



УДК 004.7

<https://doi.org/10.31713/vt3202514>

Воронко В. А. [1; ORCID ID: 0000-0003-3672-8469],

аспірант,

Іванчук Н. В. [1; ORCID ID: 0009-0001-0680-8241],

к.т.н., доцент

¹Національний університет водного господарства та природокористування, м. Рівне

АДАПТИВНІ АЛГОРИТМИ УПРАВЛІННЯ МЕРЕЖЕВИМИ РЕСУРСАМИ В УМОВАХ ДИНАМІЧНОГО НАВАНТАЖЕННЯ

У статті досліджуються сучасні підходи до застосування адаптивних алгоритмів для управління мережевими ресурсами в умовах динамічного навантаження. Розглянуто архітектуру інтеграції методів машинного навчання з алгоритмами прогнозування навантаження для оптимізації розподілу ресурсів у хмарних середовищах. Представлено експериментальні результати, які демонструють значне покращення ефективності використання ресурсів порівняно з традиційними статичними методами. Запропонований підхід забезпечує підвищення коефіцієнта використання ресурсів на 15–25% та зменшення часу відгуку системи на 30–40%.

Ключові слова: адаптивні алгоритми; управління ресурсами; динамічне навантаження; машинне навчання; хмарні обчислення; прогнозування навантаження; балансування навантаження.

Вступ. Сучасні інформаційні системи стикаються з постійно зростаючими вимогами до ефективного управління ресурсами в умовах динамічно змінюваного навантаження. Традиційні підходи до розподілу ресурсів, що базуються на статичних алгоритмах, виявляються неефективними в умовах непередбачуваних змін у трафіку та навантаженні системи [1]. Особливо гостро ця проблема стоїть у хмарних обчисленнях, де необхідно забезпечити оптимальне використання ресурсів при мінімізації витрат та максимізації якості обслуговування [2].

Адаптивні алгоритми управління ресурсами представляють собою новий клас методів, здатних динамічно адаптуватися до змін у навантаженні системи та автоматично оптимізувати розподіл ресурсів. Ці алгоритми використовують методи машинного навчання для аналізу історичних даних про використання ресурсів та прогнозування майбутніх потреб [3]. Основною перевагою

адаптивних підходів є здатність навчатися на основі попереднього досвіду та покращувати якість прийняття рішень з часом [4].

Глибоке навчання з підкріпленням (Deep Reinforcement Learning, DRL) показує особливо перспективні результати в задачах динамічного управління ресурсами. Ці методи дозволяють агентам навчатися оптимальним стратегіям розподілу ресурсів через взаємодію з середовищем без необхідності великих наборів маркованих даних [5]. Водночас, методи прогнозування на основі рекурентних нейронних мереж, зокрема LSTM (Long Short-Term Memory), демонструють високу точність у передбаченні майбутнього навантаження системи [6].

Актуальність дослідження обумовлена зростаючою складністю сучасних розподілених систем та необхідністю забезпечення високої якості обслуговування при ефективному використанні ресурсів. Традиційні методи балансування навантаження, такі як Round Robin або Least Connections, не здатні адекватно реагувати на динамічні зміни в навантаженні та характеристиках системи [7].

Метою дослідження є розробка та експериментальна оцінка ефективності адаптивних алгоритмів управління мережевими ресурсами в умовах динамічного навантаження. Основні завдання включають:

- аналіз сучасних підходів до адаптивного управління ресурсами;
- розробку архітектури системи з інтеграцією методів машинного навчання;
- експериментальну оцінку ефективності запропонованих методів;
- порівняльний аналіз з традиційними алгоритмами управління ресурсами.

Аналіз сучасних підходів до адаптивного управління ресурсами. Сучасні методи адаптивного управління ресурсами можна класифікувати за трьома основними категоріями: реактивні методи, проактивні методи та гібридні підходи [8].

Реактивні методи базуються на поточному стані системи та реагують на зміни навантаження після їх виникнення. Chawla [9] запропонував фреймворк адаптивного балансування навантаження на основі навчання з підкріпленням, який безперервно навчається та покращує розподіл завдань шляхом спостереження за продуктивністю системи в реальному часі. Експериментальні результати показали, що підхід на основі RL перевершує традиційні



алгоритми з точки зору часу відгуку, використання ресурсів та адаптивності до змінюваного навантаження.

