

СПОСІБ ПІДВИЩЕННЯ ШВИДКОДІ ВИЯВЛЕННЯ АНОМАЛІЙ В ЧАСОВИХ РЯДАХ ПОКАЗНИКІВ РОБОТИ ХМАРНИХ ТЕХНОЛОГІЙ

¹Національний університет «Львівська політехніка»

В сучасній хмарній ІТ інфраструктурі необхідно відстежувати сотні показників (метрик) на наявність аномалій для забезпечення стабільної роботи ІТ-систем. Багато з цих показників мають мультисезонний характер, і вимагають декомпозиції на компоненти: тренд, сезонність та лишки. Більшість з відомих методів декомпозиції потребують великих обчислювальних ресурсів, зокрема, широко застосовуваний метод мультисезонної тренд декомпозиції з використанням локально обчислювального згладжування діаграми розсіювання (Multiple Seasonal-Trend decomposition using Loess/MSTL). У статті пропонується альтернативний підхід виявлення аномалій у великому масиві показників хмарної інфраструктури Google Cloud Platform, ефективніший з погляду використання обчислювальних ресурсів для застосування в режимі близькому до реального часу. Сутність підходу полягає у виявленні лінійно залежних один від одного показників, моделюванні їхньої трендової та сезонної компоненти за допомогою лінійної регресії від інших показників, та пошуку аномалій у лишковому компоненті. Це дозволяє з меншими витратами обчислювальних ресурсів знаходити аномальні значення показника, та оптимізує роботу центрів моніторингу хмарної інфраструктури. В роботі продемонстровано підхід до виділення набору показників, придатних до застосування декомпозиції в поєднанні з лінійною регресією, розрахунок оптимальної довжини часового ряду для знаходження коефіцієнтів регресії та критерії виявлення аномальних значень показників. Експериментально перевірено ефективність поєднання декомпозиції та лінійної регресії в реальній хмарній інфраструктурі високонавантажених веб-застосунків в контейнерах Google Cloud Platform. Виявлено, що критерії пошуку аномалій на базі нерівності Чебишова є найприйнятнішими у таких завданнях. Знайдено зв'язок архітектури системи та статистичних особливостей показників, що впливає на їх вибір в запропонованому підході до оптимізації.

Ключові слова: декомпозиція часових рядів, MSTL, виявлення аномалій, хмарні ІТ-системи, лінійна регресія.

Вступ

Українські дослідники зазначають важливість комплексного підходу до моніторингу ІТ [1]. Саме це підтверджується дослідженнями, виконаними спеціалістами відомих Інтернет-гігантів й інших великих організацій, таких як Splunk [2] та ін. Детальний перелік останніх досліджень подано в публікації [3]. Перспективним є використання декомпозиції часових рядів з метою відділення тренду, сезонного компонента та лишків, з подальшим аналізом відхилень в лишках (аномалій) за критерієм Чебишова [4]. Серед популярних методів декомпозиції переважають такі як Multi-seasonal STL [5], Facebook Prophet [6], мультисезонної авторегресійної інтегрованої моделі ковзної середнього (Multi-SARIMA) [7]. Але і вони вимагають доволі великих обчислювальних ресурсів на кожний показник [8]. Цікавим є застосування лінійної регресії для декомпозиції часового ряду вимірів одного показника, яка застосовується в роботі [9]. Особливістю згаданої праці є накладання обмежень на гладкість тренду та сезонних компонент, та знаходження коефіцієнтів регресії, враховуючи накладені обмеження. Більшість робіт у разі декомпозиції в компоненті лишків не розділяють білий шум та аномалії, цей недолік спробували подолати фахівці, створивши алгоритм RobustSTL [10], де явно розраховується аномальна складова.

За наявності однієї періодичної компоненти цілком достатньо методу сезонної тренд-декомпозиції з використанням локально обчислювального згладжування діаграми розсіювання (STL), до того ж він

є доволі популярним серед українських дослідників. Вони використовують його і як основний метод декомпозиції для вирішення наукової проблеми [11], також як метод розвідувального аналізу даних [12], [13]. Зауважимо, що для розвідки можливе застосування й інших методів як от Холта–Вінтерса або експотенційного простору станів (ETS), а для моделювання (прогнозу) значень часового ряду — авторегресивні методи, такі як авторегресії та ковзного середнього (ARMA) [14]. У підході, що пропонується в цій роботі, не використовувались авторегресивні методи, через їх чутливість до наявності аномалій в навчальній вибірці, і необхідності збільшувати витрати ресурсів на підготовку навчальної вибірки. Ознайомитись з практичним способом конструювання навчальної вибірки з врахуванням наявних аномалій для моделі ARMA можна в роботі харківських науковців [15].

Важливим етапом у разі декомпозиції часового ряду є визначення періодів присутніх у ньому. У найпростішому випадку, це можна зробити експертним шляхом, добре знаючи прикладну галузь. Цілковито природними є періоди в 24 години, 7 діб, тощо. Але можуть зустрічатись і цикли іншої довжини. Прикладами автоматизованого визначення періоду в поєднанні із задачею декомпозиції на базі алгоритму Facebook Prophet є праці вітчизняних науковців Вінницького національного технічного університету [16], [17].

Питання оптимізації декомпозиції часових рядів в задачах моніторингу розглядається у багатьох наукових працях, але вони в переважній більшості не враховують залежність між значеннями показників. Серед них доречно зауважити:

- зменшення кількості замірів, за рахунок розрахованого зменшення частоти вибірки [18]. В деяких операціях це дозволяє зберегти до 80 % обсягу зібраних даних, але це збільшує період в який не можливо надійно виявляти аномалії;
- застосування приблизного алгоритму для розрахунку тренду, виділення сезонності після вирачування приблизного тренду, і наступний розрахунок точного тренду [8];
- штучне виділення більш низьких частот за рахунок згладжування оригінального високочастотного ряду, розрахунок їх сезонних характеристик, і відновлення характеристик більш високих частот методом вирішення оптимізаційної задачі [19].
- інкрементне обчислення тренду та сезонного компонентів, націлене на обробку показників в реальному часі в потоковому режимі [2].

Метою роботи є дослідження можливості зменшення використання обчислювальних ресурсів в процесі виявлення аномалій в часових рядах показників роботи хмарної інфраструктури за рахунок прогнозування складових тренду та сезонності одних показників через лінійну залежність від інших показників.

Огляд математичної моделі

Першим кроком у підході, який пропонується в цій статті, є вибір множини показників роботи хмарної інфраструктури, до яких можливе ефективне застосування комбінації декомпозиції та лінійної регресії. Для практичного застосування в цій роботі пораховано матрицю взаємної кореляції показників (коефіцієнт кореляції Пірсона), відфільтровані метрики з попарним значенням коефіцієнта кореляції більше ніж 0,5, та знайдено найбільший зв'язний підграф цієї множини. Можливі і інші складніші підходи, наприклад, в кілька етапів вибрати мінімальне ядро показників, які дають найбільшу інформацію про IT-систему [20]. Це є прикладом задачі міні-макса на графі, і вона є *NP*-повною, тому що зводиться до «задачі про кліку» на графах [21]. Також серед інших підходів до вибору мінімального ядра метрик можна використати алгоритми відбору властивостей (features) в ШІ, а саме методи фільтрації та обрамлення (filter and wrapping), що розглянуті в [22].

