УДК 004.942:621.396
DOI https://doi.org/10.36994/2788-5518-2025-02-10-14

МОДЕЛЮВАННЯ ГЛИБИННИХ ПІДКРІПЛЮВАЛЬНИХ МЕТОДІВ ДЛЯ АДАПТИВНОЇ МАРШРУТИЗАЦІЇ В КОМП'ЮТЕРНІЙ SDN МЕРЕЖІ З ВИКОРИСТАННЯМ PYTHON

***Полуектова Н. Р.**, д.е.н., доц., Запорізький інститут економіки та інформаційних технологій, Запоріжжя, Україна. https://orcid.org/0009-0005-9637-4916, n.poluektova@econom.zp.ua*

***Литвиненко Т. М.**, к.т.н., Запорізький інститут економіки та інформаційних технологій, Запоріжжя, Україна. https://orcid.org/0009-0002-4868-0204, t.litvinenko@econom.zp.ua*

У роботі досліджується підхід до адаптивної маршрутизації в комп'ютерних мережах, побудованих за парадигмою Software-Defined Networking (SDN), із використанням методів глибокого підкріплювального навчання (Deep Reinforcement Learning, DRL) та спеціалізованих бібліотек Python для моделювання та оптимізації мережевих процесів. Основна увага приділяється алгоритму Deep Q-Network (DQN), який поєднує можливості глибоких нейронних мереж із традиційним Q-learning для вибору оптимальних маршрутів у динамічних мережевих середовищах. Для експериментальної перевірки запропонованого підходу реалізовано симульовану SDN-топологію в середовищі Mininet, що включає сервери, клієнти та централізований контролер, який приймає рішення на основі аналізу поточних мережевих метрик, таких як затримка, пропускна здатність та завантаження каналів. Модель агента навчається мінімізувати затримку пакетів і максимізувати пропускну здатність, використовуючи багатокритерійну функцію винагороди, що інтегрує обидва параметри для забезпечення оптимальної продуктивності мережі. Результати моделювання показали, що DQN-агент здатний відтворювати поведінку класичного алгоритму Dijkstra у статичних умовах і водночас перевершує його в умовах змінного трафіку, демонструючи високу адаптивність до коливань мережевого навантаження та несподіваних збоїв. Запропонований підхід підтвердив ефективність DRL для задач інтелектуальної маршрутизації в SDN-мережах, відкриваючи перспективи для застосування в масштабних системах із високими вимогами до QoS і енергоефективності. Подальші дослідження передбачають порівняння з іншими DRL-алгоритмами, такими як Double DQN, PPO, аналіз масштабованості рішення для великих мереж та інтеграцію з IoT-середовищами для забезпечення надійної та гнучкої маршрутизації.

***Ключові слова:** адаптивна маршрутизація, глибоке навчання з підкріпленням, моделювання мережі, програмно-конфігуровані мережі, середовище Mininet.*

MODELING OF DEEP REINFORCEMENT LEARNING METHODS FOR ADAPTIVE ROUTING IN A COMPUTER SDN NETWORK USING PYTHON

***Nataliya Poluektova**, Ph.D., Ass. Prof., Zaporizhzhya Institute of Economics and Information Technologies, Zaporizhzhia, Ukraine. https://orcid.org/0009-0005-9637-4916, n.poluektova@econom.zp.ua*

***Taras Lytvynenko,** Ph.D., Zaporizhzhya Institute of Economics and Information Technologies, Zaporizhzhia, Ukraine. https://orcid.org/0009-0002-4868-0204, t.litvinenko@econom.zp.ua*

