

УДК (UDC) 519.8:004.72.052.4:004.8

Турчак**Денис Сергійович***Аспірант кафедри прикладної математики**Харківський національний університет імені В. Н. Каразіна, майдан
Свободи, 4, м. Харків, 61022**e-mail: denys.turchak@karazin.ua*<https://orcid.org/0009-0003-9144-8280>**Руккас****Кирило Маркович***Д-р, доцент**Харківський національний університет імені В. Н. Каразіна, майдан
Свободи, 4, м. Харків, 61022**e-mail: rukkas@karazin.ua;*<https://orcid.org/0000-0002-7614-0793>

Вплив архітектури GNN на робастність мережевих маршрутів у сценаріях одиничних відмов вузлів

У роботі розглянуто задачу інтелектуальної маршрутизації (пошуку шляху) в програмно-конфігурованих мережах (SDN) з використанням графових нейронних мереж. з метою підвищення ефективності використання мережевих ресурсів та адаптації до динамічних змін стану мережі (наприклад, завантаженості каналів або затримок). **Актуальність.** Сучасні програмно-конфігуровані мережі (SDN) стикаються зі зростанням обсягів трафіку та підвищеними вимогами до якості обслуговування (QoS). Традиційні алгоритми маршрутизації (наприклад, Дейкстри) є статичними та неефективними в умовах високої динаміки навантаження або раптових змін топології (відмов обладнання). Це призводить до перевантаження каналів, збільшення затримок та втрат пакетів. Використання методів машинного навчання, зокрема графових нейронних мереж (GNN) та навчання з підкріпленням (RL), відкриває нові можливості для створення адаптивних інтелектуальних агентів, здатних оптимізувати маршрутизацію в реальному часі, що робить це дослідження своєчасним та важливим для розвитку телекомунікаційних систем.

Мета. Метою дослідження є підвищення ефективності та надійності передачі даних у мережах SDN шляхом розробки та порівняльного аналізу інтелектуальних методів маршрутизації. Основний фокус зосереджено на дослідженні архітектур графових нейронних мереж (GCN, GAT) та алгоритму Q-Learning для забезпечення адаптивного керування трафіком в умовах змінного навантаження та відмов мережевих вузлів.

Методи дослідження. Методологічна основа роботи базується на комплексному застосуванні теорії графів для формалізації топології мережі, методів глибокого навчання для обробки ознак вузлів та алгоритмів навчання з підкріпленням для прийняття рішень щодо маршрутизації. Експериментальна верифікація запропонованих підходів здійснювалася шляхом емуляції програмно-конфігурованої мережі у середовищі Mininet під управлінням контролера Ryu. Програмна реалізація включала розробку моделей на основі згорткових мереж (GCN) та мереж з механізмом уваги (GAT) з використанням бібліотек глибокого навчання, а також імплементацію агента Deep Q-Learning. Оцінка ефективності алгоритмів проводилася за допомогою порівняльного аналізу ключових метрик якості обслуговування — пропускної здатності, середньої затримки та відсотка втрат пакетів — у сценаріях поступового зростання навантаження та аварійної зміни топології внаслідок відмови обладнання.

Результати. У ході дослідження встановлено, що інтеграція методів машинного навчання дозволяє суттєво покращити параметри передачі даних порівняно з класичним алгоритмом Дейкстри, особливо в умовах високого трафіку, де інтелектуальні агенти забезпечують меншу затримку та стабільність з'єднання. Критичний аналіз стійкості до відмов виявив суттєві розбіжності між досліджуваними архітектурами: модель GCN продемонструвала обмежену здатність до адаптації з показником невдалих спроб маршрутизації на рівні 30%, тоді як архітектура GAT показала кращу гнучкість, формуючи оптимальні шляхи у половині випадків. Найвищу ефективність підтвердив метод Q-Learning, який завдяки динамічній взаємодії із середовищем забезпечив побудову ідеальних маршрутів у 85% експериментів та мінімізував втрати пакетів до 5–7% навіть у критичних ситуаціях, що доводить перевагу підходів Reinforcement Learning над методами Supervised Learning у задачах адаптивного керування мережами.

Ключові слова: Програмно-конфігуровані мережі (SDN), Графові нейронні мережі (GNN), Навчання з підкріпленням (Reinforcement Learning), Інтелектуальна маршрутизація, GCN, GAT, Q-Learning, Якість обслуговування (QoS), Відмовостійкість, Адаптивне керування трафіком.

Як цитувати: Турчак Д. С Руккас. К. М., Вплив архітектури GNN на робастність мережевих маршрутів у сценаріях одиничних відмов вузлів. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління.* 2026. вип. 69. С.101-110. <https://doi.org/10.26565/2304-6201-2026-69-08>

How to quote: D. Turchak, K. Rukkas, "The impact of GNN architecture on the robustness of edge routes in scenarios of single node types", *Bulletin of V. N. Karazin Kharkiv National University, series*

Mathematical modelling. Information technology. Automated control systems, vol. 69, pp. 101-110, 2026.
<https://doi.org/10.26565/2304-6201-2026-69-08> [in Ukrainian]

1 Вступ

Використання нейронних мереж для побудови маршрутів - це одна з найбільш динамічних галузей, у сферах програмно-конфігурованих мереж (SDN), логістики та автономної навігації. Традиційні алгоритми (як-от Дейкстра або A^*) ефективні, але вони часто не встигають за динамічними змінами в реальному часі. Епоха 6G, масивних IoT-мереж та супутникового інтернету висуває нові вимоги до швидкодії та відмовостійкості маршрутизації. У 2025 році пошук "найкоротшого шляху" поступився місцем багатофакторній оптимізації, що враховує затримки, енергоспоживання та пріоритетність трафіку в реальному часі. Використання нейронних мереж дозволяє вийти за межі реактивного керування, впроваджуючи предиктивні моделі, які здатні розв'язувати конфлікти в мережі ще до їх виникнення.

