MATHEMATICAL AND COMPUTER MODELING

DOI: 10.20535/2411-1031.2018.6.1.153143
УДК 004.942:004.056.5

ДМИТРО ШАРАДКІН

АЛГОРИТМ ПОТОКОВОЇ КЛАСТЕРИЗАЦІЇ ДЛЯ МОНІТОРИНГУ ТА ДІАГНОСТИКИ СТАНУ ТЕХНІЧНИХ СИСТЕМ РЕАЛЬНОГО ЧАСУ

У роботі розглядаються особливості автоматизації процесів моніторингу та діагностики стану технічних систем реального часу, зокрема, сучасних комп'ютерних систем і мереж. Показано, що існуючі методи забезпечують вирішення задач діагностики та моніторингу з суттєвими обмеженнями, що в основному пов'язані з припущенням стаціонарності базових характеристик об'єктів моніторингу. Специфічні особливості систем, що функціонують в режимі реального часу вимагають, щоб алгоритми класифікації та кластеризації, які складають основу сучасних засобів моніторингу та діагностики, функціонували в потоковому режимі, водночас відповідаючи вимогам мінімізації обсягу задіяної пам'яті. Також ці алгоритми мають забезпечити практичну незалежність часу роботи від обсягу даних, що надходять на обробку; підтримувати роботу з кластерами відмінної від сферичної форми в просторі ознак; враховувати необхідність зберігання працездатності в умовах, коли статистичні характеристики потоку даних динамічно змінюються та при невідомій, можливо змінній, кількості кластерів у вибірці; реалізовувати процедури виявлення викидів у вхідних даних. Запропоновано алгоритм виявлення кластерної структури, що заснований на використанні відображення вхідної вибірки даних в спеціальним чином сконструйований сітковий простір, з одночасним врахуванням як характеристик щільності заповнення простору опису об'єктів, так і його метричних властивостей. Проаналізовані властивості запропонованого алгоритму і залежність його внутрішніх характеристик, що обираються при аналізі та від зовнішніх параметрів, що залежать від характеристик вибірки даних. Модифікація для випадку потокового надходження даних дозволяє адаптувати алгоритм, не вимагаючи при цьому додаткових витрат пам'яті для зберігання інформації. Для урахування динамічної зміни характеристик кластерів запропоновано використання функції забування та розглянуті можливі способи її опису. Досліджений вплив різновидів функції забування на характеристики працездатності запропонованого алгоритму. Відносна простота алгоритму і семантична прозорість його параметрів дозволяє виконувати налаштування алгоритму для різних областей застосування, включаючи задачі виявлення і запобігання інцидентам у сфері інформаційної безпеки.

Ключові слова: моніторинг та діагностика стану систем реального часу, виявлення інцидентів інформаційної безпеки, машинне навчання, класифікація, кластеризація, потокова обробка даних, динамічна зміна форм та положення кластерів.

Постановка проблеми. У зв'язку з все більш широким використанням систем реального часу виникає гостра потреба в автоматизації процесів їх моніторингу та діагностики (МтД). В загальному випадку задача моніторингу визначається як процес відстеження нормальної (типової) поведінки елементів системи і фіксація моментів, коли поведінка істотно змінюється, а завдання діагностики – як визначення, який саме варіант нетипової (аномальної) поведінки має місце. Прикладом систем описаного класу є, зокрема, комп'ютерні системи, в яких процеси МтД проявляються у вигляді завдань виявлення і запобігання інцидентам інформаційної безпеки [1]. В таких системах як ознаки, що аналізуються, використовуються інтенсивність і обсяг трафіку в мережі; опис регламентованих і заборонених дій користувача, що відбулися; кількість запитів до бази

даних, інформація про відмову елементів обладнання або зміни рівня його поточної завантаженості; фіксація запитів з певних IP-адресах [2]. Якщо при відносно низькому навантаженні систем завдання МтД успішно виконувалися в ручному режимі, то сьогодні, в зв'язку з постійним ускладненням та масштабуванням систем реального часу все гостріше постає питання про автоматизацію зазначених процесів [3].

Аналіз останніх досліджень і публікацій. З математичної точки зору МтД зводяться до задач предикативного аналізу, тобто виявлення залежності значень деякого узагальненого показника ("є вторгнення"/"немає вторгнення", "режим роботи обладнання штатний"/ "обладнання працює в позаштатному режимі" тощо) від значень ознак які описують поточний стан об'єкту моніторингу. Застосуванню методів і засобів машинного навчання для вирішення даного класу задач присвячено значний ряд робіт [1], [3] - [8]. Однак, пряме перенесення методів, навіть таких, які добре зарекомендували себе в інших предметних областях, на завдання МтД систем реального часу пов'язане з рядом труднощів, що зумовлені особливостями їх функціонування. Розглянемо основні з них [9] - [14].

1. Більшість методів класифікації і кластерного аналізу, що лежать в основі процедур МтД, засновані на *пакетному обробленні даних*. Передбачається, що набір даних одноразово подається на вхід алгоритму. В результаті виробляється відповідне рішення, тобто поділ множини станів об'єкта на класи (кластеризація) або віднесення опису стану об'єкта до одного з визначених класів (класифікація). Однак, при МтД систем реального часу дані надходять безперервно, але поступово. Відповідно, алгоритмом повинно вироблятися рішення безперервно в процесі надходження даних. Такий режим носить назву *потокового оброблення даних* [9], [10].

2. Потік даних, що постійно поповнюється, може досягти сумарного обсягу, при якому час його оброблення стає неприпустимо великим, а необхідна для зберігання пам'ять виходить за межі виділеного простору. Для уникнення цього необхідно, щоб алгоритм МтД допускав або можливість роботи тільки з обмеженим обсягом найбільш пізніх на кожен поточний момент часу даних (*алгоритми зі забуванням*), або виконував агрегацію раніше накопичених даних, формуючи їх синопсис для подальшої роботи з ним (*алгоритми зі скетч-представленням інформації*). Ідеальний потоковий алгоритм з метою економії пам'яті повинен виконувати виключно одноразову обробку порції даних що надійшла, після чого вивільняти зайняту ними пам'ять (*однопрохідні алгоритми*) [11].

3. Існуючі пакетні і однопрохідні потокові алгоритми виходять із припущення про стаціонарність моделі даних, тобто припускається, що параметри кластерів, які описують різні стани об'єкту, залишаються незмінними. Однак, якщо функціонування алгоритму розтягнуто в часі, то з великою імовірністю можуть мати місце динамічні зміни властивостей об'єктів, які класифікуються (кластеризуються). Прикладами є технічні системи, обладнання в яких схильне до поступового спрацювання (деградації). Інший приклад – процес нарощування кількості запитів, що обслуговуються в результаті зростання популярності деякого веб-ресурсу. МтД, що працюють у режимі реального часу повинні враховувати, що значення ознак об'єктів кластерів, які вважалися типовими раніше, можуть змінюватися у майбутньому, тобто раніше виявлені кластери можуть *дрейфувати простором ознак* [12], [13].

4. Більшість відомих алгоритмів припускають, що наявні кластери станів об'єкту мають "правильні" форми в просторі значень ознак, лінійно роздільні, або можуть бути описані геометрично правильними фігурами (наприклад, сферами, еліпсоїдами). Однак, для задач технічної діагностики це припущення часто порушується. Наприклад, в системах МтД комп'ютерних мереж розподіл часу з'єднань зазвичай нерегулярний. В такому випадку кластери, які породжені в просторі значень ознак, можуть мати довільні форми [14]. Аналогічно, при спостереженні за просторово-географічним поширенням явищ в соціальних мережах, розташування областей з подібними параметрами може бути будь-якої форми.

5. В інтенсивному потоці даних неминучі поодинокі події, значення ознак яких істотно відрізняються від звичайних, але кількість яких нескінченно мала (так звані *викиди*).

Такі дані можуть з'являтися в результаті впливу випадкових факторів, що не повторюються, таких як електромагнітні та інші індустріальні завади, тимчасові відмови датчиків. Алгоритм МтД повинен вміти розпізнавати викиди з метою мінімізації кількості помилкових спрацьовувань.

