

УДК 621.391

ДОСЛІДЖЕННЯ ІНТЕЛЕКТУАЛЬНОЇ МОДЕЛІ ОДНОШЛЯХОВОЇ МАРШРУТИЗАЦІЇ НА ОСНОВІ МЕТРИК OSPF



Р.С. СИТНИКОВ, Д.В. АНДРУШКО

Харківський національний університет радіоелектроніки

Abstract – The work presents and thoroughly investigates an intelligent single-path routing model based on OSPF protocol metrics and deep learning. The proposed intelligent model is based on a mathematical formalization of the problem in the form of Boolean programming, ensuring the implementation of strictly single-path routing without flow branching. The model is based on a multilayer perceptron (MLP) architecture. The integration of neural network predictive capabilities directly into the path-selection process enables dynamic optimization of interface costs. An experimental study was conducted on a topology with five nodes and six communication links. The work provides a comparative analysis of MLP regression and classification models across bandwidth ranges: 10 Mbit/s – 400 Gbit/s, 1 – 400 Gbit/s, and 1 – 100 Gbit/s. It was found that MLP forecasting accuracy reaches 99–100% when using the “optimal” data set, but decreases significantly with excessive variability in input parameters. An important aspect of the study is the comparison of software implementation environments. It has been found that using Python (TensorFlow and PyTorch) provides 5–10% higher prediction accuracy than MATLAB, which is explained by the specialization of Python libraries for network analytics tasks. The conclusions of the work justify the “accuracy limit” of MLPs due to the neglect of the network graph’s topological structure and point to the promising transition to graph neural networks (GNNs) for large systems. At the same time, it is emphasized that, due to its low computational complexity, MLP remains the optimal choice for local solutions and routers with limited resources, where microsecond-level decision-making speed is critical.

Анотація – У роботі представлено та детально досліджено інтелектуальну модель одношляхової маршрутизації, що базується на використанні метрик протоколу OSPF і технологій глибокого навчання. Запропонована інтелектуальна модель базується на математичній формалізації задачі у формі булевого програмування, що забезпечує реалізацію суворо одношляхової маршрутизації без розгалуження потоків. Основу моделі становить архітектура багatoшарового перцептрона (Multilayer Perceptron, MLP). Інтеграція предиктивних можливостей нейронних мереж безпосередньо у процес вибору шляху дозволяє здійснювати динамічну оптимізацію вартості інтерфейсів. Експериментальне дослідження проведено на топології з п’яти вузлів та шести каналів зв’язку. В роботі виконано порівняльний аналіз регресійних і класифікаційних моделей MLP при різних діапазонах пропускної здатності: 10 Мбіт/с – 400 Гбіт/с, 1 – 400 Гбіт/с та 1 – 100 Гбіт/с. Встановлено, що точність прогнозування MLP сягає 99–100% за умови використання «оптимального» ряду даних, проте суттєво знижується при надмірній варіативності вхідних параметрів. Важливим аспектом дослідження є порівняння програмних середовищ реалізації. Виявлено, що використання Python (TensorFlow та PyTorch) забезпечує на 5–10% вищу точність прогнозування порівняно з MATLAB, що пояснюється спеціалізацією бібліотек Python під задачі мережної аналітики. У висновках роботи обґрунтовано «межу точності» MLP через ігнорування топологічної структури графа мережі та вказано на перспективність переходу до графових нейронних мереж (Graph Neural Networks, GNN) для великих систем. Водночас підкреслено, що завдяки низькій обчислювальній складності MLP залишається оптимальним вибором для локальних рішень та маршрутизаторів з обмеженими ресурсами, де критичною є швидкість прийняття рішень у мікросекундному діапазоні.

Вступ

Сучасні інфокомунікаційні мережі характеризуються високою динамікою, складною топологією та постійно зростаючими вимогами до швидкості й надійності передачі даних, зумовленими розвитком технологій 5G, IoT та хмарних обчислень. Ефективне управління потоками трафіку є ключовим для забезпечення якості обслуговування (QoS), проте традиційні протоколи маршрутизації, зокрема OSPF (Open Shortest Path First), часто виявляються малоефективними через свою нездатність швидко адаптуватися до змін у реальному часі [1, 2].

Стандартний протокол OSPF базується на алгоритмі Дейкстри та використовує фіксовані метрики, що описують пропускну здатність інтерфейсів для вибору найкоротшого шляху. Такий детермінований підхід ігнорує контекстні особливості трафіку та динамічні чинники (затримки, джитер, поточну завантаженість каналів та стабільність ліній), що призводить до виникнення «вузьких місць» і неефективного використання ресурсів при пікових навантаженнях. В умовах високої мобільності вузлів або раптових сплесків трафіку традиційні алгоритми не забезпечують необхідної гнучкості, що спричиняє значне зниження загальної продуктивності мережі [1, 3, 4].

Впровадження концепції програмно-конфігурованих мереж (SDN) стало ключовим етапом у розвитку інфраструктури майбутнього, оскільки ця архітектура базується на відділенні рівня управління (Control Plane) від рівня передачі даних (Data Plane). Така декомпозиція дозволяє передати інтелектуальні функції від окремих вузлів до централізованого контролера, який має глобальне бачення топології та здатність гнучко програмувати таблиці маршрутизації через відкриті API. SDN-архітектура забезпечує фундамент для автоматизації управління ресурсами та впровадження проактивних стратегій захисту мережі, що є критичним для забезпечення високого рівня відмовостійкості та кіберстійкості в умовах динамічних навантажень [5 - 7].

Особливе місце в сучасних SDN-рішеннях посідає інтеграція нейромережних моделей, які замінюють статичні евристичні адаптивними механізмами прийняття рішень. Використання методів глибокого навчання дає змогу ефективно вирішувати задачі балансування навантаження та інжинірингу трафіку (Traffic Engineering), орієнтуючись на реальний стан мережі замість фіксованих розрахунків [4, 8]. Поєднання предиктивних можливостей нейронних мереж із гнучкістю SDN дозволяє створювати самонавчальні системи, здатні мінімізувати затримки та запобігати перевантаженням ще до їхнього виникнення, що значно підвищує масштабованість та ефективність інфокомунікаційних мереж [8, 9].

Перспективним напрямком досліджень є перехід до інтелектуальних моделей маршрутизації, що базуються на методах штучного інтелекту, зокрема глибокого навчання (Deep Learning) [4, 10]. Важливе місце в таких дослідженнях посідає використання багатоповового перцептрона MLP (Multilayer Perceptron) як інструменту для аналізу історичних даних та прогнозування стану мережі. Зокрема, дослідження показують, що навіть моделі структури «4-1-16-1» здатні з високою точністю (MAPE менше 10%) прогнозувати затримки на маршрутизаторах, що дозволяє оптимізувати вибір шляху на основі реального стану інфраструктури [11].

Інтеграція методів штучного інтелекту безпосередньо в процес вибору шляху OSPF дозволяє реалізувати динамічну оптимізацію вартості інтерфейсів. Це забезпечує проактивне перенаправлення потоків до виникнення перевантажень, що, за результатами моделювання, підвищує пропускну здатність та покращує затримку приблизно на 30% [3]. Дана стаття присвячена дослідженню інтелектуальної моделі одношляхової маршрутизації, яка трансформує статичний вибір шляху на основі метрик OSPF в адаптивну систему, здатну до самонавчання та ефективного балансування навантаження в динамічних умовах.

I. Огляд досліджень моделей інтелектуальної маршрутизації

Традиційний протокол OSPF базується на статичних метриках пропускної здатності, що обмежує його адаптивність до динамічних змін трафіку та затримок у реальному часі. Впровадження інтелектуальних моделей на основі багатошарового персептрона та інших методів глибокого навчання дозволяє трансформувати процес маршрутизації у проактивний завдяки точному прогнозуванню стану мережі. Представлений аналіз розглядає перехід від класичних евристичних алгоритмів до нейромережних архітектур, які забезпечують суттєве підвищення продуктивності та надійності сучасних інфокомунікаційних мереж.

У відомих дослідженнях інших авторів [11] було розглянуто фундаментальні обмеження протоколу OSPF та детально досліджено використання багатошарового персептрона для вирішення проблеми прогнозування затримок на маршрутизаторах. В роботі було проведено порівняльне дослідження цієї ж архітектури MLP («4-1-16-1») із нейронечіткою мережею (ANFIS) для потреб залізничного транспорту. Було зроблено висновки, що MLP, навчений за алгоритмом Левенберга-Марквардта, забезпечує високу точність прогнозування (середня похибка MAPE становить 1,94%), що дозволяє OSPF ефективно функціонувати в умовах постійної зміни обсягів трафіку.

В результаті дослідження [10] можливості заміни складної евристики глибокими нейронними мережами (DNN), які за своєю структурою є розвиненими багатошаровими персептронами (MLP), було зроблено висновок, що модель, навчена на оптимальних рішеннях математичного програмування (MILP), здатна генерувати маршрути в реальному часі з відхиленням від ідеалу лише на 1%. При цьому одна модель може обслуговувати всі пари вузлів-джерел та вузлів-призначень в мережі.

У сучасних публікаціях [3] було досліджено інтеграцію MLP разом із моделями XGBoost, Random Forest та LSTM безпосередньо у процес прийняття рішень OSPF. Було зроблено висновки, що використання ШІ для динамічного призначення вартості інтерфейсів дозволяє покращити пропускну здатність мережі на 30% порівняно зі стандартними налаштуваннями.

На основі проведеного дослідження [1] алгоритму Q-learning як альтернативи статичному алгоритму Дейкстри було зроблено висновки, що агент підкріплювального навчання RL дозволяє зменшити середню затримку передачі пакетів на 15% та забезпечити більш збалансований розподіл навантаження між вузлами розподіленої мережі. А у роботі [5] було досліджено гібридний підхід, що поєднує генетичні алгоритми та нейронні мережі Хопфілда. Було зроблено висновок, що такі еволюційні методи здатні знаходити оптимальні рішення менш ніж за 5 секунд, що є критично важливим для динамічних систем, де швидкість збіжності OSPF є недостатньою.

У відомих працях [2, 7, 12] було досліджено перехід від класичних MLP до графових нейронних мереж (Graph Neural Networks, GNN), які здатні безпосередньо обробляти топологічну структуру мережі. Так, у роботі [12] було досліджено архітектуру GENConv, де MLP використовується лише як фінальний декодер для класифікації ре-

бер графа. Було зроблено висновки, що такий підхід забезпечує успішну побудову маршрутів у 94% випадків навіть на топологіях, які не були представлені під час навчання моделі. А за результатами дослідження [2] алгоритму GNN-DRL, який поєднує структурну обізнаність GNN із гнучкістю навчання з підкріпленням, було зроблено висновок, що цей підхід знижує максимальну завантаженість каналів на 13,92% та затримку на 9,48% порівняно з існуючими інтелектуальними системами.

Попередні дослідження демонструють, що застосування нейронних мереж забезпечує еволюцію OSPF від статичного алгоритму до інтелектуальної системи управління навантаженням. Архітектура MLP залишається базовим і надійним інструментом для прогнозування затримок та оптимізації вартості шляхів завдяки простоті впровадження та низькій похибці. Водночас, розвиток GNN відкриває шлях до створення універсальних моделей маршрутизації, здатних до миттєвого узагальнення знань на нові, раніше невідомі конфігурації мереж, що є ключовим фактором для розвитку інфраструктури 5G та Інтернету речей [2, 12].

II. Математичний опис задачі маршрутизації за метрикою OSPF

Загалом протокол OSPF виконує пошук найкоротшого шляху за допомогою алгоритму Дейкстри, але при дослідженні задачу одношляхової маршрутизації з метрикою протоколу OSPF було розв'язано у формі задачі булевого програмування. Кількість каналів зв'язку в мережі (n) визначає розмірність вектора $\vec{x}$, координати $x_{i,j}$ якого характеризують долю потоку в каналі зв'язку між i -м та j -м вузлами. Розмірність вектора метрик $\vec{f}$ також відповідає числу каналів зв'язку в мережі (n), координати $f_{i,j}$ якого характеризують метрику каналу зв'язку між i -м та j -м вузлами.

Для реалізації одношляхової маршрутизації на координати вектору x накладаються наступні обмеження:

$$x_{i,j} \in \{0,1\} \quad (i, j = \overline{1, m}; i \neq j), \quad (1)$$

де m – кількість вузлів (маршрутизаторів) у мережі.

Тобто змінні $x_{i,j}$ можуть приймати лише два значення:

$$x_{i,j} = \begin{cases} 1, & \text{якщо потік використовує канал } (i, j); \\ 0, & \text{в іншому випадку.} \end{cases}$$

Булевий зміст змінних (1) гарантує відсутність розгалуження потоку за шляхами мережі, тобто всі пакети потоку будуть передаватися одним шляхом.

