УДК 004.92:004.021

ОПТИМІЗАЦІЯ ВІЗУАЛІЗАЦІЇ ВЕЛИКИХ ТРАНЗАКЦІЙНИХ ГРАФІВ ДЛЯ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ПАТЕРНІВ НА ОСНОВІ СИЛОВИХ АЛГОРИТМІВ МАКЕТУВАННЯ

DOI: 10.33842/2313-125X-2026-29-184-200
Логвиненко В.Є.,
logvinenko.js@gmail.com, ORCID: 0009-0001-0161-478X
Аушева Н.М., д-р техн. наук, професор,
nataauscheva@gmail.com, ORCID: 0000-0003-0816-2971
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", (м. Київ, Україна)

У роботі досліджено проблему візуалізації великих транзакційних графів у задачах виявлення шахрайських операцій у фінансових системах. Актуальність теми обумовлена стрімким зростанням обсягів транзакційних даних та складністю їх структурної організації, що ускладнює як автоматичний аналіз, так і експертну інтерпретацію. Показано, що класичні силові алгоритми макетування при застосуванні до графів із сотнями тисяч вузлів і зв'язків не забезпечують необхідної швидкодії, що унеможливлює їх використання в інтерактивних системах візуальної аналітики.

Обґрунтовано доцільність розділення задачі на два рівні: попередню обробку графа та оптимізацію обчислення сил у процесі макетування. Запропоновано підхід, що поєднує фільтрацію малозначущих структурних елементів графа та топологічно обумовлену апроксимацію сил відштовхування. Ключовою ідеєю є заміна попарної взаємодії між вузлами на взаємодію з центрами мас зв'язних компонент, що дозволяє суттєво зменшити обчислювальну складність без критичної втрати якості візуалізації.

Проведено експериментальне дослідження на синтетичному датасеті SAML-D, який моделює транзакційні процеси з вбудованими сценаріями відмивання коштів. У ході експерименту оцінювались такі метрики, як час однієї ітерації, кількість ітерацій до досягнення заданого рівня якості (на основі показника збереження сусідства), а також візуальна якість розміщення. Отримані результати показали суттєве скорочення часу ітерації та загальної кількості ітерацій при незначній деградації точності відображення. Доведено, що запропонований підхід забезпечує можливість інтерактивної роботи з великими графами та дозволяє зберігати ключові структурні патерни, такі як хаби та транзакційні ланцюги, що є критично важливими для задач фінансового моніторингу. Результати можуть бути використані при розробці систем візуальної аналітики для підтримки прийняття рішень у сфері AML.

Ключові слова: візуалізація графів, силові алгоритми, Fruchterman–Reingold, шахрайські транзакції, AML, транзакційні мережі.

Постановка проблеми. Проблема виявлення шахрайських транзакцій перш за все передбачає аналіз даних великої розмірності які мають складну мережу взаємозв'язків між неоднорідними вузлами. При цьому частка саме шахрайських операцій є надзвичайно низькою (менше 1% від загального обсягу транзакцій), що створює значний дисбаланс даних і ускладнює їх виявлення [9].

У задачах виявлення шахрайських транзакцій доцільно виділити дві окремі підзадачі – виявлення підозрілих об'єктів та аналіз і інтерпретація зв'язків між ними. Виявлення підозрілих об'єктів можна звести до задачі класифікації або задачі виявлення аномалій, яка вирішується повністю автоматично за допомогою методів машинного навчання або алгоритмів, заснованих на правилах. Методи вирішення такої задачі зазвичай аналізують кожен об'єкт в мережі транзакцій як набір атрибутів і можуть враховувати лише локальне оточення об'єкта.

Натомість, задача аналізу і інтерпретації зв'язків спрямована на дослідження структури мережі транзакцій в цілому. Методи, що вирішують цю задачу мають надати змогу виявляти більш складні патерни, такі як транзакційні ланцюги, цикли, групи взаємозв'язаних рахунків тощо. Розв'язання цієї задачі не може бути повністю автоматизоване, оскільки інтерпретація зв'язків і прийняття остаточного рішення потребують урахування контексту та наявності експертних знань, тож ефективне виявлення складних патернів можливе лише за рахунок участі аналітика у процесі. Крім того, згідно з рекомендаціями Financial Action Task Force [6], фінансові установи зобов'язані приймати рішення щодо підозрілої транзакції на основі експертної оцінки, що прямо передбачає участь людини.

Природнім методом візуалізації так званої «карти транзакцій» є її відображення у вигляді графа. При цьому реальні транзакційні мережі можуть містити десятки тисяч і більше вузлів та сотні тисяч зв'язків. Фактична велика кількість ребер є одним із ключових факторів, що впливає на здатність людини виконувати аналітичні задачі [11, 17], що призводить до необхідності дослідження методів ефективної та наглядної візуалізації графів великої розмірності.

Одним із найпоширеніших підходів до візуалізації графа є так звані «силові методи макетування», які моделюють граф як фізичну систему, де кожна вершина взаємодіє з усіма іншими через сили відштовхування (залежно від відстані на макеті), у той час як пов'язані вершини притягуються між собою. Таким чином ці методи мають змогу забезпечити відображення графа зберігаючи його кластерні та структурні особливості. Такі алгоритми є ітераційними і вимагають великої кількості ітерацій до отримання першої якісної візуалізації а також їх повторного виконання при

подальших внесеннях змін у структуру графу. Існуючі підходи по-різному балансують між швидкістю обчислень та здатністю зберігати структурні властивості мережі, отже виникає необхідність порівняльного аналізу існуючих алгоритмів саме в контексті аналізу великих транзакційних мереж.

Аналіз останніх досліджень і публікацій. Сучасні дослідження акцентують увагу на недостатності одних лише автоматичних методів виявлення шахрайських транзакцій. У роботі [9] показано, що складні шахрайські схеми формуються як структурні патерни у транзакційних мережах і не можуть бути виявлені лише за допомогою локального аналізу ознак. Автори пропонують підхід візуального аналізу, в якому графове відображення карти транзакцій є основним середовищем дослідження, навколо якого побудована інтерактивна взаємодія – фільтрація, масштабування, виділення підграфів, проте алгоритми побудови графа автори роботи не розглядають.

Систематичний огляд сучасних методів транзакційних мереж [10] описує класичні графові метрики, алгоритми виявлення аномалій та підходи на основі графових нейронних мереж. При цьому також залишають відкритим питання саме графової візуалізації і її важливості для аналітика. У роботі [13], натомість, показано, що саме візуальне дослідження графа є важливим для виявлення класичних та нових патернів. Візуальний аналіз також пропонують проводити через інтерактивну роботу з графом. Разом з тим, алгоритмічні аспекти побудови макету графа авторами не аналізуються.