6. У класичній, стаціонарній задачі класифікації існують чіткі межі між етапом побудови кластерів (кластеризації або попередньої ручної розмітки наявного набору даних), етапом побудови класифікатора і етапом виконання діагностичної процедури (власне класифікації). При роботі в режимі потокового аналізу з динамічною зміною значень ознак кластерів описані вище фактори руйнують ці межі. Етапи ітераційно повторюються, на кожній ітерації обробляється нова порція даних та виробляються рішення на поточний момент часу та адаптуючись до змін статистичних характеристик потоку даних.

В основі автоматизованих процесів МтД лежать алгоритми кластеризації, тобто розбиття всієї множини даних на семантично пов'язані підмножини. Існуючі алгоритми кластеризації поділяються на дві основні групи. Алгоритми першої групи, такі як *K*-means, *K*-медоідов та їх модифікації [15], засновані на аналізі подібності між екземплярами описів станів об'єкта та іншими об'єктами з цієї-ж вибірки в просторі значень ознак. Надалі назвемо екземпляр опису стану об'єкта – *точкою вибірки (в просторі значень ознак)*. У вказаній групі алгоритмів кластери являють собою сукупність точок, що характеризуються відносно малою відстанню до центру їх кластера і відносно великою відстанню до центрів інших кластерів. Алгоритми цієї групи відрізняються обчислювальною простотою, але водночас вимагають підвищених витрат на зберігання інформації, а також попереднього визначення на кількості кластерів. Оскільки кожна точка в просторі ознак призначається в кластер, центр якого розташований найближче (в обраній метриці), алгоритми цієї групи не здатні виявляти кластери з формою, істотно відмінною від сферичної.

Алгоритми другої групи, зокрема DBSCAN, SUBCLU, OPTICS та їх модифікації [16], засновані на аналізі щільності розподілу точок вибірки в просторі ознак. В них кластер визначається як множина точок вибірки, які "тяжіють" до одного локального максимуму їх розподілу в просторі ознак. Такий підхід дозволяє обробляти кластери форм, відмінних від сферичної, автоматично визначати кількість кластерів у вибірці, але є істотно більш обчислювально складний, ніж алгоритми першої групи.

Оскільки алгоритми обох груп досить вимогливі до обсягу пам'яті, їх адаптація для випадку потокових даних в основному проводилася з передумовою скорочення цього обсягу шляхом пакетування вихідного набору даних, тобто розбиття на кілька пакетів, послідовної обробки кожного пакету окремо, і подальшої агрегації отриманих результатів. Однак, такий підхід, вирішуючи завдання скорочення обсягу пам'яті, не спроможний врахувати зазначений вище фактор мінливості характеристик потоку даних, що надходять.

Метою статті є розробка алгоритму кластеризації, що може використовуватися для моніторингу та діагностики технічних систем в режимі реального часу. Алгоритм розроблений з урахуванням усіх зазначених вище вимог, а саме – *роботи в потоковому режимі; мінімізації обсягу пам'яті, що використовується; забезпечення практичної незалежності від обсягу даних, що надходять; здатність працювати в умовах, коли статистичні характеристики потоку даних динамічно змінюються; заздалегідь невідомої кількості кластерів в вибірці; здатність роботи за умов складної, не сферичної форми кластерів в просторі ознак.* Алгоритм виходить з припущення, що у вибірці об'єктивно об'єктивно існує деяка, але заздалегідь невідома кількість кластерів. При цьому самі кластери є областями в просторі значень їх описів, що характеризуються підвищеною в порівнянні з іншими областями цього простору щільністю, а метрична відстань між кластерами істотно більша за відстань між точками всередині кластера. Продемонструвати таке уявлення найпростіше для двовимірного випадку (див. рис.1). У наведеному прикладі дані мають досить чітку, інтуїтивно зрозумілу кластерну структуру, в якій точки всередині кластера розташовані близько між собою, а між кластерами присутні області, в яких щільність точок мінімальна (або дорівнює нулю).

Алгоритм представляється наступною послідовністю кроків.

Крок 1. Відображення простору значень ознак в сітковий простір грідів. Зниження обчислювальної складності алгоритму кластеризації досягається за рахунок переходу від обробки даних, що представлені у вхідному просторі значень ознак до обробки елементів сіткової структури, в якій кількість елементів сітки мала. Передбачається, що точки вибірки, які потрапили в одну клітинку сітки (*грід*), з високою імовірністю належать одному кластеру. Вказаний підхід дозволяє водночас досягти підвищення швидкість виконання, та обсягу масиву даних, що оброблюється, виділення кластерів складної форми і незалежності від порядку надходження даних[17].

Перехід з початкового простору значень ознак в сітковий простір кінцевої потужності виконується наступним чином. Зобразимо вибірку у вигляді множини точок в N - вимірному просторі, тобто $X_i \in X \subset U^N$, $X_i = (x_{i1}, x_{i2}, .., x_{ij}, .., x_{iN}), x_{ij} \in (0,1]$. . Нехай також є множина кінцевих послідовностей $\Omega_s = \left\{0, \frac{1}{k}, \frac{2}{k}, ..., \frac{k-1}{k}, 1\right\}, s \in \{0,1,..., N\}$ Побудуємо сітку Ω у вигляді декартового добутку $\Omega = \Omega_1 \times \Omega_2 \times ... \times \Omega_N$. Задаємо відображення $U^N \to \Omega$ при якому точка $X_i \in U^N$ відображається в точку $Q_l = (q_{l1}, q_{l2}, ..., q_{lN}) \in \Omega$ тоді і тільки тоді, коли $q_{(i-1)j} \le x_{ij} < q_{ij}$. Побудовану таким чином точку з множини Ω називатимемо *грідом*.

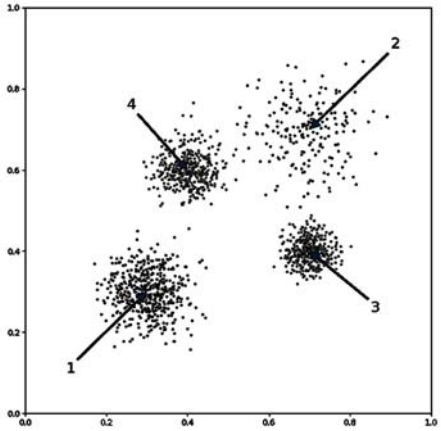

Рисунок 1 – Приклад відображення кластерної структури. Стрілки вказують на центри кластерів

Крок 2. Визначення щільності кожного гріду. Кількість точок вибірки X, що відображаються в точку Q_l називатимемо *щільністю гріду* і позначатимемо як L_l .

Оскільки алгоритм надалі оперує виключно з координатами і щільностями грідів, забезпечується вимога не зростання обчислювальної складності наступних кроків-алгоритму від кількості точок у вибірці.

Крок 3. Віднесення гріду до кластеру. Алгоритм DBCSAN [16] заснований на визначенні точок вибірки з найвищою щільністю оточення, призначенням цих точок центрами кластерів і поступовому залученні до кластеру нових точок вибірки, які потрапляють в область “близьких” точок. При зростанні кількості точок вибірки обсяг обчислювальних витрат стрімко зростає. На відміну від цього алгоритм що пропонується відносить до кластеру не кожну точку окремо, а всі точки, що на попередньому кроці відобразились до одного гріду. Таким чином, при появі нових точок повторний аналіз структури необхідно проводити тільки в разі, коли щільність гріду істотно змінюється. Крім того, точки, що відобразилися в гріди з екстремально малими значеннями щільності, можуть трактуватися як викиди, а відповідні гріди – виключатися з подальшого аналізу, що додатково зменшує обчислювальні витрати. Назвемо два гріда Q_l і Q_m *безпосередньо*

зв'язаними, якщо існує хоча б одне число і таке, що $|q_{li} - q_{mi}| = \frac{1}{k}$. Якщо $Q_{l_i}, Q_{l_{i+1}}$ і $Q_{l_{i+1}}, Q_{l_{i+2}}$ дві пари безпосередньо зв'язаних між собою грідів, назвемо гріди Q_{l_i} і $Q_{l_{i+2}}$ такими, що мають другий рівень зв'язності. Назвемо два гріди Q_l та Q_m *зв'язаними*, якщо існує така послідовність грідів $Q_{l_1}, Q_{l_2}, .., Q_{l_t}$, що для кожного $i \in \{1, .., t-1\}$ пара грідів $Q_{l_i}, Q_{l_{i+1}}$ є безпосередньо зв'язаною. Логіка віднесення гріду до кластеру заснована на порівнянні щільностей грідів і їх метричної відстані в просторі $\mathcal{K}$ та описана в вигляді квазіалгоритму (див. ліст. 1).