Аналіз сучасних робіт показує, що наукова спільнота остаточно відійшла від використання нейромереж як «чорних скриньок»[3]. Сьогодні домінує підхід інтелектуальної маршрутизації, яка поєднує знання про топологію (GNN) та стратегічне прийняття рішень (DRL).

Сучасні підходи до побудови маршрутів базуються на інтеграції графових нейронних мереж (GNN) для аналізу топології та глибокого навчання з підкріпленням (DRL) для стратегічного прийняття рішень. Ключовим етапом стала еволюція моделей сімейства RouteNet (Fermi, Erlang), які завдяки механізмам Message Passing та Attention забезпечують предиктивне моделювання QoS (затримки, джиттеру) з точністю пакетних симуляторів, але значно вищою швидкістю [1, 2, 6]. Паралельно розвиваються гібридні фреймворки, як-от Grace (2025) та багатоцільові моделі DQNR, де GNN формує ембедінги стану мережі, а DRL-агенти динамічно оптимізують маршрути за критеріями пропускної здатності (приріст до 40%), енергоефективності та стабільності лінків [4, 5, 9]. Новітні розробки також залучають каузальний висновок та архітектури LLM-NAR для підвищення логічної прозорості та масштабованості інтелектуальних систем керування у складних 3D-мережах [7].

Більшість сучасних досліджень фокусується на максимізації пропускної здатності та мінімізації затримок, залишаючи аспект відмовостійкості на другому плані. Проте для критично важливих застосунків здатність алгоритму до самовідновлення є пріоритетною над його обчислювальною ефективністю. З огляду на це, постає необхідність оцінити, наскільки існуючі GNN- та DRL-алгоритми є фактично стійкими до відмов у системі, та якою мірою вони здатні самостійно знаходити обхідні маршрути при зміні топології.

2 Середовище та об'єкт дослідження

Експериментальне середовище побудоване на інтеграції емулятора Mininet та SDN-контролера Ryu, що функціонують у закритому циклі керування трафіком. Основна увага приділена механізмам передачі стану мережі від фізичного рівня до інтелектуального агента.

2.1.1. Software architecture

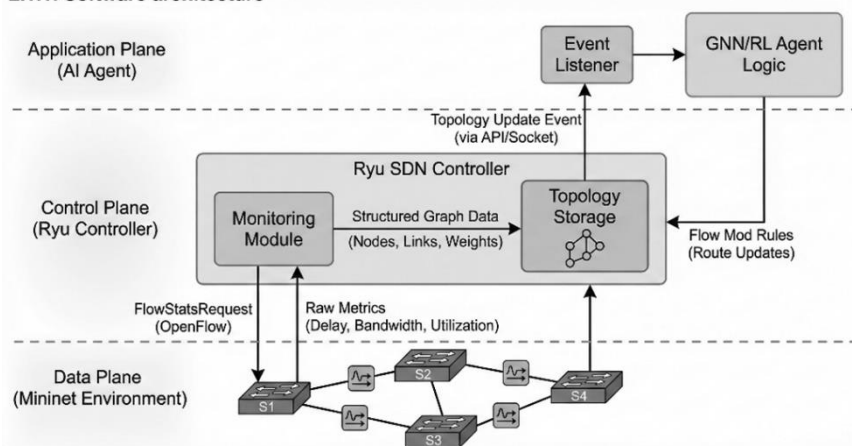


Рис 1. Архітектура експериментального середовища.

Fig. 1. Architecture of the experimental environment.

Контролер Ryu виконує роль центрального вузла збору метрик. Взаємодія реалізована через наступні модулі:

1. Модуль моніторингу: У реальному часі здійснює опитування комутаторів (FlowStatsRequest) для отримання значень затримки, доступної смуги пропускання та рівня використання лінків
2. Сховище топології: Спеціалізований буфер даних, що агрегує сирі метрики з площини даних та трансформує їх у структурований опис графа. Це дозволяє абстрагувати топологію від особливостей протоколу OpenFlow.
3. Event Listener: Програмний інтерфейс у площині застосунків, який реагує на зміни в Topology Storage. Щойно фіксується оновлення (наприклад, зміна завантаження лінка або відмова вузла), агент ініціює перерахунок маршрутів.

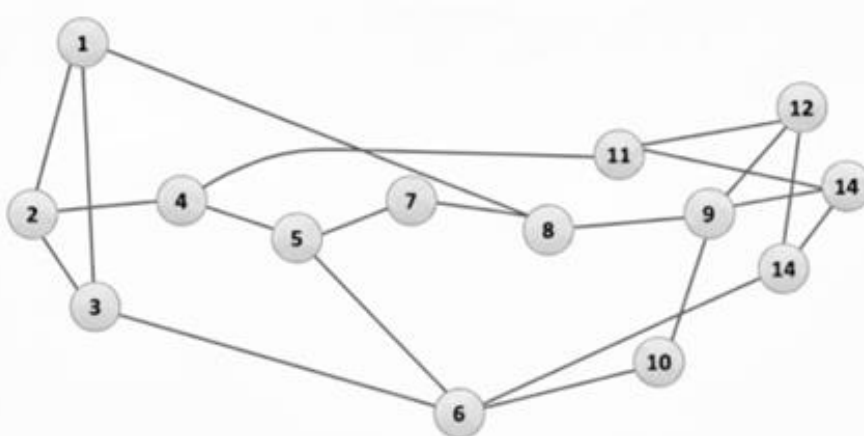


Рис 2. Мережа Національного Наукового Фонду
Fig 2. National Science Foundation Network

