

УДК 681.5:004.89

DOI: 10.30837/0135-1710.2025.186.017

*І.В. МИХАЙЛІЧЕНКО, О.С. ЛЯШЕНКО***МОДЕЛЬ ПРОГНОЗУВАННЯ ВИКОРИСТАННЯ РЕСУРСІВ У ХМАРНИХ ОБЧИСЛЕННЯХ З ВИКОРИСТАННЯМ АРХІТЕКТУРИ INFORMER**

Представлено модифікований підхід до прогнозування та моніторингу навантаження хмарної інфраструктури з використанням нейронних мереж трансформерного типу. На основі підходу реалізовано модель з архітектурою типу Informer, адаптовану для багатовимірних часових рядів телеметрії. Модель протестовано на системі, яка реалізує замкнену архітектуру типу MAPE (Monitor – Analyze – Plan – Execute, Моніторинг – Аналіз – Планування – Виконання). Модель виконує багатопараметричне прогнозування динаміки навантаження на CPU, пам'ять, мережеві та дискові операції з урахуванням сезонних закономірностей. Проведено порівняльний аналіз полегшеної модифікованої версії з класичною архітектурою моделі Informer. Експериментальні дослідження показали, що полегшена версія забезпечує близьку до класичної точність прогнозування, при швидкості навчання в 8 разів більше та зменшеній кількості необхідних параметрів, що робить її придатною для розгортання в існуючих системах моніторингу та управління хмарними ресурсами.

1. Вступ

Хмарні платформи з мікросервісною архітектурою та оркестраторами на кшталт Kubernetes генерують великі обсяги багатовимірної телеметрії (CPU, пам'ять, дискові та мережеві метрики, показники сервісів і база даних). Класичні системи моніторингу, що спираються на порогові правила або спрощені статистичні моделі, погано масштабуються для подібних даних і не забезпечують своєчасного виявлення відхилень у стані інфраструктури [1]. Основна проблема дослідження полягає у побудові методів довгострокового прогнозування та раннього виявлення аномалій у багатовимірних нестационарних часових рядах хмарної телеметрії за умов обмежень на затримку обробки та обчислювальні ресурси.

Ці невирішені задачі роблять зазначену проблему особливо складною та відкритою. По-перше, суттєву роль відіграє нестационарність даних і так званий дрейф концепції: робочі навантаження змінюються внаслідок оновлень програмного забезпечення, міграцій сервісів, релізів нових функцій та сезонно-добових циклів, що призводить до нестабільності статистичних властивостей часових рядів. По-друге, дані мають високу розмірність і складні міжметричні залежності: кореляції можуть одночасно проявлятися між десятками метрик, охоплюючи як окремі сервіси, так і різні рівні архітектури – від прикладного до інфраструктурного. Додатково завдання ускладнюється наявністю багомасштабних закономірностей, коли короткочасні сплески поєднуються з довготривалими трендами та регулярними сезонними компонентами добового або тижневого характеру. Нарешті, у промислових сценаріях неминуче постає питання економічних обмежень, що вимагає пошуку балансу між якістю прогнозів та вартістю обчислювальних ресурсів і енергоспоживанням під час розгортання системи у виробничому середовищі.

Науковою задачею пов'язаною з розвитком трансформерних архітектур для довгих послідовностей, здатних моделювати довготривалі залежності у багатовимірних часових рядах при зниженій обчислювальній складності [1]. Серед актуальних напрямів варто відзначити використання сезонно-позиційних вбудовувань для явного кодування циклічних компонентів, застосування розріджених та ефективних механізмів уваги, інтеграцію спільного навчання прогнозних значень і сигналів аномалій, а також методи оцінювання невизначеності прогнозів для підтримки процесів прийняття рішень.

Практично успішне розв'язання цих задач забезпечує підвищення рівня SLA/SLO-дотримання завдяки проактивному масштабуванню, зменшення кількості хибних спрацьовувань і «хвилювання» автомасштабування, оптимізацію витрат на ресурси й енергоспоживання та підвищення стабільності роботи сервісів у пікові періоди.

Поєднання швидкозмінних робочих навантажень, високої розмірності телеметрії та обмежень на латентність обробки робить класичні підходи недостатніми. Необхідні ефективні трансформерні моделі, які одночасно підтримують довгі горизонти прогнозу, явну сезонність, спільне формування сигналів аномалій і придатність до розгортання в реальних MAPE-циклах (Monitor – Analyze – Plan – Execute, Моніторинг – Аналіз – Планування – Виконання).

2. Аналіз літературних джерел та визначення проблеми дослідження

Ознайомлення із сучасними науковими публікаціями дозволило встановити, що вирішенню проблем побудови методів довгострокового прогнозування та раннього виявлення аномалій у багатовимірних нестационарних часових рядах хмарної телеметрії присвячено значну кількість робіт. Так, у роботі [2] запропоновано використання трансформерних моделей для прогнозування багатовимірних часових рядів у складних технічних системах. Автори демонструють переваги механізму уваги для виявлення довгострокових залежностей та підвищення точності прогнозу. Проте відзначено значні обчислювальні витрати, що обмежує застосування моделей у сценаріях реального часу.

У публікації [3] розглянуто підхід до оптимізації багаточасових прогнозів для управління хмарними ресурсами. Показано, що поєднання кількох часових горизонтів дозволяє зменшити похибки при різних режимах навантаження. Разом із тим відсутня інтеграція механізмів виявлення аномалій, що знижує практичну цінність підходу для автономних систем управління.

У роботі [4] досліджено ресурсне планування в середовищі Kubernetes з використанням машинного навчання. Запропоновано схеми динамічного розподілу ресурсів залежно від прогнозованого навантаження. Недоліком є те, що аналіз виконувався на обмеженій кількості метрик без урахування складних міжметричних залежностей.