Лістинг 1. *Процедура віднесення гріду до кластеру*

0. $i = 1$

1. $Z = \emptyset$

2.*Взяти* Q_l, *що ще не віднесені ні до одного з кластерів.*

3.*Виділити* $\widetilde{Q}_l = \{Q_{l_1}, Q_{l_2}, .., Q_{l_t}\}$ *множину (безпосередньо) зв'язаних грідів.*

4. Якщо: *існує* $Q_{l_p} \in \widetilde{Q}_l$ *такий, що* $Q_{l_p} = \underset{Q_l}{arg\,min}(L_{l_p})$ *та* $L_{l_p} > L_l$

То:

Якщо: Q_{l_p} *належить до кластеру* $K_v, v \in [1, ..., i]$

То: *Всі точки з* Z *віднести до кластеру* K_v,

Перейти до кроку 1.

Інакше: $Z = Z \cup Q_l$

$Q_l = Q_{l_p}$

Перейти до кроку 3.

Інакше*:*

Якщо: $max\, D(Q_l, Q_{l_p}) < \lambda$

То: $\widetilde{Q}_l = \widetilde{Q}_l \cup \widetilde{Q}_{l1} \cup \widetilde{Q}_{l2} \cup .... \cup \widetilde{Q}_{lt}$

Перейти до кроку 4.

Інакше: *Віднести всі точки з* Z *до кластеру* K_i

$i = i + 1$

Перейти до кроку 1.

Послідовно перебираючи гріди, які раніше не були віднесені до жодного з кластерів, алгоритм шукає найближчий до нього грід такий, щільність якого більше, ніж щільність вхідного гріду і зв'язує їх у ланцюжок *Z* переглянутих в даному циклі грідів. Якщо в ході роботи алгоритм знаходить грід, який вже був раніше віднесений до одного з кластерів, то всі гріди, що складають ланцюжок *Z* відносяться до того самого кластеру і алгоритм починає наступний цикл. В іншому випадку алгоритм перевіряє метричну відстань між грідом, що аналізується, та грідами, що зв'язані з ним. Якщо ця відстань менше заздалегідь заданої величини λ (*відстань відсікання*), то множина зв'язаних грідів $\widetilde{Q}_l$ розширюється і процедура пошуку триває. Якщо ж ця відстань виявляється більше λ, то алгоритм динамічно породжує новий кластер, до якого відносить всі елементи, які складають ланцюжок Z. На рис. 2 показані два згаданих випадки. Перший, коли ланцюжок був розпочатий з гріду Q_1 і закінчився в гріді Q_9, що ініціювало створення нового кластеру. І другий, коли ланцюжок розпочатий з гріду R_1 і побудований до раніше обробленого гріда Q_5. Величина λ є одним з зовнішніх параметрів алгоритму і не дозволяє всім грідам бути віднесеним до одного кластеру. Її вибір здійснюється експертним шляхом, виходячи з міркувань неможливості дуже близького розташування кластерів у просторі значень ознак.

Крок 4. Перевірка кластерної структури на зв'язність. Корекція кластерної структури. Оскільки точки, з яких починається побудова ланцюжків, надходять на вхід алгоритму в довільному порядку, можливий спеціальний випадок. Припустимо є два безпосередньо зв'язаних гріда, що знаходяться на межі кластерів і мають однакові значення щільності. В результаті випадкового вибору один з грідів виявився віднесеним до кластеру K_2, а другий – до кластеру K_3 (див. рис. 2). З огляду на те, що $\lambda < b$, точки кластера K_2 не були віднесені до кластеру K_3. Ця ситуація принципово відрізняється від ситуації для точок, що потрапили в кластери K_1 і K_3, де хоча $a < \lambda$, але між точками кластерів є області з грідами нульової щільності. Інтуїтивне трактування кластерної структури передбачає, що точки, попередньо віднесені до кластерів K_3 і K_2 повинні розглядатися як точки єдиного кластеру з мультимодальним розподілом об'єктів. Тому робота алгоритму завершується перевіркою наявного розбиття грідів і об'єднанням таких кластерів, точки яких взаємно пов'язані (в даному випадку – кластерів K_2 і K_3).

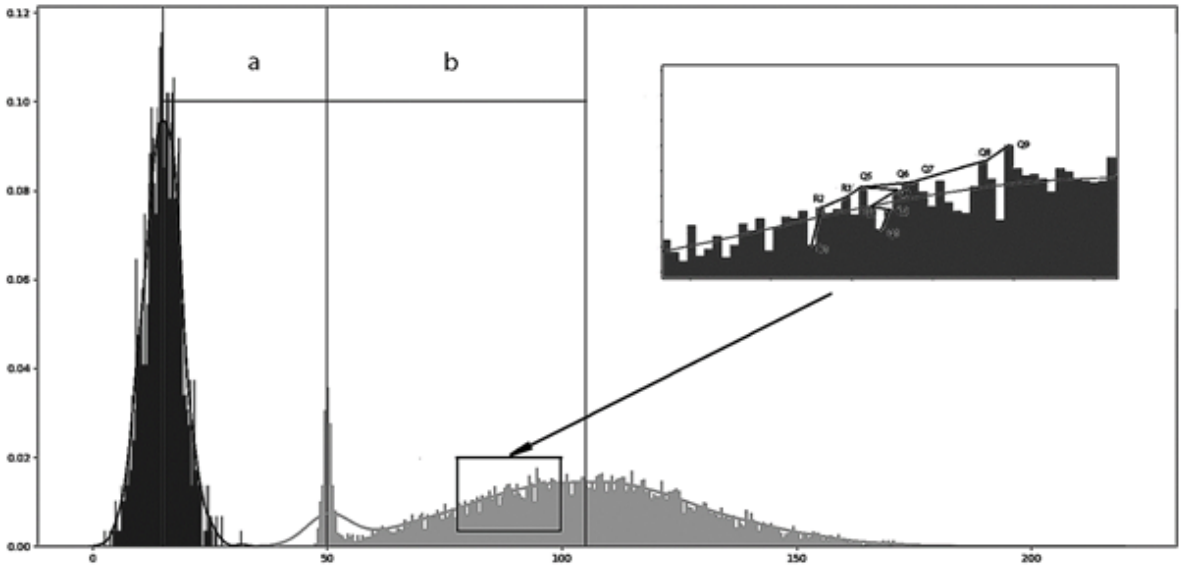

Рисунок 2 – Приклад використання алгоритму

Примітка. Для випадку кластерів, що змінюються динамічно, мультимодальність може бути ознакою того, що кластер має тенденцію до розпаду на декілька складових. Ознакою того, що кластер дійсно розпався є наявність областей простору з нульовою щільністю грідів між утвореними частинами та збереженням відношення зв'язності тільки всередині утворених множин.