Для проведення експериментів було обрано топологію NSFNET (National Science Foundation Network), Цей вибір є класичним для задач оптимізації маршрутизації, оскільки структура мережі містить декілька замкнених контурів, що дозволяє інтелектуальним агентам обирати обхідні шляхи при відмові критичних вузлів. Мережа формалізується як неорієнтований граф $G_t = (V, E, W_t)$. Множина вузлів V представляє OpenFlow комутатори. Кожен вузол має вектор ознак – обсяг буфера та поточне завантаження CPU контролера. Множина ребер E представляє фізичні або віртуальні коанали зв'язку. Матриця ваг W_t представляє динамічні характеристики лінків, такі як затримка, джиттер та доступна смуга пропускання.. На відміну від топології IEEE 39-bus[8], яка використовується для специфічних кіберфізичних систем електромереж, обрана нами топологія NSFNET дозволяє тестувати моделі в умовах магістральної мережі передачі даних загального призначення.

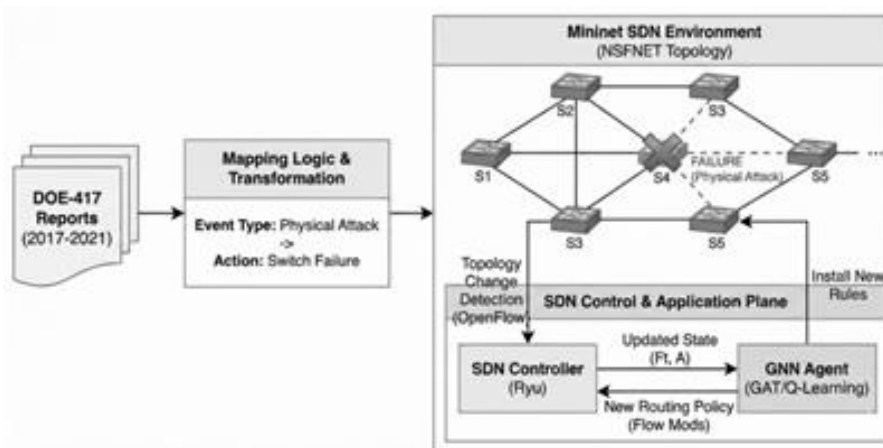


Рис 3. Процес симуляції трафіку на основі реальних звітів про відмови
Fig. 3. Traffic simulation process based on real failure reports

Процес генерації трафіку в середовищі Mininet базується на поєднанні стандартних інструментів вимірювання (iperf) та методів сокетного програмування для відтворення динамічних патернів навантаження. На відміну від традиційного використання синтетичних генераторів трафіку, у даній роботі моделювання навантаження базується на реальних історичних даних про мережеві збої та надзвичайні ситуації. Для відтворення динаміки трафіку використано звіт **Electric Emergency and Disturbance Report (DOE-417)** за період з 2017 по 2021 роки. Ці звіти містять часові мітки початку та завершення інцидентів, типи подій (вандалізм, системні збої, екстремальні погодні умови) та опис зон ураження. За допомогою сокетного програмування в Mininet реалізовано механізм «програвання» цих подій, де кожен тип аварії з репорту трансформується у відповідний сплеск трафіку або відмову вузла в топології NSFNET.

Події типу «системне перевантаження» або «затримка передачі» використовуються для моделювання стохастичного притоку пакетів (Event-Driven traffic). Події типу «фізична атака» або «диверсія» імітують повну відмову комутаторів (switch failure). Кожен сплеск черги на комутаторі, спричинений подією з DOE-417, перетворюється на функцію часу, яка вказує на наближення до критичного порогу заповнення буфера.

Такий підхід дозволяє верифікувати робастність запропонованих моделей (GAT, GCN) у сценаріях, які максимально наближені до реальних умов експлуатації магістральних мереж під час критичних збоїв.

Для навчання нейромережі використовуються три типи вхідних тензорів. Тензор трафіку D_t – матриця вимог «джерело-призначення», де кожен елемент d_{ij} вказує обсяг трафіку який необхідно передати між вузлами i та j . Тензор стану топології S_t включає інформацію про працездатність елементів. Вектор відмов F_t - бінарна маска $f \in \{0,1\}^{|E|+|V|}$, де 1 сигналізує про відмову конкретного вузла або лінка.

Завдання інтелектуальної системи полягає у знаходженні оптимальної стратегії маршрутизації π , яка відображає поточний стан мережі S_t у множину дій a_t (вибір шляхів), мінімізуючи сумарні втрати при дотриманні жорстких обмежень QoS.

Ми розглядаємо задачу багатоцільової оптимізації. Агент (нейромережа) повинен мінімізувати функцію вартості $J(\pi)$ на часовому горизонті T .

$$J(\pi) = E_{\pi}[\sum_{t=0}^T \gamma^t C(S_t, a_t)] \quad (2.1)$$

Де миттєва вартість C визначається як зважена сума критичних метрик:

$$C = \omega_1 \text{Затримка}_{\text{загальна}} + \omega_2 * \text{Втрата пакетів} + \omega_3 * \frac{1}{\text{Надійність}} \quad \gamma \in [0,1] \quad (2.2)$$

Де γ – коефіцієнт дисконтування, а ω_i - вагові коефіцієнти, що визначають пріоритет (наприклад, для VoIP-трафіку ω_1 буде максимальним).