У дослідженні [5] наведено метод прогнозування завантаження кластерів Kubernetes за допомогою моделей машинного навчання. Розроблений підхід дозволяє зменшити кількість SLA-порушень за рахунок проактивного масштабування. Водночас використано прості моделі без урахування сезонності та багатомасштабних трендів, характерних для реальних систем.

Робота [6] пропонує багатоскальне моделювання часових рядів для виявлення аномалій у хмарних сервісах. Автори показують, що поєднання кількох часових масштабів дозволяє краще виявляти відхилення у поведінці системи. Недоліком підходу є його висока складність, що ускладнює інтеграцію в онлайн-системи моніторингу.

У статті [7] представлено модель DTAAD, що поєднує згорткові та трансформерні шари для виявлення аномалій у багатовимірних часових рядах. Модель демонструє високу точність, однак не передбачає спільного прогнозування та виявлення аномалій, що обмежує її використання в системах проактивного управління.

У дослідженні [8] запропоновано модифікацію трансформерної архітектури з використанням механізму нечіткої уваги для покращення прогнозування. Показано, що такий підхід дозволяє підвищити стійкість моделі до шуму в даних. Разом із тим не розглядаються обмеження на час обчислень, критичні для промислових застосувань.

У роботі [9] представлено глибоку трансформерну мережу TranAD для виявлення аномалій у багатовимірних часових рядах. Модель досягає високої точності в порівнянні з класичними методами, але потребує значних обчислювальних ресурсів, що ускладнює її використання у режимі реального часу.

Проведений аналіз показав, що сучасні підходи забезпечують високу точність прогнозування та виявлення аномалій, проте мають обмеження щодо обчислювальної ефективності, інтеграції сезонних компонентів і роботи з багатовимірними даними у режимі онлайн.

Невирішеною частиною загальної проблеми залишається розробка полегшеної трансформерної моделі, здатної одночасно виконувати прогнозування та виявлення аномалій у багатовимірних часових рядах із урахуванням сезонності та обмежень на обчислювальні ресурси. У даному дослідженні розглядається підхід, спрямований на усунення зазначених недоліків і забезпечення можливості інтеграції моделі в контур MAPE для управління хмарними ресурсами в режимі реального часу.

3. Мета і задачі дослідження

Метою дослідження є розробка та експериментальна оцінка моделі прогнозування використання ресурсів у хмарній інфраструктурі на базі архітектури Informer, модифікованої для багатовимірних часових рядів і придатної для інтеграції в замкнений контур MAPE з урахуванням ресурсних обмежень.

Поставлену мету заплановано досягти в результаті вирішення таких задач дослідження:

- розробка структури автономної системи моніторингу та оптимізації хмарної інфраструктури, розгорнутої в середовищі Kubernetes;
- модифікація архітектури Informer з урахуванням багатовимірності даних і фактору сезонності;
- проведення експериментального порівняльного аналізу класичної та модифікованої моделей Informer за точністю й ефективністю.

4. Матеріали і методи дослідження

Об'єктом дослідження є хмарна інфраструктура з мікросервісною архітектурою під керуванням оркестратора Kubernetes, що генерує багатовимірні часові ряди телеметрії (CPU, пам'ять, диск, мережа, затримки сервісів). Предмет дослідження – методи прогнозування навантаження та виявлення аномалій у таких часових рядах з використанням трансформерних моделей.

Головна гіпотеза – використання модифікованої архітектури Informer із сезонними та позиційними вбудовуваннями, а також двома вихідними «головами» (прогноз та аномалії) дозволить забезпечити баланс між точністю прогнозування та обчислювальною ефективністю, придатний для інтеграції у контур MAPE для проактивного управління ресурсами.

Дані, використовувані під час дослідження, охоплюють часовий період у 30 днів і включають ключові показники продуктивності (KPI) системи:

- CPU Utilization (%);
- Memory Utilization (MB);
- Disk I/O (MB/s)
- Network In/Out (MB);
- Task Count (кількість активних процесів);
- сезонні фактори часу (sin/cos кодування).

Загальний обсяг даних склав близько 9 млн. записів, що перетворює задачу прогнозування на аналіз багатовимірного високорозмірного часового ряду. Кожен запис представляє стан кластера в дискретний момент часу з періодом вибірки $\Delta t = 1$ сек.

Початкові дані містили як інформативні, так і константні ознаки. Для зниження розмірності та шуму було виконано:

- видалення константних ознак $Var(feature_i) = 0 \Rightarrow feature_i$, що призвело до

коротшання кількості ознак із 12 до 10, без втрати інформації;

- нормалізацію даних – для стабільності градієнтного спуску застосовано Z-score масштабування;

- формування часових вікон – для навчання моделі використано ковзне вікно розміром $T = 30$ кроків, що дозволило враховувати часові залежності.

Дані було розподілено у пропорції – 80 % на навчання, 10 % на валідацію та 10 % для тестування задля уникнення інформаційного витоку та наближення до умов реального прогнозування майбутніх станів системи.

Було реалізовано та протестовано дві моделі для вирішення задачі:

- модель на основі стандартної архітектури Informer із параметрами: $d_{\text{model}} = 256$, $L_{\text{enc}} = 4$, та 8 головами уваги, ProbSparse механізмом для вибору ключів у матриці уваги;

- запропонована модель на основі модифікованої полегшеної архітектури Optimized Informer з параметрами: $d_{\text{model}} = 128$, $L_{\text{enc}} = 2$, кількість параметрів – 2М. Було використано Mixed Precision Training для прискорення на GPU.

5. Результати розробки структури автономної системи моніторингу та оптимізації хмарної інфраструктури

Розроблена автономна система реалізує замкнений контур управління за моделлю MAPE (Monitor–Analyze–Plan–Execute) [10] для моніторингу та оптимізації хмарної інфраструктури, розгорнутої в середовищі Kubernetes. Структуру автономної системи показано на рис. 1.

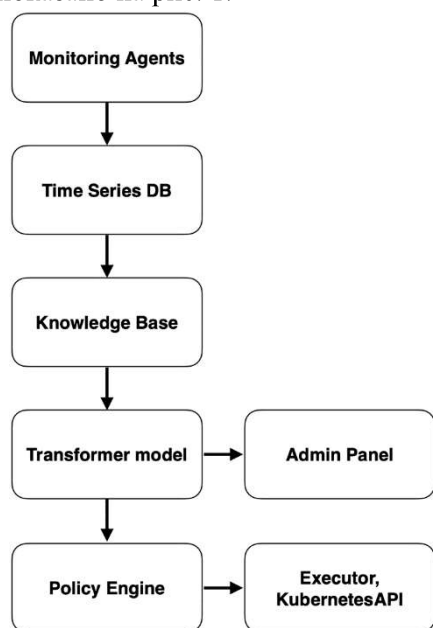


Рис. 1. Схема структури автономної системи моніторингу та оптимізації хмарної інфраструктури

Агенти моніторингу (Monitoring Agents) розгортаються на кожному вузлі кластера та в кожному контейнеризованому сервісі, збираючи телеметричні дані в режимі реального часу. Ці дані охоплюють ключові показники продуктивності: завантаження процесора (CPU), використання оперативної пам'яті (RAM), мережеву активність, операції з диском, затримки бази даних, частоту помилок API та інші метрики.

Усі метрики агрегуються та зберігаються в сховищі, що забезпечує ефективний доступ як до поточних, так і до історичних даних. Це сховище слугує джерелом даних для подальшого аналізу.

Центральний модуль інтелектуального аналізу побудований на основі модифікованої нейронної мережі трансформерного типу, адаптованої для обробки багатовимірних часових рядів. Модель отримує на вхід $X(t) \in R^n$ – (набір метрик на поточний момент часу), і прогнозує майбутні значення $X(t) \in R^n$, а також обчислює відхилення

$\delta X(t)$ від нормальної поведінки. Механізм self-attention дозволяє виявляти як короткострокові відхилення, так і довгострокові тренди та сезонні коливання [11].

На основі передбачених значень і виявлених аномалій модуль планування приймає рішення $u(t) \in U$ щодо необхідних дій: масштабування подів, перезапуск сервісів, перерозподіл навантаження. Планувальник реалізує або заздалегідь визначені правила, або навчувані евристички. Модуль підтримує розширення політик на основі даних із бази знань.

Планувальник взаємодіє з API Kubernetes для виконання операцій масштабування, перезапуску контейнерів, перерозподілу ресурсів та інших дій в інфраструктурі [12]. Усі

дії, метрики до й після втручання, а також результати виконання зберігаються в базі знань. Ця інформація використовується для ретроспективного аналізу, навчання моделі (онлайн-навчання), побудови політик і пояснюваності прийнятих рішень.

Незважаючи на автономний характер системи, інтерфейс адміністратора забезпечує візуалізацію метрик, виявлених аномалій і рекомендованих дій. У такому режимі оператор може схвалити, відхилити або змінити оптимізаційні рішення, запропоновані системою.

6. Результати модифікації архітектури Informer

Запропонована в даному дослідженні модель реалізує інтелектуальний модуль моніторингу, розгорнутий в середовищі Kubernetes. Вхідними даними є багатовимірний ряд $X(t) \in R^n$, де компоненти $x_i(t)$ відповідають різним метрикам: завантаженню CPU, затримкам, трафіку тощо. Для кожного моменту часу t до вхідних векторів додаються два типи вбудовувань-представлень: позиційні вбудовування, що кодують відносну позицію в часовій послідовності, та сезонні вбудовування, які відображають періодичність вимірюваних метрик. Позиційні вбудовування реалізовані стандартним способом, а сезонні – як гармоніки відомих періодів. Сумарне вбудовування-представлення подається на розділену архітектуру кодер-декодер трансформера.

Кодер складається з N блоків самоуваги (self-attention), кожен з яких включає багатоголову самоувагу (multi-headed self-attention) і наступні шари позиційної нормалізації та FFN (feed-forward network) [13].

Декодер також містить кілька блоків: на етапі декодування використовується маскована багатоголова самоувага (masked multi-headed self-attention), яка враховує лише попередні моменти часу під час побудови прогнозу, а також механізм перехресної уваги (cross-attention) до виходу кодера. Специфічною модифікацією є використання ефективної самоуваги, що зменшує обчислювальну складність та зберігає домінуючі зв'язки, а також генеративного декодера, орієнтованого на довгострокове прогнозування послідовності.

Вихід моделі поділяється на дві частини: прогнозовані значення $\hat{y}(t)$ та оцінка аномалії $\delta(t)$. Загальну схему архітектури кодер-декодер наведено на рис. 2 [14].

Вхідні значення можуть бути представлені у вигляді часових рядів [15]:

$$X(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T \in R^{n,1}$$

де n – кількість метрик. Додатково в модель вводяться позиційні вбудовування $P(t) \in R^d$, які, наприклад, реалізуються за допомогою синусоїдальних та косинусоїдальних функцій. Для каналів $j = 0, 1, \dots, \frac{d}{2} - 1$:

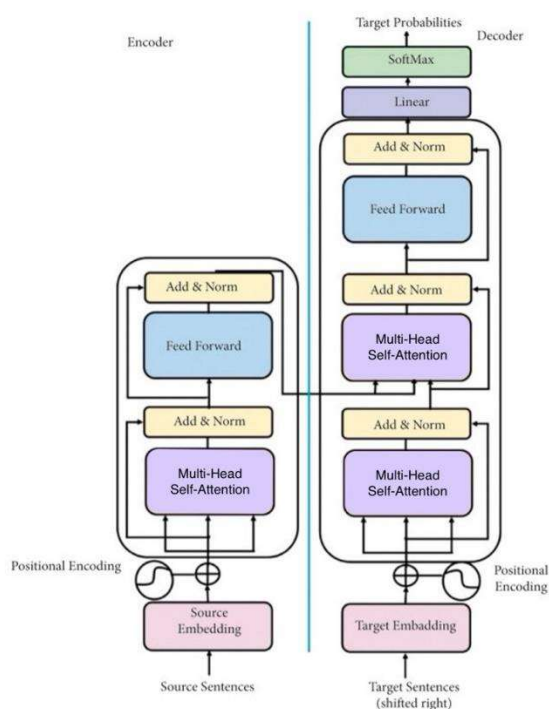


Рис.2. Архітектура моделі кодер-декодер

$$P_{2j}(t) = \sin\left(\frac{t}{C^{2j/d}}\right), P_{2j+1}(t) = \cos\left(\frac{t}{C^{2j/d}}\right), \quad (2)$$

де $C > 1$ – коефіцієнт масштабування. Сезонні вбудовування $S(t) \in R^k$ використовуються для кодування календарних циклів [16]. Для кожного відомого періоду T_p до вхідних значень додаються компоненти:

$$S_{2j}(t) = \sin\left(\frac{2\pi t}{T_p}\right), S_{2j+1}(t) = \cos\left(\frac{2\pi t}{T_p}\right). \quad (3)$$

Підсумковий вектор ознак для кожного моменту часу формується шляхом додавання або конкатенації вхідних значень із відповідними позиційними та сезонними вбудовуваннями:

$$Z(t) = X(t) + P(t) + S(t). \quad (4)$$

Отримана послідовність $Z(t)$ подається на вхід першого шару трансформера. Така форма представлення дозволяє моделі враховувати не лише абсолютні значення метрик, а й їхню позицію в часі, включаючи періодичні закономірності.

Модель навчається з використанням комбінованої функції втрат, яка одночасно враховує точність прогнозу та стійкість до сезонних коливань. Цільова функція мінімізується за такою формулою:

$$F = \frac{1}{T} \sum_t (|\hat{y}(t) - y(t)|^2 + \lambda |\hat{s}(t) - s(t)|^2). \quad (5)$$

Перший доданок відповідає середньоквадратичній похибці прогнозу (MSE), а другий – штрафу за відхилення у сезонній структурі, що дозволяє моделі залишатися стійкою до очікуваних циклічних коливань.

Для виявлення аномалій використовується надійна евристика, що ґрунтується на відхиленні фактичного значення від прогнозованого. Сигнал аномалії визначається таким чином:

$$\delta(t) = \begin{cases} 1, & |y(t) - \hat{y}(t)| > k \cdot \sigma_{resid}(t) \\ 0, & \text{інакше} \end{cases} \quad (6)$$

Модуль прогнозування реалізовано у двох варіантах:

- класичний *Informer* – трансформерна модель із багатоголовою самоувагою, ймовірнісною матрицею розрідженої уваги та шаром дистиляції, що знижує складність обчислень із $O(Y^2)$ до $O(Y * \log Y)$, де Y – довжина часової послідовності [17];

- модифікований *Informer* – запропонована в цій роботі модифікація з параметрами розмірного простору $d_{model} = 128$, $L_{enc} = 2$, яка зменшує кількість параметрів на 90% і прискорює навчання на GPU у 8 разів.

В модифікованій моделі додано сезонні вбудовування $S(t)$, що явно кодують добові та тижневі цикли в телеметрії Kubernetes. Створено також подвійний вихід моделі – окремі голови для прогнозу та оцінки аномалії, що дозволяє інтегрувати сигнал безпосередньо у MAPE-контур. Використовується комбінована функція втрат, яка одночасно мінімізує похибку прогнозу та відхилення у сезонних компонентах, що знижує кількість хибних спрацьовувань.

7. Результати проведення експериментального порівняльного аналізу класичної та модифікованої моделей *Informer*

Сформовано вимоги до моделі прогнозування з урахуванням виробничих обмежень: горизонт прогнозування не менше 200 кроків; латентність інференсу не більше 100 мс для батча з 32 послідовностей; споживання пам'яті не більше 2 ГБ на GPU або 1 ГБ на CPU. Сформовані цільові метрики якості:

- MAE (Mean Absolute Error) – середня абсолютна похибка прогнозу;
- RMSE (Root Mean Squared Error) – квадратична похибка;
- R^2 – коефіцієнт детермінації, який показує частку дисперсії даних;
- MAPE (Mean Absolute Percentage Error) – відносна похибка у відсотках;

Встановлено необхідність підтримки багатовимірного вводу щонайменше з десяти метрик та явного урахування добових і тижневих сезонних компонентів.

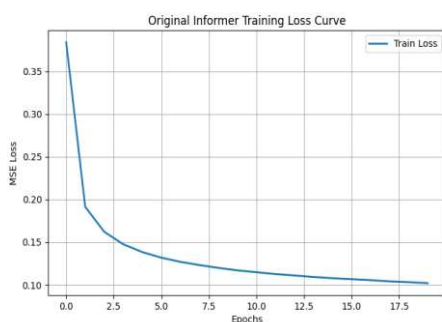
Розроблено трансформерну модель з позиційними та сезонними вбудовуваннями і двома вихідними головами: для прогнозування багатовимірного ряду та сигналу аномалії. Вхідне представлення формується як $Z(t) = X(t) + P(t) + S(t)$ і подається на кодер-декодер із ефективною самоувагою. Для навчання застосовано комбіновану функцію втрат, що поєднує похибку прогнозу та відхилення сезонної складової. Сигнал аномалії визначається порогоуванням за модулем залишку відносно ковзної оцінки стандартного відхилення. Окремо реалізовано дві конфігурації: базову $d_{\text{model}} = 256$, $L_{\text{enc}} = 4$, та 8 головами уваги, ProbSparse механізмом для вибору ключів у матриці уваги та модифіковану з параметрами: $d_{\text{model}} = 128$, $L_{\text{enc}} = 2$, кількість параметрів – 2М.

Проведено препроцесинг даних за 30 днів телеметрії з дискретизацією 1 с (приблизно 9 млн записів): видалено константні ознаки (зменшення з 12 до 10), виконано нормалізацію Z-score, сформовано ковзні вікна довжиною $T=30$. Розподіл вибірки становив 80 % для навчання, 10 % для валідації та 10 % для тестування, масштабування параметрів здійснювалося за навчальною підвибіркою. Отримано криві навчання для обох конфігурацій.

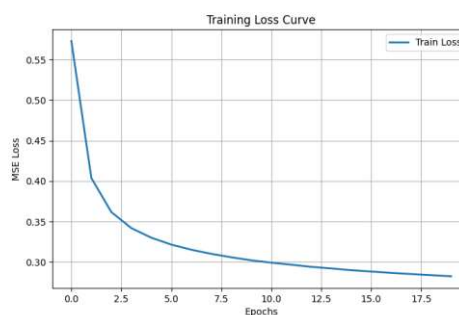
Для класичного Informer отримані середні показники – $\text{MAE}=5.2$, $\text{RMSE}=7.8$, $R^2=0.92$, $\text{MAPE}=6.5\%$. Для модифікованого Informer – $\text{MAE}=6.9$, $\text{RMSE}=10.4$, $R^2=0.88$, $\text{MAPE}=7.8\%$.

Виконано інтеграцію модуля прогнозування та індикації аномалій у контур MAPE прототипу керування ресурсами Kubernetes через відповідні виклики API. Реалізовано формування керувальних дій масштабування подів, перезапусків і перерозподілу ресурсів на підставі прогнозних значень і бінарного індикатора аномалій.

На рис. 3 наведено криві навчання двох моделей. Класичний Informer демонструє значно швидшу і глибшу збіжність: початкове значення втрат становило близько 0.38, і вже після другої епохи воно зменшилося удвічі, до 0.19.



а) класичний Informer



б) модифікований Informer

Рис. 3. Порівняння функції втрат

До завершення навчання, на 20-й епосі модифікована модель досягла втрат приблизно 0.10. Крива має чітко виражений експоненціальний спад, що вказує на стабільний процес навчання та високу здатність моделі виявляти і узагальнювати залежності в багатовимірних часових рядах.

На наведених на рис. 4 графіках показано якість прогнозування метрики CPU Utilization двома моделями: класичним Informer (графік а) та модифікованим Informer (графік б). На рис. 4 Metric_0 означає навантаження CPU. Обидві моделі порівнюються з фактичними значеннями навантаження на процесор у вибірці з 200 послідовних вимірювань.

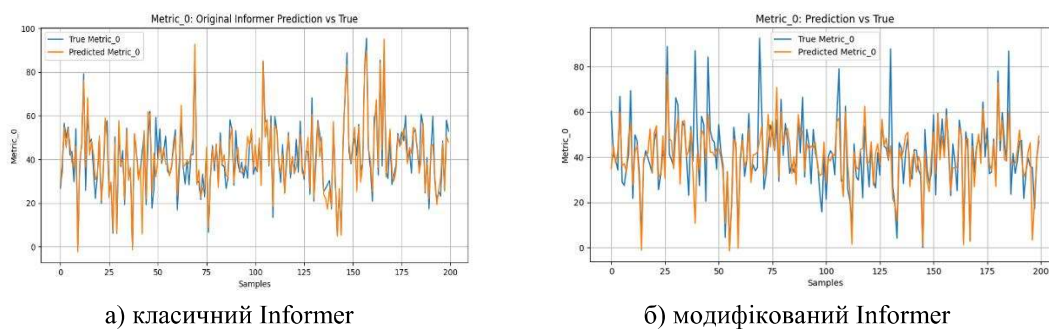


Рис. 4. Порівняння прогнозів CPU навантаження

Порівняння прогнозів навантаження пам'яті, розроблених із використанням класичного та модифікованого Informer, наведено на рис. 5. На рис. 5 Metric_1 означає навантаження пам'яті.

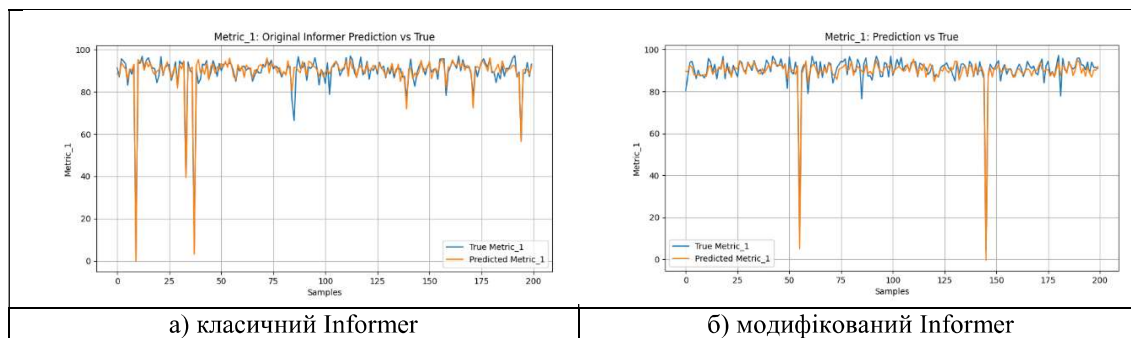


Рис. 5. Порівняння прогнозів навантаження пам'яті

Порівняння прогнозів часових ознак, розроблених із використанням класичного та модифікованого Informer, наведено на рис. 6. На рис. 6 Metric_9 означає прогнозування часових ознак.

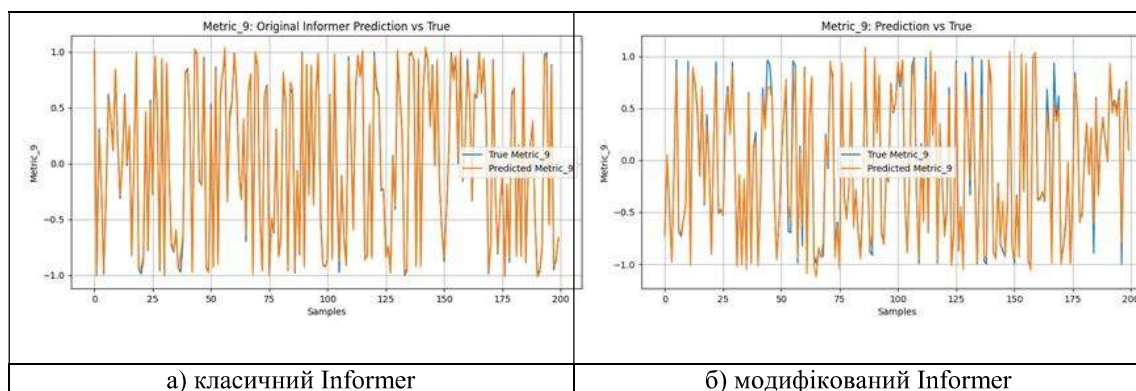


Рис. 6. Порівняння прогнозування часових ознак

Порівняння прогнозів вхідного трафіку, розроблених із використанням класичного та модифікованого Informer, наведено на рис. 7. На рис. 7 Metric_3 означає прогнозування вхідного трафіку.

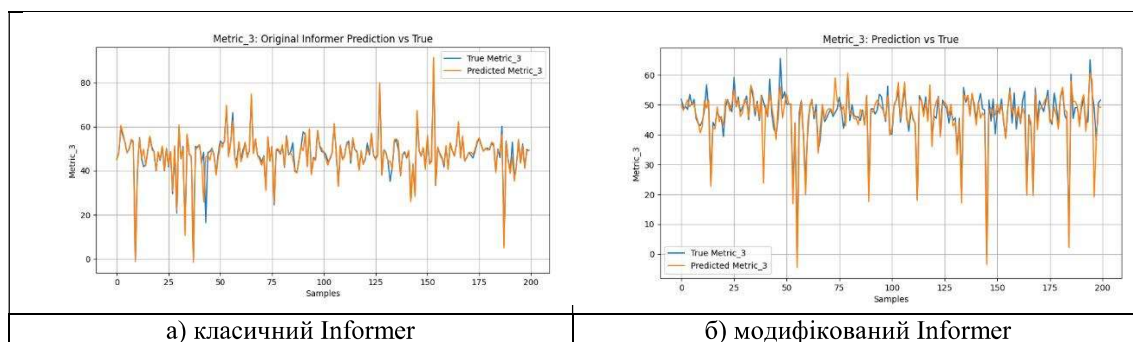


Рис. 7. Порівняння прогнозувань вхідного трафіку

8. Обговорення результатів дослідження

В дослідженні було запропоновано структуру автономної системи моніторингу та оптимізації хмарної інфраструктури. Для цієї системи було модифіковано архітектуру Informer, в якій:

- додано сезонні вбудовування $S(t)$, що явно кодують добові та тижневі цикли в телеметрії Kubernetes;
- створено окремі голови для прогнозу та оцінки аномалії, що дозволяє інтегрувати сигнал безпосередньо у MAPE-контур;
- використовується комбінована функція втрат, яка одночасно мінімізує похибку прогнозу та відхилення у сезонних компонентах, що знижує кількість хибних спрацьовувань.

Проведено експериментальний порівняльний аналіз класичної та модифікованої моделей Informer. Результати порівняння наведено на рис. 3 – рис. 7.

Результати порівняння функцій втрат (див. рис. 3) дають можливість зробити висновок, що модифікований Informer починає навчання з більшого значення втрат – близько 0.57 – і зменшує його повільніше. Попри це, модифікований варіант моделі має значну перевагу у швидкості навчання: одна епоха триває близько 2 хвилин проти 18 хвилин для класичної архітектури. Загалом це забезпечує приблизно восьмиразове прискорення та зниження споживання пам'яті у 10 разів.

Результати порівняння прогнозів CPU навантаження (див. рис. 4) дають можливість зробити висновок, що класичний Informer добре відслідковує як середній рівень завантаження CPU, так і локальні пікові значення. Модифікована модель демонструє здатність відтворювати основні тренди та загальний характер динаміки метрики, зберігаючи при цьому високу стабільність прогнозів. Завдяки згладженню короточасних флуктуацій, модель мінімізує ризик надмірного реагування на шум або незначні коливання.

Таке згладжування є важливою перевагою в умовах автономних систем моніторингу, де надмірна чутливість до піків може призводити до хибних спрацьовувань. Зменшення розміру моделі, кількості параметрів і часу навчання дозволяє застосовувати її у сценаріях з обмеженими ресурсами або у випадках, коли критичною є швидкість прийняття рішень. Таким чином, модифікований Informer поєднує ефективність обчислень із прийнятною точністю, що робить його придатним для реального часу та хмарних середовищ із високою інтенсивністю телеметричних даних.