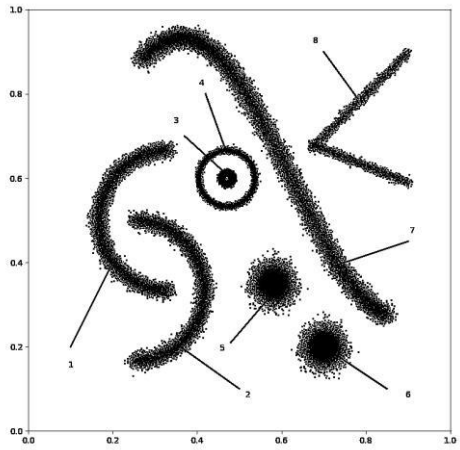

Рисунок 3 – Тестова кластерна структура

Експериментальне дослідження базового алгоритму. Для аналізу працездатності алгоритму згенеровано вибірку даних в двовимірному просторі, що включає 8 кластерів різної просторової структури (див. рис. 3). До вибірки включено вибірки “make_moons” (на рисунку–“1” та “2”) і “make_circles” (на рисунку–“3” та “4”), що стали стандартом “de-facto” при перевірці працездатності алгоритмів кластеризації [18], дві нормально розподілені сферичні вибірки (на рисунку – “5” та “6”), мультимодальна вибірка, що згенерована фрагментом синусоїди з накладанням гаусового шуму, в якій щільність точок вибірки збільшується на кінцях фрагмента і зменшується в його середині (на рисунку – “7”) і вибірка, що представляє кластер специфічної форми (на рисунку – “8”), який доволі складно піддається кластеризації багатьма відомими алгоритмами. Всього згенеровано 74000 точок. Одночасна присутність в тестовій вибірці лінійно сепарабельних кластерів, кластерів сферичної форми, вкладених кластерів і кластерів довільної структури суттєво ускладнюють, а в деяких випадках взагалі унеможливлюють застосування більшості відомих алгоритмів кластеризації.

На рис. 4 наведено результати виконання процедур кластеризації за допомогою чотирьох програм бібліотеки Scikit-Learn, що є на сьогоднішній день найбільш поширеним засобам реалізації алгоритмів машинного навчання в середовищі Python. На рисунках різні кластери відображені різними градаціями сірого кольору. Досить складний тестовий набір даних був вдало кластеризований тільки програмою, що реалізує алгоритм DBSCAN. Інші програми або виконували невиправдане злиття різних кластерів, або помилялися, розбиваючи єдиний кластер на декілька. Застосування запропонованого в роботі алгоритму до тестового набору даних дозволило кластеризувати його на 8 кластерів, в точній відповідності до структури кластерів в просторі ознак.

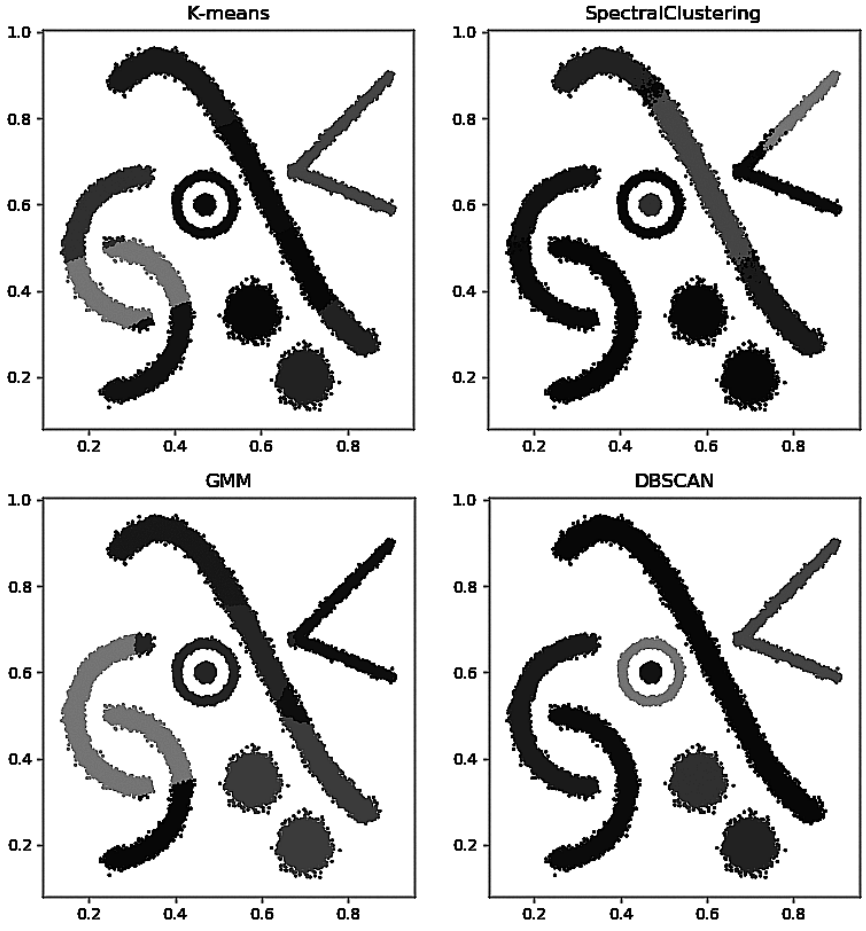

Рисунок 4 – Результати кластеризації за допомогою програм бібліотеки Scikit-Learn

У наведеній стаціонарної постановці алгоритм має два зовнішніх параметри – кількість грідів та відстань відсікання λ. Доцільно вивчити, як змінюється точність роботи алгоритму зі зміною цих параметрів. Для цього значення параметра λ будемо задавати адаптивно, у вигляді частки C від відстані максимально віддалених у вибірці грідів: $\lambda = \max_{\forall X_i X_j \in X} D(X_i, X_j) / C$. В якості функції D оберемо звичайну евклідову відстань в просторі U^N. Таким чином, для вибірок, в яких кластери розташовані близько між собою значення λ зменшується, а для вибірок, в яких кластери розкидані по всьому простору – збільшується. Ця властивість особливо придатна при динамічній зміні характеристик кластерів, коли взаємне розташування істотно змінюється в часі.

Робота алгоритму оцінювалася за кількістю кластерів, які йому вдалося виділити в тестовій вибірці. На наведеній на рис. 5 діаграмі показано залежність кількості виділених кластерів від зазначених параметрів. Видно, що присутня яскраво означена область значень параметрів C і K, при яких алгоритм забезпечує абсолютно точне розбивання простору на кластери. Подальше збільшення значень параметрів C і K виявляється недоцільним, так як при цьому виявляється явище перенавчання.

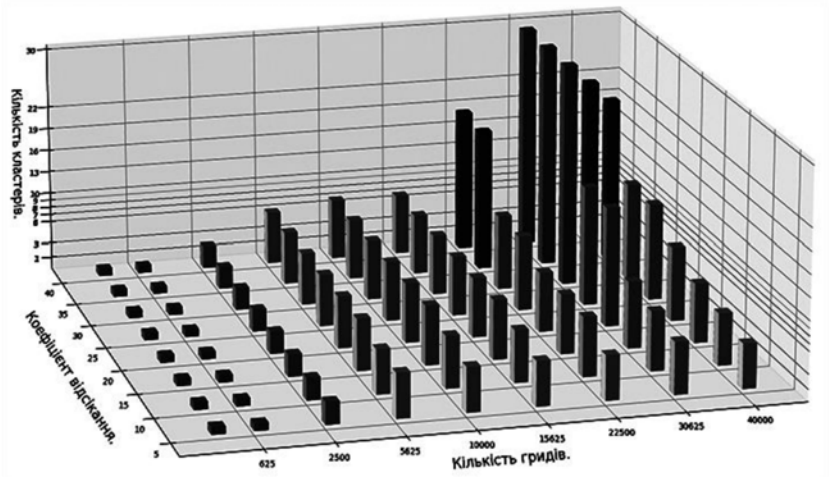

Рисунок 5 – Залежність кількості виявлених кластерів від параметрів алгоритму

На рис. 6 показані залежності часу роботи алгоритму від значень зазначених параметрів. У подальших експериментах значення параметрів приймалися рівними C=25, K=100*100.