Другим кроком до вибраної множини показників за певний період часу застосовується декомпозиція на складові тренду, сезонності та залишку методом MSTL. Запропонований в статті спосіб підвищення обчислювальної ефективності пошуку аномалій не залежить від конкретного підходу до декомпозиції. MSTL вибрано з міркувань практичної застосовності в реальних проектах хмарної інфраструктури Google, де виробник рекомендує для обробки великих масивів даних ґрунтуватись на технології паралельних обчислень Google Airflow, Dataflow, BigQuery та мову програмування Python. Для неї найпопулярніша бібліотека статистичних функцій, Statsmodels, саме імплементує MSTL. Також в технології Google BigQuery наявний метод STL [23] (основа MSTL), який теж можна використати для декомпозиції даних, тільки лише з одною сезонною компонентою (або де впливом інших сезонів можна знехтувати). Огляд наукової літератури показав, що MSTL лишається вибором великої частини дослідників, через задовільність його результатів у порівнянні з іншими методами

і одночасній простоті в реалізації. Також він є стійким до наявності викидів (outliers) бо використовує локальне згладжування (LOESS). Хоча оригінальна стаття про MSTL не містить розрахунку обчислювальної складності, з дизайну алгоритму випливає, що він використовує STL [23], і також базується на LOESS [24], що має складність в кращих випадках — $O(N * \log N)$ для випадку одновимірного часового ряду [25], де N — кількість елементів часового ряду. Складність розрахунку показників лінійної регресії в її популярних реалізаціях становить $O(N^2)$, хоча зустрічаються і показники ліпші завдяки приблизним розрахункам із завчасно заданою допустимою помилкою матричних операцій [26], або за рахунок збільшення використання оперативної пам'яті [27]. Складність прогнозу залежної величини за вже відомими коефіцієнтами лінійної регресії $O(N)$, в $O(\log N)$ разів ефективніше за декомпозицію методом MSTL. Отже, застосування методу лінійної регресії є припустимим лише за умови, що часовий інтервал для визначення показників лінійної регресії є значно меншим за часовий проміжок, на якому проводиться пошук аномалій. В нашому експерименті розрахунок лінійної регресії було достатньо зробити для часового ряду довжиною в 6 годин та ефективно застосовувати для періоду в 7 діб.

Введемо поняття часового ряду \bar{X} для показника хмарної ІТ-системи — μ . Це набір спостережень $\mu_i \in \mathbb{R}$, визначених в моменти часу $t_i \in S, i = \overline{1, N}$, t_1 — початковий момент часу; S — деяка дискретна множина значень часу; $t_{i+1} - t_i$ константа, тобто це ряд точок даних, проіндексованих в хронологічному порядку через однакові часові проміжки:

$$\bar{X} = \{\mu_i\}_{i=1}^N = \{\mu_1, \mu_2, \dots, \mu_N\} = \{\mu(t_1), \mu(t_2), \dots, \mu(t_N)\}.$$

Очевидно, що \bar{X} можна вважати вектором з кількістю координат N , та задати операції суми й різниці часових рядів, як відповідно суму, чи різницю значень координат. Розглянемо декомпозицію часового ряду \bar{X} на суму кількох часових рядів (компонентів): тренду \bar{T} , сезонні $\bar{S}_j, j = \overline{1, L}$ (де L — кількість сезонних компонент, наприклад, день, тиждень, місяць) та часовий ряд лишків \bar{R} . Тренд репрезентує залежність значень функції від часу, що змінюються неперіодично й повільно. Сезонні компоненти представляють значення функції часу з відомими періодами. Лишки є незалежними реалізаціями випадкової величини ε , що набуває значень в \mathbb{R} , має нульове математичне сподівання й дисперсію σ_ε^2 . Маємо $\bar{X} = \bar{T} + \bar{S}_1 + \bar{S}_2 + \dots + \bar{S}_L + \bar{R}$, тобто в момент часу t_i значення показника дорівнює $\mu(t_i) = \tau(t_i) + \theta_1(t_i) + \theta_2(t_i) + \dots + \theta_L(t_i) + \varepsilon(t_i)$. А часові ряди тренду, періодів та лишків відповідно: $\bar{T} = \{\tau_i\}_{i=1}^N$; $\bar{S}_j = \{\theta_{ji}\}_{i=1}^N$; $\bar{R} = \{\varepsilon_i\}_{i=1}^N$.

В роботі [25] зауважується, що для декомпозиції методом MSTL правильні такі твердження:

– $\bar{T} = \tilde{T} * \bar{X}$, де \tilde{T} матриця розміром $N \times N$, із значеннями в \mathbb{R} , а операція «*» — добуток матриці та вектора. Також сезонні компоненти є добутками відповідних матриць та вектора, яким є \bar{X}_i .

– $\bar{S}_j = \tilde{S}_j * \bar{X}$, де \tilde{S}_j — матриці розміром $N \times N$, із значеннями в \mathbb{R} , $j = \overline{1, L}$ (де L — кількість сезонних компонент, наприклад, день, тиждень, місяць).

Лишки визначаються за формулою $\bar{R} = \bar{X} - \bar{T} - (\sum \bar{S}_j)$. Метод MSTL не гарантує нормальний розподіл випадкової величини ε , але на практиці часто припускають нормальну розподіленість лишків і отримують задовільні результати у пошуку аномалій. У разі великого відхилення від нормальності лишків в роботі [28] застосували цікавий прийом — декомпозицію не оригінального часового ряду, а часового ряду, побудованого з квадратних коренів значень оригінального часового ряду, що зменшує дисперсію лишків і наближає їх розподіл до нормального. Серед інших сильних припущень, які дослідник має зробити для застосування математичного апарату в цій роботі — це гомоскедастичність часового ряду лишків, адже це значно спрощує теоретичне моделювання. Необхідно відстежувати зміну дисперсії лишків і у разі суттєвого відхилення це буде важливим сигналом для перерахунку моделі. Інші корисні методи трансформації даних розглянуті в [28] і їх варто спробувати, якщо залишковий компонент сезонної декомпозиції не відповідатиме критеріям нормального розподілу.

Продемонструємо застосування лінійної регресії для розрахунку компонентів декомпозиції. Припустимо, що є набір показників хмарної ІТ системи, що мають в своїй природі один випадковий

процес та пов'язані лінійною залежністю. Це цілком можливо, наприклад коли на кожний запит користувача іде кілька запитів до бази даних, тоді показник кількість запитів до бази даних буде мати лінійну залежність від показника кількість запитів користувача. Будемо застосовувати наш підхід оптимізації пошуку аномалій лише до вимірювань саме таких показників.