Практичним продуктом є програмна система VISFAN [3], яка створювалась саме з метою аналізу фінансових мереж. Цей продукт працює з інтерактивною графовою візуалізацією та підтримує навігацію, фільтрацію та поступове розкриття структури графа. Водночас невідомо, який підхід застосовано для відображення самого макета – ймовірніше за все силовий або гібридний, проте їх ефективність не аналізується.

Таким чином, дослідження підтверджують необхідність використання графових візуалізацій для аналізу транзакційних мереж. Проте, алгоритми побудови макету графа практично не розглядаються як окремий об’єкт дослідження. Тому ми вважаємо, що попри залежність якості аналізу від якості та швидкості візуалізації, питання ефективності силових методів макетування саме в контексті аналізу транзакційних графів залишається недостатньо дослідженим.

Формулювання цілей статті. Метою роботи є дослідження методів ефективної та наглядної візуалізації транзакційних графів великої розмірності з урахуванням вимог до інтерактивності.

У цьому контексті ключовими характеристиками є час виконання ітерації, що визначає можливість інтерактивної роботи з графом, а також якість отриманого макету. Зокрема, мінімізація кількості перетинів ребер та збереження локальної структури є критичними для аналізу. Окрему увагу

приділено здатності алгоритмів до ефективного перерахунку макету при зміні підграфа, а також необхідності адаптації силових методів до структурних особливостей транзакційних мереж.

Основна частина. Значна частина сучасних досліджень у сфері AML базується на синтетичних датасетах [10]. Причиною тому є закритість для відкритих досліджень реальної банківської інформації, включаючи інформацію про транзакції. Це призводить до необхідності в якісних синтетичних датасетах, які б відповідали вимогам масштабу, розподіленості у часі та наявності анотованих шахрайських сценаріїв.

Одним з поширених датасетів, що б відповідав таким вимогам, є датасет SAML-D [15], сформований з використанням симулятора AMLSim [2]. SAML-D є обґрунтованим вибором в ряді популярних досліджень [14, 18, 20] та зазвичай є одним із основних варіантів для тестування та валідації рішень з виявлення шахрайства. Цей датасет містить список транзакцій з наступними атрибутами – ідентифікатори відправника та отримувача, час виконання транзакції, сума переказу та деякі службові атрибути якими позначають шахрайські транзакції.

Далі в цій роботі даний датасет буде використано для формування гіпотез щодо структури транзакційних мереж та для візуального аналізу окремих підграфів.

Для побудови розміщення вузлів як базовий експеримент використовується алгоритм Fruchterman–Reingold (FR) [7], який є класичним силовим методом макетування. Алгоритм моделює граф як фізичну систему, у якій вузли взаємодіють через сили притягання та відштовхування, а кінцева конфігурація визначається як стан рівноваги цієї системи. Нехай V – множина вершин графа, E – множина ребер графа, A – площа області відображення $d(u, v)$ – евклідова відстань між вузлами u та v, $k = \sqrt{\frac{A}{|V|}}$ – оптимальна відстань між вузлами u та v. Алгоритм передбачає, що на кожну окрему вершину одночасно діють сили тяжіння від суміжних вершин та сили відштовхування – від усіх вершин. Сили тяжіння $\vec{f}_{rep}$ і відштовхування $\vec{f}_{attr}$ що діють на вершину u від суміжної вершини v обчислюються як:

$$\vec{f}_{rep}(v,u) = \frac{k^2}{d(v,u)} * \frac{\overrightarrow{p_v} - \overrightarrow{p_u}}{d(v,u)} \quad ; \tag{1}$$

$$\vec{f}_{attr}(v,u) = \frac{d(v,u)}{k^2} * \frac{\overrightarrow{p_u} - \overrightarrow{p_v}}{d(v,u)} \tag{2}$$

де $\overrightarrow{p_v}$, $\overrightarrow{p_u}$ – позиції вершин u та v відповідно. Тоді сумарна сила що діє на вершину v, буде дорівнювати:

$$\vec{F}(v) = \sum_{u \in V, u \neq v} \vec{f}_{rep}(v,u) + \sum_{(v,u) \in E,} \vec{f}_{attr}(v,u) \ . \tag{3}$$

За певну кількість ітерацій алгоритм формує граф, який відображає структурні зв'язки між вершинами шляхом просторового розміщення, що мінімізує перетини ребер та зберігає локальні сусідства вузлів.

Важливо зазначити, що в даній роботі виконання алгоритму перенесено на графічний процесор за допомогою технології WebGPU [19]. Дослідження показують, що використання графічного процесора значно ефективніше за центральний процесор для задач такого типу через природну паралельність силових алгоритмів [5, 8].

Повний датасет SAML-D містить приблизно 9 504 852 транзакції та близько 676 912 унікальних акаунтів. Також він охоплює близько 11 місяців транзакційної активності. Складність однієї ітерації алгоритма Fruchterman–Reingold визначається обчисленням сил відштовхування між усіма парами вершин і має порядок $O(|V|^2)$. Для графа з

$$|V| = 676912 \tag{4}$$

кількість пар становить:

$$\frac{|V|(|V|-1)}{2} = \frac{676\,912 * 676\,911}{2} \approx 2.29 * 10^{11}. \tag{5}$$

Тобто лише на одну ітерацію потрібно врахувати близько 229 мільярдів парних взаємодій. Якщо ж виконувати, наприклад, 500 ітерацій, то це вже

$$500 * 2.29 * 10^{11} \approx 1.15 * 10^{14} \tag{6}$$

парних обчислень, не враховуючи сили притягання вздовж ребер. За наявності понад 9.5 млн транзакцій додаткова складова, пов'язана з ребрами, також є дуже великою. Тому повна візуалізація всього датасету в межах базового FR є практично недоцільною.

Вважається, що підозрілі транзакції часто проявляються як інтенсивні короткострокові активності, які формують характерні структурні патерни у графі [8, 13]. Тому у роботі пропонується обмежити аналіз часовим підграфом тривалістю в 5 днів. Припускаючи відносно рівномірний розподіл транзакцій у часі, 5-денний інтервал охоплює приблизно 2% від загального обсягу даних, що зменшує кількість вершин до порядку 10^4, що може бути обчислювально досяжним на обмеженому обладнанні.