The study investigates an approach to adaptive routing in computer networks built on the Software-Defined Networking (SDN) paradigm, using Deep Reinforcement Learning (DRL) methods and specialized Python libraries for modeling and optimizing network processes. The main focus is on the Deep Q-Network (DQN) algorithm, which combines the capabilities of deep neural networks with traditional Q-learning to select optimal routes in dynamic network environments. For experimental validation of the proposed approach, a simulated SDN topology was implemented in the Mininet environment, including servers, clients, and a centralized controller that makes decisions based on the analysis of current network metrics, such as latency, bandwidth, and link utilization. The agent model is trained to minimize packet delay and maximize throughput, using a multi-criteria reward function that integrates both parameters to ensure optimal network performance. Simulation results showed that the DQN agent can replicate the behavior of the classical Dijkstra algorithm in static conditions while outperforming it under dynamic traffic scenarios, demonstrating high adaptability to fluctuations in network load and unexpected failures. The proposed approach confirmed the effectiveness of DRL for intelligent routing tasks in SDN networks, opening prospects for deployment in large-scale systems with stringent QoS requirements and energy efficiency considerations. Further research involves comparisons with other DRL algorithms such

as Double DQN and PPO, analysis of solution scalability for large networks, and integration with IoT environments to ensure reliable and flexible routing.

Key words: *adaptive routing, deep reinforcement learning, network modeling, software-defined networks, Mininet environment.*

Вступ

У сучасних комп'ютерних мережах, які характеризуються динамічною топологією, високою інтенсивністю трафіку та різноманіттям типів сервісів, ефективна маршрутизація даних є одним із ключових чинників, що визначає продуктивність і надійність мережевої інфраструктури.

Software-Defined Networking (SDN) – це підхід до побудови комп'ютерних мереж, який відокремлює контрольний шар (control plane) від передавального шару (data plane). Завдяки цьому мережа стає програмно керованою, що дає більшу гнучкість і спрощує автоматизацію управління потоками даних. В SDN контрольний та передавальний шари мережі розділені, що дозволяє централізовано управляти маршрутизацією через контролер.

Традиційні алгоритми маршрутизації, такі як OSPF (Open Shortest Path First), RIP (Routing Information Protocol) або BGP (Border Gateway Protocol), базуються на статичних або напівдинамічних правилах, які не завжди швидко реагують на зміни у стані мережі. Це призводить до таких проблем, як затримки у доставці пакетів, перевантаження окремих вузлів та неефективне використання мережевих ресурсів.

З розвитком технологій Інтернету речей (IoT), 5G-комунікацій та розподілених обчислень зростає потреба в інтелектуальних, адаптивних підходах до маршрутизації, які допоможуть обирати оптимальні маршрути передачі даних залежно від поточного стану мережі.

Одним із перспективних напрямів вирішення цієї проблеми є використання підкріплювального навчання з глибокими нейронними мережами (Deep Reinforcement Learning, DRL). DRL поєднує потужність глибокого навчання для аналізу складних вхідних даних із можливістю самостійного прийняття рішень, характерною для підкріплювального навчання.

Q-learning – це метод підкріплювального навчання (Reinforcement Learning), який дозволяє агенту навчитися оптимальній стратегії у середовищі без потреби знати його модель наперед. Мета агента – максимізувати суму очікуваних нагород у майбутньому.

Алгоритм ґрунтується на Q(s,a)-функції (яку також називають функцією якості), що показує очікувану суму нагород, якщо агент перебуває в стані s і виконує дію a, а потім слідує цій обраній оптимальній стратегії.

Основні компоненти:

- стан (State, S) – поточна ситуація в середовищі;
- дія (Action, A) – вибір агента, на якого впливає середовище;
- нагорода (Reward, R) – оцінка дії агента, яка сигналізує, наскільки вона корисна;
- Q(s,a)-функція– прогнозує очікувану нагороду від дії a у стані s;
- параметри алгоритму:
  - α – коефіцієнт навчання (learning rate), визначає швидкість оновлення Q-значень;
  - γ– коефіцієнт дисконтування (discount factor), враховує важливість майбутніх нагород;
  - ϵ – параметр для ϵ -жадібної політики вибору дій.