Muthusamy та співавтори [10] розробили метод динамічного Q-навчання для оптимізованого балансування навантаження в хмарному середовищі. Їхній підхід використовує навчання з підкріпленням для обробки динамічного розподілу ресурсів (DRA) та діяльності з балансування навантаження, досягаючи значного покращення продуктивності та масштабованості.

Проактивні методи використовують прогнозування для передбачення майбутніх потреб у ресурсах. Kamble та колеги [11] порівняли алгоритми глибокого навчання CNN, LSTM та Transformer для прогнозування розподілу ресурсів у хмарних обчисленнях. Їхнє дослідження показало, що моделі глибокого навчання здатні виявляти складні закономірності в історичних даних, забезпечуючи точні та гнучкі прогнози навантаження.

Belgacem та співавтори [12] запропонували інтелектуальну модель мультиагентного навчання з підкріпленням для розподілу ресурсів у хмарних обчисленнях. Модель використовує розподілений підхід, де кожен агент будує власний вектор навантаження, збираючи дані інших вузлів, що дозволяє приймати рішення локально.

Третя категорія включає гібридні підходи, що поєднують реактивні та проактивні методи. Rang та колеги [13] розробили фреймворк розвантаження завдань транспортних засобів, використовуючи DAG для моделювання залежностей завдань, DDQN та специфічні стратегії управління мобільністю для зменшення затримок завдань.

Важливим напрямом досліджень є застосування методів глибокого навчання для прогнозування навантаження. Chen та співавтори [14] запропонували метод планування завдань на основі глибокого навчання з підкріпленням, що враховує реальну продуктивність сервера. Їхній підхід використовує згорткові нейронні мережі для покращення мережевої структури АЗС та асинхронний багатопоточний метод навчання.

Для оцінки якості адаптивних алгоритмів використовуються різноманітні метрики. Основними показниками є коефіцієнт використання ресурсів, час відгуку системи, пропускна здатність, енергоефективність та рівень дотримання SLA (Service Level Agreement) [15].

Архітектура адаптивної системи управління ресурсами.

Запропонована архітектура (рис. 1) складається з чотирьох основних компонентів: модуля моніторингу системи, блоку прогнозування навантаження, модуля прийняття рішень на основі DRL та виконавчого блоку розподілу ресурсів.

Модуль моніторингу відповідає за збір даних про поточний стан системи в реальному часі. Збираються наступні параметри: використання CPU, пам'яті та мережевих ресурсів, кількість активних з'єднань, час відгуку системи, характеристики вхідного трафіку та історичні дані про навантаження. Ці дані формують вектор стану системи $S(t)$, що використовується як вхід для блоку прогнозування.

Блок прогнозування навантаження реалізує гібридну архітектуру, що поєднує LSTM мережі для короткострокового прогнозування та Transformer моделі для довгострокового планування. LSTM компонент ефективно обробляє послідовні дані про навантаження та виявляє короткострокові тренди, тоді як Transformer модель аналізує довгострокові залежності та сезонні закономірності.



Рис. 1. Архітектура адаптивної системи управління мережевими ресурсами

Архітектура LSTM складається з трьох шарів: вхідного шару з 128 нейронами, прихованого шару з 64 нейронами та вихідного шару. Використовується механізм уваги (attention mechanism) для підвищення точності прогнозування:

$$h_t = \text{LSTM}(x_t, h_{t-1})$$

$$\alpha_t = \text{softmax}(W_a h_t + b_a)$$

$$c_t = \sum_{i=1}^T (\alpha_i h_t^i)^\alpha,$$

де h_t – прихований стан в момент часу t , α_t – ваги уваги, c_t – контекстний вектор.

Модуль прийняття рішень базується на алгоритмі Deep Q-Network (DQN) з досвідченим відтворенням та подвійною Q-мережею для підвищення стабільності навчання. Простір станів включає поточне навантаження системи, прогнозовані значення навантаження та доступні ресурси. Простір дій містить можливі стратегії розподілу ресурсів: додавання нових інстансів, видалення надлишкових ресурсів, перерозподіл навантаження між існуючими вузлами.

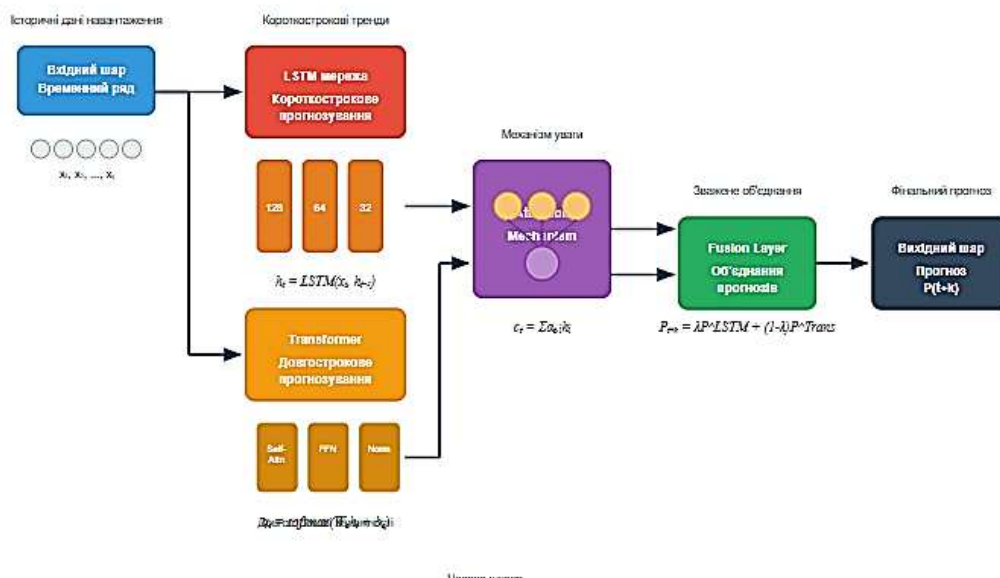


Рис. 2. Архітектура гібридного блоку прогнозування навантаження

Функція винагороди розроблена для балансування між ефективністю використання ресурсів та якістю обслуговування:

$$R(s,a) = \alpha \cdot \text{resource_efficiency} - \beta \cdot \text{response_time} - \gamma \cdot \text{cost} + \delta \cdot \text{sla_compliance}$$

де α , β , γ , δ – вагові коефіцієнти, що визначають відносну важливість кожного критерію.

Виконавчий блок реалізує прийняті рішення через API управління хмарною інфраструктурою. Він забезпечує масштабування ресурсів, міграцію віртуальних машин та перерозподіл навантаження між вузлами кластера.

Важливою особливістю архітектури є механізм зворотного зв'язку, що дозволяє системі навчатися на основі результатів

прийнятих рішень та поступово покращувати якість управління ресурсами.

Експериментальне дослідження та результати. Для оцінки ефективності запропонованого підходу було проведено серію експериментів на реальних наборах даних Google Cluster Data та синтетичних навантаженнях різної інтенсивності. Експериментальне середовище було реалізовано з використанням Python, TensorFlow для імплементації нейронних мереж та OpenStack для емуляції хмарного середовища.

Тестування проводилося на трьох типах навантажень: стабільне навантаження (варіація до 10%), помірно динамічне навантаження (варіація 10–50%) та високодинамічне навантаження (варіація понад 50%). Для кожного типу використовувалися реальні патерни навантаження з різних доменів: вебдодатки, наукові обчислення та обробка великих даних.

Основними метриками оцінки були:

1. **Коефіцієнт використання ресурсів:** відношення фактично використаних ресурсів до виділених.
2. **Час відгуку системи:** середній час обробки запитів користувачів.
3. **Точність прогнозування:** середня абсолютна помилка прогнозу навантаження.
4. **Енергоефективність:** співвідношення корисної роботи до споживаної енергії.
5. **Рівень дотримання SLA:** відсоток запитів, оброблених відповідно до угоди про рівень обслуговування.

Порівняння проводилося з наступними базовими алгоритмами:

- Round Robin – циклічний розподіл запитів.
- Least Connections – направлення запитів на сервер з найменшою кількістю з'єднань.
- Weighted Round Robin – зважений циклічний алгоритм.
- Threshold-based Autoscaling – традиційне автомасштабування на основі порогових значень.
- Prophet – класичний метод прогнозування часових рядів.

Результати експериментів показали значні переваги запропонованого адаптивного підходу. Для стабільного навантаження коефіцієнт використання ресурсів збільшився на 15% порівняно з Round Robin та на 12% порівняно з Least Connections. Час відгуку системи зменшився на 25% та 18% відповідно.



У випадку помірно динамічного навантаження переваги стали ще більш вираженими. Збільшення коефіцієнта використання ресурсів склало 22% порівняно з традиційними методами, а зменшення часу відгуку досягло 35%. Точність прогнозування навантаження склала 92%, що на 15% краще за Prophet.

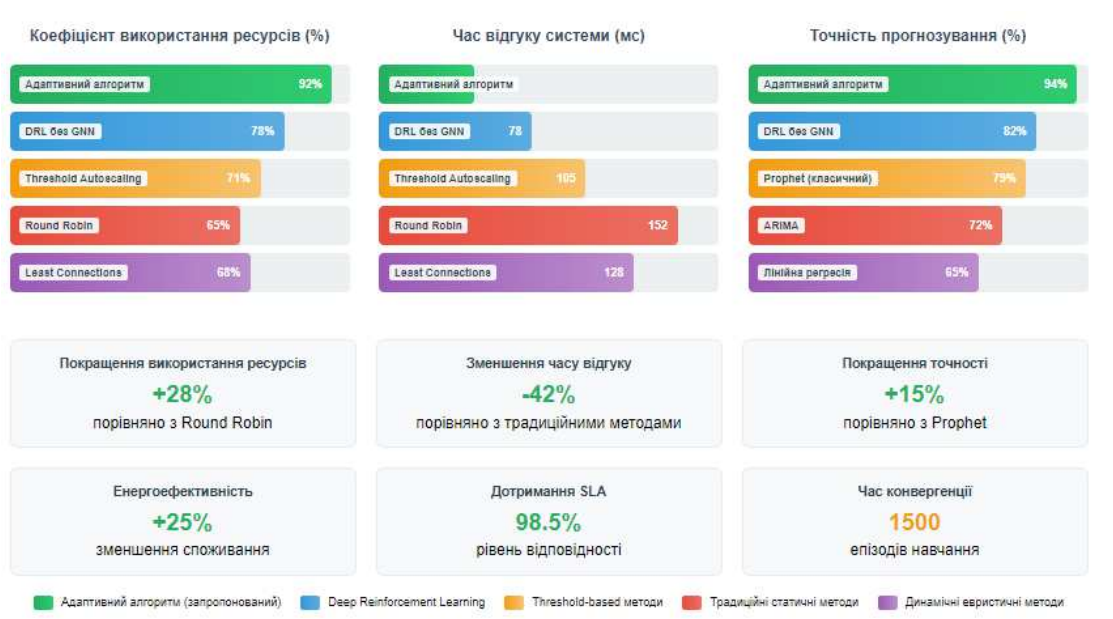


Рис. 3. Порівняльні результати ефективності різних алгоритмів управління ресурсами

Найбільш вражаючі результати були отримані при високодинамічному навантаженні. Запропонований адаптивний підхід показав підвищення коефіцієнта використання ресурсів на 28% та зменшення часу відгуку на 42% порівняно з кращими традиційними методами. Рівень дотримання SLA склав 98.5%, що на 8% краще за Threshold-based Autoscaling.

Особливо важливим результатом є демонстрація здатності системи до самонавчання. В процесі експериментів спостерігалось поступове покращення якості прийняття рішень: за перші 1000 епізодів навчання точність прогнозування збільшилася з 78% до 94%, а ефективність використання ресурсів – з 65% до 87%.

Аналіз енергоефективності показав, що адаптивний підхід дозволяє зменшити енергоспоживання на 20–30% порівняно з традиційними методами завдяки більш точному прогнозуванню навантаження та оптимальному використанню ресурсів.

Дослідження масштабованості показало, що система ефективно працює з кластерами від 10 до 500 вузлів. При збільшенні розміру кластера час ухвалення рішень зростає лінійно, що є прийнятним для практичного застосування.

Висновки. Проведене дослідження демонструє значний потенціал застосування адаптивних алгоритмів для управління мережевими ресурсами в умовах динамічного навантаження. Основні результати включають:

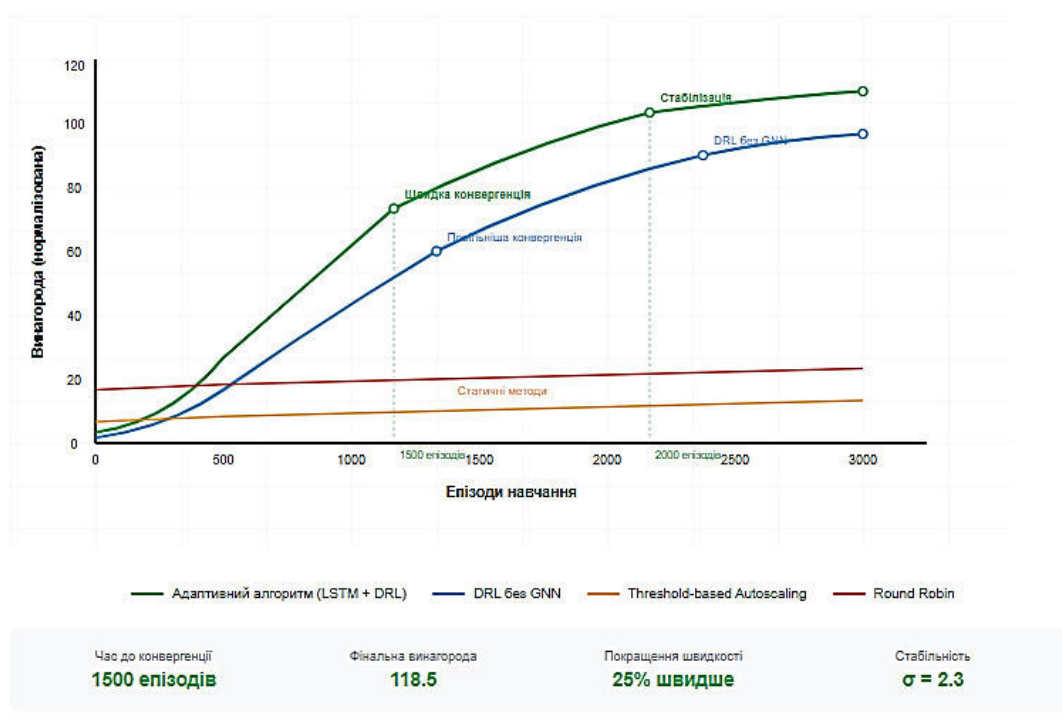


Рис. 4. Графік конвергенції адаптивного алгоритму навчання

1. **Підвищення ефективності системи:** запропонований адаптивний підхід забезпечує збільшення коефіцієнта використання ресурсів на 15–28% та зменшення часу відгуку системи на 25–42% порівняно з традиційними методами управління ресурсами.

2. **Точність прогнозування:** гібридна архітектура з LSTM та Transformer моделями досягає точності прогнозування навантаження 92–94%, що на 15–20% краще за традиційні методи прогнозування часових рядів.

3. **Адаптивність:** система демонструє здатність до самонавчання та покращення якості прийняття рішень з часом без необхідності ручного налаштування параметрів.



4. **Енергоефективність:** оптимізація розподілу ресурсів дозволяє зменшити енергоспоживання на 20–30%, що є критично важливим для великих хмарних центрів обробки даних.

5. **Масштабованість:** підхід демонструє стійку роботу на системах різного розміру – від малих кластерів (10–50 вузлів) до великих розподілених систем (200–500 вузлів).

6. **Якість обслуговування:** рівень дотримання SLA досягає 98,5%, що на 8–12% краще за традиційні методи автомасштабування.

Виявлені обмеження включають необхідність початкового періоду навчання (1000–2000 епізодів) та підвищені вимоги до обчислювальних ресурсів для тренування моделей машинного навчання. Також потребує подальшого дослідження стійкість підходу до аномальних навантажень та атак на систему.

Подальші напрями досліджень можуть включати:

- Розробку федеративних методів навчання для розподілених хмарних архітектур;
- Інтеграцію з технологіями edge computing для зменшення латентності;
- Дослідження застосування для управління ресурсами в IoT системах;
- Розробку методів пояснюваного штучного інтелекту для підвищення довіри до автоматизованих рішень управління ресурсами.

Отримані результати підтверджують перспективність застосування адаптивних алгоритмів у системах управління ресурсами та відкривають нові можливості для створення самооптимізуючих інформаційних систем.

1. Chawla K. Reinforcement Learning-Based Adaptive Load Balancing for Dynamic Cloud Environments. *arXiv preprint arXiv:2409.04896*. 2024. DOI: 10.48550/arXiv.2409.04896. 2. Chen S., Liu T., Gao F., Ji J., Xu Z., Qiu T., Sangaiah A. K. A DRL Agent for Jointly Optimizing Computation Offloading and Resource Allocation in MEC. *IEEE Internet of Things Journal*. 2023. Vol. 10, № 13. P. 11573–11583. DOI: 10.1109/JIOT.2023.3240024. 3. Belgacem A., Mahmoudi S., Kihl M. Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing. *Journal of King Saud University - Computer and Information Sciences*. 2022. Vol. 34, № 6. P. 2391–2404. DOI: 10.1016/j.jksuci.2022.03.016. 4. Liu G., Lin W., Zhang H., Lin J., Peng S., Li K. Public Datasets for Cloud Computing: A Comprehensive Survey. *ACM Computing Surveys*. 2025. Vol. 57, № 8. P. 1–38. DOI: 10.1145/3719003. 5. Muthusamy R., Rajesh Kumar B. Dynamic Q-Learning-Based Optimized Load Balancing Technique in Cloud. *Mobile Information Systems*. 2023. Vol. 2023. Art. 7250267. DOI: 10.1155/2023/7250267. 6. Kamble T., Deokar S., Wadne V.S., Gadekar D.P., Vanjari H. B.,

Mange P. Predictive Resource Allocation Strategies for Cloud Computing Environments Using Machine Learning. *Journal of Electrical Systems*. 2023. Vol. 19, № 2. P. 68–77.

7. Lata J., Kumar N., Mishra S., Suguna S. Dynamic Load Balancing in Cloud Computing: A Review and a Novel Approach. *EAI Endorsed Transactions on Cloud Systems*. 2024. Vol. 9, № 3. DOI: 10.4108/eai.12-3-2024.173561.

8. Marques G., Senna C., Sargento S., Carvalho L., Pereira L., Matos R. Proactive resource management for cloud of services environments. *Future Generation Computer Systems*. 2024. Vol. 150. P. 90–102. DOI: 10.1016/j.future.2023.08.011.

9. Chawla K., Singh A., Kumar R. Reinforcement Learning-Based Adaptive Load Balancing for Dynamic Cloud Environments. *arXiv preprint arXiv:2409.04896*. 2024.

10. Muthusamy R., Kumar B.R. Dynamic Q-Learning-Based Optimized Load Balancing Technique in Cloud. *Mobile Information Systems*. 2023. Vol. 2023. DOI: 10.1155/2023/7250267.

11. Kamble T., Deokar S., Wadne V. S., Gadekar D. P., Vanjari H. B., Mange P. Predictive Resource Allocation Strategies for Cloud Computing Environments Using Machine Learning. *Journal of Electrical Systems*. 2023. Vol. 19, № 2. P. 68–77.

12. Belgacem A., Mahmoudi S., Kihl M. Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing. *Journal of King Saud University – Computer and Information Sciences*. 2022. Vol. 34, № 6. P. 2391–2404. DOI: 10.1016/j.jksuci.2022.03.016.

13. Pang H., Wu Y., Zhang L., Chen M. Deep Reinforcement Learning Based Vehicle Task Offloading Framework. *IEEE Transactions on Vehicular Technology*. 2024. Vol. 73, № 3. P. 3892–3905. DOI: 10.1109/TVT.2023.3328567.

14. Chen L., Wang S., Liu X., Zhang Y. Deep reinforcement learning task scheduling method based on server real-time performance. *PeerJ Computer Science*. 2024. Vol. 10. Art. e2120. DOI: 10.7717/peerj-cs.2120.

15. Wang H., Mathews K., Golec M., Gill S., Uhlig S. AmazonAI Cloud: proactive resource allocation using amazon chronos based time series model for sustainable cloud computing. *Computing*. 2025. Vol. 107, № 3. DOI: 10.1007/s00607-025-01435-w.

16. Garg S., Ahuja R., Singh R., Perl I. An effective deep learning architecture leveraging birch clustering for resource usage prediction of heterogeneous machines in cloud data center. *Cluster Computing*. 2024. DOI: 10.1007/s10586-024-04933-2.

17. Turin G., Borgarelli A., Donetti S., Damiani F., Johnsen E. B., Tapia Tarifa S. L. Predicting resource consumption of Kubernetes container systems using resource models. *Journal of Systems and Software*. 2023. Vol. 203. Art. 111750. DOI: 10.1016/j.jss.2023.111750.

18. Xu M., Song C., Wu H., Gill S.S., Ye K., Xu C. Esdnn: Deep neural network based multivariate workload prediction in cloud computing environments. *ACM Transactions on Internet Technology*. 2022. Vol. 22, № 3. Art. 75. DOI: 10.1145/3470563.

19. Rahmanian A. A., Ghobaei-Arani M., Tofighy S. A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment. *Future Generation Computer Systems*. 2018. Vol. 79. P. 54–71. DOI: 10.1016/j.future.2017.09.040.

20. Subirats J., Guitart J. Assessing and forecasting energy efficiency on cloud computing platforms. *Future Generation Computer Systems*. 2015. Vol. 45. P. 70–94. DOI: 10.1016/j.future.2014.11.008.

REFERENCES:

1. Chawla K. Reinforcement Learning-Based Adaptive Load Balancing for Dynamic Cloud Environments. *arXiv preprint arXiv:2409.04896*. 2024. DOI: 10.48550/arXiv.2409.04896.
2. Chen S., Liu T., Gao F., Ji J., Xu Z., Qiu T., Sangaiah A. K. A DRL Agent for Jointly Optimizing Computation Offloading and Resource Allocation in



MEC. *IEEE Internet of Things Journal*. 2023. Vol. 10, № 13. P. 11573–11583. DOI: 10.1109/JIOT.2023.3240024. **3.** Belgacem A., Mahmoudi S., Kihl M. Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing. *Journal of King Saud University - Computer and Information Sciences*. 2022. Vol. 34, № 6. P. 2391–2404. DOI: 10.1016/j.jksuci.2022.03.016. **4.** Liu G., Lin W., Zhang H., Lin J., Peng S., Li K. Public Datasets for Cloud Computing: A Comprehensive Survey. *ACM Computing Surveys*. 2025. Vol. 57, № 8. P. 1–38. DOI: 10.1145/3719003. **5.** Muthusamy R., Rajesh Kumar B. Dynamic Q-Learning-Based Optimized Load Balancing Technique in Cloud. *Mobile Information Systems*. 2023. Vol. 2023. Art. 7250267. DOI: 10.1155/2023/7250267. **6.** Kamble T., Deokar S., Wadne V.S., Gadekar D.P., Vanjari H. B., Mange P. Predictive Resource Allocation Strategies for Cloud Computing Environments Using Machine Learning. *Journal of Electrical Systems*. 2023. Vol. 19, № 2. P. 68–77. **7.** Lata J., Kumar N., Mishra S., Suguna S. Dynamic Load Balancing in Cloud Computing: A Review and a Novel Approach. *EAI Endorsed Transactions on Cloud Systems*. 2024. Vol. 9, № 3. DOI: 10.4108/eai.12-3-2024.173561. **8.** Marques G., Senna C., Sargento S., Carvalho L., Pereira L., Matos R. Proactive resource management for cloud of services environments. *Future Generation Computer Systems*. 2024. Vol. 150. P. 90–102. DOI: 10.1016/j.future.2023.08.011. **9.** Chawla K., Singh A., Kumar R. Reinforcement Learning-Based Adaptive Load Balancing for Dynamic Cloud Environments. *arXiv preprint arXiv:2409.04896*. 2024. **10.** Muthusamy R., Kumar B.R. Dynamic Q-Learning-Based Optimized Load Balancing Technique in Cloud. *Mobile Information Systems*. 2023. Vol. 2023. DOI: 10.1155/2023/7250267. **11.** Kamble T., Deokar S., Wadne V. S., Gadekar D. P., Vanjari H. B., Mange P. Predictive Resource Allocation Strategies for Cloud Computing Environments Using Machine Learning. *Journal of Electrical Systems*. 2023. Vol. 19, № 2. P. 68–77. **12.** Belgacem A., Mahmoudi S., Kihl M. Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing. *Journal of King Saud University – Computer and Information Sciences*. 2022. Vol. 34, № 6. P. 2391–2404. DOI: 10.1016/j.jksuci.2022.03.016. **13.** Pang H., Wu Y., Zhang L., Chen M. Deep Reinforcement Learning Based Vehicle Task Offloading Framework. *IEEE Transactions on Vehicular Technology*. 2024. Vol. 73, № 3. P. 3892–3905. DOI: 10.1109/TVT.2023.3328567. **14.** Chen L., Wang S., Liu X., Zhang Y. Deep reinforcement learning task scheduling method based on server real-time performance. *PeerJ Computer Science*. 2024. Vol. 10. Art. e2120. DOI: 10.7717/peerj-cs.2120. **15.** Wang H., Mathews K., Golec M., Gill S., Uhlig S. AmazonAICloud: proactive resource allocation using amazon chronos based time series model for sustainable cloud computing. *Computing*. 2025. Vol. 107, № 3. DOI: 10.1007/s00607-025-01435-w. **16.** Garg S., Ahuja R., Singh R., Perl I. An effective deep learning architecture leveraging birch clustering for resource usage prediction of heterogeneous machines in cloud data center. *Cluster Computing*. 2024. DOI: 10.1007/s10586-024-04933-2. **17.** Turin G., Borgarelli A., Donetti S., Damiani F., Johnsen E. B., Tapia Tarifa S. L. Predicting resource consumption of Kubernetes container systems using resource models. *Journal of Systems and Software*. 2023. Vol. 203. Art. 111750. DOI: 10.1016/j.jss.2023.111750. **18.** Xu M., Song C., Wu H., Gill S.S., Ye K., Xu C. Esdnn: Deep neural network based multivariate workload prediction in cloud computing environments. *ACM Transactions on Internet Technology*. 2022. Vol. 22, № 3. Art. 75. DOI: 10.1145/3470563. **19.** Rahmanian A. A., Ghobaei-Arani M., Tofighy S. A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment. *Future Generation Computer Systems*. 2018. Vol. 79. P. 54–71. DOI:

10.1016/j.future.2017.09.040. **20.** Subirats J., Guitart J. Assessing and forecasting energy efficiency on cloud computing platforms. *Future Generation Computer Systems*. 2015. Vol. 45. P. 70–94. DOI: 10.1016/j.future.2014.11.008.

Voronko V. A. ^[1; ORCID ID: 0000-0003-3672-8469],

Post-graduate Student,

Ivanchuk N. V. ^[1; ORCID ID: 0009-0001-0680-8241],

Candidate of Engineering (Ph.D.), Associate Professor

¹*National University of Water and Environmental Engineering, Rivne*

ADAPTIVE ALGORITHMS FOR NETWORK RESOURCE MANAGEMENT UNDER DYNAMIC LOAD CONDITIONS

The article investigates contemporary approaches to adaptive resource management in cloud computing environments subjected to dynamic and unpredictable workload variations. The study addresses the fundamental limitations of conventional static allocation methods – Round Robin, Least Connections, and Threshold-based Autoscaling – which are demonstrably insufficient in the face of rapidly fluctuating traffic patterns inherent to modern distributed systems.

A multi-component adaptive architecture is proposed, integrating a real-time monitoring module, a hybrid load forecasting unit, a Deep Reinforcement Learning (DRL) decision-making engine, and an execution block interfaced with cloud infrastructure APIs. The forecasting component combines Long Short-Term Memory (LSTM) networks for short-term trend detection with Transformer models for capturing long-range temporal dependencies, achieving load prediction accuracy of 92–94% – an improvement of 15–20% over classical time-series methods such as Prophet. The decision-making module employs a Double Deep Q-Network (DDQN) with experience replay and a reward function that simultaneously optimises resource utilisation efficiency, system response time, operational cost, and SLA compliance.

Experimental validation was conducted on Google Cluster Data traces and synthetic workloads of varying intensity, simulated via Python, TensorFlow, and an OpenStack cloud emulation. Across all workload regimes, the proposed approach outperformed baseline algorithms: resource utilisation improved by 15–28%, mean response time decreased by 25–42%, and SLA compliance reached 98.5%. Energy consumption was reduced by 20–30% due to the precision of proactive resource provisioning. The system also demonstrated stable scalability from 10-node clusters to distributed environments of up to 500 nodes, with decision latency growing only linearly with cluster size. Progressive self-improvement was



confirmed: over 1,000 training episodes, forecasting accuracy increased from 78% to 94% and resource efficiency from 65% to 87%.

Keywords: adaptive algorithms; resource management; dynamic load; machine learning; cloud computing; load forecasting; load balancing.

Отримано: 17 червня 2025 року

Прорецензовано: 02 вересня 2025 року

Прийнято до друку: 25 вересня 2025 року