Використання будь-якого методу візуалізації вимагає можливості швидкої оцінки структури графа. Це означає, що суміжні вузли повинні знаходитись якомога ближче і у візуальному представленні. Кількісно це можна обчислити за допомогою метрики збереження локальних околів *(англ. Neighbor Preservation, далі NP).* Для кожної вершини порівнюються множини її найближчих сусідів у графі та у просторі розміщення:

$$NP = \frac{1}{|V|} \sum_{v \in V} \frac{|N_k^{graph}(v) \cap N_k^{layout}|}{k}, \tag{7}$$

де N_k^{graph} – множина k найближчих суміжних вузлів, N_k^{layout} – множина k найближче розташованих вузлів. Цей критерій є ключовим для оцінки якості візуалізації, адже візуальне виявлення патернів неможливе за

хаотичного розташування вершин. Саме тому ми також будемо використовувати цей критерій для визначення на якій саме ітерації алгоритм може бути зупинений.

Також, як було зазначено вище, саме наявність високого числа ребер графа є ключовим фактором, заважаючим людині виконувати аналітичні задачі [11, 17]. Тож метод візуалізації повинен мінімізувати кількість перетинів ребер задля збільшення читабельності графа і покращення інтерпретованості візуалізації. Базовою характеристикою саме швидкодії алгоритма можна вважати час ітерації. Він є обернено пропорційним частоті оновлення зображення (frames per second, FPS), З точки зору сприйняття користувачем, FPS < 10 сприймається «ривками», тоді як FPS > 60 виглядає «плавним». Діапазон 15-30 FPS можна вважати прийнятним для користувача, тож в якісній візуалізації час однієї ітерації не повинен перевищувати 33-67мс.

Базовий експеримент. Отже, профільтрувавши транзакції з датасету SAML-D, залишивши лише дані за перші 5 днів, отримано 134,844 транзакцій які були виконані 81,509 користувачем. Далі проведено експеримент з відображення цього підграфа з використанням вищенаведеного алгоритму FR. Рис. 1 відображає результат візуалізації з максимальним віддаленням, де можна побачити весь граф цілком та набір критеріїв які були пораховані в процесі візуалізації.

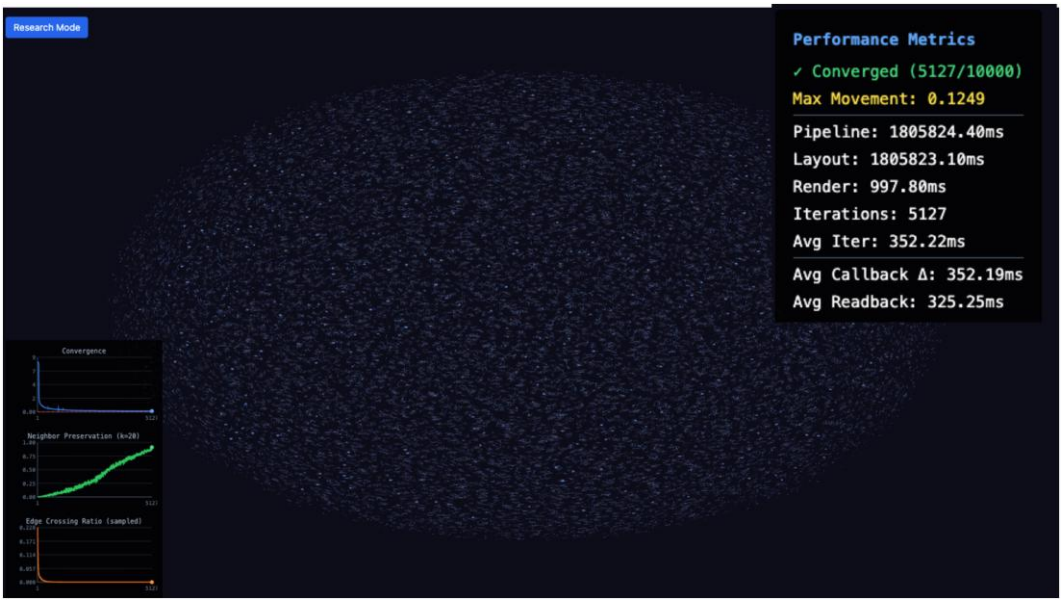

Рис. 1. Повний результат візуалізації

З віддаленого малюнку видно лише загальнорозподілену структуру графа, тож для більш детального ознайомлення на рис. 2 наведено її наближене представлення, де можна побачити структуру графа більш детально.

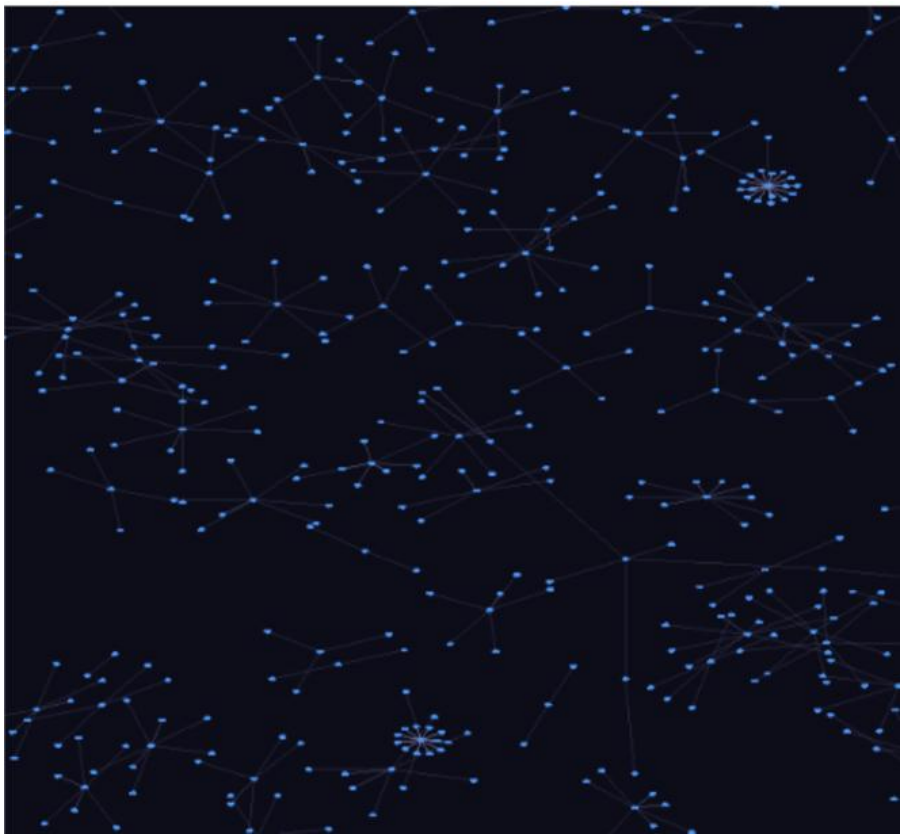

Рис. 2. Результат візуалізації з наближенням

Пораховані за допомогою побудованої програми критерії ефективності, які можна бачити з рис. 1, також наведено в таблиці 1.