Основна ідея алгоритму полягає в тому, що агент оновлює значення Q для кожної пари стан-дія за формулою (1) [1].

$$Q(s,a) = Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right], \quad (1)$$

де $\max Q(s',a')$ – максимальна очікувана винагорода для нового стану.

Це оновлення дозволяє агенту поступово наближуватись до оптимальної стратегії, навіть якщо все середовище невідоме.

Розширення класичної ідеї полягає у застосуванні глибоких нейронних мереж для роботи з великими або безперервними просторами станів. Одним з алгоритмів, який реалізує принципи навчання з підкріпленням за допомогою глибоких нейронних мереж є алгоритм DQN (Deep Q-Network).

В класичних алгоритмах Q-Learning значення Q зберігаються у табличних структурах. У DQN замість таблиць використовується глибинна нейронна мережа, яка наближує Q-функцію через перерахунок ваг нейронів:

$$Q(s,a,\theta) \approx Q^*(s,a), \quad (2)$$

де θ – ваги нейронів.

Основні компоненти DQN:

1) нейронна мережа (Q-network) яка приймає стан s і повертає Q значення для всіх можливих дій;
2) буфер (Replay Memory) де зберігаються минулі переходи (s, a, r, s');
3) копія нейронної мережі, яка оновлюється не так часто, що забезпечує стабільність навчання;
4) параметри навчання (α, γ, ϵ).

Агент спостерігає поточний стан s, використовуючі жадібну політику обирає дію a, отримує нагороду r та переходить у стан s'. Цей досвід запом'ятовується у буфері. Далі з нього береться випадковий варіант досвіду та оновлюються параметри нейронной мережі, так, що мінімізується функція витрат.

Основна перевага DRL полягає у здатності агента навчатися на основі досвіду взаємодії з мережею. Це дозволяє системі маршрутизації адаптуватися до змін трафіку, втрат пакетів, затримок та інших факторів у режимі реального часу без необхідності ручного налаштування. На відміну від класичних алгоритмів, які використовують заздалегідь визначені метрики (наприклад, довжину маршруту або пропускну здатність), DRL здатний враховувати комплексні залежності між параметрами мережі та їхню динаміку.

DRL-агенти можуть приймати рішення на основі глобального стану мережі, враховуючи різноманітні QoS-параметри, такі як затримка, пропускна здатність, втрата пакетів. Зокрема, в статті "QoS Routing Optimization Based on Deep Reinforcement Learning in SDN" пропонується архітектура, де агент DRL інтегрується в контролер SDN, а маршрутизація здійснюється на рівні передавання даних, що дозволяє ефективно обробляти вимоги QoS джерела [2]. У роботі пропонується архітектура оптимізації маршрутизації яка передбачає розділення функцій керування та передавання в SDN, централізуючи контроль за пересиланням даних і інтегруючи глибоке навчання з підкріпленням для прийняття обґрунтованих рішень щодо маршрутизації. Враховуючи такі параметри, як затримка**,** пропускна здатність**,** джитер (коливання затримки) та рівень втрати пакетів**,** розробляється функція винагороди, яка спрямовує алгоритм Deep Deterministic Policy Gradient (DDPG) до навчання оптимальної стратегії маршрутизації для забезпечення високої якості обслуговування.

G. Stampa та ін [3] також розробили агента DRL для оптимізації маршрутизації в SDN, який адаптується до поточних умов трафіку та мінімізує затримки в мережі. Експериментальні результати демонструють значне покращення продуктивності порівняно з традиційними алгоритмами. Результати моделювання продемонстрували, що запропонований алгоритм суттєво знижує мережеву затримку та підвищує загальну ефективність передавання даних, перевершуючи традиційні методи.

У роботі [4] пропонується нова схема маршрутизації в SDN, яка поєднує глибоке підкріплювальне навчання (DRL), детекцію причинно-наслідкових зв'язків та графові нейронні мережі (GNN). Це дозволяє більш ефективно враховувати взаємодії між агентами та середовищем, покращуючи якість обслуговування (QoS) у мережі.