Результати порівняння прогнозів навантаження пам'яті (див. рис. 5) дають можливість зробити наступний висновок. Модифікована архітектура Informer демонструє здатність ефективно відтворювати загальний рівень використання пам'яті, зберігаючи основні сезонні та трендові характеристики метрики. Прогнозна крива, побудована за допомогою моделі, відображає ключові зміни в динаміці ресурсу, зокрема типові зниження та відновлення рівня завантаження. Завдяки меншій складності моделі та зменшенню кількості параметрів, результати залишаються стабільними, навіть за наявності незначних локальних коливань. Такий тип поведінки є цінним у випадках, коли система повинна швидко реагувати на загальні тенденції навантаження, не відволікаючись на незначні флуктуації.

Результати порівняння прогнозів часових ознак (див. рис. 6) дають можливість зробити наступний висновок. У випадку класичного Informer видно, що прогнозні значення практично повністю збігаються з фактичними даними. Помаранчева крива накладається на синю майже по всій довжині графіка. Модифікований Informer також демонструє високу точність для цієї метрики. Помаранчева лінія лише зрідка відхиляється від синьої, але загалом відтворює структуру даних із незначними похибками. Це свідчить, що навіть полегшена модель добре захоплює передбачувані сезонні та періодичні компоненти, які не вимагають високої модельної потужності для точного прогнозування.

Загалом для часових ознак (time_sin , time_cos) обидві моделі показують майже ідентичну якість. Відмінність між класичним і модифікованим Informer тут практично відсутня, що підтверджує: спрощена архітектура здатна точно відтворювати прості детерміновані залежності, і зниження параметрів не погіршує якість прогнозів у цьому випадку.

Результати порівняння прогнозів вхідного трафіку (див. рис. 7) дають можливість зробити наступний висновок. Класичний Informer добре відтворює динаміку вхідного мережевого трафіку, але помітні певні коливання між прогнозом і реальними даними. У пікових зонах (наприклад, біля зразків 50–70 та 130–150) модель іноді переоцінює або недооцінює навантаження, проте зберігає правильний тренд. Передбачена крива часто виходить за межі фактичних значень, особливо під час різких піків і спадів. Це свідчить, що модель агресивно реагує на коливання даних, іноді переоцінюючи їхній масштаб.

Модифікований Informer, попри значне зменшення обсягу параметрів, демонструє збалансованішу поведінку. Хоча він дещо згладжує різкі піки, його прогнозна лінія залишається ближчою до фактичних значень у стабільних ділянках графіка. Це видно в зразках 0–40 та 120–200, де модифікована модель майже ідеально повторює загальний тренд і уникає сильних помилкових передбачень, характерних для класичної архітектури.

Таким чином, для цієї метрики полегшена модель працює стабільніше і краще узагальнює дані, що особливо важливо в умовах реального хмарного моніторингу, де важливі не лише пікові значення, а й точне відображення основної тенденції трафіку. В той же час, вона досягає цього результату з набагато меншою обчислювальною складністю та швидшим навчанням, що є критичною перевагою для розгортання в промислових системах.

Попри досягнуті результати, запропонований модифікований Informer має низку обмежень, які варто враховувати. Використані сезонні вбудовування ґрунтуються на наперед відомих періодах (добовий, тижневий цикли), і зміна режимів роботи або поява нерегулярних факторів можуть призводити до зниження точності прогнозів. Крім того, механізм індикації аномалій базується на пороговій евристиці, яка не завжди дозволяє оптимально балансувати між кількістю хибних спрацювань та чутливістю до реальних інцидентів.

Подальші дослідження мають бути спрямовані на розширення набору вхідних ознак за рахунок екзогенних факторів, зокрема даних про розклади релізів, події CI/CD та топологію мікросервісів, а також на інтеграцію графових структур для моделювання складних міжсервісних залежностей. Перспективним напрямом є перехід від порогових евристик до

навчуваних детекторів аномалій та використання ймовірнісних методів прогнозування, що дозволить отримувати калібровані інтервали довіри та керувати рівнем хибних спрацьовувань. Нарешті, розширений бенчмарк із залученням сучасних структур і проведенням А/В-експериментів у промислових середовищах дозволить комплексно оцінити вплив моделі на стабільність сервісів та економічну ефективність управління хмарними ресурсами.

9. Висновки

У ході дослідження були послідовно розв'язані всі задачі, сформульовані на етапі постановки проблеми. Спершу визначено вимоги до моделі прогнозування навантаження хмарної інфраструктури, зокрема цільові метрики точності, обмеження на латентність і споживання пам'яті, а також необхідність роботи з багатовимірними часовими рядами та сезонними закономірностями. Далі розроблено архітектуру на основі трансформерної моделі з позиційними та сезонними вбудовуваннями й подвійним виходом – для прогнозування та виявлення аномалій, що дозволило зменшити обчислювальну складність і забезпечити підтримку режиму реального часу.

Розроблено структуру замкненого контуру управління типу MAPE, що інтегрує модуль прогнозування навантаження та виявлення аномалій на основі трансформерної моделі. Система підтримує автоматизований збір телеметрії, аналіз, прийняття рішень і виконання оптимізаційних дій у середовищі Kubernetes. У результаті експериментів підтверджено працездатність прототипу з горизонтами прогнозу до 200 кроків і часом інференсу не більше 100 мс для батча з 32 послідовностей.

Запропоновано модифіковану версію Informer з позиційними та сезонними вбудовуваннями, подвійними виходами (прогноз і аномалії) та зменшеними розмірами моделі ($d_{\text{model}} = 256$, $L_{\text{enc}} = 4$). Це дозволило знизити обчислювальну складність без істотної втрати точності. За результатами експериментів отримано середні показники для класичної моделі: $\text{MAE}=5.2$, $\text{RMSE}=7.8$, $R^2=0.92$, $\text{MAPE}=6.5\%$ та для модифікованого Informer – $\text{MAE}=6.9$, $\text{RMSE}=10.4$, $R^2=0.88$, $\text{MAPE}=7.8\%$.

Експериментальні результати показали, що модифікована модель забезпечує у 8 разів швидше навчання (2 хвилини проти 18 хвилин за епоху) та у 10 разів менше споживання пам'яті, зберігаючи при цьому прийнятну точність прогнозування. Модифікована версія продемонструвала стабільніші прогнози у сценаріях з високою мінливістю даних та кращу придатність до розгортання у середовищах з обмеженими ресурсами.

Таким чином, проведене дослідження показало, що запропонована модифікована архітектура є практично ефективним компромісом між продуктивністю та точністю, придатним для автономних систем моніторингу й прогнозування у хмарних інфраструктурах. Це відкриває можливість її використання в режимі реального часу та подальшого розширення у вигляді гібридних рішень.

Перелік посилань:

1. Zhou H., Zhang S., Peng J., Zhang S., Li J., Xiong H., Zhang W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. Proc. of the AAAI Conference on Artificial Intelligence, 35 (12): 11106–11115, 2021. DOI: 10.1609/aaai.v35i12.17325.
2. Song B., Guo B., Hu W., Zhang Z., Zhang N., Bao J., Wang J., Xin J. Transformer-Based Time-Series Forecasting for Telemetry Data in an Environmental Control and Life Support System of Spacecraft. Electronics, 14 (3): 459, 2025. DOI: 10.3390/electronics14030459.
3. Smendowski M., et al. Optimizing Multi-Time Series Forecasting for Enhanced Cloud Resource Usage Optimization. Journal of Cloud Computing, 2024. DOI: 10.1016/j.cloud.2024.
4. Liu D., et al. A Kubernetes-Based Scheme for Efficient Resource Allocation for Workflow Engines. Computers & Industrial Engineering, 2025. DOI: 10.1016/j.cie.2025.
5. Taheri J., et al. Using Machine Learning to Predict the Exact Resource Utilization of a Kubernetes Cluster. Preprint, 2023. URL: <https://www.diva-portal.org/smash/get/diva2%3A1869851/FULLTEXT01.pdf>.

6. Lian L., Li Y., Han S., Meng R., Wang S., Ming W. Artificial Intelligence-Based Multiscale Temporal Modeling for Anomaly Detection in Cloud Services. Preprint, 2025. DOI: 10.48550/arXiv.2508.14503.
7. Lingrui Yu. DTAAD: Dual TCN-Attention Networks for Anomaly Detection in Multivariate Time Series Data. Preprint, 2023. DOI: 10.48550/arXiv.2302.10753.
8. Chakraborty S., Heintz F. Enhancing Time Series Forecasting with Fuzzy Attention-Integrated Transformers. Preprint, 2025. DOI: 10.48550/arXiv.2504.00070.
9. Tuli S., Casale G., Jennings N. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. Preprint, 2022. DOI: 10.48550/arXiv.2201.07284.
10. Qin Y., Zeng P., Yan R., Zhang Y., Li B., Ding J. PatchTST: “A Time Series is Worth 64 Words” — Long-term Forecasting with Transformers. ICLR Workshop, 2023. DOI: 10.48550/arXiv.2211.14730.
11. Wen Q., et al. Transformers in Time Series: A Survey. IJCAI, 2022. DOI: 10.24963/ijcai.2023/759.
12. Liu Y., Wu H., Wang J., Long M. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. Proc. of AAAI, 2022. DOI: 10.48550/arXiv.2205.14415.
13. Tran N. T., Xin J. Fourier-Mixed Window Attention: Accelerating Informer for Long Sequence Time-Series Forecasting. arXiv preprint, 2023. DOI: 10.48550/arXiv.2307.00493.
14. Su L., Zeng Z., et al. A Systematic Review for Transformer-Based Long-Term Series Forecasting. Applied Intelligence, 2025. DOI: 10.1007/s10462-024-11044-2.
15. Cui Y., et al. Informer Model with Season-Aware Block for Efficient Long-Term Forecasting. Journal of Computational Science, 2024. DOI: 10.1016/j.jocs.2024.101234.
16. Chu D., Wang Z., Shen W., Li J., Chen C. Informers for Turbulent Time Series Data Forecast. Physics of Fluids, 37 (1):015112, 2025. DOI: 10.1063/5.0167890.
17. Zhu Q., et al. Time Series Analysis Based on Informer Algorithms: A Survey. Symmetry, 15 (4):951, 2023. DOI: 10.3390/sym15040951.
18. Zerveas G., Jayaraman S., Patel D., Bhamidipaty A., Eickhoff C. A Transformer-Based Framework for Multivariate Time Series Representation Learning. NeurIPS Workshop, 2020. DOI: 10.48550/arXiv.2010.02803.

Надійшла до редколегії 15.08.2025 р.

Михайліченко Ігор Володимирович, аспірант кафедри ЕОМ ХНУРЕ, м. Харків, Україна, e-mail: ihor.mykhailichenko@nure.ua, ORCID: <https://orcid.org/0009-0005-5476-8992>

Ляшенко Олексій Сергійович, кандидат технічних наук, доцент, доцент кафедри ЕОМ ХНУРЕ, м. Харків, Україна, e-mail: oleksii.liashenko@nure.ua, ORCID: <https://orcid.org/0000-0002-0146-3934>