Розглянемо два часових ряди показників \bar{X} та \bar{Y} , що є вимірюваннями показників $\{\mu_i\}_{i=1}^N$ та $\{\gamma_i\}_{i=1}^N$. Їх трендові компоненти будуть $\{\tau_i\}_{i=1}^N$ та $\{\pi_i\}_{i=1}^N$, а сезонні $\{\theta_{ji}\}_{i=1}^N$ та $\{\xi_{ji}\}_{i=1}^N$. Змоделюємо залежну величину тренду τ величини γ через лінійне перетворення від π — тренду μ : $\tau_i = \beta_\tau * \pi_i + \alpha_\tau + \epsilon_i$. Де $\beta_\tau, \alpha_\tau \in \mathbb{R}$ — коефіцієнти лінійної регресії, ϵ_i — похибка регресії — випадкова величина з \mathbb{R} , з математичним сподіванням 0 та дисперсією σ_ϵ^2 . Коефіцієнти β_τ, α_τ отримаємо відомим методом найменших квадратів. Так само змоделюємо сезонні компоненти $\xi_{ji} = \beta_{j\xi} * \theta_{ji} + \alpha_{j\xi} + \omega_{ji}$, де для кожного сезонного компонента декомпозицій свої коефіцієнти лінійної регресії $\beta_{j\xi}, \alpha_{j\xi} \in \mathbb{R}$, та похибки ω_{ji} , що є випадковими величинами з математичними сподіваннями 0 та дисперсіями $\sigma_{j\omega}^2$.

Для часового ряду \bar{Y} на відріжку часу $i > N$ замість декомпозиції методом MSTL використовуватимемо моделювання тренду та сезонної компоненти методом лінійної регресії з використанням коефіцієнтів β_τ, α_τ тренду та сезонних $\beta_{j\xi}, \alpha_{j\xi}$. Декомпозицію ж ряду \bar{X} продовжимо робити методом STL. Будемо мати $\hat{\tau}_i = \beta_\tau * \pi_i + \alpha_\tau$, $\hat{\xi}_{ji} = \beta_{j\xi} * \theta_{ji} + \alpha_{j\xi}$, $j = \overline{1, L}$ (де L — кількість сезонних компонент). Виразимо величину $\gamma(t_i) = \hat{\tau}(t_i) + \epsilon(t_i) + \hat{\xi}_1(t_i) + \hat{\xi}_2(t_i) + \dots + \hat{\xi}_L(t_i) + \sum \omega_j(t_i) + \varepsilon(t_i)$. Згрупуємо всі випадкові величини лишків у величину $\rho(t_i) = \epsilon(t_i) + \sum \omega_j(t_i) + \varepsilon(t_i)$. Для лишків $\rho(t_i)$ маємо, що це реалізації випадкової величини з нульовим сподіванням та дисперсією $\sigma_\rho^2 = \sigma_\epsilon^2 + \sum \sigma_{j\omega}^2 + \sigma_\varepsilon^2$ (через незалежність випадкових величин ϵ , ε , ω_j дисперсія їх суми дорівнює сумі дисперсій).

Так отримуються ключові характеристики для значень випадкової величини $\rho(t_i) = \gamma(t_i) - \hat{\tau}(t_i) - \sum \hat{\xi}_j(t_i)$ і можливо застосування критеріїв пошуку аномалій серед значень цієї величини на значеннях часу t_i для $i > N$.

Розглянемо кілька підходів для відбору аномалій. Позначимо кількість подій за період (добу) — D , а математичне очікування кількості інцидентів ІТ-систем за період (з історичних даних) — d . Згідно з нерівності Чебишова маємо

$$P\{|\rho - E[\rho]| < \delta\} \geq 1 - \sigma_\rho^2 / \delta^2, \quad \forall \delta > 0,$$

або

$$P\{|\rho - E[\rho]| > \delta\} \leq \sigma_\rho^2 / \delta^2.$$

Візьмемо $\delta = k\sigma_\rho$, $k \in N$, врахуємо $E[\rho] = 0$ і отримаємо $P\{|\rho| > k\sigma_\rho\} \leq 1/k^2, \quad \forall k \in N$.

Можемо припустити, що великі значення розрахованих лишків декомпозиції $\gamma(t_i)$ зустрічаються з малою ймовірністю ($P = d/D$) і означають збій в ІТ-процесі. Тому виберемо найближче ціле $\dot{k} > \min \sqrt{D/d}$ і будемо показувати аналітику лише тієї події, де значення $|\rho(t_i)| > \dot{k}\sigma_\rho$. Це еквівалентно тому, що аналітик має реагувати на значення $\gamma(t_i)$ в моменти часу $i > N$ такі, де виконується нерівність $|\gamma(t_i) - \hat{\tau}(t_i) - \sum \hat{\xi}_j(t_i)| > \dot{k} \sqrt{\sigma_\epsilon^2 + \sum \sigma_{j\omega}^2 + \sigma_\varepsilon^2}$.

Якщо припустити нормальний розподіл розрахованих лишків декомпозиції $\gamma(t_i)$, то можемо розрахувати так звані Z значення лишків $\rho_z(t_i) = \frac{\rho(t_i)}{\sigma_\rho} = \rho(t_i) / \sqrt{\sigma_\epsilon^2 + \sum \sigma_{j\omega}^2 + \sigma_\varepsilon^2}$, що очікувано має

розподіл $N(0,1)$ з кумулятивною функцією розподілу CDF (кумулятивна функція розподілу). Маємо, що $P = \frac{d}{D} = P(|\rho| > r_{yz0}) = 1 - CDF(r_{yz0})$. Знаходимо відповідно r_{yz0} з таблиць розрахункових значень CDF . Аналітик має реагувати на $|\rho_z(t_i)| > r_{yz0}$, або на значення γ такі, де

$$|\gamma(t_i) - \hat{t}(t_i) - \sum \hat{\xi}_j(t_i)| > r_{yz0} \sqrt{\sigma_\epsilon^2 + \sum \sigma_{j\omega}^2 + \sigma_\epsilon^2}.$$

Серед інших підходів можна виділити застосування правило 3-х коренів з дисперсії (правило 68-95-99,7), і підсвічувати аналітику лише значення γ такі, що

$$|\gamma(t_i) - \hat{t}(t_i) - \sum \hat{\xi}_j(t_i)| > 3\sqrt{\sigma_\epsilon^2 + \sum \sigma_{j\omega}^2 + \sigma_\epsilon^2}.$$

Також для задач, в яких ймовірність двох послідовних аномалій дуже мала допустимим є застосування тестів Граббса чи Дікосна. В умовах великої кількості аномальних значень вони працюють погано [29]. Варто зауважити, що існує і багато інших підходів до пошуку аномалій, наприклад, дослідники з Львівської Політехніки запропонували оригінальний підхід на базі теорії фракталів [30]. В іншій роботі фахівці цього ж закладу використали нейронні мережі для покращання вже наявного алгоритму виявлення аномалій під час моніторингу комп'ютерної мережі [31]. Теорію хаосу для виявлення аномалій в показниках мережевого трафіку застосувала команда вітчизняних науковців у праці [32].

Важливим фактором у виявленні аномалій, є своєчасність такого виявлення. Передбачення аномалій і, відповідно, збоїв ІТ-систем, набагато корисніше ніж констатування аномалії пост-фактум. Науковці Львівського державного університету безпеки життєдіяльності показали, як нейронні мережі можуть ефективніше саме передбачувати початок аномалій, у порівнянні з іншими (статистичними) методами [33].