У статті [5] представлено алгоритм маршрутизації з множинними шляхами для SDN, який використовує DRL, механізм довіри та вдосконалений підхід до пошуку в глибину (DFS). Алгоритм покращує безпеку та ефективність маршрутизації, знижуючи затримки та варіації затримок.

Таким чином, чисельні сучасні дослідження свідчать, що у сучасних комп'ютерних мережах, зокрема в програмно-орієнтованих мережах (SDN), адаптивна маршрутизація є критично важливою для забезпечення високої якості обслуговування (QoS). Традиційні методи маршрутизації, такі як статичні алгоритми або ECMP (Equal-Cost Multi-Path), часто не здатні ефективно реагувати на динамічні зміни в мережі, такі як коливання трафіку, затримки або відмови ліній. Тому все більше уваги приділяється використанню глибокого підкріплювального навчання (DRL) для адаптивної маршрутизації.

Метою даного дослідження є розроблення та тестування агента глибокого підкріплювального навчання (DRL-агента) для задачі оптимального вибору маршруту у SDN комп'ютерній мережі за допомогою DQN алгоритму. У межах дослідження передбачалося вивчити можливості Python-бібліотек для моделювання глибинних підкріплювальних методів пошуку оптимальних маршрутів в комп'ютерній мережі. Для цього реалізовано наступні задачі:

- розроблено модель середовища, що відображає динамічну структуру мережі;
- DRL-агента навчено оптимізувати маршрути на основі метрик затримки;
- виконано порівняння результатів роботи DRL-агента з традиційними алгоритмами маршрутизації для оцінки ефективності запропонованого підходу.

Методи та моделі дослідження

У ході дослідження було реалізовано інтелектуальну систему маршрутизації на основі глибокого навчання з підкріпленням (DRL), яка використовує модифікований підхід до обчислення винагороди та специфічну топологію мережі.

Була створена симульована топологія мережі, що моделює базову архітектуру SDN у середовищі Mininet. Топологія включає три сервери, шість клієнтів та один центральний комутатор. Така конфігурація дозволяє моделювати типову ситуацію, де:

- сервери мають кращі канали зв'язку (наприклад, дата-центри);
- клієнти мають обмежену пропускну здатність і більшу затримку (наприклад, домашні користувачі).

Це створює основу для тестування алгоритмів маршрутизації, балансування навантаження, кешування контенту та навчання агентів з підкріпленням у динамічному середовищі.

На кожному етапі агент отримував інформацію про стан мережі пропускну здатність каналів, завантаження лінків і середній час відповіді – та приймав рішення щодо вибору наступного вузла маршруту. Для кожної пари вузлів вимірювалися такі параметри:

− затримка (delay) **–** через команду ping;
− пропускна здатність (bandwidth) **–** через інструмент iperf.

Основний алгоритм навчання базувався на DQN (Deep Q-Network), однак функція винагороди враховувала:

− негативну винагороду за затримку **–** чим більша затримка, тим гірше,
− позитивну винагороду за пропускну здатність **–** чим ширший канал, тим краще.

Агент реалізовано за допомогою Python-бібліотеки TensorFlow як Deep Q-Network з двома прихованими шарами по 64 нейрони з активацією ReLU. Вхідний шар формувався на основі поточних метрик затримки та пропускної здатності, а вихідний шар містив кількість нейронів, що відповідала кількості можливих дій (вибору наступного вузла). Модель оптимізувалась за допомогою алгоритму Adam з функцією втрат MeanSquaredError. Навчання відбувалось протягом 300 епізодів, у кожному з яких агент мав до 10 кроків для досягнення цільового вузла. Нагорода обчислювалась як комбінація затримки та пропускної здатності: reward = –delay + 0.01 × bandwidth. Були враховані наступні механізми стабілізації навчання:

− Replay Buffer (буфер досвіду): дозволяє навчатись на випадкових підвибірках минулих епізодів, що знижує кореляцію даних;
− Target Network: копія основної мережі, яка оновлюється періодично (кожні 10 епізодів) для стабільнішого розрахунку цільових значень Q;
− Epsilon-greedy стратегія: на початку агент часто вибирає випадкові дії (високе ε), але з кожним епізодом поступово зменшує ε, переходячи до експлуатації отриманих знань.