Таблиця 1

Результати базового експерименту

Метрика	Значення
Кількість ітерацій до NP = 0.9	> 5,127
Середній час ітерації	352.22мс

Як бачимо з результатів, попре можливість обчислити макет на обмеженому обладнанні, середній час ітерації є неприйнятно великим. В цьому випадку аналітику доводиться чекати кожного оновлення макету приблизно пів секунди (включаючи стадію відмальовування), що унеможливлює «плавне» сприйняття анімації. Також досягнення NP = 0.9 відбулось за цілих 5,127 ітерацій, що при часі ітерації в 352.22мс призводить до повного часу роботи алгоритму в приблизно 30хв. Такі результати не можуть надати користувачеві можливості комфортно аналізувати граф. Наступні кроки, описані в цій статті, спрямовані на суттєве покращення даної ситуації.

Етап попередньої обробки даних. Запропонований в цій роботі підхід передбачає етап попередньої обробки вхідних даних. Важливою особливістю транзакційних графів є наявність великої кількості окремих

зв’язних компонент графа. За обмежений проміжок часу більшість рахунків зазвичай виконує невелику кількість транзакцій і лише з кількома контрагентами, тому вони можуть не з’єднуватись з основною частиною графа.

Для підтвердження цієї гіпотези а також для подальшого аналізу характеру зв’язних компонент побудовано розподіл зв’язних компонент за їх розміром. Його зображено на рис. 3.

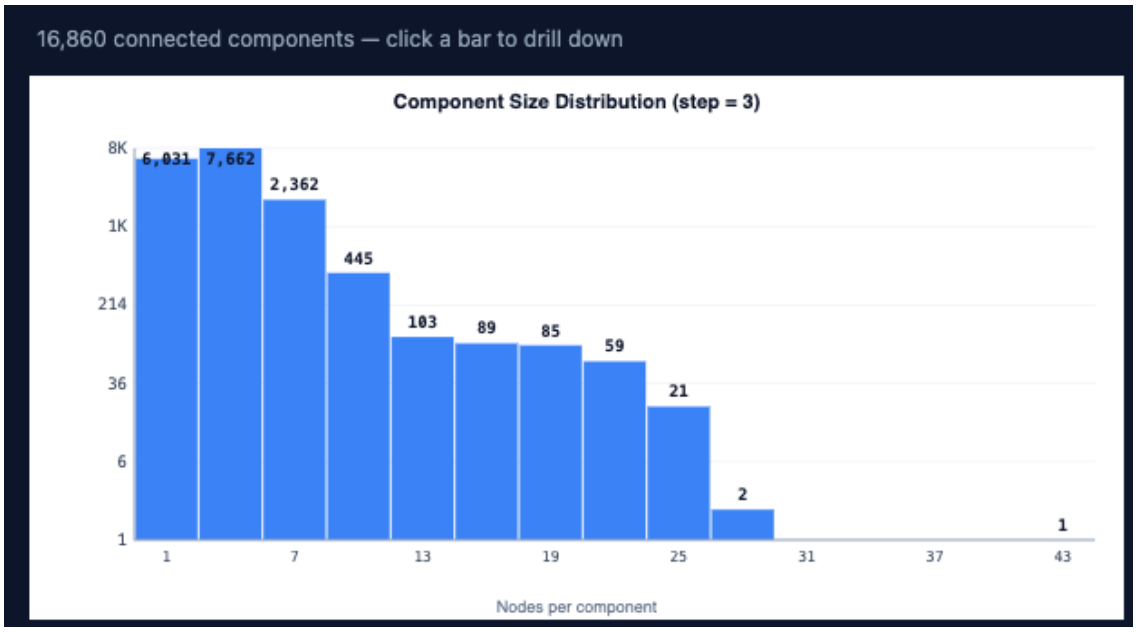

Рис. 3. Розподіл зв’язних компонент досліджуваного графа за їх кількістю

Як бачимо з рисунку, досліджуваний граф складається з 16,860 зв’язних компонент. Це підтверджує висунуту гіпотезу про характер графа, що може буде надалі застосовано для адаптації самого алгоритму. Більш того, з наведеного розподілу видно, що більша частина зв’язних компонент налічує 1-6 вузлів. Для детального погляду на такі компоненти побудовано їх розподіл за сумарною кількістю транзакцій в межах. Такий розподіл наведено на рис.4 для компонент з 1-3 вузлів та компонент з 4-6 вузлами відповідно.

Як бачимо з рис.4, переважна більшість груп з 1-3 вузлами налічує 1-3 транзакції А переважна більшість груп з 4-6 вузлами налічує до 6 транзакцій. Аномальні групи ж починаються на більшій кількості транзакцій. Це підтверджує припущення про відсутність підозрілих дій у таких транзакція, а отже, вони можуть бути відфільтровані.