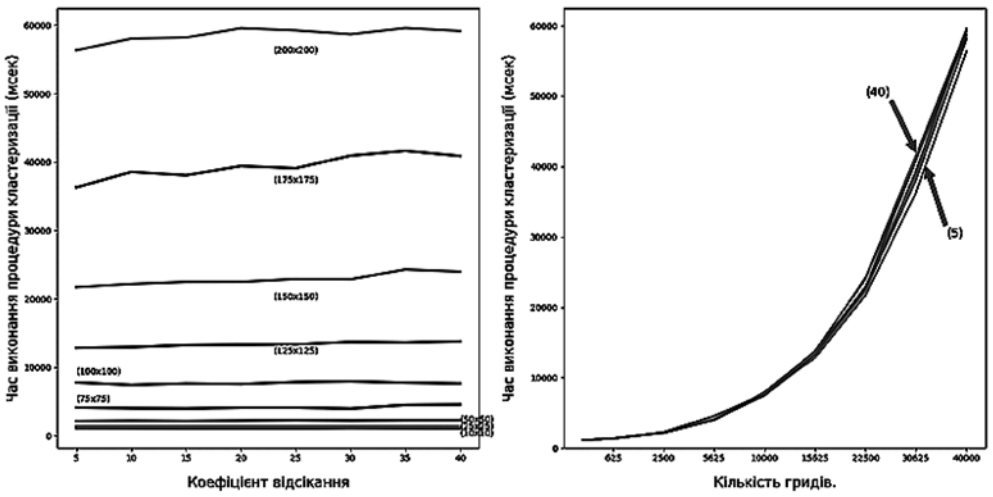

Рисунок 6 – Час виконання процедури кластеризації в залежності від параметрів алгоритму. Зліва – від значення коефіцієнту відсікання C. У дужках позначено кількість грідів. Справа – від кількості грідів K. Позначені лише крайні значення коефіцієнту відсікання

Розширення алгоритму для роботи в потоковому режимі. При роботі в потоковому режимі передбачається, що дані надходять послідовно, у вигляді підмножин множини *X*. Якщо в задачах, пов'язаних з оптимізацією використання пам'яті, кількість елементів в підмножинах може обиратися довільною, то в задачах МтД в реальному часі вона визначається частотою надходження даних. У загальному випадку ця кількість може бути довільною, та з метою зменшення обчислювальної складності алгоритму приймемо її фіксованою. А саме, значення *W* визначається з умови, що час оброблення попередніх *W* об'єктів повинен бути не більшим, ніж час надходження наступних *W* об'єктів. Оскільки час, який алгоритм витрачає на оброблення *W* об'єктів, залежить від обраної кількості грідів, а інтенсивність надходження об'єктів вхідного потоку передбачається відомою, значення *W* обирається так, щоб завадити неконтрольованому зростанню обсягу необроблених даних вхідного потоку.

З'ясуємо залежність часу, що витрачається на виявлення коректної кластерної структури і часу обробки всієї вибірки від значень параметру *W*. Для цього взявши за основу алгоритм, описаний вище, внесемо в нього відповідні зміни. Поява кожного нового пакета даних призводить до необхідності перерахунку значень щільності грідів L_l. У свою чергу це може призводити до зміни зв'язності грідів. Простежити відповідні зміни можна, відобразивши кількість знайдених кластерів в процесі появі нових даних. З наведеної на рис. 7 інформації випливає, що кількість кроків, які виконує алгоритм для виявлення коректної кластерної структури в тестовому наборі даних приблизно однакова і суттєво не залежить від розміру пакету *W*. Однак, при цьому змінюється кількість циклів, які повинен виконати алгоритм – чим більше значення *W*, тим менше циклів перерахунку виконується і тим менше часу, який витрачається до того, як буде визначена коректна кластерна структура (див. рис. 8).

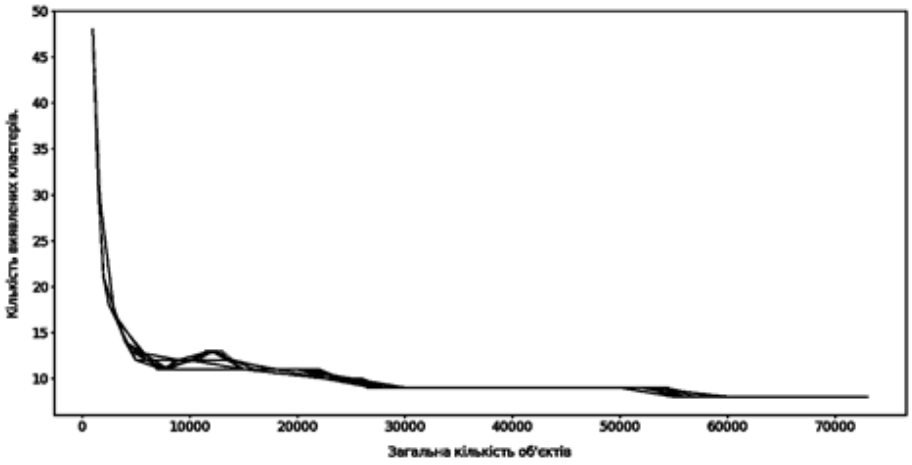

Рисунок 7 – Відображення процесу виявлення кластерів в залежності від кількості наданих об'єктів

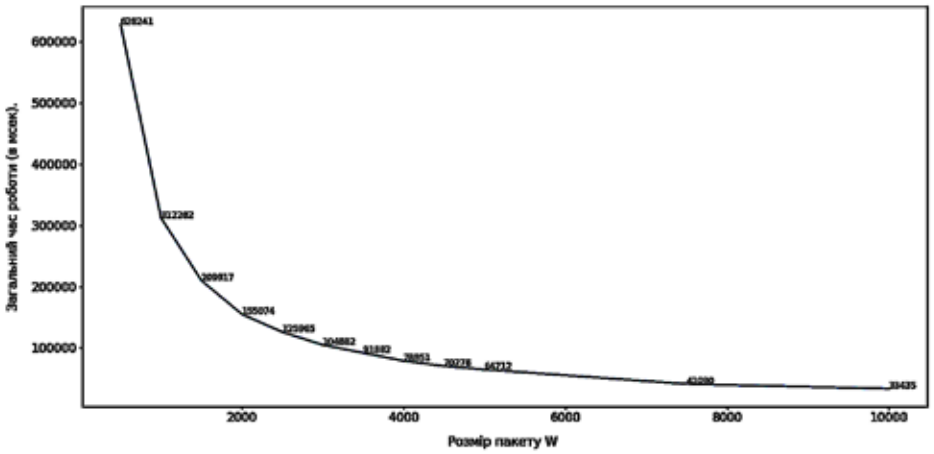

Рисунок 8 – Залежність часу роботи алгоритму від розміру пакету обробки *W*

Застосування алгоритму для кластерів з характеристиками, що динамічно змінюються. При аналізі кластерів з характеристиками, що динамічно змінюються передбачається, що чим менше часу пройшло від моменту надходження опису стану об'єкту, тим більш важлива інформація про нього. І навпаки, чим "старіші" дані, тим менш важливу інформацію для виявлення поточної кластерної структури вони несуть. Для урахування цього на кожному циклі алгоритм повинен мати властивість поступового "забування" кластерної структури, яка мала місце в минулому. Граничним випадком можна вважати випадок розгляду лише тих точок, які надійшли в останньому пакеті даних [14]. Однак, як випливає з проведеного вище аналізу, стабільна робота може досягатися тільки при досить великій кількості точок, якими оперує алгоритм. Оскільки збільшення значення W неможливо виходячи з його узгодженості з інтенсивністю надходження даних, пропонується застосувати підхід, що приписує кожній точці деякий коефіцієнт забування, що семантично враховує поточну актуальність даних.

Якщо зафіксувати поточний момент t_T , то математично коефіцієнт забування для кожної точки $t_i, i \in \{0, T\}$ задається монотонною, зворотньо-пропорційною до величини інтервалу $\Delta t_i = t_T - t_i$ функцією, яка приймає на нових даних максимальне значення рівне 1. Бажано також, що б така функція не вимагала громіздкого перерахунку, яке сповільнює роботу алгоритму.