Результати досліджень

З практичної точки зору вибрано хмарну інфраструктуру сучасного вебзастосунку в Google Cloud Platform (GCP), де автоматично проводиться збір різноманітних стандартних показників роботи цієї інфраструктури (більше 7 тисяч видів показників) та можуть додаватись власні показники за допомогою Google Cloud Monitoring API. Формат імені показника в хмарі Google схожий на URL, наприклад, "spanner.googleapis.com/api/request_count". Особливості побудови схожих систем моніторингу розглянуті в роботі вітчизняних фахівців [34], а деталі оброблення показників Google Cloud Platform в роботі [3].

Для декомпозиції та пошуку аномалій вибрані кілька показників (метрик), і проведена агрегація з інтервалом в 5 хв за період 1 тиждень для побудови моделі та другий тиждень для пошуку аномалій (але є змога застосовувати модель і в подальші періоди). Для експорту метрик написано сценарії на мові Python з використанням Google Monitoring API та Jupiter записника, де запускалась декомпозиція функцією бібліотеки Python sktime StatsForecastMSTL.

Показники GCP мають кілька типів значень, для розрахунку відібрано 252 миттєві показники з дійсними значеннями. Надалі з них відібрали ядро корельованих метрик (за період 4 години, що містив по 240 замірів кожного показника з інтервалом в 1 хв) у вигляді найбільшого зв'язного підграфу показників з взаємним коефіцієнтом кореляції, більшим за 0,5. Отримано ядро з 29 показників. Після того зібрано дані показників вимірювань за тиждень (кожні 5 хв., загалом 2018 вимірювань на показник), і порашовані декомпозиція кожного методом MSTL, та дисперсія лишків після декомпозиції. Лишки виявились не нормально розподілені, що в подальшому унеможливило ефективне використання критеріїв пошуку аномалій за тестом Граббса та трьох дисперсій. Щоб поліпшити якість моделювання декомпозиції методом лінійної регресії обчислено взаємні кореляції між компонентами тренду показників та відібрано 23 метрики з показниками взаємної кореляції трендів більше за 0,5. Для метрик розраховано взаємні показники лінійної регресії тренду та сезонної компоненти (добової).

Подальшим кроком відібрано такі базові метрики для розрахунків показників лінійної регресії:

- метрика, в якій дисперсія тренду найбільша — "spanner.googleapis.com/api/request_count". Мотивацією для цього було те, що вона може містити в собі найбільше інформації. За своєю природою — це кількість запитів до бази даних;

- метрика, в якій найбільша ентропія Шенона — "compute.googleapis.com/instance/cpu/utilization"

(таку ентропію використано в роботі [20]). За своєю природою — це кількість спожитого процесорного часу;

– метрика, для якої медіана суми значень коефіцієнтів детермінації лінійної регресії тренду та сезонної компоненти за всіма іншими метриками є максимальною “prometheus.googleapis.com/go_sched_gomaxprocs_threads/gauge”. Тобто ця метрика може пояснити максимальну кількість невизначеності серед інших метрик. За своєю природою в дизайні системи це відповідає кількості програмних потоків на оброблення запитів користувачів, в архітектурі ІТ-системи з використанням мови Go, що напряду віддзеркалює кількість запитів користувачів.

Розрахунок коефіцієнтів лінійної регресії, особливо сезонної компоненти, дуже сильно залежить від кількості періодів в часових рядах регресії. Має сенс вибирати ділянку даних лише невеликої тривалості в кілька годин, коли ділянка графіка наближена до прямої лінії. Логічно вибирати значення для розрахунків між екстремумами сезонної компоненти. Можливо ще використання плаваючого вікна і пошук найоптимальнішого відрізка базової метрики, який дає найкраще наближення. Експериментальний етап дослідження показав, що від локального мінімуму показника є сенс брати 10-ти годинний відрізок (видно з рис. 1). А від точки старту вимірювань 00-00 за Києвом — 5 годинний відрізок (цей графік не наводиться, але він подібний до рис. 1).

Точність розрахунку трендової компоненти спадає зі збільшенням періоду, але є точка зміни нахилу графіку, яка міститься так само в околі 10-ти годинного періоду (рис. 2).

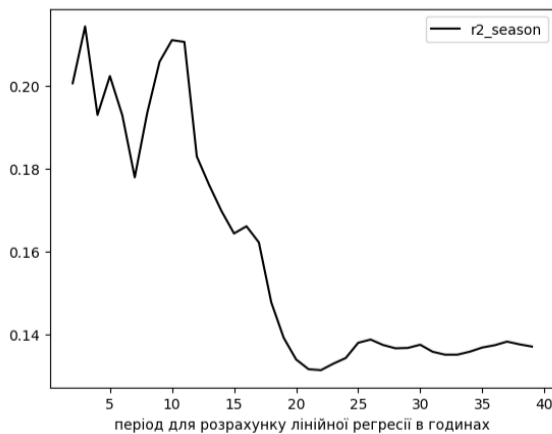


Рис. 1. Графік залежності коефіцієнта детермінації лінійної регресії сезонної компоненти (вісь Y) від періоду для її розрахунку (вісь X)

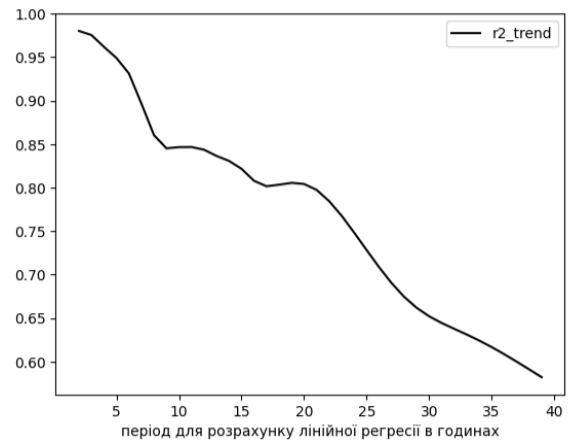


Рис. 2. Графік залежності коефіцієнта детермінації лінійної регресії трендової компоненти (вісь Y) від періоду для її розрахунку (вісь X)

В середньому за оптимальний період можливо брати 10 годин незалежно від особливостей метрики, але якщо врахувати якість результатів лінійної регресії одночасно і для тренду і для сезонної компоненти, то оптимальною виявився період в 6 годин для метрики “prometheus.googleapis.com/go_sched_gomaxprocs_threads/gauge”. Варто очікувати, що в складніших випадках сезонності (з великою кількістю екстремумів та короткими опуклими ділянками) буде менший інтервал для визначення оптимального періоду часу і моделювання лінійною регресією міститиме більшу похибку. В таблиці подані показники регресії (закруглені до 3-х знаків) для цієї метрики як базової. Нижче наведено пояснення, як інтерпретувати дані в табл.