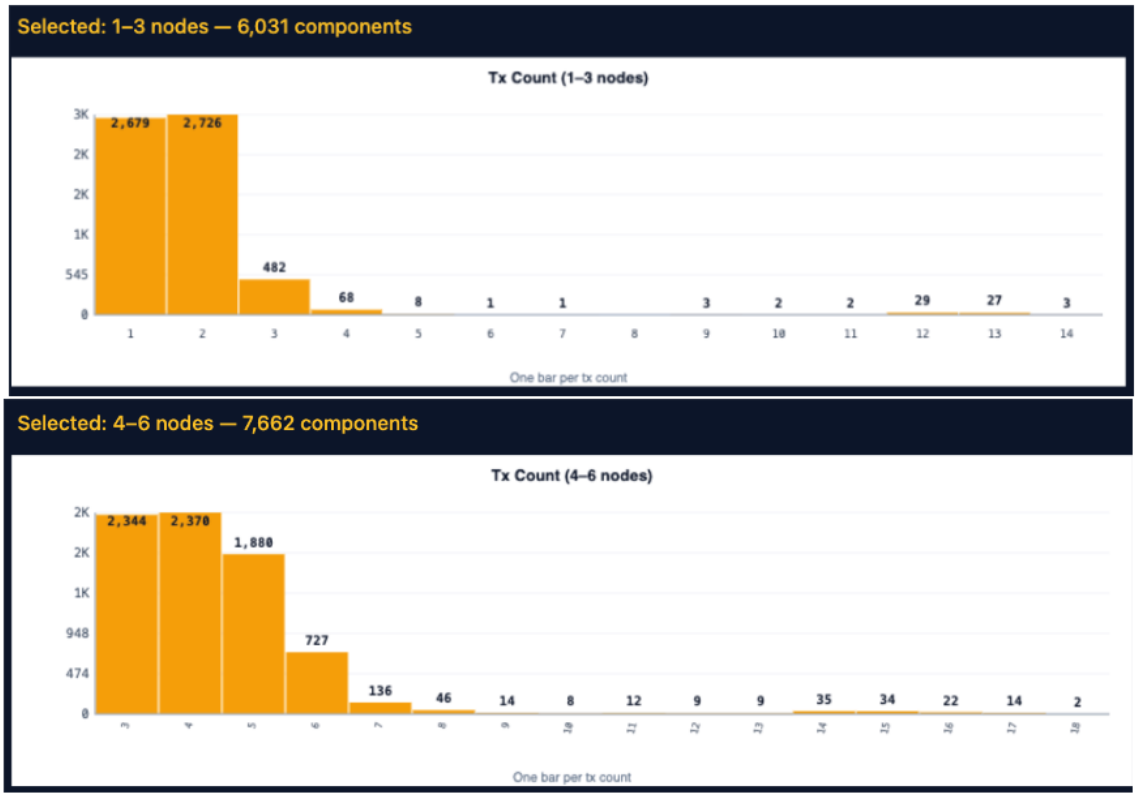

Рис. 4 Розподіл за кількістю транзакцій для компонент з 1-3 вузлами та 4-6 вузлами

Таким чином, на етапі попередньої обробки даних було вирішено провести завчасну фільтрацію, де вузли що не проходять по параметрам, наведеними в таблиці 2, видаляються з графу перед проведенням візуалізації.

Таблиця 2

Параметри фільтрації непідозрілих транзакцій.

Параметр	Значення
Мінімальна кількість вузлів в групі	3
Мінімальна кількість транзакцій в групі	6

Важливо зазначити, що для видалення вузла з графу він не повинен перевищувати жоден з параметрів. Це треба для того, щоб малі групи з великою активністю все ще потрапляли у візуалізацію, де вже аналітик зможе коректно інтерпретувати таку активність. Елемент інтерфейсу, який відображає результати фільтрування, зображено на рис. 5.

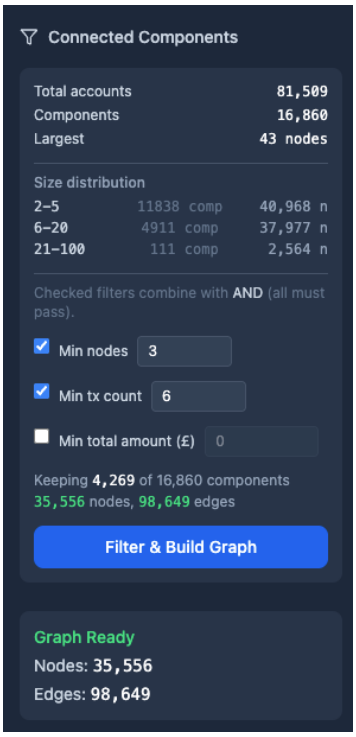

Рис. 5. Параметри і результат фільтрування

Таким чином, початковий граф з 134,844 ребер та 81,509 вузлів було спрощено до графа розміром 35,556 вузлів та 98,649 ребер.

Модифікація алгоритму. Головним недоліком алгоритму FR є його квадратична складність. Тобто на кожній ітерації для кожної вершини потрібно виконувати повний обхід графа для обчислення повної суми відштовхування кожного вузла на кожен. Сучасні підходи до прискорення алгоритмів силового макетування засновані на апроксимації сил відштовхування, тобто заміною обчисленням кожної сили окремо на обчислення агрегованої сили, що діє від групи вершин. Відмінність таких підходів визначається перш за все методом апроксимації. Найбільш поширені підходи – просторові [4], стохастичні [12], або топологічні [16], орієнтовані на загальний випадок графової структури і не враховують специфіку транзакційних мереж, зокрема наявність великої кількості зв'язних компонент.

У даній роботі пропонується модифікація алгоритму, що використовує апроксимацію сил на основі центрів мас зв'язних компонент. Нехай граф вже розбитий на деяку множину зв'язних компонент $C = \{C_1, C_2, \dots, C_k\}$ (ми фактично вже зробили це вище на етапі препроцесингу). Після цього для кожної компоненти обчислюється її центр мас:

$$\vec{m}_j = \frac{1}{|C_j|} \sum_{v \in C_j} \overrightarrow{p_v}, \tag{8}$$

де $\overrightarrow{p_v}$ – позиція вершини v. Тоді взаємодія між різними компонентами апроксимується через їх центри мас, тобто сила відштовхування, що діє на вершину $v \in C_i$ з боку іншої компоненти C_j буде дорівнювати

$$\vec{f}_{rep}^{out}(v, C_j) = -\frac{k^2 * |C_j|}{d(v, \vec{m}_j)} * \frac{\overrightarrow{p_v} - \vec{m}_j}{d(v, \vec{m}_j)}, v \in C_i, \ C_j \neq C_i. \quad (9)$$

У той же час сила тяжіння залишається незмінною (1), відштовхування між вершинами однієї компоненти обчислюється точно за класичною формулою:

$$\vec{f}_{rep}^{in}(v, u) = \frac{k^2}{d(v, u)} * \frac{\overrightarrow{p_v} - \overrightarrow{p_u}}{d(v, u)}, u, v \in C_i \, . \quad (10)$$

І сумарна сила що діє на вершину набуває вигляду

$$\vec{F}(v) = \sum_{\substack{u \in C_i \\ v \in C_i \\ u \neq v}} \vec{f}_{rep}^{in}(v, u) + \sum_{\substack{u \in C_i \\ v \in C_j \\ C_i \neq C_j}} \vec{f}_{rep}^{out}(v, u) + \sum_{(v,u) \in E,} \vec{f}_{attr}(v, u). \quad (11)$$

Таким чином, запропонований підхід є топологічно обумовленою апроксимацією сил відштовхування, близькою за ідеєю до методу Barnes–Hut [4], однак замість просторової декомпозиції використовується структурна декомпозиція графа на зв'язні компоненти.