У формулі цільової функції (Cumulative Reward) γ використовується для зважування послідовності нагород:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (2.3)$$

Де R_t - негайна нагорода, $\gamma^k R_{t+k+1}$ - нагорода, яку агент отримує через k кроків, помножена на коефіцієнт у ступені k . При побудові маршруту $P_{s,d}$ нейромережа повинна задовольняти наступні умови:

Обмеження пропускної здатності:

Для кожного ребра $e \in E$

$$\sum_{f \in \phi} x_e^f R_f \leq B_e (1 - f_e) \quad (2.4)$$

де x_e^f - індикатор використання лінка потоком f , R_f - швидкість потоку, B_e – смуга пропускання, f_e - стан відмови.

Умова цілісності потоку: Для кожного вузла $v \in V$:

$$\sum_{j:(v,j) \in E} f_{v,j} - \sum_{i:(i,v) \in E} f_{i,v} = \begin{cases} 1, \text{ якщо } v = \text{source} \\ -1, v = \text{destination} \\ 0, \text{ в іншому випадку} \end{cases} \quad (2.5)$$

Результатом роботи моделі є розподіл ймовірностей над множиною можливих наступних хопів або вибір повного шляху з набору K -найкоротших шляхів, що забезпечує максимальну стійкість до зафіксованого вектора відмов F_t

У межах цього завдання планується оцінити роботу кількох архітектур — GCN, GAT та GNN-DRL — з урахуванням їхніх принципів навчання, структури та способів взаємодії з контролером SDN. Для цього передбачається використання симуляційних середовищ (Mininet) які дозволяють змінювати стан топології в реальному часі, моделювати відмови та вимірювати ключові показники якості обслуговування (QoS), такі як середня затримка, джитер, пропускна здатність і швидкість відновлення маршрутизації.

3 Моделі

Для вирішення задачі регресії на вузлах графа було розглянуто та порівняно дві основні архітектури, що базуються на графових нейронних мережах (GNN): згорткову графову мережу (GCN) та графову мережу уваги (GAT). Усі моделі приймають стандартизований набір вхідних даних: матрицю ознак вузлів $X \in R^{N \times (F+N)}$ де N – кількість вузлів, F – кількість ознак) та зважену матрицю суміжності $A \in R^{N \times N}$. Задачею є прогнозування одного скалярного значення для кожного вузла, тобто $Y' \in R^{N \times 1}$.

3.1. Модель 1: Двошарова GCN (Baseline)

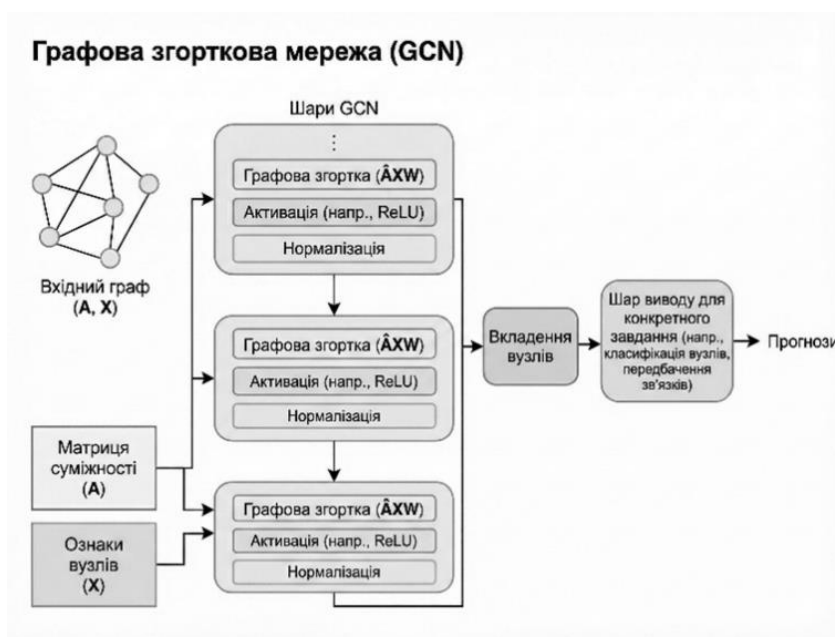


Рис 4. Архітектура GCN для вирішення задачі маршрутизації.

Fig. 4. GCN architecture for solving the routing problem

В якості базової моделі (baseline) використано стандартну двошарову архітектуру GCN. Модель складається з двох послідовних шарів GCNConv, кожен з яких використовує ReLU як функцію активації та генерує EMBEDDING_DIM ознак. Принцип агрегації: GCN виконує ізотропну (статичну) агрегацію. Інформація від сусідніх вузлів усереднюється з вагами, що жорстко визначені нормалізованою матрицею суміжності \hat{A} . Цей механізм є обчислювально ефективним, однак він не розрізняє важливість сусідів у залежності від їхніх ознак.

3.2. Модель 2: Тришарова GAT (Attention-based)

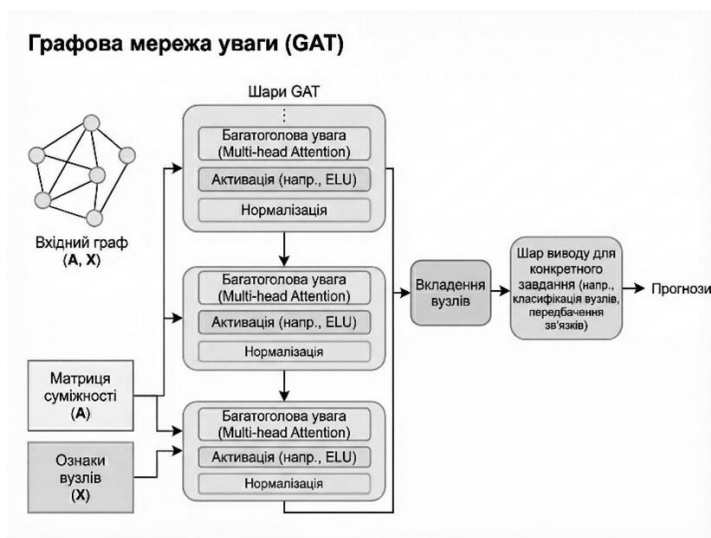


Рис 5. Архітектура GAT для вирішення задачі маршрутизації.
 Fig. 5. GAT architecture for solving the routing problem.

Для врахування динамічної важливості сусідніх вузлів було реалізовано глибшу, тришарову модель на основі GATConv (Graph Attention Network), зокрема її покращеної версії GATv2. Шари 1 та 2: Використовують 4 голови уваги (attn_heads=4). Виходи голів конкатенуються, що призводить до розширення простору ознак до $\mathbb{R}^{N \times EMBEDDING_DIM \times 4}$. Шар 3 (Фінальний): Використовує 8 голів уваги (attn_heads=8). Виходи голів усереднюються (concat_heads=False), повертаючи тензор до розмірності $\mathbb{R}^{N \times EMBEDDING_DIM}$. Принцип агрегації: GAT виконує анізотропну (динамічну) агрегацію. Модель обчислює коефіцієнти уваги для кожної пари з'єднаних вузлів "на льоту", базуючись на їхніх поточних ознаках. Це дозволяє моделі навчитися призначати вищу вагу більш релевантним сусідам та ігнорувати менш важливі, що є ключовою перевагою для складних графів.

4 Тренування

У рамках проведеного дослідження стратегія навчання інтелектуальних агентів реалізована з використанням двох комплементарних підходів: навчання з підкріпленням на основі алгоритму Q-learning для динамічної адаптації у середовищі та навчання з учителем на основі попередньо підготовленого набору даних.

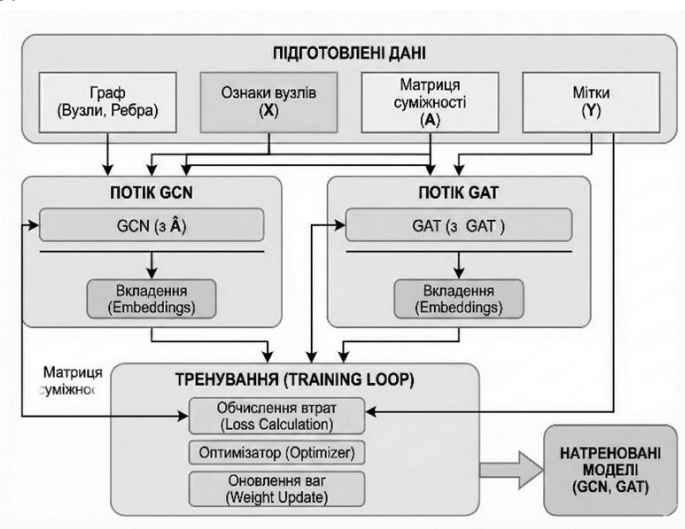


Рис 6. Алгоритм тренування моделей в навчанні з підкріпленням.
 Fig. 6. Algorithm for training models in reinforcement learning.

4.1 Навчання з підкріпленням

Процес навчання ініціюється етапом підготовки даних, де вхідна інформація декомпозується на структурні елементи графа (вузли та ребра) та матричні представлення. Ключовими

компонентами тут виступають матриця ознак вузлів (X), що містить атрибутивну інформацію, та матриця суміжності (A), яка формалізує топологію зв'язків мережі. Для контрольованого навчання також використовуються цільові мітки (Y), що слугують еталоном (ground truth) для оцінки точності прогнозів.

4.2 Q-learning

Процес прийняття рішень агентом у графі реалізовано на основі алгоритму навчання з підкріпленням (Deep Q-Learning), де навігація моделюється як послідовність переходів між вузлами мережі. На кожному часовому кроці t поточний стан агента S_t характеризується набором локальних ознак X_t , які попередньо обробляються модулем графових нейронних мереж (GCN або GAT). Цей етап дозволяє трансформувати сирі дані у компактні векторні представлення (embeddings), що кодуєть структурну інформацію про оточення вузла.

Навчання системи відбувається через механізм зворотного зв'язку: після виконання дії агент отримує сигнал винагороди R_{t+1} , який кількісно визначає успішність переходу. Ця величина разом із максимальною оцінкою нового стану використовується блоком «Q-Learning Оновлення» для обчислення помилки передбачення та коригування вагових коефіцієнтів MLP. Такий ітеративний підхід дозволяє агенту поступово оптимізувати стратегію навігації, навчаючись обирати маршрути, що максимізують довгострокову винагороду.

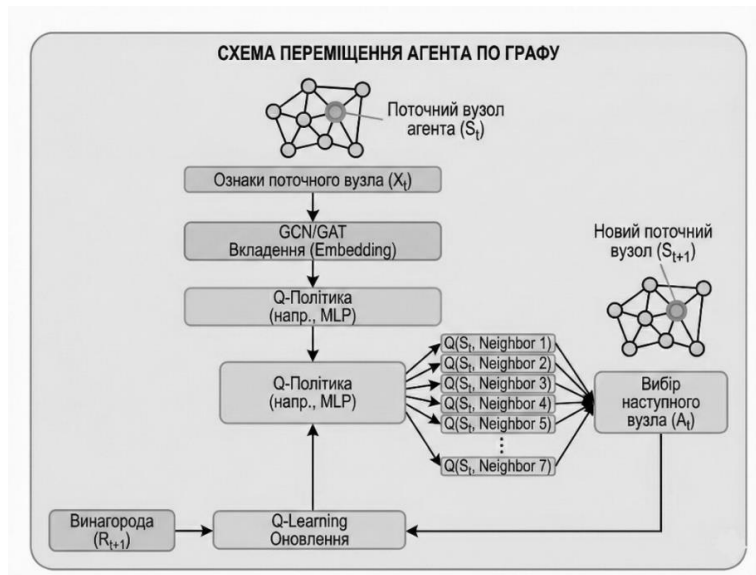


Рис 7. Алгоритм тренування моделі при Q-Learning підході.

Fig. 7. Model training algorithm using the Q-Learning approach.

5 Результати

Аналіз продуктивності алгоритмів залежно від навантаження мережі

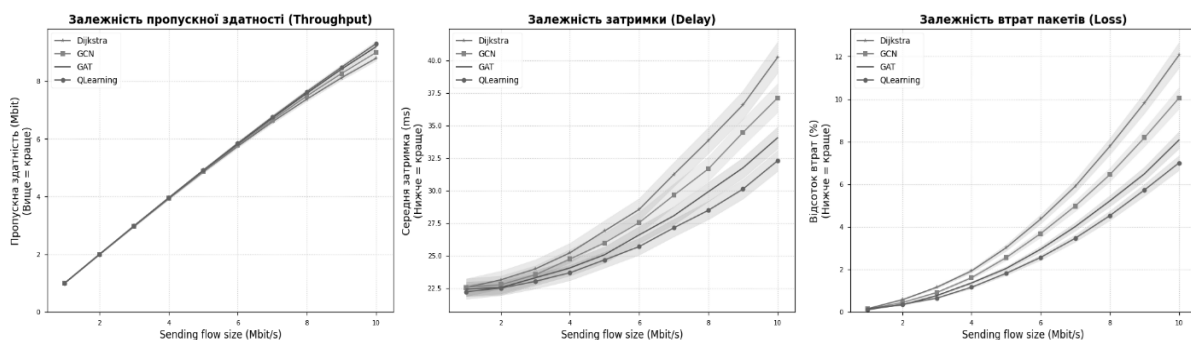


Рис 8. Порівняння ефективності моделей на тренуваній топології.

Fig. 8. Comparison of the effectiveness of models on the trained topology.

Результати експерименту демонструють порівняльну ефективність методів маршрутизації (Дейкстра, GCN, GAT, Q-Learning) при зростанні навантаження мережі від 1 до 10 Мбіт/с. Хоча пропускна здатність (Throughput) залишається зівставною для всіх підходів, показники якості

обслуговування суттєво різняться при збільшенні інтенсивності потоку. Статичний алгоритм Дейкстри виявляє найшвидшу деградацію параметрів, досягаючи критичних значень затримки та втрат пакетів (понад 12%) у пікових режимах.

Серед інтелектуальних методів найкращу масштабованість та стійкість до перевантажень продемонстрував агент Q-Learning, який забезпечив мінімальну затримку та найнижчий відсоток втрат (близько 7%). Архітектури на базі GNN також перевершили класичний підхід, причому механізм GAT виявився ефективнішим за GCN. Отримані дані підтверджують перевагу динамічної адаптації маршрутів на основі навчання з підкріпленням над статичними та суто прогнозними методами.

Аналіз продуктивності (Сценарій відмови свіча та адаптації)

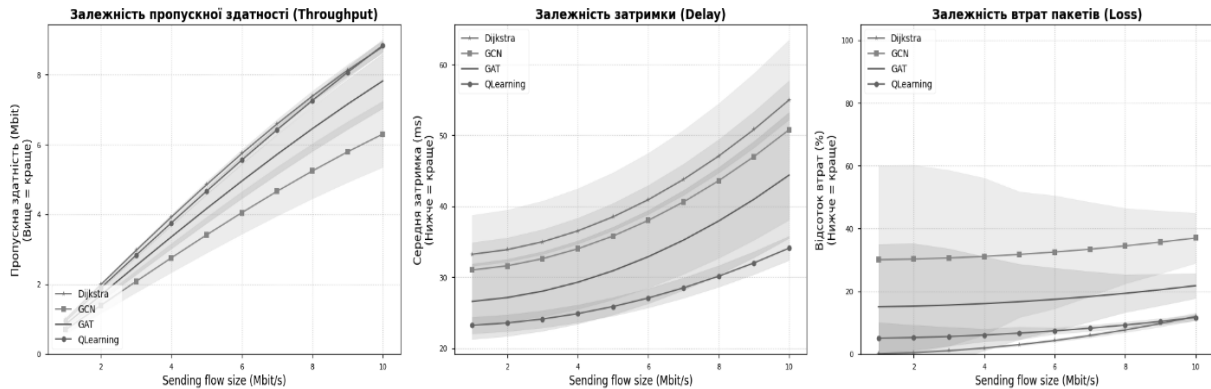


Рис 9. Порівняння ефективності моделей на тренованій топології при відмові вузлів.
 Fig. 9. Comparison of the effectiveness of models on the trained topology in the event of node failure.

Експеримент із моделюванням відмови комутатора виявив критичні відмінності в адаптивності досліджуваних методів. На відміну від штатного режиму роботи, де показники були близькими, зміна топології спричинила значну деградацію продуктивності моделей на основі GNN. Зокрема, модель GCN продемонструвала найгірші результати: суттєве зниження пропускної здатності та стабільно високий рівень втрат пакетів (близько 30%) незалежно від навантаження, що свідчить про низьку здатність до генералізації при зміні структури графа. Архітектура GAT показала кращу адаптивність (втрати близько 15%), проте також не змогла ефективно перебудувати маршрути.

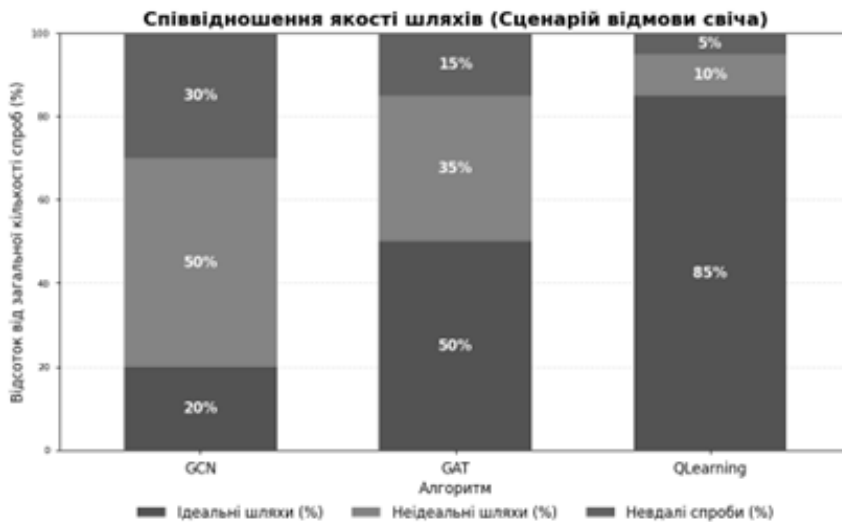


Рис 10. Співвідношення якості шляхів у сценарії відмови вузла.
 Fig. 10. Path quality correlation in node failure scenario.

Найвищу відмовостійкість продемонстрував агент Q-Learning, який не лише зберіг пропускну здатність на рівні класичного алгоритму Дейкстри, але й забезпечив найкращі показники QoS. У той час як алгоритм Дейкстри через перевантаження резервних каналів показав найбільшу затримку, Q-Learning успішно знаходив оптимальні обхідні шляхи, утримуючи мінімальні

значення затримки та втрат пакетів навіть при максимальному навантаженні. Це підтверджує перевагу динамічної стратегії RL над статичними методами та класичними GNN у критичних сценаріях.

На діаграмі наведено порівняльний аналіз ефективності маршрутизації для трьох інтелектуальних агентів (GCN, GAT, Q-Learning) в умовах аварійної зміни топології мережі. Результати класифіковано за трьома категоріями: ідеальні маршрути, субоптимальні (неідеальні) шляхи та невдалі спроби встановлення з'єднання.

Отримані дані свідчать про обмежену здатність моделі GCN до адаптації: лише 20% побудованих маршрутів були оптимальними, тоді як частка критичних збоїв досягла 30%. Архітектура GAT продемонструвала кращу стійкість, підвищивши відсоток ідеальних шляхів до 50% та знизивши рівень відмов удвічі (до 15%). Найвищу ефективність виявив алгоритм Q-Learning, який у 85% випадків забезпечив побудову ідеального маршруту, мінімізувавши частку невдалих спроб до 5%. Це підтверджує перевагу динамічного навчання з підкріпленням над підходами supervised learning при роботі з непередбачуваними змінами структури мережі.

6 Висновки

У роботі проведено комплексне дослідження ефективності методів інтелектуальної маршрутизації у програмно-конфігурованих мережах (SDN), порівнюючи класичний алгоритм Дейкстри, моделі на основі графових нейронних мереж (GCN, GAT) та підхід навчання з підкріпленням (Q-Learning). Результати моделювання підтвердили, що інтеграція машинного навчання дозволяє суттєво покращити показники якості обслуговування (QoS) порівняно зі статичними алгоритмами, особливо в умовах високого навантаження мережі.

Експериментальні дані засвідчили, що у стабільному режимі роботи архітектури GAT та Q-Learning демонструють найкращу масштабованість, забезпечуючи низьку затримку та мінімізацію втрат пакетів при зростанні трафіку до 10 Мбіт/с. При цьому механізм уваги в GAT виявився ефективнішим за спектральну згортку GCN, дозволяючи точніше враховувати вагу зв'язків між вузлами. Однак найбільш показовими стали результати тестування в аварійних сценаріях зі зміною топології мережі (відмова комутатора).

Ключовим результатом дослідження є доведення переваги динамічного підходу Q-Learning над методами навчання з учителем у нестабільних середовищах. У той час як моделі GNN зіткнулися з проблемою індуктивного узагальнення, продемонструвавши високий рівень невдалих маршрутів (до 30% для GCN), агент Q-Learning успішно адаптувався до нових умов. Завдяки механізму безперервної взаємодії з середовищем, він забезпечив побудову ідеальних шляхів у 85% випадків, утримуючи параметри продуктивності на рівні штатного режиму. Перспективи подальших досліджень спрямовані на розробку гібридних архітектур, які дозволять об'єднати можливості GNN щодо аналізу просторових ознак топології з адаптивною гнучкістю методів навчання з підкріпленням. Окремим вектором розвитку є оптимізація швидкості збіжності алгоритмів та їх масштабування для великих магістральних мереж зі складними ієрархічними зв'язками.

СПИСОК ЛІТЕРАТУРИ

1. Rusek K., et al. RouteNet: Leveraging Graph Neural Networks for network modeling and optimization. *IEEE Journal on Selected Areas in Communications*. 2020. Vol. 38, no. 10. P. 2260–2270. DOI: 10.1109/JSAC.2020.3000405.
2. Ferriol-Galmés M., et al. RouteNet-Erlang: A Graph Neural Network for Network Performance Evaluation. *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. 2022. P. 2018–2027. DOI: 10.1109/INFOCOM48030.2022.9796677.
3. Mammeri A. A Survey on Machine Learning Techniques for Routing Optimization in SDN. *IEEE Access*. 2021. Vol. 9. P. 104523–104544. DOI: 10.1109/ACCESS.2021.3098763.
4. Stampa G., et al. A Deep Reinforcement Learning Approach for Software-Defined Networking Routing Optimization. *IEEE International Conference on Communications (ICC)*. 2019. P. 1–6. DOI: 10.1109/ICC.2019.8761479.
5. Zheng S., Huang H., GNN et al. Research on Generalized Intelligent Routing Technology Based on Deep Reinforcement Learning. *Electronics*. 2022. Vol. 11, no. 3. P. 343. DOI: 10.3390/electronics11030343.

6. Ferriol-Galmés M., Barlet-Ros P., Cabellos-Aparicio A. RouteNet-Fermi: Network Modeling with Graph Neural Networks. *IEEE/ACM Transactions on Networking*. 2024. Vol. 32, no. 3. P. 1200–1215. DOI: 10.1109/TNET.2024.3392336.
7. Wu W., et al. Neural Algorithmic Reasoners informed Large Language Model for Multi-Agent Path Finding (LLM-NAR). arXiv preprint arXiv:2508.17971. 2025. URL: <https://arxiv.org/abs/2508.17971> (дата звернення: 28.01.2026).
8. Islam M., et al. Results analysis on the cyber layer of the IEEE 39-bus test system under both normal and failure condition. *IEEE Access*. 2024. Vol. 12. P. 106234–106245. DOI: 10.1109/ACCESS.2024.10623446.
9. Grace: Toward Routing in Dynamic Network Environments With Graph Embedding. *IEEE Transactions on Networking*. 2025. DOI: 10.1109/TNET.2025.11078667.

Turchak*Postgraduate student of the Department of Applied Mathematics***Denys***V. N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv, 61022***Serhiyovich***e-mail: denys.turchak@karazin.ua**<https://orcid.org/0009-0003-9144-8280>***Rukkas***Doctor of Technical Sciences, Associate Professor***Kyrylo Markovych***V. N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv, 61022**e-mail: rukkas@karazin.ua;**<https://orcid.org/0000-0002-7614-0793>*

The impact of GNN architecture on the robustness of edge routes in scenarios of single node types

The paper considers the problem of intelligent routing (path finding) in software-defined networks (SDN) using graph neural networks in order to increase the efficiency of network resource use and adapt to dynamic changes in the network state (for example, channel congestion or delays).

Topicality. Modern software-defined networks (SDN) are faced with increasing traffic volumes and increased quality of service (QoS) requirements. Traditional routing algorithms (e.g. Dijkstra) are static and inefficient in conditions of high load dynamics or sudden topology changes (hardware failures). This leads to channel congestion, increased latency, and packet loss. The use of machine learning methods, in particular graph neural networks (GNN) and reinforcement learning (RL), opens up new opportunities for creating adaptive intelligent agents capable of optimizing routing in real time, which makes this research timely and important for the development of telecommunication systems.

Goal. The aim of the research is to improve the efficiency and reliability of data transmission in SDN networks by developing and comparative analysis of intelligent routing methods. The main focus is on the study of graph neural network architectures (GCN, GAT) and the Q-Learning algorithm to provide adaptive traffic management under conditions of variable load and network node failures.

Research methods. The methodological basis of the work is based on the integrated application of graph theory to formalize the network topology, deep learning methods to process node features, and reinforcement learning algorithms to make routing decisions. Experimental verification of the proposed approaches was carried out by emulating a software-configured network in the Mininet environment under the control of the Ryu controller. The software implementation included the development of models based on convolutional networks (GCN) and attention networks (GAT) using deep learning libraries, as well as the implementation of the Deep Q-Learning agent. The effectiveness of the algorithms was assessed by comparative analysis of key quality of service metrics - throughput, average latency, and packet loss percentage - in scenarios of gradual load growth and emergency topology change due to equipment failure.

The results. The study found that the integration of machine learning methods allows for significant improvements in data transmission parameters compared to the classic Dijkstra algorithm, especially in high traffic conditions, where intelligent agents provide lower latency and connection stability. Critical analysis of fault tolerance revealed significant differences between the studied architectures: the GCN model demonstrated limited adaptability with a routing failure rate of 30%, while the GAT architecture showed better flexibility, generating optimal paths in half of the cases. The highest efficiency was confirmed by the Q-Learning method, which, thanks to dynamic interaction with the environment, ensured the construction of ideal routes in 85% of experiments and minimized packet loss to 5–7% even in critical situations, which proves the superiority of Reinforcement Learning approaches over Supervised Learning methods in adaptive network control tasks.

Key words: Software-defined networks (SDN), Graph neural networks (GNN), Reinforcement learning, Intelligent routing, GCN, GAT, Q-Learning, Quality of service (QoS), Fault tolerance, Adaptive traffic management.