В якості функцій врахування ефекту «забування», розглядають наступні альтернативи [19]:

Функція з експоненціальним забуванням:

$$f(\Delta t_i) = \begin{cases} 0, & \Delta t_i > w; \\ 1 - \dfrac{\Delta t_i}{1 + e^{-\lambda\left(\Delta t_i - \frac{w}{2}\right)}}, & 0 < \Delta t_i \le w; \\ 1, & \Delta t_i = 0. \end{cases}$$

Функція з рівномірним забуванням:

$$f(\Delta t_i) = \begin{cases} 0, & \Delta t_i > w; \\ 1 - \dfrac{\Delta t_i}{w - 1}, & 0 < \Delta t_i \le w; \\ 1, & \Delta t_i = 0. \end{cases}$$

Функція зі забуванням типу «обмежене вікно»:

$$f(\Delta t_i) = \begin{cases} 0, & \Delta t_i > w; \\ 1, & \Delta t_i \le w. \end{cases}$$

Для виконання зазначеної перевірки згенеровано три тестові вибірки даних, які виявляють п'ять основних типів змін в структурі, а саме: утворення нових кластерів; злиття кластерів; розщеплення або дроблення кластерів; зникнення кластерів; дрейф кластерів.

Зміну структури кластерів в просторі значень ознак зі зміною часу для цих вибірок наведено на рис. 9. Вибірка a) відповідає випадку послідовної зміни форми кластерів, вибірка b) – випадку, коли кластери одночасно змінюють дислокацію і кількість (з'являються нові), вибірка c) – більш загальному випадку постійної зміні положення кластерів з послідовним зближенням, злиттям, роз'єднанням, злиття інших кластерів та їх остаточним роз'єднанням. Експерименти, що були проведені з тестовими наборами даних виявили, що для різних функцій забування істотна різниця в роботі виявлялася тільки в даних набору c).

На рис. 10 показано, як при розмірі пакету W=500, використанні різних функцій забування і різних значеннях параметра w в динаміці змінювалося кількість виявлених

кластерів. Для тестових наборів a) та b) значущі відмінності в залежності від виду функції забування виявлені не були. На рис.11 показані результати роботи алгоритму на цих вибірках при $W = 500$.

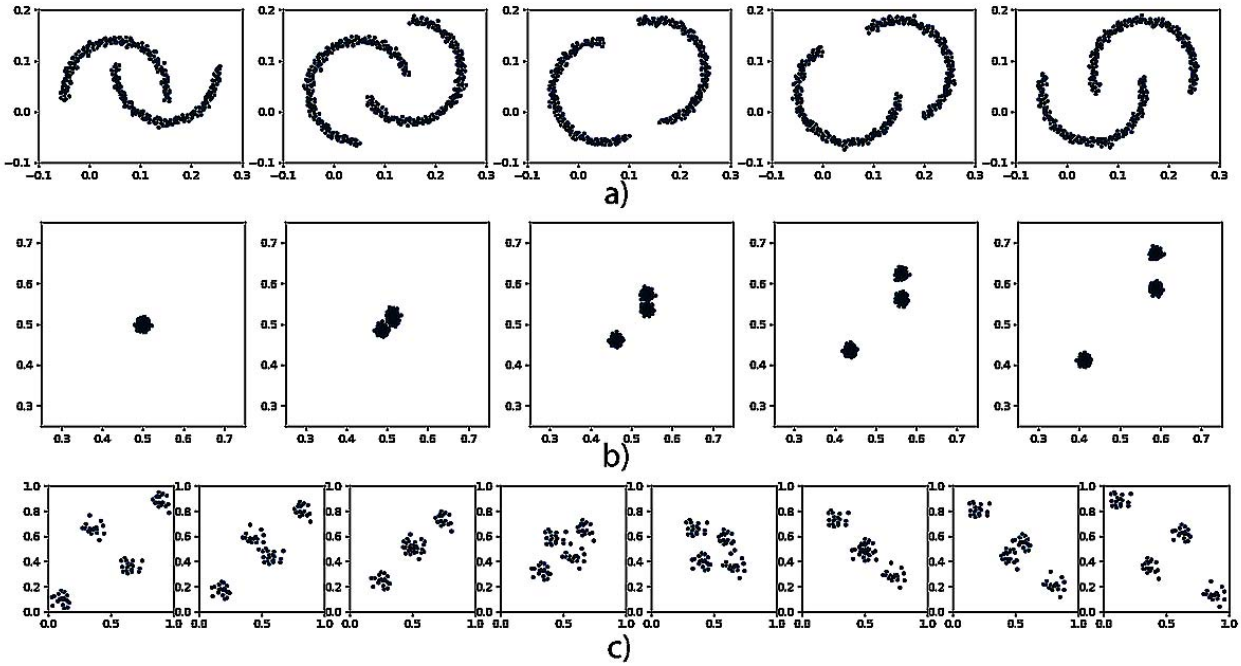

Рисунок 9 – Тестові приклади кластерних структур зі зміною параметрів. (Пояснення в тексті).

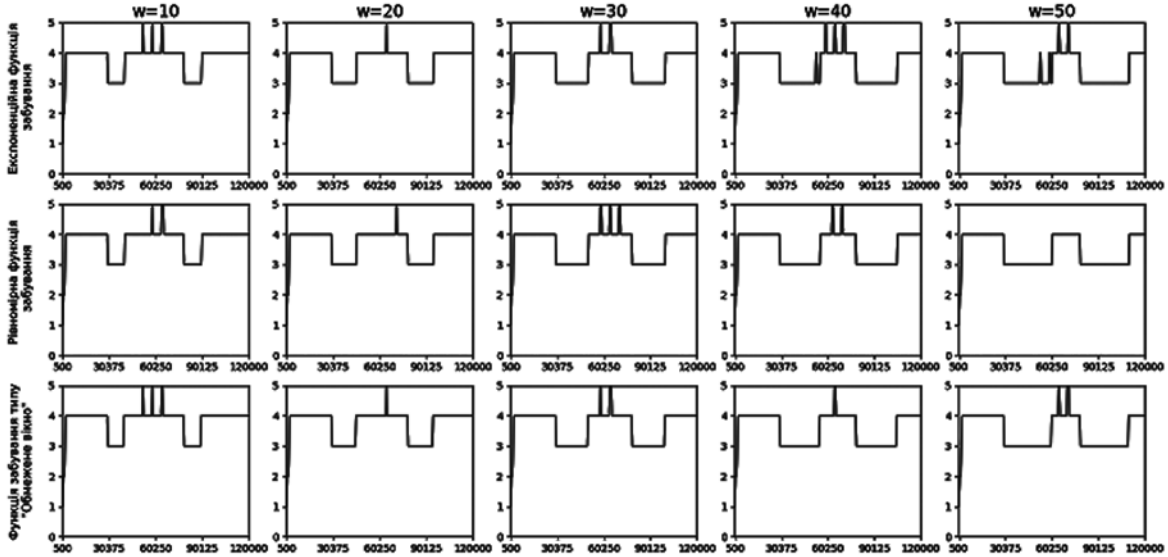

Рисунок 10 – Робота алгоритму для вибірки a) при використанні різних функцій забування та значень параметру w. Значення параметру W =500

Проведене дослідження показало, що вибір значень параметрів w і W не може бути довільним. При великих значеннях параметра W (наприклад, для досліджуваних вибірок при W>2000) і великих значеннях w спостерігається істотне збільшення часу, що витрачався на коректне розпізнавання кластерної структури, що може виявитися неприйнятним для систем реального часу.

Реалізація алгоритму та його модифікацій, а також дослідження їх працездатності проводилася на мові Python (версія 3.6.2, 64-bit, для Windows, збірка Anaconda), в середовищі розробки Spyder з використанням бібліотек NumPy, SciPy, Matplotlib, Scikit-Learn.