Складність запропонованого алгоритму буде залежати від кількості зв'язних компонент $|C|$ і визначається як

$$O\left(\sum_{i=1}^{|C|} |C_i|^2 + |V| + |C|\right), \quad (12)$$

що у випадку збалансованої кількості зв'язних компонент буде суттєво меншим за квадратичну складність $O(|V|^2)$ у класичному алгоритмі.

Основний експеримент цього дослідження проведено на фільтрованому графі, який містить кількість вершин $|V| = 31{,}248$ та кількість ребер $|E| = 50{,}578$. Попередня обробка даних (фільтрація) описана у попередньому розділі.

На рис. 6 представлено загальний вигляд графа після застосування алгоритму. Видно, що зв'язні компоненти формують окремі кластери без взаємного перекриття, що відповідає структурі даних. В таблиці 3 наведені результати вимірювання швидкості роботи алгоритму.

Таблиця 3

Критерії оцінки покращеного алгоритму

Метрика	Значення
Кількість ітерацій до NP = 0.9	> 1,425
Середній час ітерації	15.57мс

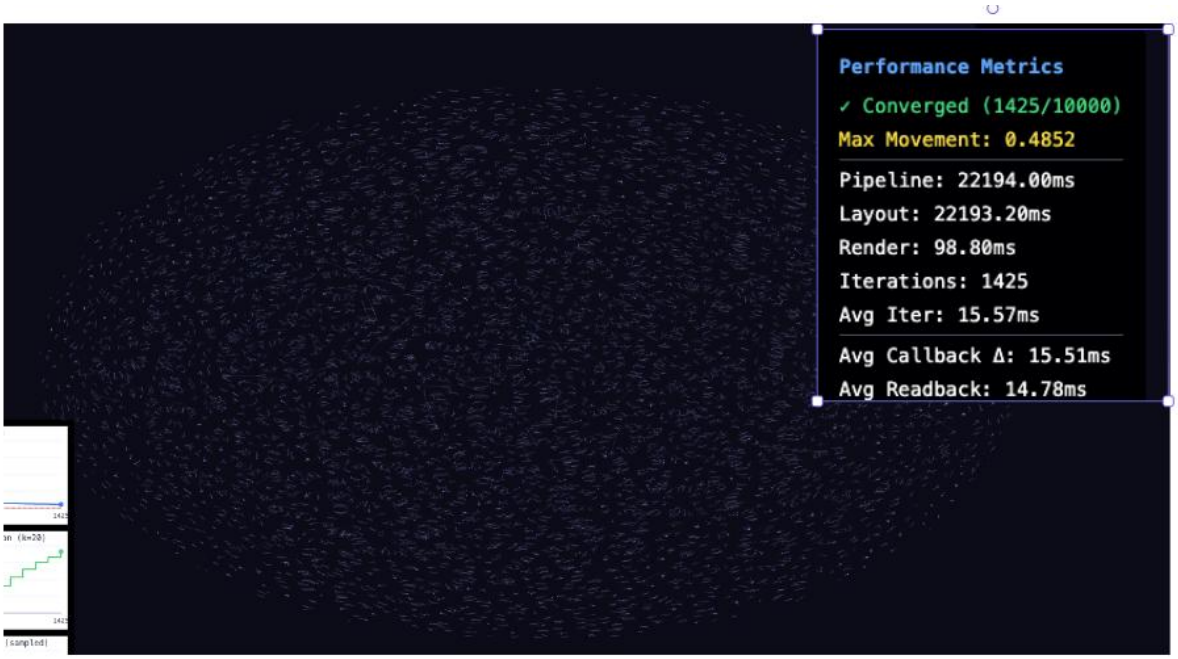

Рис. 6. Повний результат пришвидшеної візуалізації

Як бачимо, запропонований підхід зменшує час однієї ітерації більш ніж у двадцять разів, а досягнення критерію якості NP здобувається за 1,425 ітерації, що є майже в 4 рази швидше за класичний алгоритм. Це робить даний програмний інструмент можливим для інтерактивної роботи і гарантує більш стабільну і швидку збіжність.

На рис.7 зображено наближений фрагмент графа після застосування запропонованої модифікації. Може спостерігатись певна деградація структури розташування вершин, особливо у відображені так званих «хабів».

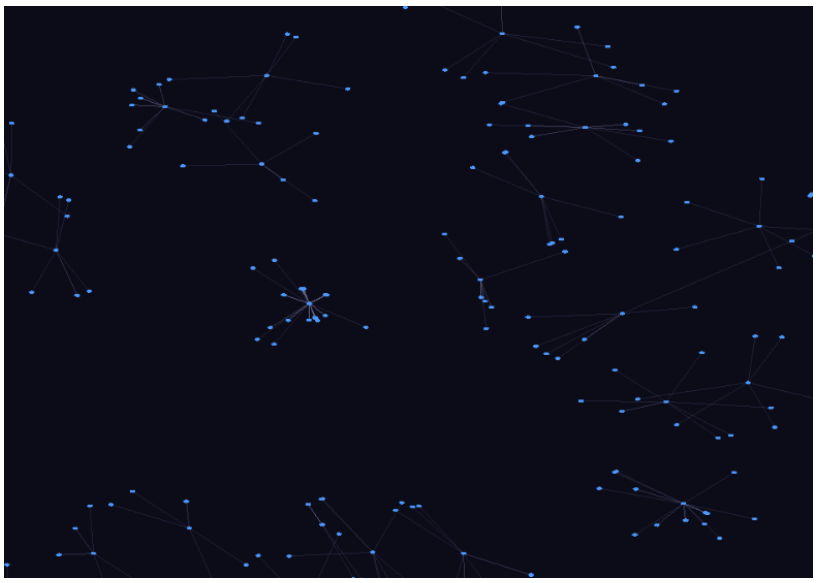

Рис. 7. Наближений фрагмент графа після пришвидшеної візуалізації

Дана деградація не виглядає значною. Видно, що навіть за умов апроксимації сил відштовхування відповідні структури залишаються чітко вираженими та легко ідентифікуються на графі. Як приклад, на рис. 8 наведено візуалізацію з підсвіткою характерних патернів типу «ланцюг» та «хаб» (fan-in).

