього завдання дозволить суттєво підвищити якість кластеризації та зробити її більш передбачуваною і надійною в умовах реальної експлуатації в компаніях.

Аналіз останніх досліджень і публікацій. Експоненціальне зростання обсягів мережевого трафіку в корпоративних інфраструктурах створює суттєві методологічні виклики під час його аналізу, оптимізації та забезпечення інформаційної безпеки. Емпіричні дослідження, результати яких були представлені в низці публікацій [5-10], демонструють високу ефективність методів кластеризації під час сегментації мережевих потоків на гомогенні групи, що сприяє виявленню латентних патернів і аномальних станів.

Сучасні наукові дослідження у сфері застосування кластерного аналізу до мережевого трафіку [11-15] характеризуються значним прогресом у розробці та модифікації алгоритмічних підходів, спрямованих на підвищення точності опрацювання багатовимірних даних. Методологічний інструментарій, використаний авторами зазначених робіт [5-15], охоплював широкий спектр алгоритмів – від класичного K-means та ієрархічної кластеризації до сучасніших підходів, таких як DBScan. Проведені авторами цих робіт емпіричні дослідження підтвердили високу ефективність методів кластеризації для сегментації мережевого трафіку, однак сучасний технологічний рівень і розвиток IT можуть сприяти впровадженню складніших методологічних підходів, зокрема інтеграції з системами машинного навчання та штучного інтелекту. Зокрема, алгоритми BIRCH і DBSCAN демонструють високу ефективність при обробці великих наборів даних і виявленні кластерів довільної конфігурації. Ці тенденції свідчать про зростаючу релевантність кластерного аналізу для дослідження мережевого трафіку та перспективність подальших наукових досліджень у цій галузі [10-17].

Метою данного дослідження є систематизація існуючих методів і підходів до аналізу трафіку компаній із використанням різних методів кластеризації даних, у тому числі колективних (ансамблевих) рішень.

МЕТОДИ ТА МОДЕЛІ

Методи дослідження. У межах дослідження передбачається використання консенсусного підходу до кластеризації даних трафіку великої компанії з метою підвищення стабільності та точності результатів. Після отримання кластеризацій від різних алгоритмів здійснюється побудова узгодженої матриці подібності, яка відображає ступінь узгодженості класифікацій. Для цього використовується

колективний алгоритм, який інтегрує результати різних методів і дозволяє мінімізувати чутливість до параметрів окремих алгоритмів. У процесі побудови коасоціативної матриці відмінностей враховуються ваги базових алгоритмів кластерного аналізу із застосуванням експоненціальної залежності, що підсилює різницю у вагах та враховує ймовірнісну залежність між характеристиками роботи ансамблю та якісними показниками кластеризації.

Результати дослідження. Для більш детального розуміння структури даних, що використовуються у кластерному аналізі, в таблиці 1 нижче наведено основні стовпці мережевих логів, які є вихідним матеріалом для кластеризації. Ці дані включали різні параметри, які були використані для ідентифікації та групування мережевих подій.