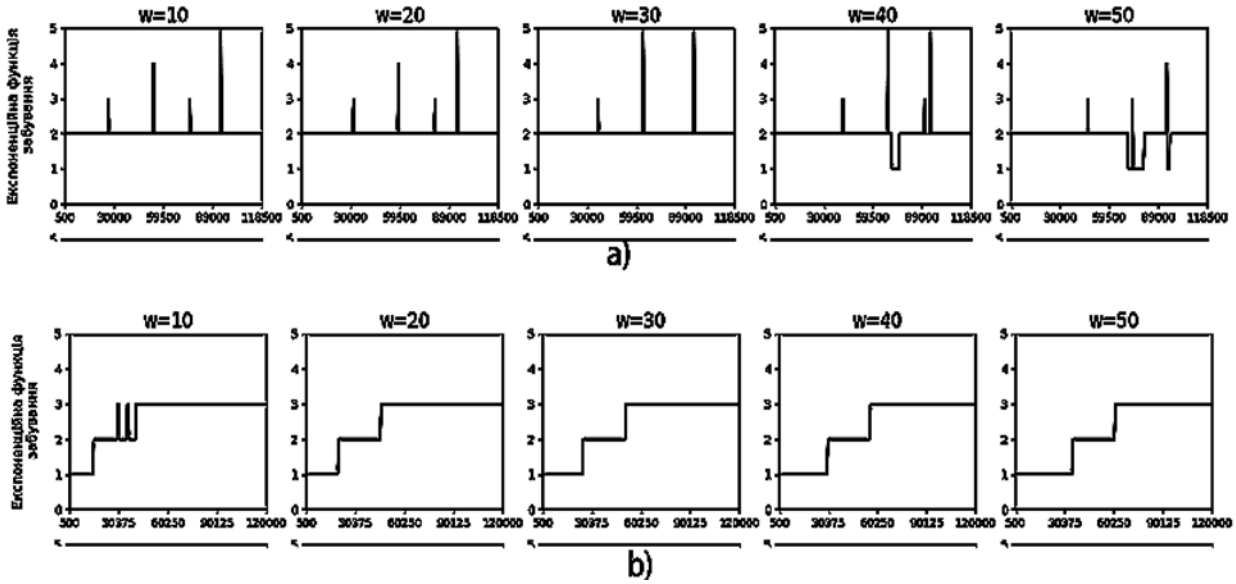

Рисунок 11 – Робота алгоритму для тестових вибірок a) та b). Значення параметру W =500

Висновки. В роботі запропонован алгоритм кластеризації, що може використовуватися для моніторингу технічних систем в режимі реального часу. Алгоритм показав результати, що не поступаються відомим алгоритмам при роботі в умовах стаціонарності та статичності вхідних даних, та на відміну від них здатен виконувати кластеризацію за умов потокового надходження даних та одночасної зміни форм та положення кластерів, що визначаються. Відносно невисока кількість помилок, що допускаються при цьому показує його стійкість до помилок першого та другого роду.

Перспективи подальших досліджень може бути направлене на виявлення залежності зовнішніх параметрів алгоритму для різних типів даних, швидкості їх надходження та поведінки динамічних кластерів у часі, а також з аналізом та вибором метрик подібності в багатовимірних просторах опису об’єктів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] О. И.Шелухин, Д. Ж.Сакалема, и А. С.Филинова. *Обнаружение вторжений в компьютерные сети(сетевые аномалии)*. Москва, Российская Федерация: Горячаялиния-Телеком, 2013.

[2] Э. Уилсон, Мониторинг и анализ сетей. *Методы выявления неисправностей*, Москва, Российская Федерация: ЛОРИ, 2002.

[3] M. Collins, *Network Security Through Data Analysis*. Sebastopol, CA: O'Reilly Media, Inc., 2014.

[4] N. Adams, and N. Hearth, *Data Analysis for Network Cyber-Security*. Danvers, MA: Imperial College Press, 2014.

[5] M. Munia, S. Samrose, P. Dey, A. Annesha, and S. Hasan, “Network Intrusion Detection using Selected Data Mining Approaches: A Review”, *International Journal of Computer Applications*, vol. 132, no.13, pp.10-17, December 2015. doi: 10.5120/ijca2015907572.

[6] А. А. Браницкий, и И. В.Котенко, “Анализ и классификация методов обнаружения сетевых атак”, *Труды СПИИРАН*, Вып. 2 (45), с. 207-243, 2016. doi: 10.15622/sp.45.13.

[7] Т. И. Булдакова, и А. Ш. Джалолов, “Выбор технологий DataMining для систем обнаружения вторжений в корпоративную сеть”, *Инженерный журнал: наука и инновации*, вып. 11, с. 1-14, 2013. doi: 10.18698/2308-6033-2013-11-987.

[8] П. М. Шипулин, и А. Н. Шниперов, “О возможности применения методов DataMining для анализа распределенных атак в сети”, *Актуальные проблемы авиации и комонавтики*, т. 1, с.782-784, 2016.
[9] M. Ghesmoune, M. Lebbah, and H. Azzag, “State-of-the-art on Clustering Data Stream”, *Big Data Analytics*, vol. 1, no. 13, pp.1-27, 2016. doi:10.1186/s41044-016-0011-3.
[10] C. C. Aggarwal, Data Streams: Models and Algorithms. London, UK: Kluwer Academic Publishers, 2007. doi: 10.1007/978-0-387-47534-9.
[11] S. Muthukrishnan. “Data Streams: Algorithms and Applications”, *Foundations and Trends in Theoretical Computer Science*, 2005, vol. 1, no.2, pp.117-236. doi: 10.1561/0400000002.
[12] В. А. Гимаров, М. И. Дли, и С. Я. Битюцкий, “Задачи нестационарной кластеризации состояния нефтехимического оборудования”, *Нефтегазовое дело*, № 2, с. 1-9, 2004.
[13] О. В. Ниссенбаум, “Алгоритм кластеризации потока данных с изменяющимися параметрами распределения”, *Вестник ТюмГУ*, № 7, с. 180-186, 2013.
[14] F. Cao, M. Estery, W. Qian, and A. Zhou, “Density-Based Clustering over an Evolving Data Stream with Noise”, *in Proc. International Conference on Data Mining*, Bethesda, 2006, pp.327-336. doi:10.1137/1.9781611972764.29
[15] A. Kassambara, *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. STHDA, 2017.
[16] H.-P. Kriegel, P. Kroeger, J. Sander, and A. Zimek, “Density-based clustering”, WIREs Data Mining and Knowledge Discovery, vol. 1, pp.231-240, 2011. doi: 10.1002/widm.30.
[17] C. C. Aggarwal, *Data clustering: algorithms and applications*. CRC Press, 2014.
[18] Examples of samples of the library Scikit-Learn. [Електронный ресурс]. Доступно: http://scikit-learn.org/stable/modules/classes.html.
[19] Е. А. Халов, “Систематический обзор четких одномерных функций принадлежности интеллектуальных систем”, Информационные технологии и вычислительные системы, № 3, с. 60-74, 2009.

Стаття надійшла до редакції 16 березня 2018 року.

REFERENCE

[1] O. I. Shelukhin, D. Z. Sakalema, and A. S. Filinova, *Detection of intrusions in computer networks (network anomalies)*. Moscow, Russian Federation: Hotline-Telecom, 2013.
[2] E. Wilson, Monitoring and analysis of networks. Methods for identifying faults, Moscow, Russian Federation: LORI, 2002.
[3] M. Collins, *Network Security Through Data Analysis*. Sebastopol, CA: O'Reilly Media, Inc., 2014.
[4] N. Adams, and N. Hearth, *Data Analysis for Network Cyber-Security*. Danvers, MA: Imperial College Press, 2014.
[5] M. Munia, S. Samrose, P. Dey, A. Annesha, and S. Hasan, “Network Intrusion Detection using Selected Data Mining Approaches: A Review”, *International Journal of Computer Applications*, vol. 132, no.13, pp.10-17, December 2015. doi: 10.5120/ijca2015907572.
[6] A. A Branitsky, and I. V. Kotenko, “Analysis and classification of methods for detecting network attacks”, in *Proc. SPIIRAN*, iss. 2 (45), pp.207-243, 2016. doi: 10.15622/sp.45.13.