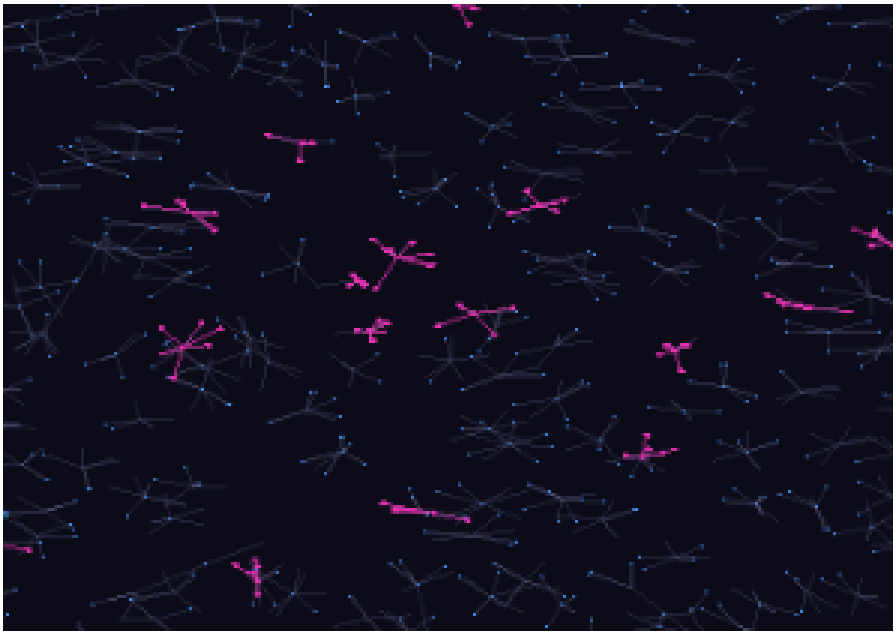

Рис. 8. Підсвітка патернів типу «хаб»

Висновки. У роботі показано, що застосування класичних силових методів макетування без урахування специфіки транзакційних графів є практично непридатним для інтерактивного аналізу. Навіть після обмеження даних у часовому вимірі базовий алгоритм Fruchterman–Reingold демонструє неприйнятні характеристики швидкодії та збіжності, що унеможливлює ефективну роботу аналітика. Це підтверджує, що проблема візуалізації великих фінансових графів не може бути вирішена лише за рахунок обчислювальних ресурсів і потребує адаптації самих алгоритмів.

Запропонований підхід, що поєднує попередню фільтрацію графа та топологічно обумовлену апроксимацію сил відштовхування на основі зв’язних компонент, дозволяє суттєво зменшити обчислювальну складність. Експериментально підтверджено значне скорочення часу ітерації та кількості ітерацій до досягнення прийнятної якості (NP), при цьому деградація візуальної структури є незначною і не впливає на можливість виявлення ключових патернів, таких як ланцюги та хаби. Це свідчить про доцільність використання структурних властивостей транзакційних мереж для оптимізації алгоритмів макетування та підтверджує ефективність запропонованого підходу для задач аналізу шахрайських транзакцій.

Подальші дослідження доцільно спрямувати на розширення

запропонованого підходу з урахуванням гетерогенності транзакційних мереж, зокрема різних типів вузлів та зв'язків. Перспективним є також дослідження методів виявлення та інтеграції характерних патернів шахрайської поведінки безпосередньо у процес побудови макету.

Література

1. Aiding Humans in Financial Fraud Decision Making. arXiv. 2024. DOI: https://doi.org/10.48550/arXiv.2408.14552.
2. AMLSim: Anti-Money Laundering Simulator. IBM. URL: https://github.com/IBM/AMLSim (дата звернення: 13.04.2026).
3. An Advanced Network Visualization System for Financial Crime Detection (VISFAN). *IEEE Pacific Visualization Symposium* (Hong Kong, 2011). 2011. P. 121–128. DOI: https://doi.org/10.1109/PACIFICVIS.2011.5742391.
4. Barnes J., Hut P. A hierarchical O(N log N) force-calculation algorithm. *Nature*. 1986. Vol. 324. P. 446–449. DOI: https://doi.org/10.1038/324446a0.
5. Brinkmann G. et al. Exploiting GPUs for Fast Force-Directed Visualization of Large Networks. *IEEE International Conference on Parallel Processing*. 2017. P. 382–391. DOI: https://doi.org/10.1109/ICPP.2017.47.
6. Financial Action Task Force. Recommendations. URL: https://www.fatf-gafi.org/recommendations.html (дата звернення: 13.04.2026)
7. Fruchterman T. M. J., Reingold E. M. Graph Drawing by Force-directed Placement. *Software: Practice and Experience*. 1991. Vol. 21, No. 11. P. 1129–1164. DOI: https://doi.org/10.1002/spe.4380211102.
8. Godiyal A., Hoberock J., Garland M. Rapid Multipole Graph Drawing on the GPU. *Lecture Notes in Computer Science*. 2009. Vol. 5876. P. 90–101. DOI: https://doi.org/10.1007/978-3-642-00219-9_10.
9. Mayr E., Kohlhammer J. et al. The Value of Visualization. *IEEE Transactions on Visualization and Computer Graphics*. 2017. Vol. 23, No. 1. P. 681–690. DOI: https://doi.org/10.1109/TVCG.2017.2744758.
10. Network Analytics for Anti-Money Laundering — A Systematic Literature Review and Experimental Evaluation. *arXiv*. 2024. DOI: https://doi.org/10.48550/arXiv.2405.19383.
11. Purchase H. C. Which Aesthetic Has the Greatest Effect on Human Understanding? *Graph Drawing: 5th International Symposium*, GD '97 (Rome, Italy, Sept. 18–20, 1997).
12. R. Gove. A Random Sampling O(n) Force-calculation Algorithm for Graph Layouts. Computer Graphics Forum. 2019. Vol. 38, No. 3. P. 1–12. DOI: https://doi.org/10.1111/cgf.13724.
13. Singh K., Best P. Anti-money laundering: Using data visualization to identify suspicious activity. Journal of Accounting and Information Systems. 2019. Vol. 34. P. 100419. DOI: https://doi.org/10.1016/j.accinf.2019.06.001.
14. Statistical Insights into Anti-Money Laundering: Analyzing Large-Scale Financial Transactions. *Zenodo*. 2025. DOI: https://doi.org/10.5281/zenodo.18107429.

15. Synthetic Transaction Monitoring Dataset (SAML-D). Kaggle. URL: https://www.kaggle.com/datasets/berkanoztas/synthetic-transaction-monitoring-dataset-aml
16. Topology-driven force-directed algorithms. *Lecture Notes in Computer Science*. 2010. Vol. 6502. P. 256–267. DOI: https://doi.org/10.1007/978-3-642-18469-7_15.
17. Ware C., Purchase H. C., Colpoys L., McGill M. Cognitive Measurements of Graph Aesthetics. *Information Visualization*. 2002. Vol. 1, No. 2. P. 103–110. DOI: https://doi.org/10.1057/palgrave.ivs.9500013.
18. Weber M., Chen J., Suzumura T. et al. Scalable Graph Learning for Anti-Money Laundering: A First Look. *arXiv*. 2019. DOI: https://doi.org/10.48550/arXiv.1812.00076.
19. WebGPU. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebGPU_API (дата звернення: 13.04.2026).
20. Zola F. et al. Synthetic Pattern Generation and Detection of Financial Crimes. *arXiv*. 2026. DOI: https://doi.org/10.48550/arXiv.2601.21446.

OPTIMIZATION OF LARGE TRANSACTION GRAPH VISUALIZATION FOR FRAUD PATTERN DETECTION USING FORCE-DIRECTED LAYOUT ALGORITHMS

Vladyslav Lohvynenko, Nataliia Ausheva

This study investigates the problem of visualizing large transaction graphs in the context of fraud detection in financial systems. The relevance of the study is driven by the rapid growth of transactional data volumes and the increasing complexity of their structural organization, which complicates both automated analysis and expert interpretation. It is shown that classical force-directed layout algorithms, when applied to graphs with hundreds of thousands of nodes and edges, fail to provide sufficient performance, making their use in interactive visual analytics systems impractical.

The study justifies the decomposition of the problem into two levels: graph preprocessing and optimization of force computation during layout. An approach is proposed that combines the filtering of low-significance structural elements with a topology-aware approximation of repulsive forces. The key idea is to replace pairwise node interactions with interactions between nodes and the centers of mass of connected components, which significantly reduces computational complexity without critical loss of visualization quality.

An experimental study is conducted on the synthetic SAML-D dataset, which models transaction processes with embedded money laundering scenarios. The evaluation considers metrics such as per-iteration time, the number of iterations required to reach a target quality level (based on neighborhood preservation), and visual layout quality. The results demonstrate a substantial

reduction in iteration time and total number of iterations, with only minor degradation in visualization accuracy.

It is shown that the proposed approach enables interactive exploration of large graphs while preserving key structural patterns, such as hubs and transaction chains, which are critical for financial monitoring tasks. The results can be applied in the development of visual analytics systems to support decision-making in the AML domain.

Keywords: graph visualization, force-directed algorithms, Fruchterman–Reingold, fraudulent transactions, AML, transaction networks.

References

1. Aiding humans in financial fraud decision making. (2024). *arXiv*. https://doi.org/10.48550/arXiv.2408.14552 [in English]
2. IBM. (n.d.). AMLSim: Anti-money laundering simulator. GitHub. https://github.com/IBM/AMLSim [in English]
3. An advanced network visualization system for financial crime detection (VISFAN). (2011). *2011 IEEE Pacific Visualization Symposium*, 121–128. https://doi.org/10.1109/PACIFICVIS.2011.5742391 [in English]
4. Barnes, J., & Hut, P. (1986). A hierarchical O(N log N) force-calculation algorithm. *Nature*, 324, 446–449. https://doi.org/10.1038/324446a0 [in English]
5. Brinkmann, G., Graser, J., & Gutwenger, C. (2017). Exploiting GPUs for fast force-directed visualization of large networks. *2017 IEEE International Conference on Parallel Processing (ICPP)*, 382–391. https://doi.org/10.1109/ICPP.2017.47 [in English]
6. Financial Action Task Force. (n.d.). Recommendations. https://www.fatf-gafi.org/recommendations.html [in English]
7. Fruchterman, T. M. J., & Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11), 1129–1164. https://doi.org/10.1002/spe.4380211102 [in English]
8. Godiyal, A., Hoberock, J., & Garland, M. (2009). Rapid multipole graph drawing on the GPU. *In Lecture Notes in Computer Science* (Vol. 5876, pp. 90–101). Springer. https://doi.org/10.1007/978-3-642-00219-9_10 [in English]
9. Mayr, E., Kohlhammer, J., Filzmoser, P., & Miksch, S. (2017). The value of visualization. *IEEE Transactions on Visualization and Computer Graphics,* 23(1), 681–690. https://doi.org/10.1109/TVCG.2017.2744758 [in English]
10. Network analytics for anti-money laundering — A systematic literature review and experimental evaluation. (2024). *arXiv*. https://doi.org/10.48550/arXiv.2405.19383 [in English]

11. Purchase, H. C. (1997). Which aesthetic has the greatest effect on human understanding*? In Lecture Notes in Computer Science* (Vol. 1353, pp. 248–261). Springer. https://doi.org/10.1007/3-540-63938-1_67 [in English]
12. Gove, R. (2019). A random sampling O(n) force-calculation algorithm for graph layouts. *Computer Graphics Forum*, 38(3), 1–12. https://doi.org/10.1111/cgf.13724 [in English]
13. Singh, K., & Best, P. (2019). Anti-money laundering: Using data visualization to identify suspicious activity. *International Journal of Accounting Information Systems*, 34, 100419. https://doi.org/10.1016/j.accinf.2019.06.001 [in English]
14. Statistical insights into anti-money laundering: *Analyzing large-scale financial transactions*. (2025). Zenodo. https://doi.org/10.5281/zenodo.18107429 [in English]
15. Oztas, B. (n.d.). Synthetic transaction monitoring dataset (SAML-D). Kaggle. https://www.kaggle.com/datasets/berkanoztas/synthetic-transaction-monitoring-dataset-aml [in English]
16. Topology-driven force-directed algorithms. (2010). *In Lecture Notes in Computer Science* (Vol. 6502, pp. 256–267). Springer. https://doi.org/10.1007/978-3-642-18469-7_15 [in English]
17. Ware, C., Purchase, H. C., Colpoys, L., & McGill, M. (2002). Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2), 103–110. https://doi.org/10.1057/palgrave.ivs.9500013 [in English]
18. Weber, M., Chen, J., Suzumura, T., Pareja, A., Ma, T., Hirose, H., & Kaler, T. (2019). Scalable graph learning for anti-money laundering: A first look. *arXiv*. https://doi.org/10.48550/arXiv.1812.00076 [in English]
19. MDN Web Docs. (n.d.). WebGPU. https://developer.mozilla.org/en-US/docs/Web/API/WebGPU_API [in English]
20. Zola, F., et al. (2026). Synthetic pattern generation and detection of financial crimes. *arXiv*. https://doi.org/10.48550/arXiv.2601.21446 [in English]

Матеріал надійшов до редакції 27.04.2026

Прийнято до друку 13.05.2026 р.