Кожному каналу зв'язку буде присвоюватися метрика, що відповідає знаходженню шляху між заданою парою вузлів з мінімальною вартістю ($fval$) за аналогією з властивостями протоколу OSPF:

$$f_{i,j} = bw/c_{i,j}; \quad i, j = \overline{1, m}; i \neq j, \quad (2)$$

де bw – константа, що відповідає значенню максимальної пропускну здатності каналу мережі (для OSPF за замовчуванням 10^8); $c_{i,j}$ – номінальна пропускна здатність (ПЗ) каналу зв'язку між i -м та j -м вузлами.

У ході розв'язання маршрутної задачі необхідно забезпечити виконання умови збереження потоку в кожному з мережних вузлів та в мережі в цілому:

$$\begin{cases} \sum_{j:(i,j)} x_{i,j} - \sum_{j:(j,i)} x_{j,i} = 1 & \text{– для вузла-відправника;} \\ \sum_{j:(i,j)} x_{i,j} - \sum_{j:(j,i)} x_{j,i} = 0 & \text{– для транзитних вузлів;} \\ \sum_{j:(i,j)} x_{i,j} - \sum_{j:(j,i)} x_{j,i} = -1 & \text{– для вузла-одержувача.} \end{cases} \quad (3)$$

Задача пошуку найкоротшого шляху в мережі може формалізуватися як задача булевого програмування, для розв'язання якої може використовуватися інструментарій «*Optimization Toolbox*» середовища *MATLAB*, представлений підпрограмою «*intlinprog*» або аналогічний пакет «*scipy.io*» Python (підпрограма «*linprog*»).

У ході розв'язання задач змішаного цілочисельного лінійного програмування необхідно мінімізувати цільову функцію, представлену лінійною формою

$$\min_x \vec{f}^t \vec{x},$$

при виконанні ряду умов, які представлені у вигляді обмежень рівнянь і нерівностей

$$A \cdot \vec{x} \leq \vec{b}; Aeq \cdot \vec{x} = \vec{beq} \text{ і } \vec{lb} \leq \vec{x} \leq \vec{ub},$$

де $\vec{f}, \vec{x}, \vec{b}, \vec{beq}$ – вектори, A і Aeq – матриці відповідної розмірності; $\vec{lb}$ і $\vec{ub}$ – вектори-стовбці розміру n .

Для опису задачі маршрутизації в формалізмах середовища *MATLAB* або *Python* умови збереження потоку (3) необхідно представити у векторно-матричній формі $Aeq \cdot \vec{x} = \vec{beq}$. У такий спосіб матриця Aeq має розмірність $m \times n$, координати якої приймають числові значення $\{-1; 0; 1\}$ у такий спосіб ($j = \overline{1, m}, i = \overline{1, n}$):

$a_{ji} = 1$, якщо i -й канал зв'язку виходить із j -го вузла;

$a_{ji} = -1$, якщо i -й канал зв'язку входить у j -й вузол;

$a_{ji} = 0$, якщо i -й канал зв'язку не є інцидентним j -му вузлу.

Розмірність вектора beq відповідає числу вузлів у мережі (m), а його координати формується в такий спосіб ($j = \overline{1, m}$):

$beq_j = 1$, якщо j -й вузол є вузлом-відправником пакетів;

$beq_j = -1$, якщо j -й вузол є вузлом-одержувачем пакетів;

$beq_j = 0$, якщо j -й вузол є транзитним вузлом.

У випадку, коли умов у вигляді нерівностей немає, то $A = []$ та $b = []$.

Дослідження інтелектуальної моделі одношляхової маршрутизації на основі метрик OSPF відбувалося на прикладі мережної топології й пропускної здатності її каналів зв'язку, наведеної на рис. 1. Тоді загальне число вузлів у мережі дорівнює п'яти ($m = 5$), а число каналів зв'язку – шести ($n = 6$). Нехай також вузол-відправник пакетів – вузол 1, а вузол-одержувач – вузол 5.

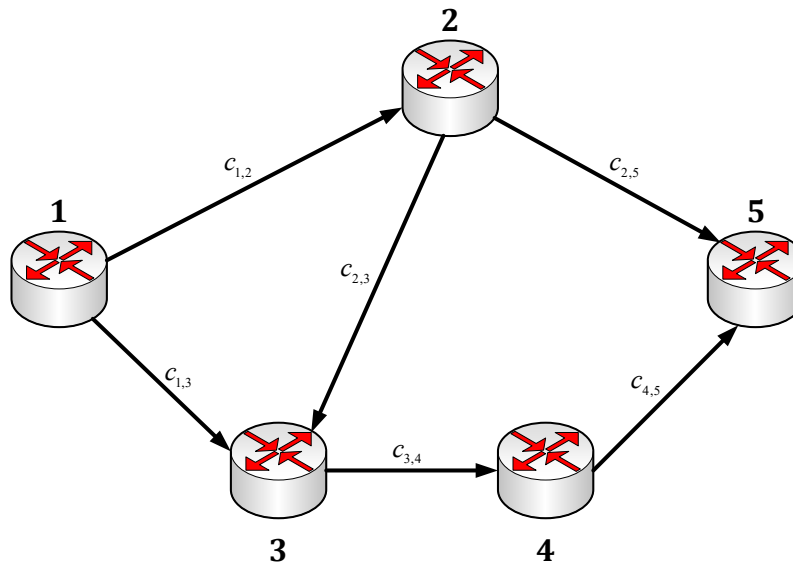


Рис. 1. Топологія досліджуваної мережі

Формуємо шуканий вектор $\vec{x}$. Нехай у рамках моделі, що представлена виразами (1)-(3), він має вигляд:

$$\vec{x} = \begin{bmatrix} x_{1,2} \\ x_{1,3} \\ x_{2,3} \\ x_{2,5} \\ x_{3,4} \\ x_{4,5} \end{bmatrix}. \quad (4)$$

Відповідно до метрики протоколу OSPF вектор вагових коефіцієнтів $\vec{f}$ для прикладу (4) матиме вигляд:

$$\vec{f} = \begin{bmatrix} f_{1,2} \\ f_{1,3} \\ f_{2,3} \\ f_{2,5} \\ f_{3,4} \\ f_{4,5} \end{bmatrix} = \begin{bmatrix} bw/c_{1,2} \\ bw/c_{1,3} \\ bw/c_{2,3} \\ bw/c_{2,5} \\ bw/c_{3,4} \\ bw/c_{4,5} \end{bmatrix}. \quad (5)$$

Формалізуємо умови збереження потоку у вузлах мережі (2):

$$\begin{cases} x_{1,2} + x_{1,3} = 1; \\ -x_{1,2} + x_{2,3} + x_{2,5} = 0; \\ -x_{1,3} - x_{2,3} + x_{3,4} = 0; \\ -x_{3,4} + x_{4,5} = 0; \\ -x_{2,5} - x_{4,5} = -1. \end{cases} \quad (6)$$

Тоді згідно з системою (6) формуємо матрицю Aeq і вектор $\vec{beq}$:

$$Aeq = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 & 0 & -1 \end{bmatrix}; \quad \vec{beq} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}. \quad (7)$$

У векторно-матричному поданні система нерівностей буде мати вигляд $A \cdot \vec{x} \leq \vec{b}$, де

$$A = []; \quad \vec{b} = []. \quad (8)$$

Оскільки всі координати вектора $\vec{x}$ є булевими (1), то

$$\vec{lb} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{та} \quad \vec{ub} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \quad (9)$$

III. Експериментальне дослідження задачі інтелектуальної маршрутизації за метрикою OSPF

Генерація навчальних та тестових даних

Експериментальне дослідження задачі інтелектуальної маршрутизації за метрикою OSPF складалося з трьох послідовних етапів:

- генерація навчальних (10000 комбінацій) та тестових (1000 комбінацій) даних;
- навчання та тренування нейронної мережі з багатошарових перцептронів MLP;
- тестування нейронної мережі та аналіз результатів експериментів.

Генерацію навчальних та тестових даних було зроблено у середовищі MATLAB за допомогою підпрограми `intlinprog` окремо для моделей регресії та класифікації. При генерації як вихідні дані було використано випадкові вектори пропускних здатностей каналів зв'язку $c_{i,j}$. Було розглянуто три варіанти вхідних даних мережі:

- 1) «Повний» ряд стандартних пропускних здатностей 10 Мбіт/с – 100 Мбіт/с – 1 Гбіт/с – 10 Гбіт/с – 40 Гбіт/с – 100 Гбіт/с – 400 Гбіт/с.
- 2) «Максимальний» ряд (1 Гбіт/с – 400 Гбіт/с).
- 3) «Оптимальний» рекомендований ряд (1 Гбіт/с – 100 Гбіт/с).

Оскільки за замовчуванням при розрахунку метрики OSPF використовується коефіцієнт 10^8 (біт/с), то при для сучасних високошвидкісних каналів зв'язку від 1000 Мбіт/с спостерігається некоректна ситуація, коли метрика дорівнює 1 для всіх каналів зі швидкістю від 100 Мбіт/с та більше. Тому адміністраторам рекомендується на всіх маршрутизаторах мережі (area) вручну встановлювати такий коефіцієнт для обчислення метрики OSPF, що дорівнює максимальній пропускної здатності каналу зв'язку мережі (area). При генерації наборів даних для розрахунку метрик OSPF як коефіцієнт (базової пропускної здатності, bw) було використано максимальні пропускні здатності каналів (100 Гбіт/с та 400 Гбіт/с відповідно).

Як видно з топології мережі рис. 1, для одношляхової маршрутизації доступні тільки 3 шляхи:

- 1 – 2 – 5: вектор шляху $\vec{x}^t = [0 \ 1 \ 0 \ 0 \ 1 \ 1]$ – номер комбінації 1;
- 1 – 3 – 4 – 5: вектор шляху $\vec{x}^t = [1 \ 0 \ 0 \ 1 \ 0 \ 0]$ – номер комбінації 2;
- 1 – 2 – 3 – 4 – 5: вектор шляху $\vec{x}^t = [1 \ 0 \ 1 \ 0 \ 1 \ 1]$ – номер комбінації 3.

Як видно з рис. 2 та рис. 3, де наведено статистику згенерованих навчальних та тестових комбінацій, варіанти рішень оптимізаційної задачі при випадковій генерації пропускних здатностей каналів мають різну частку (28%-70%-2%), тобто імовірність появи після розрахунку різних варіантів рішень суттєво відрізняється.