[7] T. I. Buldakova, and A. Sh. Jalolov, “Choice of Data Mining Technologies for Intrusion Detection Systems in the Corporate Network”, *Engineering Journal: Science and Innovation*, iss. 11, pp. 1-14, 2013.
doi: 10.18698/2308-6033-2013-11-987.
[8] P. M. Shipulin, and A. N. Shniperov, “About possibility of application of Data Mining methods for the analysis of the distributed attacks in a network”, *Actual problems of aviation and komonavtiki*, vol. 1, p. 782-784, 2016.
[9] M. Ghesmoune, M. Lebbah, and H. Azzag, “State-of-the-art on Clustering Data Stream”, *Big Data Analytics*, vol. 1, no. 13, pp.1-27, 2016.
doi:10.1186/s41044-016-0011-3.
[10] C. C. Aggarwal, Data Streams: Models and Algorithms. London, UK: Kluwer Academic Publishers, 2007.
doi: 10.1007/978-0-387-47534-9.
[11] S. Muthukrishnan. “Data Streams: Algorithms and Applications”, *Foundations and Trends in Theoretical Computer Science*, 2005, vol. 1, no.2, pp.117-236.
doi: 10.1561/0400000002.
[12] V. A. Gimarov, M.I. Dli, and S.Y.Bityutsky, “Problems of non-stationary clusterization of the petrochemical equipment states”, *Neftegazovoedelo*, no. 2, pp. 1-9, 2004.
[13] O. V. Nissenbaum, “Algorithm for the clustering of a data stream with changing distribution parameters”, *Bulletin of Tyumen State University*, no.7, p.180-186, 2013.
[14] F. Cao, M. Estery, W. Qian, and A. Zhou, “Density-Based Clustering over an Evolving Data Stream with Noise”, *in Proc. International Conference on Data Mining*, Bethesda, 2006, pp.327-336.
doi:10.1137/1.9781611972764.29.
[15] A. Kassambara, *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. STHDA, 2017.
[16] H.-P. Kriegel, P. Kroeger, J. Sander, and A. Zimek, “Density-based clustering”, WIREs Data Mining and Knowledge Discovery, vol. 1, pp.231-240, 2011.
doi: 10.1002/widm.30.
[17] C. C. Aggarwal, *Data clustering: algorithms and applications*. CRC Press, 2014.
[18] Examples of samples of the library Scikit-Learn. [Електронный ресурс]. Доступно: http://scikit-learn.org/stable/modules/classes.html.
[19] E. A. Khalov, “A systematic review of clear one-dimensional functions of the ownership of intelligent systems”, Information technologies and computer systems, no. 3, pp. 60-74, 2009.

ДМИТРИЙ ШАРАДКИН

АЛГОРИТМ ПОТОКОВОЙ КЛАСТЕРИЗАЦИИ ДЛЯ МОНИТОРИНГА И ДИАГНОСТИКИ СОСТОЯНИЯ ТЕХНИЧЕСКИХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

В работе рассматриваются особенности автоматизации процессов мониторинга и диагностики состояния технических систем реального времени, в частности, современных компьютерных систем и сетей. Показано, что существующие методы обеспечивают решение задач диагностики и мониторинга с существенными ограничениями, которые в основном связаны с предположением о стационарности базовых характеристик объектов мониторинга. Специфические особенности систем, функционирующих в режиме реального времени, требуют, чтобы алгоритмы классификации и кластеризации, которые составляют основу современных средств мониторинга и диагностики, функционировали в потоковом режиме, одновременно отвечая требованиям минимизации объема задействованной памяти. Эти алгоритмы должны обеспечить практическую независимость времени работы от объема данных, поступающих на обработку; поддерживать работу с кластерами отличной от сферической формы в пространстве признаков; учитывать необходимость сохранения работоспособности в условиях, когда статистические характеристики потока данных

динамически изменяются и при неизвестном, возможно переменном, количестве кластеров в выборке; реализовывать процедуры выявления выбросов в исходных данных. Предложен алгоритм выявления кластерной структуры, основанный на использовании отображения входной выборки данных в специальным образом сконструированное сетевое пространство, с одновременным учетом как характеристик плотности заполнения пространства описания объектов так и его метрических свойств. Проанализированы свойства предложенного алгоритма и зависимость его внутренних характеристик, выбираемых при анализе, и от внешних параметров, зависящих от характеристик выборки данных. Модификация для случая потокового поступления данных позволяет адаптировать алгоритм, не требуя дополнительных затрат памяти для хранения информации. Для учета динамического изменения характеристик кластеров предложено использование функции забывания, рассмотрены возможные способы ее описания. Исследовано влияние разновидностей функции забывания на характеристики работоспособности предложенного алгоритма. Относительная простота алгоритма и семантическая прозрачность его параметров позволяют выполнять настройки алгоритма для различных областей применения, включая задачи обнаружения и предотвращения инцидентов в сфере информационной безопасности.

Ключевые слова: мониторинг и диагностика состояния систем реального времени, выявление инцидентов информационной безопасности, машинное обучение, классификация, кластеризация, поточная обработка данных, динамическое изменение форм и положения кластеров.

DMYTRO SHARADKIN

STREAMING CLUSTERING ALGORITHM FOR MONITORING AND CONDITION'S DIAGNOSTICS OF TECHNICAL REAL-TIME SYSTEMS

Special features of automatization of the states monitoring and diagnostics processes in technical systems which are executed in real-time mode, in particular modern computer systems and networks, are investigated and described in this paper. It is shown that the existing methods provide a solution for diagnostics and monitoring with significant limitations, which are mainly related to the stationary assumption of the basic characteristics of the objects of monitoring. The specific features of real-time systems require that the classification and clustering algorithms, which form the basis of modern monitoring and diagnostic tools, have to execute in a streaming mode, while simultaneously requirements for minimizing the amount of involved memory. These algorithms should provide practical independence of the execution time from the amount of data. They have to handling with clusters of spherical form in the feature space; to preserve the performance under conditions of dynamically changing of statistical characteristics of the data flow and with an unknown, possibly variable, number of clusters in the sample. Outliers and anomalies in data have to be detected and processed. An algorithm based on the simultaneous use of the mapping of the original data sample into a specially designed finite grid space, using both the fill density characteristics of the object description space and its metric properties for detecting the cluster structure is proposed. The properties of the algorithm and the dependence of its characteristics from the specified parameters are analysed. Some modification of the algorithm allows execute streaming data processing, easily adapt the algorithm without utilization extra memory. For handling of the clusters' parameters dynamic changes the attenuation function was introduced. Some variants of its specification were considered, their influence on proposed algorithm's performance was analyzed. The relative simplicity of the algorithm and the semantic transparency of its external parameters make it possible simple configure the algorithm for various areas of its application, including the tasks of IT-security incidents detecting and preventing in computer systems and networks.

Keywords: monitoring and diagnostics of real-time systems, detection of information security incidents, machine learning, classification, clustering, streaming data processing, dynamic changes of cluster’s form and position.

Дмитро Михайлович Шарадкін, кандидат технічних наук, доцент, доцент кафедри кібербезпеки та застосування автоматизованих інформаційних систем і технологій, Інститут спеціального зв'язку та захисту інформації Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна.

ORCID: 0000-0001-6407-8040.

E-mail:dmsh@ukr.net.

Дмитрий Михайлович Шарадкин, кандидат технических наук, доцент, доцент кафедры кибербезопасности и применения автоматизированных информационных систем и технологий, Институт специальной связи и защиты информации Национального технического университета Украины "Киевский политехнический институт имени Игоря Сикорского", Киев, Украина.

DmytroSharadkin, candidate of technical sciences, associate professor, associate professor at the cybersecurity and application of automated information systems and technologies academic department, Institute of special communication and information protection National technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.