Після завершення навчання модель зберігалась у файл. У фазі тестування мережа запускалась з тією ж топологією, збирались поточні метрики, і навчений агент використовувався для побудови маршруту від джерела до цілі. Для кожного маршруту обчислювались загальна затримка та пропускна здатність, які порівнювались з результатами алгоритму Dijkstra, що обирає маршрут з мінімальною сумарною затримкою, але не враховує пропускну здатність.

Під час навчання будувався графік винагород, який демонстрував прогрес агента та стабілізацію його політики (рис. 1)

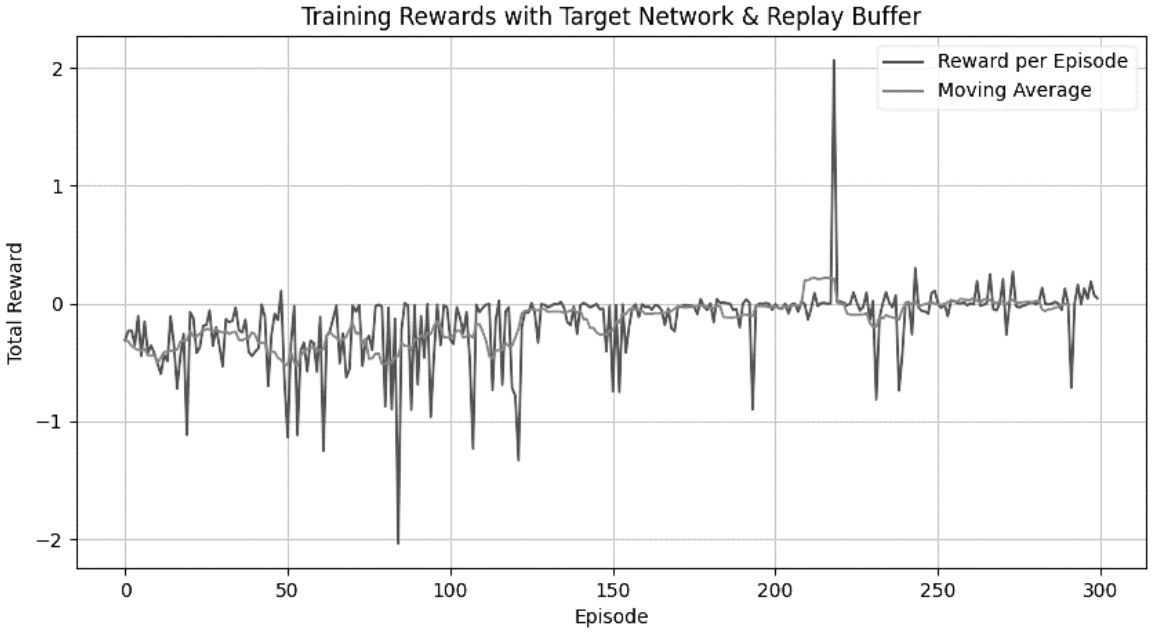

Рис.1. Стабілізація графіку винагород в процесі навчання

Результати дослідження

Під час першого експерименту з тестування було виявлено, що маршрути, обрані DQN-агентом, практично повністю збігаються з маршрутами, визначеними алгоритмом Dijkstra. Графічні результати, що відображали затримку і пропускну здатність, також збігалися (рис.3). Це свідчить про те, що модель навчилася правильно інтерпретувати метрики мережі та відтворила логіку класичного алгоритму найкоротшого шляху. Такий результат пояснюється тим, що експерименти проводилися у статичному середовищі з однорідними характеристиками каналів, де оптимальний маршрут був єдиним і незмінним.