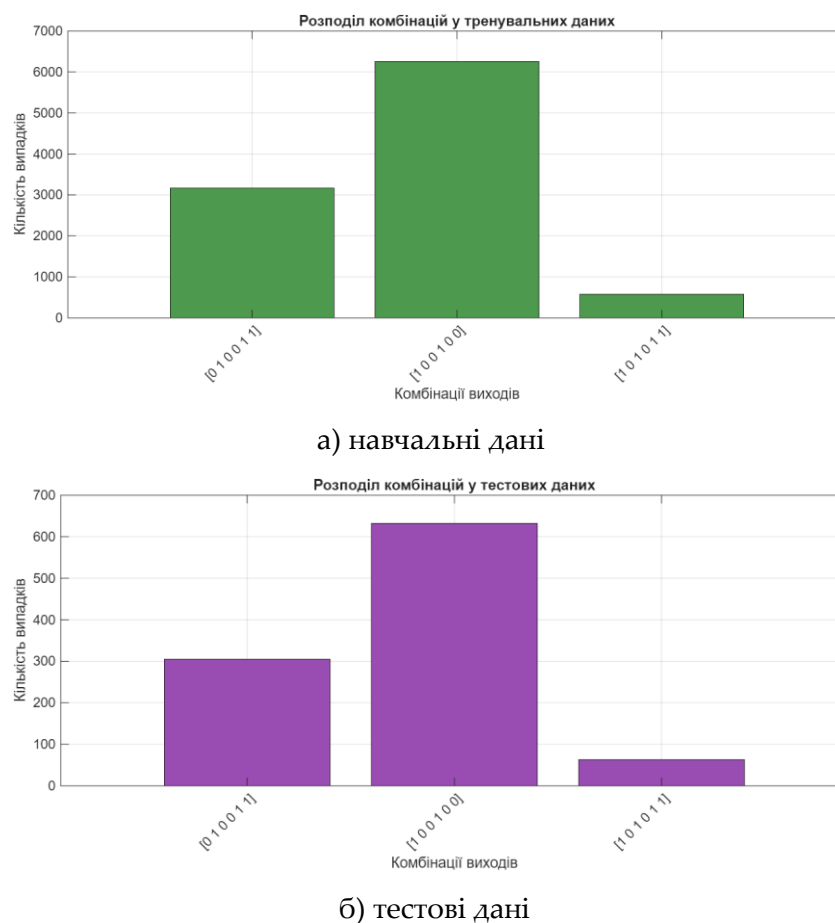


Рис. 2. Розподіл комбінацій даних для регресійної моделі

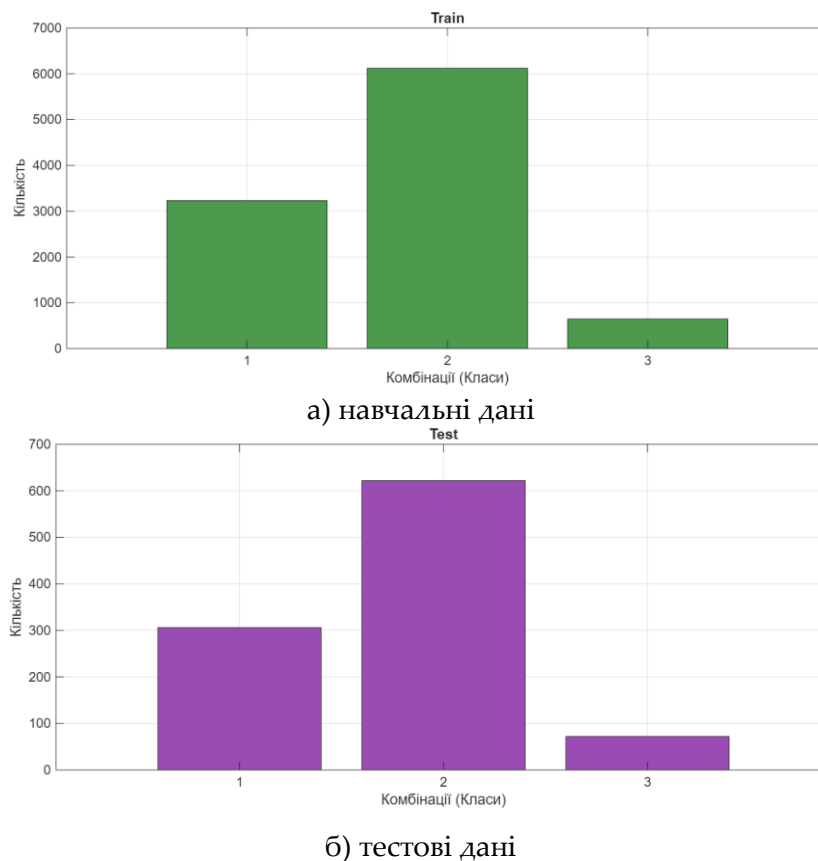
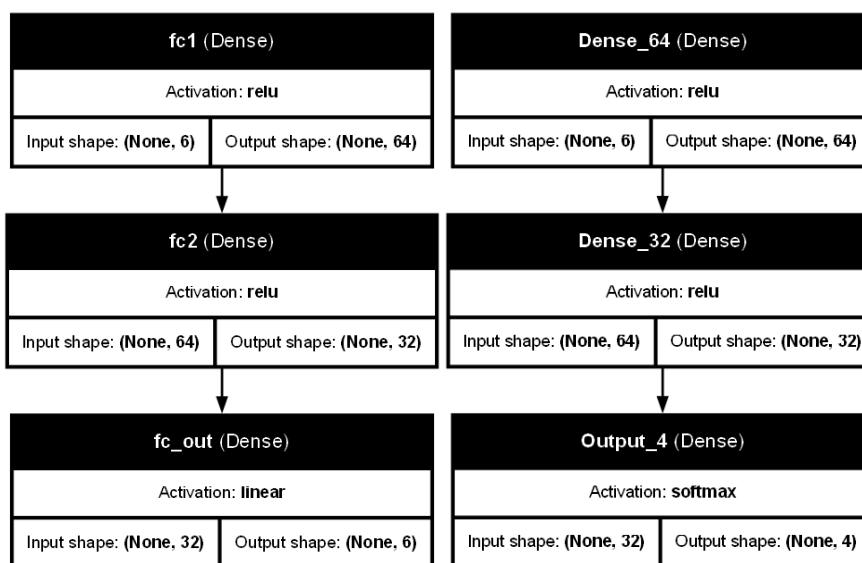


Рис. 3. Розподіл комбінацій даних для класифікаційної моделі

Для дослідження було обрано нейронну мережу MLP – багатошаровий перцептрон, що є класичною нейронною мережею прямого поширення. Мережа працює з векторами і таблицями та припускає, що кожен вхідний приклад є незалежним. Архітектура складається з послідовних повнозв'язних (Dense) шарів, де кожен нейрон одного шару з'єднаний з усіма нейронами наступного. За умови зміни вхідних даних MLP доведеться перенавчати. MLP представлена у середовищі MATLAB через Deep Learning Toolbox та вбудовані безпосередньо в базові класи `tf.keras.layers.Dense` (TensorFlow) або `torch.nn.Linear` (PyTorch). На рис. 4 наведено структури MLP, що використовувалися при моделюванні.

Як видно з рис. 4, структури нейронної мережі обох моделей прості та відрізняються тільки вихідним шаром: якщо класифікатор обирає одну з 3 категорій (4 категорія відсутня у навчальних та тестових наборах), то регресія намагається точно вирахувати 6 числових значень (вектор шляху). Вони складаються з:

- вхідного шару: 6 нейронів (пропускна здатність каналів);
- шару 1 (Dense): 64 нейрони + ReLU;
- шару 2 (Dense): 32 нейрони + ReLU;
- вихідного шару: 6 нейронів із функцією активації linear/ 4 нейрона із функцією softmax (ймовірності класів).



а) модель регресії

б) модель класифікації

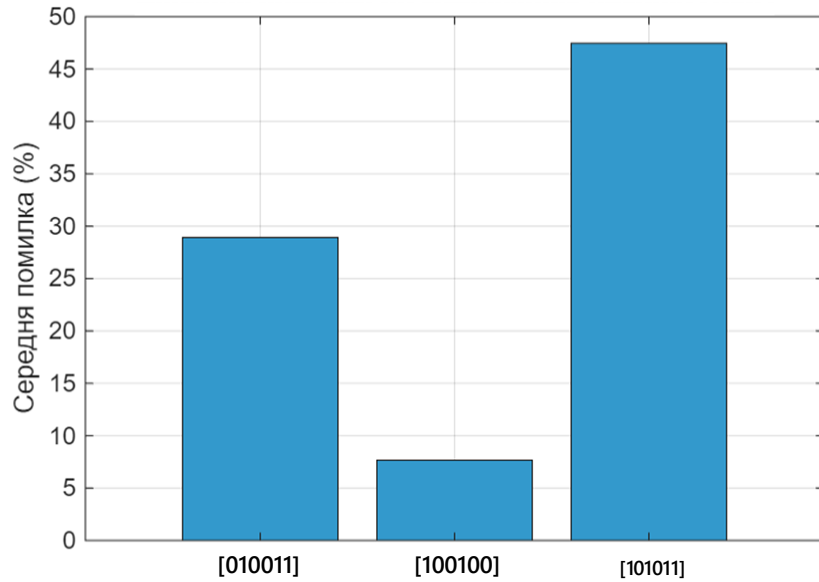
Рис. 4. Структури нейронної мережі

Тестування регресійної моделі нейронної мережі при розв'язанні задачі маршрутизації за метрикою OSPF

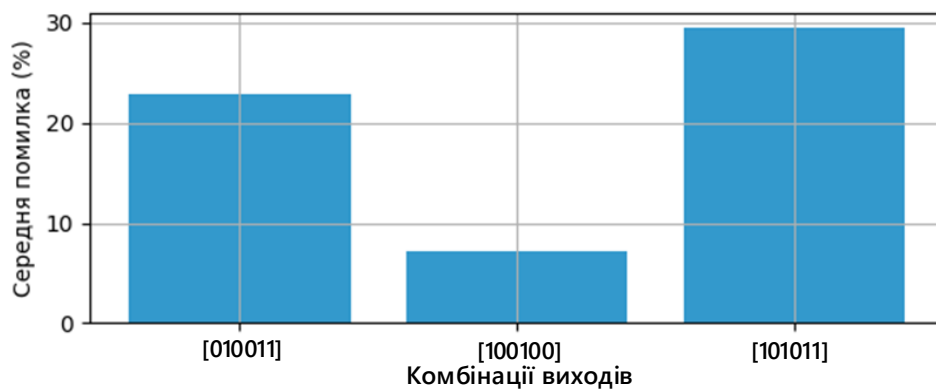
На рис. 5 – рис. 10 наведено діаграми за результатами тестування регресійної моделі нейронної мережі при розв'язанні задачі маршрутизації за метрикою OSPF з використанням трьох варіантів вхідних даних: точність прогнозу та середня помилка по комбінаціях виходів. Діаграма «Середня помилка по комбінаціях виходів» (рис. 5 – рис. 7) характеризує ступінь невизначеності моделі – це середнє арифметичне абсолютних відхилень прогнозних значень мережі від цільових значень (0 або 1) у межах певної групи тестових векторів. Вона відображає середню відстань між неперервним виходом нейромережі та бінарним еталоном до моменту округлення та характеризує ступінь невизначеності моделі. Якщо середня помилка наближається до 50%, це свідчить про те, що для даної комбінації модель генерує значення на межі прийняття рішення, що призводить до високої ймовірності помилкового округлення. В залежності від середньої помилки прогнозний стан моделі можна класифікувати як:

- зона високої точності (0% – 10%): модель «впевнена», значення на виході близькі до цілих чисел (наприклад, 0,05 або 0,92), ймовірність помилки буде близькою до нуля;
- зона допустимих відхилень (10% – 25%): модель має певні сумніви, але все ще тримається правильного «боку» від порогу 0,5, точність залишається високою, але з'являються поодинокі випадки помилкового округлення;
- зона невизначеності (25% – 40%): «слабке передбачення», виходи часто потрапляють у діапазон 0,3–0,7, ймовірність помилки різко зростає;

– критична зона (40% – 50% і вище): модель видає значення, близькі до 0,5, або взагалі помилкові значення (видає 1 там, де має бути 0), модель не справляється з цією комбінацією.

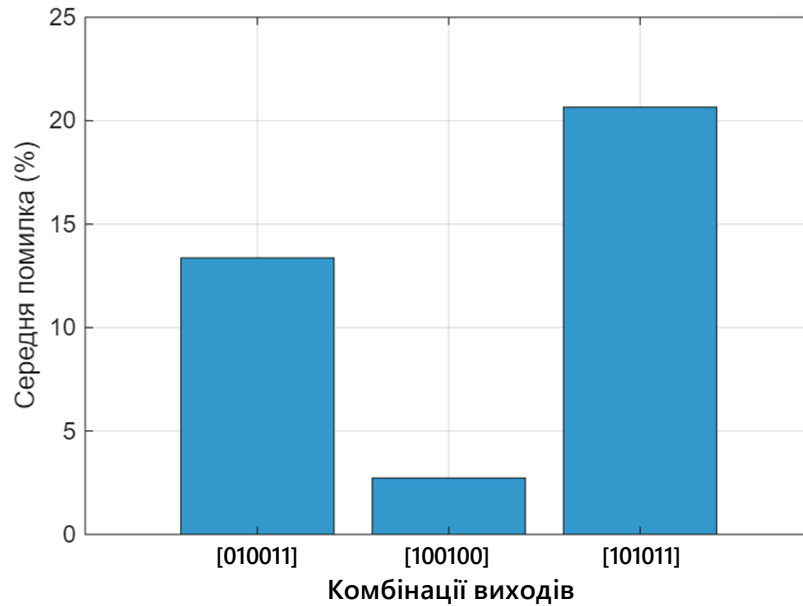


а) неймережа MATLAB

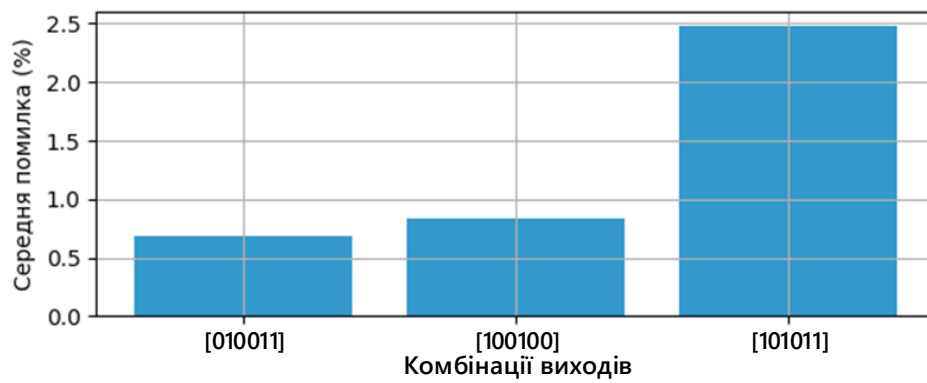


б) неймережа Python

Рис. 5. Середня помилка по комбінаціях регресійної моделі (10 Мбіт/с – 400 Гбіт/с)