Таблиця 1

Опис стовпців таблиці з даними мережевих логів synthetic_data.csv

№	Параметр	Пояснення	Тип
1	timestamp	Час створення події. Це поле може містити дату й час у діапазоні від вчора до теперішнього моменту.	Дата/Час (datetime)
2	event_id	Унікальний ідентифікатор події, що генерується у форматі UUID.	UUID (унікальний ідентифікатор)
3	event_type	Тип події, який визначає її категорію (наприклад, "auth", "traffic" або "IDS/IPS").	String
4	protocol	Протокол передачі даних, який використовується в події. Можливі значення: 'TCP', 'UDP', 'ICMP'.	String
5	source_ip	IP-адреса джерела події. Можливі значення: IPv4.	IP-адреса
6	destination_ip	IP-адреса отримувача події. Якщо подія типу «IDS/IPS», завжди дорівнює 192.168.0.1. Можливі значення: IPv4 або фіксована 192.168.0.1.	IP-адреса
7	application	Застосунок, пов'язаний із подією. Якщо тип події «IDS/IPS», це один із заздалегідь вибраних застосунків. Можливі значення: 'ssh', 'https', 'ftp', 'email', 'database'.	String
8	bytes_sent	Кількість байтів, надісланих під час події. Можливі значення: від 100 до 1 000 000.	integer
9	bytes_received	Кількість байтів, отриманих під час події. Можливі значення: від 100 до 1 000 000.	integer
10	duration	Тривалість події у секундах. Можливі значення: від 1 до 3600 секунд.	integer
11	status	Статус виконання події. Можливі значення: 'success', 'failure'.	String
12	error_code	Код помилки, якщо статус події – "failure", а додаток – "https". Можливі значення: 400, 401, 403, 404, 500, 502, 503, 504.	integer
13	user_id	Ідентифікатор користувача, пов'язаний із подією, окрім подій типу "IDS/IPS". Можливі значення: UUID або None для "IDS/IPS".	UUID
14	device_id	Ідентифікатор пристрою, пов’язаний із подією, окрім подій типу "IDS/IPS". Можливі значення: UUID або None для "IDS/IPS".	UUID
15	location	Локація (країна), з якої здійснюється подія, окрім "IDS/IPS". Можливі значення: назва країни або None для "IDS/IPS".	String
16	signature_id	Унікальний ідентифікатор підпису події.	UUID
17	user_agent	Інформація про браузер або програмне забезпечення користувача, якщо подія не є "IDS/IPS". Можливі значення: стандартні рядки user-agent або None.	String
18	http_method	HTTP-метод, використаний у події, якщо вона не є "IDS/IPS". Можливі значення: 'GET', 'POST', 'PUT', 'DELETE' або None.	String
19	url	URL, на який було надіслано запит, якщо подія не є "IDS/IPS". Можливі значення: URI або None.	String-URL

Продовження таблиці 1

20	query_params	Параметри запиту в URL, якщо подія не є "IDS/IPS". Можливі значення: URI-шлях або None.	String
21	content_type	Тип контенту, що передається через HTTP, якщо додаток – "https". Можливі значення: 'text/html', 'application/json', 'image/png' або None.	String
22	sensor_id	Ідентифікатор сенсора, що зафіксував подію, якщо тип події – "IDS/IPS". Можливі значення: UUID або None.	UUID
23	device_type	Тип пристрою, що бере участь у події, окрім подій "IDS/IPS". Можливі значення: 'server', 'workstation', 'mobile' або None.	String
24	software	Програмне забезпечення, яке використовується на пристрої, окрім подій типу "IDS/IPS" та додатків "email". Можливі значення: наприклад, "Microsoft Office 365", "Google Workspace", "Salesforce" тощо, або None.	String
25	anomaly_score	Оцінка аномалії, яка показує, наскільки подія відрізняється від звичайної поведінки. Можливі значення: від 0 до 100.	integer

У рамках проведення обчислювальних експериментів була використана вибрана хмарна середа розробки Google Colaboratory (Colab), яка надає низку суттєвих переваг для реалізації алгоритмів машинного навчання (МН) та аналізу даних. Ця платформа забезпечує доступ до високопродуктивних обчислювальних ресурсів, зокрема графічних процесорів (GPU) та оперативної пам'яті достатнього обсягу для обробки великих наборів даних, що є особливо важливим під час роботи з багатовимірними даними мережевого трафіку. Важливо, що Colab підтримує інтерактивний режим розробки мовою Python із використанням популярних бібліотек для аналізу даних і МО, що суттєво спрощує процес впровадження та тестування алгоритмів кластеризації.

У ході експериментальних досліджень була реалізована та апробована в обчислювальних експериментах програма для кластеризації мережевого трафіку із використанням декількох методів, а результати кластеризації були візуалізовані за допомогою двовимірного представлення даних, отриманих після застосування методу головних компонент (PCA). Візуалізації на графіках, наведених на рис. 1 та 2, які відображають різні аспекти кластеризації, а також результати консенсусної кластеризації. На рис. 1 показано результати кластеризації, виконаної із використанням різних алгоритмів. Кольорова диференціація для кожного з використаних алгоритмів відображається окремим кольором, що дозволяє візуально оцінити, як алгоритми розділяють дані. Як видно з рисунка 1, консенсусна кластеризація забезпечує більш стабільні результати, оскільки поєднує множинні незалежні класифікації та знижує чутливість до індивідуальних особливостей кожного алгоритму. Консенсусна кластеризація включала кілька етапів, зокрема побудову матриці подібності між результатами кластеризації різних алгоритмів. Ця матриця (див. рис. 2) відображає ступінь узгодженості кластерних міток між різними алгоритмами. У межах застосованого алгоритму використовувалися ваги, отримані з експоненціальної функції подібності (1), які підсилюють відмінності між кластерами залежно від ступеня їхньої узгодженості. Це дозволяє алгоритму більше покладатися на ті алгоритми, які продемонстрували вищу стабільність у класифікації.

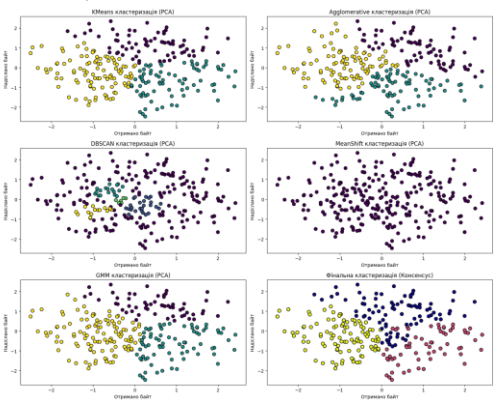

Рис. 1. – Приклад результатів кластеризації мережевого трафіку великої компанії, виконаної за допомогою різних алгоритмів та консенсусної кластеризації.

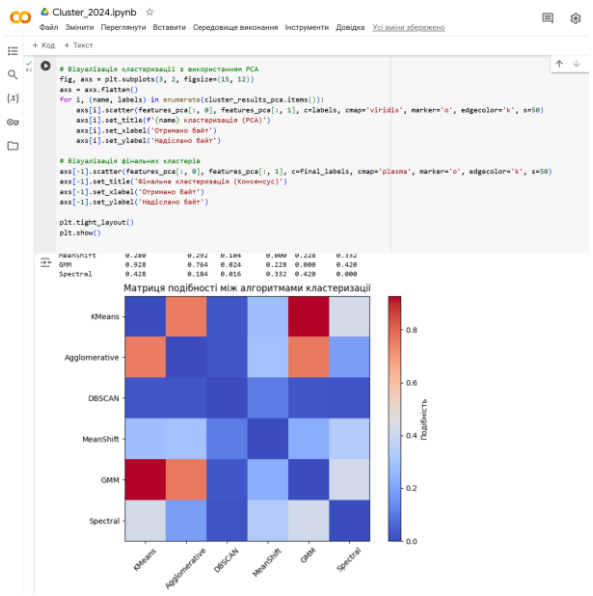

Рис. 2. – Ступінь узгодженості кластерних міток між різними алгоритмами.

Основна ідея полягає в тому, що у нашому випадку вага кожного з алгоритмів в ансамблі залежить від його продуктивності, визначеної метриками якості, такими як ARI та NMI. Пропонується використовувати експоненційну залежність, що дозволить посилити різницю у вагах алгоритмів, особливо коли різниця в метриках якості незначна. Це потенційно зробить метод ансамблевої кластеризації (колективного рішення) більш чутливим до різних рівнів продуктивності. Експоненціальна залежність (1) використовується у наших міркуваннях для підсилення відмінностей у вагах. Парадигма такого підходу ґрунтується на тому, що незначні зміни у значеннях метрик якості призводять до суттєвіших змін у вагах, що є важливим, оскільки більш продуктивні алгоритми повинні мати помітно більшу вагу, адже їхній вплив на фінальний результат буде сильнішим.

Тоді, враховуючи вищесказане, можна записати:

$$w_i = \frac{exp(\alpha \cdot Q_i)}{\sum_{j=1}^{N} exp(\alpha \cdot Q_j)}, \quad (1)$$

де w_i – вага i – го алгоритму; Q_i – відповідна метрика якості (наприклад, ARI або NMI) i – го алгоритму; α – параметр, який регулює ступінь підсилення різниць у вагах (можна підбирати експериментально); N – загальна кількість алгоритмів в ансамблі (колективному рішенні).

Після обчислення ваг для кожного алгоритму та кожної мітки відбувається процес голосування, під час якого кожна мітка голосує за один із кластерів, а підсумкове розбиття формується шляхом вибору найбільш ймовірного кластера на основі сумарної ваги.

Комбінуючи результати декількох методів, вдалося досягти більш точних і збалансованих кластерів, що особливо важливо для складних і багатозначних даних, як у випадку аналізу мережевого трафіку. На відміну від методів, чутливих до викидів (наприклад, KMeans чи DBSCAN), консенсусний підхід є більш стійким, оскільки в ньому інтегруються результати, отримані різними методами.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Запропоновано вдосконалений підхід до ансамблевої кластеризації мережевого трафіку та методику побудови узгодженої матриці подібності для інтеграції результатів різних алгоритмів кластеризації на основі експоненціальної залежності з метою посилення відмінностей у вагах алгоритмів, що підвищує точність фінальної кластеризації. Реалізовано програмну систему мовою Python, яка поєднує кілька методів кластеризації, що забезпечує вищу стабільність результатів завдяки застосуванню консенсусного підходу, підтверджену результатами обчислювальних експериментів. Показано, що перевагою розробленого підходу є підвищена стійкість до викидів порівняно з традиційними методами (KMeans, DBSCAN), які використовуються для кластерного аналізу мережевого трафіку компанії, а також збалансована кластеризація для складних багатовимірних даних. У роботі запропоновано формулу розрахунку ваг алгоритмів із використанням експоненціальної залежності, що дає змогу комплексно

підходити до інтеграції результатів різних методів кластеризації, а розроблене програмне рішення розширює методи аналізу мережевого трафіку та забезпечує практичний інструмент для підвищення ефективності функціонування корпоративних інформаційних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Takyi, K., Bagga, A., & Goopta, P. (2018, August). Clustering techniques for traffic classification: A comprehensive review. In *Proceedings of the 7th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO 2018)* (pp. 224-230). IEEE. https://doi.org/10.1109/ICRITO.2018.8748772
2. Li, J., Zhang, H., Tang, D., & Lin, C. (2021, September). Traffic classification using cluster analysis. In *Proceedings of the International Conference on Computer Information Science and Artificial Intelligence (CISAI 2021)* (pp. 463-467). IEEE. https://doi.org/10.1109/CISAI54367.2021.00094
3. Rodríguez-Rodríguez, J. E., García, V. H. M., & Usaquén, M. A. O. (2018). Corporate networks traffic analysis for knowledge management based on random interactions clustering algorithm. In *Knowledge management in organizations (KMO 2018)* (pp. 523-536). Springer. https://doi.org/10.1007/978-3-319-95204-8_44
4. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281-297).
5. Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129-137. https://doi.org/10.1109/TIT.1982.1056489
6. Cheeseman, P. C., & Stutz, J. C. (1996). Bayesian classification (AutoClass): Theory and results. In *Advances in knowledge discovery and data mining* (pp. 153-180).
7. Zhang, T., Ramakrishnan, R., & Livny, M. (1997). BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, *1*(2), 141-182. https://doi.org/10.1023/A:1009783824328
8. Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. *ACM SIGMOD Record*, *27*(2), 73-84. https://doi.org/10.1145/276305.276312
9. Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD 1996)* (pp. 226-231).
10. Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *ACM SIGMOD Record*, *28*(2), 49-60. https://doi.org/10.1145/304181.304187
11. Subramani, K., Velkov, A., Ntoutsi, I., Kröger, P., & Kriegel, H.-P. (2011, December). Density-based community detection in social networks. In *Proceedings of the IEEE International Conference on Internet Multimedia Systems Architecture and Application (IMSAA 2011)* (pp. 1-8). IEEE. https://doi.org/10.1109/IMSAA.2011.6156334
12. Zander, S., Nguyen, T., & Armitage, G. (2005, November). Automated traffic classification and application identification using machine learning. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN 2005)* (pp. 250-257). IEEE. https://doi.org/10.1109/LCN.2005.35
13. McGregor, A., Hall, M., Lorier, P., & Brunskill, J. (2004). Flow clustering using machine learning techniques. In *Passive and active network measurement (PAM 2004)* (pp. 205-214). Springer. https://doi.org/10.1007/978-3-540-24668-8_21
14. Erman, J., Mahanti, A., Arlitt, M., Cohen, I., & Williamson, C. (2007). Offline/realtime traffic classification using semi-supervised learning. *Performance Evaluation*, *64*(9-12), 1194-1213. https://doi.org/10.1016/j.peva.2007.06.014
15. Wang, Y., Xiang, Y., Zhang, J., Zhou, W., Wei, G., & Yang, L. T. (2013). Internet traffic classification using constrained clustering. *IEEE Transactions on Parallel and Distributed Systems*, *25*(11), 2932-2943. https://doi.org/10.1109/TPDS.2013.307
16. Wang, P., Lin, S. C., & Luo, M. (2016, June). A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs. In *Proceedings of the IEEE International Conference on Services Computing (SCC 2016)* (pp. 760-765). IEEE. https://doi.org/10.1109/SCC.2016.105

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (33), 2026
ISSN 2663 – 4023

Denys Redko
Postgraduate student of the Department of Software Engineering and Cybersecurity
State University of Trade and Economics, Kyiv, Ukraine
ORCID: 0009-0003-5827-264X
d.redko@knute.edu.ua

Alona Desiatko
Doctor of Philosophy in Computer Science,
Head of the Department of Software Engineering and Cybersecurity
State University of Trade and Economics, Kyiv, Ukraine
ORCID: 0000-0002-2284-3418
desyatko@gmail.com

Baituma Bissarinov
Doctor of Philosophy in Computer Science, Department of Information Systems
Al-Farabi Azadli National University, Almaty University of Energy
ORCID: 0000-0002-2218-0749
baituma_bai@gmail.com

ENSEMBLE CLUSTERING OF NETWORK TRAFFIC BASED ON A CONSENSUS APPROACH

Abstract. In the context of the rapid growth of network traffic volumes and the complexity of corporate information systems (IS), the development of effective methods for data analysis and clustering is of particular relevance. Modern approaches to network traffic processing require a comprehensive consideration of numerous parameters and characteristics, which necessitates the improvement of existing cluster analysis methods. The paper proposes an improved approach to ensemble clustering of network traffic and a method for constructing a consistent similarity matrix for integrating the results of different clustering algorithms based on exponential dependence in order to enhance the differences in the weights of the algorithms, which significantly increases the accuracy of the final clustering. The software system implemented in the Python language as part of the study combines several clustering methods, which allows achieving significantly greater stability of the results through the use of a consensus approach, the effectiveness of which has been confirmed by the results of large-scale computational experiments. During the research and computational experiments, it was demonstrated that the key advantage of the developed approach is increased resistance to outliers compared to traditional methods (KMeans, DBSCAN) used for cluster analysis of the company's network traffic, as well as more balanced clustering for complex multidimensional data. A dependence for calculating algorithm weights using the exponential function is also proposed, which allows for a comprehensive approach to integrating the results of different clustering methods. The developed software solution significantly expands the methods of network traffic analysis and provides an effective practical toolkit for increasing the productivity of corporate information systems. The proposed approach can be successfully adapted to solve a wide range of data analysis problems that require processing large volumes of multidimensional information.

Keywords: network traffic, big data, data analysis methods, clustering, collective decisions, ensemble models.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Takyi, K., Bagga, A., & Goopta, P. (2018, August). Clustering techniques for traffic classification: A comprehensive review. In *Proceedings of the 7th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO 2018)* (pp. 224-230). IEEE. https://doi.org/10.1109/ICRITO.2018.8748772
2. Li, J., Zhang, H., Tang, D., & Lin, C. (2021, September). Traffic classification using cluster analysis. In *Proceedings of the International Conference on Computer Information Science and Artificial Intelligence (CISAI 2021)* (pp. 463-467). IEEE. https://doi.org/10.1109/CISAI54367.2021.00094
3. Rodríguez-Rodríguez, J. E., García, V. H. M., & Usaquén, M. A. O. (2018). Corporate networks traffic analysis for knowledge management based on random interactions clustering algorithm. In *Knowledge*

management in organizations (KMO 2018) (pp. 523-536). Springer. https://doi.org/10.1007/978-3-319-95204-8_44

4. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281-297).
5. Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129-137. https://doi.org/10.1109/TIT.1982.1056489
6. Cheeseman, P. C., & Stutz, J. C. (1996). Bayesian classification (AutoClass): Theory and results. In *Advances in knowledge discovery and data mining* (pp. 153-180).
7. Zhang, T., Ramakrishnan, R., & Livny, M. (1997). BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, *1*(2), 141-182. https://doi.org/10.1023/A:1009783824328
8. Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. *ACM SIGMOD Record*, *27*(2), 73-84. https://doi.org/10.1145/276305.276312
9. Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD 1996)* (pp. 226-231).
10. Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *ACM SIGMOD Record*, *28*(2), 49-60. https://doi.org/10.1145/304181.304187
11. Subramani, K., Velkov, A., Ntoutsi, I., Kröger, P., & Kriegel, H.-P. (2011, December). Density-based community detection in social networks. In *Proceedings of the IEEE International Conference on Internet Multimedia Systems Architecture and Application (IMSAA 2011)* (pp. 1-8). IEEE. https://doi.org/10.1109/IMSAA.2011.6156334
12. Zander, S., Nguyen, T., & Armitage, G. (2005, November). Automated traffic classification and application identification using machine learning. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN 2005)* (pp. 250-257). IEEE. https://doi.org/10.1109/LCN.2005.35
13. McGregor, A., Hall, M., Lorier, P., & Brunskill, J. (2004). Flow clustering using machine learning techniques. In *Passive and active network measurement (PAM 2004)* (pp. 205-214). Springer. https://doi.org/10.1007/978-3-540-24668-8_21
14. Erman, J., Mahanti, A., Arlitt, M., Cohen, I., & Williamson, C. (2007). Offline/realtime traffic classification using semi-supervised learning. *Performance Evaluation*, *64*(9-12), 1194-1213. https://doi.org/10.1016/j.peva.2007.06.014
15. Wang, Y., Xiang, Y., Zhang, J., Zhou, W., Wei, G., & Yang, L. T. (2013). Internet traffic classification using constrained clustering. *IEEE Transactions on Parallel and Distributed Systems*, *25*(11), 2932-2943. https://doi.org/10.1109/TPDS.2013.307
16. Wang, P., Lin, S. C., & Luo, M. (2016, June). A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs. In *Proceedings of the IEEE International Conference on Services Computing (SCC 2016)* (pp. 760-765). IEEE. https://doi.org/10.1109/SCC.2016.105

Отримано редакцією журналу / Received: 06.02.26
Прорецензовано / Revised: 21.02.26
Схвалено до друку / Accepted: 25.06.26