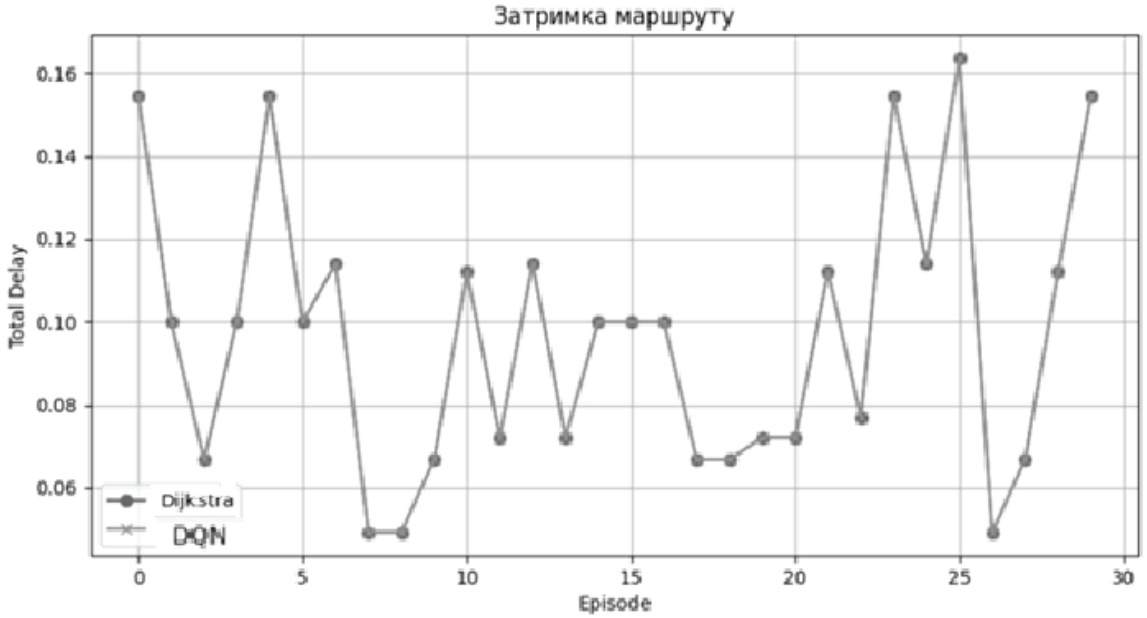

Рис. 2. Результат експерименту у статичному середовищі

Щоб імітувати реальні зміни в мережі (флуктуації затримок, перевантаження каналів), було додано варіацію параметрів delay і bandwidth випадковим чином в процес донавчання моделі.

На етапі тестування знову були обчислені та візуалізовані значення загальної затримки маршрутів, які отримані для DQN на основі максимального Q-значення, яке прогнозує навчена модель, а, для Dijkstra – за мінімальною сумарною затримкою у поточній матриці delay.

Результати наведні на рис. 3.