Позначимо виміряні значення в моменти часу $t_i, i = \overline{1..72}$ (що відповідають моментам часу з п'ятихвилинним інтервалом) базового показника “prometheus.googleapis.com/go_gc_gomemlimit_bytes/gauge” як $\{\mu_i\}_{i=1}^N$, $N = 72$, проведемо його декомпозицію методом MSTL з періодом 1 доба і отримаємо часові ряди значень тренду $\{\tau_i\}_{i=1}^N$ і однієї сезонної компоненти $\{\theta_i\}_{i=1}^N$ (бо у нас 1 період в плані експерименту). Далі візьмемо заміри іншої метрики, наприклад, з першого рядка таблиці — “compute.googleapis.com/guest/cpu/load_15m”, позначимо їх $\{\gamma_i\}_{i=1}^N$, і компоненти декомпозиції розраховані методом MSTL — $\{\pi_i\}_{i=1}^N$ та $\{\xi_i\}_{i=1}^N$. Змоделюємо величину $\{\pi_i\}_{i=1}^N$ за допомогою лінійної регресії від $\{\tau_i\}_{i=1}^N$. Під час моделювання отримаємо коефіцієнт детермінації R_τ^2 . Змоделюємо величину

$\{\xi_i\}_{i=1}^N$ за допомогою лінійної регресії від $\{\theta_i\}_{i=1}^N$. В процесі моделювання отримуємо коефіцієнт детермінації R_0^2 . Вони показують якість моделювання. Також під час моделювання отримуємо дисперсії помилок лінійної регресії тренду та сезонної компонент — σ_ϵ^2 та σ_ω^2 . Для метрики σ_ω^2 також після декомпозиції можна визначити дисперсію лишків σ_ϵ^2 . Таблиця містить значення округлені до 3 знаків після нуля. Видно, що тренд моделюється доволі точно — значення R_t^2 близькі до 1, а сезонні компоненти навпаки — з великою помилкою.

**Показники лінійної регресії та декомпозиції ядра метрик для базової метрики
“prometheus.googleapis.com/go_gc_gomemlimit_bytes/gauge”**

Показник	R_t^2	σ_f^2	R_0^2	σ_ω^2	σ_ϵ^2
compute.googleapis.com/guest/cpu/load_15m	0,948	0	0,236	0,017	1,028
compute.googleapis.com/guest/cpu/load_1m	0,948	0	0,18	0,035	2,371
compute.googleapis.com/guest/cpu/load_5m	0,947	0	0,208	0,029	1,842
compute.googleapis.com/guest/cpu/runnable_task_count	0,948	0	0,176	0,035	2,371
compute.googleapis.com/instance/cpu/utilization	0,98	0	0,319	0	0,001
dataflow.googleapis.com/job/aggregated_worker_utilization	0,962	0	0,684	0	0,022
kubernetes.io/container/cpu/limit_utilization	0,996	0	0,011	0	0
kubernetes.io/node/cpu/allocatable_utilization	0,976	0	0,205	0,001	0,001
prometheus.googleapis.com/ xxx_fastapi_endpoint_latency_seconds/gauge	1	0	0,082	0,005	0,053
prometheus.googleapis.com/ go_gc_duration_seconds/summary	1	0	0,059	0	0
prometheus.googleapis.com/ go_gc_gomemlimit_bytes/gauge	0,868	80826	0,896	44629404	12060736463
prometheus.googleapis.com/ go_memstats_buck_hash_sys_bytes/gauge	0,885	51,25	0,385	16015	4038465,163
prometheus.googleapis.com/ go_memstats_mcache_inuse_bytes/gauge	0,909	12607094	0,225	309849757	87102582538
prometheus.googleapis.com/ go_memstats_other_sys_bytes/gauge	0,864	0	0,873	0,03	16,665
prometheus.googleapis.com/ nginx_ingress_controller_ nginx_process_connections/gauge	0,995	0	0,758	3,87	198,52
prometheus.googleapis.com/ process_open_fds/gauge	0,997	0	0,771	1,419	81,409
prometheus.googleapis.com/ pushxxx_http_push_duration_seconds/summary	0,727	0	0,074	0,006	0,607
prometheus.googleapis.com/ pushxxx_http_push_size_bytes/summary	1	83	0,146	532285801	5216829806
prometheus.googleapis.com/ rabbitmq_queue_messages/gauge	0,985	0,001	0,211	105311	232127
prometheus.googleapis.com/scrape_series_added/gauge	0,999	0	0,098	0,02	2,273
spanner.googleapis.com/api/request_count	0,995	0	0,056	0,954	102
spanner.googleapis.com/instance/peak_split_cpu_usage_score	0,898	0	0,02	0,019	2,037

Нижче наведено результати пошуку аномалій. Математичне сподівання кількості збоїв за тиждень з історичних даних взято — 7. Кількість показників компоненти, декомпозиції яких змодельовано за допомогою лінійної регресії — 22 (10 % від можливих 223 показників). Загальна кількість вимірювань цих показників за 7 діб — 44,352. Застосовуючи критерій пошуку аномальних величин показників за нерівністю Чебишова, розраховане k дорівнює 79 (найменше ціле число, що більше за $\sqrt{44,352/7}$). Часові відмітки аномально великих значень лишків показників сконцентровані навколо піків (спалахів) в кількох часових моментах (загалом 233 аномальних значень показників за тиждень).

Під час використання методу пошуку аномалій за допомогою *CDF* стандартного розподілу Гауса

отримано значення параметра. Знайдено 11,106 подій з аномально великими лишками. Це занадто багато для ручної обробки аналітиком (в 48 разів більше аномальних точок ніж із застосуванням нерівності Чебишова). Неefективність використання методів виявлення аномальних значень на базі нормального розподілу зумовлена тим, що лишки мають інший розподіл ймовірності.

Під час використання правила трьох дисперсій для пошуку аномалій знайдено надто велику кількість аномальних подій, яка унеможливило їх ручну обробку аналітиком.

Детальне вивчення знайдених аномалій показало, що в деяких випадках розраховані лишки мали яскраво виражені пікові значення (періодичність), спричинені додатковими сезонними компонентами з іншими періодами, які не враховані під час декомпозиції. На рис. 3—5 подані приклади аномалій:

- реальні аномалії, виявлені критерієм Чебишова (рис. 3—5):
- рис. 3 ілюструє збій в роботі ланцюга компонент, що призвів до різкого зростання кількості необроблених повідомлень;
- рис. 4 ілюструє виявлені аномалії, створені двома незапланованими завданнями з імпортування великого обсягу даних, що призвели до пікового навантаження;
- рис. 5 віддзеркалює спад навантаження через впровадження ефективнішого алгоритму роботи ІТ-компонента.
- хибні періодичні пікові значення (аномалії), знайдені алгоритмом через неврахування періоду довжиною в тиждень показані на рис. 6.

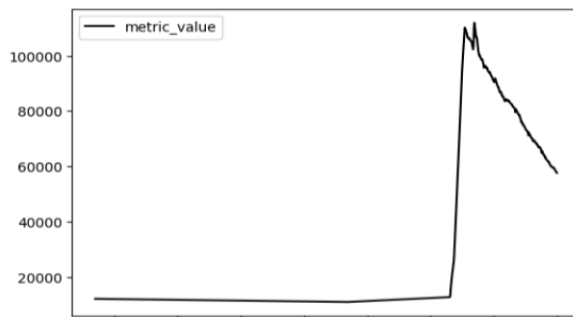


Рис. 3. Приклад виявленої аномалії (неочікуваний пік)

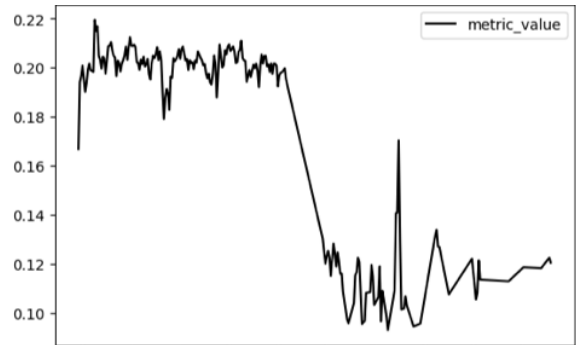


Рис. 4. Приклад виявленої аномалії (два неочікувані піки)

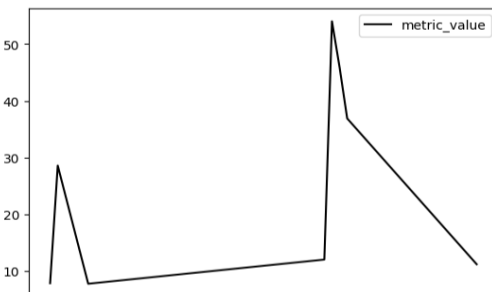


Рис. 5. Приклад виявленої аномалії (неочікуваний провал у значенні метрики)

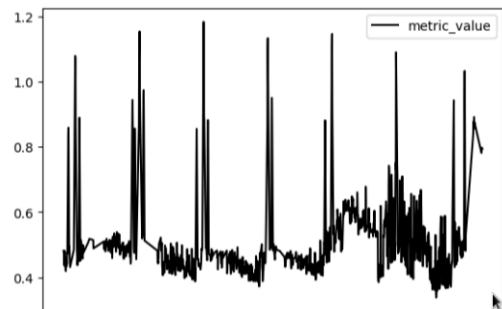


Рис. 6. Приклад хибних аномалій через неврахування додаткового періоду і появу регулярних піків

Через конфіденційність даних про реальну систему замовника не показано, яка саме метрика містила аномалії, в якій період часу і в яких одиницях вимірювалась.

Якщо порівняти отримані результати з іншими відомими техніками можна виділити такі переваги та недоліки:

- в роботі [8] оптимізація виконується за рахунок ефективнішого обчислення тренду (але з більшою помилкою). Спосіб дослідників застосовний для будь-якої кількості метрик, а не лише для тих, які добре моделюються методом лінійної регресії, і вимагає $O(T*N)$ арифметичних операцій, де T – число елементів у вікні медіанного згладжування (періоді). На практиці воно доволі велике і прямий розрахунок методом лінійно регресії потребує лише $O(N)$ арифметичних операцій.

- потокові методи декомпозиції [2], [35], [36] пропонують набагато ефективніші способи декомпозиції складністю $O(1)$, але вони не так ефективно працюють з часовими рядами, в яких наявні стрибки тренду та зсуви в сезонності. Спосіб, запропонований у цій статті, тривіально можна адаптувати для використання в потоковому режимі, замінивши MSTL на один зі згаданих вище

методів, і отримати приріст у швидкості через надзвичайно простий механізм розрахунку прогнозу лінійної регресії.

Висновки

Запропоновано підхід щодо удосконалення прийнятої в індустрії практики пошуку аномальних показників роботи ІТ-систем з використанням декомпозиції часових рядів, який ґрунтується на моделюванні трендової та сезонної компонент часових рядів значень показників за допомогою лінійної регресії.

Установлено, що використання лінійної регресії у поєднанні з декомпозицією методом MSTL часових рядів, створених показниками ІТ хмарної інфраструктури дозволяє ефективніше в $O(\log N)$, з точки зору обчислювальних ресурсів, знаходити аномальні значення показників та підвищує ефективність працівників центрів спостереження за роботою ІТ-систем (N — кількість вимірювань часового показника у вибраному періоді часу). При чому, кількість показників, до яких можливе застосування комбінації MSTL та лінійної регресії в хмарній інфраструктурі становить 10 % всіх можливих показників, що важливо для розрахунку окупності інвестицій в програмне забезпечення для оптимізації завдань моніторингу. Встановлено, що знайдені показники за своєю природою сильно пов'язані з архітектурою додатку та діями користувачів (віддзеркалюють роботу з базою даних та використання певної технології вебсервера). Знайдено оптимальний часовий інтервал для розрахунку показників лінійної регресії — 6 годин (1/4 від періоду декомпозиції в 24 години). Близьким до нього за значеннями отриманих коефіцієнтів детермінації лінійної регресії є десятигодинний відрізок часу (близько 1/2 періоду декомпозиції). Встановлено, що придатним для практичного застосування підходом пошуку аномалій є використання нерівності Чебишова для абсолютних значень компонента лишків в декомпозиції. Використання підходів на базі стандартного розподілу непридатний для практичного застосування через ненормальність розподілу випадкової величини лишків, що призводить до надмірної кількості хибних аномалій (кількість аномальних замірів знайдених за допомогою стандартного нормального розподілу в 48 разів більша за кількість, знайдену за допомогою нерівності Чебишова).

Найсуттєвішою для практичних застосувань є експериментально підтверджена ідея щодо можливості моделювання компонент декомпозиції часових рядів значень показників хмарних ІТ-систем для пошуку аномальних значень. У перспективі подальших досліджень — знаходження точніших способів моделювання компоненти лишків часових рядів не через білий шум з нормальним розподілом, а інші процеси, наприклад, Пуассона.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] I. Danylyuk, and L. Budnyk, “Technology of carrying out a comprehensive IT monitoring of the company,” *Galician economic journal*, vol. 87, no. 2, pp. 40-49, 2024, https://doi.org/10.33108/galicianvisnyk_tntu2024.02.040. Available: <https://galicianvisnyk.tntu.edu.ua/index.php?art=1280>. Accessed: Dec. 11, 2025.
- [2] A. Mishra, R. Sriharsha, and S. Zhong, “OnlineSTL: scaling time series decomposition by 100x,” *Proc. VLDB Endow.*, vol. 15, no. 7, pp. 1417-1425, Mar. 2022, <https://doi.org/10.14778/3523210.3523219>. Available: <https://dl.acm.org/doi/10.14778/3523210.3523219>. Accessed: Dec. 11, 2025.
- [3] Г. Пахаренко, «Використання декомпозиції часових рядів в задачах моніторингу хмарної інфраструктури,» in *Future of Work: Technological, Generational and Social Shifts. Proceedings of the 4th International Scientific and Practical Internet Conference*, May 2025, pp. 160-163.
- [4] T. Mathonsi, and T. L. V. Zyl, “Multivariate anomaly detection based on prediction intervals constructed using deep learning,” *Neural Comput & Applic*, vol. 37, no. 2, pp. 707-721, Jan. 2025, <https://doi.org/10.1007/s00521-021-06697-x>. Available: <https://link.springer.com/10.1007/s00521-021-06697-x>. Accessed: Dec. 11, 2025.
- [5] K. Bandara, R. J. Hyndman, and C. Bergmeir, “MSTL: A Seasonal-Trend Decomposition Algorithm for Time Series with Multiple Seasonal Patterns.” *arXiv*, 2021. <https://doi.org/10.48550/ARXIV.2107.13462>. Available: <https://arxiv.org/abs/2107.13462>. Accessed: Dec. 11, 2025.
- [6] S. J. Taylor, and B. Letham, “Forecasting at scale.” Sept. 27, 2017. <https://doi.org/10.7287/peerj.preprints.3190v2>. Available: <https://peerj.com/preprints/3190v2>. Accessed: Dec. 11, 2025.
- [7] A. T. Williams, R. E. Sperl, and S. M. Chung, “Anomaly Detection in Multi-Seasonal Time Series Data,” *IEEE Access*, vol. 11, pp. 106456-106464, 2023, <https://doi.org/10.1109/ACCESS.2023.3317791>. Available: <https://ieeexplore.ieee.org/document/10256098/>. Accessed: Dec. 11, 2025.
- [8] Z. Zhang, K. Nie, and T. T. Yuan, “Moving Metric Detection and Alerting System at eBay,” *arXiv*, 2020. <https://doi.org/10.48550/ARXIV.2004.02360>. Available: <https://arxiv.org/abs/2004.02360>. Accessed: Dec. 11, 2025.
- [9] A. Dokumentov, and R. J. Hyndman, “STR: Seasonal-Trend Decomposition Using Regression.” *arXiv*, 2020. <https://doi.org/10.48550/ARXIV.2009.05894>. Available: <https://arxiv.org/abs/2009.05894>. Accessed: Dec. 11, 2025.
- [10] Q. Wen, J. Gao, X. Song, L. Sun, H. Xu, and S. Zhu, “RobustSTL: A Robust Seasonal-Trend Decomposition Algorithm for Long Time Series.” *arXiv*, 2018. <https://doi.org/10.48550/ARXIV.1812.01767>. Available: <https://arxiv.org/abs/1812.01767>.

Accessed: Dec. 11, 2025.

- [11] O. Yunkova, and P. Kucher, "Modeling and forecasting of commodity markets based on the decomposition method," *MISE*, no. 101, pp. 182-191, Dec. 2021, <https://doi.org/10.33111/mise.101.15>. Available: <https://mise.kneu.ua/archive/2021/101.15>. Accessed: Dec. 11, 2025.
- [12] O. P. Gozhy, I. O. Kalinina, V. O. Gozhy, and V. V. Dymo, "System approach to forecasting electricity demand based on machine learning," *System technologies*, vol. 3, no. 158, pp. 36-47, Apr. 2025, <https://doi.org/10.34185/1562-9945-3-158-2025-05>. Available: <https://journals.nmetau.edu.ua/index.php/st/article/view/1994>. Accessed: Dec. 11, 2025.
- [13] I. Kalinina, P. Bidyuk, A. Gozhyj, and P. Malchenko, "Combining Forecasts Based on Time Series Models in Machine Learning Tasks," *MoMLeT+DS 2023*, Lviv, June 2023. Available: <https://ceur-ws.org/Vol-3426/paper3.pdf>
- [14] I. Koblianska, L. Kalachevska, S. Minta, N. Strochenko, and S. Lukash, "Modelling and forecasting of potato sales prices in Ukraine," *Agric. resour. econ.*, vol. 7, no. 4, pp. 160-179, Dec. 2021, <https://doi.org/10.51599/are.2021.07.04.09>. Available: <https://are-journal.com/are/article/view/483>. Accessed: Dec. 11, 2025.
- [15] T. Marynych, "Comparative Analysis of Univariate Time Series Modeling and Forecasting Techniques for Short-Term Unstable Data," *Математичне моделювання в техніці та технологіях. Вісник НТУ «ХПІ»*, vol. 6, no. 1128, pp. 63-69, 2017.
- [16] Шмундяк Д. О., Мокін В. Б. «Метод ідентифікації параметрів гармонік та аномалій періодичного часового ряду на основі адаптивної декомпозиції,» *Вісник Вінницького політехнічного інституту*, № 6, с. 46-56, 2023, <https://doi.org/10.31649/1997-9266-2023-171-6-46-56>.
- [17] Лосенко А. В., «Інформаційна технологія прогнозування часового ряду кількості хворих на коронавірус на основі моделі Facebook Prophet,» *Вісник Вінницького політехнічного інституту*, № 5, с. 50-59, 2023. <https://doi.org/10.31649/1997-9266-2023-170-5-50-59>. Дата звернення: груд., 11, 2025.
- [18] A. Anwar, A. Sailer, A. Kochut, and A. R. Butt, "Anatomy of Cloud Monitoring and Metering: A case study and open problems," in *Proceedings of the 6th Asia-Pacific Workshop on Systems*, Tokyo Japan: ACM, July 2015, pp. 1-7. <https://doi.org/10.1145/2797022.2797039>. Available: <https://dl.acm.org/doi/10.1145/2797022.2797039>. Accessed: Dec. 11, 2025.
- [19] L. Yang, Q. Wen, B. Yang, and L. Sun, "A Robust and Efficient Multi-Scale Seasonal-Trend Decomposition," 2021, <https://doi.org/10.48550/ARXIV.2109.08800>. Available: <https://arxiv.org/abs/2109.08800>. Accessed: Dec. 11, 2025.
- [20] A. Singal, D. Pathak, K. Ray, F. George, M. Verma, and P. Moogi, "Metric Criticality Identification for Cloud Microservices." arXiv, 2025. <https://doi.org/10.48550/ARXIV.2501.03547>. Available: <https://arxiv.org/abs/2501.03547>. Accessed: Dec. 11, 2025.
- [21] R. M. Karp, "Reducibility Among Combinatorial Problems," in 50 Years of Integer Programming 1958-2008, M. Jünger, T. M. Lieblich, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 219-241. https://doi.org/10.1007/978-3-540-68279-0_8. Available: http://link.springer.com/10.1007/978-3-540-68279-0_8. Accessed: Dec. 11, 2025.
- [22] R. Kohavi, and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273-324, Dec. 1997, [https://doi.org/10.1016/S0004-3702\(97\)00043-X](https://doi.org/10.1016/S0004-3702(97)00043-X). Available: <https://linkinghub.elsevier.com/retrieve/pii/S000437029700043X>. Accessed: Dec. 11, 2025.
- [23] R. Cleveland, *STL: A Seasonal-Trend Decomposition Procedure Based on Loess*, 1990.
- [24] W. Cleveland, "Robust Locally Weighted Regression and Smoothing Scatterplots," *J. Am. Stat. Assoc.*, vol. 74, no. 368, pp. 829-836, 1979.
- [25] Z. Ouyang, M. Jabloun, and P. Ravier, "STLformer: Exploit STL Decomposition and Rank Correlation for Time Series Forecasting," in *2023 31st European Signal Processing Conference (EUSIPCO)*, Helsinki, Finland: IEEE, Sept. 2023, pp. 1405-1409. <https://doi.org/10.23919/EUSIPCO58844.2023.10290126>. Available: <https://ieeexplore.ieee.org/document/10290126/>. Accessed: Dec. 11, 2025.
- [26] M. Braverman, "The Gradient Complexity of Linear Regression" *Proceedings of Thirty Third Conference on Learning Theory*, PLMR, 2020, pp. 627-647.
- [27] A. Frank, D. Fabregat-Traver, and P. Bientinesi, "Large-scale linear regression: Development of high-performance routines," *Applied Mathematics and Computation*, vol. 275, pp. 411-421, Feb. 2016, <https://doi.org/10.1016/j.amc.2015.11.078>. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0096300315015805>. Accessed: Dec. 11, 2025.
- [28] R. P. Hafen, et al., "Syndromic surveillance: STL for modeling, visualizing, and monitoring disease counts," *BMC Med Inform Decis Mak*, vol. 9, no. 1, p. 21, Dec. 2009, <https://doi.org/10.1186/1472-6947-9-21>. Available: <https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/1472-6947-9-21>. Accessed: Dec. 11, 2025.
- [29] F. E. Grubbs, "Procedures for Detecting Outlying Observations in Samples," *Technometrics*, vol. 11, no. 1, pp. 1-21, Feb. 1969, <https://doi.org/10.1080/00401706.1969.10490657>. Available: <http://www.tandfonline.com/doi/abs/10.1080/00401706.1969.10490657>. Accessed: Dec. 11, 2025.
- [30] М. Бешлей, А. Прислупський, М. Медвецький, і Г. Бешлей, «Інтелектуальна система моніторингу та аналізу трафіку для виявлення атак в програмно-конфігурованих мережах,» *ICTEE*, т. 2, № 1, с. 1-11, 2022, <https://doi.org/10.23939/ictee2022.01.001>. Available: <http://ictee.arleons.com/?journal=ictee&page=issue&op=view&path%5B%5D=ictee-2-1-22&path%5B%5D=ictee-2-1-22-st1>. Accessed: Dec. 11, 2025.
- [31] A. Senyk, Y. Pyrih, and O. Shpur, "Study of the Intelligent Monitoring Algorithm of Qos in the Mass Service Systems," *ICTEE*, vol. 4, no. 2, pp. 103-112, Oct. 2024, <https://doi.org/10.23939/ictee2024.02.103>. Available: <https://science.lpnu.ua/ictee/all-volumes-and-issues/volume-4-number-2-2024/study-intelligent-monitoring-algorithm-qos-mass>. Accessed: Dec. 11, 2025.
- [32] А. Гребенник, «Виявлення та прогнозування рівня загроз для корпоративної комп'ютерної мережі,» *Технічні науки та технології*, т. 2, № 20, с. 175-185, 2020.
- [33] O. M. Shopyskiy, and R. R. Golovaty, "Application of machine learning models for early detection of emergency situations based on streaming big data," *NMetAU Journals*, vol. 4, no. 159, pp. 85-98, May 2025, <https://doi.org/10.34185/1562-9945-4-159-2025-09>. Available: <https://journals.nmetau.edu.ua/index.php/st/article/view/2043>. Accessed: Dec. 11, 2025.
- [34] O. Yu. Tarnovetska, K. P. Hazdiuk, S. M. Balen, and K. M. Dmytrashchuk, "Study of Internet System Connection to Monitoring Using Modern Devops Technologies," *Scientific notes of Taurida National V.I. Vernadsky University. Series:*

Technical Sciences, vol. 1, no. 1, pp. 295-302, 2024, <https://doi.org/10.32782/2663-5941/2024.1.1/44>. Available: https://www.tech.vernadskyjournals.in.ua/journals/2024/1_2024/part_1/46.pdf. Accessed: Dec. 11, 2025.

[35] X. He, Y. Li, J. Tan, B. Wu, and F. Li, "OneShotSTL: One-Shot Seasonal-Trend Decomposition for Online Time Series Anomaly Detection And Forecasting," *Proc. VLDB Endow.*, vol. 16, no. 6, pp. 1399-1412, Feb. 2023, <https://doi.org/10.14778/3583140.3583155>. Available: <https://dl.acm.org/doi/10.14778/3583140.3583155>. Accessed: Dec. 11, 2025.

[36] H. Wang, H. Guo, Z. Zhu, Y. Zhang, Y. Zhou, and X. Zheng, "BacktrackSTL: Ultra-Fast Online Seasonal-Trend Decomposition with Backtrack Technique," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Barcelona Spain: ACM, Aug. 2024, pp. 5848-5859. <https://doi.org/10.1145/3637528.3671510>. Available: <https://dl.acm.org/doi/10.1145/3637528.3671510>. Accessed: Dec. 11, 2025.

Рекомендована кафедрою системного аналізу та інформаційних технологій ВНТУ

Дата надходження: 2.11.2025

Дата прийняття до друку після рецензування: 27.03.2026

Дата публікації: 8.04.2026

Ця робота ліцензується відповідно до

[Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

Пахаренко Гліб Аркадійович — аспірант кафедри комп'ютеризованих систем автоматики Інституту комп'ютерних технологій, автоматики та метрології, e-mail: hlib.a.pakharenko@lpnu.ua . <https://orcid.org/0009-0003-5118-2537>;

Шпак Оксана Іванівна — канд. техн. наук, доцент кафедри комп'ютеризованих систем автоматики Інституту комп'ютерних технологій, автоматики та метрології, e-mail: oksana.i.shpak@lpnu.ua . <https://orcid.org/0000-0002-2093-9029> .

Національний університет «Львівська політехніка», Львів

G. A. Pakharenko¹

O. I. Shpak¹

Anomaly Detection Optimization in Cloud Technology Metric Time Series

In modern cloud IT infrastructures, hundreds of metrics must be monitored to detect anomalies and ensure stable operation. Many of these metrics exhibit multi-seasonal characteristics, requiring decomposition into three components: trend, seasonality, and residuals. However, most known decomposition methods, including the well-known Multiple Seasonal-Trend decomposition using Locally Estimated Scatterplot Smoothing (MSTL), require significant computational resources. This article proposes an alternative, more computationally efficient approach for detecting anomalies in a large array of metrics, particularly for real-time applications. The method's core idea is to identify linearly dependent metrics and model their trend and seasonal components using linear regression with other metrics. Anomaly detection is then performed on the residual component. This reduces the required computational power, thereby optimizing the expenses of cloud infrastructure monitoring centers. This work also demonstrates how to select a suitable subset of metrics for linear regression modeling and decomposition, calculate the optimal time series length for determining linear regression model coefficients, and establish reliable criteria for detecting anomalous metric values. To verify the effectiveness of this linear regression-based decomposition approach, the experiment was conducted on a real cloud infrastructure. The experiment involved a containerized web application with heavy traffic on the Google Cloud Platform. The results showed that the Chebyshev inequality-based approach was the most suitable anomaly detection criterion in this case. Furthermore, a connection was established between the system's architecture, which defines the nature of the metrics, and their statistical properties, which influences their selection for the proposed optimization approach.

Keywords: Time series decomposition, MSTL, anomaly detection, cloud IT systems, linear regression.

Pakharenko Glib A. — Post-Graduate Student, of the Chair of Computerized Automation Systems of Institute of Computer Technologies, Automation and Metrology, e-mail: hlib.a.pakharenko@lpnu.ua . <https://orcid.org/0009-0003-5118-2537>;

Shpak Oksana I. — Cand. Sc. (Eng.), Associate Professor, of the Chair of Computerized Automation Systems of Institute of Computer Technologies, Automation and Metrology, e-mail: oksana.i.shpak@lpnu.ua . <https://orcid.org/0000-0002-2093-9029>