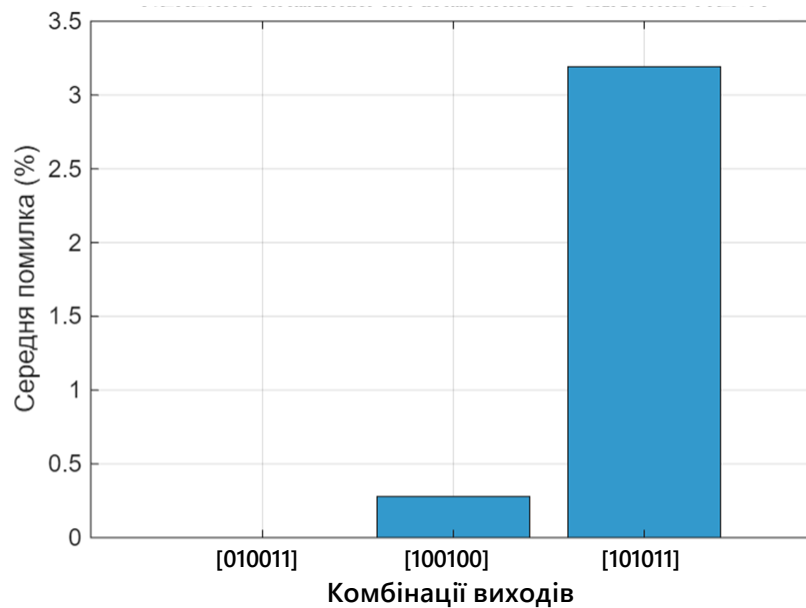


а) нейромережа MATLAB



б) нейромережа Python

Рис. 6. Середня помилка по комбінаціях регресійної моделі (1 Гбіт/с – 400 Гбіт/с)



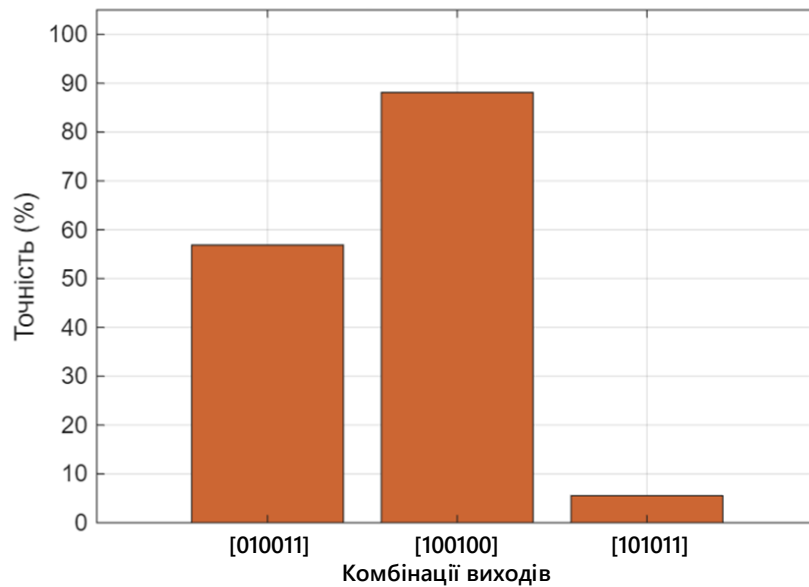
а) нейромережа MATLAB



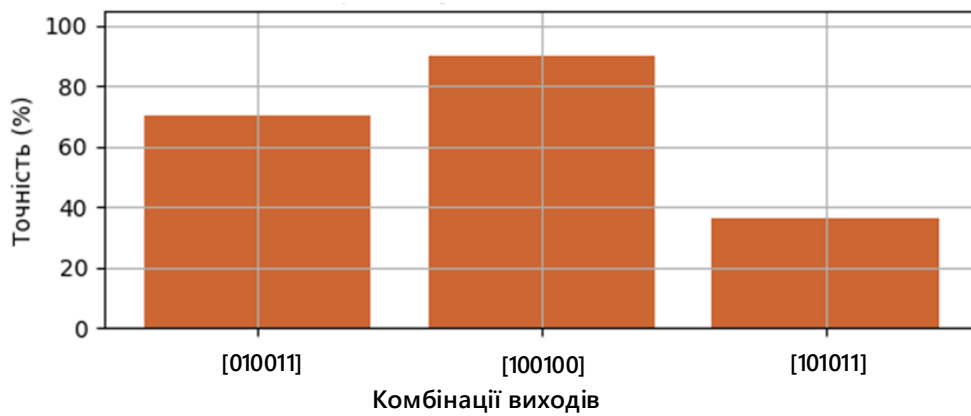
б) нейромережа Python

Рис. 7. Середня помилка по комбінаціях регресійної моделі (1 Гбіт/с – 100 Гбіт/с)

Діаграма «Точність прогнозу по комбінаціях виходів» (рис. 8 – рис. 10) показує, наскільки надійно нейронна мережа відтворює конкретні топологічні рішення, тобто це точність відтворення вектору шляху з 6 каналів зв'язку (обернена величина до ймовірності помилки прогнозу). Помилка лише одного каналу зв'язку призводить до суттєвого зниження загальної точності відтворення шляху, оскільки вектор шляху втрачає узгодженість із реальною топологією. При цьому навіть незначні відхилення у прогнозі можуть спричинити вибір неефективного шляху, що збільшує затримку та знижує пропускну здатність мережі.

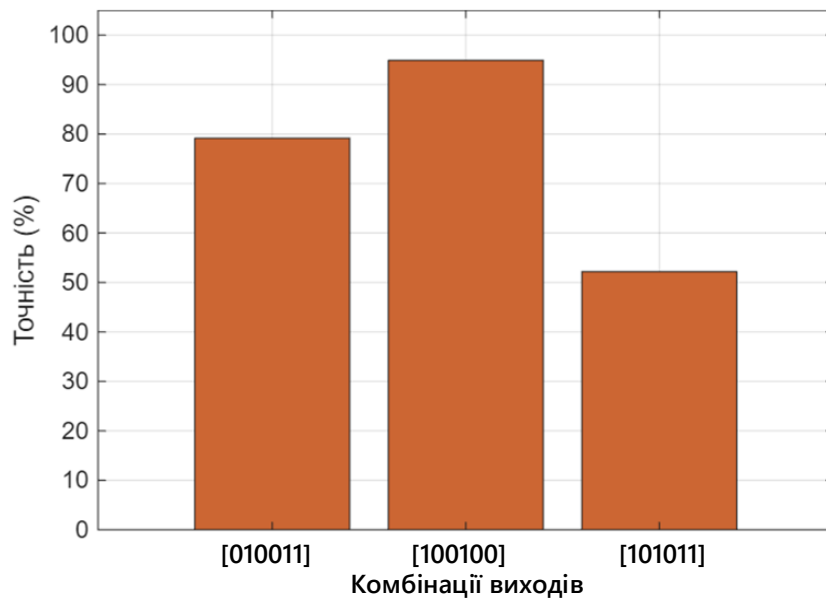


а) нейромережа MATLAB

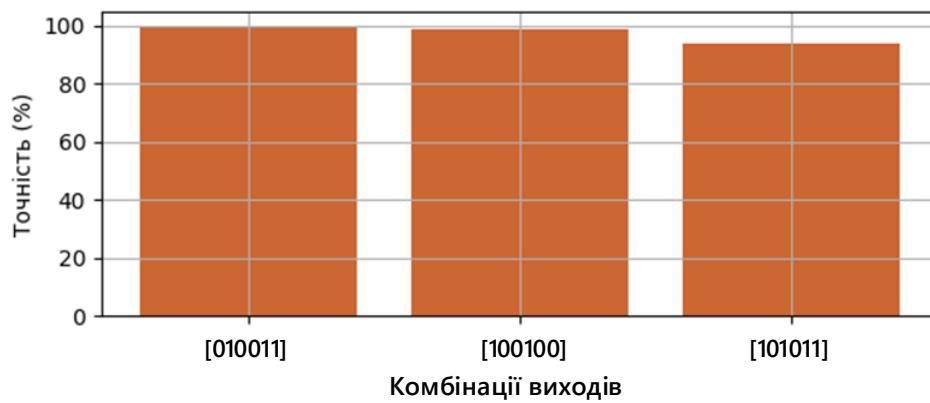


б) нейромережа Python

Рис. 8. Точність прогнозу по комбінаціях регресійної моделі (10 Мбіт/с – 400 Гбіт/с)

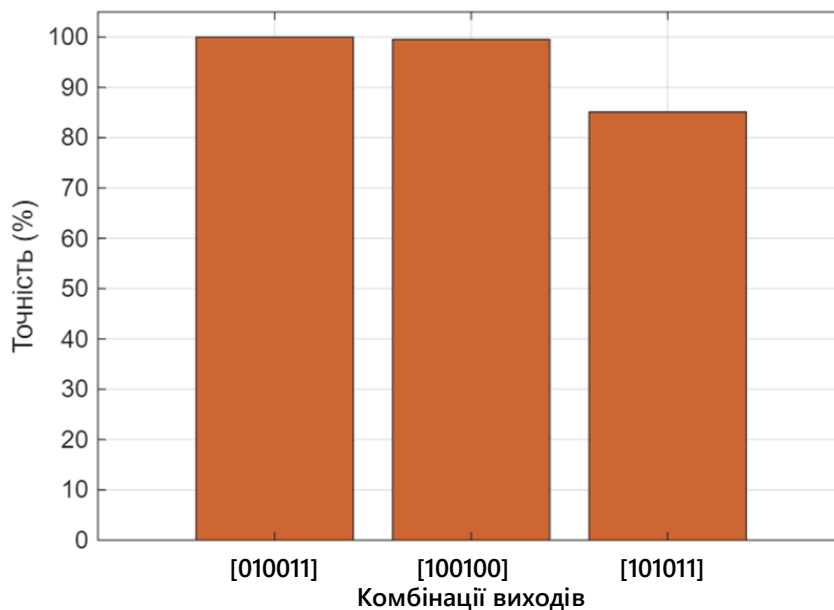


а) неймережа MATLAB

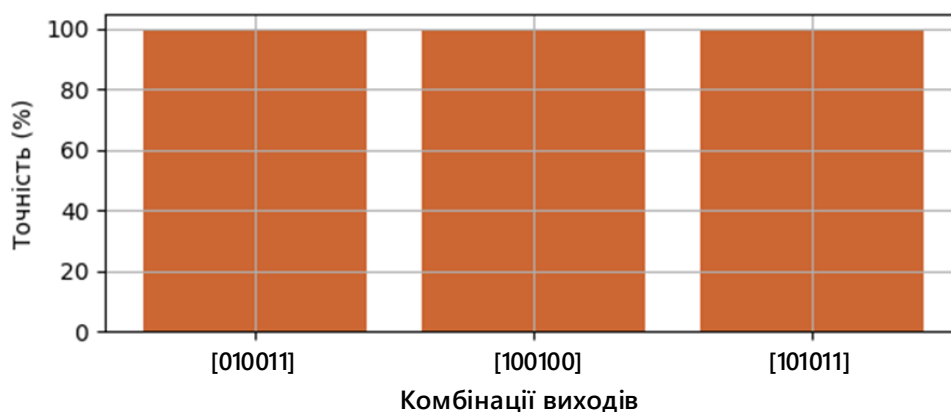


б) неймережа Python

Рис. 9. Точність прогнозу по комбінаціях регресійної моделі (1 Гбіт/с – 400 Гбіт/с)



а) неймережа Matlab



б) неймережа Python

Рис. 10. Точність прогнозу по комбінаціях регресійної моделі (1 Гбіт/с – 100 Гбіт/с)

Як видно з рис. 5 – рис. 10, регресійна модель загалом спроможна розв’язувати задачу одношляхової маршрутизації, але результат сильно залежить від рівномірності комбінацій вхідних даних: найнижча точність прогнозу (найбільша ймовірність помилки) спостерігалася для найменш ймовірної комбінації у наборі вхідних даних.

Також спостерігається явна залежність точності прогнозу від набору навчальних та тестових даних (ряду ПЗ каналів зв’язку): найкращі результати було досягнуто при використанні «оптимального» ряду (1 Гбіт/с – 100 Гбіт/с), коли пропускні здатності каналів відрізняються не більш, ніж на 2 порядки. У такому випадку досягається висока точність прогнозів (до 100%), модель «впевнена» та знаходиться у зоні високої точності. При використанні «максимального» ряду (1 Гбіт/с – 400 Гбіт/с) спостерігається явна перевага нейронної мережі Python (точність прогнозів зменшується до 98-99%) над нейронною мережею MATLAB (точність прогнозів до 90%) – це також підкреслюється

тим, що нейронна мережа Python залишається у зоні високої точності, а нейронна мережа MATLAB за деякими комбінаціями працює у зоні допустимих відхилень. Найгірша ситуація спостерігається при використанні «повного» ряду пропускних здатностей, коли точність прогнозів падає до 72-84%. Це відбувається тому, що нейронна мережа Python за деякими комбінаціями працює у зонах допустимих відхилень та невідзначеності, а нейронна мережа MATLAB – за однією найменш ймовірною комбінацією взагалі у критичній зоні, тобто, мережа не справляється з цією комбінацією.

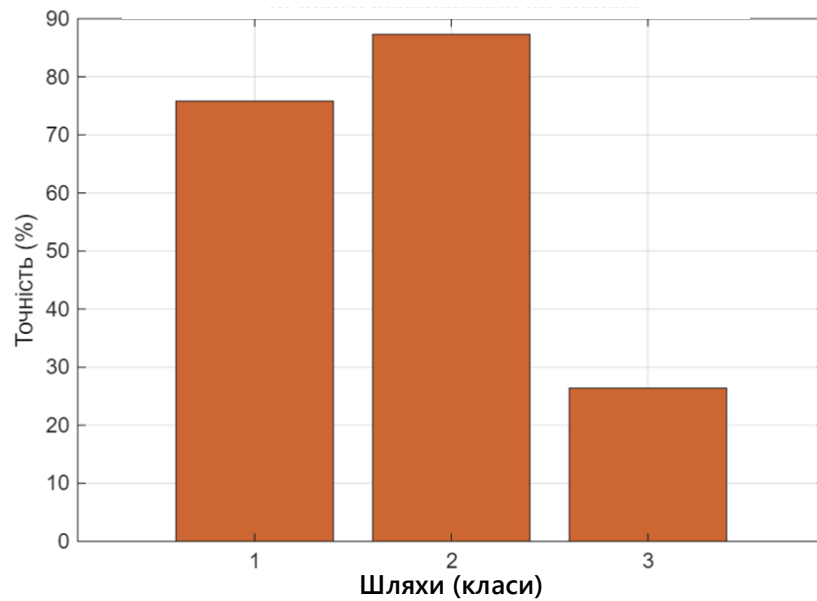
Результати, що відображають точність пошуку вектора шляху регресійними моделями, наведені у табл. 1, за якою чітко видно перевагу нейронної мережі Python над нейронною мережею MATLAB.

Таблиця 1. Результати моделювання регресії (точність пошуку вектору шляху)

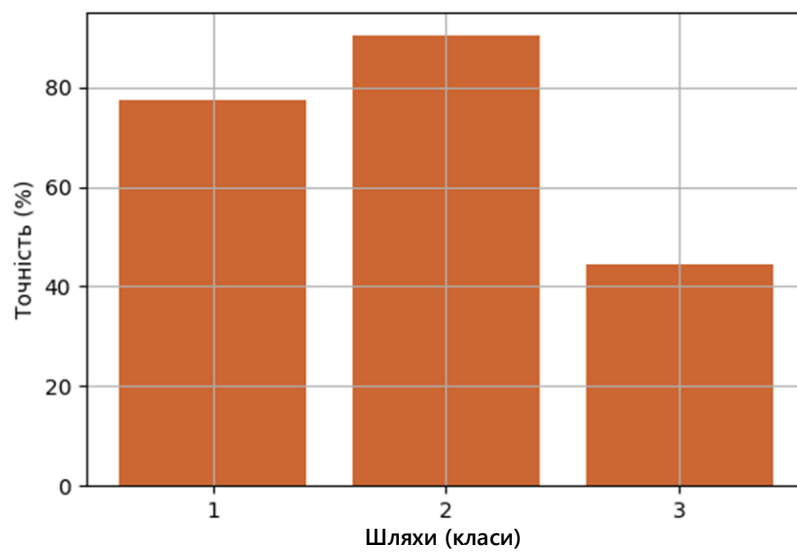
Вхідні дані (ПЗ каналів)	Нейронна мережа		
	MATLAB	Python TensorFlow	Python PyTorch
10 Мбіт/с – 400 Гбіт/с	72,6% -75,5%	79,7%-81,9%	79,3%-83,8%
1 Гбіт/с – 400 Гбіт/с	88,4% -89,2%	97,4%-99,2%	98%-98,6%
1 Гбіт/с – 100 Гбіт/с	98,7%-99,5%	100%	100%

Тестування класифікаційної моделі нейронної мережі при розв'язанні задачі маршрутизації за метрикою OSPF

На рис. 11 – рис. 16 наведено діаграми за результатами тестування класифікаційної моделі нейронної мережі при розв'язанні задачі маршрутизації за метрикою OSPF з використанням трьох варіантів вхідних даних: точність класифікації та ймовірність помилки по класах. На відміну від моделі регресії, де мережа вгадувала та округляла числа, у моделі класифікації мережа обирає конкретний тип (номер) шляху. Теоретично, це фундаментальна зміна, яка значно краще підходить для завдань маршрутизації, тому що класифікатор завжди обирає дискретні рішення із скінченного набору варіантів (топологій). Вихідний шар softmax (рис. 4 б) перетворює виходи в ймовірності, що дозволяє відсікати сумнівні рішення. Замість того, щоб мінімізувати відстань (MSE – середній квадрат різниці), як у моделі регресії, мережа максимізує ймовірність правильного вибору, що виглядає логічніше для маршрутизації. Діаграма «Точність класифікації по класах» (рис. 11 – рис. 13) показує, наскільки точно нейронна мережа ідентифікує кожен конкретний сценарій маршрутизації (топологію) залежно від вхідних параметрів мережі. Вона дозволяє визначити, які саме комбінації каналів модель вивчила досконало, а які викликають у неї труднощі через схожість вхідних даних або через недостатню кількість прикладів у навчальній вибірці.

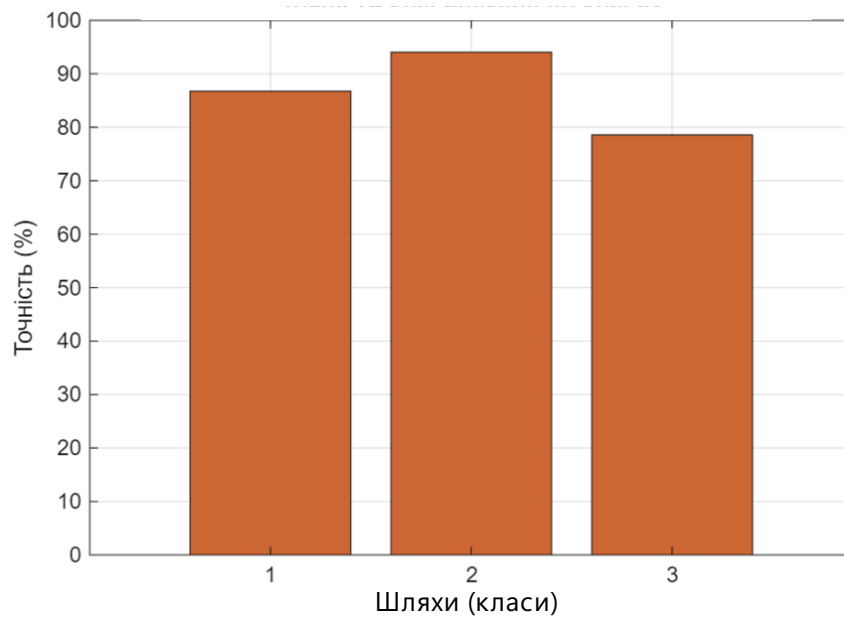


а) неймерережа MATLAB

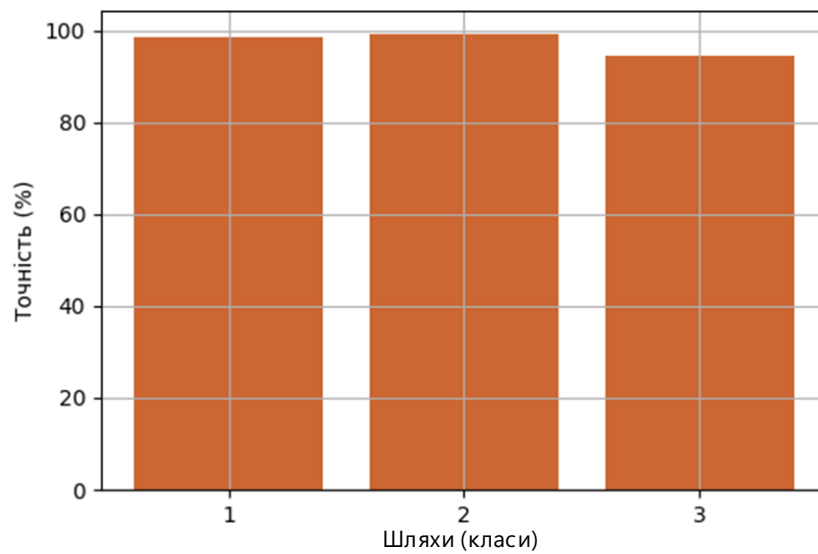


б) неймерережа Python

Рис. 11. Точність класифікації по класах класифікаційної моделі (10 Мбіт/с – 400 Гбіт/с)

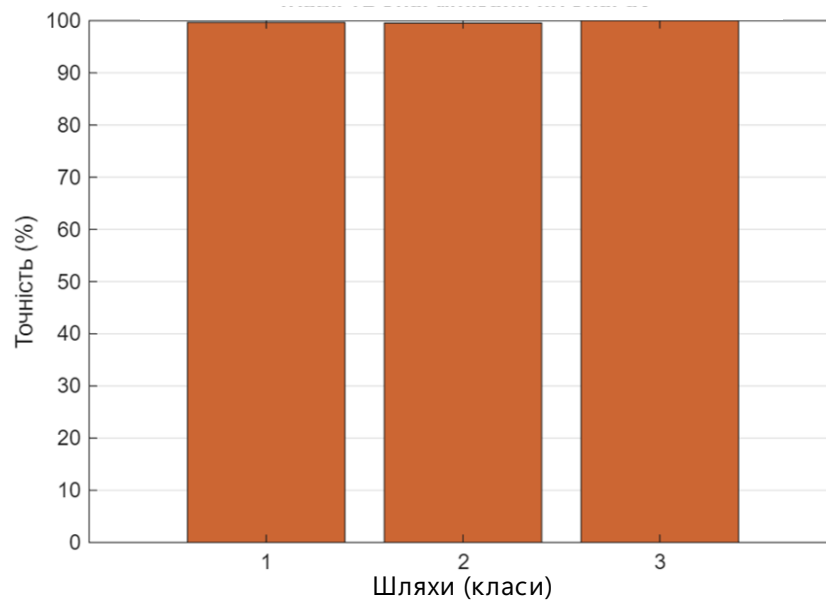


а) неймережа MATLAB

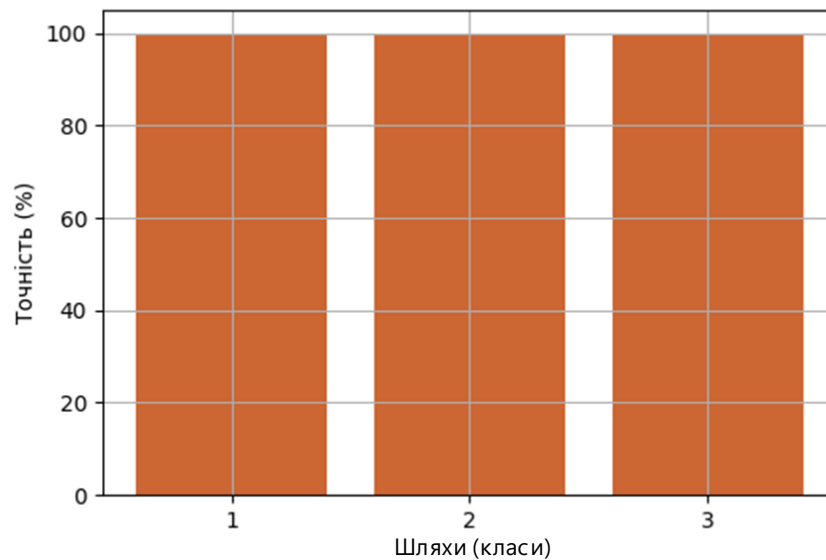


б) неймережа Python

Рис. 12. Точність класифікації по класах класифікаційної моделі (1 Гбіт/с – 400 Гбіт/с)



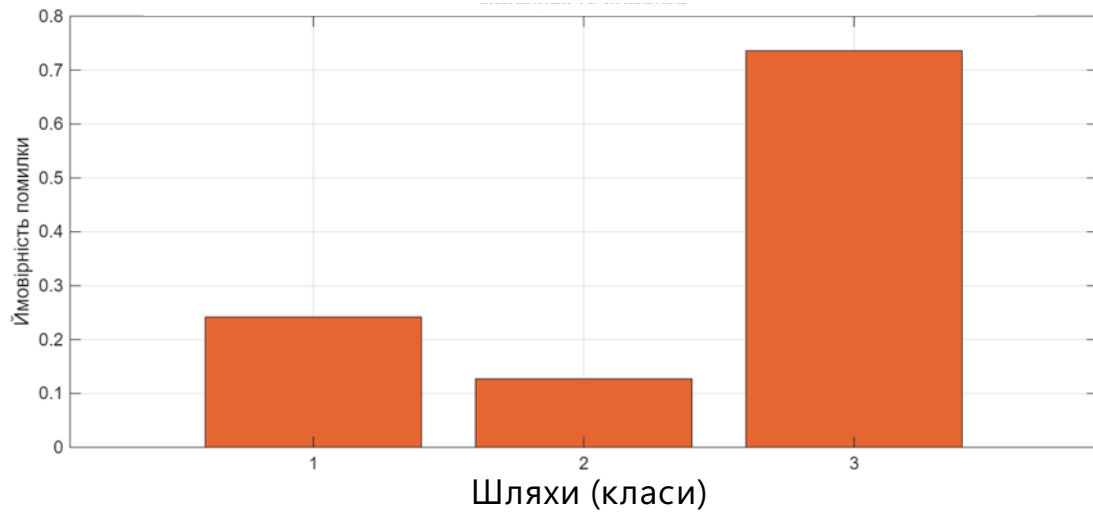
а) нейромережа MATLAB



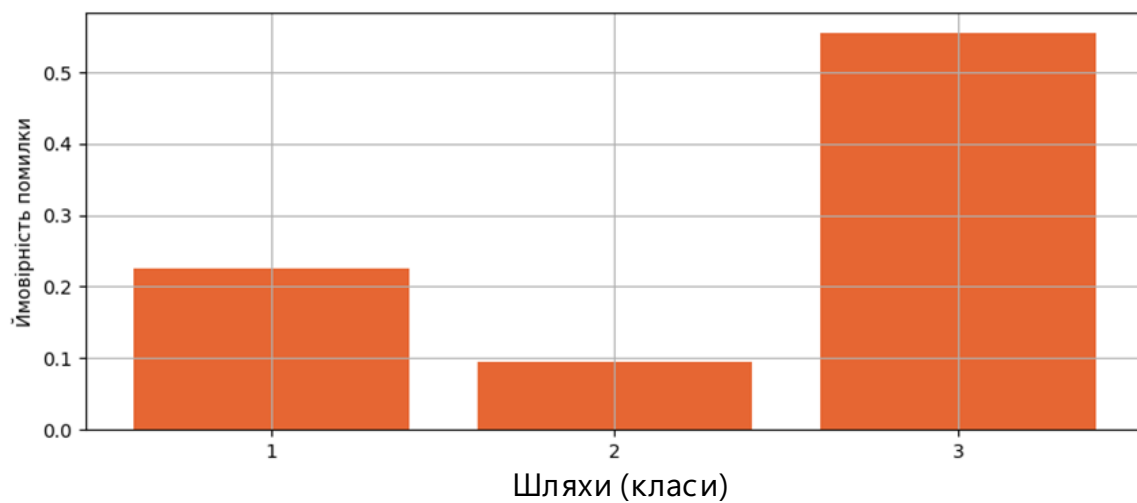
б) нейромережа Python

Рис. 13. Точність класифікації по класах класифікаційної моделі (1 Гбіт/с – 100 Гбіт/с)

Діаграма «Ймовірність помилки» (рис. 14 – рис. 16) по суті є оберненою до діаграми точності та показує ймовірність, з якою мережа обрала інший (неправильний) шлях.

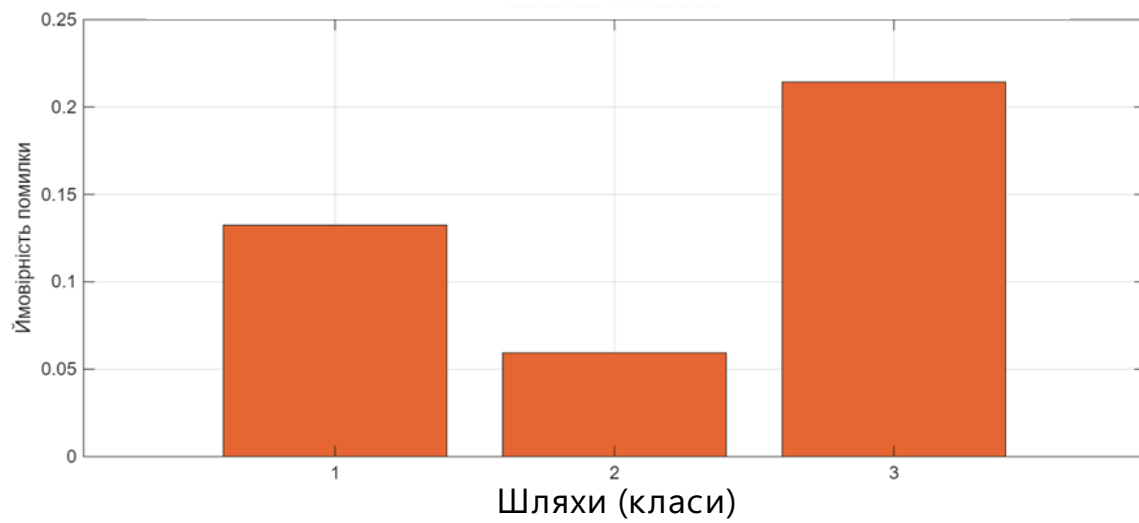


а) нейромережа MATLAB

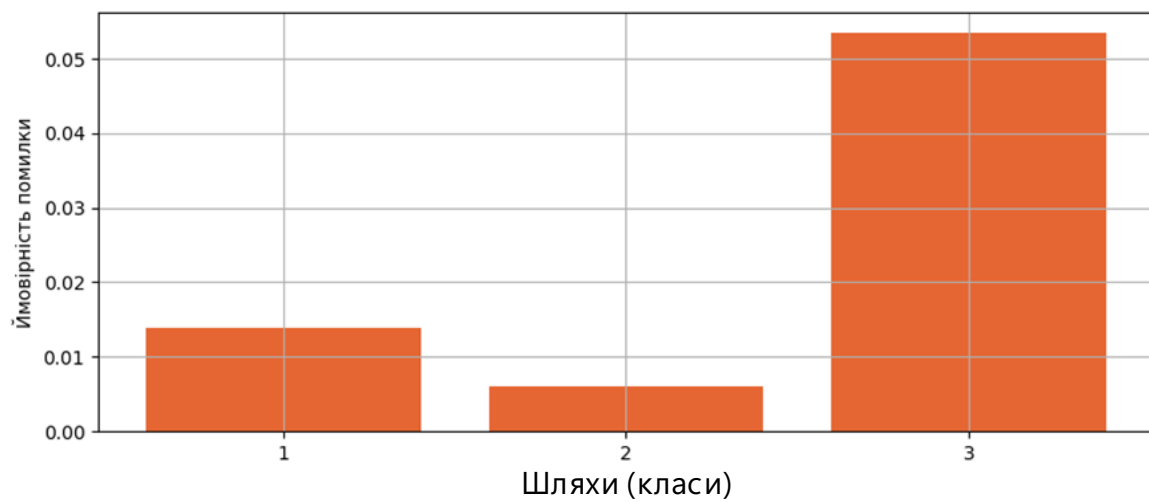


б) нейромережа Python

Рис. 14. Ймовірність помилки по класах класифікаційної моделі (10 Мбіт/с – 400 Гбіт/с)

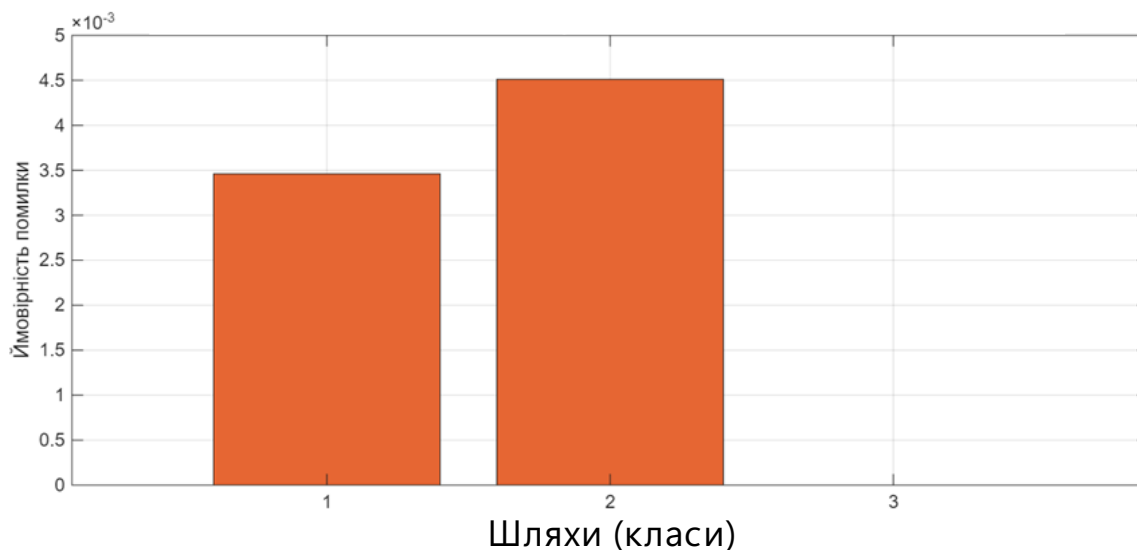


а) нейромережа MATLAB

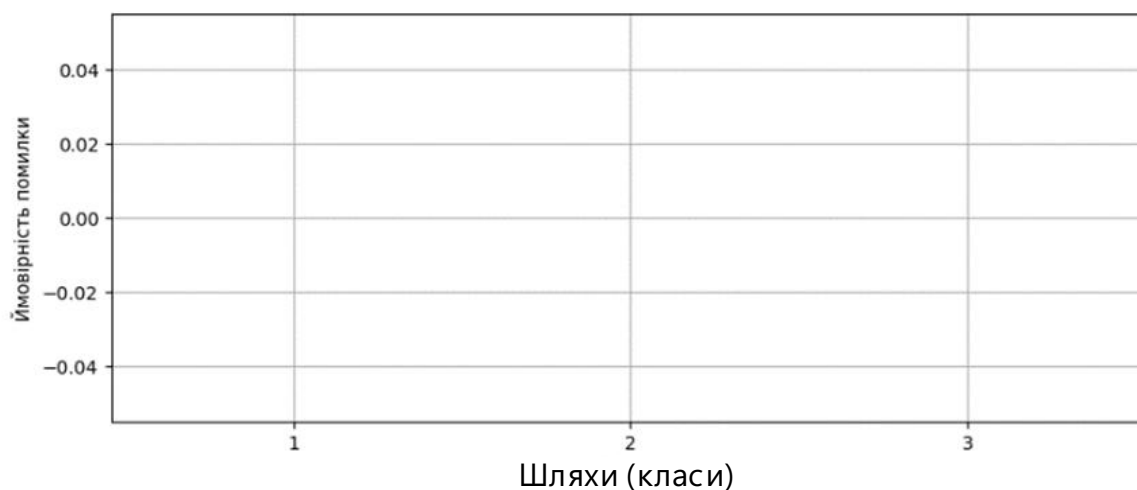


б) нейромережа Python

Рис. 15. Ймовірність помилки по класах класифікаційної моделі (1 Гбіт/с – 400 Гбіт/с)



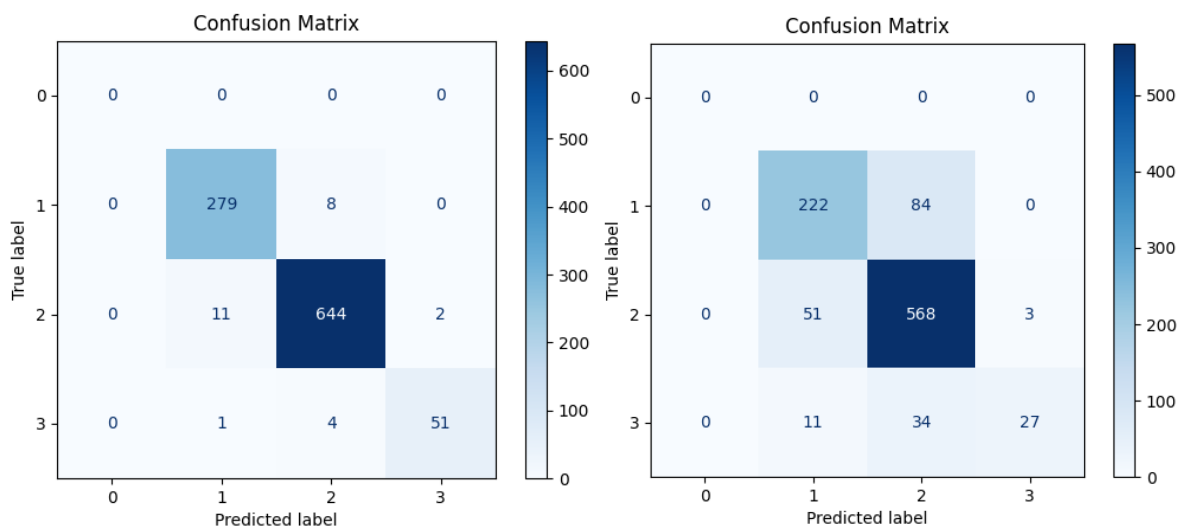
а) нейромережа MATLAB



б) нейромережа Python

Рис. 16. Ймовірність помилки по класах класифікаційної моделі (1 Гбіт/с – 100 Гбіт/с)

Як видно з рис. 11 – рис. 16, класифікаційна модель показує дещо більш високу точність розв'язання задачі одношляхової маршрутизації, ніж регресійна модель, але все ж таки зберігаються ті ж самі залежності результатів від рівномірності комбінацій вхідних даних та від набору навчальних та тестових даних (ряду ПЗ каналів зв'язку). До того ж видно більш ефективну роботу нейронної мережі Python (незалежно від застосованих бібліотек). На рис. 17 наведені матриці невідповідностей (Confusion Matrix), що є ключовим інструментом для оцінки якості моделі класифікації маршрутів. Матриця показує, наскільки часто модель прогнозує правильний клас (шлях) і де саме вона припускається помилок. Так, головна діагональ (зліва направо) – це кількість правильних передбачень для класів 1, 2 та 3 відповідно (прикладів класу 0 немає).



а) 1 Гбіт/с – 400 Гбіт/с

б) 10 Мбіт/с – 400 Гбіт/с

Рис. 17. Матриці невідповідностей класифікаційної моделі нейромережі Python

Чим вищі ці значення і чим темніший колір клітинок, тим краще працює модель. З рис. 17 а (ПЗ каналів 1 Гбіт/с – 400 Гбіт/с) можна зробити висновок, що клас 2 найкраще прогнозується, тому що має найбільшу кількість правильних прогнозів (644) та найтемніший колір – модель дуже добре розпізнає цей сценарій. Клас 1 також демонструє високу точність (279), але має "плутанину" з класом 2 (8). Клас 3 – найменш чисельний клас у вибірці (всього 56 прикладів: $51 + 4 + 1$), але попри малу кількість, точність досить висока ($\approx 91\%$). Помилки (елементи поза головної діагоналі) показують, які класи є "схожими" з точки зору нейромережі:

- клас 1 та клас 2: 8 разів модель помилково обрала клас 2 замість класу 1 та 11 разів модель обрала клас 1 замість класу 2 – ці два маршрути мають дуже схожі вхідні параметри (пропускну здатність каналів), що змусило модель коливатися між ними.
- клас 2 та клас 3: 2 рази модель сплутала клас 2 з класом 3 та 4 рази – клас 3 з класом 2.
- клас 3 та клас 1: лише 1 випадок помилки, що означає, що ці маршрути мають суттєві відмінності і мережа їх майже не плутає.

Оскільки загальна кількість тестів 1000, а кількість правильних прогнозів 974, то загальна точність склала 97,4%, тобто модель демонструє високу ефективність, але існує дисбаланс класів, що змушує модель обирати клас 2 у сумнівних ситуаціях.

З рис. 17 б (ПЗ каналів 10 Мбіт/с – 400 Гбіт/с) видно, що цей варіант моделі демонструє значно вищий рівень «плутанини» між класами. Хоча модель все ще зберігає працездатність, її надійність помітно знизилася (точність 81,7%).

Щоб уникнути дисбалансу класів було проведено навчання нейронної мережі класифікаційної моделі збалансованими даними. Це відбувалося доповненням тренувальних даних найменш ймовірних комбінацій дублями – oversampling та з викорис-

танням аналізу ваги (частоти появи) комбінації – під час навчання loss-функція масштабується за вагами класів, тобто рідкісні класи мають більшу вагу у втраті, щоб мережа "звертала більше уваги" на них.

На рис. 18 – рис. 20 наведено діаграми ймовірності помилки по класах за результатами тестування класифікаційної моделі нейронної мережі при виконанні задачі маршрутизації за метрикою OSPF з використанням трьох варіантів збалансованих вхідних даних. Але, як можна побачити з рис. 18 – рис. 20, хоча точність прогнозу найбільш рідкісних комбінацій зростає (ймовірність помилок знизилася), загальна точність значно погіршилася.

Це можна пояснити тим, що навчали модель на збалансованих даних, але тестували її на незбалансованих (природних) даних. У результаті це призвело до розриву між тим, що модель очікує, і тим, що вона бачить. Оригінальні незбалансовані дані містили інформацію про частоту появи комбінацій і ця частота є важливою ознакою, яку модель використовує підсвідомо (якщо вона знає, що комбінація з'являється у 80 % випадків, це допомагає їй у прогнозуванні), тобто при тренуванні збалансованими даними втрачається найважливіша статистична інформація: пріоритетність домінуючого класу, що змушує модель ігнорувати реальну ймовірність класів.

Результати, що відображають точність пошуку шляху класифікаційними моделями, наведені у табл. 2, з якою чітко видно перевагу нейронної мережі Python над нейронною мережею MATLAB та погіршення точності прогнозування при використанні при навчанні збалансованих вхідних даних.

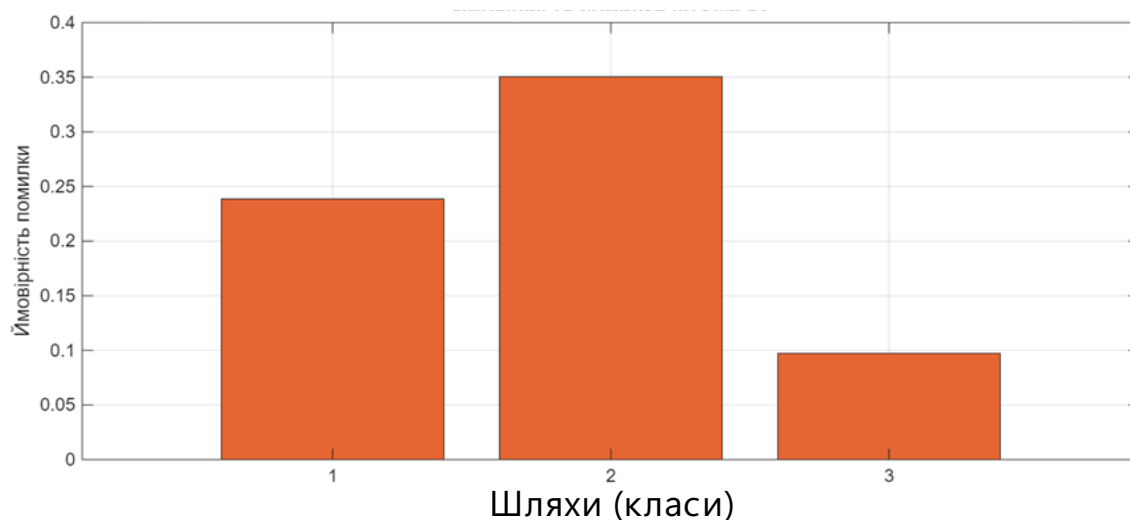


Рис. 18. Ймовірність помилки по класах класифікаційної моделі (10 Мбіт/с – 400 Гбіт/с) нейромережі Python із збалансованими вхідними даними

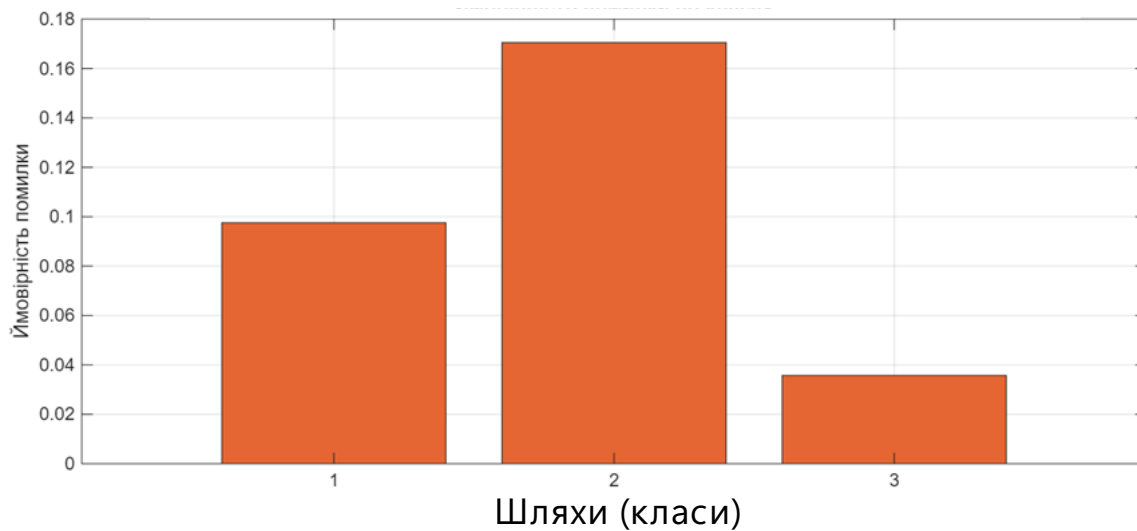


Рис. 19. Ймовірність помилки по класах класифікаційної моделі (1 Гбіт/с – 400 Гбіт/с) нейромережі Python із збалансованими вхідними даними

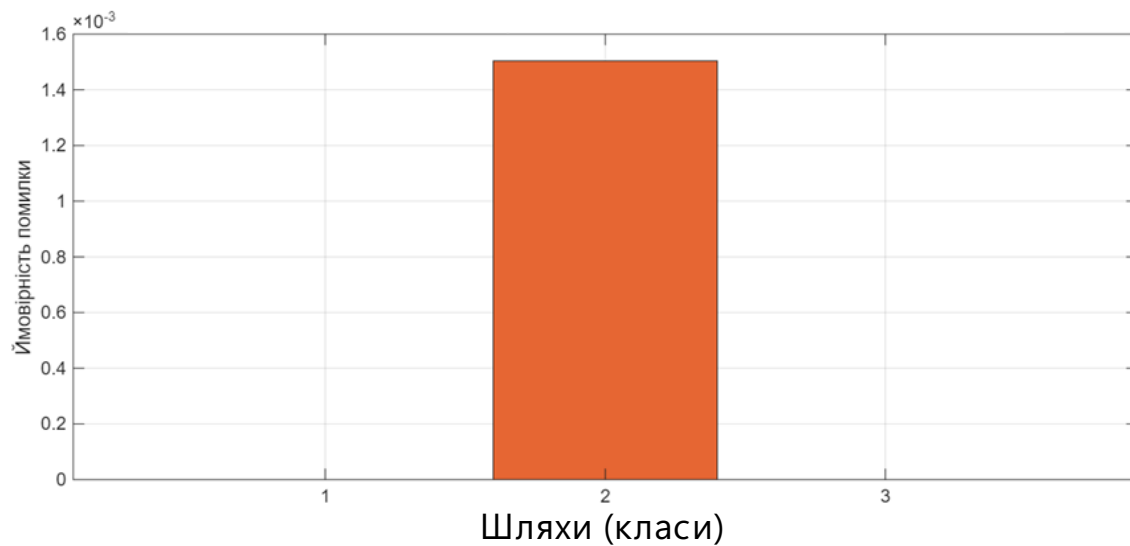


Рис. 20. Ймовірність помилки по класах класифікаційної моделі (1 Гбіт/с – 100 Гбіт/с) нейромережі Python із збалансованими вхідними даними

Таблиця 2. Результати моделювання класифікації (точність пошуку номеру шляху)

Вихідні дані (ПЗ каналів)	Нейронна мережа				
	MATLAB	MATLAB збалансовані дані	Python PyTorch	Python TensorFlow	Python TensorFlow з вагами класів
10 Мбіт/с – 400 Гбіт/с	76,9%-78,2%	66,1%-69,5%	81,9%-83,5%	81%-83,4%	75,9%-78,4%
1 Гбіт/с – 400 Гбіт/с	89,9%-93,1%	84,9%-92,0%	92,5%-98,7%	96,7%-99%	96%-99%
1 Гбіт/с – 100 Гбіт/с	99,3%-100%	99,7%-100%	100%	100%	100%

Для підвищення надійності (точності прогнозування) класифікаційної моделі нейронної мережі було ускладнено її структуру (рис. 21). Як видно з рис. 21, до структури нейронної мережі було додано перший шар зі 128 нейронами та додано метод регуляризації Dropout, який запобігає перенавчанню моделі, що потенційно повинно було підвищити стабільність та точність прогнозування. Але, як показали подальші експерименти з «найгіршим» варіантом вхідних даних (ПЗ каналів 10 Мбіт/с – 400 Гбіт/с), ускладнення структури мережі не призвело до підвищення точності. Це можна пояснити тим, що модель MLP вичерпала статистичний ресурс аналізу вхідних даних (6 параметрів) без урахування топології мережі.

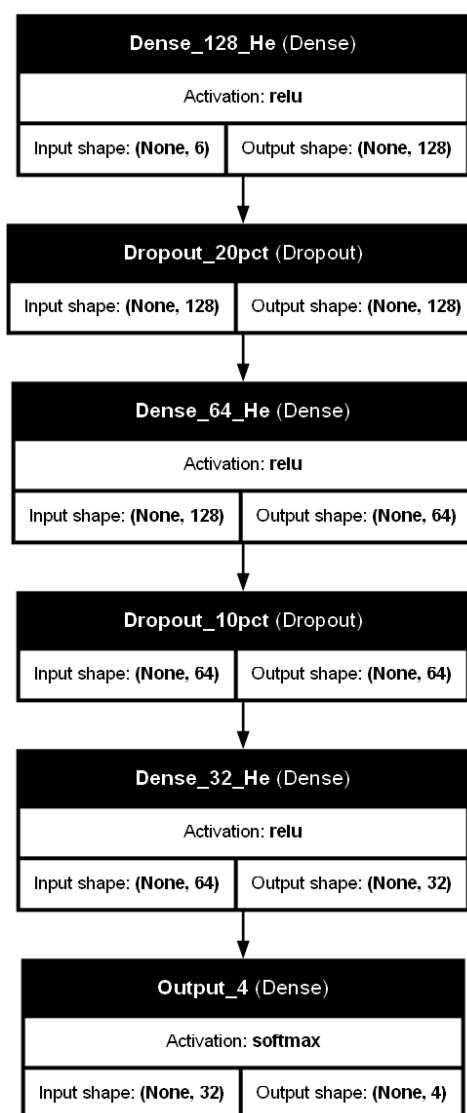


Рис. 21. Структура ускладненої нейронної мережі моделі класифікації

Для підвищення точності прогнозування потрібно або додаткові вхідні параметри, або перейти до використання графової нейронної мережі GNN.

Висновки

Як показав проведений аналіз, застосування технологій штучних нейронних мереж дає змогу перетворити OSPF із пасивного протоколу на активну систему управління трафіком. Архітектура багатошарового персептрона і надалі виступає фундаментальним та перевіреним засобом прогнозування затримок і оптимізації вартості маршрутів завдяки простоті реалізації та високій точності. Паралельно, поява графових нейронних мереж відкриває перспективи побудови універсальних моделей маршрутизації, здатних оперативного узагальнювати знання для нових, раніше невідомих топологій, що має вирішальне значення для розвитку інфраструктури 5G та IoT.

Експерименти проведені на мережній топології (рис. 1), яка включала п'ять вузлів та шість каналів зв'язку, що створювали три шляхи між вузлом-відправником пакетів (вузол 1) та вузлом-одержувачем (вузол 5). Розглядалося три варіанти вихідних даних щодо топології мережі (стандартних пропускних здатностей каналів зв'язку) та дві моделі (регресійна та класифікаційна) нейронної мережі, які впливали на точність вибору (прогнозування) оптимального шляху за метрикою OSPF.

Результати експериментального дослідження підтвердили ефективність інтелектуальної моделі одношляхової маршрутизації на основі метрик OSPF. Показано, що точність прогнозування нейронної мережі MLP залежить від того, наскільки сильно відрізняються між собою пропускні здатності каналів зв'язку. Найкращий результат пошуку вектору шляху регресійною моделлю (99% - 100%) досягається при «оптимальному» ряді пропускних здатностей, що відрізняються не більш, ніж на два порядки (1 Гбіт/с – 100 Гбіт/с), найгірший (73% – 84%) – при «повному» ряді пропускних здатностей (10 Мбіт/с – 400 Гбіт/с). Класифікаційна модель нейронної мережі загалом показала більш високі результати пошуку шляху, але зберігається залежність точності пошуку шляху від розбіжності навчальних даних.

Продемонстровано залежність точності прогнозування шляху від рівномірності варіантів рішень у навчальних даних. Тренування нейронної мережі нерівномірним набором навчальних даних (дисбаланс даних) призводить до зміщення моделі в бік домінуючих варіантів, коли спостерігається висока ймовірність помилки прогнозу для рідкісних комбінацій. Це особливо помітно при великій розбіжності у вихідних даних.

Перевірено навчання класифікаційної моделі нейронної мережі збалансованими даними. Проте, як можна побачити з результатів (табл. 2) та діаграм ймовірності помилок (рис. 18 – рис. 20), хоча точність найбільш рідкісних комбінацій зросла, загальна точність значно погіршилася за рахунок падіння точності домінуючих комбінацій, що вивано втратою інформації про частоту появи комбінацій. Тобто, при тренуванні збалансованими даними втрачається найважливіша статистична інформація: пріоритетність домінуючого класу, що змушує модель ігнорувати реальну ймовірність класів.

Визначено, що використання нейронних мереж Python для задач одношляхової маршрутизації дає на 5% – 10% кращий результат точності прогнозування, ніж

MATLAB. Це можна пояснити тим, що нейронні мережі Python спеціалізовані для задач мережної маршрутизації, а нейронні мережі MATLAB більш універсальні для вирішення багатьох задач.

З'ясовано, що ускладнення структури нейронної мережі MLP не призводить до покращення точності прогнозування шляху при одношляховій маршрутизації, тому що модель MLP досягає межі вичерпання статистичного ресурсу аналізу вхідних даних без урахування топології мережі. Для подолання цієї межі точності потрібно або залучати додаткові вхідні параметри, що містять інформацію про завантаженість сусідніх вузлів, або виконати перехід до використання графової нейронної мережі GNN, оскільки на відміну від MLP, графова мережа використовує матрицю суміжності як структурний фільтр, що дозволяє моделі «розуміти», як потоки в одному каналі фізично впливають на сусідні канали. Використання GNN дозволить моделі вивчити фізичні закони розподілу трафіку (наприклад, закон збереження потоку у вузлах), що є ключовим фактором для досягнення точності пошуку шляху.

Незважаючи на виявлену «межу» точності, завдяки низькій обчислювальній складності MLP залишається актуальною архітектурою для локальних рішень, вбудованих систем та маршрутизаторів з обмеженими ресурсами, де критичною є швидкість прийняття рішення (мікросекунди). MLP ефективно показала себе як швидка заміна лінійного програмування у невеликих сегментах мережі з фіксованою структурою з невеликою розбіжністю параметрів каналів зв'язку. Проте GNN наразі вважається найбільш перспективним напрямком у телекомунікаціях через свою "графову природу", яка ідентична структурі мереж [2, 7, 12].

Список літератури

1. Заблоцький, С., Пограничний, В., Тарасенко, А., Колодій, Р. (2025), "Оптимізація процесу маршрутизації в розподілених мережах з використанням машинного навчання", Інфокомунікаційні технології та електронна інженерія, 2025, No. 5, С. 64–74. DOI: <https://doi.org/10.23939/ict2025.01.064>.
2. Zheng, X., Huang, W., Li, H., Li, G. (2022), "Research on Generalized Intelligent Routing Technology Based on Graph Neural Network", Electronics, No. 11, 2952. DOI: <https://doi.org/10.3390/electronics11182952>.
3. Hama Amin, R.R. (2025), "Intelligent Optimization of OSPF Path Selection Using Machine Learning Models for Adaptive Network Routing", Kurdistan Journal of Applied Research, No. 10(2), P. 31–42. DOI: <https://doi.org/10.24017/science.2025.2.3>.
4. Jiang, W., Han, H., Zhang, Y., Wang, J., He, M., Gu, W., Mu, J., Cheng, X. (2024), "Graph Neural Networks for Routing Optimization: Challenges and Opportunities", Sustainability, No. 16, 9239. DOI: <https://doi.org/10.3390/su16219239>.
5. Колесніков, К., Карпетян, А., Баган, В. (2016), "Аналіз результатів дослідження реалізації задачі маршрутизації на основі нейронних мереж та генетичних алгоритмів", Вісник Черкаського державного технологічного університету, No. 21(1), С. 28–34.
6. Лемешко, О. В., Єременко, О. С., Невзорова, О. С. (2020), Потоківі моделі та методи маршрутизації в інфокомунікаційних мережах: відмовостійкість, безпека, масштабованість, Харків: ХНУРЕ, 308 с. DOI: <https://doi.org/10.30837/978-966-659-282-1>.

7. Liang, K., Myers, M. (2021), "Machine Learning Applications in the Routing in Computer Networks", arXiv preprint, P. 1–14. DOI: <https://doi.org/10.48550/arXiv.2104.01946>.
8. Pawar, L., Bajaj, R. and Anand, D. (2025), "Resource Allocation Load Balancing Secure Schemes in Software Defined Networks: An Analytical Review", International Journal of Computer Networks and Applications, No. 12, P. 523–537. DOI: <https://doi.org/10.22247/ijcna/2025/32>.
9. Craveirinha, J., Clímaco, J., Girão-Silva, R., Pascoal, M. (2024), "Multiobjective Path Problems and Algorithms in Telecommunication Network Design—Overview and Trends", Algorithms, No. 17, 222. DOI: <https://doi.org/10.3390/a17060222>.
10. Reis, J., Rocha, M., Phan, T.K., Griffin, D., Le, F. and Rio, M. (2019), "Deep Neural Networks for Network Routing", Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, P. 1–8. DOI: <https://doi.org/10.1109/IJCNN.2019.8851733>.
11. Пахомова, В., Хрестян, А. (2025), "Дослідження двох підходів щодо прогнозування затримки на маршрутизаторі комп'ютерної мережі залізничного транспорту з використанням нейромережної технології", Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки, No. 2(1), P. 184–190. URL: <https://doi.org/10.32782/2663-5941/2025.1.2/27>.
12. Shtangey, S., Melnikova, L., Marchuk, A., Linnyk, O. and Sokolov, O. (2025), "Modeling and analysis of graph neural networks for optimizing routing in infocommunication networks", Management Information System and Devices, No. 1(186), P. 40–54. DOI: <https://doi.org/10.30837/0135-1710.2025.186.040>.
13. Лемешко, О. В., Єременко, О. С., Євдокименко, М. О., Шаповалова, А. С., Слейман, Б. (2022), Моделювання та оптимізація процесів безпечної та відмовостійкої маршрутизації в телекомунікаційних мережах: Монографія. Х.: ХНУРЕ, 198 с. DOI: <https://doi.org/10.30837/978-966-659-378-1>.
14. Singh, S., Roushan, A., Mehta, D., Agarwal, P., Singh, B. (2025), "Advanced Networking: Dynamic Hybrid Routing Protocol Empowered by Machine Learning for Enhanced Security and Fault Detection", In: Yadav, A., Joshi, A.M., Ergezer, M., Balas, V.E. (eds) Artificial Intelligence and Applications. ICAIA 2024. Algorithms for Intelligent Systems, Springer, Singapore, P. 359–371. DOI: https://doi.org/10.1007/978-981-97-8074-7_26.
15. Yang, S., Tan, C., Madsen, D.Ø., Xiang, H., Li, Y., Khan, I., Choi, B.J. (2022), "Comparative analysis of routing schemes based on machine learning", Mobile Information Systems, No. 2022(1), p.4560072. DOI: <https://doi.org/10.1155/2022/4560072>.