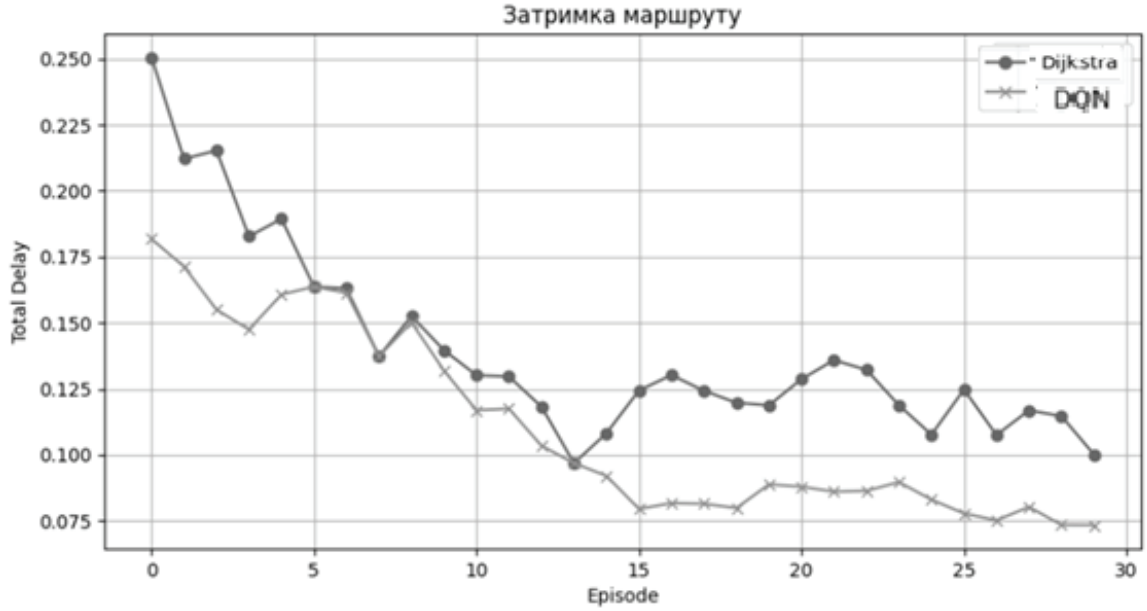

Рис. 3. Результат експерименту у динамічному середовищі

Результати експериментів показали, що навчена DQN-модель здатна адаптивно оптимізувати маршрутизацію трафіку у мережі з непостійними параметрами. DQN демонструє кращі результати завдяки своїй адаптивності до динамічних умов мережі. Він навчається розпізнавати шаблони змін у метриках, таких як затримка та пропускна здатність, і приймати рішення, що забезпечують оптимальний компроміс між ними навіть у випадку варіацій трафіку. На відміну від нього, алгоритм Dijkstra кожного разу виконує перерахунок найкоротшого шляху за статичною метрикою, що робить його менш гнучким у змінних умовах. Крім того, DQN ураховує кілька факторів одночасно, оптимізуючи

багатофакторну функцію винагороди, яка балансує між швидкістю та якістю маршруту, тоді як Dijkstra мінімізує лише один показник – затримку.

Завдяки навчанню через досвід DQN поступово накопичує інформацію про вигідні та невигідні маршрути, формуючи політику, яка забезпечує найкращий середній результат, навіть якщо деякі лінки тимчасово деградують. У складних топологіях мережі DQN також здатен приймати гнучкі рішення, обираючи неочевидні маршрути, які можуть бути коротшими за сумарним часом передачі, хоча й не обов'язково мінімальними за кількістю переходів.

Висновки

Отже, у процесі дослідження було підтверджено, що DQN-підхід здатний відтворити поведінку традиційного алгоритму маршрутизації у детермінованому середовищі. Разом з тим результати показали обмеження такої моделі в умовах статичної топології. Для виявлення переваг глибокого підкріплювального навчання над класичними методами доцільно надалі проводити експерименти у динамічних мережевих сценаріях, де параметри затримки, пропускної здатності та втрат пакетів змінюються з часом, що створює ситуації, у яких агент повинен адаптувати свою політику маршрутизації в реальному часі.

Попри продемонстровану ефективність DQN у задачах маршрутизації в динамічних мережах, існує низка перспективних напрямів для подальшого дослідження, які дозволять глибше оцінити доцільність та межі застосування методів глибокого навчання з підкріпленням у мережевих системах.

1. Порівняння з іншими алгоритмами навчання з підкріпленням. DQN є лише одним із представників класу алгоритмів DRL. Подальші дослідження можуть включати порівняння з такими методами, як Double DQN, Dueling DQN, A3C або PPO, з метою виявлення більш стабільних або швидко адаптивних стратегій у складних мережевих середовищах.

2. Вивчення масштабованості. Необхідно дослідити, як змінюється ефективність DQN при збільшенні кількості вузлів, складності топології та кількості одночасних потоків. Це дозволить оцінити придатність підходу для реальних мережевих інфраструктур.

3. Адаптація до QoS-вимог. Варто дослідити, як агент може враховувати специфічні вимоги до якості обслуговування (QoS), такі як джиттер, втрата пакетів loss або пріоритетність трафіку, що є критично важливим для мультимедійних або критичних сервісів.

4. Енергоефективність та обчислювальні витрати. Важливо оцінити, наскільки використання DQN є енергетично доцільним у порівнянні з класичними алгоритмами, особливо в умовах обмежених ресурсів (наприклад, у периферійних пристроях або IoT-середовищах).

Література

1. Sutton R. S., Barto A. G. Reinforcement Learning: An Introduction. 2nd ed. Cambridge, MA, USA: MIT Press, 2018. URL: https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

2. Song Y., Qian X., Zhang N., Wang W., Xiong A. QoS Routing Optimization Based on Deep Reinforcement Learning in SDN. Computers, Materials & Continua, 2024, Vol. 79, No 2, P. 3007–3021. DOI: 10.32604/cmc.2024.051217

3. Stampa G., Arias M., Sanchez-Charles D., Muntes-Mulero V., Cabellos A. A Deep-Reinforcement Learning Approach for Software-Defined Networking Routing Optimization. arXiv preprint arXiv:1709.07080, 2017. URL: https://arxiv.org/abs/1709.07080

4. He Y., Xiao G., Zhu J., Zou T., Liang Y. Reinforcement learning-based SDN routing scheme empowered by causality detection and GNN. Frontiers in Computational Neuroscience, 2024, Vol. 18, Article 1393025. URL: https://www.frontiersin.org/articles/10.3389/fncom.2024.1393025/full

5. Kim B., Kong J. H., Moore T. J., Dagefu F. T. Deep Reinforcement Learning Based Routing for Heterogeneous Multi-Hop Wireless Networks. arXiv preprint arXiv:2508.14884 [eess.SP], 2025. URL: https://arxiv.org/abs/2508.14884

References

1. Sutton R. S., Barto A. G. (2018). Reinforcement Learning: An Introduction. 2nd ed. Cambridge, MA, USA: MIT Press. URL: https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

2. Song Y., Qian X., Zhang N., Wang W., Xiong A. (2024). QoS Routing Optimization Based on Deep Reinforcement Learning in SDN. Computers, Materials & Continua, 79(2), s. 3007–3021. DOI: 10.32604/cmc.2024.051217

3. Stampa G., Arias M., Sanchez-Charles D., Muntes-Mulero V., Cabellos A. (2017). A Deep-Reinforcement Learning Approach for Software-Defined Networking Routing Optimization [Підхід глибокого підкріплювального навчання для оптимізації маршрутизації в SDN]. arXiv preprint arXiv:1709.07080. URL https://arxiv.org/abs/1709.07080

4. He Y., Xiao G., Zhu J., Zou T., Liang Y. (2024). Reinforcement learning-based SDN routing scheme empowered by causality detection and GNN. Frontiers in Computational Neuroscience, 18, Article 1393025. URL: https://www.frontiersin.org/articles/10.3389/fncom.2024.1393025/full

5. Kim B., Kong J. H., Moore T. J., Dagefu F. T. (2025). Deep Reinforcement Learning Based Routing for Heterogeneous Multi-Hop Wireless. arXiv preprint arXiv:2508.14884 [eess.SP]. URL: https://arxiv.org/abs/2508.14884

Дата першого надходження статті до видання: 22.10.2025
Дата прийняття статті до друку після рецензування: 24.11.2025
Дата публікації (оприлюднення) статті: